# Proceedings of the 2007 International Conference on Computational and Mathematical Methods in Science and Engineering

**Illinois Institute of Technology**
**Chicago, Illinois, USA**
**20-23, June, 2007**

CMMSE
Computational and Mathematical
Methods in Science and Engineering

**Editor:**
Bruce A. Wade

**Associate Editors:**
Greg Fasshauer, Abdul Q.M. Khaliq, Jesús Vigo Aguiar

# Proceedings of the 2007 International Conference on Computational and Mathematical Methods in Science and Engineering

**Illinois Institute of Technology**
**Chicago, Illinois, USA**
**20-23, June, 2007**

G. Fasshauer, A.Q.M. Khaliq, J. Vigo Aguiar, B. A. Wade, eds.

## Preface

We are honored to bring you this collection of articles and extended abstracts from the *Seventh International Conference on Computational and Mathematical Methods in Science and Engineering* (CMMSE 2007), held at the Illinois Institute of Technology, Chicago, Illinois, USA, June 20-23, 2007. The primary focus of CMMSE is on new ideas and interdisciplinary interaction in rapidly growing fields of computational mathematics, mathematical modeling, and applications.

CMMSE 2007 special sessions represent advances in financial mathematics and engineering, industrial mathematics, computational spectral theory, multiscale modeling and high performance computing in mathematical biology, algorithms and computation for complex networks, and novel finite difference and hybrid methods for ordinary and partial differential equations.

We would like to thank the plenary speakers for their excellent contributions in research and leadership in their respective fields. We express our gratitude to the special session organizers, who have been a very important part of the conference, and, of course, to all participants.

Chicago, Illinois, USA, June 20, 2007

G. Fasshauer, A.Q.M.Khaliq, J. Vigo Aguiar, B.A. Wade

# Acknowledgements

We are indebted to many people who have helped with the conference:

### CMMSE 2007 Plenary Speakers

M. Anitescu, Argonne National Laboratory, USA
H.T. Banks, North Carolina State University, USA
E. Brändas, Uppsala Universitet, Uppsala, Sweden
P. Forsyth, University of Waterloo, Canada
Y. Jiang, Los Alamos National Laboratory, USA
G. Papanicolaou, Stanford University, USA
J. Pasciak, Texas A & M University, USA

### CMMSE 2007 Special Session Organizers

H.T. Banks, North Carolina State University, USA
R. Bawa, Punjabi University, India
T. R. Bielecki, Illinois Institute of Technology, USA
S. Boccaletti, Istituto Nazionale di Ottica Applicata, Italy
J. Burns, Virginia Tech, USA
B. Chanane, King Fahd University of Petroleum and Minerals, Saudi Arabia
R. Criado, Universidad Rey Juan Carlos, Spain
S. Damelin, Georgia Southern University, USA
R.K. Dash, Medical College of Wisconsin, USA
J. Davis, Baylor University, USA
F. Hickernell, Illinois Institute of Technology, USA
A.Q.M. Khaliq, Middle Tennessee State University, USA
I. Lauko, University of Wisconsin-Milwaukee, USA
J. Martín Vaquero, Universidad de Salamanca, Spain
M. Siddique, Virginia Union University
Z. Sinkala, Middle Tennessee State University, USA
Q. Sheng, Baylor University, USA
J. Vigo Aguiar, Universidad de Salamanca, Spain
D. Xie, University of Wisconsin-Milwaukee, USA

ISBN 978-84-690-6887-8

# Contents

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE2007*
*Chicago, 20 - 23 June 2007*

# A Theory of Non-Gaussian Option Pricing

Gil Adams, Michael Kelly
*Illinois Institute of Technology, Stuart School of Business*

emails : giladams@stuart.iit.edu, mkelly@stuart.iit.edu

## Abstract

The Black - Scholes model has been the option pricing standard for three decades and continues to be even with its acknowledged deficiencies. One such deficiency is its dependence on Gaussian distributions. This article uses *Mathematica*'s symbolic programming language to implement an alternative model, one similar in structure to Black - Scholes, but based upon a more realistic non - Gaussian distribution.

Key Words : Non - Gaussian, Black - Scholes, Options, Entropy, Tsallis, Borland

## ■ 1. Introduction

In the literature of financial instruments and especially of option pricing, the Black-Scholes model [1] is recognized as the basis of the interlocking theory that ties together no-arbitrage risk-neutral pricing with stochastic calculus applied to risky assets and with the reduction of the partial differential equation (PDE) for financial derivatives to the well known heat equation. This intersection of the established fields, mathematics and statistics, has given a well justified and prominent position to the Black-Scholes pricing of European options. However like all models, the Black-Scholes is based upon assumptions which represent idealizations that do not apply well to all the markets. The two most important of these assumptions are: first, the invariance of the volatility $\sigma$ or annualized standard deviation of the returns of the underlying equity. The second assumption is the stochastic component of the equity, often called the Wiener process coefficient. There has therefore been a strong need for a new model capable of better representing the observed values and higher option prices.

It has long been known [2] that the volatility is stochastic and that changes in its value can significantly alter pricing. In those markets where the volatility is especially high, such as the Nasdaq 100 index [www.Nasdaq.com] and technology stocks, the Black-Scholes formula leads to significant underpricing. One traditional method [3] of dealing with this has been to introduce another stochastic equation for the volatility that involves further stochastic calculus difficulties. The method introduced here is to change the volatility coefficient or Wiener process $\omega$. The assumption of lognormality in equity prices has led to the adoption of the Gaussian distribution $\mathcal{N}$ for the Wiener process. However the very fat tails at either extreme of the stock distribution suggest non-Gaussian distributions. There have been many attempts at replacing the Gaussian distribution with T distributions [4 & 5], stable distributions [6] and other fractal measures [7]. Based upon earlier work in the application of maximum entropy [8] to financial evaluation, it is apparent that *entropy* can generate non-Gaussian distributions. Entropy is a measure of the missing information in the stochastic behavior of a market variable. The work described here extends this notion by replacing the noise process with a generalized Wiener process governed by a non-Gaussian fat-tailed Tsallis distribution of index q>1 associated with the Tsallis non-extensive entropy.

For the purposes of exposition we utilize the same notation and functional definitions as occur in the original work of Tsallis [9] and Borland [10]. For consistency we replicate these definitions using the *Mathematica* code.

## 2. Tsallis Entropy

In order to describe Tsallis or non-extensive entropy we first need to define *extensivity*. Given two independent systems $A$ and $B$, for which the joint probability density satisfies

$$p(A,B) = p(A)\,p(B)$$

the Tsallis entropy of this system satisfies

$$H_q(A, B) = H_q(A) + H_q(B) + (1-q)\,H_q(A)\,H_q(B) \tag{2.1}$$

From this result, it is evident that the parameter $q$ is a measure of the departure from extensivity. For an extensive system we take the limit as $q$ –> 1,

$$H(A, B) = H(A) + H(B)$$

The Tsallis entropy $H_q(p)$ is a generalization of the standard Boltzmann-Gibbs entropy $H(p)$ put forward by Constantino Tsallis in 1988 [9]. These different versions of entropy are defined as

$$\begin{aligned} H(p) &= -\int p(x)\,dx \\ H_q(p) &= \frac{1-\int p^q(x)\,dx}{(q-1)} \end{aligned} \tag{2.2}$$

Where $p(x)$ denotes the probability distribution of the underlying asset at maturity and $q$ is the real parameter associated with non-extensivity. In the limit as $q$ –> 1, the normal Boltzmann-Gibbs entropy is recovered. In this paper q is shown to have a financial interpretation consistent with different levels of volatility and different markets.

## 3. Borland Model

The purpose of this article is to expand upon and implement in *Mathematica* code the contents of Borland's "A theory of non-Gaussian option pricing" [10]. What follows in this section is a shortened version of the original paper, emphasizing the Feynman-Kac perspective. The standard model for stock movement is

$$dS_t = \mu\,S_t\,dt + \sigma\,S_t\,d\omega,\ t \geqslant 0 \tag{3.1}$$

$S_t$ represents the value of a stock $S$ at time $t$, its mean rate of drift by $\mu$ and the returns' variance by $\sigma^2$. The $\omega$ represents a zero-mean Gaussian process with variance $t$, i.e. $\mathbb{E}\left[(d\,\omega)^2\right] = d\,t$. When $\mu$ and $\sigma$ are linear functions, the Feynman-Kac formula yields exact solutions for *functions* of $S_t$. Since $\mu$ and $\sigma$ are constants here then the Feynman-Kac formula is applicable to option pricing as demonstrated in the next section 4.
The Borland model for stock movement is

$$dS_t = \left(\mu + \frac{\sigma^2}{2}\,\mathrm{P}_q{}^{1-q}\,(\Omega(t))\right) S_t\,dt + \sigma\,S_t\,d\Omega$$

$$dS_t = \left(\mu + \frac{\sigma^2}{2}\,\mathrm{P}_q{}^{1-q}\,(\Omega(t))\right) S_t\,dt + \sigma\,S_t\,\mathrm{P}_q{}^{\frac{1-q}{2}}(\Omega(t))\;d\omega \tag{3.2}$$

with the symbols as above. The driving noise $\Omega$ is a non-Gaussian statistical feedback process

$$d(\Omega(t)) = \mathrm{P}^{\frac{1-q}{2}}(\Omega(t))\;d\omega$$

Clearly, it is heavily reliant on the Gaussian process $\omega$. When $q = 1$, $\Omega$ reduces to $\omega$ and the standard process is recovered. To more closely match actual returns we restrict $q > 1$. This lower limit insures fat tails under the Tsallis distribution. The upper limit, $q < 5/3$, keeps the variance $\mathbb{E}\left[\Omega^2(t)\right] = \frac{1}{(5-3\,q)\,\beta(q,t)}$ convergent. Using the definition of c(q) found in [13], the Tsallis probability is now given by

$$\begin{aligned}
&\mathrm{P}_q{}^{1-q}\,(\Omega(t)) = \frac{\left(1-(1-q)\,\beta(q,t)\,(\Omega(t))^2\right)^{\frac{1}{1-q}}}{Z(q,t)}\\
&\text{with}\; \beta(q,\,t) \;=\; c(q)^{\frac{1-q}{3-q}}\,((2-q)\,(3-q)\;t)^{\frac{-2}{3-q}}\\
&Z(q,\,t) \;=\; ((2-q)\,(3-q)\;c(q)\,t)^{\frac{1}{3-q}}\\
&c(q) \;=\; \frac{\pi}{q-1}\,\left(\Gamma\left[\frac{1}{q-1}\;-\;\frac{1}{2}\right]\right)^2\Big/\left(\Gamma\left[\frac{1}{q-1}\right]\right)^2
\end{aligned} \tag{3.3}$$

Applying Feynman-Kac to the non-Gaussian model means modifying equation (3.2) into a form similar to (3.1). This requires the changing of a variable, a transformation of the discounted stock price into a martingale via a change in measure, a stochastic integration and the conversion back to S in the original measure.

Beginning with the discounted stock price

$$\begin{gathered}
\mathrm{G} = \mathrm{S}\;\mathbb{e}^{-\mathrm{r}\,\mathrm{t}}\\
\text{where}\;\; dG \;=\; \left(\mu\;-r\;+\;\sigma^2\,\frac{\mathrm{P}_q{}^{1-q}}{2}\right)G\,dt\;+\sigma\,G\,d\Omega = \sigma\,G\,\mathrm{P}_q{}^{1-q}\;dz\\
\text{with}\,dz \;=\; \frac{\left(\mu-r+\sigma^2\;\frac{\mathrm{P}_q{}^{1-q}}{2}\right)}{\sigma\,P_q{}^{\frac{1-q}{2}}}\,dt\;+\;d\omega
\end{gathered} \tag{3.4}$$

$$d\ln G \;=\; \frac{\sigma^2}{2}\,P_q{}^{1-q}\,dt + r\,P_q{}^{1-q}\,dz \tag{3.5}$$

Taking the stochastic integration of equation (3.5) using the Radon-Nikodym derivative where $\mathrm{P}_q$ is a function of $\Omega$, which in turn depends upon $S_t$, which is represented in terms of G in equation (3.4) yields

$$\begin{gathered}
S_t = \;S(0)\,\mathrm{Exp}\left[\int_0^t\sigma\,\mathrm{P}_q{}^{\frac{1-q}{2}}(\Omega\,(z))\,dz\;+\;\int_0^t\left(r\;-\;\frac{\sigma^2}{2}\,\mathrm{P}_q{}^{1-q}(\Omega\,(x))\right)dx\right]\\
\text{with}\,\alpha = \frac{1}{2}\,(3-q)\,((2-q)\,(3-q)\;c)^{\frac{q-1}{3-q}}\\
\text{and}\;\;\Omega(t)\;=\int_0^t\mathrm{P}_q{}^{\frac{1-q}{2}}(\Omega\,(z))\,dz
\end{gathered}$$

and $\mathrm{P}_q\,(\Omega\,(\tau))$ at an arbitrary time $\tau$ can be mapped onto $\mathrm{P}_q\,(\Omega\,(T))$ at a fixed time T via the appropriate variable transformation $\Omega(\tau) \;=\; \sqrt{\frac{\beta(T)}{\beta(\tau)}}\;\Omega(T)$. Then

$$\begin{aligned}
&S_T \;=\; S(0)\,\mathrm{Exp}\left[\sigma\,\Omega\,(T) +\; r\,T - \frac{\sigma^2}{2}\,\alpha\,T^{\frac{2}{3-q}} +\;(1-q)\,\frac{\sigma^2}{2}\int_0^T\frac{\beta(t)}{Z(t)^{1-q}}\,\Omega^2(t)\,dt\right]\\
&\text{where}\,\alpha\;=(3-q)\,((2-q)\,(3-q)\,c)^{\frac{q-1}{3-q}}\Big/2\;,\;\;\text{Hence}
\end{aligned}$$

$$S_T \;=\; S(0)\,\mathrm{Exp}\left[\sigma\,\Omega\,(T) +\; r\,T\;-\frac{\sigma^2}{2}\,\alpha\,T^{\frac{2}{3-q}}\left(1-\;(1-q)\,\beta(T)\,\Omega^2(T)\right)\right] \tag{3.6}$$

Note that once again, when $q = 1$ the standard model is recovered. Note that there is a misprint in [10] so that the first occurrence of $\sigma$ was missing from the two equations above.

# ■ 4. Similarities Between the Black-Scholes and Borland Models

## □ Solutions to Stochastic Differential Equations

### Feynman-Kac Solution to Black-Scholes

In the Black-Scholes version of equation (3.1) $\mu$ can be replaced by the risk-neutral interest rate $r$, so that we now have

$$dS_t = \sigma S_t \, d\omega + r S_t \, dt\,, \quad t \geqslant 0 \tag{4.1}$$

Here we implement Lyasoff's [11] version of the Feynman-Kac option pricing formula. The European option as a function of $S_t$ described by equation (4.1) can be written as the integral representation of

$$\text{OptionPrice}\,[t,\, S_t,\ K] \ = \ \mathbb{E}\left[e^{-rt}\,\text{Max}\,[S_t\ -\ K,\, 0]\ \middle|\ S_0\ =\ S\ (0)\right],\ t\ \geq\ 0, \tag{4.2}$$

$$\text{where } S_t = S_0\, e^{\left(r-\frac{1}{2}\sigma^2\right)t\,+\,\sigma W_t}\ ,\quad t \geqslant 0 \tag{4.3}$$

With $S_t$ equal to the stock price and $K$ equal to the strike price, then for a Black-Scholes European call option on a non-dividend paying asset the payoff function is Max[$S_t - K$, 0] and using equation (4.3) we can write this simply in *Mathematica* code as

$$\texttt{BSCall[t\_, s\_, k\_] :=}$$
$$\frac{e^{-r\,t}}{\sqrt{2\,\pi}}\ \texttt{NIntegrate}\left[\texttt{Max}\left[s * e^{\left(r-\frac{\sigma^2}{2}\right)t\,+\,\sigma\sqrt{t}\;y} - k,\ 0\right] * e^{-\frac{y^2}{2}},\right.$$
$$\{y,\ -\infty,\ \infty\}\ ,\ \ \texttt{Method} \to \texttt{GaussKronrod},\ \texttt{MaxRecursion} \to 15,$$
$$\left.\texttt{WorkingPrecision} \to 20,\ \texttt{PrecisionGoal} \to 10\ \right];$$

For the specific set of values below, the European Black-Scholes call on a non-dividend paying stock is

```
So = 50; K = 40; r = 6 / 100; T = 6 / 10;
σ = 3 / 10;   BSCall[T, So, K] // N
```

```
12.091
```

### Feynman-Kac Solution to Borland

Since neither the call option nor Feynman-Kac place any restrictions on the stock price that would exclude the non-Gaussian model, we can apply equation (3.6) to equation (4.2).

$$\texttt{BorCall1[t\_, s\_, k\_] :=}\ \frac{e^{-\,r\,t}}{\texttt{Z[q, t]}}$$
$$\texttt{NIntegrate}\left[\texttt{Max}\left[s * e^{\left(\sigma\,\Omega\,+\,r\,t\ -\ \frac{\sigma^2}{2}\,\alpha\,t^{\frac{2}{3-q}}\,+\,(1-q)\ \alpha\,t^{\frac{2}{3-q}}\,\beta[q,t]\ \frac{\sigma^2}{2}\,\Omega^2\right)}\ -k,\ 0\right] *\right.$$
$$\left(1 - (1-q)\ \beta[q,\ t]\ \Omega^2\right)^{\frac{1}{1-q}},\ \{\Omega,\ -\infty,\ \infty\},\ \ \texttt{Method} \to \texttt{GaussKronrod},$$
$$\left.\texttt{MaxRecursion} \to 15,\ \texttt{WorkingPrecision} \to 20,\ \texttt{PrecisionGoal} \to 10\right];$$

For the set of values below, with $\alpha$ dependent on $q$, the European non-Gaussian call is:

```
So = 50; K = 40; r = 6 / 100; T = 6 / 10; σ = 3 / 10;
q = 1.15; α = 1.06307;   BorCall1[T, So, K] // N
```

```
12.214
```

Interchanging the strike and the stock prices in the payoff functions will exchange the call for a put option: PayOff[$S_t$, $K$] = Max[$K - S_t$, 0]. However, for the purposes of simplicity in this article, we choose to demonstrate only European call functions for underlying assets that pay no dividends as in [10]. For the Borland model, just as for the standard Gaussian one, modifying the results to include dividend-paying assets, with a dividend yield $\delta$, by replacing $r$ with $r-\delta$ is a simple matter.

It should be noted that when the $q \to 1$, equation (3.2) → equation (4.1) and equation (3.6) → equation (4.3). In other words as $q \to 1$, Tsallis distribution recovers normal distribution and Borland recovers the traditional Black-Scholes model. By example, for the specific set of values below, with $\alpha$ dependent on $q$ being very close to 1, the European Black-Scholes call and the European Borland call are equal.

```
So = 50; K = 40; r = 6 / 100; T = 6 / 10;
σ = 3 / 10; q = 1.000001; α = 1.0000;
TableForm[{{"BorlandCall", "BlackScholesCall"},
   {BorCall1[T, So, K], BSCall[T, So, K]}}] // N
```

```
        BorlandCall      BlackScholesCall
        12.091           12.091
```

□ Conversion to CDFs

### Black-Scholes

The Gaussian and the non-Gaussian pricing models can also be expressed in the form of the difference between CDFs of functions which correspond to the probabilities of the stock price being in and out of the money. For example, the standard Black-Scholes format draws on two ubiquitous financial functions: "done" and "dtwo". The normal CDF of *done* is related to the probability of the stock price being in the money. Shaw [12] codes the European Black-Scholes model for non-dividend paying assets very much like the following.

```
done[s_, σ_, k_, t_, r_] :=
  (r * t + Log[s / k]) / (σ * Sqrt[t]) + (σ * Sqrt[t]) / 2;
dtwo[s_, σ_, k_, t_, r_] := done[s, σ, k, t, r] - (σ * Sqrt[t]);
BlackScholesCall[s_, k_, σ_, r_, t_] :=
  s * η[done[s, σ, k, t, r]] - k * Exp[-r * t] * η[dtwo[s, σ, k, t, r]];
```

For the specific set of values below, the Black-Scholes European call on a non-dividend paying asset is

```
So = 50; K = 40; r = 6 / 100; T = 6 / 10;
σ = 3 / 10;  BlackScholesCall[So, K, σ, r, T] // N
```

```
        12.091
```

### Borland

The non-Gaussian model for a non-dividend asset draws upon two functions: "NQ1" and "MQ".

```
NQ1[D1_, D2_, q_, t_] :=
  NIntegrate[(1 - (1 - q) β[q, t] x^2)^(1/(1-q)), {x, D1, D2}] / Z[q, t];
MQ[α_, σ_, D1_, D2_, q_, t_] := With[{β1 = β[q, t]},
   NIntegrate[Exp[σ x - σ^2/2 α t^(2/(3-q)) (1 - (1 - q) β1 x^2)]
      (1 - (1 - q) β1 x^2)^(1/(1-q)), {x, D1, D2}] / Z[q, t]];
```

This is similar to Black-Scholes in concept and form.

```
BorCall3[s_, k_, σ_, r_, t_, q_] :=
  Module[{α, s1, s2}, α = (3 - q)/2 ((2 - q) (3 - q) c[q])^((q-1)/(3-q));
   s1 = S1[α, k, q, r, s, σ, t]; s2 = S2[α, k, q, r, s, σ, t];
   s MQ[α, σ, s1, s2, q, t] - ⅇ^(-r t) k NQ1[s1, s2, q, t] ];
```

For the specific set of values below, the non-Gaussian European call for a non-dividend paying asset is

```
So = 50; K = 40; r = 6 / 100; T = 6 / 10; σ = 3 / 10;
q = 1.15;   BorCall3[So, K, σ, r, T, q] // N
```

```
        12.214
```

We can reduce the NQ1 integral for better speed and more efficient execution. With $D_1$ and $D_2$ substituting as dummy values for the functions $S_1$ and $S_2$, the integral becomes

$$\text{NQ1}(D_1, D_2, q, t) = \frac{e^{-rT}\ K}{Z(q,T)} \int_{D_1}^{D_2} \left(1-(1-q)\,\beta(q,t)\,\Omega_t^{\,2}\right)^{\frac{1}{1-q}}\, d\Omega_t$$

Two assumptions must be met to achieve the desired results from the integration. The first is that $\sqrt{\beta - \beta\, q}$ 's imaginary part is not zero (dropping some arguments for clarity). Since $q$ is always greater than 1, the first assumption is verified. The second assumption requires the upper limit of integration to be larger than the lower limit, specifically $D_2 > D_1$. This assumption is always satisfied when $q$ is not equal to one and when the following parameters are not equal to zero: $\alpha$, $\beta$, $\sigma$, $r$ or $T$. Therefore

```
Integrate[(1 - (1 - q) B x^2)^(1/(1-q)), {x, D₁, D₂},
 Assumptions → { D₂ > D₁ , Im[√(B - B q)] ≠ 0}]
```

$$_2F_1\left(\frac{1}{2}, \frac{1}{q-1}; \frac{3}{2}; -B\,(q-1)\,D_2^2\right) D_2 - {_2F_1}\left(\frac{1}{2}, \frac{1}{q-1}; \frac{3}{2}; -B\,(q-1)\,D_1^2\right) D_1$$

Using the `Hypergeometric2F1` functions

```
NQ[D1_, D2_, q_, t_] := With[{β1 = β[q, t]},
   (D2 * Hypergeometric2F1[1/2, 1/(-1 + q), 3/2, -D2^2 (-1 + q) β1] - D1 *
       Hypergeometric2F1[1/2, 1/(-1 + q), 3/2, -D1^2 (-1 + q) β1]) / Z[q, t]];
BorlandCall[s_, k_, σ_, r_, t_, q_] :=
  Module[{α, s1, s2}, α = (3 - q)/2 ((2 - q) (3 - q) c[q])^((q-1)/(3-q));
   s1 = S1[α, k, q, r, s, σ, t]; s2 = S2[α, k, q, r, s, σ, t];
   s MQ[α, σ, s1, s2, q, t] - ⅇ^(-r t) k NQ[s1, s2, q, t]];
```

On average the hypergeometric version is faster:

```
So = 50; K = 40; r = 6 / 100; T = 6 / 10; σ = 3 / 10; q = 1.01;
TableForm[
 {{"Borland: Hypergeometric version", "Borland: NIntegrate version"},
  Mean[Table[{Timing[BorlandCall[##]], Timing[ BorCall3[##]]} & @@
     {So, K, σ, r, T, q}, {500}]]}]
```

```
Borland: Hypergeometric version   Borland: NIntegrate version
       0.00503 Second                0.008502 Second
          12.0986                       12.0986
```

## 5. The Option Greeks

A commonly used and powerful mathematical tool in assessing an option's risk is its sensitivity to changes either in market conditions or in the underlying asset itself. An arsenal of five distinct tools has been developed based upon these sensitivities. Four correspond to the first derivatives of the option with respect to: stock price (Delta), time (Theta), interest rates (Rho) and volatility (Vega). Theta actually is the negative of the first derivative with respect to time. Vega is commonly referred to as either Lambda or Kappa. The fifth, Gamma, is calculated as the second derivative with respect to the stock price. Collectively these five are known as the *Greeks* for obvious reasons. Once again, we modified Shaw's [12] code to accommodate the case of non-dividend paying assets.

### The Black-Scholes Greeks

```
BlackScholesCallDelta[s_, k_, σ_, r_, t_] =
  D[BlackScholesCall[s, k, σ, r, t], s];
BlackScholesCallTheta[s_, k_, σ_, r_, t_] =
  - D[BlackScholesCall[s, k, σ, r, t], t];
BlackScholesCallRho[s_, k_, σ_, r_, t_] =
  D[BlackScholesCall[s, k, σ, r, t], r];
BlackScholesCallVega[s_, k_, σ_, r_, t_] =
  D[BlackScholesCall[s, k, σ, r, t], σ];
BlackScholesCallGamma[s_, k_, σ_, r_, t_] =
  D[BlackScholesCall[s, k, σ, r, t], {s, 2}];
```

For a specific set of values, the Greeks associated with the European Black-Scholes call options are

```
So = 50; K = 40; r = 6 / 100; T = 6 / 10; σ = 299 / 1000;
TableForm[{{"Gaussian", "Distribution"},
   greeks = {"Delta", "Theta", "Rho", "Vega", "Gamma"},
   Through[ToExpression[StringJoin["BlackScholesCall", #] & /@
       greeks][So, K, σ, r, T]]}] // N
```

```
Gaussian   Distribution
 Delta        Theta        Rho       Vega      Gamma
0.89153     −3.74597    19.4956   7.20964   0.016075
```

### The Borland Greeks

A similar arsenal has been created for the non-Gaussian model. In addition Borland [10] defines a “ newGreek" - *Upsilon* - as the option's first derivative with respect to *q*.

```
BorlandCallDelta[s_, k_, σ_, r_, t_, q_] =
  D[BorlandCall[s, k, σ, r, t, q], s];
BorlandCallVega[s_, k_, σ_, r_, t_, q_]  =
  D[BorlandCall[s, k, σ, r, t, q], σ];
BorlandCallTheta[s_, k_, σ_, r_, t_, q_]  =
  - D[BorlandCall[s, k, σ, r, t, q], t];
BorlandCallRho[s_, k_, σ_, r_, t_, q_]  =
  D[BorlandCall[s, k, σ, r, t, q], r];
BorlandCallGamma[s_, k_, σ_, r_, t_, q_] =
  D[BorlandCall[s, k, σ, r, t, q], {s, 2}];
BorlandCallUpsilon[s_, k_, σ_, r_, t_, q_]  =
  D[BorlandCall[s, k, σ, r, t, q], q];
```

For a specific set of values, the Greeks associated with the European non-Gaussian call options are

```
So = 50; K = 40; r = 6 / 100; T = 6 / 10; σ = 299 / 1000; q = 1.01;
TableForm[{{"Non-Gaussian", "Distribution"},
   greeks = {"Delta", "Theta", "Rho", "Vega", "Gamma", "Upsilon"},
   Through[ToExpression[StringJoin["BorlandCall", #] & /@ greeks][
     So, K, σ, r, T, q]]}] // N
```

```
Non−Gaussian   Distribution
    Delta          Theta        Rho       Vega      Gamma      Upsilon
  0.891386      −3.76197     19.4868   7.24104   0.0160282   0.757974
```

## 6. Graphical Results - The Options and Their Greeks

The results in this article are visual and fall into two main categories. The first deals with the options themselves and their relationship to various parameters: strikes, expiration time, volatility and $q$, etc. The images in the first section generally compare the standard Gaussian model against the non-Gaussian model. The second category deals primarily with the traditional five Greeks and Upsilon as defined in section 5.

### The Options

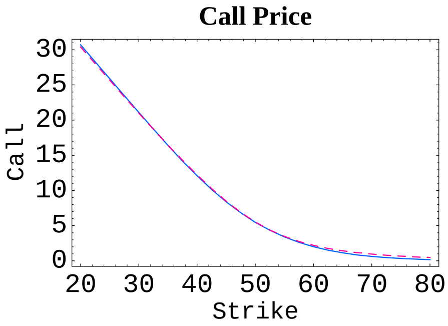


**Figure 1** demonstrates two sets of Calls over a range of Strike Prices, one generated by Black-Scholes and one by the non-Gaussian model. S(0), abbreviated as $S_0$, is set to 50, $r = 0.06$ and $T = 0.6$. For the B-S case (solid blue) $q = 1$ and $\sigma = 0.3$. For the non-Gaussian case (dashed red) $q = 1.5$ and $\sigma = 0.299$.

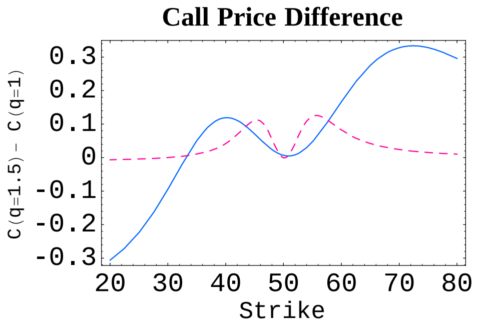


**Figure 2** demonstrates two sets of Call Price Differences over a range of Strike Prices: Borland - BlackScholes. $S_0 = 50$, $r = 0.06$. The solid blue line represents time $T = 0.6$. Borland is calculated using $q = 1.5$ and $\sigma = 0.297$; BS is calculated using $\sigma = 0.3$. For the dashed red line, time $T = 0.05$, Borland using $q = 1.5$ and $\sigma = 0.41$; BS using $\sigma = 0.3$.

The image directly above demonstrates that the fatter tails of a Tsallis distribution ($q = 1.5$)

allows for greater probabilities of a call going either deep in-the-money or deep out-of-the-money.

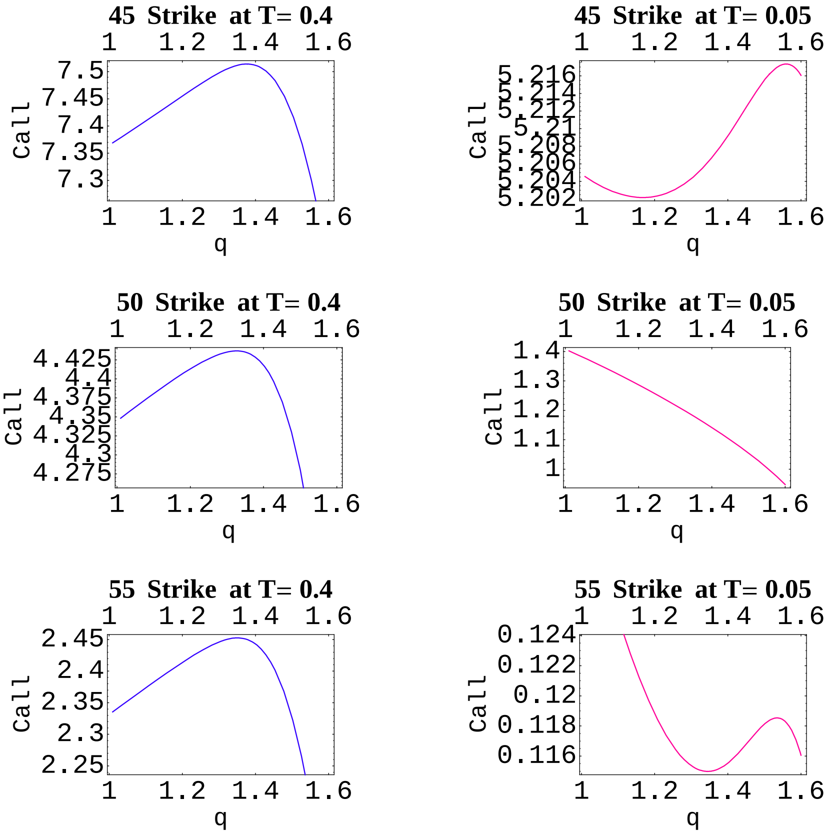


**Figure 3** demonstrates Call Options versus $q$ for two different expiry times and for three distinct strikes. The left column is calculated with $t = 0.06$, the right column with $t = 0.05$. The three rows represent $K = 45$ (in-the-money), $K = 50$ (at-the-money) and $K = 55$ (out-of-the-money), all with $S_0 = 50$, $r = 0.06$ and $\sigma = 0.299$.

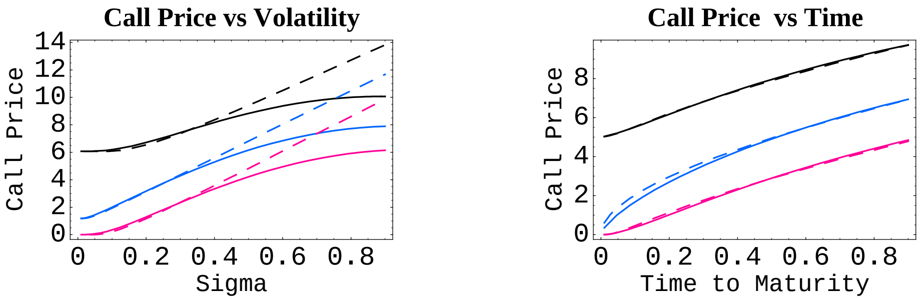


**Figure 4** demonstrates Call Prices vs Volatility and Call Prices vs Time-to-Maturity for three strikes. Black lines represent in-the-money calls (top), $K = 45$. Blue represents at-the-money (middle), $K = 50$. Red represents out-of-the-money (bottom), $K = 55$ all with $S_0 = 50$ and $r = 0.06$. Solid lines represent Black-Scholes. Dashed lines represent Borland with $q = 1.5$.

### The Greeks

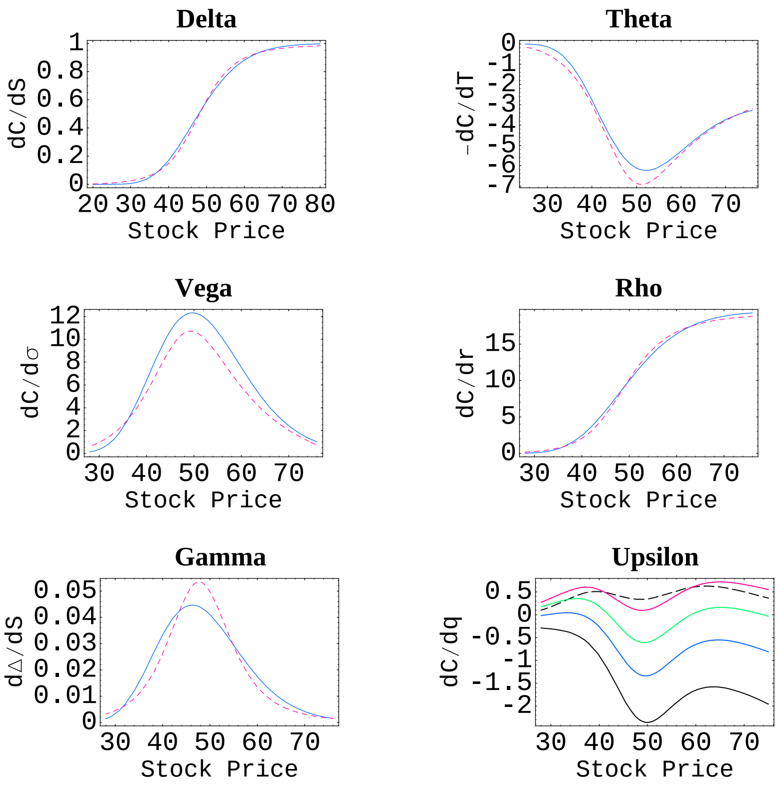


**Figure 5** demonstrates the five traditional Greeks - Delta $\left(\Delta = \frac{\delta c}{\delta S}\right)$,Theta $\left(\theta = -\frac{\delta c}{\delta T}\right)$, Vega $\left(V = \frac{\delta c}{\delta \sigma}\right)$, Rho $\left(\rho = \frac{\delta c}{\delta r}\right)$ and Gamma $\left(\Gamma = \frac{\delta \Delta}{\delta S}\right)$. Solid blue lines represent Black-Scholes and dashed red lines represent Borland with $q = 1.5$. For the "new" Greek Upsilon $\left(\Upsilon = \frac{\delta c}{\delta q}\right)$ the dashed line represents Borland with $q = 1.1$. The solid lines represent, in order, Borland with $q = 1.3$ (red), $q = 1.4$ (green), $q = 1.45$ (blue) and $q = 1.5$ (black). All are calculated over a range of Stock Prices with $K = 50$, $r = 0.06$, $T = 0.4$ and $\sigma = 0.300$

## 7. Conclusion

It has long been observed that the Black-Scholes model, while adequate in describing stable markets, is no longer reliable for highly volatile markets and that it has become necessary to introduce additional stochastic volatility models to account for this. Here we offer another solution that posits the observed variability as a function of the non-extensive parameter $q$. This additional parameter allows an explanation of the moneyness bias in modern markets. Furthermore the need for proper hedging in volatile markets is often poorly modeled by traditional measures of the greeks. Here we show that values of $q \neq 1$ can accommodate the extreme sensitivity for option prices close to maturity and the strike value by providing greeks that reflect this observed responsiveness. Lastly both the option prices and their greeks can be determined rapidly using Mathematica's `Integrate[]` function.

## References

[1] F. Black and M. Scholes, *The Pricing of Options and Corporate Liabilities*, Journal of Political Economy, **81**, 1973 pp. 637– 659.

[2] J.C.Hull and A.White, *The Pricing of Options on Assets with Stochastic Volatility*, Journal of Finance, **42**( June), 1987 pp. 281-300.

[3] Alan L.Lewis, *Option Valuation Under Stochastic Volatility with Mathematica Code*, 2nd Edition, Finance Press, California USA, 2005.

[4] K. Fergusson and E. Platen, *On the Distributional Characterisation of Daily Log-Returns of a World Stock Index*, Applied Mathematical Finance, **13**(1), 2006 pp. 19-38.

[5] S.Hurst, *The characteristic function of the Student T distribution*, Mathematical Sciences Institute, Financial Mathematics Research Reports, FMRR95-006, 1995, http://wwwmaths.anu.edu.au/research.reports/fmrr/95/

[6] D. Edelman, *Natural Generalisation of Black-Scholes in the Presence of Skewness, Using Stable Distributions*, Abacus, **31**(1), 1995 pp. 113-119.

[7] R.J. Elliott and J. van der Hoek, *A general Fractional White Noise Theory and Applications to Finance*, Mathematical Finance, **13**, 2003 pp. 301-330.

[8] P.Buchen and M. Kelly, *The Maximum Entropy Distribution of an Asset Inferred from Option prices*, Journal of Financial and Quantitative Analysis, **31**(1), 1996 pp. 143-159.

[9] C. Tsallis, *Nonextensive Statistics: Theoretical, Experimental and Computational Evidences and Connections*, Brazilian Journal of Physics, **29**, 1, March 1999.

[10] L. Borland, *A theory of non-Gaussian option pricing*, Quantitative Finance, **2**, 2002 pp. 415-431.

[11] A. Lyasoff, *Path Integral Methods for Parabolic Partial Differential Equations with Examples from Computational Finance*, The Mathematica Journal, **9**(2), 2004 pp. 399-422.

[12] W. T. Shaw, *Modelling Financial Derivatives with Mathematica*, Cambridge University Press, 1998.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# Iterative Refinement for Neville Elimination

**Pedro Alonso**[1], **Jorge Delgado**[1], **Rafael Gallego**[1] **and Juan Manuel Peña**[2]

[1] *Department of Mathematics, University of Oviedo, Spain*

[2] *Department of Mathematics, University of Zaragoza, Spain*

emails: `palonso@uniovi.es`, `delgadojorge@uniovi.es`, `rgallego@uniovi.es`, `jmpena@unizar.es`

### Abstract

We provide a sufficient condition for the convergence of iterative refinement using Neville elimination.

*Key words: Iterative refinement, Neville elimination, Total positivity*
*MSC 2000: 65F05, 65F10*

## 1 Introduction

Let us consider a linear system of equations $Ax = b$, with $A \in \mathbb{R}^{n \times n}$. Then, solving this system with some direct method, in floating point arithmetic, we get an approximation $\widehat{x}^{(0)}$ to the solution. Iterative refinement is a well established and studied technique to improve the accuracy of the computed solution $\widehat{x}^{(0)}$ of the linear system $Ax = b$. This process can be summarized in the following algorithm

**Algorithm** Iterative refinement

```
Input $A$, $b$, $nTol$, $xTol$
Compute an approximation to the solution of $Ax = b$: $\widehat{x}^{(0)}$
$k = 0$; $x^{(-1)} = \infty$
While $k \leq nTol$ and $\|\widehat{x}^{(k)} - \widehat{x}^{(k-1)}\| \geq xTol$ do
    Compute the residual $r^{(k)} = b - A\widehat{x}^{(k)}$
    Solve the system $A\,y^{(k)} = r^{(k)}$: $\widehat{y}^{(k)}$
    Update the solution $\widehat{x}^{(k+1)} = \widehat{x}^{(k)} + \widehat{y}^{(k)}$
    $k = k + 1$
End-While
Output $\widehat{x}^{(k)}$
```

It is necessary to compute the residual with extra precision to avoid the errors produced by cancellation of significant figures. For more details in this technique see, for example [12] and [7].

The usual method to solve a linear system of equations $Ax = b$ is Gaussian elimination. So, in the literature it has been considered the study of iterative refinement using Gaussian elimination from several points of view: convergence (see [7], [15], [18] and [19]), stability (see [17] and [12]) and error analysis (see [14] and [19]).

The main purpose of this work is to study the convergence of iterative refinement using Neville elimination. This method is an alternative procedure to Gaussian elimination to transform a square matrix $A$ into an upper triangular matrix $U$. Neville elimination makes zeros in a column of the matrix $A$ by adding to each row a multiple of the previous one. Here we only give a brief description of this procedure (for a detailed and formal introduction of it we refer to [10]). If $A \in \mathbb{R}^{n\times n}$, the Neville elimination procedure consists of at most $n-1$ steps:

$$A = A^{(1)} \to \widetilde{A}^{(1)} \to A^{(2)} \to \widetilde{A}^{(2)} \to \cdots \to A^{(n)} = \widetilde{A}^{(n)} = U.$$

On one hand, $\widetilde{A}^{(t)}$ is obtained from the matrix $A^{(t)}$ by moving to the bottom the rows with a zero entry in column $t$, if necessary, to get that

$$\widetilde{a}_{it}^{(t)} = 0, \quad i \geq t \quad \Rightarrow \quad \widetilde{a}_{ht}^{(t)} = 0, \quad \forall\, h \geq i.$$

On the other hand, $A^{(t+1)}$ is obtained from $\widetilde{A}^{(t)}$ making zeros in the column $t$ below the main diagonal by adding an adequate multiple of the $i$th row to the $(i+1)$th for $i = n-1, n-2, \ldots, t$. If $A$ is nonsingular, the matrix $A^{(t)}$ has zeros below its main diagonal in the first $t-1$ columns. It has been proved that this process is very useful with totally positive matrices, sign-regular matrices and other related types of matrices (see [8] and [3]).

A real matrix is called totally positive if all its minors are nonnegative. Totally positive matrices arise in a natural way in many areas of Mathematics, Statistics, Economics, etc. Specially, their application to approximation theory and Computer Aided Geometric Design (CAGD) is of great interest. For example, coefficient matrices of interpolation or least square problems with a lot of representations in CAGD (the Bernstein basis, the B-spline basis, etc.) are totally positive. Some recent applications of such kind of matrices to CAGD can be found in [13], [5] and [16]. For applications of totally positive matrices to other fields see [8].

In [6], [11], [9] and [10] it has been proved that Neville elimination is a very useful alternative to Gaussian elimination when working with totally positive matrices. In addition, there are some studies that prove the high performance computing of Neville elimination (see [2]).

Then, taking into account the convenience of using Neville elimination with totally positive matrices and that, as far as we know, no study of the convergence of the iterative refinement through Neville elimination exists, the main goal of this work is to perform that task.

Let $A$ be a $n \times n$ nonsingular matrix, in [1] it has been proved that the computed solution $\widehat{x}$ of $Ax = b$ by Neville elimination satisfies

$$(A + H)\widehat{x} = b, \tag{1}$$

with $H$ verifying different bounds depending on the matrix $A$. Considering (1) and the system

$$(A + H_k)\widehat{y}^{(k)} = r^{(k)}, \tag{2}$$

we will study the convergence of the iterative refinement. We will prove that in the general case the procedure converges if

$$\|H_k\| \leq \frac{1}{2\|A^{-1}\|}. \tag{3}$$

In the case that $A$ is a totally positive matrix and taking into account that it can be proved that

$$\|H\| \leq \frac{61}{16}\gamma_n \|A\|, \tag{4}$$

with $\gamma_n := \dfrac{nu}{1 - nu}$, where $u$ is the unit of roundoff, we deduce that the following condition on $A$ ensures the convergence of the iterative refinement using Neville elimination:

$$\frac{61}{16}\gamma_n \|A\| \, \|A^{-1}\| < \frac{1}{2}. \tag{5}$$

We point out that (4) is of the same kind as the bounds obtained by de Boor and Pinkus in [4] for Gaussian elimination.

The bound (5) depends on the condition number of the matrix $A$ as every equivalent bound corresponding to Gaussian method. But, in contrast to most of the equivalent bounds for this method, our bound does not depend on the growth factor of the elimination procedure.

## Acknowledgements

This work has been partially supported by the Spanish Research Grant MTM2006-03388 and under MEC and FEDER Grant TIN2004-05920.

## References

[1] P. Alonso, M. Gasca and J. M. Peña, *Backward Error Analysis of Neville Elimination*, Appl. Numer. Math. **23** (1997) 193–204.

[2] P. Alonso, R. Cortina, I. Díaz and J. Ranilla, *Neville Elimination: a Study of the Efficiency Using Checkerboard Partitioning*, Linear Algebra Appl. **393** (2004) 3–14.

[3] T. Ando, *Totally Positive Matrices*, Linear Algebra Appl. **90** (1987) 165–219.

[4] C. de Boor and A. Pinkus, *Backward Error Analysis for Totally Positive Linear Systems*, Numer. Math. **23** (1997) 193–204.

[5] J. Delgado and J. M. Peña, *Progressive Iterative Approximation and Bases with the Fastest Convergence Rates*, Comp. Aided Geom. D. **24** (2007) 10–18.

[6] J. Demmel and P. Koev, *The Accurate and Efficient Solution of a Totally Positive Generalized Vandermonde Linear System*, SIAM J. Matrix Anal. Appl. **27** (2005) 142–152.

[7] G. E. Forsythe and C. B. Moler, *Computer Solutions of Linear Algebraic Systems*, Prentince-Hall, Englewood Cliffs, NJ, 1967.

[8] M. Gasca and and C. A. Micchelli, eds., *Total Positivity and its Applications*, Kluwer Academic Publishers, Boston, 1996.

[9] M. Gasca and J. M. Peña, *Total positivity and Neville Elimination*, Linear Algebra Appl. **165** (1992) 25–44.

[10] M. Gasca and J. M. Peña, *A Matricial Description of Neville Elimination with Applications to Total Positivity*, Linear Algebra Appl. **202** (1994) 33–53.

[11] M. Gassó and J. R. Torregrosa, *A Totally Positive Factorization of Rectangular Matrices by the Neville elimination*, SIAM J. Matrix Anal. Appl. **25** (2004) 986–994.

[12] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[13] H. Lin, H. Bao and G. Wang, *Totally positive bases and progressive iteration approximation*, Comput. Math. Appl. **50** (2005) 575–586.

[14] C. B. Moler, *Iterative Refinement in Floating Point*, J. Assoc. Comput. Mach. **14** (1967), 316–321.

[15] J. M. Ortega, *Numerical Analysis A Second Course*, SIAM, Philadelphia, 1990.

[16] J. M. Peña, *Shape preserving representations in Computer Aided-Geometric Design*, Nova Science Publishers, Inc., New York, 1999.

[17] R. D Skeel, *Iterative Refinement Implies Numerical Stability for Gaussian Elimination*, Math. Comput. **35** (1980) 817–832.

[18] J. H. Wilkinson, *Errors Analysis of Direct Methods of Matrix Inversion*, J. Assoc. Comput. Mach. **8** (1961) 281-330.

[19] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Notes on Applied Science **32** (Her Majesty's Stationery Office, 1963).

*Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2007
Chicago, 20–23 June 2007.*

# A mathematical model of the static pantograph/catenary interaction

**Enrique Arias[1], Angelines Alberto[2], Tomás Rojo[1], Fernando Cuartero[1] and Jesús Benet[1]**

[1] *Escuela Poliécnica Superior de Albacete, University of Castilla-La Mancha*

[2] *Albacete Research Institute of Albacete, University of Castilla-La Mancha*

emails: `earias@dsi.uclm.es`, `Angelines.Alberto@uclm.es`, `trojo@dsi.uclm.es`, `fernando@dsi.uclm.es`, `Jesus.Benet@uclm.es`

### Abstract

In this paper, a mathematical model of the static pantograph/catenary interaction for high speed railways. Also, a High Performance Computing Algorithm associated to the model has been developed to obtain the solution of the static equilibrium equation of the pantograph/catenary system after an exhaustive study of the tradictionl mechanical approach based on a set of coupled strings. In order to obtain an adequate behaviour in the pantograph/catenary system, it is necessary the existence of adequate conditions in the line, and this requires a very precise mechanical calculus. The resulting stiffness matrix has a high sparsity degree. This circumstance can be exploited into two senses: less memory storage requeriments, and the use of suitable methods for solving the static equilibrium equation as projection methods.

*Key words: high performance computing, interaction pantograph/catenary, static equilibrium equation, sparse linear algebra libraries*

## 1 Introduction

The evolution in the transport market resulting from the globalization of the economy, the increasing deregulation of the markets, the competition on the base of a customer service, the growing environmental concerns, and the need to ensure a long term operational profitability consistent with prevailing economic reality are driving to a radical change (structural and cultural) in the railway sector, centered on innovative approaches to business and services, conducive to a future-oriented and market-driven posture in the global transportation marketplace.

The survival in the ever evolving and highly competitive transport market, implies a continuous search for the roots of excellence enabling railway operators and supplying industry to achieve a world-class profile. This quest for excellence has to cover the entire range of business activities, beginning with market demand and ending with customer satisfaction. It will entail the need for an integrated development and timely deployment of the adequate organizational, technological and skill infrastructures.

Thus, the fulfillment of this strategy needs new construction concepts, among them, the development of more performant coupled pantograph/ catenary systems and a wide utilization of new technologies.

Furthermore, the multitude of electrification systems currently in use throughout Europe and subsequently the massive capital investment that would be necessary in order to implement any sort of European wide harmonised solution, preclude to envisage any major change in this field in the foreseeable future. A detailed study of electrification systems and catenary/pantograph technology considering economic aspects may be found in [4].

During the recent last years, passengers transportation by railway has experienced a considerable increase in some European countries (Germany, France, Spain, ...). For that reason, reaching of higher velocities in railways has become a very important target. In that scenario, the pantograph/catenary system, with its dynamic behaviour, becomes a crucial component (see [5, 2, 3]), because at high speed it is very difficult to guarantee the permanent contact of the pantograph head and contact the wire, more over without the increasing of noise and wear.

In order to obtain an adequate behaviour in the pantograph/catenary system, it is necessary the existence of adequate conditions in the line, and this requires, among other aspects, a very precise mechanical calculus. Recent investigations have focused on dynamical behaviour by dynamical simulations in order to allow a better interaction of the pantograph and the catenary [3, 6]; in this paper we will follow a more traditional approach, focusing in the catenary, modeled, as usual, by a set of coupled strings.

The best conditions in which the pantograph would obtain electric energy from the line are when the contact wire is parallel to the ground, and then, an important problem is to determine the exact length of the droppers in order to allow the contact wire to acquire the correct shape. So, our objective is the development of a technique which allows us to implement a high precision calculation algorithm, and thus to develop a software tool to design high quality catenaries.

In this work, a High Performance Computing (HPC) Algorithm has been developed for solving the static equilibrium equation of the pantograph/catenary interaction in order to obtain good performances.

This paper is structured as follows. In Section 2, the catenary model is described. Section 3 introduces some aspects on High Performance Computing. Section 4 presents the standard linear algebra libraries BLAS and SPARSKIT. In section 5 the experimental results are presented. Finally, some conclusions and future work guidelines are outlined.

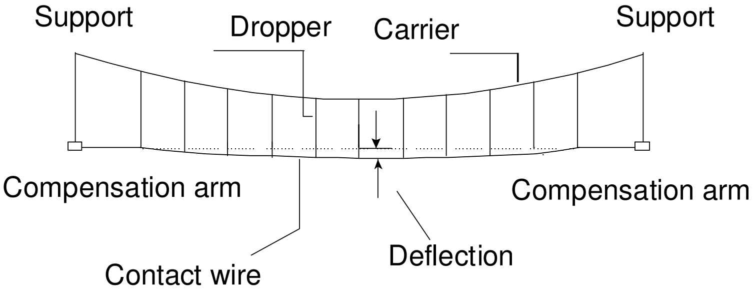


Figure 1: A span of catenary

## 2 The catenaty model

The conventional catenary electrification system is designed for heavy-traffic mainline operation and it is useful for train speeds well above 200 kph. For such high-speed operation an essentially constant contact force must be maintained between the overhead contact-wire and the locomotive's pantograph power-collecting apparatus.

As we have previously indicated, in this paper we will use a classical model of the catenary wire appearing in railways, so we consider the catenary composed from a reduced range of elements, such as carrier, droppers, contact wire and compensation arms (see Figure 1). The droppers are supporting the contact wire in order to obtain a horizontal line. The interval between two compensation arms will be called the span.

In this section an introduction to the problems related to the study of railways catenaries is outlined. After that, the modelization of the carrier, the contact wire, the droppers and the compensation arm is carried out. Finally, the static problem is defined.

### 2.1 Problems to consider in the mechanical study of railway catenaries

A span of catenary is composed by three types of cables (see Figure 1): the carrier, the droppers and the contact wire. The carrier is fixed at the supports, while the contact wire is upheld by the compensation arm.

In the mechanical study of the catenary system, three differents problems can be considered:

- **The calculation of the droppers length:** This problem consists on the determination of the droppers length for obtaining an adecuate position in the contact wire, as parallel to the ground, as parabolic shape, in order to compesate the difference of stiffness between the supports and the center. This problem requires a study of the static forces in the wires.
- **The static problem:** It consists on the determination of the static position of the catenary when a force is applied. This allows to know the variation of the

stiffness along the line.

- **The dynamic problem:** This allows to simulate the behaviour of the pantograph-catenary in time.

In the study of the last two problems, a discretization using `FEM (Finite Element Methods)` must be used. In order to be able to deal with a great number of variables, some method for getting adecuate computational efficiency is required.

## 2.2 Modelization of the carrier and contact wire

The carrier and the contact wire can be considered as a pretensed beam. Under this assumption, the following Euler-Bernouilli equation is used:

$$\frac{p}{g}\ddot{y} = -EIy^{IV} + T_x y'' - p, \tag{1}$$

Where $p$ is the uniform load of the wire, $T_x$ is the horizontal tension of the wire, $E$ is the elastic module, $g$ represents the gravity force and $I$ the diametral moment of inertia.

Equation 1 allows us to discretize the system using `FEM` in order to obtain the stiffness matrix and the static equation of an element of the wire with a length of $l$:

$$kq = r,$$

In this case, each node has two variables: the vertical position $y_i$ and the angle $\theta_i$

$$k = \frac{EI}{l^3}\begin{bmatrix} 12 & 6l & -12 & 6l \\ 6l & 4l^2 & -6l & 2l^2 \\ -12 & -6l & 12 & -6l \\ 6l & 2l^2 & -6l & 4l^2 \end{bmatrix} + \frac{T_x}{30l}\begin{bmatrix} 36 & 3l & -36 & 3l \\ 3l & 4l^2 & -3l & -l^2 \\ -36 & -3l & 36 & -3l \\ 3l & -l^2 & -3l & 4l^2 \end{bmatrix}, \tag{2}$$

$$r = \begin{bmatrix} -\frac{pl}{2} \\ -\frac{pl^2}{l2} \\ -\frac{pl}{2} \\ \frac{pl^2}{l2} \end{bmatrix}, \tag{3}$$

$$q = \begin{bmatrix} y_i \\ \theta_j \\ y_i \\ \theta_j \end{bmatrix}. \tag{4}$$

In the case that we do not consider the bending stiffness (see Figure 2), the cables are considered as a pretensed string, then each node has a variable, that is, the vertical position $y_i$. In this case, the system is represented as

$$k = \frac{T_x}{l}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad r = \begin{bmatrix} -\frac{pl}{2} \\ -\frac{pl}{2} \end{bmatrix}, \quad q = \begin{bmatrix} y_i \\ y_j \end{bmatrix}. \tag{5}$$

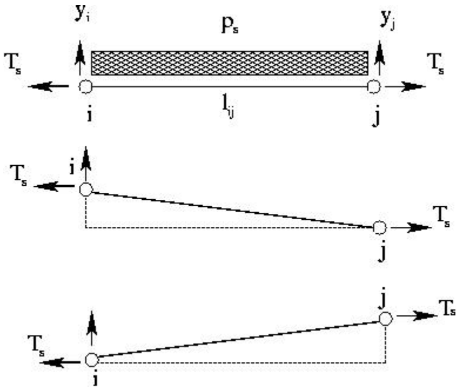


Figure 2: String elements

### 2.3 Modelization of the droppers

The droppers can be considered as an elastic bar with a length of $l$, which are deformated from an initial length $l_0$ by an initial load $F$. The stiffness matrix, the independent term and the vector variables are:

$$k = \frac{EA}{l}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad r = \begin{bmatrix} \frac{EAl_0}{l} \\ -\frac{EAl_0}{l} - P \end{bmatrix} = \begin{bmatrix} EA - F \\ -EA + F - P \end{bmatrix}, \quad q = \begin{bmatrix} y_i \\ y_j \end{bmatrix}, \tag{6}$$

where $P$ is the weight of the dropper and $A$ the cross area. The initial load $F$ is known by the static analysis of the forces.

The droppers only can work in a traction mode. Their effect of the opposite case is not considered.

### 2.4 Modelization of the campensation arm

The effect of the compensation arm can be considered as an spring with a vertical force $f_b$over the contact wire:

$$f_b = r_0 + (y_h - y_A)k_b, \tag{7}$$

being $k_b$ the apparent stiffness of the compensation arm, $y_A$ is the dynamic position of the holding point, $y_h$ is the static position of the holding point, a known data, and $r_o$ is the weight of the cable that supports the compensation arm.

### 2.5 The static problem

In the static problem the static position of the system is determined when we apply a vertical force over the contact wire. So, it is necessary to configurate the stiffness matrix $K$ and the independent term $R$ of the system and then solving the lineal system for the position of the nodes $Q$ of the cables:

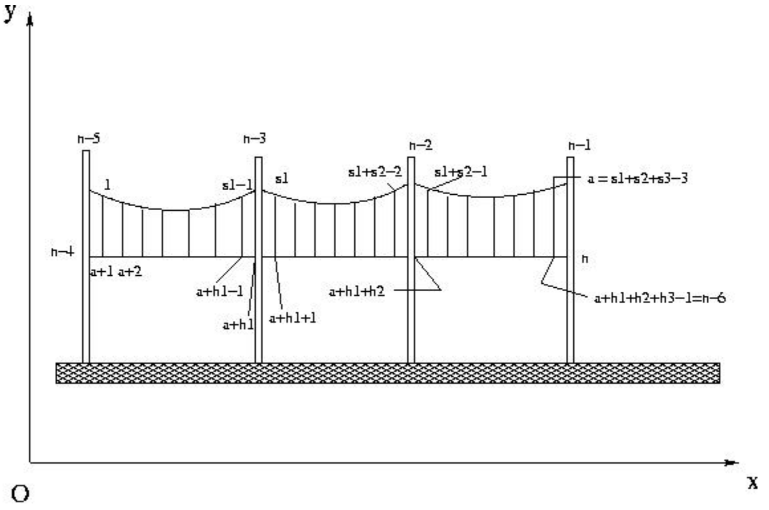


Figure 3: Model of Catenary. Notation

$$KQ = R. \tag{8}$$

Once the equilibrium position of the system is obtained, it is needed to check if all droppers work in a traction mode. In an affirmative case the problem is solved, but in the negative case, the stiffness matrix and the independent term must be reconfigured in order to eliminate the terms of the droppers that do not work, obtaining the new position and repeating this problem until all droppers work in a traction mode.

### 2.6 Discretization process

The carrier and the contact wires are discretized according to a finite element method (`FEM`), from left to right in a progressive way ([7, 8, 9, 10]. First, inner nodes of the carrier are numbered obtaining $s$ elements and $n_p$ droppers $(os_1, os_2, os_{np})$. Making the same numeration for the contact wire, $s+h-2$ elements are obtained (the numeration of droppers for the contact wire is $oh_1, oh_2, ..., oh_{np})$ (see Figure 3).

Considering the conections between nodes (see Figure 4 and 5) the stiffness matrix $K$ is conjugate.

From a general point of view, the stiffnes matrix has the following structure

$$\left( \begin{array}{cc} K_{11} & K_{12} \\ K_{21} & K_{22} \end{array} \right) \left( \begin{array}{c} Y_1 \\ Y_2 \end{array} \right) = \left( \begin{array}{c} R_1 \\ R_2 \end{array} \right) \tag{9}$$

where $K_{11} \in R^{nl \times nl}$, $K_{12} \in R^{nl \times na}$, $K_{21} \in R^{na \times nl}$ and $K_{22} \in R^{na \times na}$, being $nl$ the number of free nodes and $na$ the number of nodes subjected to constrains. $Y_1$ represents the unknows in this equation and $Y_2$ are the boundary conditions.

Operating by blocks, the following pair of equations are obtained

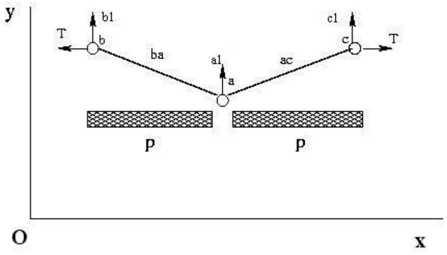


Figure 4: Nodes linked by means of two string elements.

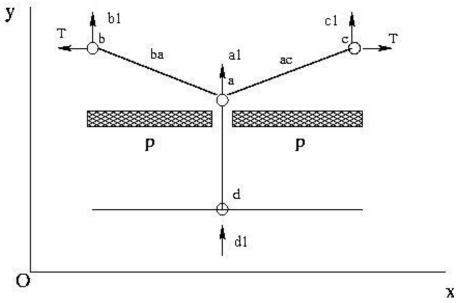


Figure 5: Nodes linked by means of two string elements and one elastic bar element.

$$K_{11}Y_1 + K_{12}Y_2 = R_1, \tag{10}$$

$$K_{21}Y_1 + K_{22}Y_2 = R_2. \tag{11}$$

From this pair of equations, only the first one has interest in order to calculate the unknows $Y_1$. Actually, the final system of equations to be solved is

$$K_{11}Y_1 = R_1 - K_{12}Y_2. \tag{12}$$

In 12, all terms except $Y_1$ are known.

One example of $K_{11}$ matrix is shown in Figure 6. This matrix has a high degree of sparsity. So, a good treatment of the storage space, and the application of a suitable method to solve sparse systems lead us to the High Performance Computing approach, aim of this paper.

## 3 High Performance Computing Approach

In order to solve a mathematical problem in a efficient way on a computer, the following steps are involved [13]

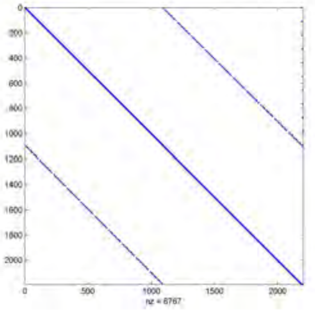


Figure 6: Example of stiffness matrix (test 4)

1. Making a mathematical model of the problem, translating the problem into a mathematical language, eg. ordinary differential equations.

2. Finding or developing constructive methods for solving the mathematical model, that is, a literature search to find what methods are available for the problem.

3. Identifying the best method from a numerical point of view.

4. Implementing on the computer the numerically effective method identified in the previous step.

In general, the developed software has to be a *high-quality mathematical software* which guarantees a good solution to the problem. This high quality mathematical software should have the following features: Power and flexibility, easily read and modified, portability, robustness, efficient and economic in use of storage.

The two last points are specially important in the problem solved in this work. In particular, the sparsity and simmetry of the stiffness matrix has been exploted, improving the efficiency of the implementation and dramatically reducing the memory storage requirements.

Finally, a High Performance Implementation (HPI) has to take into account the features of current architectures like, for example, cache memory. These features are particularly important when rebuilding the traditional algorithms to a block-oriented implementations. Block-oriented algorithms reduce drastically the data flow between main memory and secondary memory enhancing the performance of the final implementation. These HPI have been carried out by using BLAS and SPARSKIT standard linear algebra libraries.

The BLAS [11](*Basic Linear Algebra Subroutines*) library includes subroutines for common linear computations such as dot-products (BLAS-I), matrix-vector multiplication (BLAS-II), and matrix-matrix multiplication (BLAS-III).

Sparse matrices appear on a lot of current problems in science and engineering. Due to that fact, an intensive research is being carried out in this area producing lot of storage schemes and methods to deal with sparse matrices. SPARSKIT [12] is a software packet which allows us to work with different storage schemes (COO, CSR, CSC, etc) and iterative methods for solving sparse systems of equations. This packet is divided into several modules for conversion of storage scheme (*FORMAT* module), basic linear algebra operations over sparse matrices (*BLASSM* and *MATVEC* module), system of equations solvers (*ITSOL* module), etc.

## 4 Experimental Results

In this section, the experimental results obtained with the new HPC implementation of the algorithm for solving the static equilibrium equation are presented.

The test battery used in the experiments is shown in Table 1

| Test | nv | np | nps | nph | lv |
|---|---|---|---|---|---|
| 1 | 10 | 10 | 1 | 2 | 59 |
| 2 | 15 | 11 | 1 | 2 | 64 |
| 3 | 3 | 15 | 1 | 1 | 64 |
| 4 | 10 | 10 | 10 | 10 | 59 |

Table 1: Test Battery

Where $nv$ is the number of sections, $np$ is the number of droppers, $nps$ represents the number of elements between droppers in the carrier, $nph$ is the number of elements between droppers in the contact and $lv$ is the length of the section.

The experiments have been carried out in a Pentium III-650MHz with Red Hat Linux V. 7.2 operating system. The experimental platform has 385 MBytes of RAM memory and a cache memory of 256 KBytes. BLAS and SPARSKIT libraries have been compiled in this machine in order to obtain better performances.

Test 1 to 3 are low dimension problems used to verify the results, and test 4 is a more realistic example (see Figure 6). Thanks to the sparse storage scheme used in this work, the amount of used memory has been considerably decreased. This reduction in the storage space compared with the original algorithm is summarized in Table 2. In Table 2, $N$ represents the number of nodes and $nz$ the number of non-zero elements in the stiffnes matrix.

| Test | N | nz | % Reduction of memory |
|---|---|---|---|
| 1 | 332 | 1157 | 98,95 |
| 2 | 542 | 1902 | 99,35 |
| 3 | 98 | 366 | 96,20 |
| 4 | 2202 | 6767 | 99,86 |

Table 2: Reduction of memory requirements

The execution time has also been drastically reduced. Table 3 shows the execution time for the different tests and the percentage of time reduction with respect to the original implementation

| Test | Number of iterations | Execution time (msecs) | % Reduction of memory |
|---|---|---|---|
| 1 | 27 | 6,379 | 98,94 |
| 2 | 15 | 20,124 | 99,67 |
| 3 | 24 | 1,389 | 99,04 |
| 4 | 57 | 228,58 | * |

Table 3: Reduction of execution time

The test 4 could not be executed by using the original implementation.

## 5 Conclusions and Future Work

In this work a High Performance Computing Algorithm has been developed for solving the static equilibrium equation of the pantograph/catenary system of High Speed Railways. This new approach is based on the implementation philosophy of High Quality Sotfware.

The experimental results show that the HPC resulting algorithm provides spectacular reduction in the memory requirements as well as in execution time.

By using BLAS and SPARSKIT standard linear algebra libraries, two secondary objectives, but not less important, are achieved, i.e., portability and efficiency.

This work is the initial point of a lot of computational efforts in order to apply the HPC philosophy to different algorithms developed by the authors [17] following tradicional implementations. So, the future work could be outlined in the following points:

- Consider the solution of the static equilibrium equation when the spans contains different number of droppers and these are not equispaced.
- To extent this work considering the stitched catenary.
- To deal with the dynamical problem with guarantees.
- According to the results obtained for the dynamical problem, think about parallel implementations on shared memory platforms based on threads [14] or on distributed memory platforms based on MPI [15] and using the standard library PETSC [16].

The obtained algorithms will be used by RENFE, the Spanish Railway Company, in the design of high speed railways (AVE program).

## 6 Acknowledgments

We will thank Jesús Montesinos, engineer of RENFE, for its technical support and good knowledge of the real problem beyond the simulation world.

# References

[1] Poetsch G., J. Evans, R. Meisinger, W. Kortum, M. Baldauf, A. Veitl and J. Wallaschek. *"Pantograph/catenary dynamics and control"*, Vehicle System Dynamics, 28:159-195, 1997.

[2] Poetsch G. and J. Wallaschek. *"Symulating the dynamic behaviour of electrical lines for high-speed trains on parallel computers"*, International Symposium on Cable Dynamics, Lige, 1993.

[3] Simeon, B. and Arnold M. *"The simulation of pantograph and catenary: a PDAE approach"*, Technical Report 1990, Fachbereich Mathematik Technische Universitat Darmstadt, 1998.

[4] Garfinkle, M. *"Tracking pantograph for branchline electrification"*, Technical Report -, School of Textiles and Materials Technology, University of Philadelphia, 1998.

[5] Poetsch G., J. Evans, R. Meisinger, W. Kortum, M. Baldauf, A. Veitl, & J. Wallaschek. *"Pantograph/catenary dynamics and control"*, Vehicle System Dynamics, 28:159–195, 1997.

[6] Carsten, N. J. *"Nonlinear systems with discrete and continuous elements"*, PhD thesis, University of, 1997.

[7] K. J. Bathe. *"Finite Element Procedures in Engineering Analysis"*, Ed. Prentice-Hall, 1996.

[8] Thomas J. R. Hughes. *"The Finite Element Method"*, Ed. Prentice-Hall, 1987.

[9] Zienkiewics, R. L. Taylor. *"El Metodo de los Elementos Finitos"*, Ed. McGraw Hill, 1980.

[10] Cook D. C., Malkus D. S., Plesha M. E. *"Concepts and Applications of Finite Element Analysis"*, Ed. John Wiley and Sons, 1989.

[11] J. J. Dongarra, J. Du Croz, I. S. Duff and S. Hammarling. *"A set of Level 3 Basic Linear Algebra Subprograms"*, ACM Trans. Math. Soft, 1990.

[12] Yousef Saad. *"SPARSKIT: a basic tool kit for sparse matrix computations"*, University of Illinois and NASA Ames Research Center, 1994.

[13] B. Nath Datta, *"Numerical Linear Algebra and Applications"*, Broks/Cole Publishing Company, 1995.

[14] Mueller, F. *"Pthreads Library Interface"*, Institut fur Informatik, March, 1999.

[15] Gropp, W. and Lusk, E. and Skjellum, A. *"Using MPI: Portable Parallel Programming with the Message-Passing Interface"*, MIT Press, 1994.

[16] Batish Balay, William Group, Lois Curfman McInnes, B. Innes. *"PETSC 2.0 User's Manual"*, Mathematic and Computer Science Division, 2000.

[17] J. Benet, F. Cuartero and T. Rojo. *"A tool to calculate catenaries in railways"*, Seventh International Conference on Computer in Railways, COMPRAIL-2000, 2000.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# Phenomenological to Molecular Modelling of Hysteresis in Viscoelastic Polymers: Elastomers to Biotissue

H.T. Banks
Center for Research in Scientific Computation
North Carolina State University
Raleigh, NC 27695-8205

## Extended Abstract

In control and systems theory, delay systems or systems with memory (hysteresis) have played an important role for many years because of the early realizations by Minorsky and others [34, 35, 36, 27, 28, 38] that feedback design based on dynamics wherein one ignores any delays may fail catastrophically to stabilize or control a system in which delays or hysteresis are present in the dynamics. This is true whether the hysteresis is a fundamental part of the underlying dynamics or a part of the input or control operator. For the latter there is a growing body of literature [5, 6, 7, 26, 33, 45] on the Preisach and related theories for hysteretic control input such as arises in smart material systems [16, 18, 41]. Here we shall focus on the delays or hysteresis arising in the fundamental dynamics of the systems to be stabilized or controlled. In particular we consider viscoelastic materials that are polymeric in nature. This includes a wide range of materials of current importance such as (rubber or silicone based) filled elastomers and all types of biotissue (soft tissue, ligaments, cartilage, etc.).

The mathematical modelling of *viscoelasticity* (sometimes also loosely referred to as *hysteresis*) in materials using ideas from elasticity has attracted the attention of a large number of investigators over the past century. Among significant contributors (see the many references in [17, 19, 20, 22, 23, 32, 37, 39, 42, 44, 46, 47]) have been some of the true giants from the fields of engineering and material sciences. One of the most widely used empirical models for viscoelasticity in materials is the Boltzmann convolution law [12, 20, 22, 23, 46], one form of which is given in equation (1)

$$\sigma(t) = g_e(\epsilon(t)) + C_D\dot{\epsilon}(t) + \int_{-\infty}^{t} Y(t-s)\frac{d}{ds}g_v\left(\epsilon(s), \dot{\epsilon}(s)\right) ds, \tag{1}$$

where $\epsilon$ is the infinitesimal strain, $Y$ is the convolution memory kernel, and $g_e$ and $g_v$ are nonlinear functions accounting for the elastic and viscoelastic responses of the elastomers, respectively; for summaries and further references, see Chapter 2 of [23] as well as [12]. This form of model, when incorporated into force balance laws, results in *integro-partial differential equations* which are most often phenomenological in nature as well as being computationally challenging both in simulation and control design. This stress-strain law implies that the stress depends not only on the current strain and strain rate but also on the history of the strain and the strain-rate. It is very important to note that the stress-strain law (1) contains various standard *internal strain* or *internal variable* formulations as special cases. The anelastic displacement field (ADF) models of Lesieutre [30, 31] for composite materials exhibiting both elastic and anelastic displacement fields are formulated on the assumption that the host elastic material contains anelastic materials with internal strains $\varepsilon_1$ which are elastic strain driven. That is, the constitutive laws have the form

$$\sigma(t) = E\varepsilon(t) - E_1\varepsilon_1(t), \tag{2}$$

where the internal strain is given by

$$\dot{\varepsilon}_1(t) + \frac{1}{\tau}\varepsilon_1(t) = c_2\varepsilon(t), \quad \varepsilon_1(0) = 0, \tag{3}$$

or equivalently,

$$\varepsilon_1(t) = \int_0^t c_2 e^{-\frac{t-s}{\tau}}\varepsilon(s)ds.$$

Several generalizations of this formulation exist, e.g., Johnson, et al., [24, 25], suggest that the internal strain is strain *rate* driven, i.e.,

$$\dot{\varepsilon}_1(t) + \frac{1}{\tau}\varepsilon_1(t) = c_2\dot{\varepsilon}(t). \tag{4}$$

The Boltzmann-type law (1) (under appropriate assumptions on the past memory from $-\infty$ to 0) corresponds to an internal strain model of the form

$$\dot{\varepsilon}_1(t) + \frac{1}{\tau}\varepsilon_1(t) = \frac{d}{dt}g_v(\varepsilon(t), \dot{\varepsilon}(t)), \quad \varepsilon_1(0) = 0. \tag{5}$$

This form is often chosen since one finds that neither (3) nor (4) provide laws that readily describe experimental data, especially in the cases of filled elastomers, biotissues and other molecular polymers.

Fung, in his extensive efforts [23] with biomechanics and biotissue, develops and presents the quasi-linear viscoelastic constitutive equation

$$S_{ij}(t) = \int_{-\infty}^t G_{ijkl}(t-\tau)\frac{\partial S_{kl}^{(e)}[\bar{E}(\tau)]}{\partial \tau}d\tau, \tag{6}$$

where $S_{ij}$ is the Kirchoff stress tensor, $\bar{E}$ is the Green's strain tensor, $G_{ijkl}$ is a reduced relaxation function, and $S_{kl}^{(e)}$ is the "elastic" stress tensor. For the scalar components $G_{ijkl}$, Fung proposes the reduced relaxation function $G(t)$ given in the form

$$G(t) = \left\{1 + C[E_1(\frac{t}{\tau_2}) - E_1(\frac{t}{\tau_1})]\right\}[1 + c\ln(\frac{\tau_2}{\tau_1})]^{-1}. \tag{7}$$

Here $E_1(z) = \int_z^\infty \frac{e^{-t}}{t}dt$, $C$ represents the degree to which viscous effects are present, and $\tau_1$ and $\tau_2$ represent fast and slow viscous time phenomena. We note that the internal strain variable formulation (2), (5) is equivalent to the constitutive relationship proposed by Fung if one considers an approximation of the relaxation function $G$ by a sum of exponential terms. Various internal strain variable models are investigated in [1] and a good agreement is demonstrated between a two internal strain variable model (e.g., of the form $\sigma = E\varepsilon - E_1\varepsilon_1 - E_2\varepsilon_2$) and undamped simulated data based on the Fung kernel $G$.

Since its introduction, this quasi-linear viscoelastic (QLV) theory of Fung has been applied successfully in stress-strain experiments to several types of biological tissue. A benefit to using (6) as a constitutive equation is that, unlike simpler models for viscoelasticity, it allows for the consideration of a continuous spectrum (e.g., see the discussions in [23]) of relaxation times and frequencies (this is also true of the probabilistic-based internal variable approach developed in [13] and described below). (The need for a continuum of relaxation times in certain materials was observed many years ago [21, 40, 43, 47].) While Fung's theory has been successfully employed for fitting hysteretic stress-strain curves, for control applications one is interested in using it in a full dynamical model. Unfortunately, the QLV, as presented by Fung, leads to exceedingly difficult computations within full dynamical partial differential equations, especially in estimation and control problems. This motivated the development of the internal variable approach described in [1, 13, 30] (which permits discrete approximation to a continuum) in attempts to approximate well the corresponding *dynamic responses* even in cases where the *stress-strain curves alone* do not produce adequate approximations – see [23].

The probabilistic based internal variable alternative [13] to Fung's kernel involves a parameter dependent kernel with a continuous distribution of parameters and internal variables. In the case of a finite combination of Dirac $\delta$ distributions, one obtains a finite summation of exponential functions as the approximation kernel (see the discussions below). This method can be extended to allow for consideration of a continuous spectrum of relaxation times and frequencies by utilizing absolutely continuous parameter distributions in place of the $\delta$ distributions.

The internal variable approach to overcome both conceptual and computational challenges is consistent with the belief that hysteresis is actually a manifestation of the presence of multiple scales in a physical or biological material system that is frequently modelled (and masked) with a phenomenological representation such as an hysteresis integral for the macroscopic stress-strain constitutive law. The *internal variable* modelling leads to an efficient computational alternative for the corresponding integro-partial differential equation models. In addition, it provides a "molecular" basis for the models (for a comparison of models of viscoelastic damping via hysteretic integrals versus internal variable representations, see [12] and the references therein).

Our own interest in viscoelasticity in polymeric materials has been motivated by projects in our Industrial Applied Mathematics Program with at least two of our industrial partners: The Lord Corporation and Medacoustics, Inc. The collaborations with polymer scientists and engineers at Lord involved the dynamic modelling of filled rubbers which experimentally exhibit both significant hysteresis and nonlinearity in tensile and shear deformations as depicted in the sample stress-strain curves in Figure 1. The efforts with engineers at Medacoustics used some of the viscoelastic models we have investigated in attempts to understand the propagation of arterial stenosis induced shear waves in composite biotissue in a sensor development and characterization project.

In some of our earlier efforts [14, 15], the models for hysteretic damping in elastomers employed a phenomenological Boltzmann-type constitutive law of the form (1). As explained in [11, 14], our nonlinear materials undergoing large deformations required the use of *finite* (as opposed to *infinitesimal*) strain theories [39]. However, since the nonlinearity between the stress and finite strain is an unknown to be estimated (using inverse problem algorithms) and since the finite strain can be expressed in terms of known nonlinearities as a function of the infinitesimal strain (at least in the problems of interest here), one can effectively formulate the problem as one of estimating the unknown nonlinearity between stress and infinitesimal strain (see [14]). Hence one can develop models for stress in terms of infinitesimal strain. Our previous efforts as summarized in [11] have shown, through com-

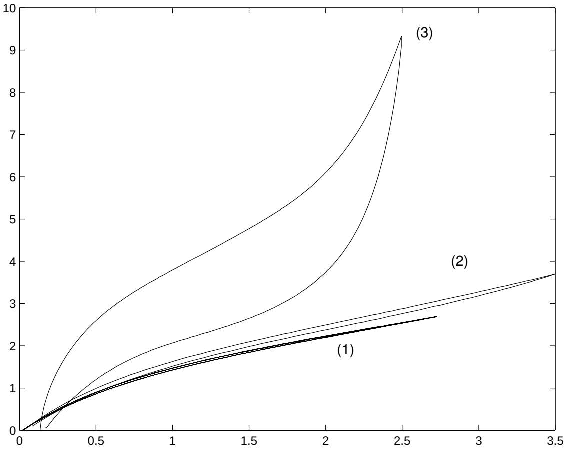


Figure 1: Experimental stress-strain curves for (1) unfilled, (2) lightly filled and (3) highly filled rubber in tensile deformations.

parison with experimental data, that the best fit to filled elastomer data occurs when $g_e$ and $g_v$ are cubic, along with $Y$ as a *distribution* of decaying exponentials. We subsequently [9, 10, 11] developed nonlinear models based on stick-slip "molecular" ideas of Johnson and Stacer [24] and Doi and Edwards [19] which resulted in a form for $g_e$, $g_v$ and $Y$ in (1) that matched the empirical findings reported in [11, 14, 15]. These models allow for multiple relaxation times present in polymer strands of composite materials within a virtual compartmental model of entangled chemically cross-linked/physically constrained system of long chain "molecules". While accounting for multiple relaxation parameters, these models do not include physically or chemically based parameters in representations of the polymer strands.

In the current review, we summarize the historical development of hysteresis laws outlined above and briefly outline two recent advances: $(i)$ a new constitutive model [4] that has been developed which combines the virtual stick-slip continuum "molecular-based" ideas of Johnson and Stacer [24] with the Rouse bead chain (see Figure 2) ideas as described in Doi and Edwards [19]; $(ii)$ a two dimensional version [8, 29] of a model that accounts for stenosis driven shear wave propagation in biotissue.

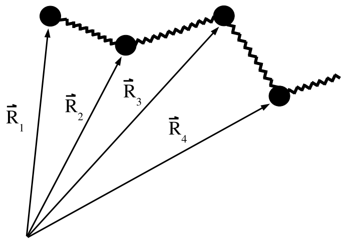


Figure 2: Representation of vectors for a bead-spring polymer molecule.

The new molecular-based constitutive model, in which polymer chains are treated as Rouse type strings of interconnected beads (a reasonable approximation for many materials), permits the incorporation of many important physical parameters (such as temperature, segment bond length, internal friction, and segment density) in the overall hysteretic constitutive relationship. Its form is similar to that developed in [11, 10] and does have the general form (1) of Boltzmann type, even though the kernel is *not* of convolution type.

In the discussions of the biotissue efforts, we recount an internal variable formulation of Boltzmann type hysteresis laws to investigate the propagation of stenosis generated waves in biotissue where it has been demonstrated that a viscoelastic (as opposed to an elastic) formulation is important and that waves generated in a two-dimensional cylindrical geometry with inner radius partial occlusions can be readily modelled and simulated.

The early models and the nonlinear extensions of the Boltzmann law did not provide insight into the underlying mechanisms for tensile and/or shear deformations in filled rubber or biotissue. This is not unexpected since the approaches described above are based on pseudo-phenomenological formulations. We then ([10, 11]) turned to a different approach based on molecular arguments which, as we shall see, lead precisely to the class of models based on a Boltzmann hysteresis formulation. As usual, one begins with force and moment balance and seeks constitutive laws for the viscoelastic stress term $\sigma_{visco}$ in

$$\sigma(t;\tau) = \sigma_{elast}(\varepsilon, \dot{\varepsilon}) + \sigma_{visco}(\varepsilon_1(\cdot)),$$

where $\varepsilon = \frac{\partial u}{\partial x}$ is the infinitesimal strain and $\varepsilon_1$ is an "internal strain" variable on which $\sigma_{visco}$ depends in an hysteretic manner. As described above, we found that a reasonable description of the data of interest could be given with the typical stress-strain relationship

$$\sigma(t) = g_e(\varepsilon(t), \dot{\varepsilon}(t)) + \int_0^t \gamma e^{-\frac{t-s}{\tau}} \frac{d}{ds} g_v(\varepsilon(s), \dot{\varepsilon}(s)) ds,$$

where $\tau$ is a relaxation parameter, $g_v$ is defined with cubic polynomials and $g_e = \hat{g}_e(\frac{\partial u}{\partial x}) + C_D \frac{\partial^2 u}{\partial t \partial x}$. We have already observed that this expression is equivalent to

$$\sigma(t) = \tilde{g}_e(\varepsilon(t), \dot{\varepsilon}(t)) + \gamma \varepsilon_1(t;\tau), \tag{8}$$

where, for a given "relaxation parameter" $\tau$, the internal strain $\varepsilon_1(t;\tau)$ satisfies (5). In fact, we found that highly filled rubbers required multiple relaxation times $\tau_1, \tau_2$ in an approximation to (7) to obtain good model fits to the data. As one might expect, molecular based formulations, where microscopic relaxation parameters vary across the population of molecules in the material, lead to internal dynamics of the form (5),(8) that involve multiple values of $\tau$. When combined with a Prohorov metric framework (see [2, 3]) for uncertainty in internal dynamics, these ideas lead to the computational models we have used. Indeed, the molecular based approach leads to a general class of models with uncertainty or randomness in the stress

$$\sigma(t, x; P) = \tilde{g}_e(\varepsilon(t, x), \dot{\varepsilon}(t, x)) + \gamma \int_{\mathcal{T}} \varepsilon_1(t, x; \tau) dP(\tau), \tag{9}$$

where $P$ is a probability distribution over the set $\mathcal{T}$ of possible relaxation parameters, and $\varepsilon_1(t;\tau)$ satisfies, for each $\tau \in \mathcal{T}$,

$$\dot{\varepsilon}_1(t, x; \tau) + \frac{1}{\tau} \varepsilon_1(t, x; \tau) = \dot{\varepsilon}(t, x) h(\varepsilon(t, x)).$$

For the reptation model derivation in [10], one begins with the Doi/Edwards [19] stick-slip molecular models as embodied in the continuous tube reptation models of Johnson/Stacer [24] wherein polymer materials such as rubber are postulated to be composed of two types of molecules. In tensile deformations, one denotes by $L(t)$ the length of *chemically cross-linked* or CC molecules, while $\ell(t)$ denotes the length of *physically constrained* or PC molecules. To use stick-slip models in continuum simulations of reptation in rubbers, one considers networks of "cells" or boxes of parallel-sided CC boxes and PC boxes with sides of length (principal stretches)

$$\lambda_c = 1 + \varepsilon = 1 + \frac{\partial u_c}{\partial x}, \;\; \lambda_p = 1 + \varepsilon_1 = 1 + \frac{\partial u_p}{\partial x},$$

respectively. Here $u_c$ denotes the deformations of the CC box and $u_p$ denotes the deformations of the PC box. Using a linear stick-slip assumption as in [24], and strain energy densities based on experiments of Young and Danik (see [9, 11] for details), one obtains as a limit of PC response to step tensile deformations of the CC molecules, the $\varepsilon, \varepsilon_1$ coupled dynamics

$$\dot{\varepsilon}_1 + \frac{1}{\tau}\varepsilon_1 = \dot{\varepsilon}\frac{1+\varepsilon_1}{1+\varepsilon}.$$

However, if one replaces the linear assumption of [9] by a nonlinear stick-slip hypothesis (which is the basis of the work in [10]), one obtains a more general nonlinear, dynamical relationship between $\varepsilon$ and $\varepsilon_1$ given by

$$\dot{\varepsilon}_1 + \frac{1}{\tau}\varepsilon_1 = \dot{\varepsilon} f((1+\varepsilon_1)/(1+\varepsilon)).$$

Expansion and truncation of higher order terms lead to equations of the form

$$\dot{\varepsilon}_1 + \frac{1}{\tau}\varepsilon_1 = \dot{\varepsilon}(\alpha_0 + \alpha_1\varepsilon + \alpha_2\varepsilon^2 + \alpha_3\varepsilon^3), \tag{10}$$

which are of the same form as the internal variable model (5),(8) with $g_v$ a cubic polynomial. For the corresponding contributions to $\sigma$ from the strain energy densities of Young-Danik/ Johnson-Stacer with the nonlinear stick-slip hypothesis, one obtains a contribution to the rate independent strain $g_v^s$ (after expanding $f$ in a Taylor series and dropping higher order terms) of the form

$$g_v^s(\varepsilon, \varepsilon_1) = g_{cubic}(\varepsilon) + \gamma_1\varepsilon_1,$$

where $\varepsilon_1$ is as before (i.e., the internal strain satisfying (10)). Thus, the total stress-strain relationship can be written in the form (9). If the measure P of (9) has atoms at $\tau_1$ and $\tau_2$, (i.e., the measure is composed of Dirac measures concentrated at $\tau_1$ and $\tau_2$), then the constitutive law leads precisely to the model

$$\sigma(t, x; P) = \tilde{g}_e(\varepsilon(t, x), \dot{\varepsilon}(t, x)) + \gamma_1\varepsilon_1(t, x; \tau_1) + \gamma_2\varepsilon_2(t, x; \tau_2),$$

which was used in the data fits in [1, 11].

**Summary**

In the presentation we will expand the details of the history and results from ideas ranging from Boltzmann to Doi-Edwards/Stacer-Johnson to Fung as outlined above. We then report on two recent efforts in the modelling of viscoelastic polymers. First, we outline a new constitutive model which combines the "molecular-based" ideas of Johnson and Stacer with the Rouse bead chain ideas and explain its relation to the Boltzmann phenomenological models. Second, we discuss a two-dimensional version of an internal variable model that accounts for shear wave propagation in biotissue and the model's relationship to the Fung kernel model. A brief summary of these two presentations is:

(i) A Stick-Slip/Rouse Hybrid Model:

We give a brief outline of a new constitutive model; more details of the derivation can be found in the report [4]. We model a polymer material undergoing directional deformation by assuming it is composed of two *virtual* compartments as depicted in Figure 3. One compartment consists of a *constraining tube* which is a macroscopic compartment containing both CC (chemically cross-linked) and PC (physically constrained) molecules. The other compartment is microscopic in nature and consist of those PC molecules aligned with the direction of the deformation. These molecules will at first "stick" to the constraining tube and be carried along with its motion, but will very quickly "slip" and begin to "relax" back to a configuration of lower strain energy. In the model derivation to obtain the constitutive law one computes the contributions of both "compartments" to the overall stress of this polymer material undergoing deformations.

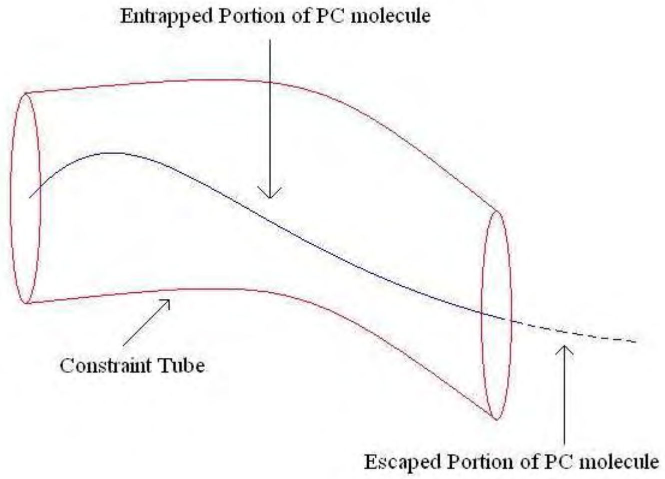

Entrapped Portion of PC molecule
Constraint Tube
Escaped Portion of PC molecule

Figure 3: PC molecule entrapped by the surrounding constraining tube.

(ii) Stenosis-Driven Shear Wave Propagation in Biotissue:

In a second discussion, we turn to recent results on the viscoelastic models for propagation of stenosis-driven biotissue waves mentioned above. Specifically we report on two dimensional models that employ an internal variable approach to model wave propagation. To motivate this, we recall [1] that coronary artery disease (CAD) is caused by atherosclerosis, the gradual accumulation of plaque along the walls of an artery. This buildup, known as a stenosis, restricts the flow of blood, leading to a decrease in the oxygen supply to the heart muscle. It is well known that arterial stenoses produce sounds due to turbulent blood flow in partially occluded arteries. In principle, turbulent normal wall forces exist at and downstream from an arterial stenosis, exerting pressure on the wall of the artery which then causes a small displacement in the surrounding body tissue. The goal is to model the propagation of the wave generated from the stenosis to the chest wall, and ultimately, to create an inverse problem methodology which can be utilized to determine the location of an arterial stenosis. We also discuss comparison of the viscoelastic model to an elastic one as well as present typical simulations for a biologically motivated example.

## Acknowledgements

This research was supported in part by the US Air Force Office of Scientific Research under grant AFOSR-FA9550-04-1-0220.

## References

[1] H.T. Banks, J.H. Barnes, A. Eberhardt, H. Tran and S. Wynne, Modeling and computation of propagating waves from coronary stenoses, *Computational and Applied Mathematics*, **21** (2002), 767–788.

[2] H.T. Banks and K. Bihari, Modeling and estimating uncertainty in parameter estimation, CRSC-TR99-40, NCSU, December 1999; *Inverse Problems*, **17** (2001), 1–17.

[3] H.T. Banks, D. Bortz, G.A. Pinter and L.K. Potter, Modeling and imaging techniques with potential application in bioterrorism, CRSC-TR03-02, NCSU, January 2003; Chapter 6 in *Bioterrorism: Mathematical Modeling Applications in Homeland Security,* (H.T. Banks and C. Castillo-Chavez, eds.), Frontiers in Applied Mathematics, Vol.28, SIAM, Philadelphia, 2003, 129–154.

[4] H.T. Banks, J.B. Hood, N.G. Medhin and J.R. Samuels, A stick-slip/Rouse hybrid model for viscoelasticity in polymers, Technical Report CRSC-TR06-26, NCSU, November, 2006; *Nonlinear Analysis: Real World Applications*, submitted.

[5] H.T. Banks and A.J. Kurdila, Hysteretic control influence operators representing smart material actuators: Identification and approximation , CRSC-TR96-23, August, 1996; Proc. 35th IEEE Conf. on Decision and Control (Kobe, Japan), December, 1996, 3711–3716.

[6] H.T. Banks, A.J. Kurdila and G. Webb, Identification of hysteretic control influence operators representing smart actuators, Part I: Formulation , CRSC-TR96-14, April 1996; *Mathematical Problems in Engineering*, **3** (1997), 287–328.

[7] H.T. Banks, A.J. Kurdila and G. Webb, Identification of hysteretic control influence operators representing smart actuators: Part II, Convergent approximations , CRSC-TR97-7, April, 1997; *J. of Intelligent Material Systems and Structures*, **8** (1997), 536–550.

[8] H.T. Banks and N.S. Luke, Simulations of propagating shear waves in biotissue employing an internal variable approach to dissipation, Technical Report CRSC-TR06-28, NCSU, December, 2006; *Communications in Computational Physics*, submitted.

[9] H.T. Banks and N.G. Medhin, A molecular based dynamic model for viscoelastic responses of rubber in tensile deformations, CRSC-TR00-27, NCSU, October, 2000; *Communications on Applied Nonlinear Analysis*, **8** (2001), 1–18 .

[10] H.T. Banks, N.G. Medhin and G.A. Pinter, Nonlinear reptation in molecular based hysteresis models for polymers, *Quarterly Applied Math.*, **62** (2004), 767–779.

[11] H.T. Banks, N.G. Medhin and G.A. Pinter, Multiscale considerations in modeling of nonlinear elastomers, Technical Report CRSC-TR03-42, NCSU, October, 2003; *J. Comp. Meth. Engr. Sci. and Mech.*, **8** (2007), 1–10.

[12] H.T. Banks and G.A. Pinter, Damping: hysteretic damping and models, CRSC-TR99-36, NCSU, December, 1999; in *Encyclopedia of Vibration*, ( S.G. Braun, D. Ewins and S. Rao, eds.), Academic Press, London, 2001, pp. 658-664.

[13] H.T. Banks and G.A. Pinter, A probabilistic multiscale approach to hysteresis in shear wave propagation in biotissue, *Multiscale Modeling and Simulation*, **3** (2005), 395–412.

[14] H.T. Banks, G.A. Pinter, L.K. Potter, M.J. Gaitens and L.C. Yanyo, Modeling of quasi-static and dynamic load responses of filled viscoelastic materials, Technical Report CRSC-TR98-48, NCSU, December, 1998; Chapter 11 in *Mathematical Modeling: Case Studies from Industry* (E. Cumberbatch and A. Fitt, eds.), Cambridge University Press, 2001, pp. 229–252.

[15] H.T. Banks, G.A. Pinter, L.K. Potter, B.C. Muñoz and L.C. Yanyo, Estimation and control related issues in smart material structures and fluids, Technical Report CRSC-TR98-02, NCSU, January, 1998; *Optimization Techniques and Applications* (L. Caccetta, et al., eds.), Curtain Univ. Press, July, 1998, pp. 19–34.

[16] M.Brokate and J. Sprekels, *Hysteresis and Phase Transitions*, Springer-Verlag, New York, 1996.

[17] M. Doi, *Introduction to Polymer Physics*, Clarendon Press, Oxford, 1996.

[18] M.V. Gandhi and B.S. Thompson, *Smart Materials and Structures*, Chapman and Hall, London, 1992.

[19] M. Doi and M. Edwards, *The Theory of Polymer Dynamics*, Oxford, New York, 1986.

[20] J.D. Ferry, *Viscoelastic Properties of Polymers*, John Wiley and Sons, Inc., New York, 1961.

[21] J.D. Ferry, E.R. Fitzgerald, L.D. Grandine and M.L. Williams, Temperature dependence of dynamic properties of elastomers: relaxation distributions, *Ind. Engr. Chem.*, **44** (1952), 703–706.

[22] Y.C. Fung, *Foundations of Solid Mechanics*, Prentice-Hall, Englewood Cliffs, NJ, 1965.

[23] Y.C. Fung, *Biomechanics: Mechanical Properties of Living Tissues*, Springer-Verlag, New York, 1993.

[24] A.R. Johnson and R.G. Stacer, Rubber viscoelasticity using the physically constrained system's stretches as internal variables, *Rubber Chemistry and Technology*, **66** (1993), 567–577.

[25] A.R. Johnson, C.J. Quigley and J.L. Mead, Large strain viscoelastic constitutive models for rubber, part I: Formulations, *Rubber Chemistry Technology*, **67** (1994), 904–917.

[26] M.A. Krasnosel'skii and A.V. Pokrovskii, *Systems with Hysteresis*, Nauka, Moscow, 1983; translated, Springer-Verlag, Berlin, 1989.

[27] A. Halanay, *Differential Equations*, Academic Press, New York, 1966.

[28] J.K. Hale, *Functional Differential Equations*, Springer-Verlag, New York, 1971.

[29] N.S. Luke, *Modeling Shear Wave Propagation in Biotissue: An Internal Variable Approach to Dissipation*, Ph.D. dissertation, NC State University, August, 2006.

[30] G.A. Lesieutre, Modeling frequency-dependent longitudinal dynamic behavior of linear viscoelastic long fiber components, *J. Composite Materials*, **28** (1994), 1770–1782.

[31] G.A. Lesieutre and K. Govindswamy, Finite element modeling of frequency-dependent and temperature-dependent dynamic behavior of viscoelastic materials in simple shear, *Int. J. Solids Structures*, **33** (1996), 419–432.

[32] J.E. Marsden and T.J.R. Hughes, *Mathematical Foundations of Elasticity*, Prentice-Hall, Englewood Cliffs, NJ, 1983.

[33] I.D. Mayergoyz, *Mathematical Models of Hysteresis*, Springer-Verlag, New York, 1991.

[34] N. Minorsky, Self-excited oscillations in dynamical systems possessing retatded actions, *J. Applied Mechanics*, **9** (1942), A65–A71.

[35] N. Minorsky, On non-linear phenomenon of self-rolling, *Proc. National Academy of Sciences*, **31** (1945), 346–349.

[36] N. Minorsky, *Nonlinear Oscillations*, Van Nostrand, New York, 1962.

[37] R.W. Ogden, *Non-Linear Elastic Deformations*, Ellis Horwood Limited, Chichester, 1984.

[38] M.N. Oguztorelli, *Time-Lag Control Systems*, Academic Press, New York, 1966.

[39] R.S. Rivlin, Large elastic deformations of isotropic materials, I, II, III, *Phil. Trans. Roy. Soc. A* **240**, (1948), 459–525.

[40] F. Schwarzl and A.J. Staverman, Higher approximation methods for the relaxation spectrum from static and dynamic measurements of viscoelastic materials, *Appl. Sci. Res.*, **A4** (1953), 127–141.

[41] R.C. Smith, *Smart Material Systems: Model Development*, Frontiers in Applied Mathematics, Vol. FR32, SIAM, Philadelphia, 2005.

[42] J.K. Stille, *Introduction to Polymer Chemistry*, John Wiley and Sons, Inc., New York, 1962.

[43] D. Ter Haar, A phenomenological theory of viscoelastic behavior, *Physica*, **16** (1950), 839–850.

[44] L.R.G. Treloar, *The Physics of Rubber Elasticity*, Clarendon, Oxford 1975.

[45] A. Visitin, *Differential Models of Hysteresis*, Springer-Verlag, New York, 1994.

[46] I.M. Ward, *Mechanical Properties of Solid Polymers*, J. Wiley & Sons, New York, 1983.

[47] M.L. Williams and J.D. Ferry, Second approximation calculations of mechanical and electrical relaxation and retardation distributions, *J. Poly. Sci.*, **11** (1953), 169–175.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# Functional quantization method using low discrepancy points with application to option pricing

**Marta B. Bastrzyk**[1], **Ben Niu**[1] **and Fred J. Hickernell**[2]

[1] *Department of Applied Mathematics, Illinois Institute of Technology*

[2] *Department of Applied Mathematics, Illinois Institute of Technology*

emails: `bastmar1@iit.edu`, `nben@iit.edu`, `fred@math.iit.edu`

### Abstract

We are interested in the use of functional quantization method with low discrepancy points. Two different sampling techniques with respect to Time Discretization Method and Eigenfunction Expansion Truncation Method are discussed. We apply our new algorithm to price European and Asian style derivatives. Numerical results from function quantization method with low discrepancy points are compared to Monte Carlo simulation and Quasi-Monte Carlo simulation results.

*Key words: Functional Quantization, low discrepancy points, Monte Carlo, Quasi Monte Carlo.*

## 1 Introduction

Generally, there are two ways to approximate a stochastic process $X_t$, where $X_t$ is assumed to be the Brownian motion because of our interest in derivative pricing. These two different ways are: time discretization method and eigenfunction expansion truncation method. The discretization method is done by discretizing time interval $[0, T]$ into equal subintervals. The idea behind the second method is Karhunen-Loeve expansion of Gaussian process. Specially speaking, Karhunen-Loeve eigenbasis and corresponding eigenvalues of Brownian motion admit a closed form. This particular method has been extensively explored by Gilles Pages and Jacques Printems, they used product quantizers to numerically solve option pricing problems. Our goal is to use low discrepancy points instead of product quantizers. We apply our new algorithm to Asian and European options, and we get competitive result compared to the method using product quantizers.

## 2 Time Discretization method

We discuss the time discretization techniques with respect to the application for Asian options. We consider continuous Asian call option defined on $[0, T]$ with initial asset price $S(0)$, strike price $K$, constant risk-free rate $r$ and volatility $\sigma$. The asset price can be formulated as:

$$S(X(t)) = S(0) \exp^{(r-\sigma^2/2)t+\sigma X(t)}.$$

The payoff depends on the whole path of the asset price process, which is

$$f(X(\cdot)) = \max \quad \frac{1}{T}\int_0^T S(X(t))dt - K, 0 \quad e^{-rT}.$$

The price of Asian call option is computed as

$$I = \mathbb{E}(f(X(\cdot)))$$

Here $X(t)$ is a Brownian motion defined on $[0, T]$, then we discretize the time interval $[0, T]$ to $s$ subintervals equally: $t_k = \frac{kT}{s}, k = 1, 2, \ldots, s$. After time discretization, we can approximate $X(t)$ at each discretized time by

$$X(t_k) = \sqrt{\frac{T}{s}}(Z_1 + \ldots + Z_k).$$

where $Z_1, \ldots, Z_s$ are i.i.d standard normal distributed random variables. For purpose of simulation, we can sample $X(t_k)$ by,

$$X_i(t_k) = \sqrt{\frac{T}{s}}(z_{i,1} + z_{i,2} + \cdots + z_{i,k}). \quad \mathbf{z}_i = (z_{i,1}, \ldots, z_{i,s}).$$

Here the $\{\mathbf{z}_i\}_{i=1}^n$ can be any sequence of points in $\mathbb{R}^s$. For example, $\{\mathbf{z}_i\}_{i=1}^n$ maybe i.i.d standard normal random variables, the inverse of uniformly distributed low discrepancy sequence on $[0, 1]^s$, e.g. Sobol' sequence, Halton sequence, or centralized Voronoi quantizers. $i$ means the $i^{th}$ simulated path. The choice of a good sequence of points is an activate research topic. For option pricing, this method will be suitable for European call and put option since the payoff depends only on $X(T)$. However, for a continuous Asian option where the payoff depends on the whole path of the asset price process, there is significant discretization error.

We assume the payoff function $f_{D,s} : \mathbb{R}^s \to \mathbb{R}$, where $f_{D,s}$ means payoff function for discretization method with s discretized time intervals. Then, the Asian call option pricing $I$ is approximately computed as

$$I = \mathbb{E}(f(X(\cdot))) \approx I_{D,s,n} = \mathbb{E}(f_{D,s}(X(\cdot))) = \sum_{i=1}^{n} f_{D,s}\left(X_i(t_1), \ldots, X_i(t_s)\right) w_i.$$

where

$$f_{D,s}(X(\cdot)) = \max\left(\frac{1}{s}\sum_{k=1}^{s} S(X(t_k) - K, 0\right) e^{-rT}.$$

In this equation, $n$ means the number of asset paths and $w_i$ means the weight function which can be chosen as $w_i = \frac{1}{n}$. We can see from above that $f_{D,s}$ is a $s-$dimensional function. This method admits time discretization error.

## 3 Eigenfunction Expansion Truncation Method

In the above method the accuracy and convergence will be affected by discretization error. In the second method, which is called functional quantization method studied by Professor Gilles Pages, we can ignore discretization error for one moment, however, we focus more on truncation error.

The basic principle under this idea is the Karhunen-Loeve expansion of Gaussian process, here, we consider about Brownian motion $X(t)$ defined on $[0, T]$. We expand $X(t)$ based on its eigenfunction and eigenvalue got from Karhunen-Loeve expansion of $X(t)$, which is

$$X(t) = \sum_{j=1}^{\infty} Z_j \sqrt{\lambda_j} e_j(t), \quad Z_j \sim \text{i.i.d} \quad N(0,1).$$

The above expansion has infinitely many eigenvalues and eigenfunctions, and for numerical computation purpose, we have to truncate it at some level, say, truncate it as $d$ dimensional expansion, which is

$$X(t) = \sum_{j=1}^{\infty} Z_j \sqrt{\lambda_j} e_j(t) \approx \sum_{j=1}^{d} Z_j \sqrt{\lambda_j} e_j(t).$$

Specifically for Brownian motion on $[0, T]$, the Karhunen-Loeve eigenfunction and its eigenvalues admit a closed form given by

$$e_j(t) = \sqrt{\frac{2}{T}} \sin \quad \pi \quad j - \frac{1}{2} \quad \frac{t}{T} \quad , \quad \lambda_j = \quad \frac{T}{\pi(j-1/2)} \quad , \quad j \geq 1.$$

Having the above close form expansions, we can sample $X(t)$ by choosing some standard normal distributed random variable which can either be generated by simple random number generator or be transformed from some low discrepancy sequence by normal inverse function. For $i = 1, 2, \cdots, N$, we have

$$X_i(t) \approx \sum_{j=1}^{d} z_{ij} \sqrt{\lambda_j} e_j(t), \quad \mathbf{z}_i = (z_{ij}) = (z_{i1}, z_{i2}, \cdots, z_{id}), \quad z_{ij} \sim \text{i.i.d} \quad N(0,1).$$

then, we can evaluate the Brownian motion $X(t)$ at any time t by

$$X_i(t_k) \approx \sum_{j=1}^{d} z_{ij} \sqrt{\lambda_j} e_j(t_k), \quad \mathbf{z}_i = (z_{ij}) = (z_{i1}, z_{i2}, \cdots, z_{id}), \quad z_{ij} \sim \text{i.i.d} \quad N(0,1).$$

Here we can use functional quantization based technique to simulate the stock path. With the above sampled Brownian motion path, we can solve the above Asian option problem. .

## 4 Numerical Experiments

We test the above algorithm on European style option and Asian style option. For European option, we do not need to use time discretization technique because the payoff of European option only depends on the state at expiration date. We use pseudo-random sequence, Vander Corput sequence and randomized Vander Corput sequence for Monte Carlo and Quasi-Monte Carlo simulation, then we use the same sequence on functional quantization based method, our numerical test shows that functional quantization method using low discrepancy points is the fastest method to approach the true value. For Asian option, we use both of the time discretization method and functional quantization method. For time discretization techniques, we use middle point rule and rectangular rule. Our numerical results show that functional quantization method using low discrepancy points, and with middle point rule discretization technique are the most efficient compared to Monte Carlo simulation using pseudo random sequence, Quasi-Random sequence(Sobol'), and functional quantization method using pseudo random sequence.

## Acknowledgements

This work is jointly done by Professor Fred J. Hickernell, Ben Niu, Marta B. Bastrzyk.

## References

[1] G. Pagés, J. Printems, *Functional Quantization for Pricing Derivatives*, Research Report,CERMICS/ENPC, (2004), no 264

[2] Lapeyre B., Team E., *Competitive Monte Carlo methods for the pricing of Asian Options*, Journal of Computational Finance. Geom. **5** (2001) 39-57.

[3] Glasserman, P., *Monte Carlo Methods in Financial Eigneering*, Springer (2003).

*Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2007
Chicago, 20–23 June 2007.*

# Robust Computational Techniques For Global Solution and Normalized Flux of Singularly Perturbed Reaction-Diffusion Problems

**Rajesh K. Bawa**[1]

[1] *Department of Computer Science, Punjabi University, Patiala, Punjab, INDIA 147002*

emails: `rajesh_k_bawa@yahoo.com`

### Abstract

In this talk, First, An $\varepsilon$-uniform computational technique will be presented for singularly perturbed two-point boundary-value (BVP) problems of reaction diffusion type with natural boundary conditions. This technique combines a cubic spline scheme and classical difference scheme. In the inner region, we are using cubic spline approximation for solution at mesh points, whereas in outer region, classical difference scheme is used. This variable mesh scheme is applied on well-known piecewise shishkin mesh. This hybrid scheme is generalized for problems having Robin type boundary conditions. Also, Techniques to find global solution and normalized flux of the problems with natural boundary conditions will be presented. Detailed error analysis is provided and various numerical example are taken to show the efficiency of these techniques.

*Key words: Singular perturbation problems (SPPs), reaction-Diffusion Problems, Cubic spline, Global Solution, Normalized Flux.*

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# Second-Order $\varepsilon$-Uniformly Convergent Scheme for Singularly Perturbed Convection-Diffusion Problems

**R.K. Bawa**[1] **and S. Natesan**[2]

[1] *Department of Computer Science, Punjabi University, Patiala - 147 002, India.*

[2] *Department of Mathematics, Indian Institute of Technology, Guwahati - 781 039, India.*

emails: `rajesh_k_bawa@yahoo.com`, `natesan@iitg.ernet.in`

**Abstract**

In this paper, we have considered singularly perturbed two-point boundary-value (BVP) problems of convection diffusion type. A variable mesh hybrid scheme is proposed for these types of problems. This scheme combines a cubic spline scheme and mid-point scheme. In the inner region, the convective term is approximated by three-point differences by spline approximation of solution at mesh points, whereas in outer region the mid-point approximations are used for convective term and the classical central difference scheme is used for the diffusive term. The first-order derivative in the left boundary point is approximated by the cubic spline. This scheme is applied on the boundary layer resolving Shishkin mesh. In order to show the second-order $\varepsilon$-uniform convergence of the scheme, a numerical example is taken. Maximum errors and computational order of convergence are obtained for various values of the perturbation parameter $\varepsilon$ and the number of mesh points $N$.

*Key words: Singular perturbation problems, cubic spline, mid-point scheme, piece-wise uniform mesh, uniform convergence.*
*MSC 2000: 65L10*

## 1 Introduction

Singular perturbation problems (SPPs) arise often in applied areas like fluid mechanics, chemical reactor theory, quantum mechanics, etc. The solution of these problems has a multiscale character, it has two components, one slowly varying and another fastly varying in some parts of the domain of interest, which creates difficulties in solving these problems numerically. Basically, the classical finite difference/element schemes fail to capture the steep gradients in the boundary layer regions. Therefore, special attention is required for the numerical approximations of these problems. For a detailed discussion on the analytical and numerical treatment of SPPs we may refer the reader to the books of O'Malley [7], Doolan et al. [1], Roos et al. [8] and Miller et al. [3].

In this article, we propose a hybrid numerical scheme for the following singularly perturbed two-point BVP:

$$Lu \equiv \varepsilon u''(x) + a(x)u'(x) = f(x), \quad x \in D = (0,1) \tag{1.1}$$

$$-a(0)u'(0) = -f(0), \quad u(1) = \beta, \tag{1.2}$$

where $\varepsilon > 0$ is a small parameter, $a$ and $f$ are sufficiently smooth functions such that $a(x) \geq \alpha > 0$, $x \in \overline{D} = [0, 1]$. Under these assumptions, the BVP (1.1-1.2) has a unique solution $u(x) \in C^2(D) \bigcap C^1(\overline{D})$ exhibiting a weak boundary layer at $x = 0$, see for example [8].

The application of second order cubic spline difference scheme on whole domain using Shishkin mesh may result in oscillations in the coarser region due to involvement of three-point approximations of convective term in the scheme for smaller values of $\varepsilon$. Whereas, the use of midpoint scheme in whole domain results in oscillation free scheme but with first order convergence rate. In order to retain the second order convergence of cubic spline scheme together with non-oscillating behavior of mid-point scheme, we club these two schemes by taking cubic spline scheme in inner region and mid-point scheme in outer region. The value of the transition parameter $\sigma_0$ is chosen in such a way that the resultant hybrid scheme is second order $\varepsilon$-uniformly convergent through out the domain. More precisely, we take $\sigma_0 = 2/\alpha$.

The convection-diffusion BVP (1.1-1.2) has been studied earlier by Natesan et al. [5, 6, 9]. In all these articles, the BVP is solved by using a domain decomposition method. The domain of the differential equation is divided into non-overlapping subdomains and the differential equation is solved on each subdomain with suitable conditions at the interfaces of the domain. These methods are suitable for parallel computers, indeed, in [9], the numerical scheme is implemented in a parallel machine. Bawa et al. have derived difference schemes for SPPs using cubic splines in [2].

In the following section $K$ and $C$ denote generic positive constants independent of nodal points, mesh size and the perturbation parameter $\varepsilon$.

## 2 The Continuous Problem

As the proposed scheme can be generalized easily for problems containing reactive terms, we study the analytical behavior of the solution of the following BVP, which will be used to derive error bounds for the derivatives of the solution.

$$Lu \equiv \varepsilon u''(x) + a(x)u'(x) - b(x)u(x) = g(x, \varepsilon), \quad x \in \Omega \tag{2.1}$$

$$B_0u(0) \equiv b(0)u(0) - a(0)u'(0) = -g(0, \varepsilon), \quad B_1u(1) \equiv u(1) = \beta. \tag{2.2}$$

**Definition 2.1** *A function $g(x, \varepsilon)$ is said to be of Class$(K, j)$, if the derivatives of $g$ with respect to $x$ satisfy*

$$\mid g^{(i)}(x, \varepsilon) \mid \leq K[1 + \varepsilon^{-i} \exp(-\alpha x/\varepsilon)], \quad 0 \leq i \leq j, \quad x \in \overline{\Omega}$$

**Lemma 2.2** *[3]. Let $v$ be a smooth function satisfying $B_0v(0) \geq 0$, $B_1v(1) \geq 0$ and $Lv(x) \leq 0, \forall x \in \Omega$. Then $v(x) \geq 0, \forall\, x \in \overline{\Omega}$.*

**Lemma 2.3** *[3]. Let $v$ be a smooth function. Then, we have the following uniform stability estimate*

$$\mid v(x) \mid \leq C[\mid B_0v(0) \mid + \mid B_1v(1) \mid + \max_{y \in \overline{\Omega}} \mid Lv(y) \mid], \quad \forall\, x \in \overline{\Omega}.$$

**Lemma 2.4** *Let $g \in$ Class$(K, 0)$. Then the solution $y$ of (2.1-2.2) satisfies*

$$\mid y^{(i)}(x) \mid \leq C, \quad i = 0, 1.$$

**Proof.** The proof can be seen in [6]. ■

**Lemma 2.5** *[6] Let $g \in Class(K, j)$. Then the solution $y$ of (2.1-2.2) satisfies*

$$\mid y''(0) \mid \leq C \quad \textit{and} \quad \mid y^{(i)}(0) \mid \leq C\varepsilon^{-i+1}, \quad i = 3(1)j + 1.$$

**Theorem 2.6** *Let $g$ be of $Class(K, j)$. Then the solution $y$ of (2.1-2.2) satisfies*

$$\mid y^{(i)}(x) \mid \leq C[1 + \varepsilon^{-i+1} \exp(-\alpha x/\varepsilon)], \quad i = 2(1)j + 1, \quad x \in \overline{\Omega}.$$

**Proof.** Refer [6] for the proof. ∎

**Corollary 2.7** *If $u(x)$ is the solution of (1.1-1.2) and $a, b$ and $f$ are in $C^j(\overline{\Omega})$, then $u$ satisfies*

$$\mid u^{(i)}(x) \mid \leq C[1 + \varepsilon^{-i+1} \exp(-\alpha x/\varepsilon)], \quad i = 1(1)j + 1, \quad x \in \overline{\Omega}.$$

# 3 The Discrete Problem

In this section, first, we derive the cubic spline scheme on variable meshes, and then propose the hybrid scheme. Finally, we provide the piece-wise uniform Shishkin meshes for the SPP (1.1-1.2).

## 3.1 Cubic Spline Difference Scheme

Let the mesh points of $\overline{\Omega} = [0, 1]$ be

$$x_0 = 0,\ x_i = \sum_{k=0}^{i-1} h_k,\ h_k = x_{k+1} - x_k,\ x_N = 1,\ i = 1, 2, \ldots, N-1. \tag{3.1}$$

We derive the difference scheme in the following.

For given values $u(x_0), u(x_1), \cdots, u(x_N)$ of a function $u(x)$ at the nodal points $x_0, x_1, \ldots, x_N$, there exists an interpolating cubic spline $S(x)$ with following properties:

(i) $S(x)$ coincides with a polynomial of degree three on each subinterval $[x_i, x_{i+1}], i = 0, \ldots, N-1$;

(ii) $S(x) \in C^2[0, 1]$;

(iii) $S(x_i) = u(x_i),\ i = 0, \ldots, N$.

The cubic spline can be given by

$$\begin{aligned} S(x) = \quad & \frac{(x_{i+1} - x)^3}{6h_i} M_i + \frac{(x - x_i)^3}{6h_i} M_{i+1} + \left( u(x_i) - \frac{h_i^2}{6} M_i \right) \left( \frac{x_{i+1} - x}{h_i} \right) + \\ & + \left( u(x_{i+1}) - \frac{h_i^2}{6} M_{i+1} \right) \left( \frac{x - x_i}{h_i} \right), \quad x_i \leq x \leq x_{i+1},\ i = 0, \cdots, N-1 \end{aligned} \tag{3.2}$$

where $M_i = S''(x_i),\ i = 0, \cdots, N$.

The first derivative of $S(x)$ is given by

$$\begin{aligned} S'(x) = -M_i \frac{(x_{i+1} - x)^2}{2h_i} + M_{i+1} \frac{(x - x_i)^2}{2h_i} + \frac{u(x_{i+1}) - u(x_i)}{h_i} - \frac{(M_{i+1} - M_i)}{6} h_i, \\ x_i \leq x \leq x_{i+1},\ i = 0, \cdots, N-1. \end{aligned} \tag{3.3}$$

and the second derivative is

$$S''(x) = M_i \frac{(x_{i+1} - x)}{h_i} + M_{i+1} \frac{(x - x_i)}{h_i}. \tag{3.4}$$

For the one sided limit of the first derivative, from (3.3), we have

$$S'(x_i-) = \frac{h_{i-1}}{6} M_{i-1} + \frac{h_{i-1}}{3} M_i + \frac{u(x_i) - u(x_{i-1})}{h_{i-1}}, \tag{3.5}$$

and

$$S'(x_i+) = -\frac{h_i}{3} M_i - \frac{h_i}{6} M_{i+1} + \frac{u(x_{i+1}) - u(x_i)}{h_i}. \tag{3.6}$$

From (3.2) and (3.4), the functions $S(x)$ and $S''(x)$ are continuous on $\overline{\Omega}$ and for $S'(x)$ to be continuous at the interior nodes $x_i$, we have from (3.5)-(3.6), the following well-known 'continuity condition':

$$\frac{h_{i-1}}{6} M_{i-1} + \left( \frac{h_i + h_{i-1}}{3} \right) M_i + \frac{h_i}{6} M_{i+1} = \left( \frac{u_{i+1} - u_i}{h_i} \right) - \left( \frac{u_i - u_{i-1}}{h_{i-1}} \right), \; i = 1, \cdots, N-1. \tag{3.7}$$

This equation ensures the continuity of the first order derivative of the spline $S(x)$ at the interior nodes.

For obtaining second order approximations of the first order derivatives at the nodal points in terms of approximate values $u_i$ of $u(x)$ at $x_i$, we do the following.

Taking usual Taylor series expansion for $y$ around $x_i$, and neglecting the third and forth order terms, we get the following approximations for $u_{i+1}$ and $u_{i-1}$

$$u_{i+1} \simeq u_i + h_i u_i' + \frac{h_i^2}{2} u_i'', \tag{3.8}$$

$$u_{i-1} \simeq u_i - h_{i-1} u_i' + \frac{h_{i-1}^2}{2} u_i''. \tag{3.9}$$

Multiplying (3.9) by $h_i^2 / h_{i-1}^2$ and subtracting from (3.8), we get the following approximation for $u_i'$:

$$u_i' \simeq \frac{1}{h_i h_{i-1} (h_{i-1} + h_i)} [h_{i-1}^2 u_{i+1} + (h_i^2 - h_{i-1}^2) u_i - h_i^2 u_{i-1}] \tag{3.10}$$

Multiplying (3.9) by $h_i^2 / h_{i-1}^2$ and adding it to(3.8), we get the following approximation for$u_i''$:

$$u_i'' \simeq \frac{2}{h_i h_{i-1} (h_{i-1} + h_i)} [h_{i-1} u_{i+1} - (h_{i-1} + h_i) u_i + h_i u_{i-1}] \tag{3.11}$$

Also, we have

$$u_{i+1}' \approx u_i' + h_i u_i'', \tag{3.12}$$

$$u_{i-1}' \simeq u_i' - h_{i-1} u_i''. \tag{3.13}$$

Using the expressions for $u_i'$ and $u_i''$ from (3.10) and (3.11)respectively, and putting them in (3.12), we get the following approximation for $u_{i+1}'$:

$$u_{i+1}' \simeq \frac{1}{h_i h_{i-1} (h_i + h_{i-1})} [(h_{i-1}^2 + 2 h_i h_{i-1}) u_{i+1} - (h_{i-1} + h_i)^2 u_i + h_i^2 u_{i-1}] \tag{3.14}$$

Similarly, using the expressions for $u_i'$ and $u_i''$ from (3.10)and (3.11)respectively, and putting them in(3.13) , we get the following approximation for $u_{i-1}'$:

$$u_{i-1}' \simeq \frac{1}{h_i h_{i-1}(h_i + h_{i-1})}[-h_{i-1}^2 u_{i+1} - (h_{i-1} + h_i)^2 u_i - (h_{i-1} + h_i)^2 u_i - (h_i^2 + 2h_i h_{i-1})u_{i-1}] \quad (3.15)$$

Substituting

$$\varepsilon M_j = -a(x_j)u_j' + f(x_j), \quad j = i, i \pm 1, \quad (3.16)$$

in (3.7) and using (3.10),(3.14),(3.15) for the first order derivatives,we get the following system which gives the approximations $u_1, u_2, \cdots u_{N-1}$ of the solution $u(x)$ at $x_1, x_2, \cdots x_{N-1}$:

$$\begin{cases} \left[\frac{-3\varepsilon}{h_{i-1}(h_i + h_{i-1})} - \frac{2h_{i-1} + h_i}{2(h_i + h_{i-1})^2} a_{i-1} - \frac{h_i}{h_{i-1}(h_i + h_{i-1})} a_i + \frac{h_i^2}{2h_{i-1}(h_i + h_{i-1})^2} a_{i+1}\right] u_{i-1} \\ + \left[\frac{3\varepsilon}{h_i h_{i-1}} - \frac{1}{2h_i} a_{i-1} + \frac{h_i - h_{i-1}}{3h_i h_{i-1}} - \frac{1}{2h_{i-1}} a_{i+1}\right] u_i + \\ + \left[\frac{-3\varepsilon}{h_i(h_i + h_{i-1})} - \frac{h_{i-1}^2}{2h_i(h_i + h_{i-1})^2} a_{i-1} + \frac{h_{i-1}}{h_i(h_i + h_{i-1})} a_i + \frac{2h_i + h_{i-1}}{2(h_i + h_{i-1})^2} a_{i+1}\right] u_{i+1} = \\ \left[\frac{h_{i-1}}{2(h_i + h_{i-1})}\right] f_{i-1} + f_i + \left[\frac{h_i}{2(h_i + h_{i-1})}\right] f_{i+1}. \end{cases} \quad (3.17)$$

Now, using expressions (3.5) and (3.6) for approximation of the first derivative at boundary points, we obtain by following:

$$\left[-\frac{3\varepsilon a_0}{h_0^2} - \frac{a_0^2}{h_0} - \frac{a_0 a_1}{2h_0}\right] u_0 + \left[\frac{3\varepsilon a_0}{h_0^2} + \frac{a_0^2}{h_0} + \frac{a_0 a_1}{2h_0}\right] u_1 = \left[\frac{3\varepsilon}{h_0} + a_0\right] f_0 + \frac{a_0}{2} f_1 \quad (3.18)$$

Finally, the equations (3.17) and (3.18) constitute the system of linear algebraic equations, which gives the approximations $u_0, u_1, \cdots, u_N$ of the solution $u(x)$ at $x_0, x_1, \cdots, x_N$.

## 3.2 Piece-wise uniform Shishkin mesh

The cubic spline difference scheme derived in the previous subsection 3.1 is on variable meshes and it is a more general one. For SPPs one need finer mesh in the boundary layer regions and coarse mesh in the regular region which can be easily obtained *viz.* the piece-wise uniform Shishkin mesh. More precisely, the domain $\overline{\Omega}$ is divided into two subintervals as

$$\overline{\Omega} = [0, \sigma) \cup [\sigma, 1],$$

for some $\sigma$ such that $0 < \sigma \leq 1/2$. On the subinterval $[0, \sigma)$ a uniform mesh with $N/2$ mesh–intervals is placed, where $[\sigma, 1 - \sigma]$ has a uniform mesh with $N/2$ mesh intervals. It is obvious that the mesh is uniform when $\sigma = 1/2$, and it is fitted to the problem by choosing $\sigma$ be the following function of $N$, $\varepsilon$ and $\sigma_0$

$$\sigma = \min\left\{\frac{1}{2}, \sigma_0 \varepsilon \ln N\right\}, \quad (3.19)$$

where $\sigma_0 > 0$ is a constant. Further, we denote the mesh size in the regions $[0, \sigma)$ as $h^{(1)} = 2\sigma/N$, and in $[\sigma, 1]$ by $h^{(2)} = 2(1 - \sigma)/N$.

## 3.3 The hybrid scheme

As the use of the second-order cubic spline scheme derived in Section 3.1 in the whole domain gives satisfactory results only when the values of $\varepsilon$ and $N$ are compatible, *i.e.*, when $N^{-1} \leq \varepsilon$, and it is difficult to get uniform convergence. Therefore, taking advantage of its higher order convergence and uniform convergence of the mid-point scheme, we propose the following hybrid scheme, where cubic spline scheme is used only in the boundary layer region and mid-point scheme in the outer region.

More precisely, the hybrid scheme is given as

$$\begin{cases} L^N u_i \equiv r_i^- u_{i-1} + r_i^c u_i + r_i^+ u_{i+1} = q_i^- f_{i-1} + q_i^c f_i + q_i^+ f_{i+1}, \quad i = 1, \cdots, N/2 - 1, \\ L^N u_i \equiv r_i^- u_{i-1} + r_i^c u_i + r_i^+ u_{i+1} = f_{i+1/2}, \quad i = N/2, \cdots, N-1 \end{cases} \tag{3.20}$$

along with the following equations corresponding to the boundary points

$$\begin{cases} B_0^N u_i \equiv r_0^c u_0 + r_0^+ u_1 = q_0^c f_0 + q_0^+ f_1, \\ B_1^N u_i \equiv u_N = \beta, \end{cases} \tag{3.21}$$

for $i = 1, \cdots, N/2 - 1$

$$\begin{cases} r_i^- = \dfrac{-3\varepsilon}{h_{i-1}(h_i + h_{i-1})} - \dfrac{2h_{i-1} + h_i}{2(h_i + h_{i-1})^2} a_{i-1} - \dfrac{h_i}{h_{i-1}(h_i + h_{i-1})} a_i + \dfrac{h_i^2}{2h_{i-1}(h_i + h_{i-1})^2} a_{i+1} \\ r_i^c = \dfrac{3\varepsilon}{h_i h_{i-1}} - \dfrac{1}{2h_i} a_{i-1} + \dfrac{h_i - h_{i-1}}{3h_i h_{i-1}} - \dfrac{1}{2h_{i-1}} a_{i+1}; \\ r_i^+ = \dfrac{-3\varepsilon}{h_i(h_i + h_{i-1})} - \dfrac{h_{i-1}^2}{2h_i(h_i + h_{i-1})^2} a_{i-1} + \dfrac{h_{i-1}}{h_i(h_i + h_{i-1})} a_i + \dfrac{2h_i + h_{i-1}}{2(h_i + h_{i-1})^2} a_{i+1}; \\ q_i^- = \dfrac{h_{i-1}}{2(h_i + h_{i-1})}; \quad q_i^c = 1; \quad q_i^+ = \dfrac{h_i}{2(h_i + h_{i-1})}, \end{cases} \tag{3.22}$$

and for $i = N/2, \cdots, N-1$

$$\begin{cases} r_i^- = \dfrac{2\varepsilon}{h_{i-1}(h_i + h_{i-1})}; \quad r_i^c = \dfrac{-2\varepsilon}{h_{i-1}(h_i + h_{i-1})} - \dfrac{2\varepsilon}{h_i(h_i + h_{i-1})} - \dfrac{a_{i+1/2}}{h_i}; \\ r_i^+ = \dfrac{2\varepsilon}{h_i(h_i + h_{i-1})} + \dfrac{a_{i+1/2}}{h_i}; \end{cases} \tag{3.23}$$

and

$$\begin{cases} r_0^c = -\dfrac{3\varepsilon a_0}{h_0^2} - \dfrac{a_0^2}{h_0} - \dfrac{a_0 a_1}{2h_0}; \quad r_0^+ = \dfrac{3\varepsilon a_0}{h_0^2} + \dfrac{a_0^2}{h_0} + \dfrac{a_0 a_1}{2h_0}; \\ q_0^c = \dfrac{3\varepsilon}{h_0} + a_0; \quad q_0^+ = \dfrac{a_0}{2}; \end{cases} \tag{3.24}$$

The tri-diagonal system of linear algebraic equations (3.20-3.21) can be solved by any existing codes.

## 3.4 Truncation Error

Here, we derive the truncation error for the difference scheme proposed in Section 3.3. The discrete stability analysis, and error estimates will be carried out in our forthcoming article [4].

For $i = 1, \cdots, N/2 - 1$, the truncation error of the hybrid scheme is given by

$$\tau_{i,u} = [r_i^- u(x_{i-1}) + r_i^c u(x_i) + r_i^+ u(x_{i+1})] - [q_i^- f(x_{i-1}) + q_i^c f(x_i) + q_i^+ f(x_{i+1})]. \tag{3.25}$$

Using the differential equation (1.1) for $f$ in the above expression, we get

$$\begin{aligned}\tau_{i,u} = [r_i^- u(x_{i-1}) + r_i^c u(x_i) + r_i^+ u(x_{i+1})] - [q_i^- (\varepsilon u''(x_{i-1}) + a_{i-1} u'(x_{i-1})) +& \\ + q_i^c (\varepsilon u''(x_i) + a_i u'(x_i)) + q_i^+ (\varepsilon u''(x_{i+1}) + a_{i+1} u'(x_{i+1}))].&\end{aligned} \tag{3.26}$$

Now, making use of the Taylor series expansion, we have

$$u(x_{i-1}) = u(x_i) - h_{i-1} u'(x_i) + \frac{h_{i-1}^2}{2!} u''(x_i) - \frac{h_{i-1}^3}{3!} u^{(iii)}(x_i) + \frac{h_{i-1}^4}{4!} u^{(iv)}(x_i) + \cdots,$$

and

$$u(x_{i+1}) = u(x_i) + h_i u'(x_i) + \frac{h_i^2}{2!} u''(x_i) + \frac{h_i^3}{3!} u^{(iii)}(x_i) + \frac{h_i^4}{4!} u^{(iv)}(x_i) + \cdots$$

Using the values of $u(x_{i-1})$, $u(x_{i+1})$ in (3.26), we have

$$\tau_{i,u} = T_{0,i} u(x_i) + T_{1,i} u'(x_i) + T_{2,i} u''(x_i) + T_{3,i} u^{(iii)}(x_i) + T_{4,i} u^{(iv)}(x_i) + \text{h.o.t.}, \tag{3.27}$$

where

$$\begin{aligned}
T_{0,i} &= r_i^- + r_i^c + r_i^+, \\
T_{1,i} &= -h_{i-1} r_i^- + h_i r_i^+ - (q_i^- a_{i-1} + q_i^c a_i + q_i^+ a_{i+1}), \\
T_{2,i} &= \frac{h_{i-1}^2}{2!} r_i^- + \frac{h_i^2}{2!} r_i^+ + \varepsilon (q_i^- + q_i^c + q_i^+) - \left(h_{i-1} q_i^- a_{i-1} - h_i q_i^+ a_{i+1}\right), \\
T_{3,i} &= -\frac{h_{i-1}^3}{3!} r_i^- + \frac{h_i^3}{3!} r_i^+ - \varepsilon (q_i^- h_{i-1} - q_i^+ h_i) + \left(\frac{h_{i-1}^2}{2!} q_i^- a_{i-1} + \frac{h_i^2}{2!} q_i^+ a_{i+1}\right), \\
T_{4,i} &= \frac{h_{i-1}^4}{4!} r_i^- + \frac{h_i^4}{4!} r_i^+ + \varepsilon (q_i^- \frac{h_{i-1}^2}{2!} + \frac{h_i^2}{2!} q_i^+) - \left(\frac{h_{i-1}^3}{3!} q_i^- a_{i-1} - \frac{h_i^4}{4!} q_i^+ a_{i+1}\right).
\end{aligned}$$

It can be easily seen that

$$T_{0,i} = T_{1,i} = T_{2,i} = T_{3,i} = 0, \quad T_{4,i} = -3\varepsilon \left(\frac{h_i^3 + h_{i-1}^3}{h_i + h_{i-1}}\right) \left[\frac{1}{4!} - \frac{1}{2!6}\right].$$

Thus, we have

$$\tau_{i,u} = -3\varepsilon \left(\frac{h_i^3 + h_{i-1}^3}{h_i + h_{i-1}}\right) \left[\frac{1}{4!} - \frac{1}{2!6}\right] u^{(iv)}(x_i) + O(N^{-3}). \tag{3.28}$$

For $i = N/2, \cdots, N - 1$, we can proceed in similar manner to show that

$$\tau_{i,u} = -\varepsilon \left(\frac{h_i - h_{i-1}}{3}\right) u^{(iii)}(x_i) + \frac{2\varepsilon}{4!} \left(\frac{h_i^3 + h_{i-1}^3}{h_i + h_{i-1}}\right) u^{(iv)}(x_i) + O(N^{-3}). \tag{3.29}$$

The truncation error at the boundary point $x_0$ is given by

$$\tau_{0,u} = r_0^c u(x_0) + r_0^+ u(x_1) - q_0^c f_0 - q_0^+ f_1 \tag{3.30}$$

Again using (1.1), and the Taylor series expansion for $u(x_1)$, the truncation error at $x_0$ can be given as

$$\tau_{0,u} = -3\varepsilon h_0^2 \left(\frac{1}{4!} - \frac{1}{2!6}\right) u_0^{(iv)}(x_0) + O(N^{-3}). \tag{3.31}$$

Using the bounds of the solution obtained in Section 2, one can prove the following proposition.

**Proposition 3.1** *Let $u(x)$ and $u_i$ be respectively the solutions of (1.1-1.2) and (3.20-3.21). Then, the local truncation error satisfies the following bounds:*

$$\begin{aligned}
|\tau_{i,u}| &\leq CN^{-2}\sigma_0^2 \ln^2 N, \quad \textit{for} \quad 0 \leq i < N/2, \\
|\tau_{i,u}| &\leq C(N^{-2}\varepsilon + N^{-\alpha\sigma_0}), \quad \textit{for} \quad N/2 \leq i \leq N-1, \quad \textit{and} \quad h^{(2)} \geq \sqrt{\varepsilon}, \\
|\tau_{i,u}| &\leq C(N^{-1}\varepsilon + N^{-\alpha\sigma_0}), \quad \textit{for} \quad N/2 \leq i \leq N-1 \quad \textit{and} \quad h^{(2)} < \sqrt{\varepsilon}.
\end{aligned}$$

# 4 Numerical Experiments

To show the accuracy of the present method, here we have implemented it on the following example. The results are presented in the form of tables with maximum point–wise errors and rate of convergent, and figures showing the maximum point-wise error in the loglog scale.

**Example 4.1** *Consider the following convection-diffusion Neumann BVP:*

$$\begin{aligned}
\varepsilon u''(x) + (1+2x)^2 u'(x) = -(x^3 + \exp(-x)), \quad x \in (0,1) \\
-u'(0) = -1, \quad u(1) = 0.
\end{aligned}$$

To calculate the maximum point-wise error and rate of convergence, we use the double mesh principle. We calculate the numerical solution $U^N$ on $\Omega^N$ and the numerical solution $\widetilde{U}^N$ on the mesh $\widetilde{\Omega}^N$ where the transition parameter is now given by

$$\widetilde{\sigma} = \min\left\{\frac{1}{2}, \sigma_0 \varepsilon \ln(N/2)\right\}.$$

Define the double mesh differences to be

$$G_\varepsilon^N = \max_{x_j \in \overline{\Omega}_\varepsilon^N} |U^N(x_j) - U^{2N}(x_j)|, \quad \text{and} \quad G^N = \max_\varepsilon G_\varepsilon^N,$$

where $U^N(x_j)$ and $U^{2N}(x_j)$ respectively denote the numerical solutions obtained using $N$ and $2N$ mesh intervals. Further, we calculate the parameter-robust orders of convergence as

$$p = \log_2\left(\frac{G_\varepsilon^N}{G_\varepsilon^{2N}}\right) \quad \text{and} \quad p_{uni} = \log_2\left(\frac{G^N}{G^{2N}}\right).$$

The numerical results for the present example are presented in Table 1. The maximum point-wise errors are plotted in Figure 1.

## 5 Discussion

In this paper, we proposed a hybrid scheme for the numerical solution of convection dominated two-point boundary-value problems. This scheme consists of both the cubic spline and mid-point schemes. This scheme is applied on a layer resolving Shishkin mesh. In the boundary layer region, where the mesh is fine, the cubic spline scheme is used. In the outer region, *i.e.*, in the coarse mesh region, the mid-point scheme is used. Truncation errors are derived. The numerical results reveal the second-order $\varepsilon$-uniform convergence of the scheme throughout the domain.

## Acknowledgements

This work is supported by the Department of Science and Technology, Government of India under research grant SR/S4/MS:318/06.

## References

[1] E.P. Doolan, J.J.H. Miller, and W.H.A. Schildres, *Uniform Numerical Methods for Problems with Initial and Boundary Layers*, Boole Press, Dublin, 1980.

[2] M.K. Kadalbajoo and R.K. Bawa, *Variable Mesh Difference Scheme Cubic for Singularly Perturbed Boundary-Value Problems Using Splines*, Jl. Optim. Theory. and Appl., **9** (1996) 405–416.

[3] J.J.H. Miller, E. O'Riordan, and G.I. Shishkin, *Fitted Numerical Methods for Singular Perturbation Problems*, World Scientific, Singapore, 1996.

[4] S. Natesan and R.K. Bawa, *Second-Order Parameter-Uniform Error Estimate for Convection Dominated Two-Point Boundary-Value Problems*, Working Paper, 2007.

[5] S. Natesan and N. Ramanujam, *'Shooting Method' for the Solution of Singularly Perturbed Two-Point Boundary-Value Problems Having Less Severe Boundary Layers*, *Appl. Math. Comput.*, **133** (2002) 623–641.

[6] S. Natesan, J. Vigo-Aguiar, and N. Ramanujam, *A Numerical Algorithm for Singular Perturbation Problems Exhibiting Weak Boundary Layers*, Comput. Math. Appl. **45** (2003) 469–479.

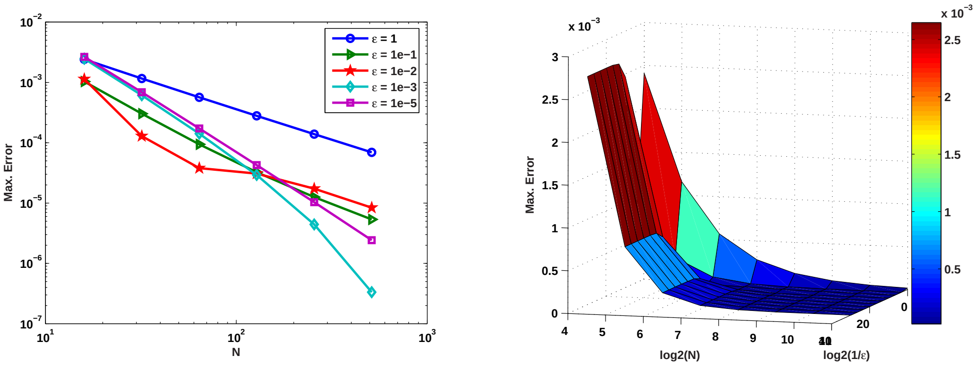
[7] R.E. O'Malley, *Singular Perturbation Methods for Ordinary Differential Equations*, Springer, New York, 1991.

[8] H.-G. Roos, M. Stynes, and L. Tobiska, *Numerical Methods for Singularly Perturbed Differential Equations*, Springer, Berlin, 1996.

[9] J. Vigo-Aguiar and S. Natesan, *A Parallel Boundary Value Technique for Singularly Perturbed Two-Point Boundary Value Problems*, The Journal of Supercomputing **27** (2004) 195–206.

Table 1: *Maximum point-wise errors $E_\varepsilon^N$, rate of convergence $p$ and $\varepsilon$ uniform errors $E^N$ for Example 4.1.*

| $\varepsilon$ | Number of mesh points $N$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 |
| $10^{-0}$ | 2.4092e-3<br>1.0558 | 1.1589e-3<br>1.0334 | 5.6619e-4<br>1.0179 | 2.7960e-4<br>1.0093 | 1.3891e-4<br>1.0047 | 6.9227e-5<br>1.0024 | 3.4557e-5<br>1.0012 | 1.7264e-5 |
| $10^{-2}$ | 1.1338e-3<br>3.1362 | 1.2896e-4<br>1.7622 | 3.8017e-5<br>0.3032 | 3.0810e-5<br>0.8323 | 1.7304e-5<br>1.0454 | 8.3840e-6<br>1.1256 | 3.8425e-6<br>1.1593 | 1.7205e-6 |
| $10^{-4}$ | 2.6484e-3<br>1.9434 | 6.8858e-4<br>1.9959 | 1.7263e-4<br>2.0188 | 4.2598e-5<br>2.0437 | 1.0332e-5<br>2.0910 | 2.4250e-6<br>2.1986 | 5.2827e-7<br>2.5000 | 9.3387e-8 |
| $10^{-6}$ | 2.6687e-3<br>1.9363 | 6.9728e-4<br>1.9837 | 1.7630e-4<br>1.9961 | 4.4196e-5<br>1.9994 | 1.1054e-5<br>2.0006 | 2.7623e-6<br>2.0016 | 6.8981e-7<br>2.0033 | 1.7206e-7 |
| $10^{-8}$ | 2.6689e-3<br>1.9363 | 6.9737e-4<br>1.9836 | 1.7634e-4<br>1.9959 | 4.4212e-5<br>1.9990 | 1.1061e-5<br>1.9997 | 2.7657e-6<br>1.9999 | 6.9145e-7<br>2.0000 | 1.7286e-7 |
| $10^{-10}$ | 2.6689e-3<br>1.9363 | 6.9737e-4<br>1.9836 | 1.7634e-4<br>1.9959 | 4.4212e-5<br>1.9990 | 1.1061e-5<br>1.9997 | 2.7657e-6<br>1.9999 | 6.9146e-7<br>1.9999 | 1.7288e-7 |
| $10^{-12}$ | 2.6689e-3<br>1.9363 | 6.9737e-4<br>1.9836 | 1.7634e-4<br>1.9959 | 4.4212e-5<br>1.9990 | 1.1061e-5<br>1.9997 | 2.7657e-6<br>1.9999 | 6.9146e-7<br>1.9999 | 1.7288e-7 |
| $E^N$<br>$p_{uni}$ | 2.6689e-3<br>1.2035 | 1.1589e-3<br>1.0334 | 5.6619e-4<br>1.0179 | 2.7960e-4<br>1.0093 | 1.3891e-4<br>1.0047 | 6.9227e-5<br>1.0024 | 3.4557e-5<br>1.0012 | 1.7264e-5 |



*(a) N Vs. Maximum Error.*

*(b) N, $\varepsilon$ Vs. Maximum Error.*

Figure 1: *Plots of Maximum Point-wise Error for Example 4.1.*

# Deadlock Detection Algorithm for Grid Resource Management

**Seema Bawa, Anju Sharma**
*Computer Science & Engineering Department*
*Thapar University, Patiala Punjab- 147004, India*
{sbawa, anjusharma }@tiet.ac.in

## Abstract

Distributed environments are touching newer heights; becoming more useful, popular and complex with the emergence of technologies like peer-to-peer computing, autonomic computing, pervasive computing and Grid computing. Grid computing is emerging as the new paradigm to provide collaborative and resource sharing environment over multiple geographically distributed environments. These technologies aim to enable large resource sharing. Due to the heterogeneous and dynamic nature of resources, we need to develop highly distributed and extensible framework for resource management. This paper presents an algorithm for Grid Resource Management based on deadlock detection mechanism used in operating systems. The proposed algorithm aims to achieve the major objectives like: fairness of resource distribution among nodes, scalability, flexibility and simplicity. A prototype implementation of the algorithm is also presented in the paper.

***Keyword:*** *Distributed environments, Grid computing, Resource Management, Resource Allocation Graph (RAG)*

## 1. Introduction

Grid infrastructures and computing environments have progressed significantly in the past few years [1]. Grid is a distributed system involving heterogeneous resources located in different geographical domains that are potentially managed by different organizations. In Grid systems many users run their application at the same time and compete for the finite number of resources.

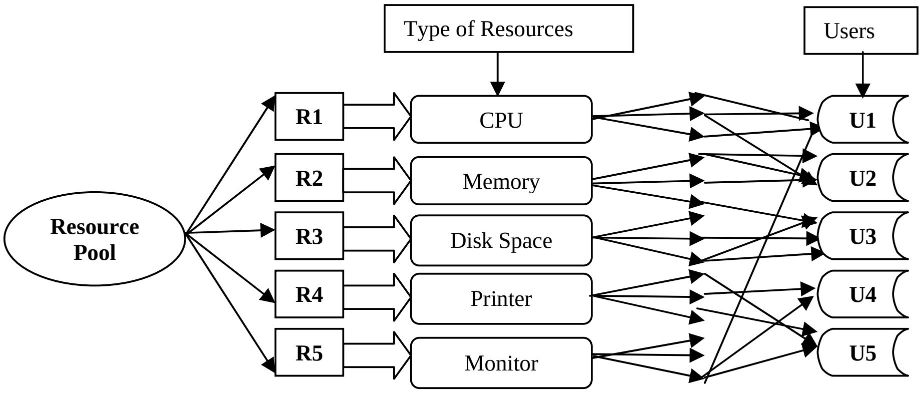


**Figure 1. Schematic view of Resource Pool**

The key problem here is to find the node that offer the desired type and amount of resources like cpu, memory, disk space etc. – required by a user to start and complete the execution of its application. Owing to the heterogeneous environment of the Grid, it has large set of resources. These resources can be represented by resource pool as shown in the Figure 1.

These resources are represented as R1, R2, R3, R4 and R5. Categorization may be done as CPU, Memory, Printer, Disk Space, and Monitor. A user may request as many resources as it requires to carry out its designated task. Apparently, the number of resources requested may not exceed the total number of resources available in the system. For example, a process can not request three printers if the system has only two. If a system has two CPUs, then the resource type CPU has two instances. If a user requests an instance of a resource type, the allocation of any instance of the type will satisfy the request. If it does not, then the instances are not identical, and the resource type classes have not been defined properly. For example, a system may have two printers; both may be defined to be in the same resource class, if no one cares which printer prints the output. However, if one printer is on the ninth floor and other is in the basement, then the people of ninth floor may not see both printers as equivalent, and separate resource classes may need to be defined for each printer.

Deadlock is one of the most serious problems in multitasking concurrent programming systems [2]. Consider a simple case study of an organization: In an organization thousands of employees work for a shift and they need some resources for specific amount of time. Suppose a user U1 uses the memory for half a day and in between this half day user U4 need, the same memory. So, to use the memory user U4 has to wait or U1 and U4 may have to compete for the memory at the same time. Once they are competing a condition occurs which is known as deadlock. Deadlock can be described through a Resource Allocation Graph (RAG) [3]. It can be shown that, if the graph contains no cycles, then no process in the system is deadlock. If the graph does not contain a cycle, then a deadlock may exist. Suppose that process P3 requests an instance of resource type R2. Since no resource instance is currently available, a request edge P3→ P2 is added to the graph (Figure 2). At this point, two minimal cycles exist in the system: $P_1 \rightarrow R_1 \rightarrow P_2 \rightarrow R_3 \rightarrow P_3 \rightarrow R_2 \rightarrow P_1$ and $P2 \rightarrow R_3 \rightarrow P_3 \rightarrow R_2 \rightarrow P_2$. Processes $P_{1,}$ $P_2$, $P_3$ are deadlocked.

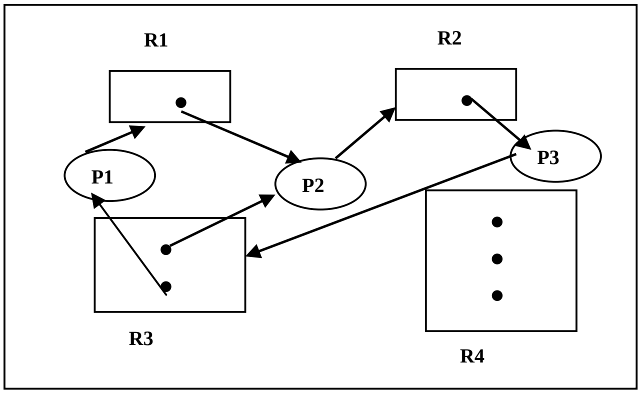


**Figure 2. Resource Allocation Graph (RAG) with Deadlock**

Deadlock is constant problem often offsetting the advantages of resource sharing [4]. A set of threads is deadlocked if each thread is waiting for an event that can only be generated by another thread in the set [5]. Detecting the deadlock is one of the major problems in Grid Computing [6]. When a deadlock is detected message is propagated to whole cycle of the deadlock as described in the Section 3.1. Some algorithms for deadlock detection are described in following sections.

## 2. Deadlock Detection Algorithms

### *2.1 Assessment Algorithm*

This algorithm is used for finding out whether or not a system is in assessing state. It can be described as follows:

```
I.   Initialize Start := Open and Finish [i] := False
             for i =1,2,…,n.
II.  Let Start and Finish be vectors of length v and n, respectively.
III. Find an i such that both
         a. Finish[i] = False
         b. $Require_i$ <= Start
     If no such i exists, go to step IV
IV.  Start := Start + Allotment;
       Finish[i] := True;
       Go to step II.
V.   If Finish[i] = True for all i, then calculate that the system is in safe state.
```

This algorithm may require an order of v*$n^2$ operations to decide whether the state is safe.

### *2.2 Demand Resource Algorithm*

Let $Request_i$ be the request vector for user $P_i$. If $Request_i [j]$ = k, then user $P_i$ wants k instances of resource type $R_j$. When a request for resources is made by user $P_i$, the following actions are taken:

a. If $Request_i$ <= $Require_i$, go to step b. Otherwise, raise an error condition, since the user has exceeded its maximum.
b. If $Request_i$ <= $Require_i$, go to step c. Otherwise $P_i$ must wait, since resources are not available.
c. Have the system pretend to have allotment the request resources to user *Pi* by modifying the state as follows:

$$Open :=Open – Request_i;$$

$$Allotment := Allotment_i + Request_{i;}$$

$$Require_i := Require_i - Request_{i;}$$

If the resulting resource allocation state is safe, the transaction is completed and user $P_i$ is allocated the resources. However, if the new state is not assess, then $P_i$ must wait for $Request_i$ and the old resource-allocation state is restored.

## 3. Proposed Approach and Data Structures for the Algorithm

Based on the above algorithms the proposed approach and data structures for the algorithms are as follows. When a new user enters into the system, it must declare the maximum number of instances of each resource type that it may need. This number may not exceed the total number of resources in the system. When a user requests a set of resources, the system must determine whether the allotment of these resources will leave the system in assess state. If it will, the resources are allocated; otherwise, the user must wait until some other user releases enough resources.

Several data structures must be maintained to implement the algorithm. These data structures encode the state of the resource-allocation system. Let n be the number of users in the system and v be the number of resource types. We need the following data structures.

- ***Open:*** A vector of length v indicates the number of available user of each type. If available [Resource j] = k, there are k instances of resource type Rj open.
- ***Max:*** An n*v matrix defines the maximum demand of each user. If Max [i, j] = k, then user Qi may request at most k instances of resource type Rj.
- ***Allotment:*** An n*v matrix defines the number of resources of each type currently allocated to each user. If Allocation [i,j] = k, then user Qi is currently allocated k instances of resource type Rj.
- ***Require:*** An n*v matrix indicates the remaining resource need of each user. If require [i,j] = k, then user Qi may need k more instances of resource type Rj to complete its task. Note that Require [i,j] = Max [i,j] – Allotment [i,j].

### *3.1 Type of messages*

- ***Inadequate number of resources:*** Once the user request for resources, if resources are not available then popup message is generated “inadequate number of resources”.
- ***Over allocation of resources:*** If available resources are less then or equal to zero then ‘over allocation of resources’ message is popped up.
- ***Failed to satisfy the Customer:*** the message ‘failed to satisfy’ the customer is popped up when the requested resources are not available.
- ***Satisfy the Customer:*** If not failed to satisfy the customer then the message generated is “satisfy the customer”.
- ***State is Safe:*** Once the customer is satisfied the ‘state is safe’**.**
- ***State is Unsafe:*** If the Customer is not satisfied the ‘state is unsafe’.

## 4. Implementation Details

We have considered the standard C library function, malloc. Initially implementation of this algorithm is done using ‘C’ programming language. Five cases are considered i.e. ‘failed to allocate the existing resources’, ‘failed to allocate the available resources’, ‘state is safe’, ‘state is unsafe’, and ‘overallocation of resource’. In order to do this we need to declare the argument count (argc) and the array of strings (argv). We have used two 1-dimensional arrays and two 2-dimensional arrays, as pointers, and use malloc call later to make them into arrays with actual storage. The pseudocode of the proposed algorithm is given below.

### *4.1 The pseudocode for the proposed Deadlock Detection Algorithm*

```
procedure Deadlock Detection Algorithm (nr, nc, argc, argv)
        {Enter the number of resource ,number of customer for the computation, argc (argument
count), argv (array of strings)}
      var
        nr, nc, ii, jj, existing_resourrces, available_resources, current_allocation,
requested_resources, done=0, trials=0, freed=0
        begin
   Step 1:    nr, nc; { PLEASE SPECIFY THE NUMBER OF
              RESOURCES AND NUMBER OF CUSTOMERS}
   Step 2:    if (! (existing_resources)  || (! (available_resources) ||(! (current_allocation) ||
                  (! (requested_resources) then               //if Resources are not present
             failure: “inadequate number of resources”        //Message pop up
   Step 3:    for ii :=0 to< nr do
                  how many of existing_resources exists?
              available_resources [ ]= existing_resources [ ]
   Step 4:    for ii:=0 to < nc
                  {tell me about the resources for customers}
                  {how many of resources are currently allocated to customer}
                      available_resources[ ] -= current_allocation[ ] [ ]
                      //available_resources =   available_resources – current_allocation
                  if available_resources [ ] <  0
                  {overallocation of resources}
                  exit (1)
                  else
                  how many maximum of resources would customer like to have?
                  Requested_resource[ ] [ ] -= current_allocation [ ] [ ]
                  //Requested_resources = Requested_resources-current_allocation
   Step 5:     while (!done)
                      do
                      trials =0;
                      freed = 0; do begin
                      for ii := 1 to <nc
                      if requested_resources
                       {trials++;
                         failed = 0;
                      for jj := 1 to < nr
                      if requested_resources [ ] [ ] > available_resources [ ] [ ]
                      {failed = 1
                      failed to satisfy the customer}
                      break;
   Step 6:             if (!failed) {
                      { free++;
                      for jj := to < nr
                      { available_resources [ ] += current_allocation [ ] [ ]
                      //available_resources = available_resources + current_allocation
                      }
                      free (requested_resources) ;
                      requested_resources [ ] = NULL;
```

```
                    {satisfying customer}
Step 7:            if (trials = = 0) {
                    {the state is safe}
               exit (0);
Step 8:          if (freed = = 0)
                    {the state is unsafe}
                    exit(1);
end; {Deadlock Detection Algorithm}
```

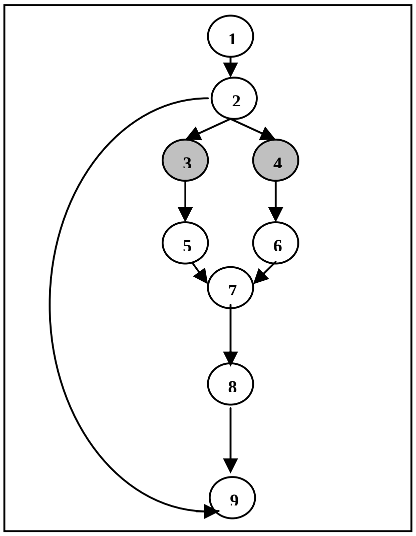


**Figure 3. An Example of Execution of Algorithm**

```
1.  User Initiates the Algorithm
2.  If  (!Available_Resources) || (!Existing_Resources ) ||
    (!Current_Allocation) || (!Requested_Resources) then
    "EXIT"
    Message pop up, "inadequate number of resources"
3.  For the number_of_resources
          Available_Resources[ ] = Existing_Resourecs [ ]
4.  For the number_of_customers
          If available_resources<0
    Msg "Overallocation of resources"
    Else
    How many maximum of resources would customer like to
     have?
    Requested_Resources [ ]-= Current_Allocation [ ]
5.  While (!done)
    If Requested_Resources [ ] > Available_Resources [ ]
    Msg "Failed to satisfy the customer"
6.  if(!failed)
    Messge pop up,  "satisfying the customer"
7.  State is Safe
8.  State is Unsafe
9.  Exit
```

**Figure 4. Detail Description of Execution of the Algorithm Shown in Figure 3.**

## 5. Conclusion

We have proposed a Deadlock Detection Algorithm, in Grids which is well suited for applications such as Resource Management in Grids. Deadlock Detection algorithm is capable of meeting the identified requirements and the approach is workable. By implementing the Algorithm proposed above in Grid Computing, we can establish a Fairness, Scalability, Flexibility and Simplicity in Resource Management. Figure 3. shows example of execution of the algorithm. Figure 4. Shows the detail description of the execution of the algorithm.

### ACKNOWLEDGEMENT

The authors are deeply indebted to Thapar University, Patiala (India). Words are not sufficient to acknowledge our gratitude towards TU Grid Group actively working at the Centre of Excellence in Grid Computing (CoEGC) the Thapar Deemed University, Patiala for providing us motivation and base for carrying out this work. The free and open discussions and brainstorming sessions held during weekly meetings there had been extremely helpful in this endeavor.

## References:

[1] http://arxiv.org/pdf/cs.CE/0301018

[2] E. Knapp, “Deadlock Detection in Distributed Database Systems,” ACM Computing Surveys, Vol. 19, No. 4, pp. 303-327, Dec. 1987.

[3] Galvin Gagna. “Operating System Concepts”, Edition 7.

[4] M. Singhal, “Deadlock Detection in Distributed Systems,” IEEE Computer, Vol. 22, pp. 37–48, 1989.

[5] Tanenbaum. “Modern Operating Systems”, Prentice Hall, 2001.

[6] Nacer Farajzadeha, Mehdi Hashemzadeha, Morteza Mousakhania, Abolfazl T. Haghighat,” An Efficient Generalized Deadlock Detection and Resolution Algorithm in Distributed Systems” Proceedings Fifth International Conference on Computer and Information Technology (CIT’05) 2005 IEEE.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE2007*
*Chicago, 20-23 June 2007*

# Analysis of Computational Grid Environments

Seema Bawa
*Department of Computer Science & Engineering*
*Thapar University, Patiala*
sbawa@tiet.ac.in

## Abstract

The Grid is computing and data management infrastructure, which provides the electronic underpinning for a global society in business, government and research. To realize the global grid environment we need a standard architecture to cope up the heterogeneity and the interoperatability The middleware component is of extreme importance while building the Grid environment, as the majority of the Grid specific features are implemented using the Grid middleware. The grid middlewares act as glue between the grid resource components and the application components. The Globus, Condor, Nimrod/g, Alchemi are the popular middlewares. The comparative analysis of these middlewares let the grid community know the choices available.

*Keywords: Grid computing, Middleware, Globus, Nimrod/g, Alchemi, Condor*

## 1. Introduction

Grids are becoming platforms for high-performance and distributed computing [7]. A Grid benefits users by permitting them to acces0s heterogeneous resources, such as machines, data, people and devices that are distributed geographically and organizationally. They allow users to execute compute intensive problems whose computational requirements cannot be satisfied by a single machine. Grid Computing has emerged as a new and important field and can be visualized as an enhanced form of Distributed Computing. With the advent of new technology, it has been realized that paralleling sequential applications could yield faster results and sometimes at a lower cost [2].

Grid computing is the next generation IT infrastructure that promises to transform the way organizations and individuals compute, communicate and collaborate [8][9]. It offers untapped processing cycles from networks of computers spanning vast geographical boundaries. Sharing in a Grid is not just a simple sharing of files but of hardware, software, data, and other resources [6]. Thus a complex yet secure sharing is at the heart of the Grid.

## 2. Grid Components

The various components s that are necessary to form a Grid are as follows.

### 2.1 Grid Fabric

This consists of all the globally distributed resources that are accessible from anywhere on the Internet. These resources could be computers (such as PCs or Symmetric Multi-Processors) running a variety of operating systems (such as UNIX or Windows), storage devices, databases, and special scientific instruments such as a radio telescope or particular heat sensor [17].

### 2.2 Core Grid middleware

This offers core services such as remote process management, co-allocation of resources, storage access, information registration and discovery, security, and aspects of Quality of Service (QoS) such as resource reservation and trading. These services abstract the complexity and heterogeneity of the fabric level by providing a consistent method for accessing distributed resources.

### 2.3 User-level Grid middleware

User level middleware utilizes the interfaces provided by the low-level middleware to provide higher-level abstractions and services. This includes application development environments, programming tools, and resource brokers for managing resources and scheduling application tasks for execution on global resources.

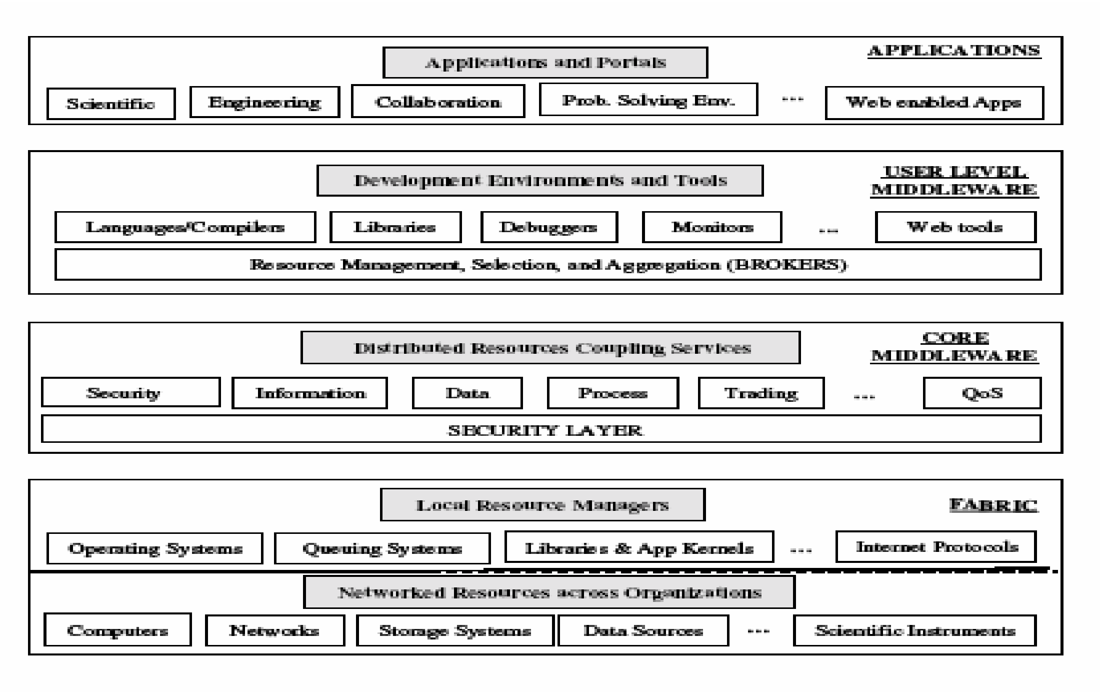


*Fig 2.2 Grid components [17]*

### 2.4 Grid applications and Portal

Grid applications are typically developed using Grid-enabled languages and utilities such as HPC++ or MPI. An example application, such as parameter simulation or a grand-challenge problem, would require computational power, access to remote data sets, and may need to interact with scientific instruments. Grid portals offer Web-enabled application services, where users can submit and collect results for their jobs on remote resources through the Web [17].

## 3. Grid Middleware

Grids have *middleware stacks*, which are a series of cooperating programs, protocols and agents designed to help users access the resources of a Grid[28]. Grid Middleware refers to the security, resource management, data access, instrumentation, policy, accounting, and other services required for applications, users, and resource providers to operate effectively in a Grid environment. Middleware acts as a sort of 'glue' which binds these services together.

Formally Grid middleware can be define [23] as:

"***A mediator layer that provide a consistent and homogeneous access to resources managed locally with different syntax and access methods***"

Till today several implementation of Grid middleware have been achieved. These implementations have well identified basic services. These middleware implementations are now moving their focus from proprietary/ adhoc solutions to standard based solutions. The brief overview of the some popular middleware's Globus, Alchemi, and Condor is discussed in this section.

### 3.1 Globus

The Globus [27] toolkit is designed to enable people to create computational Grids. It has been developed over several years chiefly at the Argonne National Laboratory Illinois USA. As an open source project any person can download the software, examine it, install it and hopefully improve it. By this constant stream of comments and improvements, new versions of the software can be developed with increased functionality and reliability. In this way the Globus project itself will be on going with constant evolution of the toolkit [27].

### 3.1.1 Globus Pyramids

Globus Toolkit has three pyramids of support built on top of a security infrastructure, as illustrated. They are:

- Resource management
- Data management
- Information services

All of these pyramids are built on top of the underlying Grid Security Infrastructure (GSI). This provides security functions, including single/mutual authentication, confidential communication, authorization, and delegation.

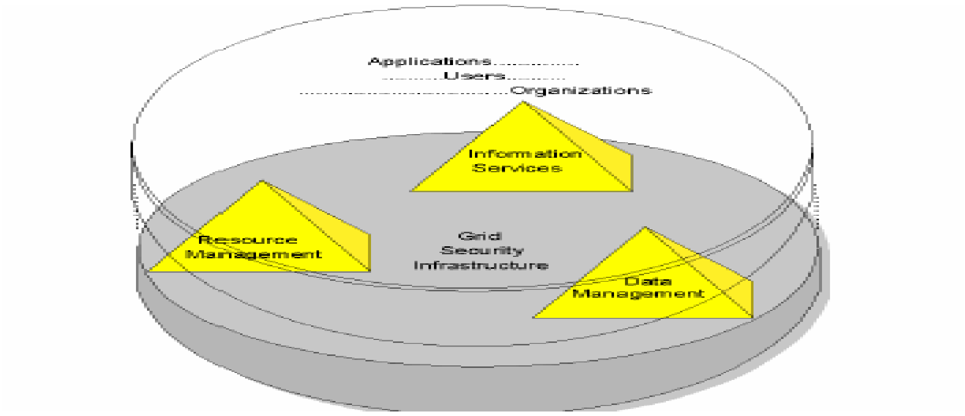

Applications.............
..........Users..........
....................................Organizations
Information Services
Resource Management
Grid Security Infrastructure
Data Management

*Fig3.1 Globus pyramids [24]*

### Resource management

The resource management pyramid provides support for:

- Resource allocation
- Submitting jobs: Remotely running executable files and receiving results
- Managing job status and progress

### Information services

The information services pyramid provides support for collecting information in the Grid and for querying this information, based on the Lightweight Directory Access Protocol (LDAP).

### Data management

The data management pyramid provides support to transfer files among machines in the Grid and for the management of these transfers. [24]

### 3.1.2 Components of Globus Toolkit

For each pyramid previously presented, Globus provides a component to implement resource management, data management, and information services. The various Components are:
GRAM, MDS, GridFTP and GSI.

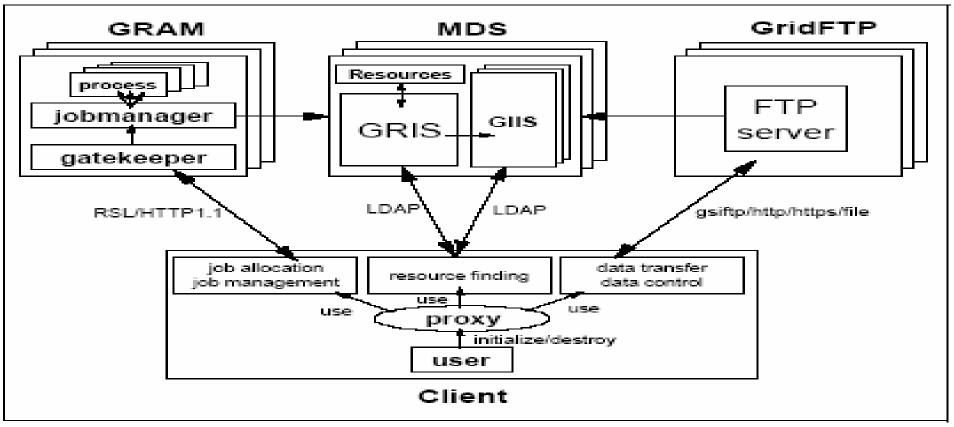


*Fig 3.2 System overview of Globus [24]*

#### 3.1.2.1 Grid Security Infrastructure (GSI)

GSI provides elements for secure authentication and communication in a Grid. The infrastructure is based on the SSL protocol (Secure Socket Layer), public key encryption, and x.509 certificates. For a single sign-on, Globus add some extensions on GSI. It is based on the Generic Security Service API, which is a standard API promoted by the Internet Engineering Task Force (IETF).
These are the main functions implemented by GSI are:
***Single/mutual authentication, Confidential communication, Authorization, Delegation***

#### 3.1.2.2 Grid Resource Allocation Manager (GRAM)

GRAM is the module that provides the remote execution and status management of the execution. When a client submits a job, the request is sent to the remote host and handled by the gatekeeper daemon located in the remote host. Then the gatekeeper creates a job manager to start and monitor the job. When the job is finished, the job manager sends the status information back to the client and terminates [24]. GRAM contains the following elements:
***globusrun command, Resource Specification Language (RSL), gatekeeper daemon, job manager, forked process, Global Access to Secondary Storage (GASS)***

#### 3.1.2.3 Monitoring and Discovery Service (MDS)

MDS provides access to static and dynamic information of resources. Basically, it contains the following components:
***Grid Resource Information Service (GRIS), Grid Index Information Service (GIIS), Information Provider, MDS client***

#### 3.1.2.4 GridFTP

GridFTP provides a secure and reliable data transfer among Grid nodes [24].

## 3.2 Alchemi

Alchemi is an open-source .Net based Enterprise Grid computing framework developed by researchers at the GRIDS lab, in the Computer Science and Software Engineering Department at the University of Melbourne, Australia. It allows you to painlessly aggregate the computing power of networked machines into a virtual supercomputer and to develop applications to run on the grid with no additional investment and no discernible impact to users. It has been designed with the primary goal of being easy to use without sacrificing power and flexibility.

### 3.2.1 Architecture

Alchemi follows the master-worker parallel programming paradigm [14] in which a central component dispatches independent units of parallel execution to workers and manages them. This smallest unit of parallel execution is a grid thread, which is conceptually and programmatically similar to a thread object (in object-oriented sense) that wraps a "normal" multitasking operating system thread.

### 3.2.2 Alchemi components

Alchemi has the following components designed for the grid construction:

- Manager
- Executor
- Owner
- Cross Platform Manger

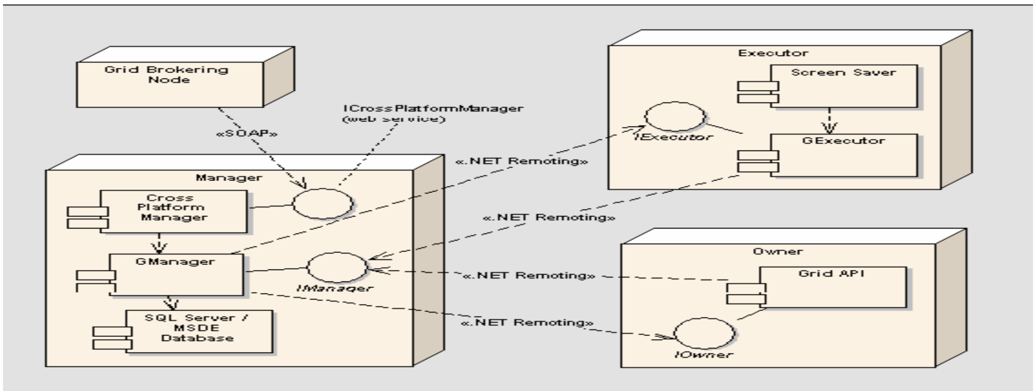


*Fig. 4.4 Alchemi architecture [1]*

#### 3.2.2.1 Manger

The Manager manages the execution of grid applications and provides services associated with managing thread execution. The Executors register themselves with the Manager, which in turn keeps track of their availability. Threads received from the Owner are placed in a pool and scheduled to be executed on the various available Executors. Threads are scheduled on a Priority and First Come First Served (FCFS) basis, in that order.

#### 3.2.2.2 Executor

The Executor accepts threads from the Manager and executes them. An Executor can be configured to be dedicated, meaning the resource is centrally managed by the Manager, or non-dedicated, meaning that the resource is managed on a volunteer basis via a screen saver or by the user. For non-dedicated execution, there is one-way communication between the Executor and the Manager. Thus, Alchemi's execution model provides the dual benefit of:

- Flexible resource management i.e. centralized management with dedicated execution vs. decentralized management with non-dedicated execution; and

- Flexible deployment under network constraints i.e. the component can be deployment as non dedicated where two-way communication is not desired or not possible (e.g. when it is behind a firewall or NAT/proxy server).

Thus, dedicated execution is more suitable where the Manager and Executor are on the same Local Area Network while non-dedicated execution is more appropriate when the Manager and Executor are to be connected over the Internet.

#### 3.2.2.3 Owner

Grid applications created using the Alchemi API are executed on the Owner component. The Owner provides an interface with respect to grid applications between the application developer and the grid. Hence it "owns" the application and provides services associated with the ownership of an application and its constituent threads. The Owner submits threads to the Manager and collects completed threads on behalf of the application developer via the Alchemi API [1].

#### 3.2.2.4 Cross-Platform Manager

The Cross-Platform Manager, an optional sub-component of the Manager, is a generic web services interface that exposes a portion of the functionality of the Manager in order to enable Alchemi to manage the execution of platform independent grid jobs (as opposed to grid applications utilizing the Alchemi grid thread model). Jobs submitted to the Cross-Platform Manager are translated into a form that is accepted by the Manager (i.e. grid threads), which are then scheduled and executed as normal in the fashion described above.

## 3.3 Condor

Condor is a high-throughput distributed batch computing system. Condor is a sophisticate job scheduler developed by the condor research project at the university of Wisconsin-Madison Department of Computer science. Users submit their serial or parallel jobs to Condor, Condor places them into a queue, chooses when and where to run the jobs based upon a policy, carefully monitors their progress, and ultimately informs the user upon completion [6].

### 3.3.1 Condor architecture

Condor's key activities - job-resource allocation, job startup and execution, and metadata collection and display – are kept separate, allowing compartmentalization of Condor into clearly defined components, distributed amongst submission site, central manager and execution site, as illustrated in figure: [5]

*Central Manager*: For every condor pool a single central manager is responsible for collecting resource characteristics and usage information (i.e. accounting)

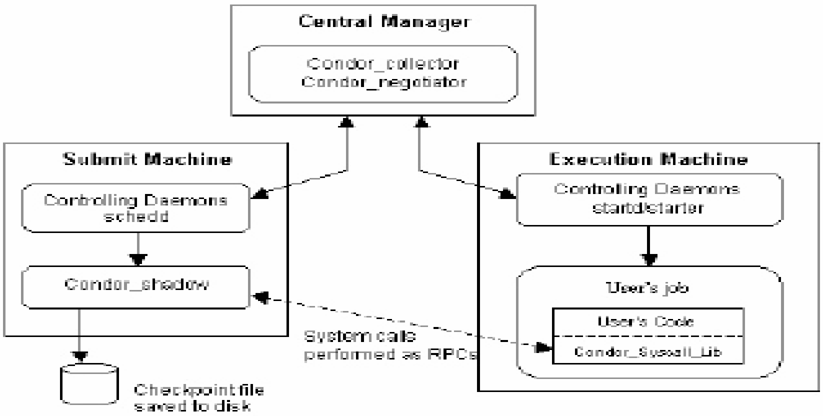


*Fig. 4.8 Condor Architecture [5]*

from all machines in the pool and enforcing *community policies [5]*. It is based on this collected information, and on *user priorities*, that job execution requests can be *matched* to suitable resources for execution during a negotiation cycle.

*Submit Machine:* This system client allows users to submit jobs to a local virtual ‘queue’ (scheduler - *schedd*). The scheduler will request resource allocations for its jobs from the central manager during a negotiation cycle [5]. Once a resource has been allocated to a job, the scheduler will spawn a *shadow* daemon responsible for managing the remote execution that job.

*Execute Machine*: The execute machine, represented by the *startd* daemon, runs jobs on behalf of clients [5]. It advertises its capabilities and usage information - as well as requirements and preferences upon a match - to the central manager, and manages the local execution of the job (via a spawned *starter* daemon), whilst protecting resource owner policies (e.g. a job may be vacated if the user touches the keyboard).

### 3.3.2 Condor daemons

A Condor workstation (machine) [21] runs two Condor daemons, the scheduler daemon *Schedd* and the starter daemon *Startd*. One Condor machine is designated to run Central Manager (CM), which consists of two daemons, the *Negotiator* and the *Collector*. The Condor daemons cooperate with each other by exchanging messages. The tasks of the daemon are:

- The *Schedd* [21][26] maintains a queue of jobs submitted on its machines, prioritizes them and controls the remote startup of these jobs. The requirements of each job are stored in a job context.
- The *Startd* monitors [21][26] the state of its machines, advertises its resources towards the CM, handles the startup and monitors the execution of a job submitted at another machine.
- The *Collector* [21][26] gathers information of the machines in the pool. The information, which is sent by the *Schedd* (job queue information), the *Startd* (machine state and resources), and the *Negotiator* (machine properties), is stored in a machine context.
- The *Negotiator* [15][16] prioritizes the machines, and matches contexts of jobs and contexts of available machines.

### 3.4 Nimrod-G

Nimrod-G is a tool for automated modeling and execution of parameter sweep applications (parameter studies) over global computational Grids. It provides a simple declarative parametric modeling language for expressing parametric experiments. A domain expert can easily create a plan for a parametric experiment and use the Nimrod-G system to deploy jobs on distributed resources for execution [19]. It uses novel resource management and scheduling algorithms based on economic principles. Specifically, it supports user-defined deadline and budget constraints for schedule optimizations and manages supply and demand of resources in the Grid using a set of resource trading services

### 3.4.1 Architecture

Nimrod-G has been developed as a Grid resource broker based on the GRACE framework. It leverages services provided by Grid middleware systems such as Globus, Legion, and the GRACE trading mechanisms. The middleware systems provide a set of low-level protocols for secure and uniform access to remote resources; and services for accessing resources information and storage management. The modular and layered architecture of Nimrod-G is shown in Figure 4.2.

The key components of Nimrod-G resource broker consist of:

- Nimrod-G Clients, which can be:
    - Tools for creating parameter sweep applications.
    - Steering and control monitors, and

  - Customized end user applications (e.g., Active Sheets).
- The Nimrod-G Resource Broker, that consists of:
  - A Task Farming Engine (TFE),
  - A Scheduler that performs resource discovery, trading, and scheduling,
  - A Dispatcher and Actuator, and
  - Agents for managing the execution of jobs on resources.

The Nimrod-G broker architecture leverages services provided by lower-level different Grid middleware solutions to perform resource discovery, trading, and deployment of jobs on Grid resources [19].

### 3.5 Grid Middleware Characteristics

The various characteristics, which the grid middleware's should have, are:

- Transparency: The grid middleware should provide the users an environment where they are unaware of the underlying complexities like where the resources are located, who is the owner of that particular resource.
- Robustness: As the very nature of the grid, they need to operate in very dynamic environment. The virtual organizations are constructed dynamically at run time depending upon the requirement of a particular application. The chances of the failure of the grid nodes increase; the middleware should be able to deal with the faults. It should be able to dynamically relocate the load if a node becomes unavailable.
- Security: The grid spans over a large geographical area and multiple organizations, which dynamically collaborate at run time. Every organization has its own security policies, authentication mechanisms and security requirements. The middleware must take care of these security policies so that the integrity and the confidentiality of these organizations are maintained.
- Persistency: The middleware should be able to mange the states and keep track of them, so that the user can retrieve the desired information, like how much the job is completed, how long will it still take to finish the job etc, at any time.
- Scalability: the middleware should scale the grids in a very nice manner as the grids are the talk of the day, and they are expanding.
- Ease to use/ program: The middleware should provide the users an environment where they can write code very easily so that it can be run on grid environment. The programming environment should be easy to learn.

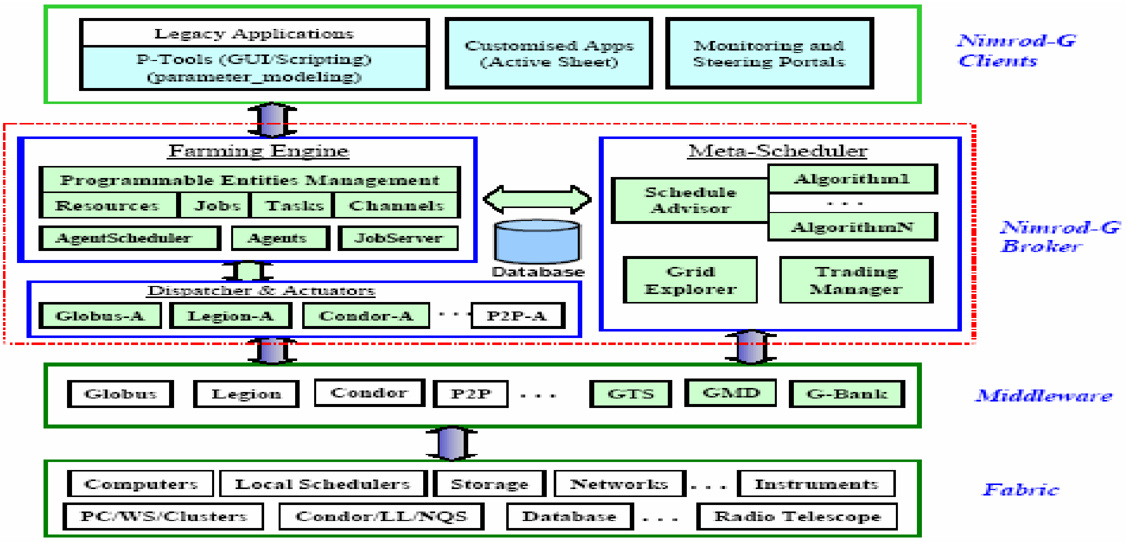


*Fig4.9 Nimrod/G Layered Architecture [19]*

## 4. Comparative analysis of Middlewares

As discussed in the problem statement, the middlewares are compared on the basis of following characteristics:

### 4.1 Category

The category defines whether the middleware is user level or core middleware. The globus, alchemi, condor are core middlewares where as the nimrod/G is a user level middleware. Nimrod/g requires a core middleware for it’s functioning. Here we use it with the globus.

### 4.2 Security

Security is key to the grid environment. The globus has GSI pyramid for the security. It uses Public Key infrastructure and X.509 certificates for the authentication purposes. The Alchemi has role-based security. The manager is configured to support anonymous or non-anonymous Executors. The Alchemi administrator configures user, group and permission. In Nimrod/G there is as such no security mechanisms. It uses the security provided by the low-level middleware services. Condor provides high support for authorization, authentication, encryption. When the condor is installed the default configuration setting include none of them. The administrator, through the use of macros, enables these features.

### 4.3 Architecture

The globus follows the hourglass model, in which the core functionalities are in the center. It has a layered and modular architecture. Each module and layer focuses on a particular aspect. The alchemi has hierarchal, master slave architecture. It has a centralized manager and the executors, which connect to this manager. Condor too has a hierarchal architecture. The nimrod/g has a component based layered architecture. These components interact with each other to deliver the functionality.

### 4.4 Scalability

Scalability indicates the ability of a system to increase total throughput under an increased load and when hardware or software resources are added. The globus is very close to the hardware. The scalability is direct result. It can scale till Internet. Alchemi has a centralized manger so the load on the manger increases if the grid is expanded to a large extent. Similarly in the condor the centralized architecture limits the scalability. Nimrod/G is extensible enough to use the underlying middleware services.

### 4.5 Programming environment

Globus provides the replacement libraries for UNIX & C libraries, Special MPI library (MPICH –G), CoG (Commodity Grid) kits in Java, Python, CORBA, Matlab, Java Server Pages, Perl and Web Services. The Alchemi uses the Grid multithreaded model for the execution of the applications. The condor has support for the C, Java, and MPI environment. The Nimrod provides a parametric language to describe the executable written in C or java.

### 4.6 Run Time Platform

The run time platform for the globus is the unix like platforms and the windows. The Alchemi run on the windows environment having .net framework installed. The Condor has support for both widows and unix platforms. The run time platform plays an important role. As if a user familiar with linux environment cannot use the alchemi.

### 4.7 Ease of use/Understand

The globus is very close to hardware level. It uses the complex calls and the commands to run the jobs and it operates on the command line interface. So it is not user friendly. The alchemi has a graphical interface and offers the simple API to use the environment. The condor too works only on command line interface and user need to learn commands to operate.

### 4.8 Scheduling policy

Globus does not have its own scheduler it uses the above layer functionalities for the scheduling. The Condor Scheduling policy is Performance centric i.e. it tries to maximize the throughput of overall system. The nimrod/g has market driven scheduling policy i.e. it uses the economical principles. Alchemi has system centric Scheduling policy

### 4.9 Type of applications

The condor is used for high throughput computing applications. The nimrod/g is used for scientific parametric sweep applications. The Globus and Alchemi are used for the applications requiring large computations.

### 4.10 Implementation Technologies

These all are open source projects. The Globus has been implemented using C and java technologies. The Alchemi has been implemented using the C#, perl. The Condor has been implemented using C and java. The above comparison is summarized in the table given next.

## 5. Conclusion

This paper provides the introduction to the grid computing. It emphasizes the importance of the middleware components, discusses the major grid middleware technologies At last, the comparative analysis between these middleware has been done on the basis of various factors. This paper provides, the in depth detail of the Grid Middleware that act as an interface between the resources and the applications. This paper provides the architectural and philosophical differences, between these middleware technologies and thus educates the grid community about the choices available.

## References

[1] Akshay Luther, Rajkumar Buyya, Rajiv Ranjan, and Srikumar Venugopal, "*Alchemi: NET-based Grid Computing Framework and its Integration into Global Grids*" Technical Report, GRIDS-TR-2003-8, Grid Computing and Distributed Systems Laboratory, University of Melbourne, Australia, December 2003.

[2] B. Bansal, "*Design and Development of Grid Portal*", thesis Submitted at Computer Science & Engineering Deptt., TIET Patiala, May 2005.

[3] Bart Jacob, "*Grid computing: What are the key components?*" .

[4] Clovis Chapman1, Paul Wilson2, "*Condor services for the Global Grid: Interoperability between Condor and OGSA*", Proceedings of the 2004 UK e-Science All Hands Meeting, ISBN 1-904425-21-6, pages 870-877, Nottingham, UK, August 2004

[5] Douglas Thain, Todd Tannenbaum, and Miron Livny, "*Distributed Computing in Practice: The Condor Experience*" *Concurrency and Computation: Practice and Experience*, Vol. 17, No. 2-4, pages 323-356, April, 2005.

[6] Foster, I., Kesselman, C., Nick, J. and Tuecke, S..”The Anatomy of the Grid: Enabling Scalable Virtual Organizations” *International J. Supercomputer Applications*, 15(3), 2001.

| Middleware<br>Property | Globus | Alchemi | Condor | Nimrod/G |
|---|---|---|---|---|
| **Category** | Core Level | Core Level | Core Level | User Level |
| **Security** | PKI, X.509<br>No Authorization | Role Based Security | Both Authentication and Authorization | Underlying Core middleware |
| **Architecture** | Layered & Modular | Hierarchal | Hierarchal | Component based |
| **Scalability** | High | Limited | Limited | Extensible to underlying Middleware |
| **Programming Environment** | C, Java, Cog, Matlab, Perl, Corba, MPICH-G Python, | C#, OOP, threaded approach | C, MPI, JAVA | Parametric Language |
| **Platform** | Windows, Unix Like | Windows | Unix, Windows | Unix |
| **Ease of Use** | Low | High | Low | High |
| **Scheduling Policy** | Uses user level middleware | System Centric | System centric | Market driven |
| **Application Type** | Computational | HTC | Computational | Parametric |
| **Implementation technology** | C, java | C#, Perl | C, Java | - |

Table 1: Comparison of Related Middleware

[7] Foster, I., Kesselman, C., Nick, J. and Tuecke, S**.** **"**The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration**"**, Open grid service Infrastructure WG Global Grid Forum

[8] Foster, I., **"**What is the Grid? A Three Point Checklist," GRIDToday, July 20, 2002

[9] Francois Grey, Matti Heikkurinen, Rosy Mondardini, Robindra Prabhu, "Brief History of Grid"

[10] Jay Haunger, Matt Haynos, "A visual tour of Open Grid Services Architecture"

[11] Jean-Christophe Durand, "*Grid Computing: A Conceptual And Practical Study*" , University de Lausanne, Switzerland, November 2004

[12] Jospeh, Joshy, Fellenstine Craig, "*Grid Computing*", Prentice Hall/IBM Press, Edition 2004, India.

[13] Livny and R. Raman. The GRID: Blueprint for a New Computing Infrastructure, chapter High Throughput Resource Management, pages 311–336. Morgan Kaufman, 1999.

[14] Livny, M., Raman, R. and Solomon, M. *Matchmaking: Distributed Resource Management for High Throughput Computing*. Proc. of the Seventh IEEE International Symposium on High Performance Distributed Computing, Chicago, IL., July 28th-31st, 1998.

[15] Mark Baker, Rajkumar Buyya, and Domenico Laforenza , " Grids and Grid technology for wide-area distributed computing", Software Practice and experience, 2002

[16] Rajkumar Buyya, "The Nimrod-G Resource Broker: an economic Based Grid Scheduler"

[17] R.J.M.Boer, "Resource Management in the Condor System", Delft University of Technology, Netherlands and National Institute for Nuclear Physics and High Energy Physics, Netherlands, May-1996

[18] Thierry Prioi, "Grid Middleware", Advanced Grid Reseach Workshops through Europeon and Asian Co-operation,

[19] Viktors Berstis, et al, "*Introduction to Grid Computing with Globus* ", http://www.redbooks.ibm.com/ redbooks/pdfs/sg246895.pdf

[20] "Condor R Version 6.6.10 Manual", Condor Team, University of Wisconsin– Madison, Jan 31, 2006

[21] www.globus.org

[22] www.wiki.gridpp.ac.uk/wiki/Grid_middleware

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# An Optimization Problem in Deregulated Electricity Markets solved with the Nonsmooth Maximum Principle

**L. Bayón[1], J.M. Grau[1], M.M. Ruiz[1] and P.M. Suárez[1]**

[1] *Department of Mathematics, University of Oviedo, Spain*

emails: bayon@uniovi.es, grau@uniovi.es, mruiz@uniovi.es, pedrosr@uniovi.es

**Abstract**

In this paper, the new short-term problems that are faced by a generation company in a deregulated electricity market are addressed and a optimization algorithm is proposed. Our model of the spot market explicitly represents the price of electricity as an uncertain exogenous variable. We consider a very complex problem of hydrothermal optimization with pumped-storage plants, so the problem deals with non-regular Lagrangian and non-holonomic inequality constraints. To obtain a necessary minimum condition, the problem was formulated within the framework of nonsmooth analysis using the generalized (or Clarke's) gradient and the Nonsmooth Maximum Principle. The optimal control problem is solved by means of an algorithm implemented in the commercial software package Mathematica. Results of the application of the method to a numerical example are presented.

*Key words: Nonsmooth Analysis, Control Problem, Electricity markets*
*MSC 2000: 49M20, 49J24, 91B24, 91B26*

## 1 Introduction

Over the last decade, the electricity industry has experienced significant changes in terms of deregulation and competition. In this paper, we focus on the problem that a generation company faces when preparing its offers for the *day-ahead* market. Several methods have been proposed for simulating competitive generation markets. Most of these models [1] can be categorized into two major groups: models that represent all the generation companies and models that focus on a particular generation company. Two approaches can be adopted to represent spot market auctions when only one company is considered: price modeled as an *exogenous* variable and price modeled as a *function of the demand supplied* by the firm under study. In the former, the price of electricity does not depend on the company's decisions. This can be acceptable if the company is small enough. These models can again be classified into two sub-groups, depending on whether they use a deterministic [2] or probabilistic [3] price representation.

In this paper, we only represent the operation of *one company* in detail, including each of the company's generation units. Our model of the spot market explicitly represents the price of electricity as an *uncertain exogenous* variable. We represent generation units at a high level of detail and our model distinguishes *individual generation units* and considers *inter-temporal constraints* such as hydro reserves. In addition, we also consider *pumped-storage* hydro-plants.

The Spanish activity rules [4] have been used as a reference model for the market. The day-ahead market in the Spanish wholesale electricity market is organized as a set of twenty-four simultaneous *hourly auctions*. The *simple* bid format consists of a pair of (hourly) values: quantity $q[MWh]$ and price $p[euro/MWh]$. The utility company that inspires our paper, *HC*, controls approximately only 7% of all the electricity that is generated. So, we consider our company as a price-taker, and under this assumption, the volatility of the spot market price of electricity is represented by a *stochastic model.* Price forecasting techniques in power systems are relatively recent procedures [5] [6]. Although the problem of constructing the probability distribution exceeds the purpose of this paper, we suggest the following simplified approach based on [7]. The idea is to search for past spot market sessions that can be considered similar to the session that the company is about to face. To identify the days, we classify the entire collection of sessions (using *clustering techniques*) according to the values of an *explanatory variable.* The most relevant information about the current session for our problem is the vector of 24 prices that has resulted from the day-ahead market clearing. Once a group of $S$ similar days has been identified, the company can assume that the probability distribution for the market session under study is completely defined by these past $S$ market sessions (*probability distributions with finite support*). If we now focus on a particular auction, it is easy to understand that the $S$ quantities and $S$ prices decided by the company for that hour constitute the *offer curve* (nondecreasing) that the company must submit to that auction.

This paper addresses a very complex problem of hydrothermal optimization with pumped-storage plants. In this kind of problem (see the previous paper [8]), the Lagrangian is piecewise continuous and we consider constraints for the admissible generated power. Hence, this paper considers non-regular Lagrangian and non-holonomic inequality constraints (differential inclusions). To obtain a necessary minimum condition, the problem is formulated within the framework of nonsmooth analysis [9] using the generalized (or Clarke's) gradient and the Nonsmooth Maximum Principle. This characteristic distinguishes our work from all the above.

## 2 Statement of the Problem

In this section the optimization problem of one company is described, the objective function of which can be defined as its *profit maximization.* Let us assume that our hydrothermal system accounts for $n$ hydro-plants and $m$ thermal plants: the $(H_{\mathbf{n}} - T_{\mathbf{m}})$ *problem.*

Let $\Psi_i : D_i \subseteq \mathbb{R}^+ \longrightarrow \mathbb{R}^+$ $(i = 1, \ldots, m)$ be the cost functions $(euro/h)$ of the $m$

thermal plants. The most usual cost function of each generator can be represented as a quadratic function:

$$\Psi_i(P_i(t)) = \alpha_i + \beta_i P_i(t) + \gamma_i P_i^2(t); \quad i = 1, ..., m$$

where $P_i(MW)$ is the power generated, and we consider the thermal plants to be constrained by *technical restrictions* of the type

$$P_{i\,\min} \leq P_i(t) \leq P_{i\,\max}; \quad i = 1, ..., m, \ \forall t \in [0, T]$$

$[0, T]$ being the optimization interval. In prior studies [10], it was proven that the problem with $m$ thermal plants may be reduced to the study of a hydrothermal system made up of one single thermal plant, called the *thermal equivalent*: the $(H_{\mathbf{n}} - T_{\mathbf{1}})$ *problem.* We shall denote as the *equivalent minimizer* of $\{\Psi_i\}_1^m$, the function $\Psi : D_1 + \cdots + D_m \to \mathbb{R}$ defined by

$$\Psi(P(t)) = \min \sum_{i=1}^{m} \Psi_i(P_i(t)); \ P_{\min} \leq P(t) \leq P_{\max}$$

with $P(t)$ the power generated by said thermal equivalent.

We assume that our system accounts for $n$ hydro-plants that have a pumping capacity. The mapping $H : \Omega_H \longrightarrow \mathbb{R}$

$$H(t, z_1(t), \ldots, z_i(t), \ldots, z_n(t), \dot{z}_1(t), \ldots, \dot{z}_i(t), \ldots, \dot{z}_n(t)) = H(t, \mathbf{z}(t), \dot{\mathbf{z}}(t))$$

is called the function of *effective hydraulic contribution* and is the power contributed to the system at the instant $t$ by the set of hydro-plants, $z_i(t)$ being the volume that is discharged up to the instant $t$ by the $i$-th hydro-plant, $\dot{z}_i(t)$ the rate of water discharge at the instant $t$ by the $i$-th hydro-plant, and $\Omega_H \subset [0, T] \times \mathbb{R}^n \times \mathbb{R}^n$ the domain of definition of $H$.

We say that $\dot{\mathbf{z}} = (z_1, \ldots, z_n)$ is admissible for $H$ if $z_i$ belong to the class $\widehat{C}^1[0, T]$ (the set of piecewise $C^1$ functions), and $(t, \mathbf{z}(t), \dot{\mathbf{z}}(t)) \in \Omega_H$, $\forall t \in [0, T]$. The volume $b_i$ that must be discharged up to the instant $T$ is called the admissible volume of the $i$-th hydro-plant. Let $\mathbf{b} = (b_1, \ldots, b_n) \in \mathbb{R}^n$ be the vector of admissible volumes. In a general model, with hydraulic coupling between the $n$ hydro-plants, we call $H_i(t, z_i(t), \dot{z}_i(t)) : \Omega_{H_i} = [0, T] \times \mathbb{R} \times \mathbb{R} \longrightarrow \mathbb{R}$ the function of effective hydraulic contribution by the $i$-th hydro-plant, being

$$H(t, \mathbf{z}(t), \dot{\mathbf{z}}(t)) = \sum_{i=1}^{n} H_i(t, z_i(t), \dot{z}_i(t))$$

Besides, we consider $H_i(t, z_i(t), \dot{z}_i(t))$ to be bounded by technical constraints

$$H_{i\,\min} \leq H_i(t, z_i(t), \dot{z}_i(t)) \leq H_{i\,\max}; \ i = 1, \cdots, n, \ \forall t \in [0, T]$$

Throughout the paper, no *transmission losses* will be considered; a crucial aspect when addressing the optimization problem from a centralized viewpoint. From the perspective of a generation company, and within the framework of the new electricity market, said losses are not relevant, since the generators currently receive payment for all the energy they generate in power plant bars.

Let us assume that the cost function $\Psi : \mathbb{R}^+ \longrightarrow \mathbb{R}^+$ satisfies $\Psi'(x) > 0$, $\forall x \in \mathbb{R}^+$, i.e. it is strictly increasing. This constraint is absolutely natural: it reads more cost to more generated power. Let us assume as well that $\Psi''(x) > 0$, $\forall x \in \mathbb{R}^+$, i.e. it is strictly convex. The models traditionally employed meet this constraint.

Let us assume that the function $H_i$ is strictly increasing with respect to the rate of water discharge $\dot{z}_i$, i.e. $\partial H_i/\partial \dot{z}_i > 0$ (more power to a higher rate of water discharge) and that $[\partial H_i/\partial z_i]_{\dot{z}_i=0} = 0$. Let us also assume that $\partial^2 H_i/\partial \dot{z}_i^2 < 0$, i.e. $H_i$ is concave with respect to $\dot{z}_i$. The real models meet these three constraints. In addition, pumped-storage plants are considered, and in this kind of problem, the derivative of $H_i$ with respect to $\dot{z}_i$ $(\partial H_i/\partial \dot{z}_i)$ presents discontinuity at $\dot{z}_i = 0$, which is the border between the power generation zone (positive values of $\dot{z}_i$) and the pumping zone (negative values of $\dot{z}_i$). In the real models, it is verified that $H^+_{\dot{z}} \leq H^-_{\dot{z}}$. In the $(H_\mathbf{n} - T_\mathbf{1})$ problem, the *objective function* is given by revenue minus cost during the optimization interval $[0, T]$

$$F(P, \mathbf{z}) = \int_0^T [p(t)(P(t) + H(t, \mathbf{z}(t), \dot{\mathbf{z}}(t))) - \Psi(P(t))]\, dt$$

Revenue is obtained by multiplying the total production (thermal and hydraulic) of the company by the clearing price $p(t)$ in each hour $t$. Cost is given by $\Psi$, the cost function of the thermal equivalent, where $P(t)$ is the power generated by said plant. With this statement, our objective functional in *continuous time form* is

$$\max_{P,\mathbf{z}} F(P, \mathbf{z}) = \max_{P,\mathbf{z}} \int_0^T L(t, P(t), \mathbf{z}(t), \dot{\mathbf{z}}(t)) dt \tag{2.1}$$

with $L(t, P(t), \mathbf{z}(t), \dot{\mathbf{z}}(t)) = p(t)(P(t) + H(t, \mathbf{z}(t), \dot{\mathbf{z}}(t))) - \Psi(P(t))$, on the set

$$\Omega = \left\{ \mathbf{z} \in \left(\widehat{C}^1[0, T]\right)^n \mid \begin{array}{c} z_i(0) = 0,\ z_i(T) = b_i \\ H_{i\,\min} \leq H_i(t, z_i(t), \dot{z}_i(t)) \leq H_{i\,\max},\ \forall t \in [0, T] \\ \forall i = 1, \ldots, n \end{array} \right\} \tag{2.2}$$

# 3 The $(H_\mathbf{1} - T_\mathbf{1})$ Problem

We begin the development in this section by presenting the simple problem with one pumped-storage hydro-plant $(i = 1)$. In the $(H_\mathbf{1} - T_\mathbf{1})$ problem, we have $\mathbf{z} = z$ and our objective functional is

$$F(P, z) = \int_0^T L(t, P(t), z(t), \dot{z}(t)) dt$$

with $L(t, P(t), z(t), \dot{z}(t)) = p(t)(P(t) + H(t, z(t), \dot{z}(t))) - \Psi(P(t))$ on the set

$$\Omega = \left\{ z \in \widehat{C}^1[0, T] \mid \begin{array}{c} z(0) = 0, z(T) = b \\ H_{\min} \leq H(t, z(t), \dot{z}(t)) \leq H_{\max},\ \forall t \in [0, T] \end{array} \right\}$$

where $L(\cdot,\cdot,\cdot,\cdot)$ and $L_z(\cdot,\cdot,\cdot,\cdot)$ are the class $C^0$ and $L_{\dot z}(t,P,z,\cdot)$ is piecewise continuous ($L_{\dot z}(t,P,z,\cdot)$ is discontinuous in $\dot z = 0$). The problem involves non-holonomic inequality constraints (differential inclusions) and the previous assumptions guarantee that: $L_{\dot z\dot z}(t,P,z,\dot z) < 0$; $L_{\dot z}(t,P,z,\dot z) > 0$. We also assume that

$$H(t,b,\dot z(t)) \le H(t,z(t),\dot z(t)) \le H(t,0,\dot z(t)),\ \forall z \in \Omega$$

These suppositions are fulfilled in all real hydrothermal problems, and bearing in mind the weak influence of $z(t)$, ($H(t,b,\dot z) \simeq H(t,z,\dot z) \simeq H(t,0,\dot z)$), it is reasonable to substitute the restriction: $H_{\min} \le H(t,z(t),\dot z(t)) \le H_{\max}$ by others of the type: $H_{\min} \le H(t,b,\dot z)$; $H(t,0,\dot z) \le H_{\max}$. Thus, it is reasonable to substitute $\Omega$ by

$$\Omega^* = \left\{ z \in \widehat{C}^1[0,T] \mid \begin{array}{c} z(0)=0, z(T)=b \\ H_{\min} \le H(t,b,\dot z);\ H(t,0,\dot z) \le H_{\max},\ \forall t \in [0,T] \end{array} \right\}$$

The solution to the problem in $\Omega^*$ will be very close to that obtained with the set $\Omega$, the advantage being that the mathematical treatment of sets of type $\Omega^*$ is much simpler than of those of type $\Omega$. We shall focus in the present paper on the development of the applications of Optimal Control Theory (OCT) and nonsmooth analysis to this problem. Let us term as the *coordination function* of $z \in \Omega^*$ the function in $[0,T]$, defined by:

$$\mathbb{Y}_z(t) = L_{\dot z}(t,P(t),z(t),\dot z(t)) - \int_0^t L_z(s,P(s),z(s),\dot z(s))ds$$

denoting by $\mathbb{Y}_z^+(t)$ and $\mathbb{Y}_z^-(t)$ the expressions obtained when considering the lateral derivatives of $L$ with respect to $\dot z$. Let us now see the fundamental result, which is the basis for elaborating the optimization algorithm. We present the problem considering the *state variables* to be $z(t)$ and $P(t)$ and the *control variables* $u_1(t) = \dot z(t)$ and $u_2(t) = \dot P(t)$. The *optimal control problem* is thus:

$$\max_{u_1(t),u_2(t)} \int_0^T L(t,P(t),z(t),u_1(t))dt; \quad \text{with} \left\{ \begin{array}{l} \dot z = u_1; \quad \dot P = u_2 \\ z(0)=0, \quad z(T)=b \end{array} \right.$$
$$u_1(t) \in \Theta = \{x \mid H_{\min} \le H(t,b,x); H(t,0,x) \le H_{\max}\}\,;\ u_2(t) \in (-\infty,\infty)$$

We shall use the nonsmooth version of Pontryagin's Minimum Principle (PMP) [9] as the basis for proving this theorem.

**Theorem 1 (Theorem of Coordination).** *If $(z^*,P^*) \in (\widehat{C}^1, C^1)$ is a solution of our problem, then $\exists K \in \mathbb{R}^+$ such that:*

*i) If* $\dot z^*(t) = 0 \implies \mathbb{Y}_{z^*}^+(t) \le K \le \mathbb{Y}_{z^*}^-(t)$

*ii) If* $\dot z^*(t) \ne 0 \implies \mathbb{Y}_{z^*}(t)$ *is* $\left\{ \begin{array}{lll} \ge K & \textit{if} & H(t,b,\dot z^*(t)) = H_{\min} \\ = K & \textit{if} & H_{\min} < H(t,z^*(t),\dot z^*(t)) < H_{\max} \\ \le K & \textit{if} & H(t,0,\dot z^*(t)) = H_{\max} \end{array} \right.$

*and* $\dot\Psi\,(P^*(t)) = p(t)$

We shall call this relation

$$L_{\dot z}(t,P(t),z(t),\dot z(t)) - \int_0^t L_z(s,P(s),z(s),\dot z(s))ds = K \in \mathbb{R}^+, \forall t \in [0,T] \tag{3.1}$$

the *coordination equation* for $z(t)$, and the positive constant $K$ will be termed the *coordination constant* of the extremal.

**Note**. It is very important to stress that the problem is thus easily broken down into the two sub-problems: Thermal and Hydro. In the thermal sub-problem, the power $P(t)$ of the equivalent thermal plant is distributed (as we see in [10]) between the $m$ thermal plants, and so is completely resolved. In the next section, we consider once more the general problem $(H_{\mathbf{n}} - T_{\mathbf{1}})$ with $n$ hydro-plants, which is the problem to be solved.

## 4 Generalization to the $(H_{\mathbf{n}} - T_{\mathbf{1}})$ Problem. The Optimization Algorithm

In this section, we present an algorithm of the numerical resolution of the problem of optimization of a hydrothermal system that involves $n$ hydro-plants. The associated variational problem is related to solving a boundary-value problem for a system of differential equations. The algorithm uses a particular strategy related to the method of *cyclic coordinate descent* (CCD). The CCD method minimizes a function cyclically with respect to the coordinate variables. With our method, a problem of the type $H_{\mathbf{n}} - T_{\mathbf{1}}$ could be solved (under certain conditions) if we start out from the resolution of a sequence of problems of the type $H_{\mathbf{1}} - T_{\mathbf{1}}$. The algorithm for the $H_{\mathbf{n}} - T_{\mathbf{1}}$ problem carries out several iterations and at each *j-th* iteration calculates $n$ stages, one for each hydro-plant. At each stage, it calculates the optimal functioning of a hydro-plant, while the behavior of the rest of the plants is assumed fixed. For every $\mathbf{z} = (z_1, \ldots, z_n) \in \Omega$, we consider the functional $F^i_{\mathbf{z}}$ defined by

$$F^i_{\mathbf{z}}(P, v_i) = \int_0^T \left[p(t)(P(t) + H^i_{\mathbf{z}}(t, v_i(t), \dot{v}_i(t))) - \Psi\left(P(t)\right)\right] dt$$

$$\text{with } H^i_{\mathbf{z}}(t, v_i, \dot{v}_i) = H(t, z_1, \ldots, z_{i-1}, v_i, z_{i+1}, \ldots, z_n, \dot{z}_1, \ldots, \dot{z}_{i-1}, \dot{v}_i, \dot{z}_{i+1}, \ldots, \dot{z}_n)$$

where $H^i_{\mathbf{z}}$ represents the power generated by the hydraulic system as a function of the rate of water discharge and the volume turbinated by the $i$-th plant, under the assumption that the rest of the plants behave in a definite way. We call the $i$-th *minimizing mapping* the mapping $\phi_i : \Omega \longrightarrow \Omega$, defined in the following way: for every $\mathbf{z} \in \Omega$

$$\phi_i(P, z_1, \ldots, z_i, \ldots, z_n) = (P^*, z_1, \ldots, z^*_i, \ldots, z_n)$$

where $(P^*, z^*_i)$ minimizes $F^i_{\mathbf{z}}$. If we set $\Phi = (\phi_n \circ \phi_{n-1} \circ \cdots \circ \phi_2 \circ \phi_1)$ and

$$(P^j, \mathbf{z}^j) = \Phi(P^{j-1}, \mathbf{z}^{j-1})$$

beginning with some admissible $(P^0, \mathbf{z}^0)$, we construct a sequence of $(P^j, \mathbf{z}^j)$ via successive applications of $\{\phi_i\}_{i=1}^n$ and the algorithm will search

$$\lim_{j \to \infty} (P^j, \mathbf{z}^j)$$

It is simple to justify the convergence of the algorithm in a finite number of steps, simply by considering the following solution set:

$$\{\mathbf{z} \mid F(P, \mathbf{z}) - F(\Phi\,(P, \mathbf{z}))< \varepsilon\}$$

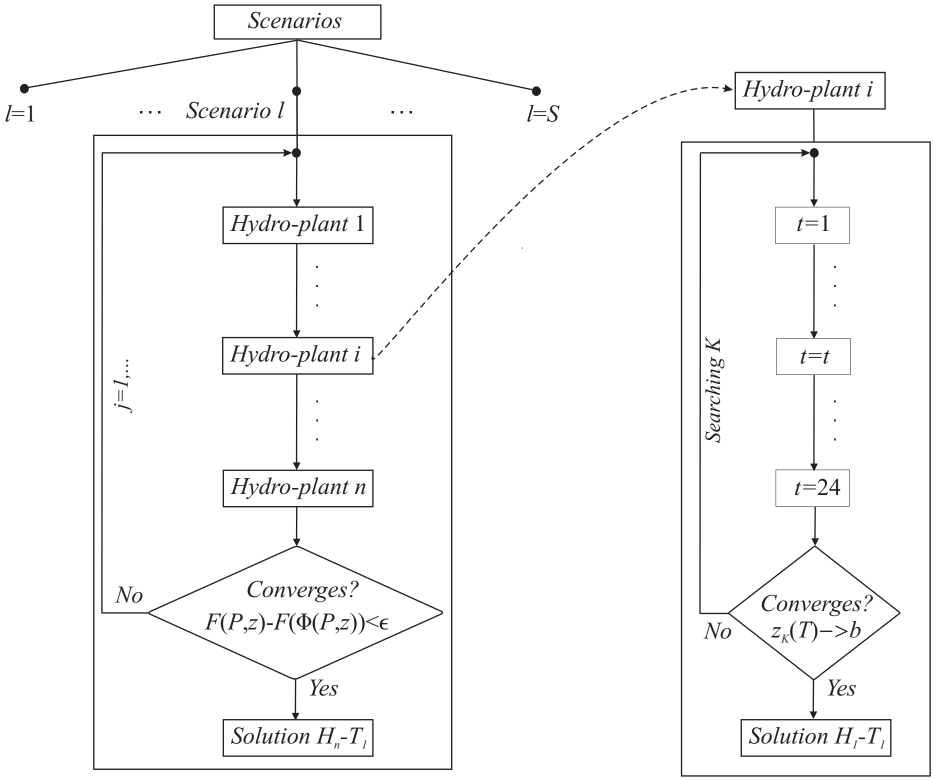


Fig. 1. The Optimization Algorithm.

The application of every $\phi_i$ involves solving a problem of the type $(H_{\mathbf{1}} - T_{\mathbf{1}})$. To obtain the optimum operating conditions of the hydro-plant, we shall use the coordination equation (3.1). To undertake the approximate calculation of the solution, expressed in Theorem 1, we use a similar numerical method to those used to solve differential equations in combination with an appropriate adaptation of the classical shooting method.

*Step* 1) Approximate construction of $z_K$ (*the adapted Euler method*).

The problem will consist in finding for each $K$ the function $z_K$ that satisfies $z_K(0) = 0$, and the conditions of Theorem 1. From the computational point of view, the construction of $z_K$ can be performed with the use of a discretized version of Equation (3.1). The approximate construction of each $z_K$, which we shall call $\widetilde{z}_K$, is carried out by means of polygonals (Euler's method). In general, the construction of $\dot{z}_K$ must be carried out by constructing and successively concatenating the extremal arcs and boundary arcs until completing the interval $[0, T]$.

*Step* 2) Construction of a sequence $\{K_j\}_{j\in\mathbb{N}}$ such that $z_{K_j}(T)$ converges to $b$ (*the adapted shooting method*).

Varying the coordination constant $K$, we would search for the extremal that fulfils the second boundary condition $z_K(T) = b$. The procedure is similar to the shooting method used to resolve a two-point boundary value problem (TPBVP). A number of methods exist for solving these problems, including shooting, collocation and finite difference methods. Among the shooting methods, the Simple Shooting Method (SSM) and the Multiple Shooting Method (MSM) appear to be the most widely known and used methods. We implemented a SSM and obtained good results. Effectively, we may consider the function $\varphi(K) := z_K(T)$ and calculate the root of $\varphi(K) - b = 0$, which may be realized approximately using elemental procedures. The secant method was used in the present paper, and the algorithm shows a rapid convergence to the optimal solution for a wide range of $K_{\min}$ and $K_{\max}$.

# 5 Application to a Real Hydrothermal System

A computer program was written (using the Mathematica package) to apply the results obtained in this paper to a real power system. As an example, we shall use the hydrothermal system that the electricity company *HC* has in Asturias (Spain), which is made up of 2 classic thermal plants: *Aboño* (with two groups of 360 and 543 of power ($Mw$) respectively) and *Soto* (with two groups of 254 and 350 of power ($Mw$) respectively) and 9 hydro-plants. For our optimization problem, we shall only use the 3 *variable-head* (the generation is function of $z$ and $\dot{z}$) hydro-plants of the utility company *HC*: *Salime*, *Tanes* (pumped-storage) and *La Barca*. We do not consider the remaining hydro-plants, because they are *run-of-river* type (without reservoir) and power generation is not controllable. Let us see the models of different subsystems used in our study. For the cost functions, we use a second-order polynomial

$$\Psi_i(P_i(t)) = \alpha_i + \beta_i P_i(t) + \gamma_i P_i^2(t)$$

The hydro-network has the three hydro-plants on different rivers, so the rate of discharge at the upstream plant does not affect the behaviour at the downstream plants: the hydraulic system has no *hydraulic coupling*. We use a *variable head* model and the $i$-th function of effective hydraulic generation $H_i$ (for a conventional hydro-plant) is given by

$$H_i(t, z_i(t), \dot{z}_i(t)) = A_i(t)\dot{z}_i(t) - B_i\dot{z}_i(t)z_i(t) - C_i\dot{z}_i^2(t); \quad \dot{z}_i(t) \geq 0$$

where $A_i(t)$, $B_i$ and $C_i$ are the coefficients

$$A_i(t) = \frac{1}{G_i} B_{y_i}(S_{0i} + t \cdot i_i); \quad B_i = \frac{B_{y_i}}{G_i}; \quad C_i = \frac{B_{t_i}}{G_i}$$

For the pumped-storage plant, $H_i$ is defined piecewise, taking in the pumping zone ($\dot{z}_i(t) < 0$): $M \cdot H_i(t, z_i(t), \dot{z}_i(t))$. The parameters that appear in this formula are: the efficiency $G$ in $(m^4/h.Mw)$, the natural inflow $i$ in $(m^3/h)$, the initial volume $S_0$ in $(10^6 m^3)$, and the coefficients $B_y$ in $(10^{-7} m^{-2})$ and $B_t$ in $(10^{-5} hm^{-2})$, parameters that depend on the geometry of the reservoir.

Let us consider the construction of the scenario structure for the day-ahead market problem faced by the *company HC* in the Spanish spot market. In particular, the market session of February 15th 2006 is considered as the current session. The past market sessions [4] that are considered relevant range from February 1st to February 14th. Table I presents the results of the clustering analysis performed on this range of days. The classification provided by the $S$-means algorithm for $S = 4$ (four clusters) is presented below.

Table I. Clustering Analysis.

| Date | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Day | W | Th | F | Sa | S | M | T | W | Th | F | Sa | S | M | T | W |
| Cluster | 4 | 4 | 4 | 1 | 2 | 3 | 4 | 4 | 4 | 4 | 1 | 2 | 3 | 4 | 4 |

As can be seen, the four day types provided by the clustering analysis are quite reasonable: Cluster 1 and Cluster 2 corresponds to low-price days (Saturdays and Sundays, respectively), Cluster 3 includes one type of weekday: Mondays, and Cluster 4 comprises the other type of weekdays. This analysis suggests considering eight scenarios (eight realizations) for the day-ahead market problem faced by the company on February 15th. We consider short-term hydrothermal scheduling (24 hours) with an optimization interval $[0, 24]$ and we consider a discretization of 24 subintervals. The total optimal hydro and thermal power generation for the company HC are shown in Figure 2-a and Figure 2-b respectively. The eight scenarios considered are presented in both figures.

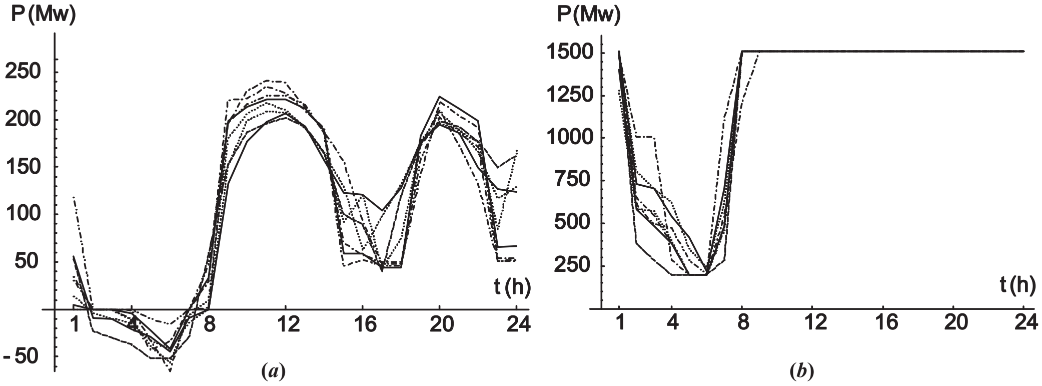
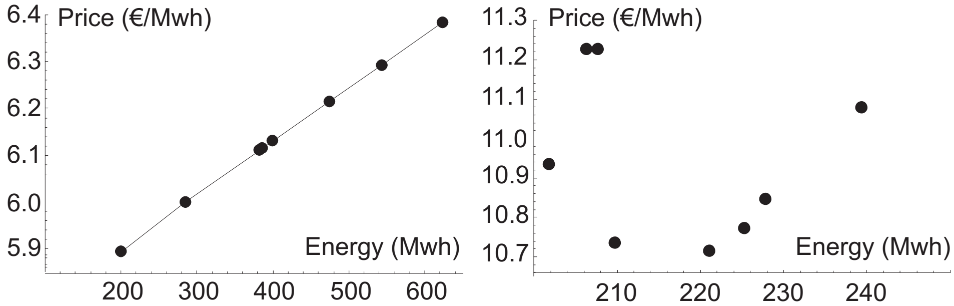


Fig. 2. (a) Optimal hydro-power. (b) Optimal thermal-power.

The solution yields the optimal offers that the company must submit to each of the day-ahead market auctions. Figure 3-a shows the offers corresponding to the 4th auction for the total optimal thermal-power, and for the eight possible realizations. The 8 quantities and 8 prices for that hour constitute the offer curve (nondecreasing) that the company must submit to that auction.



(*a*) (*b*)

Fig. 3. (a) Thermal-offers. (b) Hydro-offers.

These results can be easily analyzed. Figure 3-a shows that the offer curve obtained for the 4th hourly auction is quite flat, thus making the company rather uncertain about the amount of energy that it will finally sell. This is confirmed by Figure 2-b, in which the company's eight possible levels of sales for the 4th hour are very different. However, it is not possible to construct an offer curve (nondecreasing) for the company's optimal hydro-power. Figure 3-b shows the offers corresponding to the 12th auction for the total optimal hydro-power, and for the eight possible realizations. It is easy to understand that this behaviour is due to the inter-temporal constraints for the hydro-plants, besides the pumped-storage character of some of the hydro-plants (the optimal hydro-solution of one of the auctions influences the rest of the auctions). Therefore, we suggest that the optimal offers that the company must submit, for the hydro-plants, must be a half value of the optimal hydro-power generation that we present in Figure 2-a.

## References

[1] M. Ventosa, A. Baíllo, M. Rivier and A. Ramos, *Electricity Market Modeling Trends*, Energy Policy **33(7)** (2005) 897–913.

[2] G. Gross and D.J. Finlay, *Optimal Bidding Strategies in Competitive Electricity Markets*, Proc. 12th PSCC (1996) 815–823.

[3] J. Valenzuela and M. Mazumdar, *Probabilistic Unit Commitment under a Deregulated Market*, in: B. F. Hobbs, M. H. Rothkopf, R. P. O'Neill and H.-P. Chao, (Eds.), *The Next Generation of Electric Power Unit Commitment Models*, Kluwer Academic Publishers, Boston, (2001) 139–152.

[4] Compañía Operadora del Mercado Español de Electricidad, S.A. *Electricity market activity rules.* http://www.omel.es/es/pdfs/EMRules.pdf.

[5] A.J. Conejo, F.J. Nogales and J.M. Arroyo, *Price-taker bidding strategy under price uncertainty*, IEEE Trans. Power Sys. **17(4)** (2002) 1081–1088.

[6] A. Mateo, A. Muñoz and J. García, *Modeling and forecasting electricity prices with input/output hidden Markov models*, IEEE Trans. Power Sys. **20(1)** (2005) 13–24.

[7] A. Baíllo, M. Ventosa, A. Ramos and M. Rivier, *Optimal offering strategies for generation companies operating in electricity spot markets*, IEEE Trans. Power Sys. **19(2)** (2004) 745–753.

[8] L. Bayón, J. Grau, M.M. Ruiz and P.M. Suárez, *Nonsmooth Optimization of Hydrothermal Problems*, Journal of Computational and Applied Mathematics **192(1)** (2006) 11–19.

[9] F.H. Clarke, *Optimization and Nonsmooth Analysis*, SIAM, Philadelphia, 1990.

[10] L. Bayón, J. Grau, M.M. Ruiz and P.M. Suárez, *New developments on equivalent thermal in hydrothermal optimization: an algorithm of approximation*, Journal of Computational and Applied Mathematics **175(1)** (2005) 63–75.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# Carr's randomization for American options in regime-switching models

**Svetlana Boyarchenko**[1] **and Sergei Levendorskiĭ**[1]

[1] *Department of Economics, The University of Texas at Austin*

emails: `author1@example.com`, `author2@example.com`

**Abstract**

In the paper, we solve the pricing problem for American options in Markov-modulated Lévy models. The early exercise boundaries and prices are calculated using a generalization of Carr's randomization for regime-switching models. An efficient iteration pricing procedure is developed. The computational time is of order $m^2$, where $m$ is the number of states, and of order $m$, if the parallel computations are allowed. The payoffs, riskless rates and class of Lévy processes may depend on a state. Special cases are stochastic volatility models and models with stochastic interest rate; both must be modelled as finite-state Markov chains.

*Key words: regime-switching, Lévy processes, American options, stochastic interest rates, stochastic volatility*
*MSC 2000: AMS codes 60J75, 90 08, 91B28*

## 1 Introduction

Different pricing problems in regime switching models were considered in a number of papers. In the majority of publications, Markov-modulated (geometric) Brownian motion models are studied. For closed form solutions for perpetual lookback options, see [11]. Perpetual American put in the geometric Brownian motion model coupled by a two-state Markov chain was studied in [13]. In [9], a pricing procedure for the American put with finite time horizon was designed, for GBM modulated by a two state chain, under assumption that the early exercise boundary in one state is below the early exercise boundary in the other state. For further references, see the op. cit. The results for switching models with jumps are scarce due to technical difficulties, although the literature on pricing of American options under processes with jumps is fairly extensive by now – see, e.g., [1]-[8], [14], [15] and the bibliography therein. In [2], the perpetual American put in regime-switching Lévy models with phase-type jumps is studied. The technique of the paper relies on the special elegant structure of this class of models and a simple explicit structure of the payoff function. It is not clear how to generalize the technique of the cited paper for more general classes of payoff functions and processes.

In [7], [8], closed form solutions for perpetual American and real options are obtained under uncertainty modelled as a monotone function of a process with i.i.d. increments. As an application of the technique, explicit recurrent formulas for Carr's randomization under Lévy processes are derived. These models of uncertainty exhibit mean-reverting and switching features, switches being endogenous in the sense that the characteristics of the price process change as the price arrives in another region of the state space. In the paper, we extend the method of [5]-[8] coupling (monotone functions of) Lévy processes by an exogenous finite-state Markov chain, and construct regime-switching models with both endogenous and exogenous switching. Note that not only the parameters of the process but the interest rate and payoff functions are allowed to depend on the state of the Markov chain. Thus, we allow for jumps in the payoff, interest rate, volatility (and other parameters of the price process or the type of the process), and jumps in the price process itself.

In the case of perpetual American options, the optimal exercise rules can be viewed as natural extensions of the record-setting news principles in [4] and universal bad news principle in [8] to regime switching models; these principles are applied in each state of the Markov chain assuming that the value functions in all other states are given. Then the equation for the exercise boundary and explicit analytical expression for the value function are as simple as in the no-regime-switching case. After that, we prove that the resulting system of equations for the exercise boundaries and option values in all states can be easily solved using the iteration method. If the transition rates are small, iterations converge very fast. Finally, we prove that in the limit, the optimal exercise rule and option value obtain. In the case of American options with finite time horizon, we reduce to a sequence of perpetual American options. The solution of options in the sequence is similar to [5], [6], [7], [15] but simpler even for the non-switching case. Also, we improve the scheme in op.cit. allowing for time steps of varying sizes; in op.cit., the time grid was equally spaced.

# 2 PRELIMINARIES

## 2.1 Lévy processes and Expected Present Value operators

The moment generating function of a Lévy process can be represented in the form $E\left[e^{zX_t}\right] = e^{t\Psi(z)}$; the function $\Psi$ is called the Lévy exponent. The latter naturally appears when one calculates the action of the infinitesimal generator of $X_t$, denoted $L$, on exponential functions: $Le^{zx} = \Psi(z)e^{zx}$. As a basic example, we will use the process which appeared in [10] for the first time:

$$\Psi(z) = \frac{\sigma^2}{2}z^2 + bz + \frac{c^+z}{\lambda^+ - z} + \frac{c^-z}{\lambda^- - z}; \tag{1}$$

the results are valid for Lévy processes satisfying the (ACP)-property.

Let $T \sim \mathrm{Exp}(q)$. Then

$$E^x[g(X_T)] := qE^x\left[\int_0^{+\infty} e^{-qt} g(X_t)dt\right].$$

Introduce the normalized resolvent or *expected present value operator: EPV-operator* of a stochastic process $X$:

$$\mathcal{E}g(x) = qE^x\left[\int_0^{+\infty} e^{-qt} g(X_t)dt\right].$$

This operator calculates the EPV of a stream $qg(X_t)$. Applying $\mathcal{E}$ to $g(x) = e^{zx}$ and using the equality $E\left[e^{zX_t}\right] = e^{t\Psi(z)}$, we obtain that $\mathcal{E}$ acts on exponential functions as the multiplication operator by the number $q(q-\Psi(z))^{-1}$. To ensure that the expectation is finite, it is necessary and sufficient that the real part of $q - \Psi(z)$ is positive. Since $(q - L)e^{zx} = (q - \Psi(z))e^{zx}$, we conclude that $q^{-1}(q - L)$ and $\mathcal{E}$ are mutual inverses. To make this statement precise, we need to specify function spaces between which $q^{-1}(q-L)$ and $\mathcal{E}$ act. We will also need the normalized EPV-operators of the supremum process $\bar{X}_t = \sup_{0\le s\le t} X_s$ and the infimum process $\underline{X}_t = \inf_{0\le s\le t} X_s$. These EPV-operators act as follows:

$$\mathcal{E}^+g(x) := qE^x\left[\int_0^{\infty} e^{-qt} g(\bar{X}_t)dt\right], \quad \mathcal{E}^-g(x) := qE^x\left[\int_0^{\infty} e^{-qt} g(\underline{X}_t)dt\right].$$

Evidently, $\mathcal{E}^+g(x) = E^x[g(\bar{X}_T)]$ and $\mathcal{E}^-g(x) = E^x[g(\underline{X}_T)]$, where $T$ is the exponential random variable introduced at the beginning of this subsection. It is straightforward to check that $\mathcal{E}^+$ and $\mathcal{E}^-$ also act on an exponential function $e^{zx}$ as multiplication operators by numbers, which we denote $\kappa^+(z)$ and $\kappa^-(z)$, respectively:

$$\mathcal{E}^+e^{zx} = \kappa^+(z)e^{zx}, \quad \mathcal{E}^-e^{zx} = \kappa^-(z)e^{zx}. \tag{2}$$

These numbers are $\kappa^+(z) = E\left[e^{z\bar{X}_T}\right], \kappa^-(z) = E\left[e^{z\underline{X}_T}\right]$. Note that to simplify the notation, we suppress the dependence of the EPV-operators $\mathcal{E}, \mathcal{E}^\pm$ and numbers $\kappa^\pm(z)$ on $q$ (and on the process $X$).

## 2.2 Wiener-Hopf factorization

The Wiener-Hopf factorization formula reads:

$$E[e^{zX_T}] = E[e^{z\bar{X}_T}]E[e^{z\underline{X}_T}],\ \forall\ z \in i\mathbb{R}.$$

Equivalently, $\forall\ z \in i\mathbb{R}$

$$q/(q - \Psi(z)) = \kappa^+(z)\kappa^-(z). \tag{3}$$

Applying $\mathcal{E}$, $\mathcal{E}^+$ and $\mathcal{E}^-$ to $g(x) = e^{zx}$ and using (2)-(3), we obtain the third version of the Wiener-Hopf factorization formula:

$$\mathcal{E}g(x) = \mathcal{E}^+\mathcal{E}^-g(x) = \mathcal{E}^-\mathcal{E}^+g(x). \tag{4}$$

By linearity, (4) holds for linear combinations of exponents and integrals of exponents, hence for wide classes of functions. Equation (4) means that the normalized EPV-operator of a Lévy process admits a factorization into a product of the normalized EPV-operators of the supremum and infimum processes.

### 2.3 Example

For the Lévy process with the characteristic exponent (1), the Wiener-Hopf factors are

$$\kappa^+(z) = \prod_{l=1,2} \frac{\beta_l^+}{\beta_l^+ - z}, \quad \kappa^-(z) = \prod_{l=1,2} \frac{\beta_l^-}{\beta_l^- - z},$$

where $\beta_2^- < \lambda^- < \beta_1^- < 0 < \beta_1^+ < \lambda^+ < \beta_2^+$ are the roots of the "characteristic equation" $q - \Psi(\beta) = 0$. EPV-operators acts as follows

$$\mathcal{E}^+ u(x) = \sum_{l=1,2} a_l^+ \beta_l^+ \int_0^{+\infty} e^{-\beta_l^+ y} u(x+y)dy,$$

$$\mathcal{E}^- u(x) = \sum_{l=1,2} a_l^- (-\beta_l^-) \int_{-\infty}^{0} e^{-\beta_l^- y} u(x+y)dy,$$

where $a_1^\pm, a_2^\pm > 0$ come from the decomposition of $\kappa^\pm(z)$ into the sum of simple fractions.

## 3 CARR'S RANDOMIZATION IN REGIME-SWITCHING MODELS

Let $\lambda_{jk}$ be the transition rate from state $j$ to state $k$. The riskless rate in state $j$ is $q_j$. The infinitesimal generator of the driving Lévy process $X_t^{(j)}$ in state $j$ is denoted by $L_j$, and the Lévy exponent of the process $X_t^{(j)}$ – by $\Psi_j$. We assume that a switch from state $j$ to state $k$ and a jump of $X_t^{(j)}$ do not happen simultaneously, a.s. However, we may produce simultaneous switches and jumps playing with different payoff functions in different states of the Markov chain.

Consider the American option with the payoff $G_j(X_t)$ in state $j$ and time horizon $T$. Here and below, we assume that an index $j$ means that $X_t$ evolves as $X_{t-\tau^j}^{(j)}$, where $\tau^j$ is the last moment of a switch to the state $j$, starting at the point, where switch happened. Assume that all functions $G_j$ are continuous, non-increasing and change sign; then, at sufficiently low levels of the stochastic factor, in some states (possibly, not all), it may be optimal to exercise the option. The standard example is $G_j(x) = K_j - B_j e^x$, where $K_j$ and $B_j$ are positive. $K_j$ can be interpreted as the strike price in state $j$, and $B_k/B_j$ as a jump factor in the stock price at a moment of a switch from state $j$ to state $k$. For simplicity, we assume that if the discount rate $q$ is sufficiently large, then each payoff $G_j(X_j^{(j)})$ can be represented as the EPV of some stream:

$$G_j(x) = E_j^x \left[ \int_0^{+\infty} e^{-qt} g_j(X_t^{(j)})dt \right], \tag{5}$$

where the function $g_j = g_{q,j}$ is continuous, non-increasing, positive in a neighborhood of $-\infty$, and changes sign. The subscript $j$ in $E_j^x$ means that the expectation is calculated assuming that $X_t$ will follow the Lévy process $X_t^{(j)}$ and there will be no switches. If $|G_j(x)|$ grows as $x \to \infty$ then (5) imposes certain restrictions on the Lévy exponents $\Psi_j$. For instance, if $G_j$ and their derivatives $G_j^{(s)}$, $s \le 2$, satisfy the bound

$$|g(x)| \le C(e^{\sigma^- x} + e^{\sigma^+ x}), \tag{6}$$

where $\sigma^- \le 0 \le \sigma^+$, then (5) can be satisfied with any $q > \max\{\Psi_j(\sigma^-), \Psi_j(\sigma^+)\}$ and $g_j = (q - L_j)G_j$ provided $\Psi_j(z)$ are defined for $z \in [\sigma^-, \sigma^+]$. For each state $j$, the option owner needs to find a stopping time $\tau_j \le T$, which maximizes

$$V_j(t,x) = E^{j,x}\left[\int_0^{\tau_j} e^{-(q_j+\Lambda_j)t}\sum_{k\neq j}\lambda_{jk}V_k(t,X_t)dt\right] + E^{j,x}\left[e^{-\tau_j(q_j+\Lambda_j)}G_j(X_{\tau_j})_+\right].$$

to approximate the optimal stopping problem with finite time horizon by sequences of optimal stopping problems with infinite time horizon. A sequence of optimal stopping problems with infinite time horizon is determined by a partition $0 = t_0 < t_1 < \cdots < t_N = T$ of the interval $[0,T]$, and a "staircase" $x = h^s, t_s < t \le t_{s+1}, s = 0, 1, \ldots, N-1$, which approximates the early exercise boundary. In the regime-switching version, we need to introduce a staircase for each state $j$: $x = h_j^s, t_s < t \le t_{s+1}, s = 0, 1, \ldots, N-1$. In the case of put-like options, $G_j$ are non-increasing functions, which change sign, and in Carr's approximation, the option is exercised when the state $j$ staircase is reached or crossed from above, in state $j$; in the case of call-like options, $G_j$ are non-decreasing functions, which change sign, and in Carr's approximation, the option is exercised when the state $j$ staircase is reached or crossed from below, in state $j$.

Let $v_j^s(x)$ be the approximation to the option value $V_j(t_s, x)$ in state $j$, at time $t_s$, and set $q_j^s = \Delta_s^{-1} + q_j + \Lambda_j$. Assume that the approximations $v_{i,*}^{s+1}$ on the next time step are known for all $i$, and, for a given $j$, the approximations $v_{k,*}^s, k \neq j$, are known. We assume that these functions are continuous and satisfy bound (6). For $s = N$, $v_{j,*}^N(x) = G_j(x)_+$ are known and satisfy bound (6). We can interpret $v^s = (v_j^s)_{j=1}^m$ as the value of the option to swap the stream $(\Delta_s^{-1}v_{j,*}^{s+1}(X_t))_{j=1}^m$ for the instantaneous payoff $(G_j(X_t))_{j=1}^m$; the value function, stream and payoff are functions on $\mathbb{R}^m$, the state space of the regime switching process $X_t$.

For $s = N-1, N-2, \ldots$, we need to choose a stopping time, $\tau_j^s$ so that $v_j^s(x)$ is maximized, and we will show that $\tau_j^s$ is the hitting time of a semi-infinite interval of the form $(-\infty, h_j^s]$. The corresponding free boundary problem is

$$\begin{aligned}(q_j^s - L_j)v_j^s(x) &= \Delta_s^{-1}v_{j,*}^{s+1}(x) + \sum_{k\neq j}\lambda_{jk}v_{k,*}^s(x),\\ v_j^s(x) &= G_j(x),\end{aligned} \tag{7}$$

for $x > h_j^s$ and $x \le h_j^s$, respectively. We replaced $G_j(x)_+$ in (7) with $G_j(x)$ because the stream $\Delta_s^{-1}v_{j,*}^{s+1}(X_t) + \sum_{k\neq j}\lambda_{jk}v_{k,*}^s(X_t)$ is non-negative, hence, it is non-optimal to exercise the option unless $G_j(X_{\tau_j^s}) > 0$.

Note that we do not use a scheme which is implicit w.r.t. $v_j^s$ but explicit w.r.t. $v_k^s, k \neq j$, that is, the scheme with $v_{k,*}^{s+1}, k \neq j$, in the RHS, because this scheme is less accurate than the scheme which is implicit w.r.t. all $v_j^s$. However, the latter requires more iterations. If the transition probabilities $\lambda_{jk}$ are small, the number of iterations on each time step is small for both schemes.

## 3.1 Solution of the sequence of perpetual options

For each $s = N-1, N-2, \ldots$, we find the set of optimal exercise boundaries $\{h_j^s\}_{j=1}^m$ in three steps. First, we derive a system for the optimal exercise boundaries and option values $\{v_j^s\}_{j=1}^m$ assuming that the option values are sufficiently regular. Then we construct an iteration procedure, which converges to a solution of this system. We use the procedure to prove that this solution is sufficiently regular, and that it gives optimal exercise rules and option values. Finally, we prove that the solution is unique.

Introduce $\tilde{v}_j^s = v_j^s - G_j$, $\tilde{v}_{k,*}^s = v_{k,*}^s - G_k$, $\tilde{g}_j = \sum_{k\neq j} \lambda_{jk} G_k - (q_j + \Lambda_j - L_j)G_j$. Assume that functions $G_j$ and their derivatives up to order 2 satisfy bound (6); then functions $g_j^s = (q_j^s - L_j)G_j$ are well-defined. We choose $\Delta_s$ sufficiently small so that $q_j^s$ is sufficiently large in the sense that for $j = 1, 2, \ldots, m$,

$$q_j^s - \Psi_j(\sigma) > 0, \quad \forall\, \sigma \in [\sigma^-, \sigma^+]. \tag{8}$$

Using $q_j^s$ and $L_j$ in place of $q$ and $L$, we define the EPV-operators $\mathcal{E}_j^s, \mathcal{E}_j^{s,-}, \mathcal{E}_j^{s,+}$.

**Theorem 3.1** *Let the following conditions hold:*

*(i)* $\Delta_s$ *and the Lévy exponents* $\Psi_j$ *satisfy (8);*

*(ii)* $g_j^s = (q_j^s - L_j)G_j$ *are non-decreasing functions that change sign, and* $G_j(+\infty) = -\infty$*;*

*(iii)* $\tilde{g}_j$ *are continuous non-decreasing functions, and* $\tilde{g}_j(-\infty) < 0$;

*(iv) for* $s = l+1$ *and all* $k$*, functions* $\tilde{v}_{k,*}^s$ *are known;*

*(v) for some* $j$*, functions* $\tilde{v}_{k,*}^l$ *in states* $k \neq j$ *are known.*

*Then for the same* $j$ *and* $s = l$*,*

*a) function* $\tilde{g}_j^s = \sum_{k\neq j} \lambda_{jk}\tilde{v}_{k,*}^s + \Delta_s^{-1}\tilde{v}_{j,*}^{s+1} + \tilde{g}_j$ *is a non-decreasing continuous function satisfying bound (6); in addition,*

$$\tilde{g}_j^s(-\infty) < 0 < \tilde{g}_j^s(+\infty) = +\infty; \tag{9}$$

*b) function* $\tilde{w}_j^s := \mathcal{E}_j^{s,+}\tilde{g}_j^s$ *is continuous. It increases and satisfies (9), therefore equation* $\tilde{w}_j^s(h) = 0$ *has a unique solution, denote it* $h_{j,*}^s$*;*

*c) the hitting time of* $(-\infty, h_{j,*}^s]$, $\tau_-(h_{j,*}^s)$*, is a unique optimal stopping time;*

*d) Carr's approximation to state-j option value is*

$$v_{j,*}^s = (q_j^s)^{-1}\mathcal{E}_j^{s,-}\mathbf{1}_{(h_{j,*}^s,+\infty)}\tilde{w}_j^s + G_j;$$

*e)* $\tilde{v}_{j,*}^s = v_{j,*}^s - G_j$ *is a positive non-decreasing function that admits bound (6) and satisfies* $\tilde{v}_{j,*}^s(+\infty) = +\infty$*; it vanishes below* $h_{j,*}^s$ *and increases on* $[h_{j,*}^s, +\infty)$*.*

### 3.2 Iteration procedure

For $s = N$, the state-$j$ option value is known: $v^N_{j,*}(x) = G_j(x)_+$, and the state-$j$ exercise boundary $h^n_{j,*}$ is a unique root of the equation $G_j(h) = 0$, $j = 1, 2, \ldots, m$. In the regime-switching version of Carr's randomization procedure, we need to calculate the (approximations to the) exercise boundaries $h^s_{j,*}$ and option values $v^s_{j,*}$, $j = 1, 2, \ldots, m$, for $s = l < N$, assuming that for $l + 1 \le s \le N$ and $j = 1, 2, \ldots, m$, $h^s_{j,*}$ and $v^s_{j,*}$ are known. In the non-switching case, we were able to derive explicit formulas for the (Carr's approximations to the) exercise boundaries and option values on step $s$ in terms of the option values on the previous step. Here, in the regime-switching case, we cannot derive explicit analytical formulas but we can design an efficient iteration procedure instead. For each $s = n - 1, n - 2, \ldots$ and $j = 1, 2, \ldots, m$, we construct sequences $\{h^{sn}_j\}^\infty_{n=0}$ and $\{v^{sn}_j\}^\infty_{n=0}$, such that the limits

$$\bar{h}^s_j = \lim_{n\to+\infty} h^{sn}_j, \quad \bar{v}^s_j = \lim_{n\to+\infty} v^{sn}_j \tag{10}$$

are the optimal exercise boundary, $h^s_{j,*}$, and option value, $v^s_{j,*}$ (in the switching analog of Carr's randomization procedure, at time step $s$ and in state $j$). Thus, for each $s$, we need to introduce an additional cycle in $n$; and inside the cycle in $n$, we will use additional cycles in $j = 1, 2, \ldots, m$.

Fix $s < N$. Below, we formulate the iteration procedure for $w^s_{j,*}$, $\tilde{w}^s_{j,*}$, $h^s_{j,*}$, $v^s_{j,*}$ and $\tilde{v}^s_{j,*}$. Calculate

$$w^s_{0j} = -\mathcal{E}^{s,+}_j(q^s_j - L_j)G_j = q^s_j(\mathcal{E}^{s,-}_j)^{-1}G_j,$$

set $v^{s0}_j = 0, h^{s0}_j = +\infty, j = 1, 2, \ldots, m$, and for $n = 1, 2, \ldots,$ define, step by step, in the interior cycle in $j = 1, 2, \ldots, m$,

(i) functions $\tilde{w}^{sn}_j = w^{sn}_j + w^s_{0j}$, where

$$w^{sn}_j = \mathcal{E}^{s,+}_j\left(\sum_{k\neq j}\lambda_{jk}v^{s,n-1}_k + \Delta^{-1}_s v^{s+1}_{j,*}\right);$$

(ii) $h^{sn}_j$, a solution of the equation $\tilde{w}^{sn}_j(h) = 0$;

(iii) functions

$$\begin{aligned} v^{sn}_{0j} &= (q^s_j)^{-1}\mathcal{E}^{s,-}_j\mathbf{1}_{(-\infty,h^{sn}_j]}(-w^s_{j0}), \\ v^{sn}_{1j} &= (q^s_j)^{-1}\mathcal{E}^{s,-}_j\mathbf{1}_{(h^{sn}_j,+\infty)}w^{si}_j, \\ v^{sn}_j &= v^{sn}_{1j} + v^{sn}_{0j}, \\ \tilde{v}^{sn}_j &= v^{sn}_j - G_j. \end{aligned}$$

**Theorem 3.2** *Let conditions (i)–(iii) of Theorem 3.1 hold. Then* $\forall\ s = N-1, N-2, \ldots$, $j = 1, 2, \ldots, m$ *and* $n = 1, 2, \ldots$

Table 1: Parameters of the processes

| $m$ | $\sigma$ | $b$ | $c_-$ | $\lambda_-$ | $c_+$ | $\lambda_+$ |
|---|---|---|---|---|---|---|
| 1 | 0.25 | -0.03 | 0.02 | -5 | 0.015 | 10 |
| 2 | 0.15 | -0.02 | 0.01 | -4 | 0.01 | 12 |
| 3 | 0.2 | -0.02 | 0.15 | -4.5 | 0.1 | 13 |

*a) $\tilde{w}^{sn}_j$ is continuous increasing function that changes sign, hence, $h^{sn}_j$ is well-defined;*

*b) $w^{s,n+1}_j > w^{s,n}_j$, $h^{sn}_j < h^{s,n-1}_j$, and $v^{sn}_j > v^{s,n-1}_j$ on $[h^{sn}_j, +\infty)$;*

*c) $\tilde{v}^{sn}_j(x) = 0$, $x \le h^{sn}_j$, and $\tilde{v}^{sn}_j$ increases on $[h^{sn}_j, +\infty)$;*

*d) sequences $\{h^{sk}_j\}^{+\infty}_{k=1}$ and $\{v^{sk}_j\}^{+\infty}_{k=1}$ are bounded from below and above, respectively; hence, there exist finite limits (10);*

*e) limits (10) are the optimal time-step-s-state-j exercise boundary, $h^s_{j,*}$, and option value, $v^s_{j,*}$; the latter is positive;*

*f) function $\tilde{v}^s_{j,*} = v^s_{j,*} - G_j$ is continuous; it satisfies (6), vanishes below $h^s_{j,*}$, increases on $[h^s_{j,*}, +\infty)$, and $\tilde{v}^s_{j,*}(+\infty) = +\infty$.*

### 3.3 Numerical example

The generator of the Markov chain is

$$\begin{pmatrix} -0.03 & 0.01 & 0.02 \\ 0.015 & -0.025 & 0.01 \\ 0.015 & 0.03 & -0.045 \end{pmatrix}.$$

In Table 1, we list parameters of processes in different states. Riskless rates: $q_1 = 0.05, q_2 = 0.055, q_3 = 0.045$; payoff functions: $G_1(x) = 2-e^x, G_2(x) = 2.5-2e^x, G_3(x) = 3 - 1.5e^x$. Numerical experiments demonstrate that the method is sufficiently accurate even for a very long time interval $\tau = 50$ with more than 150 time steps. In the example below, we divided interval $[0, 25]$ into 25 subintervals using the geometric progression with the first term 0.005, then we used the uniform spacing $\Delta_s = \Delta_{25}$ till $t = 15$, and after that, the uniform spacing with $\Delta_s = 2 * \Delta_{25}$ till $t = 50.1186$. Close to expiry, state-1 boundary seems lower than Figure 1 and Table 2 show because it drops down rather fast, and the part adjacent to 0 almost coincides with the vertical axis. $h_{j,*}(0+)$ denote the limit of the early exercise boundary at expiry. The limits are calculated using the following theorem, which generalizes the result in [5], [15] for Lévy processes.

**Theorem 3.3** *Let $(j, x)$ be in the money region, that is, $G_j(x) > 0$, but*

$$0 \;<\; (q_j + \Lambda_j - L_j)(-G_j)(x) + \sum_{k \neq j} \lambda_{jk} G_k(x)_+$$

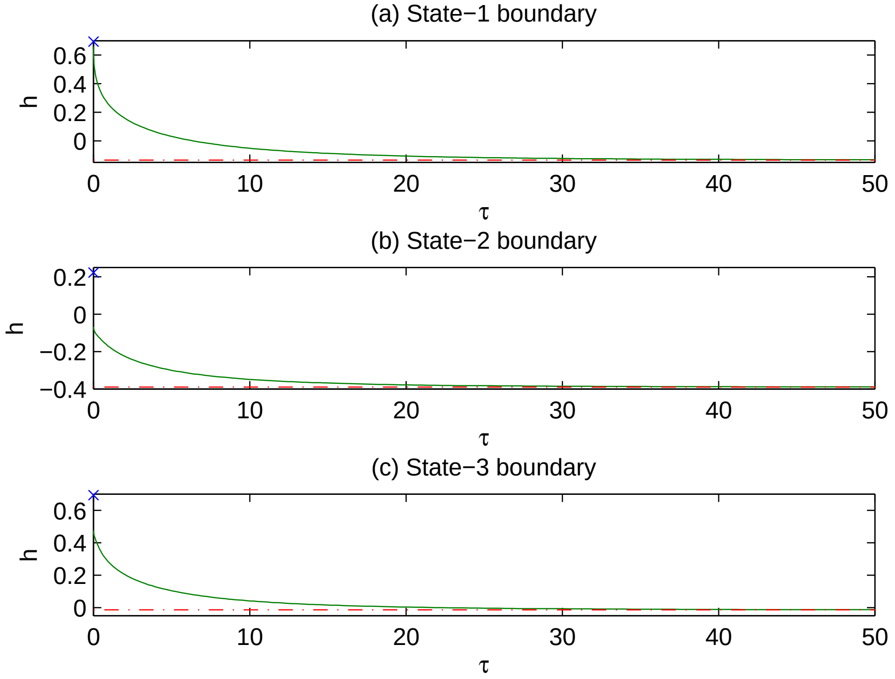


Figure 1: Early exercise boundaries $h_{j,*}(\tau)$ for $\tau \le 50$ to expiry. Parameters are in Table 1. Crosses: $h_{j,*}(0)$. Dash-dotted lines: early exercise boundaries for the perpetual option. Solid lines: early exercise boundaries.

$$+\int_{-\infty}^{+\infty}(-G_j(x+y))_+F_j(dy). \tag{11}$$

*Then there exists $\tau_0 = \tau_0(j,x)$ such that for each $\tau \in (0,\tau_0)$, it is not optimal to exercise the option as $X_{T-\tau} = (j,x)$.*

Here $F_j(dy)$ is the state-$j$ Lévy density. In our example, (11) has a unique solution, and it follows from Theorem 3.3, that this solution is not larger than the limit of the early exercise boundary at expiry. We cannot prove that it is the limit and not just an upper bound but the numerical results shown below support the claim that the solution is the limit. In this example, $h_{j,*}(0+)$ is strictly below $h_{j,*}(0)$ for each $j$, which is caused by the interaction of three factors: the stock pays dividends because $q_j - \Psi_j(1) > 0$ in this example; each Lévy process $X_t^{(j)}$ has the non-trivial positive jumps component; at a moment of a jump, the payoff may jump up. We will analyze the interaction among these three factors in a separate publication.

## 4 Conclusion

In the paper, we derived the iteration procedure for the American put with finite the horizon in Markov-modulated Lévy models. The procedure does not use any apriori assumption on the relative location of the early exercise boundaries in different states. The riskless rates, payoff functions and Lévy processes depend on a state. Lévy processes are assumed to satisfy the (ACP)-property, and payoff functions can be more general than for the standard put and call options. The procedures are efficient if the transition rates are not large w.r.t. the riskless rates. The CPU time is of order $m^2$, where $m$ is the number of states; if parallel computations are used, then the CPU time is of order $m$. Numerical example show that the method is accurate for options of different maturities. Method can be applied to stochastic interest models and stochastic volatility models; both must be modelled as finite state Markov chains. Jumps at moment of switching can be used to model a mean-reverting component of the price process; this component may correlate with the stochastic interest rate and volatility.

The method of the paper admit extension in several directions: approximation of the standard stochastic interest rate/volatility models (with possible switching) by models constructed in the paper; credit risk models; optimal exercise with learning the current state; real options, especially, with strategic interactions; and other.

## Acknowledgements

This work has been partially supported by NSF grant #DMS-0631628.

## References

[1] L. Alili and A. Kyprianou, *Some remarks on first passage of Lévy process, the American put and pasting principles*, Annals of Applied Probability **15** (2005) 2062–2080.

[2] S. Asmussen, F. Avram and M. R. Pistorius, *Russian and American put options under exponential phase-type Lévy models*, Stochastic Processes and Applications **109** (2004) 79–111.

[3] F. Avram, A. E. Kyprianou and M. R. Pistorius, *Exit problems for spectrally negative Lévy processes and applications to (Canadized) Russian options*, The Annals of Applied Probability **14** (2004) 215–238.

[4] S. I. Boyarchenko, *Irreversible Decisions and Record-Setting News Principles*, American Economic Review, **94** (2004) 557–568.

[5] S. I. Boyarchenko and S. Z. Levendorskiĭ, *Perpetual American Options under Lévy Processes*, SIAM Journal on Control and Optimization **40** (2002) 1663–1696.

[6] S. I. Boyarchenko and S. Z. Levendorskiĭ, *Non-Gaussian Merton-Black-Scholes theory*, World Scientific, Singapore, 2002.

[7] S. I. Boyarchenko and S. Z. Levendorskiĭ, *American options: the EPV pricing model*, Annals of Finance **1** (2005) 267–292.

[8] S. I. Boyarchenko and S. Z. Levendorskiĭ, *General Option Exercise Rules, with Applications to Embedded Options and Monopolistic Expansion*, Contributions to Theoretical Economics **6** (2006) Article 2

[9] J. Buffington and R.J. Elliott, *American options with regime switching*, International Journal of Theoretical and Applied Finance, **5** (2002) 497–514.

[10] D. Duffie, J. Pan and K. Singleton, *Transform Analysis and Asset Pricing for Affine Jump Diffusions*, Econometrica **68** (2000) 1343–1376.

[11] X. Guo, *An explicit solution to an optimal stopping problem with regime switching*, Journal of Applied Probability **38** (2001) 464–481.

[12] X. Guo, J. Miao and E. Morellec, *Irreversible investment with regime shifts*, Journal of Economic Theory **122** (2005) 37–59.

[13] X. Guo and Q. Zhang, *Closed form solutions for perpetual American put options with regime switching*, SIAM Journal of Applied Mathematics **64** (2004) 2034–2049.

[14] A. E. Kyprianou and M. R. Pistorius, *Perpetual options and Canadization through fluctuation theory*, Annals of Applied Probability **13** (2003) 1077-1098.

[15] S. Z. Levendorskiĭ, *Pricing of the American put under Lévy processes*, International Journal of Theoretical and Applied Finance **7** (2004) 303–336.

# ANALYSIS OF THE PML METHOD APPLIED TO SCATTERING PROBLEMS AND THE COMPUTATION OF RESONANCES IN OPEN SYSTEMS.

JAMES H. BRAMBLE, **JOSEPH E. PASCIAK**, AND SEUNGIL KIM

**Abstract:** In this talk, I shall consider the application of the so-called perfectly matched layer (PML) to acoustic and electromagnetic scattering problems as well as discuss how PML can be used to reformulate problems of computing resonances in open systems.

One of the main difficulty with the numerical approximation of scattering problems is that such problems are posed on unbounded domains with a boundary condition at infinity given by either the Sommerfeld condition or, in the case of electromagnetics, the Silver-Müller condition. There are numerous approaches for dealing with the boundary condition at infinity, e.g., truncating the domain and applying an approximate boundary condition, truncating the domain and using boundary integral techniques, using infinite element techniques, or truncating the domain and using a ficticious absorbing boundary layer (PML). A discussion of these approaches can be found in [1, 2, 8, 9, 12] and the included references. Computational evidence suggests that the PML approach is competitive and effective. Moreover, it is simple to implement in any suitably general code developed for bounded domain computations.

For scattering problems, the PML method has a long history starting from papers by Bérenger [1, 2]. The original approach involed the introduction of additional variables so that the equations could be split in an appropriate way. Subsequently, PML formulations were developed which avoided this splitting [6, 13]. Chew and Weedon [6] provided significant insight into the understanding of PML by viewing the method as a complex coordinate shift.

One can view the PML process as two distinct steps. First, a PML problem is posed on the infinite domain whose solution coincides with that of the original problem on a finite domain (region of interest) while decaying rapidly (exponentially) outside. It is this decay that allows one to truncate the problem to a bounded domain not much larger than the domain of interest and to apply any convenient boundary condition on the resulting ficticious boundary. Many authors have studied applications of PML showing the existence and uniqueness of solutions for the truncated problem and its exponential convergence to the infinite domain PML problem (on the region of interest) [5, 7, 10, 11]. However, existence and uniqueness results alone are not enough to guarantee stable numerical approximation. In [4, 5], we show that under appropriate conditions on the PML parameters, the PML method leads to continuous and discrete systems which satisfy inf-sup conditions with uniform constants independent of the truncation domain size and mesh size. Such estimates are necessary to conclude stable and convergent numerical approximation and will be presented in this talk.

I will also consider the application of PML to the problem of computing resonances in open systems, specifically, acoustic resonances. Resonances are related to improper eigenfunctions for the Helmholtz operator satisfying an outgoing boundary condition. They are improper in the sense that they grow exponentially at infinity. We shall see that the application of PML converts the resonance problem to a standard eigenvalue problem (still on an infinite domain). This new eigenvalue

problem involves an operator which resembles the original Helmholtz equation transformed by a complex shift in coordinate system. Our goal will be to approximate the shifted operator first by replacing the infinite domain by a finite (computational) domain with a convenient boundary condition and second by applying finite elements on the computational domain. I will give theorems which show that the first of these steps leads to eigenvalue convergence to the desired resonance values and are free from spirious computational eigenvalues provided that the size of computational domain is sufficiently large. The analysis of the second step is classical (see [3] and the included references).

I will conclude the talk with numerical results for PML applied to scattering problems and the resonance problem. These results will show that PML leads to an effective solution technique for both of these problems.

## REFERENCES

[1] J.-P. Bérenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 114(2):185–200, 1994.

[2] J.-P. Bérenger. Three-dimensional perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 127(2):363–379, 1996.

[3] J. H. Bramble and J. E. Osborn. Rate of convergence estimates for nonselfadjoint eigenvalue approximations. *Math. Comp.*, 27:525–549, 1973.

[4] J. H. Bramble and J. E. Pasciak. Analysis of a finite element PML approximation for the three dimensional time-harmonic Maxwell problem. *Math. Comp.* (to appear).

[5] J. H. Bramble and J. E. Pasciak. Analysis of a finite PML approximation for the three dimensional time-harmonic Maxwell and acoustic scattering problems. *Math. Comp.*, 76(258):597–614, 2007.

[6] W. Chew and W. Weedon. A 3d perfectly matched medium for modified Maxwell's equations with streched coordinates. *Microwave Opt. Techno. Lett.*, 13(7):599–604, 1994.

[7] F. Collino and P. Monk. The perfectly matched layer in curvilinear coordinates. *SIAM J. Sci. Comp.*, 19(6):2061–2090, 1998.

[8] M. Grote and J. Keller. Nonreflecting boundary conditions for Maxwell equations. *J. Comput. Phys.*, 139:327–342, 1998.

[9] F. Ihlenburg. *Finite element analysis of acoustic scattering*, volume 132 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1998.

[10] M. Lassas and E. Somersalo. On the existence and convergence of the solution of PML equations. *Computing*, 60(3):229–241, 1998.

[11] M. Lassas and E. Somersalo. Analysis of the PML equations in general convex geometry. *Proc. Roy. Soc. Edinburgh Sect. A*, 131(5):1183–1207, 2001.

[12] P. Monk. A finite element method for aproximating the time-harmonic Maxwell equations. *Numer. Math.*, 63:243–261, 1992.

[13] Z. Sacks, D. Kingsland, R. Lee, and J. Lee. A perfectly matched anisotropic absorber for use as an absorbing boundary condition. *IEEE Trans. Antennas Propagat.*, 43:1460–1464.

DEPARTMENT OF MATHEMATICS, TEXAS A&M UNIVERSITY, COLLEGE STATION, TX 77843-3368.
*E-mail address*: bramble@math.tamu.edu

DEPARTMENT OF MATHEMATICS, TEXAS A&M UNIVERSITY, COLLEGE STATION, TX 77843-3368.
*E-mail address*: pasciak@math.tamu.edu

DEPARTMENT OF MATHEMATICS, TEXAS A&M UNIVERSITY, COLLEGE STATION, TX 77843-3368.
*E-mail address*: sgkim @math.tamu.edu

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE2007*
*Chicago, 20-23 June 2007*

# The Jordan Form and Its Use in Chemical Physics and Physical Chemistry

**Erkki J. Brändas**

*Department of Quantum Chemistry, Uppsala University, Uppsala, Sweden*

email: Erkki.Brandas@kvac.uu.se

### Abstract

We derive and discuss a general complex symmetric form of the most general Jordan block representation and propose various examples in physics and chemistry where this extension appears necessary.

*Key words: complex symmetry, Jordan blocks, ODLRO off-diagonal long-range order, special- and general relativity*
*MSC2000: 83-A05*

## 1. Introduction

The query whether Jordan blocks of Segrè characteristics larger than one should appear in normal Jordan form representations of extended quantum dynamical formulations has been perceived many times in the past decades, see e.g. [1,2] and references therein. The question has to be elucidated a bit further. Jordan block representations of nilpotent operators are of course commonly used in quantum mechanics e.g. step operators in angular momentum algebra as well as annihilation and creation operators in second quantization. Here we focus on a different theme.

The Hamiltonian, or Liouvillian, of a dynamical system generally serves two purposes: they represent a measurable observable, the energy, and simultaneously they generate the time evolution of the system. In standard quantum mechanics the Hamiltonian-Liouvillian is by definition self-adjoint yielding real eigenvalues (with the Segrè characteristic equal to one) and unitary time evolution, see e.g. [3]. Within this framework it is clear that no Jordan blocks could or should come into view.

However, original advances in non-selfadjoint extensions of Hamiltonian/Liouvillian dynamics, [4,5] calls for the immediate incorporation of general classical canonical forms of the Jordan type [1,6]. In these extended applications, the possible occurrence of degeneracies, with Segrè characteristics larger than one, would result in an unwanted computational breakdown - usually considered as a numerical accident brought forward by the self-orthogonality of the transformed vectors.

In the next section we will take into account the form of a complex symmetric representation, which is commensurate with the non-self-adjoint extensions mentioned above. We will suggest some examples in chemistry and physics where the spectral map promotes Jordan block formations. In the final section the group theoretical structure of the transformations indicating further interesting organizations within a biological frame will be presented.

## 2. The complex symmetric form of the Jordan block

Since complex symmetric representations are routine in the majority of non-selfadjoint quantum treatments, e.g. complex scaling applications [4], it is important to know that a complex symmetric matrix invokes no constraint on the general secular problem. Gantmacher [7] proved that every square matrix is similar to a symmetric matrix and that every symmetric matrix is orthogonally similar to an explicitly given normal form.

The author has for many years considered situations in which particular complex symmetric forms have been employed, see e.g. Ref. [8]. Particular applications concern proton transfer processes and dynamics and quantum correlations in condensed matter systems, see e.g. Refs. [1,9]. The specific symmetric matrix $\mathbf{Q}$ of dimension $m$ derived and used, see Refs. [1, 6] has a very simple and useful structure defined by

$$\mathbf{Q}_{kl} = (\delta_{kl} - \frac{1}{m})\, e^{i\frac{\pi}{m}(k+l-2)};\ k,l = 1,2,..m \tag{1}$$

In other words one can prove that

$$\mathbf{Q} = \mathbf{B}^{-1}\mathbf{J}\mathbf{B} \tag{2}$$

with $\mathbf{J}$ defined as

$$\mathbf{J} = \begin{pmatrix} 0 & 1 & 0 & . & 0 \\ 0 & 0 & 1 & . & . \\ . & . & . & . & 0 \\ . & . & . & . & 1 \\ 0 & . & . & . & 0 \end{pmatrix} \tag{3}$$

and where the unitary matrix $\mathbf{B}$ that connects the standard Jordan form $\mathbf{J}$ with $\mathbf{Q}$ becomes

$$\mathbf{B} = \frac{1}{\sqrt{m}}\begin{pmatrix} 1 & \omega & \omega^2 & . & \omega^{m-1} \\ 1 & \omega^3 & \omega^6 & . & \omega^{3(m-1)} \\ . & . & . & . & . \\ . & . & . & . & . \\ 1 & \omega^{2m-1} & \omega^{2(2m-1)} & . & \omega^{(m-1)(2m-1)} \end{pmatrix}; \quad \omega = \mathrm{e}^{\frac{i\pi}{m}} \tag{4}$$

The form can be generalized to various powers $\mathbf{Q}^r$, i.e.

$$\mathbf{Q}^r_{kl} = \omega^{r(k+l-2)}[\delta_{kl} - (\mathbf{R}^r)_{kl}];\ k,l = 1,2,..m \tag{5}$$

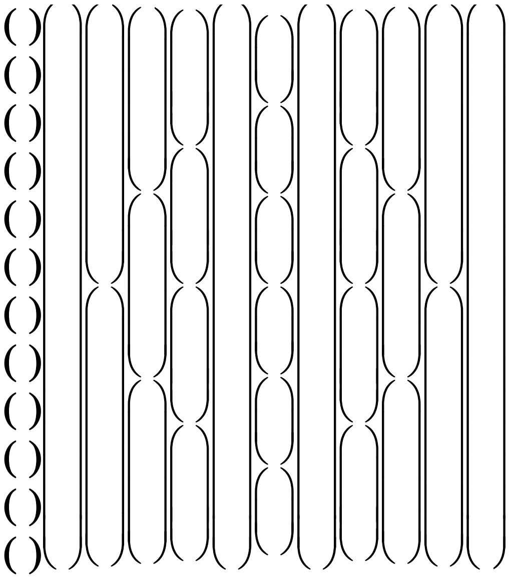
$$(\mathbf{R}^r)_{kl} = \begin{cases} \frac{1}{m}\dfrac{\sin(\frac{\pi r(l-k)}{m})}{\sin(\frac{\pi(l-k)}{m})} & k \neq l \\ \frac{r}{m} & k = l \end{cases} \tag{6}$$

## 3. The group structure – an example

To see how the transformation above lends itself to an interesting structure we consider again the transformation formula Eq. (1-4) above. Not only does the unitary transformation **B** show that a complex symmetric matrix is similar to a real matrix that display the standard Jordan form, but it does further exhibit interesting properties as a simple example will demonstrate. First we introduce a simple notation that will display the cyclic structure incorporated in **B.** Let us e.g. denote the simple column $(\omega, \omega^{3}, \omega^{5}, \cdots, \omega^{2\ -1})^{\dagger}$ , for an arbitrary *n*, with the symbol $(n)^{\dagger}$ where $n \leq m$. Choosing $m=12$ we can write for $\sqrt{12}$ **B** the symbolic form

$$\begin{array}{cccccccccccc} (1) & & & & & & 2 & & & & & \\ (1) & & & 4 & 3 & & & & 3 & 4 & & \\ (1) & & 6 & & & & 2 & & & & 6 & \\ (1) & & & & & & & & & & & \\ (1) & & & & 3 & & 2 & & 3 & & & \\ (1) & 12 & & 4 & & 12 & & 12 & & 4 & & 12 \\ (1) & & & & & & 2 & & & & & \\ (1) & & & & 3 & & & & 3 & & & \\ (1) & & 6 & & & & 2 & & & & 6 & \\ (1) & & & 4 & & & & & & 4 & & \\ (1) & & & & 3 & & 2 & & 3 & & & \\ (1) & & & & & & & & & & & \end{array} \tag{7}$$

or simply



(8)

Note that the columns (2, 12), (3, 11), (4, 10), (5,9), and (6.8) are related by the operation of multiplying the complex conjugate with a minus sign. The 2-cycle is here simply (i,-i). In general this relation is connected with the columns *s*, for *1<s≤n/2*, and *n+2-s*. The 2-cycle is always (i,-i) if n is even. The rows have a similar symmetry, i.e. for *m=12* the rows (1,12), (2,11), (3,9)…. (6,7) are complex conjugate of each other or generally row *s*, for *1<s≤n/2*, and *n+1-s* are complex conjugate of each other. For *m* odd the middle cycle is (1,-1). One might speculate what would be the consequence for the appearance of a large prime number *m=p* occurring above, since the only “repetitive vector” would be in the middle containing sub-blocks of (1,-1).

The present structure suggests tempting interpretations in the biological field, e.g. proton correlations in DNA, the origin of the screw like symmetry of the double helix and possible long term correlations of the smallest microscopic self-organizing units co-operating *in vivo* systems, see e.g. Ref. [10] for some more detailed suggestions and associated time evolutionary consequences. It is indeed an invigorating feature that Jordan blocks appear as fundamental units in an extended quantum description that also incorporates characteristics of the special and general theories of relativity [11].

## Acknowledgements

The author is indebted to Prof. Bruce Wade and Prof. Jésus Vigo-Aguiar for generous support and hospitality. Supported in part by the Swedish Foundation for Strategic Research.

## References

[1] E. BRÄNDAS, *Resonances and Dilatation Analyticity in Liouville Space,*

Adv. Chem. Phys. **99** (1997) 211-244.

[2] A. BOHM, M. GADELLA, M. LOEVE, S. MAXON, P. PATULEANU, C. PUNTMANN, *Gamow-Jordan Vectors and Nonreducible Density Operators from Higher Order S-Matrix Poles,* J. Math. Phys. **38** (1997) 6072-6100.

[3] P. O. LÖWDIN, *Linear Algebra for Quantum Theory*, Wiley, New York, 1998.

[4] E. BALSLEV, J. M. COMBES, *Spectral properties of many-body Schrödinger operators with dilatation-analytic interactions,* Commun. Math. Phys. **22** (1971) 280-294.

[5] A. BOHM, M. GADELLA, *Dirac Kets, Gamow vectors and Gel´fand Triplets,* Springer Verlag, Berlin, 1989.

[6] C.E. REID, E. BRÄNDAS, *On a Theorem for Complex Symmetric Matrices and its Relevance in the Study of Decay Phenomena,* Lecture Notes in Physics, **325** (1989) 476-483.

[7] F. R. GANTMACHER, *The theory of matrices,* Vol II, Chelsea Publishing Company, New York, 1959.

[8] E. BRÄNDAS, *Relaxation Processes and Coherent Dissipative Structures,* in Dynamics during Spectroscopic Transitions. Eds E. Lippert and J. D. Macomber, Springer Verlag, (1995) 148-193.

[9] E. BRÄNDAS, *Applications of CSM Theory,* in Dynamics during Spectroscopic Transitions. Eds E. Lippert and J. D. Macomber, Springer Verlag, (1995) 194-241.

[10] E. BRÄNDAS, *Dissipative Systems and Microscopic Selforganisation,* Adv. Quant. Chem. **41** (2002) 121-138.

[11] E. BRÄNDAS, *Quantum Mechaanics and the Special- and General Theory of Relativity,* Adv. Quant. Chem. **54** (2007) in press.

# Valuation of guaranteed annuity options in affine term structure models

**Chi Chiu Chu and Yue Kuen Kwok**[⋆]

*Department of Mathematics, Hong Kong University of Science and Technology*
*Clear Water Bay, Hong Kong*

**Abstract**

We propose three analytic approximation methods for numerical valuation of the guaranteed annuity options in deferred annuity pension policies. The approximation methods include stochastic duration approach, Edgeworth expansion and analytic approximation in affine diffusions. The payoff structure in the annuity policies is similar to a quanto call option written on a coupon bearing bond. To circumvent the limitations of the one-factor interest rate model, we model the interest rate dynamics by a two-factor affine interest rate term structure model. The numerical accuracy and computational efficiency of these approximation methods are analyzed. We also investigate the value sensitivity of the guaranteed annuity option with respect to different parameters in the pricing model.

JEL classification: G13; G23

Keywords: Guaranteed annuity option, affine term structure models, coupon-bond options, stochastic duration, Edgeworth approximation, affine approximation

## 1. Introduction

A guaranteed annuity option (GAO) provides the policyholder the right to either receive at retirement an assured accumulated funds or a life annuity at a fixed rate. This is one of the many examples of minimum return guarantees (embedded options) in life insurance policies. Pension-type policies with GAO's were popular in UK retirement saving contracts in the 1970's and 1980's. Between 1975 and 1985, UK interest rates were at a high level (typically above 10%). It was then generally perceived that the GAO's have insignificant value since these options are deeply out-of-the-money. However, for pension-type contracts having a long term, which may last for 30 years or more, the change in financial and other variables may cause the embedded GAO to become an uncontrollable liability. There are a number of factors that contribute to the acute increase in the value of these GAO's. First, the current UK interest rates stay at lower level so that the annuity value becomes higher. Second, the accumulated equity value of these contracts may increase substantially with a strong return in the stock market. Third, the improvement of mortality compared

[⋆] Corresponding author. Tel: 852-2358-7418; fax: 852-2358-1643. E-mail address: maykwok@ust.hk

to the anticipated mortality assumption when these contracts were written. Due to the significant increase in liabilities in these pension-type contracts, Equitable Life (a leading UK insurance firm) had to close for new business. Detailed accounts of the issues faced by the issuance of GAO's can be found in the review articles by O'Brien (2001) and Wilkie *et al.* (2004).

There have been numerous works on the pricing and hedging of GAO's using the option valuation approach. Boyle and Hardy (2003) provide an insightful review on the issues of pricing, reserving and hedging GAO's under interest rate risk, equity risk and mortality risk. Pelsser (2003) shows how to construct a replicating portfolio of interest rate swaptions that replicates the GAO. The swaption is seen to mimick the type of interest rate exposure faced by the GAO issuer. However, his swaption replication technique still faces problems with the equity risk and mortality risk. When the equity return increases, the hedger has to acquire more swaptions for hedging.

The payoff structure of the GAO resembles a quanto call option written on a coupon-bearing bond. The "quanto" feature appears since the payoff is in units of "stock" (like units of foreign currency) rather than in cash. The moneyness of the option is entirely dependent on the interest rate risk. Ballotta and Haberman (2003a) apply the one-factor Ho-Lee model and Vasicek model to price GAO in unit-linked deferred annuity contracts that are purchased on the grant date by a single premium [with later extension to include stochastic mortality effects (Ballotta and Haberman (2003b))]. However, an one-factor interest rate model would implicitly imply that all future interest rates are perfectly correlated. Since pension policies are long term contracts, it is generally known in the literature that a two-factor interest rate model performs much better in hedging long-term interest rate derivatives.

In this paper, we employ a two-factor interest rate model of the affine class (Dai and Singleton, 2000) to characterize the stochastic interest rate movement. Similar to Ballotta-Haberman framework (2003a), we do not incorporate the insurance company expenses, tax effects and pre-retirement death benefits into our model. Also, the mortality risk is assumed to be unsystematic and independent of the equity and interest rate risks. Under the simplicity of the one-factor interest rate model, Ballotta and Haberman (2003a) are able to apply the decomposition technique of Jamshidian (1989) on pricing options on coupon-bearing bonds, thanks to the observation that the annuity option payoff can be written as the payoff generated by a portfolio of zero-coupon bond options with appropriate strike prices. However, since the interest rates become correlated under the two-factor interest rate model, the Jamshidian decomposition technique cannot be applied.

There will be no closed form analytic price formula for the GAO when the interest rate dynamics is modeled by a two-factor interest rate model. However, several analytic approximation methods are known in the literature for pricing bond options or swaptions under multi-factor affine interest rate models. One method uses a single zero-coupon bond as a proxy for the original coupon-bearing bond. The approximation error is minimized by choosing the maturity of the zero-coupon bond equal to the stochastic duration (Cox *et al.*, 1979; Wei, 1997) of the coupon-bearing bond. Another method makes use of the Edgeworth approximation of the probability distribution of the value of the coupon-bearing bond (Collin-Dufresne and Goldstein, 2002). The third method approximates the conditional distributions of the risk factors in affine diffusions. The exercise probability of the annuity option is approximated through an approximation of the exercise region. This is achieved by the linearization of the exercise region, whose boundary is approximated by a hyperplane. One then compute the relevant probabilities needed for pricing options on coupon-bearing bonds by the same numerical method used in the pricing of options on zero-coupon bonds (Singleton and Umantsev, 2002). We adopt and modify these analytic approximation methods for numerical valuation of the GAO in deferred annuity pension policies. The numerical accuracy and computational efficiency of these approximation schemes are compared, and the impact of different parameter values in the pricing model on the GAO value are investigated.

The paper is organized as follows. In the next section, we present the model setup of the GAO and the formulation of the multi-factor affine interest rate model. Nice analytic tractability of the affine term structure model for finding present values of annuity payments are demonstrated. In Section 3, we discuss the method of minimum variance duration. The GAO is priced under the measure associated with the numeraire that is related to the annuity payment paid at $\tau$-period after retirement. A judicious analytic approximation is made in the expectation calculations so that closed form formula can be obtained. The pricing error between the exact and approximate solutions is minimized by choosing $\tau$ such that the variance of the value of the future stream of annuity payments normalized by the price of the $\tau$-maturity bond is minimized. In Section 4, we illustrate how to perform the Edgeworth expansion of the distribution of the annuity value at the maturity of the policy. Under the affine diffusion assumption, the bond prices are exponential affine functions of the risk factors. The moments of the annuity value are also exponential affine so that the coefficients can be solved through the solution of a system of Ricatti equations. In Section 5, we apply the affine approximation approach to the valuation of GAO. Section 6 reports the numerical experiments that were performed to compare the numerical accuracy and computational efficiency of the minimum variance duration approach, Edgeworth series approximation and affine approximation. Pricing behaviors of the GAO are also examined. The last section summarizes and concludes the main results of the paper.

## 2. Model setup of the guaranteed annuity option

The payoff structure of a guaranteed annuity option (GAO) is similar to a call option on a coupon bond, where the "coupons" are the future stream of annuity payments and the "maturity of the bond" is related to the mortality of the policyholder. Besides the interest rate risk as in usual options on a coupon bond, the GAO also has exposure in equity risk and mortality risk. The equity risk arises since the payoff is in units of stock rather than in cash so that the payoff is essentially in the form of a quanto option (the equity risk in GAO resembles the exchange rate risk in quanto option). For the mortality risk, we assume that it is independent of the financial risk so that it is diversifiable. It is quite acceptable to use deterministic mortality for valuing options dependent on death of the policyholder (Boyle and Hardy, 2003).

We consider a single premium equity-linked policy whose policy's maturity date is $T$. The maturity date $T$ coincides with the retirement age $R$ of the policyholder. The premium is invested in equity whose value $S_t$ is assumed to follow a Geometric Brownian process. We let $a_R(t)$ denote the market value at time $t$ of a life annuity of one dollar per annum starting at age $R$. Let ${}_np_R$ denote the probability that a person aged $R$ survives $n$ years and $D_{T+n}(t)$ denote the market value of the unit par default free zero-coupon bond at time $t$ with maturity date $T+n$. Also, we let $\omega$ denote the maximum age in the mortality table. By constructing a portfolio of default free bonds that match exactly with the expected cash flows of the annuity, the value of annuity $a_R(T)$ is given by

$$a_R(T) = \sum_{n=0}^{\omega-R-1} {}_np_R D_{T+n}(T) = 1 + \sum_{n=1}^{\omega-R-1} {}_np_R D_{T+n}(T), \tag{2.1}$$

since ${}_0p_R = D_T(T) = 1$. Provided that the policyholder survives to maturity $T$, he either receives $S_T$ or $\frac{S_T}{g}a_R(T)$ at $T$, whichever has a high value. Here, $g$ is called the guaranteed conversion rate (say, $g = 9$). When the policyholder exercises the GAO, the equity fund $S_T$ is used to purchase an annuity of $S_T/g$. The value of the GAO at maturity $T$ is then given by

$$\text{terminal value of GAO } = \frac{S_T}{g}(a_R(T) - g)^+, \tag{2.2}$$

where $x^+ = \max(x, 0)$. By assuming deterministic mortality rates, the payoff of the form $(a_R(T) - g)^+$ resembles an option on a coupon bond with strike $g$ and coupon payment of amount ${}_np_R$ at time $T + n, n = 0, 1, \cdots$. The factor $S_T/g$ behaves like the exchange rate factor in a quanto option. The GAO has two types of financial risk exposure: interest rate risk and equity risk.

When the interest rate dynamics is modeled by an one-factor short rate model, it is relatively straightforward to obtain closed form formula for the GAO using the Jamshidian decomposition technique for coupon bearing bond (Boyle and Hardy, 2003; Ballotta and Haberman, 2003a). Unfortunately, the one-factor assumption of the short rate would imply full correlation of all future interest rates. Such feature invites criticism when the one-factor short rate model is employed to price long term interest rate derivatives. In this paper, we use the multi-factor affine term structure framework to model the interest rate derivatives. The affine framework has become more popular due to its analytic tractability and flexibility. Also, the multi-factor affine model can be easily calibrated through fitting of the current term structure of traded bond prices.

### Multi-factor affine term structure model

Let $r_t$ denote the short rate. The risk neutral processes of $r_t$ and the $\ell$-component vector of risk factors $\boldsymbol{x}(t)$ are governed by

$$\begin{aligned} r_t &= \boldsymbol{a}(t)^T\boldsymbol{x}(t) + \boldsymbol{b}(t) \\ d\boldsymbol{x}(t) &= \boldsymbol{\mu}(\boldsymbol{x},t)\,dt + \sigma(\boldsymbol{x},t)\,d\boldsymbol{Z}(t), \end{aligned} \tag{2.3}$$

where the parameter function

$$\boldsymbol{a}(t) = \begin{pmatrix} a_1(t) \\ a_2(t) \\ \vdots \\ a_\ell(t) \end{pmatrix}$$

is a deterministic $\ell$-component vector function, $b(t)$ is a scalar function and

$$\boldsymbol{\mu}(\boldsymbol{x},t) = \begin{pmatrix} \mu_1(\boldsymbol{x},t) \\ \mu_2(\boldsymbol{x},t) \\ \vdots \\ \mu_\ell(\boldsymbol{x},t) \end{pmatrix} \quad \text{and} \quad \sigma(\boldsymbol{x},t) = \begin{pmatrix} \sigma_{11}(\boldsymbol{x},t) & \cdots & \sigma_{1m}(\boldsymbol{x},t) \\ \vdots & & \vdots \\ \sigma_{\ell 1}(\boldsymbol{x},t) & \cdots & \sigma_{\ell m}(\boldsymbol{x},t) \end{pmatrix}$$

are the drift rate vector and volatility matrix for $\boldsymbol{x}(t)$. Also, the $m$ components in the random vector

$$\boldsymbol{Z}(t) = \begin{pmatrix} Z_1(t) \\ Z_2(t) \\ \vdots \\ Z_m(t) \end{pmatrix}$$

are independent Wiener processes under the risk neutral measure $Q$. Under certain conditions on $\boldsymbol{\mu}$ and $\sigma$, the time-$t$ value of the zero-coupon bond maturing at time $T$ has the following exponential affine form (Dai and Singleton, 2000)

$$D_T(t) = \exp(-\boldsymbol{A}_T(t)^T\boldsymbol{x}(t) - B_T(t)), \tag{2.4$a$}$$

where $\boldsymbol{A}_T(t)$ and $B_T(t)$ are governed by a system of Ricatti differential equations. To be more precise on the functional dependence, $D_T(t)$ is a function of $\boldsymbol{x}, t$ and time to maturity $T-t$, $\boldsymbol{A}_T(t)$ is a function of $T-t$ while $B_T(t)$ is a function of both $t$ and $T-t$. The volatility vector of the bond price is given by

$$\boldsymbol{\sigma}_D(\boldsymbol{x},t;T) = \begin{pmatrix} \sigma_{D,1}(\boldsymbol{x},t;T) \\ \sigma_{D,2}(\boldsymbol{x},t;T) \\ \vdots \\ \sigma_{D,m}(\boldsymbol{x},t;T) \end{pmatrix} = -\sigma(\boldsymbol{x},t)^T\boldsymbol{A}_T(t). \tag{2.4$b$}$$

For the equity fund, its time-$t$ value under the risk neutral measure $Q$ is modeled by

$$\frac{dS_t}{S_t} = (r - q)\,dt + \boldsymbol{\sigma}_S(t)^T\,d\boldsymbol{Z}, \tag{2.5}$$

where $q$ is the constant dividend yield and $\boldsymbol{\sigma}_S(t)^T = (\sigma_{S,1}(t)\ \sigma_{S,2}(t)\cdots\sigma_{S,m}(t))$ is the vector of equity volatilities.

**Risk neutral valuation of GAO value**

Under the risk neutral measure $Q$, the time-$t$ value $V(S, \boldsymbol{x}, t)$ of the GAO is given by the risk neutral expectation of the payoff at time $T$ times the probability of survival of the policyholder over the next $T-t$ years. The probability of survival is given by ${}_{T-t}p_{R-(T-t)}$ since the policyholder reaches age $R$ in $T-t$ years later. We assume that the company has well diversified the sale of annuity products so that the mortality risk can be taken to be independent of the financial risk under the risk neutral measure $Q$. Given the terminal payoff defined in Eq. (2.2), we then have

$$\begin{aligned} V(S, \boldsymbol{x}, t) &= {}_{T-t}p_{R-(T-t)} E_Q\left[e^{-\int_t^T r_u\,du}\frac{S_T}{g}(a_R(T) - g)^+\right] \\ &= {}_{T-t}p_{R-(T-t)}\left\{E_Q\left[e^{-\int_t^T r_u\,du}\frac{S_T}{g}\sum_{n=1}^{\omega-R-1}{}_np_R D_{T+n}(t)\mathbf{1}_{\{a_R(T)>g\}}\right]\right. \\ &\quad \left.-\left(1-\frac{1}{g}\right)E_Q\left[e^{-\int_t^T r_u\,du}S_T\mathbf{1}_{\{a_R(T)>g\}}\right]\right\}. \end{aligned} \tag{2.6}$$

It will be illustrated that the above expectation calculations can be simplified by the method of change of numeraire.

Let $F_n(S, \boldsymbol{x}, t)$ denote the time-$t$ value of a security that pays $S_T D_{T+n}(T)$ at time $T$ so that $F_n(S, \boldsymbol{x}, t)/g$ gives the time-$t$ value of the annuity payment at time $T+n$. By using the equity fund value $S_t$ as the numeraire, we obtain

$$\begin{aligned} F_n(S, \boldsymbol{x}, t) &= E_Q\left[e^{-\int_t^T r_u\,du}S_T D_{T+n}(T)\right] \\ &= S_t e^{-q(T-t)}E_{Q_S}[D_{T+n}(T)], \end{aligned} \tag{2.7}$$

where $Q_S$ is the measure associated with the numeraire $S_t$. In a similar manner, the expectation of the second term in Eq. (2.6) can be expressed as

$$\begin{aligned} E_Q\left[e^{-\int_t^T r_u\,du}S_T\mathbf{1}_{\{a_R(T)>g\}}\right] &= S_t e^{-q(T-t)}E_{Q_S}\left[\mathbf{1}_{\{a_R(T)>g\}}\right] \\ &= S_t e^{-q(T-t)}P_{Q_S}[a_R(T) > g], \end{aligned} \tag{2.8}$$

where $P_{Q_S}[A]$ denotes the probability of event $A$ occurring under the measure $Q_S$. To compute the expectation of the first term in Eq. (2.6), it is more appropriate to choose $F_n(S, \boldsymbol{x}, t), n = 1, 2, \cdots, \omega - R$, as the numeraire. Let $Q_{F_n}$ denote the measure associated with the numeraire $F_n$. We then have

$$\begin{aligned}
& E_Q\left[e^{-\int_t^T r_u\,du}\frac{S_T}{g}\sum_{n=1}^{\omega-R-1} {}_np_R D_{T+n}(T)\mathbf{1}_{\{a_R(T)>g\}}\right] \\
= & \sum_{n=1}^{\omega-R-1}\frac{{}_np_R}{g}F_n(S, \boldsymbol{x}, t)P_{Q_{F_n}}[a_R(T) > g]. \qquad (2.9)
\end{aligned}$$

In our subsequent discussion, we limit the multi-factor affine term structure model to the Gaussian type model, where the volatility matrix $\sigma(\boldsymbol{x}, t)$ defined in Eq. (2.3) is a function of $t$ only. For a Gaussian type model, the bond price volatility vector becomes

$$\boldsymbol{\sigma}_D(t; T) = -\sigma(t)^T\boldsymbol{A}_T(t).$$

**Stochastic differential equations**

Under the risk neutral measure $Q$, the stochastic differential equation (SDE) of $D_T(t)$ is given by

$$\frac{dD_T(t)}{D_T(t)} = r_t\,dt + \boldsymbol{\sigma}_D(t; T)^T\,d\boldsymbol{Z}.$$

Using the Girsanov Theorem, the SDE of $D_T(t)$ under the measure $Q_S$ is given by

$$\frac{dD_T(t)}{D_T(t)} = [r_t + \boldsymbol{\sigma}_D(t; T)^T\boldsymbol{\sigma}_S(t)]\,dt + \boldsymbol{\sigma}_D(t; T)^T\,d\boldsymbol{Z}_{Q_S}, \qquad (2.10)$$

where $\boldsymbol{Z}_{Q_S}$ is a vector of Brownian processes under $Q_S$. The SDE of $F_n(S, \boldsymbol{x}, t)$ under the risk neutral measure $Q$ is given by

$$\begin{aligned}
\frac{dF_n}{F_n} &= r_t\,dt + [\boldsymbol{\sigma}_S(t) + \boldsymbol{\sigma}_D(t; T+n) - \boldsymbol{\sigma}_D(t; T)]^T\;d\boldsymbol{Z} \\
&= r_t\,dt + \left\{\boldsymbol{\sigma}_S(t) + \sigma(t)^T[\boldsymbol{A}_T(t) - \boldsymbol{A}_{T+n}(t)]\right\}^T\;d\boldsymbol{Z}. \qquad (2.11)
\end{aligned}$$

Next, we would like to solve for $F_n(S, \boldsymbol{x}, t)$. We consider $\ln\frac{D_{T+n}(t)}{D_T(t)}$, whose dynamics under $Q_S$ is given by

$$d\left(\ln\frac{D_{T+n}(t)}{D_T(t)}\right)$$

$$
\begin{aligned}
&= \left\{\boldsymbol{\sigma}_S(t)^T[\boldsymbol{\sigma}_D(t;T+n)-\boldsymbol{\sigma}_D(t,T)] - \frac{1}{2}\left(\|\boldsymbol{\sigma}_D(t;T+n)\|^2 - \|\boldsymbol{\sigma}_D(t;T)\|^2\right)\right\} dt \\
&\quad + [\boldsymbol{\sigma}_D(t;T+n)-\boldsymbol{\sigma}_D(t;T)]^T \, d\boldsymbol{Z}_{Q_S} \\
&= \left\{[\boldsymbol{\sigma}_S(t)-\boldsymbol{\sigma}_D(t;T)]^T[\boldsymbol{\sigma}_D(t;T+n)-\boldsymbol{\sigma}_D(t;T)] - \frac{1}{2}\|\boldsymbol{\sigma}_D(t;T+n)-\boldsymbol{\sigma}_D(t;T)\|^2\right\} dt \\
&\quad + [\boldsymbol{\sigma}_D(t;T+n)-\boldsymbol{\sigma}_D(t;T)]^T \, d\boldsymbol{Z}_{Q_S}. \qquad (2.12)
\end{aligned}
$$

The solution to $F_n(S,\boldsymbol{x},t)$ is readily found to be

$$
\begin{aligned}
F_n(S,\boldsymbol{x},t) = {} & \frac{D_{T+n}(t)}{D_T(t)} S_t e^{-q(T-t)} \\
& \exp\left(\int_t^T [\boldsymbol{\sigma}_S(u)-\boldsymbol{\sigma}_D(u;T)]^T[\boldsymbol{\sigma}_D(u;T+n)-\boldsymbol{\sigma}_D(u;T)]\,du\right). \qquad (2.13)
\end{aligned}
$$

Lastly, the SDE of $\boldsymbol{x}$ under $Q_{F_n}$ can be deduced to be

$$
\begin{aligned}
d\boldsymbol{x} &= \{\boldsymbol{\mu}(\boldsymbol{x},t) + \sigma(t)[\boldsymbol{\sigma}_S(t)+\boldsymbol{\sigma}_D(t;T+n)-\boldsymbol{\sigma}_D(t;T)]\}\,dt + \sigma(t)\,d\boldsymbol{Z}_{Q_{F_n}} \\
&= \left\{\boldsymbol{\mu}(\boldsymbol{x},t) + \sigma(t)\boldsymbol{\sigma}_S(t) + \sigma(t)\sigma(t)^T[\boldsymbol{A}_T(t)-\boldsymbol{A}_{T+n}(t)]\right\} dt + \sigma(t)\,d\boldsymbol{Z}_{Q_{F_n}}, \qquad (2.14)
\end{aligned}
$$

where $\boldsymbol{Z}_{Q_{F_n}}$ is a vector of Brownian processes under the measure $Q_{F_n}$.

Under the Gaussian type affine term structure model, the bond prices are lognormally distributed [see Eq. (2.4a,b)]. Since the future annuity payment stream can be visualized as a portfolio of discount bonds and the density of the sum of lognormal distributions has no closed form representation, so there is no closed form analytic solution to the GAO value under the multi-factor affine term structure model. In the next three sections, we explore three different analytic methods for finding approximate solution to $V(S,\boldsymbol{x},t)$.

## 3. Method of minimum variance duration

We adopt the idea of minimum variance duration similar to that proposed by Munk (1999). The minimum variance duration approach has been shown to give highly accurate approximation solution to an option on coupon bearing bond under the multi-factor interest rate model. The minimum variance duration may be considered as an extension of the concept of stochastic duration. Recall that the stochastic duration of a coupon bearing bond in a multi-factor diffusion model is defined to be the time to maturity of the zero-coupon bond with the same relative volatility as that of the coupon bearing bond (Wei, 1997).

The solution of the GAO value may be sought by pricing under the measure associated with the numeraire corresponding to the security that pays $S_T a_R(T)$ at $T$. However, the pricing under such measure is not analytically tractable. Instead, we consider an alternative numeraire that corresponds to the security that pays $S_T D_{T+\tau}(T)$ at maturity time $T$. Here, $\tau$ represents the time to maturity of the underlying bond at time $T$. Later, we illustrate how to choose the parameter $\tau$ such that the error in the approximate solution is minimized in some sense. Let $F_\tau(S, \boldsymbol{x}, t)$ denote the time-$t$ value of such security and $Q_{F_\tau}$ denote the pricing measure when $F_\tau(S, \boldsymbol{x}, t)$ is used as the numeraire. Under $Q_{F_\tau}$, the time-$t$ value of the GAO is given by [see Eqs. (2.6) and (2.9)]

$$\begin{aligned} V(S, \boldsymbol{x}, t) &= {}_{T-t}p_{R-(T-t)} E_Q \left[ e^{-\int_t^T r_u \, du} S_T \left( \frac{a_R(T)}{g} - 1 \right)^+ \right] \\ &= {}_{T-t}p_{R-(T-t)} F_\tau(S, \boldsymbol{x}, t) E_{Q_{F_\tau}} \left[ \left( \frac{a_R(T)}{g D_{T+\tau}(T)} - \frac{1}{D_{T+\tau}(T)} \right)^+ \right]. \end{aligned} \tag{3.1}$$

Nice analytic tractability can be achieved if we set $\dfrac{a_R(T)}{g D_{T+\tau}(T)}$ be some constant $K$. Here, $K$ is judiciously chosen to be the mean of $\dfrac{a_R(T)}{g D_{T+\tau}(T)}$ under $Q_{F_\tau}$. The analytic approximate solution to $V(S, \boldsymbol{x}, t)$ is taken to be

$$V_a(S, \boldsymbol{x}, t) = {}_{T-t}p_{R-(T-t)} F_\tau(S, \boldsymbol{x}, t) E_{Q_{F_\tau}} \left[ \left( K - \frac{1}{D_{T+\tau}(T)} \right)^+ \right], \tag{3.2$a$}$$

where

$$K = E_{Q_{F_\tau}} \left[ \frac{a_R(T)}{g D_{T+\tau}(T)} \right]. \tag{3.2$b$}$$

The remaining procedures include (i) the derivation of closed form analytic expression for $V_a(S, \boldsymbol{x}, t)$, (ii) the determination of the parameter $\tau$ such that the pricing error $|V(S, \boldsymbol{x}, t) - V_a(S, \boldsymbol{x}, t)|$ is minimized based on the minimization of variance.

**Approximate price formula**

First, $K$ can be readily found to be

$$\begin{aligned} K &= \frac{1}{g} \sum_{n=0}^{\omega-R-1} {}_n p_R E_{Q_{F_\tau}} \left[ \frac{D_{T+n}(T)}{D_{T+\tau}(T)} \right] \\ &= \frac{1}{g} \frac{S_t e^{-q(T-t)}}{F_\tau(S, \boldsymbol{x}, t)} \sum_{n=0}^{\omega-R-1} {}_n p_R E_{Q_S}[D_{T+n}(T)] \\ &= \frac{1}{g F_\tau(S, \boldsymbol{x}, t)} \sum_{n=0}^{\omega-R-1} {}_n p_R F_n(S, \boldsymbol{x}, t). \end{aligned} \tag{3.3}$$

Next, the expectation in Eq. (3.2a) is found to be

$$\begin{aligned} & E_{Q_{F_\tau}} \left[ \left( K - \frac{1}{D_{T+\tau}(T)} \right)^+ \right] \\ &= K P_{Q_{F_\tau}} \left[ D_{T+\tau}(T) > \frac{1}{K} \right] - \frac{S_t e^{-q(T-t)}}{F_\tau(S, \boldsymbol{x}, t)} P_{Q_S} \left[ D_{T+\tau}(T) > \frac{1}{K} \right]. \end{aligned} \tag{3.4}$$

By combining Eqs. (3.3) and (3.4) together, we obtain

$$
\begin{aligned}
V_a(S,\boldsymbol{x},t) = {}_{T-t}p_{R-(T-t)} \Bigg\{ & \sum_{n=0}^{\omega-R-1} \frac{{}_n p_R F_n(S,\boldsymbol{x},t)}{g} P_{Q_{F_\tau}} \left[ D_{T+\tau}(T) > \frac{1}{K} \right] \\
& - S_t e^{-q(T-t)} P_{Q_S} \left[ D_{T+\tau}(T) > \frac{1}{K} \right] \Bigg\}. \qquad (3.5)
\end{aligned}
$$

Similar to Eq. (2.11), the dynamics of $\ln \dfrac{D_{T+\tau}(t)}{D_T(t)}$ under $Q_{F_\tau}$ is found to be

$$
\begin{aligned}
d\left(\ln \frac{D_{T+\tau}(t)}{D_T(t)}\right) = & \Big\{ [\boldsymbol{\sigma}_S(t) - \boldsymbol{\sigma}_D(t;T)]^T [\boldsymbol{\sigma}_D(t;T+\tau) - \boldsymbol{\sigma}_D(t;T)] \\
& + \frac{1}{2} \|\boldsymbol{\sigma}_D(t;T+\tau) - \boldsymbol{\sigma}_D(t;T)\|^2 \Big\} \, dt \\
& + [\boldsymbol{\sigma}_D(t;T+\tau) - \boldsymbol{\sigma}_D(t;T)]^T \, d\boldsymbol{Z}_{Q_{F_\tau}}. \qquad (3.6)
\end{aligned}
$$

The mean of $\ln D_{T+\tau}(T)$ under $Q_{F_\tau}$ and $Q_S$ are obtained as follows:

$$
E_{Q_{F_\tau}}[\ln D_{T+\tau}(T)] = \overline{c}(\tau) + \frac{\overline{v}^2(\tau)}{2} + \ln \left[ \frac{D_{T+\tau}(t)}{D_T(t)} \right] \qquad (3.7a)
$$

$$
E_{Q_S}[\ln D_{T+\tau}(T)] = \overline{c}(\tau) - \frac{\overline{v}^2(\tau)}{2} + \ln \left[ \frac{D_{T+\tau}(t)}{D_T(t)} \right], \qquad (3.7b)
$$

where

$$
\overline{c}(\tau) = \int_t^T [\boldsymbol{\sigma}_S(u) - \boldsymbol{\sigma}_D(u;T)]^T [\boldsymbol{\sigma}_D(u;T+\tau) - \boldsymbol{\sigma}_D(u;T)] \, du
$$

$$
\overline{v}^2(\tau) = \mathrm{var}\,[\ln D_{T+\tau}(T)] = \int_t^T \|\boldsymbol{\sigma}_D(u;T+\tau) - \boldsymbol{\sigma}_D(u;T)\|^2 \, du.
$$

Also, we may express $F_n(S,\boldsymbol{x},t)$ and $K$ in the following forms:

$$
F_n(S,\boldsymbol{x},t) = \frac{D_{T+n}(t) S_t}{D_T(t)} e^{-q(T-t)+\overline{c}(n)}
$$

$$
K = \frac{e^{-\overline{c}(\tau)}}{g D_{T+\tau}(t)} \sum_{n=0}^{\omega-R-1} {}_n p_R D_{T+n}(t) e^{\overline{c}(n)} = \frac{e^{-\overline{c}(\tau)}}{g D_{T+\tau}(t)} \overline{a}_R(t).
$$

Here, the quantity

$$
\overline{a}_R(t) = \sum_{n=0}^{\omega-R-1} {}_n p_R D_{T+n}(t) e^{\overline{c}(n)}
$$

can be interpreted as the equity-risk-adjusted annuity. Now, it becomes readily to compute the two probability values in Eq. (3.4) and obtain

$$P_{Q_{F_\tau}}\left[D_{T+\tau}(T) > \frac{1}{K}\right] = N\left(-\frac{\ln\frac{1}{K} - \ln\frac{D_{T+\tau}(t)}{D_T(t)} - \overline{c}(\tau) - \frac{\overline{v}^2(\tau)}{2}}{\overline{v}(\tau)}\right) = N(d)$$

and

$$P_{Q_S}\left[D_{T+\tau}(T) > \frac{1}{K}\right] = N(d - \overline{v}(\tau)),$$

where

$$d = \frac{\ln\dfrac{\overline{a}_R(t)}{gD_T(t)} + \dfrac{\overline{v}^2(\tau)}{2}}{\overline{v}(\tau)}.$$

Finally, the analytic expression for $V_a(S, \boldsymbol{x}, t)$ is found to be

$$V_a(S, \boldsymbol{x}, t) = {}_{T-t}p_{R-(T-t)} S e^{-q(T-t)}\left[\frac{\overline{a}_R(t)}{gD_T(t)} N(d) - N(d - \overline{v}(\tau))\right]. \qquad (3.8)$$

**Determination of $\tau$ using minimization of variance duration**

The error in the approximation of $V(S, \boldsymbol{x}, t)$ by $V_a(S, \boldsymbol{x}, t)$ is quantified by $E_{Q_{F_\tau}}[|Y|]$, where

$$Y = \left(\frac{a_R(T)}{gD_{T+\tau}(T)} - \frac{1}{D_{T+\tau}(T)}\right)^+ - \left(K - \frac{1}{D_{T+\tau}(T)}\right)^+. \qquad (3.9)$$

Following a similar approach as proposed by Munk (2000), the pricing error is minimized by choosing $\tau$ so as to minimize the variance of $\dfrac{da_R(t)}{a_R(t)} - \dfrac{dD_{T+\tau}(t)}{D_{T+\tau}(t)}$. That is, the optimal value of $\tau$ is given by

$$\tau^* = \underset{\tau \geq 0}{\text{argmin}}\left\|\text{var}_{Q_{F_\tau}}\left(\frac{da_R(t)}{a_R(t)} - \frac{dD_{T+\tau}(t)}{D_{T+\tau}(t)}\right)\right\|. \qquad (3.10)$$

We present the justification of the above argument, then followed by the derivation of the analytic procedures to obtain $\tau^*$.

Let $\widehat{m} = \min\left(\dfrac{a_R(T)}{gD_{T+\tau}(T)}, K\right)$ and $\widehat{M} = \max\left(\dfrac{a_R(T)}{gD_{T+\tau}(T)}, K\right)$. The following three events are mutually exclusive and exhaustive:

$$E_1 = \left\{\frac{1}{D_{T+\tau}(T)} \geq \widehat{M}\right\}, \quad E_2 = \left\{\widehat{m} < \frac{1}{D_{T+\tau}(T)} < \widehat{M}\right\} \quad \text{and} \quad E_3 = \left\{\frac{1}{D_{T+\tau}(T)} \leq \widehat{m}\right\},$$

and from which we deduce that

$$E_{Q_{F_\tau}}\left[|Y|\right] = E_{Q_{F_\tau}}\left[|Y|\mathbf{1}_{E_1}\right] + E_{Q_{F_\tau}}\left[|Y|\mathbf{1}_{E_2}\right] + E_{Q_{F_\tau}}\left[|Y|\mathbf{1}_{E_3}\right].$$

Each of the above expectation calculations is analyzed below.

(i) $E_{Q_{F_\tau}}\left[|Y|\mathbf{1}_{E_1}\right] = 0$ since $Y$ becomes zero when $E_1$ occurs.

(ii) $E_{Q_{F_\tau}}\left[|Y|\mathbf{1}_{E_2}\right] \leq \{E_{Q_{F_\tau}}\left[|Y|\right] P_{Q_{F_\tau}}[E_2]\}^{1/2}$ and $P_{Q_{F_\tau}}[E_2]$ has a smaller value when $\dfrac{a_R(T)}{gD_{T+\tau}(T)}$ stays closer to its mean $K$. This occurs when $\mathrm{var}_{\,Q_{F_\tau}}\left(\dfrac{a_R(T)}{gD_{T+\tau}(T)}\right)$ is minimized.

(iii)
$$\begin{aligned} E_{Q_{F_\tau}}\left[|Y|\mathbf{1}_{E_3}\right] &= E_{Q_{F_\tau}}\left[\left|\frac{a_R(T)}{gD_{T+\tau}(T)} - K\right|\mathbf{1}_{E_3}\right] \\ &\leq \left\{E_{Q_{F_\tau}}\left[\left(\frac{a_R(T)}{gD_{T+\tau}(T)} - K\right)^2\right] P_{Q_{F_\tau}}[E_3]\right\}^{1/2} \\ &= \left\{\mathrm{var}_{\,Q_{F_\tau}}\left(\frac{a_R(T)}{gD_{T+\tau}(T)}\right) P_{Q_{F_\tau}}[E_3]\right\}^{1/2}. \end{aligned}$$

Therefore, one can minimize the pricing error by minimizing $\mathrm{var}_{\,Q_{F_\tau}}\left(\dfrac{a_R(T)}{gD_{T+\tau}(T)}\right)$ over the choice of $\tau$. However, the minimization procedure appears to be intractable due to the complex expressions for $a_R(T)$ and $D_{T+\tau}(T)$. Instead, we attempt to minimize the relative change of value in $\dfrac{a_R(T)}{D_{T+\tau}(T)}$, which can be measured by the variance of $\dfrac{da_R(t)}{a_R(t)} - \dfrac{dD_{T+\tau}(t)}{D_{T+\tau}(t)}$.

Under the risk neutral measure $Q$, the dynamics of $a_R(t)$ and $D_{T+\tau}(t)$ are given by

$$\begin{aligned} \frac{da_R(t)}{a_R(t)} &= r_t\,dt + \boldsymbol{\sigma}_a(t;T)^T\,d\boldsymbol{Z} \\ \frac{dD_{T+\tau}(t)}{D_{T+\tau}(t)} &= r_t\,dt + \boldsymbol{\sigma}_D(t;T+\tau)^T\,d\boldsymbol{Z}, \end{aligned}$$

where the volatility vector of annuity $\boldsymbol{\sigma}_a$ is given by

$$\boldsymbol{\sigma}_a(t;T) = \sum_{n=0}^{\omega-R-1} \frac{{}_np_R D_{T+n}(t)}{a_R(t)} \boldsymbol{\sigma}_D(t;T+n).$$

For an one-factor interest rate model, it is readily seen that the solution to $\tau^*$ defined in Eq. (3.10) is given by

$$\boldsymbol{\sigma}_a(t;T) = \boldsymbol{\sigma}_D(t;T+\tau^*), \tag{3.11}$$

which is just the stochastic duration of the annuity (Wei, 1997). For the general multi-factor case, the minimization of $\operatorname{var}_{Q_{F_\tau}}\left(\frac{da_R(t)}{a_R(t)} - \frac{dD_{T+\tau}(t)}{D_{T+\tau}(t)}\right)$ leads to the following non-linear algebraic equation for $\tau$:

$$[\boldsymbol{\sigma}_a(t;T) - \boldsymbol{\sigma}_D(t;T+\tau)]^T \frac{\partial \boldsymbol{\sigma}_D(t;T+\tau)}{\partial \tau} = 0. \tag{3.12}$$

### Two-factor Gaussian model

We illustrate how to compute $V_a(S, \boldsymbol{x}, t)$ using the two-factor Gaussian interest rate model ($G2$++) as an example. For the $G2$++ model, the interest rate $r_t$ is given by

$$r_t = x_{1,t} + x_{2,t} + b(t) \tag{3.13$a$}$$

where the dynamics of the risk factors are governed by

$$\begin{aligned} dx_1 &= -\kappa_1 x_1\, dt + \sigma_1\, dZ_1 \\ dx_2 &= -\kappa_2 x_2\, dt + \sigma_2(\rho\, dZ_1 + \sqrt{1-\rho^2}\, dZ_2). \end{aligned} \tag{3.13$b$}$$

Here, $b(t)$ is a function which is determined by fitting the current interest rate term structure and $\rho$ is the correlation coefficient between the risk factors. For Gaussian type models, one of the drawbacks is the possibility of negative interest rates. For the $G2$++ model, the corresponding solution of $\boldsymbol{A}_T(t)$ and $B_T(t)$ as defined in Eq. (2.4a) are found to be

$$\boldsymbol{A}_T(t) = \begin{pmatrix} \frac{1-e^{-\kappa_1(T-t)}}{\kappa_1} \\ \frac{1-e^{-\kappa_2(T-t)}}{\kappa_2} \end{pmatrix}, \tag{3.14$a$}$$

and

$$\begin{aligned} B_T(t) = &-\ln\frac{D_T(0)}{D_t(0)} - \frac{\sigma_1^2}{\kappa_1}\left(\frac{1-e^{-2\kappa_1 t}}{2\kappa_1}\right)\left[\frac{1-e^{-2\kappa_1(T-t)}}{2\kappa_1}\right] \\ &- \frac{\sigma_2^2}{\kappa_2}\left(\frac{1-e^{-2\kappa_2 t}}{2\kappa_2}\right)\left[\frac{1-e^{-2\kappa_2(T-t)}}{2\kappa_2}\right] \\ &- \rho\sigma_1\sigma_2\left(\frac{1}{\kappa_1}+\frac{1}{\kappa_2}\right)\left(\frac{1-e^{-(\kappa_1+\kappa_2)t}}{\kappa_1+\kappa_2}\right)\left[\frac{1-e^{-(\kappa_1+\kappa_2)(T-t)}}{\kappa_1+\kappa_2}\right] \\ &+ \left(\frac{\sigma_1^2}{\kappa_1}+\frac{\rho\sigma_1\sigma_2}{\kappa_2}\right)\left(\frac{1-e^{-\kappa_1 t}}{\kappa_1}\right)\left[\frac{1-e^{-\kappa_1(T-t)}}{\kappa_1}\right] \\ &+ \left(\frac{\rho\sigma_1\sigma_2}{\kappa_1}+\frac{\sigma_2^2}{\kappa_2}\right)\left(\frac{1-e^{-\kappa_2 t}}{\kappa_2}\right)\left[\frac{1-e^{-\kappa_2(T-t)}}{\kappa_2}\right]. \end{aligned} \tag{3.14$b$}$$

Once $\boldsymbol{A}_T(t)$ and $B_T(t)$ are known, the bond prices $D_{T+n}(t)$ and $\overline{a}_R(t)$ can be determined. It remains to find $\overline{c}(\tau)$ and $\overline{v}^2(\tau)$ for the G2++ model. The corresponding volatility vector $\boldsymbol{\sigma}_D(t;T)$ is given by

$$\boldsymbol{\sigma}_D(t;T) = -\begin{pmatrix} \frac{\sigma_1}{\kappa_1}[1-e^{-\kappa_1(T-t)}] + \frac{\rho\sigma_2}{\kappa_2}[1-e^{-\kappa_2(T-t)}] \\ \frac{\sigma_2\sqrt{1-\rho^2}}{\kappa_2}[1-e^{-\kappa_2(T-t)}] \\ 0 \end{pmatrix}. \tag{3.14$c$}$$

Suppose the stochastic component of the equity fund value $S_t$ under the risk neutral measure is $\sigma_S\,dZ_S$, where $dZ_S\,dZ_1 = \rho_{S1}\,dt$ and $dZ_S\,dZ_2 = \rho_{S2}\,dt$, then

$$\frac{dS_t}{S_t} = (r-q)\,dt + \boldsymbol{\sigma}_S^T\,d\boldsymbol{Z}, \tag{3.15$a$}$$

where the volatility vector $\boldsymbol{\sigma}_S$ is given by

$$\boldsymbol{\sigma}_S^T = \begin{pmatrix} \sigma_S\rho_{S1} & \sigma_S\rho_{S2} & \sigma_S\sqrt{1-\rho_{S1}^2-\rho_{S2}^2} \end{pmatrix}. \tag{3.15$b$}$$

For the $G2$++ model, we obtain

$$\begin{aligned}
\overline{c}(\tau) &= \int_t^T \left[\boldsymbol{\sigma}_S - \boldsymbol{\sigma}_D(u;T)\right]^T \left[\boldsymbol{\sigma}_D(u;T+\tau) - \boldsymbol{\sigma}_D(u;T)\right]\,du \\
&= \frac{\sigma_1^2}{\kappa_1}\left(\frac{1-e^{-\kappa_1\tau}}{\kappa_1}\right)\left[\frac{1-e^{-2\kappa_1(T-t)}}{2\kappa_1}\right] + \frac{\sigma_2^2}{\kappa_2}\left(\frac{1-e^{-\kappa_2\tau}}{\kappa_2}\right)\left[\frac{1-e^{-2\kappa_2(T-t)}}{2\kappa_2}\right] \\
&+\ \rho\sigma_1\sigma_2\left(\frac{2-e^{-\kappa_1\tau}-e^{-\kappa_2\tau}}{\kappa_1\kappa_2}\right)\left[\frac{1-e^{-(\kappa_1+\kappa_2)(T-t)}}{\kappa_1+\kappa_2}\right] \\
&-\ \left(\frac{\sigma_1^2}{\kappa_1} + \frac{\rho\sigma_1\sigma_2}{\kappa_2} + \sigma_1\sigma_S\rho_{S1}\right)\left(\frac{1-e^{-\kappa_1\tau}}{\kappa_1}\right)\left[\frac{1-e^{-\kappa_1(T-t)}}{\kappa_1}\right] \\
&-\ \left[\frac{\sigma_2^2}{\kappa_2} + \frac{\rho\sigma_1\sigma_2}{\kappa_1} + \sigma_2\sigma_S(\rho_{S1}\rho + \rho_{S2}\sqrt{1-\rho^2})\right]\left(\frac{1-e^{-\kappa_2\tau}}{\kappa_2}\right)\left[\frac{1-e^{-\kappa_2(T-t)}}{\kappa_2}\right],
\end{aligned} \tag{3.16}$$

and

$$\begin{aligned}
\overline{v}^2(\tau) &= \int_t^T \left\|\boldsymbol{\sigma}_D(u,T+\tau) - \boldsymbol{\sigma}_D(u,T)\right\|^2\,du \\
&= \sigma_1^2\left(\frac{1-e^{-\kappa_1\tau}}{\kappa_1}\right)^2\left[\frac{1-e^{-2\kappa_1(T-t)}}{2\kappa_1}\right] + 2\rho\sigma_1\sigma_2\left(\frac{1-e^{-\kappa_1\tau}}{\kappa_1}\right)\left(\frac{1-e^{-\kappa_2\tau}}{\kappa_2}\right) \\
&\left[\frac{1-e^{-(\kappa_1+\kappa_2)(T-t)}}{\kappa_1+\kappa_2}\right] + \sigma_2^2\left(\frac{1-e^{-\kappa_2\tau}}{\kappa_2}\right)^2\left[\frac{1-e^{-2\kappa_2(T-t)}}{2\kappa_2}\right].
\end{aligned} \tag{3.17}$$

## 4. Edgeworth expansion

From Eq. (2.6), the calculation of the GAO value amounts to the determination of $P_{Q_{F_n}}[a_R(T) > g], n = 0, 1, 2, \cdots$ (note that $Q_{F_n}$ becomes $Q_S$ when $n = 0$). Let $\pi^{(n)}(a)$ denote the density function of $a_R(T)$ under the measure $Q_{F_n}$. We estimate $P_{Q_{F_n}}[a_R(T) > g]$ by performing a cumulant expansion of $\pi^{(n)}(a)$. The cumulants of a distribution are related to the moments of a distribution. The first two cumulants of a distribution are simply the mean and variance of the distribution, and there exists an one-to-one relationship between moments and cumulants. Let $m_j^{(n)}$ and $c_j^{(n)}$ denote the $j^{\text{th}}$ moment and $j^{\text{th}}$ cumulants of $\pi^{(n)}(a)$. It is well known that (Collin-Dufresne and Goldstein, 2002)

$$c_1^{(n)} = m_1^{(n)}, \quad c_2^{(n)} = m_2^{(n)} - (m_1^{(n)})^2, \quad c_3^{(n)} = m_3^{(n)} - 3m_1^{(n)}m_2^{(n)} + 2(m_1^{(n)})^3, \text{ etc.} \tag{4.1}$$

We would like to approximate $P_{Q_{F_n}}[a_R(T) > g]$ in terms of the first three cumulants. Also, we illustrate how to find the first three moments by solving a system of Ricatti equations. By virtue of Eq. (2.6), we then obtain an approximate price formula of the GAO value based on the Edgeworth expansion of $\pi^{(n)}(a)$.

1. Approximation of $P_{Q_{F_n}}[a_R(T) > g]$ in terms of cumulants

   Let $\Pi^{(n)}(\lambda)$ denote the characteristic function of $a_R(T)$ under $Q_{F_n}$, where

$$\Pi^n(\lambda) = E_{Q_{F_n}}[e^{i\lambda a_R(T)}] = \int_{-\infty}^{\infty} e^{i\lambda a}\pi^{(n)}(a)\,da. \tag{4.2}$$

   The cumulants are defined as the coefficients of a Taylor series expansion of the logarithm of the characteristic function, where

$$\ln \Pi^{(n)}(\lambda) = \sum_{j=1}^{\infty} c_j \frac{(i\lambda)^j}{j!}. \tag{4.3}$$

   By taking the Fourier inversion of $\Pi^{(n)}(\lambda)$ and keeping cumulants only up to the third order, we obtain

$$\begin{aligned}
\pi^{(n)}(a) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\lambda a}\Pi^{(n)}(\lambda)\,d\lambda \\
&= \frac{1}{2\pi}\int_{-\infty}^{\infty} \exp\left(-i\lambda a + i\lambda c_1^{(n)} - \frac{c_2^{(n)}}{2}\lambda^2 - i\frac{c_3^{(n)}}{6}\lambda^3 + o(\lambda^3)\right) d\lambda \\
&\approx \frac{1}{2\pi}\int_{-\infty}^{\infty} \exp\left(-i(a - c_1^{(n)})\lambda - \frac{c_2^{(n)}}{2}\lambda^2\right)\left(1 - \frac{ic_3^{(n)}}{6}\lambda^3\right) d\lambda.
\end{aligned}$$

   After some tedious integration procedure, we obtain

$$\pi^{(n)}(a) \approx \left[\frac{1}{\sqrt{c_2^{(n)}}} - \frac{c_3^{(n)}(a - c_1^{(n)})}{2(c_2^{(n)})^{5/2}} + \frac{c_3^{(n)}(a - c_1^{(n)})^3}{6(c_2^{(n)})^{7/2}}\right] n\left(\frac{a - c_1^{(n)}}{\sqrt{c_2^{(n)}}}\right), \tag{4.4}$$

where $n(x) = \dfrac{1}{\sqrt{2\pi}} e^{-x^2/2}$. Furthermore, we deduce that

$$\begin{aligned} P_{Q_{F_n}}[a(T) > g] &= \int_g^\infty \pi^{(n)}(a)\, da \\ &\approx N(z_1) + \frac{c_3^{(n)}}{6(c_2^{(n)})^{3/2}}(z_1^2 - 1)n(z_1), \end{aligned} \tag{4.5}$$

where

$$z_1 = \frac{c_1^{(n)} - g}{\sqrt{c_2^{(n)}}}.$$

2. Determination of the moments of $a_R(T)$ under the measure $Q_{F_n}$

We would like to find the $j^{\text{th}}$ moment of $a_R(T)$ under the measure $Q_{F_n}$ as defined by

$$m_j^{(n)} = E_{Q_{F_n}}[a(T)^j]. \tag{4.6}$$

Note that

$$\begin{aligned} a(T)^j &= \left[ \sum_{n=0}^{\omega-R-1} {}_n p_R D_{T+n}(T) \right]^j \\ &= \sum_{n_1,n_2,\cdots,n_j=0}^{\omega-R} ({}_{n_1}p_R\, {}_{n_2}p_R \cdots {}_{n_j}p_R)[D_{T+n_1}(T) D_{T+n_2}(T) \cdots D_{T+n_j}(T)] \end{aligned}$$

so that

$$\begin{aligned} m_j^{(n)} = &\sum_{n_1,n_2,\cdots,n_j=0}^{\omega-R-1} ({}_{n_1}p_R\, {}_{n_2}p_R \cdots {}_{n_j}p_R) \\ &E_{Q_{F_n}}\left[ \exp\left( -\sum_{k=1}^{j} \left[ \boldsymbol{A}_{T+n_k}(T)^T \boldsymbol{x}(T) + B_{T+n_k}(T) \right] \right) \right]. \end{aligned} \tag{4.7}$$

The moments are seen to have the exponential affine form. For nice analytical tractability associated with the Gaussian type models, we assume that the drift term $\boldsymbol{\mu}(\boldsymbol{x}, t)$ takes the linear form

$$\boldsymbol{\mu}(\boldsymbol{x}, t) = \boldsymbol{\mu}_0(t) + \mu_1(t)\boldsymbol{x},$$

where $\boldsymbol{\mu}_0(t)$ is a $\ell$-component vector and $\mu_1(t)$ is a $\ell \times \ell$ matrix. The expectation term in Eq. (4.7) can be evaluated by solving a system of Ricatti equations.

By following the standard evaluation procedures in affine term structure models, we obtain

$$\begin{aligned} & E_{Q_{F_n}}\left[\exp\left(-\sum_{k=1}^{j} \boldsymbol{A}_{T+n_k}(T)^T \boldsymbol{x}(T) - B_{T+n_k}(T)\right)\right] \\ &= \exp\left(-\boldsymbol{G}_T(t)^T \boldsymbol{x}(t) - G_T^0(t)\right) \qquad (4.8) \end{aligned}$$

where $\boldsymbol{G}_T(t)$ and $G_T^0(t)$ have dependence on $n_1, n_2, \cdots, n_j$ and $n$, and they satisfy the following systems of Ricatti equations.

(i)
$$\begin{aligned} & \frac{d\boldsymbol{G}_T(t)}{dt} + \mu_1(t)^T \boldsymbol{G}_T(t) = \mathbf{0} \\ & \boldsymbol{G}_T(T) = \sum_{k=1}^{j} \boldsymbol{A}_{T+n_k}(T); \qquad (4.9) \end{aligned}$$

(ii)
$$\begin{aligned} & \frac{dG_T^0}{dt} + \boldsymbol{G}_T(t)^T \left\{\boldsymbol{\mu}_0(t) + \sigma(t)\sigma_S(t) + \sigma(t)\sigma(t)^T\left[\boldsymbol{A}_T(t) - \boldsymbol{A}_{T+n}(t)\right]\right\} \\ &= \frac{1}{2}\boldsymbol{G}_T(t)^T \sigma(t)\sigma(t)^T \boldsymbol{G}_T(t) \\ & G_T^0(T) = \sum_{k=1}^{j} B_{T+n_k}(T). \qquad (4.10) \end{aligned}$$

Let $\Phi_T(t)$ be the solution to the following system of differential equations

$$\begin{aligned} & \frac{d\Phi_T(t)}{dt} = -\mu_1(t)^T \Phi_T(t) \\ & \Phi_T(T) = I \qquad (4.11) \end{aligned}$$

where $\Phi_T(t)$ is a $\ell \times \ell$ matrix and $I$ is the $\ell \times \ell$ identity matrix. It can be shown that

$$\Phi_T(t) = \exp\left(\int_t^T \mu_1(u)^T \, du\right). \qquad (4.12)$$

Now, the closed form solution to $\boldsymbol{G}_T(t)$ and $G_T^0(t)$ can be expressed in terms of $\Phi_T(t)$ as follows

$$\boldsymbol{G}_T(t) = \Phi_T(t)\boldsymbol{G}_T(T) = \sum_{k=1}^{j} \exp\left(\int_t^T \mu_1(u)^T \, du\right) \boldsymbol{A}_{T+n_k}(T) \qquad (4.13a)$$

$$\begin{aligned} G_T^0(t) = G_T^0(T) + \int_t^T \boldsymbol{G}_T(u)^T & \Big\{\boldsymbol{\mu}_0(u) + \sigma(u)\boldsymbol{\sigma}_S(u) \\ & + \sigma(u)\sigma(u)^T \left[\boldsymbol{A}_T(u) - \boldsymbol{A}_{T+n}(u) - \frac{\boldsymbol{G}_T(u)}{2}\right]\Big\}\, du. \qquad (4.13b) \end{aligned}$$

**Two-factor Gaussian model**

We illustrate how to compute $\boldsymbol{G}_T(t)$ and $G_T^0(t)$ using the two-factor Gaussian interest rate model defined by Eqs. (3.13a,b) and the equity fund dynamics defined by Eqs. (3.15a,b). The volatility matrix $\sigma(t)$ is given by

$$\sigma(t) = \begin{pmatrix} \sigma_1 & 0 & 0 \\ \sigma_2\rho & \sigma_2\sqrt{1-\rho^2} & 0 \end{pmatrix}$$

so that

$$\sigma(t)\sigma(t)^T = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}.$$

The solution to $\Phi_T(t)$ is found to be

$$\Phi_T(t) = \begin{pmatrix} e^{-\kappa_1(T-t)} & 0 \\ 0 & e^{-\kappa_2(T-t)} \end{pmatrix}$$

while $\boldsymbol{A}_T(t)$ and $B_T(t)$ are given by Eqs. (3.14a,b).

Finally, the solution to $\boldsymbol{G}_T(t)$ and $G_T^0(t)$ are given by

$$\boldsymbol{G}_T(t) = \begin{pmatrix} G_{T,1}(T)e^{-\kappa_1(T-t)} \\ G_{T,2}(T)e^{-\kappa_2(T-t)} \end{pmatrix} \text{ where } G_T(T) = \begin{pmatrix} G_{T,1}(T) \\ G_{T,2}(T) \end{pmatrix}, \tag{4.14$a$}$$

and

$$\begin{aligned} G_T^0(t) = {} & G_T^0(T) + \sigma_1\sigma_S\rho_{S1}G_{T,1}(T)\left(\frac{1-e^{-\kappa_1(T-t)}}{\kappa_1}\right) \\ & + \sigma_2\sigma_S(\rho_{S1}\rho + \rho_{S2}\sqrt{1-\rho^2})G_{T,2}(T)\left(\frac{1-e^{-\kappa_2(T-t)}}{\kappa_2}\right) \\ & - \sigma_1^2 G_{T,1}(T)\left[\frac{1-e^{-\kappa_1 n}}{\kappa_1} + \frac{G_{T,1}(T)}{2}\right]\left(\frac{1-e^{-2\kappa_1(T-t)}}{2\kappa_1}\right) \\ & - \sigma_2^2 G_{T,2}(T)\left[\frac{1-e^{-\kappa_2 n}}{\kappa_2} + \frac{G_{T,2}(T)}{2}\right]\left(\frac{1-e^{-2\kappa_2(T-t)}}{2\kappa_2}\right) \\ & - \rho\sigma_1\sigma_2\Bigg[G_{T,1}(T)\left(\frac{1-e^{-\kappa_2 n}}{\kappa_2}\right) + G_{T,2}(T)\left(\frac{1-e^{-\kappa_1 n}}{\kappa_1}\right) \\ & + G_{T,1}(T)G_{T,2}(T)\Bigg]\left[\frac{1-e^{-(\kappa_1+\kappa_2)(T-t)}}{\kappa_1+\kappa_2}\right]. \end{aligned} \tag{4.14$b$}$$

## 5. Affine approximation approach

Unlike the Edgeworth expansion approach, Singleton and Umantsev (2002) propose to approximate the probability of exercising the option $P_{Q_{F_n}}[a_R(\boldsymbol{x}, T) > g]$ through an approximation of the exercise region itself. They show that if all the future cashflows are positive, then the boundary of the in-the-money region $\{a_R(\boldsymbol{x}, T) > g\}$ is a concave surface. Their method involves the linearization of the exercise boundary by fitting a hyperplane $\boldsymbol{\beta}^T\boldsymbol{x} = 1$ that approximates the exercise boundary $a_R(\boldsymbol{x}, T) = g$. The probability of exercising $P_{Q_{F_n}}[a_R(\boldsymbol{x}, t) > g]$ is then approximated by either $P_{Q_{F_n}}[\boldsymbol{\beta}^T\boldsymbol{x} > 1]$ or $P_{Q_{F_n}}[\boldsymbol{\beta}^T\boldsymbol{x} < 1]$ (whose choice depends on the location of the exercise region). For the Gaussian type models, $\boldsymbol{\beta}^T\boldsymbol{x}(T) = \beta_1 x_1(T) + \cdots + \beta_\ell x_\ell(T)$ is normally distributed whose mean and variance are given by $\boldsymbol{\beta}^T\overline{\boldsymbol{\mu}}_{\boldsymbol{x}}$ and $\boldsymbol{\beta}^T\sigma_{\boldsymbol{x}}\boldsymbol{\beta}$, where $\overline{\boldsymbol{\mu}}_{\boldsymbol{x}}$ and $\sigma_{\boldsymbol{x}}$ are the conditional mean vector and covariance matrix of $\boldsymbol{x}(T)$ given $\boldsymbol{x}(t)$ under $Q_{F_n}$.

**Fitting algorithm**

Consider a two-factor interest rate model with two risk factors, the fitting algorithm involves the following steps.

1. Choose a level of significance $\alpha$ (say, 1%), then find the two values $x_{2,\alpha/2}$ and $x_{2,1-\alpha/2}$ such that
$$P_{Q_{F_n}}[x_{2,\alpha/2} < x_2(T) < x_{2,1-\alpha/2}] = 1 - \alpha.$$
2. Once $x_{2,\alpha/2}$ and $x_{2,1-\alpha/2}$ are known, solve for $x_{1,\alpha/2}$ and $x_{1,1-\alpha/2}$ so that the two points $(x_{1,\alpha/2}, x_{2,\alpha/2})$ and $(x_{1,1-\alpha/2}, x_{2,1-\alpha/2})$ fall on the exercise boundary: $a(\boldsymbol{x}, T) = g$.
3. Fit a hyperplane (a line in the case of a two-factor interest rate model)
$$\beta_1 x_1 + \beta_2 x_2 = 1$$
to the two points determined in Step 2 by solving for the parameters $\beta_1$ and $\beta_2$ through
$$\boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} = \begin{pmatrix} x_{1,\alpha/2} & x_{2,\alpha/2} \\ x_{1,1-\alpha/2} & x_{2,1-\alpha/2} \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$
Choose the appropriate region $\{\boldsymbol{\beta}^T\boldsymbol{x} > 1\}$ or $\{\boldsymbol{\beta}^T\boldsymbol{x} < 1\}$ so as to approximate the exercise region $\{a_R(\boldsymbol{x}, t) > g\}$.

## 6. Numerical results

In this section, we present our numerical experiments that were performed to compare the numerical accuracy and computational efficiency of the three analytic approximation methods. Also, we explore how the GAO value depends on the guaranteed conversion rate $g$ and various correlation coefficients in the pricing model.

In our numerical calculations, we use the following set of parameter values in the pricing model (unless otherwise specified).

*Parameters in the equity and interest rate models*

$$\begin{array}{lllll} S_t = 100, & q = 5\%, & \sigma_S = 10\%, & \kappa_1 = 0.77, & \kappa_2 = 0.08, \\ \sigma_1 = 2\%, & \sigma_2 = 1\%, & \rho = -0.7, & \rho_{S1} = 0.5, & \rho_{S2} = 0.5. \end{array}$$

Current yield curve, $Y(T) = r_0 + 0.04(1 - e^{-0.2T})$ where $r_0$ is taken to assume different constant values.

*Mortality data*

retirement age = 65, maximum age = 100

| $n$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ${}_nP_R$ | 0.9871 | 0.9730 | 0.9578 | 0.9411 | 0.9229 | 0.9029 | 0.8808 | 0.8567 | 0.8304 | 0.8018 | 0.7708 | 0.7374 | 0.7015 | 0.6632 | 0.6226 | 0.5798 | 0.5351 | 0.4889 | 0.4414 | 0.3934 | 0.3454 | 0.2981 | 0.2523 | 0.2088 | 0.1684 | 0.1319 | 0.0998 | 0.0725 | 0.0503 | 0.0330 | 0.0203 | 0.0115 | 0.0059 | 0.0027 | 0.0011 |

*Other parameters*

$g = 9$, time to expiry $= T - t = 15$,

significant level in the affine approximation = 0.01,

100,000 trial runs are performed in each Monte Carlo simulation,

Edgeworth expansion is taken up to the third order.

### Computational efficiency and numerical accuracy

In a typical run in the numerical calculations of the GAO value at a given value of $r_0$, the following computer running times (in minutes) for various methods are recorded in Table 1. Since tedious iterative calculations are required to calculate the higher order moments in the Edgeworth expansion method, our experience shows that even only up to the third order expansion, the running time required by the Edgeworth expansion is longer than that of the Monte Carlo simulation with $100,000$ trials. Since closed form formulas are available in the minimum variance duration method, the required running time is significantly shorter than that of the other numerical methods.

| method | Monte Carlo simulation | Edgeworth expansion | affine approximaton | min. var. duration |
|---|---|---|---|---|
| running time | 0.4305 | 1.136 | 0.1812 | 0.0016 |

**Table 1** Comparison of computer running time (in minutes) of various numerical methods that compute the GAO value.

In Table 2, we list the numerical results of GAO value obtained by various numerical methods at different choices of $r_0$ in the assumed functional form of the yield curve. The agreement between the numerical values is quite well acceptable.

| $r_0\%$ | Monte Carlo Simulation | Edgeworth expansion | affine approximation | min. var. duration |
|---|---|---|---|---|
| 0.5 | 11.7750 | 11.8161 | 11.7913 | 11.8100 |
| 1.0 | 9.7568 | 9.7502 | 9.7412 | 9.7714 |
| 1.5 | 7.8952 | 7.8479 | 7.8529 | 7.8958 |
| 2.0 | 6.1543 | 6.1293 | 6.1418 | 6.1946 |
| 2.5 | 4.6735 | 4.6199 | 4.6313 | 4.6860 |
| 3.0 | 3.3793 | 3.3408 | 3.3464 | 3.3911 |
| 3.5 | 2.3257 | 2.2999 | 2.3044 | 2.3273 |
| 4.0 | 1.5116 | 1.4897 | 1.5057 | 1.5008 |
| 4.5 | 0.9222 | 0.8942 | 0.9310 | 0.9008 |
| 5.0 | 0.5201 | 0.4922 | 0.5439 | 0.4984 |

**Table 2** Comparison of numerical results of GAO value obtained by various numerical methods.

We also explore the pricing errors of the three analytic approximation methods. Using the Monte Carlo results as the benchmark, we calculate the GAO value at 100 different value of $r_0$ ($r_0 = 0.1\%, 0.2\%, \cdots, 10\%$) and compute the percentage error of each analytic approximation method. The variation of the percentage error with respect to $\dfrac{\overline{a}}{gD_T(t)}$ is plotted in Fig. 1. The pricing error is typically less than 1% when $\dfrac{\overline{a}}{gD_T(t)} > 1$ (the annuity option is currently in-the-money) while the accuracy deteriorates when $\dfrac{\overline{a}}{gD_T(t)}$ falls below 1. Similar behaviors on numerical accuracy are exhibited in swaption calculations using the affine approximation method (Singleton and Umantsev, 2002) and minimum variance duration method (Munk, 1999).

**Pricing behaviors of the guaranteed annuity option**

We investigate the pricing behaviors of the GAO with respect to various parameters in the pricing model. In Fig. 2, we plot the GAO value against $g$ at varying values of $T-t$. The curves exhibit consistency with the intuition that the GAO value should be a decreasing function of $g$. The rate of decrease of GAO value is higher at a lower value of $g$. Also, the GAO has a higher value when the policyholder enters the contract closer to retirement (smaller value of $T-t$). This is related to the time value of money since smaller $T-t$ means shorter time horizon over which the annuity payments are discounted. This effect counteracts the usual theta effect of option value, which implies that a longer-lived option usually has a higher value.

Our GAO pricing model assumes that the interest rate dynamics is governed by two risk factors (G2++ model). Therefore, there are 3 correlation coefficients in the model, namely, the correlation coefficient $\rho$ between the interest rate risk factors, the correlation coefficients $\rho_{S1}$ and $\rho_{S2}$ between the stock price process and the risk factors. In Fig. 3, we plot the GAO value against $\rho$ with different sets of values of $\rho_{S1}$ and $\rho_{S2}$. In the analytic approximation price formula, the functional dependence of the GAO value on these correlation coefficients appears to be so highly complicated that any theoretical analysis is intractable. From the plots in Fig. 3, it is quite disquieting to observe that the GAO value is highly sensitive to the correlation coefficients. Similar phenomena of price sensitivity to the correlation coefficient have also been reported by Ballotta and Haberman (2003).

## 7. Conclusions

Since there is no closed form analytic price formula for a guaranteed annuity option when the interest rate dynamics is modeled by a multi-factor short rate model, the numerical valuation of the guarantee in deferred annuity pension policies is resorted to either Monte Carlo simulation or analytic approximation methods. In this paper, we construct three analytic approximation methods for effective valuation of the annuity option value when the interest rate dynamics is modeled by a multi-factor affine term structure model. The method of minimum variance duration starts with a judicious analytic approximation so that closed form formula can be obtained. The pricing error is minimized by choosing the period $\tau$ of a reference bond such that the variance of the value of the annuity payment normalized by the price of the $(T+\tau)$-maturity bond is minimized. The Edgeworth expansion method seeks the Edgeworth approximation of the probability distribution of the annuity value at option's maturity. In the affine approximation, the exercise probability of the annuity option is approximated through the approximation of the concave exercise boundary by a hyperplane. When these three analytic approximation methods are compared in terms of numerical accuracy and computational efficiency, the method of minimum variance duration seems to have the best performance among them. When the annuity option is in-the-money or slightly out-of-the-money, the pricing error of these approximation methods are within a few percentage points. Though the three-term Edgeworth expansion demonstrates sufficient accuracy, the computational time required is even longer than that of the Monte Carlo simulation method using $100,000$ simulation paths.

Our numerical studies showed that the value of the annuity option is highly dependent on the guaranteed conversion rate of the annuity and the correlation coefficients among the risk factors. As future works, one may use a more accurate model to characterize the equity return process, like the use of stochastic volatility model or regime switching model. Also, we may incorporate stochastic mortality effects and other market factors (tax, expenses, etc.) into the pricing model.

## References

Ballotta, L., Haberman, S., 2003a. Valuation of guaranteed annuity conversion options. Insurance: Mathematics and Economics 33, 87-108.

Ballotta, L., Haberman, S. 2003b. The fair valuation problem of guaranteed annuity options: the stochastic mortality environment case. Working paper of City University, London.

Boyle, P., Hardy, M., 2003. Guaranteed annuity options. ASTIN Bulletin 33(2), 125-152.

Collin-Dufresne, P., Goldstein, R.S., 2002. Pricing swaptions within an affine framework. Journal of Derivatives, Fall issue, 1-18.

Cox, J.C., Ingersoll, J.E., Ross, S.A., 1979. Duration and the measurement of basis risk. Journal of Business 51(1), 51-61.

Dai, Q., Singleton, K.J., 2000. Specification analysis of affine term structure models. Journal of Finance 55, 1943-1978.

Heath, D., Jarrow, R., Morton, A., 1992. Bond pricing and term structure of interest rates: a new methodology for contingent claims valuation. Econometrica 60(1), 77-105.

Jamshidian, F., 1989. An exact bond option formula. Journal of Finance, 44(1), 205-209.

Munk, C., 1999. Stochastic duration and fast coupon bond option pricing in multi-factor models. Reviews of Derivatives Research 3, 157-181.

O'Brien, C., 2001. Guaranteed annuity options: Five issues for resolution. Working paper of University of Nottingham.

Pelsser, A., 2003. Pricing and hedging guaranteed annuity options via static option replication. Insurance: Mathematics and Economics 33(2), 283-296.

Singleton, K.J., Umantsev, L., 2002. Pricing coupon-bond options and swaptions in affine term structure models. Mathematical Finance 12(4), 427-446.

Wei, J., 1997. A simple approach to bond option pricing. Journal of Futures Markets 17(2), 131-160.

Wilkie, A.D., Waters, H.R., Kong, S., 2004. Reserving, pricing and hedging for policies with guaranteed annuity options. British Actuarial Journal 9(2), 263-425.

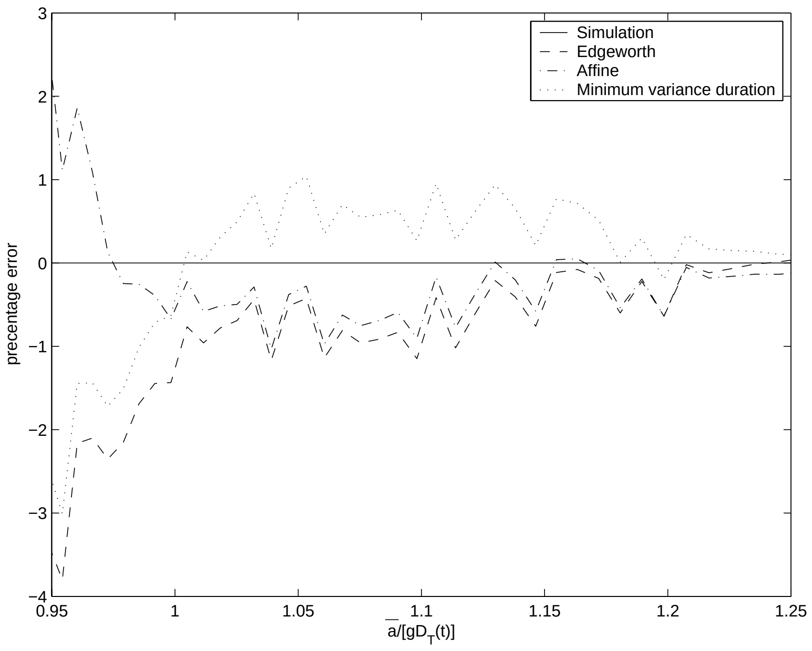


**Fig. 1.** Comparison of the pricing error (in percentage) of various analytic approximation methods. Good numerical accuracy is revealed when $\dfrac{\overline{a}}{gD_T(t)} > 1$ (the annuity option is currently in-the-money).

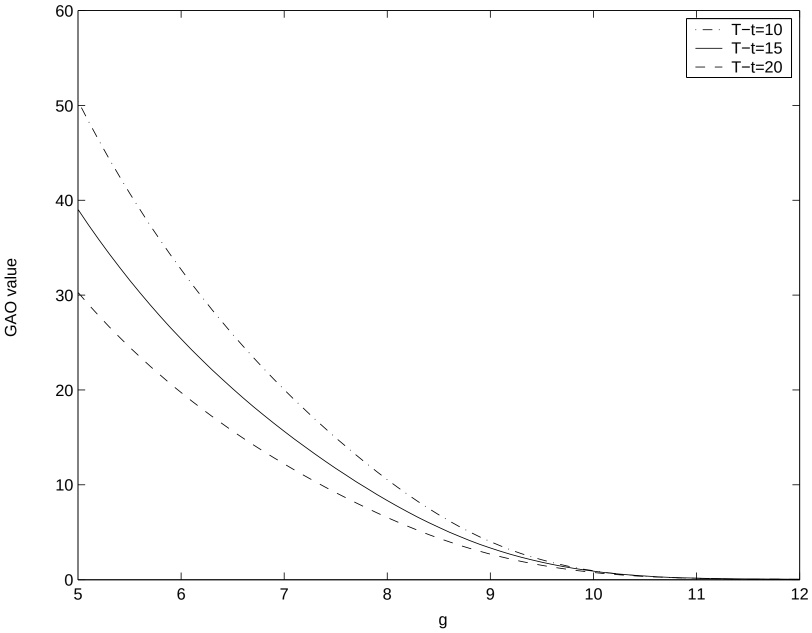


**Fig. 2.** Plot of GAO value against guaranteed conversion rate $g$ at varying values of time to expiry $T - t$.

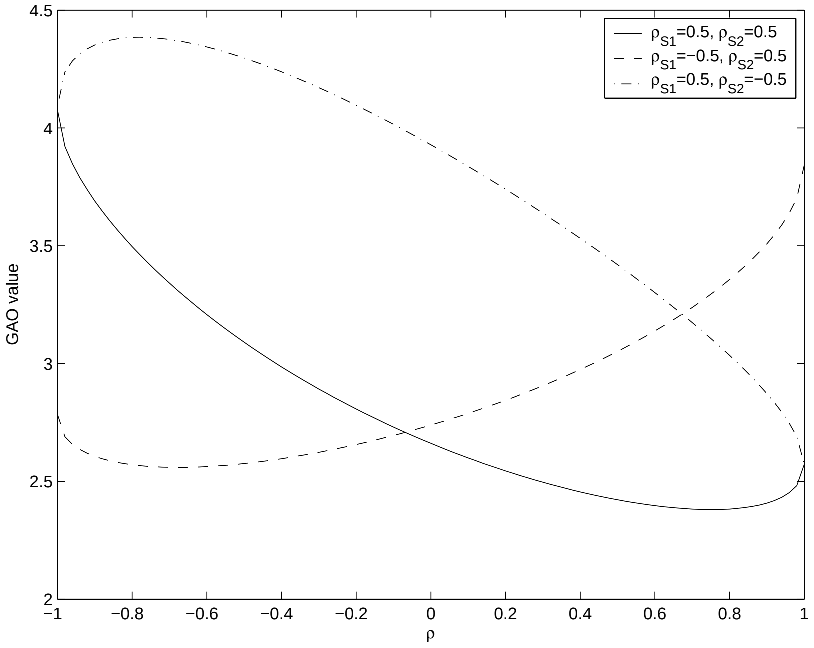


**Fig. 3.** Price sensitivity of the GAO value to the correlation coefficients in the pricing model.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE2007*
*Chicago, 20-23 June 2007*

# Non-linear analysis of cable networks by FEM and experimental validation

**J. J. del Coz Díaz[1], P. J. García Nieto[2], D. Castro Fresno[3] and E. Blanco Fernández[3]**

[1] *Construction and Production Engineering Department, University of Oviedo, Viesques Department Building no. 7, 33204 Gijón (Spain)*
[2] *Department of Mathematics, Faculty of Sciences, University of Oviedo, 33007 Oviedo (Spain)*
[3] *Department of Construction, ETSIPCC, University of Cantabria, 39005 Santander (Spain)*

emails: juanjo@constru.uniovi.es, pauli@constru.uniovi.es, daniel.castro@unican.es, blancoe@unican.es

## Abstract

This work studies the analysis of the resistant capacity of cable nets for the stabilization of slopes. Two tests have been carried out, one with a distributed longitudinal load and the other with distributed transversal load, in order to simulate in situ the working conditions of these systems. Tensile tests were also carried out on the cable elements of the network in order to obtain the non-linear mechanical properties. On the one hand, the proposed numerical procedure uses the finite element method (FEM) and it takes into account the material and geometrical non-linearities due to the geometrical change in the cable net substructure. On the other hand, the cable network is modelled by one-dimensional beam elements with joints between them by means of multibody coupling. The laboratory tests only provide information about the strain and maximum resistance, but they do not establish a relationship between the values of stresses of each net element. These data have been obtained through the computational simulation by FEM. A reliable model of the interaction of the flexible contour beam with the cable network enables the achievement of more efficient solutions in the design analysis. Finally, we compare the structural behaviour of the numerical and experimental results by means of the equivalent elastic modulus and the equivalent Poisson's ratio. Reasonable agreement between the predicted results by FEM and test observations was found.

*Key words: Finite element analysis; cable networks; Material and geometrical non-linearities; Soil stabilization;*
*MSC2000: 74S05, 74C05, 35J60*

## 1. Introduction

On gentle slopes, erosion blankets, both natural and man-made, or hydro-seeding can be employed to hold soil and seed in place until vegetation gets established. These lightweight products may not be enough to control erosion caused by high water velocities associated with steep slopes. Where steeper slopes and high water velocities prevail, a wire mesh and a cable net slope protection system may be used [1-2]. Wire mesh and cable net slope protection have been in use for more than 50 years along highways to control rockfall on actively eroding slopes. The present work studies these last flexible systems of slopes' surface stabilization, the levelling of grounds and loose materials by the finite element method (FEM) as well as the analysis and a subsequent comparison with the experimental results.

The cable network systems provide: (a) control of the erosion; (b) covering; (c) fastening of the ground and (d) surface stabilization. The system works like a lightly tightened continuous surface that receives from the ground the loads due to the earth pressure on the cable networks and it transmits them to the head of the anchorages, which at the same time transmit them to the stable area of the hillside or slope.

Once the system object of the study has been described, the most important characteristics in the numerical simulation will be discussed. Next, we shall carry out an analysis of the obtained results and to conclude we shall do a comparison of numerical results with the experimental tests. Finally we shall analyse the causes of the possible discrepancies present between the experimental and numerical results.

## 2. Mathematical model

The finite element method is used to model the cable network and the boundary structure. Approximation of the structure by a discrete number of finite size straight and curved elements, connected at nodes, is a quite natural simulation since the original cable network consists of turns made of elements and curves. According to stiffness properties the structural system is divided in several substructures (one per wire). Therefore, the resolution of this problem implies the simultaneous study of two non-linearities [3]: (1) material non-linearity (elastoplastic behaviour in this case), and (2) geometrical non-linearity or large displacements.

The most straightforward way to create the mathematical model of a cable network with a flexural contour structure is to include the cable and the beam elements in a common system of equations in general matrix form. If load $P$ is applied at the nodes of the original undeflected structure, an unbalanced resultant force is produced:

$$R = F - P \tag{1}$$

where $F$ is the vector of internal forces at nodes and $P$ is the vector of external forces at nodes. $R$ is a non-linear function with respect to the displacements of the structure. In order to find the equilibrium state geometry, Eq. (1) has to be solved, using relevant numerical methods [4]. According to the Newton-Raphson method we calculate the correction of the solution at every iteration cycle:

$$d^{i} = \left(K^{i}\right)^{-1} R^{i} \tag{2}$$

where $K^i$ is the stiffness matrix of the structure. The new geometry after cycle *i* is determined by the vector of nodal co-ordinates:

$$u^{i+1} = u^i + d^i \tag{3}$$

Nodal displacements produce elastoplastic elongation [3] of the cable elements and change of the tension and bending forces according to axial and flexural stiffness. Therefore stiffness matrix *K* and unbalanced force *R* have to be updated for every iteration cycle according to the current geometry. Convergence of the described method is usually rapid near the equilibrium state, but may be disturbed if elements of the stiffness matrix vary significantly or if the initial geometry is inappropriate.

2.1. Plasticity

The plastic behaviour is characterized by irreversibility of stress paths and the development of permanent (i.e. non-recoverable) deformation (or strain), known as *yielding* (or plastic flow). A hardening plastic material model provides a refinement of the ideal plastic material model. In this model, it is assumed that the yield stress depends on some parameter $\kappa$ (e.g. plastic strain $\varepsilon^p$), called the *hardening parameter*. The general yield criterion is expressed in the form [5]:

$$F(\sigma_{ij}, \kappa) = 0 \tag{4}$$

After initial yielding, the stress level at which further plastic deformation occurs may be dependent on the current degree of plastic straining, known as *strain hardening* [5]. Thus, the yield surface will vary (i.e. expand) at each stage of plastic deformation. If the subsequent yield surfaces are a uniform expansion of the original yield surface, the hardening model is said to be *isotropic*. The behaviour is initially linear elastic with slope *E* (Young's modulus) until onset of yielding at the uniaxial yield stress $\sigma_Y$. Thereafter, the material response is elasto-plastic with the local tangent to the curve, $E_T$, called the elasto-plastic tangent modulus, continually changing.

At some stress level σ in the plastic range, if the load is increased to induce a stress of $d\sigma$, it results in a corresponding strain $d\varepsilon$. This increment of strain contains two parts: elastic $d\varepsilon^e$ (recoverable) and plastic $d\varepsilon^p$ (non-recoverable) [3,5]:

$$d\varepsilon = d\varepsilon^e + d\varepsilon^p, \quad d\varepsilon^e = \frac{d\sigma}{E}, \quad \frac{d\sigma}{d\varepsilon} = E_T \tag{5}$$

The strain-hardening parameter, *H*, is defined by [6-7]:

$$H = \frac{d\sigma}{d\varepsilon^p} = \frac{\frac{d\sigma}{d\varepsilon}}{1 - \frac{d\varepsilon^e}{d\varepsilon}} = \frac{E_T}{1 - \frac{E_T}{E}} \tag{6}$$

The element stiffness for the linear elastic portion is, say $\left[K^e\right]$ [6-7]:

$$\left[K^e\right] = \int_{x_a}^{x_b} [B]^T \left[D^e\right][B]\, dx \tag{7}$$

where $\left[D^e\right]$ is the linear elasticity matrix ($D^e = E$ for the uniaxial case). When the element deforms plastically, $\left[D^e\right]$ reflects the decreased stiffness. This is computed, for uniaxial material behaviour, by the following procedure. The increment in load $dF$ causes an incremental displacement $du$:

$$du = h_e \, d\varepsilon_{xx} = h_e \left( d\varepsilon^e + d\varepsilon^p \right), \quad dF = A \, d\sigma = A_e \, H \, d\varepsilon^p \qquad (8)$$

where $h_e$ is the length and $A_e$ the area of cross-section of the element. The effective stiffness is:

$$E^{ep} = \frac{dF}{du} = \frac{A_e \, H \, d\varepsilon^p}{h_e \left( d\varepsilon^e + d\varepsilon^p \right)} = \frac{E \, A_e}{h_e} \left[ 1 - \frac{E}{(E+H)} \right] \qquad (9)$$

The element stiffness for the plastic range becomes [5-7],

$$\left[K^{ep}\right] = \int_{x_a}^{x_b} [B]^T \left[D^{ep}\right] [B] dx \qquad (10)$$

where $\left[D^{ep}\right]$ is the material stiffness in the plastic range. For the uniaxial case $D^{ep} = E^{ep}$. Eq. (7) is valid when $\sigma < \sigma_Y$ and Eq. (10) is valid for $\sigma > \sigma_Y$. Note that $d\sigma = \sigma - \sigma_Y$ when $\sigma > \sigma_Y$.

## 2.2. Large displacements

Whether the displacements (or strains) are large or small, equilibrium conditions between internal and external 'forces' have to be satisfied. Thus, if the displacements are prescribed in the usual manner by a finite number of nodal parameters $\vec{a}$, we can obtain the necessary equilibrium equations using the *virtual work principle* [7]:

$$\Psi(\vec{a}) = \int_V \overline{B}^T \vec{\sigma} \, dV - \vec{f} = 0 \qquad (11)$$

where $\Psi$ once again represents the sum of external and internal generalized forces, and in which $\overline{B}$ is defined from the strain definition $\vec{\varepsilon}$ as:

$$d\vec{\varepsilon} = \overline{B} \, d\vec{a} \qquad (12)$$

The bar suffix has now been added for, if displacements are large, the strains depend non-linearly on displacement, and the matrix $\overline{B}$ is now dependent on $\vec{a}$. We see that it can be conveniently write:

$$\overline{B} = B_0 + B_L(\vec{a}) \qquad (13)$$

in which $B_0$ is the same matrix as in linear infinitesimal strain analysis and only $B_L$ depends on the displacement. In general, $B_L$ will be found to be a *linear function* of such displacements.

Clearly the solution of Eq. (11) will have to be approached iteratively. If, for instance, the Newton-Raphson process is to be adopted we have to find the relation between $d\vec{a}$ and $d\Psi$. Thus taking appropriate variations of Eq. (11) with respect to $d\vec{a}$ we have [6]:

$$d\Psi = \int_V d\overline{B}^T \, \vec{\sigma} \, dV + \int_V \overline{B}^T \, d\vec{\sigma} \, dV = K_T \, d\vec{a} \qquad (14)$$

and using equation $d\vec{\sigma} = D \, d\vec{\varepsilon}$ and Eq. (12) it is obtained:

$$d\vec{\sigma} = D \, d\vec{\varepsilon} = D \, \overline{B} \, d\vec{a}$$

and taking into account the Eq. (13), it is verified that $d\overline{B} = dB_L$ . Therefore,

$$d\Psi = \int_V dB_L^T \, \vec{\sigma} \, dV + \overline{K} \, d\vec{a} = K_\sigma \, d\vec{a} + \overline{K} \, d\vec{a} \tag{15}$$

where

$$\overline{K} = \int_V \overline{B}^T D \, \overline{B} \, dV = K_0 + K_L$$

in which $K_0$ represents the usual, small displacements stiffness matrix and the matrix $K_L$ is due to the *large displacements*, and are given by [7]:

$$K_0 = \int_V B_0^T \, D \, B_0 \, dV \,, \qquad K_L = \int_V \left( B_0^T \, D \, B_L + B_L^T \, D \, B_L + B_L^T \, D \, B_0 \right) dV \tag{16}$$

To summarize, Eq. (15) can be expressed globally as:

$$d\Psi = \left( K_0 + K_\sigma + K_L \right) d\vec{a} = K_T \, d\vec{a} \tag{17}$$

where $K_T$ represents the total, *tangential stiffness*, matrix. Newton-type iteration can once more be applied precisely in order to solve the final non-linear problem.

### 2.3. Multibody coupling by joints

One of the major areas of nonlinear analysis is the solution of problems in which separate bodies or structures may come in contact with each other. Several methods have been developed to handle such problems and, in this paper, the multibody coupling has been adopted [7-8].

Often it is desirable to have two (or more) rigid bodies connected in some specified manner. For example, in our case, each turn in the cable network is hooked to another. Both turns are treated as flexible non-linear bodies and it is necessary to consider coupling among them in order to obtain the global structural behavior of the network. This type of interconnection is commonly referred to as a *joint*. When generating our model, we define the relationships among different degrees of freedom by using elements to connect the nodes.

In this work, we use a joint which is a linear connection where one body may freely move around the other but relative translation is prevented (see Fig. 1 below). Thus each turn may not translate relative to another in any direction. If a full translation constraint is imposed a simple relation may be introduced as [7]:

$$\vec{C}_j = \vec{x}^{(a)} - \vec{x}^{(b)} = \vec{0} \tag{18}$$

where *a* and *b* denote two rigid bodies.

## 3. Geometrical model

Several geometrical models were analyzed in this work. In order to carry out a right numerical simulations, it is necessary to reproduce accurately both the geometrical and mechanical characteristics of the system object of study. In the present work, the characteristics of meshes are shown in Fig. 2.

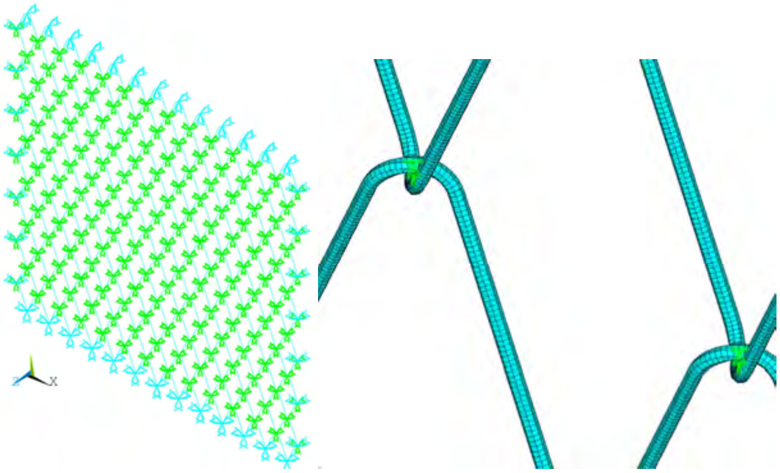


Fig.1. Couplings among turns of the cable network.

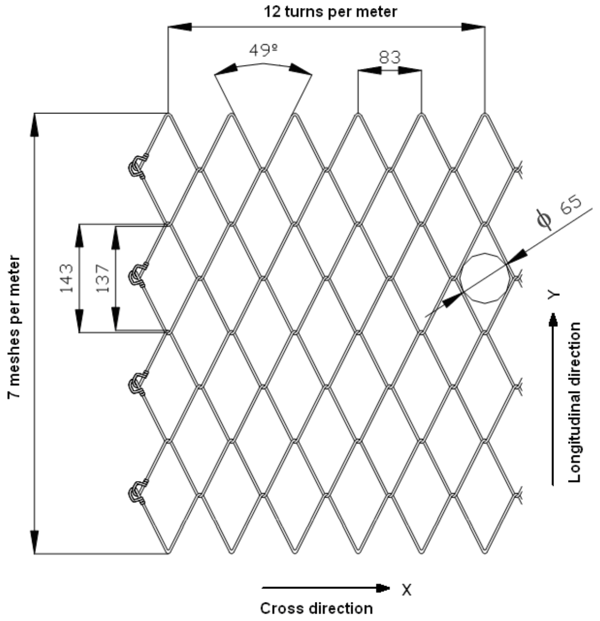


Fig. 2. Geometrical model of a cable network.

## 4. Finite element model and analysis

Based on the geometrical model of a piece of cable network (see Fig.2) the finite element model was built, following a four-step process. Firstly, the definition of mechanical properties is based on real tests. Secondly, the selection of the element types, formulations and physical properties was made. Thirdly, the geometrical model was meshed. Finally, loads and boundary conditions were applied and the model was solved.

4.1. Mechanical properties

The mechanical properties of the high strength steel such as its Young's modulus, its yield stress and its ultimate stress have been obtained by means of the experimental results from direct tensile tests carried out to these wires. In this way, for the 3 mm diameter wire the mechanical characteristics obtained are shown in Table 1.

Table 1. Mechanical properties of the 3 mm diameter wire.

| **Stress [MPa]** | **Strain** | **Young's Modulus [MPa]** |
|---|---|---|
| 0.000 | 0.0000 | |
| 1,101.328 | 0.0056 | 196,906 |
| 1,482.037 | 0.0093 | |
| 1,794.777 | 0.0139 | |
| 1,826.427 | 0.0300 | |

4.2. Element types and meshing

The element used in this study is a quadratic (3-node) beam element with six degrees of freedom at each node (include translations in the *x*, *y*, and *z* directions and rotations about the *x*, *y*, and *z* directions) and it includes stress stiffness terms [3,9], since for its characteristics is the most appropriate for the problem. This is an one-dimensional element well-suited for large rotation and large strain nonlinear applications as in this work. This element is based on Timoshenko beam theory and shear deformation effects are included. In this work a extremely fine mesh has been used in order to obtain a good accurateness. This mesh has an element size of 1 millimetre, giving rise to a mesh of approximately 70,000 elements (see Fig. 3).

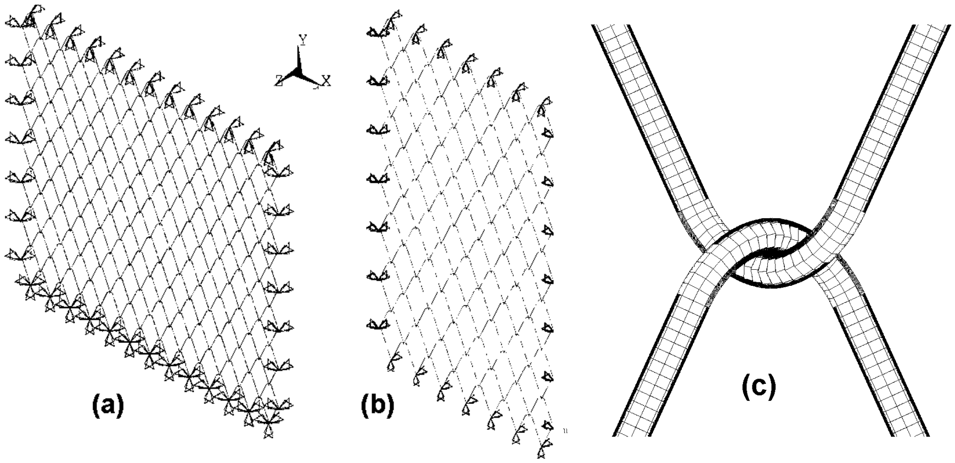


Fig. 3. Finite element mesh and boundary conditions: (a) entire mesh for the direct tensile test (left), (b) entire mesh for the cross tensile test (middle) and (c) detail of a connection (right).

4.3. Loads and boundary conditions

Two different boundary conditions have been applied in the numerical model:

- On the one hand, in order to simulate the direct tensile test, several boundary conditions have to be imposed in the finite element model (see Fig. 3(a)). Firstly, at the lower side of the mesh, displacements in directions *x*, *y* and *z*, are constrained. Secondly, at the left and right sides, displacements in directions *x* and *z* are constrained, in order to allow the displacement in vertical direction *y*. Finally, at the upper side, the displacement in *z* direction is constrained and it is imposed a displacement in *y* direction of 60 mm.
- On the other hand, the cross tensile test requires different boundary conditions (see Fig. 3(b)). In the first place, at the upper and lower sides, displacements in directions *z* and *y* are constrained, allowing the displacement in direction *x*. Secondly, at the left side, displacements in directions *x* and *z* are constrained. Finally, at the right side, a displacement in direction *x* of 60 mm is imposed.

## 5. Non-linear analysis of cable networks

A non-linear analysis was performed taking into account the geometrical and material non-linearities: large displacements and plasticity.

The solution controls were also adjusted to improve convergence. Thus the parameter time was set to the value of one, corresponding to the value of maximum displacement applied (60 mm), the geometrical non-linearity was activated, the inertial effects were not included, the number of equilibrium iterations was specified and the convergence tolerance values of displacements were delimited as well as the time step for the analysis. In this work, the total number of iterations in order to get convergence was about 160.

## 6. Analysis of results and discussion

On the one hand, we show the von-Mises stresses obtained by FEM for the direct tensile test and the cross tensile test (see Fig. 4). On the other hand, we show the experimental patent device with the cable networks used in the Cantabria University laboratory in order to determine the experimental results (see Fig. 5).

Graphs representing both the numerical and experimental results are shown in Fig. 6, in order to get the longitudinal and transversal equivalent elastic moduli by means of a linear fitting from data. The values obtained are:

1) From the direct tensile test (longitudinal equivalent elastic modulus):

| | |
|---|---|
| Numerical simulation: | 1,690.6 kN/m. |
| Experimental test: | 1,886.3 kN/m. |

2) From the cross tensile test (transversal equivalent elastic modulus):

| | |
|---|---|
| Numerical simulation: | 319.9 kN/m. |
| Experimental test: | 222.47 kN/m. |

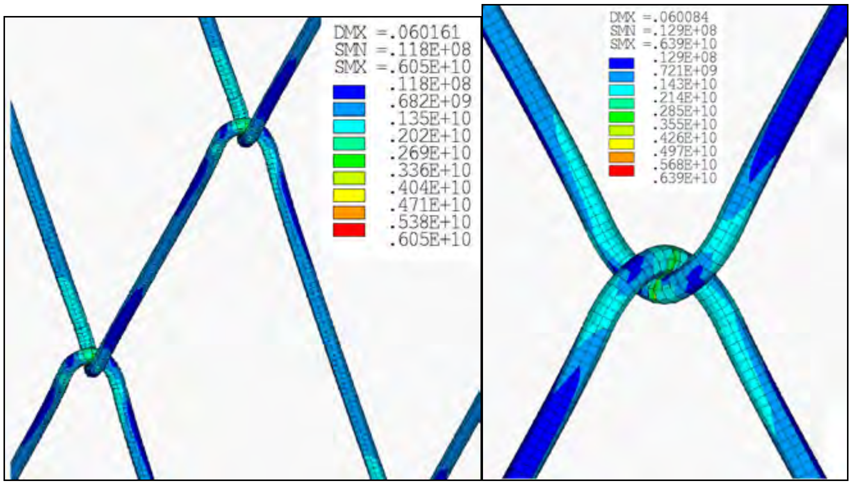


Fig. 4. Detail of von-Mises stresses (Pa) for the direct tensile test (left) and for the cross tensile test (right).

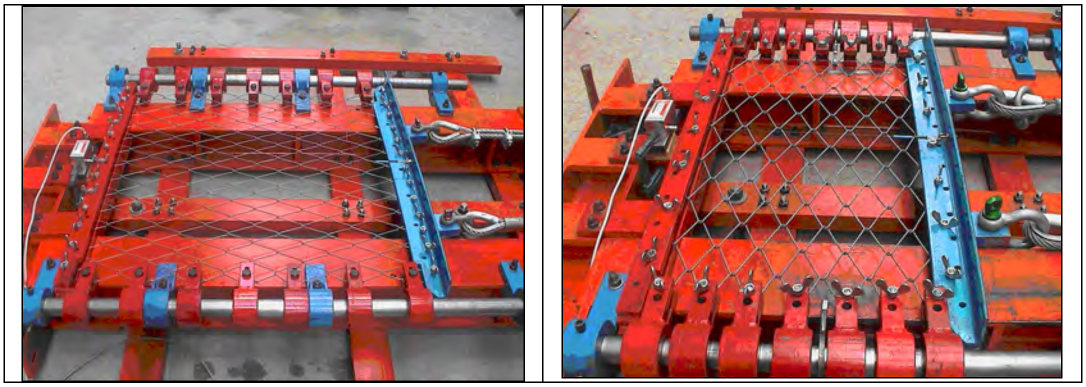


Fig. 5. The cable network tested: direct tensile test (left) and cross tensile test (right).

In order to simulate the membrane behaviour of the cable networks, it is necessary to calculate the relationship between the longitudinal and transversal stresses for the two types of tests: direct tensile test and cross tensile test. The quotient of the transversal stress divided by the longitudinal stress are named *equivalent Poisson's ratio*:

$$\mu_{xy}^{eq} = \frac{\sigma_y}{\sigma_x} \tag{19}$$

Fig. 7 shows the $\mu_{xy}^{eq}$ calculated by linear fitting from experimental and numerical data for the direct tensile test and cross tensile test. The values obtained are:

1) From the direct tensile test (longitudinal equivalent elastic modulus):
(a) numerical simulation: 0.33; (b) experimental test: 0.217
2) From the cross tensile test (transversal equivalent elastic modulus):
(a) Numerical simulation: 0.4437; (b) experimental test: 0.4721

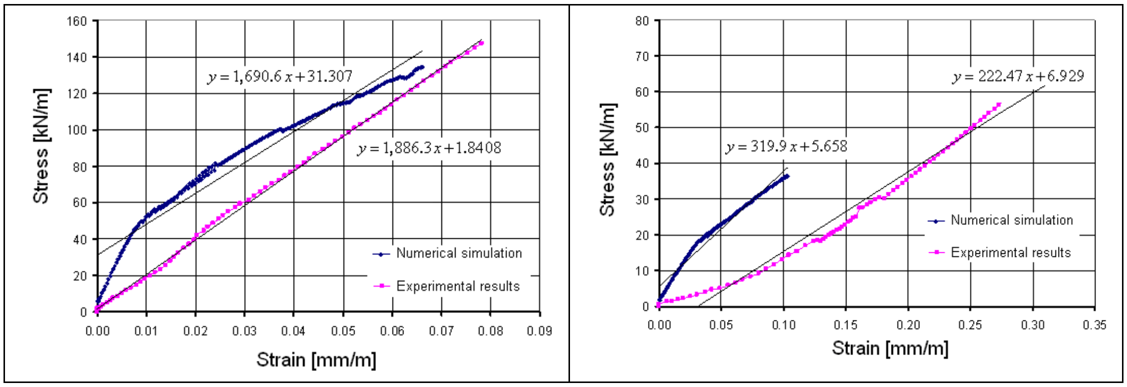


Fig. 6. Stress vs. strain curves obtained from the numerical simulation by FEM and from experimental test: (a) direct tensile test (left) and (b) cross tensile test (right).

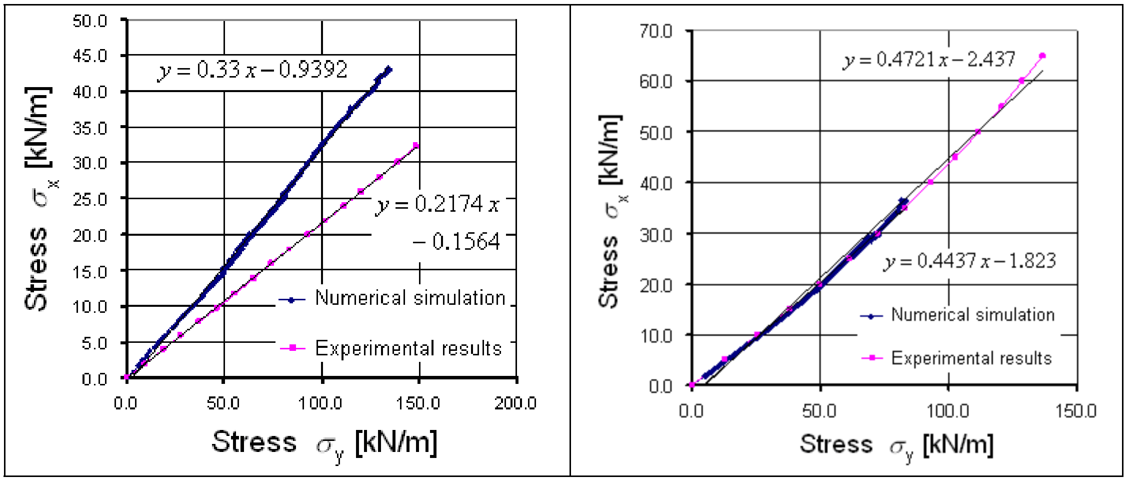


Fig. 7. Equivalent Poisson´s ratio: (a) direct tensile test (left) and (b) cross tensile test (right).

## 7. Conclusions

A method for modelling static behaviour of cable networks has been developed and verified here. The equilibrium equations of elasto-plastic network are derived in the incremental form. The procedure can serve as an alternative tool in order to avoid expensive physical tests in the laboratory of different configurations and geometries of cable networks. From the results obtained, the following conclusions can be drawn:

- In the first place, the finite element method (FEM) has been shown as suitable tool in the modelling and analysis of singular structures, such as the complex structural behaviour of cable networks with strong non-linearities.
- The definition of the cable network geometry is very cumbersome using a finite elements analysis program. For this reason, a three-dimensional parameter design program was used in order to design the turn appropriately as well as the entire assembly of the cable network.

- We have compared the numerical results with the experimental ones. The following aspects are observed:
  - A good agreement between the longitudinal equivalent moduli for both techniques: numerical and experimental.
  - A small deviation in the value of the transversal equivalent moduli, due to the Bauschinger effect [5] in turns, residual stresses, etc.
  - A good performance is observed with respect to the equivalent Poisson's ratios.

Finally, in view of the obtained results for both tests, it can be considered that a numerical simulation like this one can provide accurately results that will help us to understand the behaviour of these stabilization systems. In future works, it would be necessary to take into account other local phenomena in turns such as the effects due to residual stresses, the hysteresis of materials, etc.

**Acknowledgements**

We gratefully acknowledge the financial support provided by the Construction Technology Research Group (GITECO) at the University of Cantabria, Malla Talud Cantabria Ltd. (MTC) and SODERCAN. We thank the Construction and Production Engineering Department of the University of Oviedo for the computational support.

**References**

[1] H.A. BUCHHOLDT, *An introduction to cable roof structures*, Thomas Telford, London, 1999.

[2] S. SHU, B. MUHUNTHAN, T.C. BADGER, R. GRANDORFF, *Load testing of anchors for wire mesh and cable net rockfall slope protection systems*, Engineering Geology **79** (2005) 162-176.

[3] K.J. BATHE, *Finite element procedures*, Prentice Hall, Englewood Cliffs, 1996.

[4] J. STOER, R. BULIRSCH, *Introduction to numerical analysis,* Springer-Verlag, New York, 2004.

[5] J.C. SIMO, T.J.R. HUGHES, *Computational inelasticity*, Springer-Verlag, New York, 1998.

[6] J.N. REDDY, *An introduction to non-linear finite element analysis*, Oxford University Press, New York, 2005.

[7] O.C. ZIENKIEWICZ, R.L TAYLOR, *The finite element method: solid mechanics*, Butterworth-Heinemann, Oxford, 2000.

[8] J.J. DEL COZ DIAZ, P.J. GARCIA NIETO, C. BETEGON BIEMPICA, G. FERNANDEZ ROUGEOT, *Non-linear analysis of unbolted base plates by the FEM and experimental validation*, Thin-walled Structures **44** (2006) 529-541.

[9] D. BRAESS, *Finite elements: theory, fast solvers, and applications in solid mechanics*, Cambridge University Press, Cambridge, 2001.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# Optimal design of robust and efficient complex networks

**R. Criado, J. Pello, M. Romance and M. Vela-Pérez**

*Departamento de Matemática Aplicada, ESCET, Universidad Rey Juan Carlos,*
*C/ Tulipán s/n, 28933, Móstoles, Madrid, Spain*

emails: regino.criado@urjc.es, javier.pello@urjc.es,
miguel.romance@urjc.es, maria.vela@urjc.es

**Abstract**

A relevant problem in network design is how to construct or improve a complex network in order to optimise some structural properties. In this work we study this design problem when we deal with efficiency and vulnerability and we present some characterisation of the extremal networks and improvements. Finally we apply these result to the optimal improvement of the Spanish and German airport networks.

*Key words: Optimal networks, robustness, efficiency, vulnerability, network topology, network design.*
*MSC 2000: 05C75, 05C90, 94C15, 94C30*

## 1 Introduction

In the last years, much progress has been made to describe the complex structure of real world networks [1, 2, 3, 9, 10]. Complex networks have applications in fields ranging from biology (which include issues such as metabolic pathways, genetic regulatory networks or protein folding) to the Internet or the World Wide Web and other technological systems, as well as the study of social or economic relationships, to name a few. Hence a detailed analysis of the underlying network is central for the understanding of the modelled complex system [1, 9]. The investigation of such issues must necessarily embrace a diversity of viewpoints that include different complementary aspects of the network structure.

There are several mathematical parameters that give structural properties of the network, but in this work we will consider the efficiency and vulnerability as main analysis tools. If $G = (V, E)$ is a complex network with $n$ nodes and $m$ links, it is considered that the *performance* of $G$ is a single function $\Phi(G) > 0$ that measures the behaviour of $G$. Some examples of the performance of a network $G$ are the characteristic path length of the network $L(G)$, the mean flow-rate information over $G$, but we will use the efficiency $E^{+}(G)$, defined (see [6, 7]) as

$$E^{+}(G) = \frac{1}{n(n-1)} \sum_{i \neq j \in V} \frac{1}{d_{ij}}, \tag{1}$$

where $d_{ij}$ stands for the shortest distance between the nodes $i$ and $j$. This concept plays the role of measuring its ability for the exchange of information and its response for the spread of perturbations in diverse applications [3, 4].

Another important parameter is the *vulnerability* (as the opposite concept to robustness) which is related with the ability of a network to avoid malfunctioning when a fraction of its constituents is damaged due to random failures or intentional attacks [5, 7]. There are several different approaches in the literature to measure the vulnerability of a complex network [4, 5] but, in general, they can be divided in two types. On the one hand we find the *static vulnerability* which analyses the response of the structural properties of the networks when some of its nodes or links are removed, while, on the other hand, the *dynamical vulnerability* is considered to measure the redistribution of flow in the network when a failure or attack occurs.

In this paper we will consider static vulnerability related to structural properties of the complex network that allows us to spot its critical components in order to improve the security. In [5], an axiomatic description of the robustness is presented and some candidates for vulnerability functions are proposed based on the network regularity. Roughly speaking, a *vulnerability function* is a normalised function $v(G)$ intrinsic to the topology of $G$ that increases if we remove some components of the network. In [5] it is considered that the vulnerability is related to the node regularity and the number of alternative links that can balance a random or intentional attack. The basic idea is that the *more similar* the nodes are, the more robust the network is, assumed that we had fixed the number of links and nodes in the network. Hence, in addition to the number of nodes and links, also the dispersion of the degree distribution should play a central role in the vulnerability of the network, and it was introduced the vulnerability function $V_1(G)$ of a network $G = (V, E)$ with $n$ nodes and $m$ links as

$$V_1(G) = \exp\left(\frac{M-a}{n} + n - m - 2 + \frac{2}{n}\right), \tag{2}$$

where $M = \max\{gr(v_i);\, i \in V\}$, $a = \min\{gr(i);\, i \in V\}$ and $gr(v_i)$ is the degree of node $i \in V$. This definition can be computed easily and gives a good estimation of the robustness of a complex network but the fact that only the nodes of extremal degrees are considered makes it not as sharp as desirable from a statistical point of view. To avoid this problem, a sharper estimator of the regularity of the degree distribution must be considered, leading to the vulnerability function $V_2(G)$ given by

$$V_2(G) = \exp\left(\frac{\sigma}{n} + n - m - 2 + \frac{2}{n}\right), \tag{3}$$

where $n$ is the number of nodes of $G$, $m$ stands for the number of links and $\sigma$ denotes the standard deviation of the degree distribution, i.e.

$$\sigma = \left(\frac{1}{n}\sum_{i\in V}\left(gr(v_i) - \frac{2m}{n}\right)^2\right)^{1/2}. \tag{4}$$

Once that we have fixed some quantitative parameters for the properties of the network, it is natural to ask how a network must be designed or improved to get the best possible result, possibly with a given set of constrains. For example, if we had the chance to reinforce the airline network of a country by adding a new link between two airports, it would be desirable to know which airports we should connect in order to optimize the efficiency of the whole network.

In this work we present some results regarding the optimal design of networks with respect to efficiency and vulnerability and also, how to improve a given network to optimise those parameters. The approaches for optimising vulnerability and efficiency are rather different since the problems have different nature. While we get a complete characterisation of the extremal network for the vulnerability function, the case of efficiency function is much deeper and we present some approximation algorithms that we apply to the optimal improvement of the Spanish and German airport networks.

## 2 Extreme networks for vulnerability and efficiency

In this section we consider the set of all connected networks $G$ with $n$ nodes and $m$ links and we find the extreme graphs for vulnerability and efficiency. That is, we find those networks with maximal vulnerability, with minimal vulnerability and with maximal efficiency for a given number of nodes $n$ and links $m$.

When dealing with a vulnerability function we work with definition $V_2(\cdot)$, since for definition $V_1(\cdot)$ the results are straightforward. On the other hand, for efficiency, we use the additive definition $E^+(\cdot)$ given by Latora and Marchiori (see [6, 7]), but note that since there is a relationship between the different definitions for the efficiency, we can transfer the results from one to the other.

Our first result is about vulnerability and its extreme values. We will use the following inequality.

**Proposition 2.1** *Let $G = (V, E)$ be a network with $n > 1$ nodes and $m$ links and let $gr = (gr(v_1), ..., gr(v_n))$ its degree vector. Then*

$$\sum_{i=1}^{n} gr(v_i)^2 = \|gr\|^2 \geq \frac{4m^2}{n}. \tag{5}$$

Note that we have the equality in (5) for the $K$-regular graphs, simply by the equality case in Cauchy-Schwartz inequality. By using this result we can characterise the minimal vulnerability networks with a given number of nodes and links, as the following result shows.

**Theorem 2.2** *Let $G = (V, E)$ be a network with $n > 1$ nodes and $m$ links and let $gr = (gr(v_1), ..., gr(v_n))$ its degree vector. Then $G$ has the minimum vulnerability if its degree vector is the most parallel to the vector $(1, \ldots, 1)$, that is, $G$ is the closest network to the $K$-regular one with $m$ links. Hence, the degree vector for $G$ is of the form $(a, \ldots, a, b)$, $a, b \leq n-1$, with $\sum_{i=1}^{n} gr(v_i) = 2m$.*

Similarly, by using some geometric arguments in $\mathbb{R}^k$, we have a result that characterises the network with maximum vulnerability.

**Theorem 2.3** *Let $G = (V, E)$ be a network with $n > 1$ nodes and $m$ links and let $gr = (gr(v_1), ..., gr(v_n))$ its degree vector. Then $G$ has the maximum vulnerability if its degree vector is the most parallel to the vector $(n-1, 1, \ldots, 1)$, that is, $G$ is the closest network to the Star. Hence, the degree vector for $G$ is of the form $(n-1, \ldots, n-1, a, 1, \ldots, 1)$, $a \leq n-1$, with $\sum_{i=1}^{n} gr(v_i) = 2m$.*

When dealing with extremal networks for the efficiency function, a first analysis should include the local structure of the graph. By using this approach the degree vector is the natural tool and we show that for the networks having a node with maximum degree there is a simple formula for the efficiency function:

**Theorem 2.4** *Let $G$ be a simple network with $n > 1$ nodes and $m$ links. If there exists a node with maximum degree (i.e. $gr(v_i) = n-1$) then*

$$E^+(G) = \frac{m}{n(n-1)} + \frac{1}{2}.$$

Note that last theorem shows that we have the upper equality in [4, theorem 2.2] for the simple networks with a node with maximum degree. As a consequence we deduce that for a network $G$ (with $n > 1$ nodes) to have maximum efficiency it is enough that $G$ has the $n$-Star as a subgraph or another $K$-complete bipartite subgraph.

We find also an inequality for efficiency in terms of the maximum degree of the network (not necessarily equal to $n-1$).

**Proposition 2.5** *Let $G = (V, E)$ be a a network with $n > 1$ nodes and $m$ links. Let $v_k \in V$ be the node with maximum degree in the network (not necessarily equal to $n-1$), $gr(v_k) = a$. Then $d_{ij} \leq n-a+1$ for every $v_i, v_j \in V$, where $d_{ij}$ is the minimum distance between $v_i$ and $v_j$. Furthermore,*

$$E^+(G) \geq \frac{2m(n-a)}{n(n-1)(n+1-a)} + \frac{1}{n+1-a}.$$

# 3 Optimal improvement of complex network

A major problem in network design is spotting the critical element to be added to a given (real-life) complex network that gets the best possible network for some parameters. By cost restrictions, the elements to be added to the network (nodes or links) are usually limited and hence the problem of finding the optimal improvement of the network is related to a discrete and conditioned critical points problem which is hard to solve by direct methods. In this section we will consider such problems when we want to get an optimal improvement of the network that maximises its robustness (i.e. minimises some vulnerability function) or maximises its performance (i.e. the efficiency function) and we will give some computationally effective criteria to determine these conditioned critical points.

If we consider a complex network $G$ and we want to add a single link $\ell$ such that we get a network $G' = G \cup \{\ell\}$ with minimal vulnerability or maximal efficiency, a first naive approach leads us to compute all possible improvements of type $G \cup \{\ell\}$ and spot the optimal, but in real networks this can be computationally non-effective. Note, for example, that if we want to locate the improvement of a complex network with $n$ nodes which has maximal efficiency, an exhaustive analysis of all possible candidates uses an algorithm of computational complexity of order $n^7$, which is far from being effective when dealing with real networks with thousand (or million) of nodes. Therefore it is necessary to develop new strategies of design that reduce the complexity of locating the critical component to be added in order to get effective tools for the network optimisation.

Locating the critical single link $\ell$ that gets the most robust improvement $G \cup \{\ell\}$ of a network $G$ is related to the degree of the nodes to be linked. If we want to minimise the vulnerability function $V_1(G \cup \{\ell\})$, it is straightforward that the optimal design strategy is to decrease the range of the degree distribution of $G$ by adding a link joining the node of minimal degree with other node which has no maximal degree. If we consider the vulnerability function $V_2(\cdot)$, we could think that the same idea should work, but since this vulnerability function uses the whole degree distribution, it can be checked that this is not the optimal strategy for network improvement. Despite this fact, the optimal computationally effective strategy is also related to the minimal degree of the nodes involved as the following result shows.

**Theorem 3.1** *Let $G = (V, E)$ be a graph and $\ell_0 = \{v_{i_0}, v_{j_0}\}$ such that $v_{i_0}, v_{j_0} \in V$ and $\ell_0 \notin E$. Then the following assertions are equivalent:*

*(i)* $G' = G \cup \{\ell_0\}$ *has minimal vulnerability* $V_2(\cdot)$ *among all improvements* $G \cup \{\ell\}$.

*(ii)* $\gamma(v_{i_0}, v_{j_0}) = \min\{\gamma(v_i, v_j);\ \{v_i, v_j\} \notin E\}$, *where* $\gamma(v_i, v_j) = gr(v_i) + gr(v_j)$.

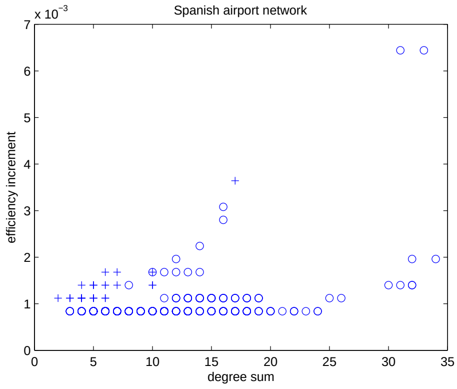
Note that the computational complexity to find the minimum of $\gamma(v_i, v_j)$ directly is of order $n^4$, while the exhaustive computation of the optimal improvement of type $G \cup \{\ell\}$ has complexity $n^5$.

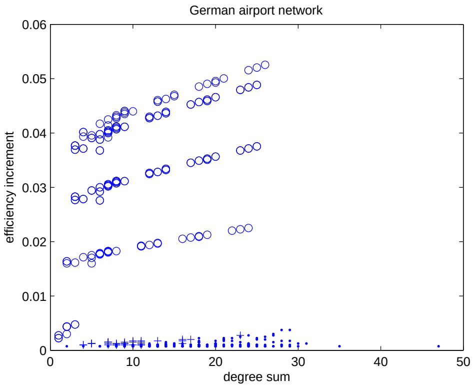
Locating the maximal efficiency improvement of type $G \cup \{\ell\}$ is, by far, a much more complicated problem. In this case, it is clear that the addition of a single link $\ell$ to a network $G$ can produce deep changes in its geodesic structure. We could naively expect that the optimal improvement occurs when we link the most distant nodes, but it is easy to find simple example where this idea fails. Actually, it seems that there is no other clear-enough criterium for locating the improvement of the network with maximal efficiency. As an alternative approach, we propose to give other computationally effective method that gives an approximation of the optimal improvement. If we want to get a near-optimal improvement of type $G \cup \{\ell\}$ we have to mix two different facts:

*(i)* Nodes to be mixed have to be far in order to produce a significant increase in the efficiency.

*(ii)* The new link should produce the biggest change in the geodesic structure of the network. Note that this geodesic sensitivity is related again to the degree of the nodes involved.

However, these conditions do not, by themselves, guarantee that a certain edge will provide the greatest, or close to the greatest, efficiency increase. The following graphs show the increase in efficiency in the Spanish and German airport networks when a single link is added, against the sum of the degrees of the nodes connected. It can be seen that linking nodes already close always has a small impact on efficiency, while picking distant nodes may have a larger effect, but it may as well not be the case.





So, while choosing the two most connected nodes does not always bring the highest efficiency, at least we know we have to take two of the most connected, and preferably distant, nodes. There is a bound to how much the efficiency can increase when adding a single edge, based on their initial distance and their degrees. Therefore, a suitable course of action to find the best edge would be to sort the possible edges to be added according to distance and node degree and run through them in decreasing order, testing the change in efficiency and stopping when the bound ensures that we have already come across the best choice. The graphs above suggest that this best choice will actually be one of the first pairs to be tested.

# References

[1] R. Albert and A. L. Barabási, "Statistical mechanics of complex networks", *Rev. Mod. Phys.* ***74*** (2002), 47–97.

[2] Y. Bar-Yam, *Dynamics of Complex Systems*, Addison-Wesley, 1997.

[3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D.-U. Hwang, "Complex networks: Structure and dynamics", *Physics Reports* ***424*** (2006), 175–308.

[4] R. Criado, A. García del Amo, B. Hernández-Bermejo, M. Romance, "New results on computable efficiency and its stability for complex networks", *Journal of Computational and Applied Mathematics* ***192***, 59 (2006).

[5] R. Criado, J. Flores, B. Hernández-Bermejo, J. Pello, M. Romance, "Effective measurement of network vulnerability under random and intentional attacks", *Journal of Mathematical Modelling and Algorithms* Vol 4, No 3 (2005), 307–316.

[6] V. Latora and M. Marchiori, "Efficient behaviour of small-world networks", *Phys. Rev. Lett.* ***87*** (2001), art. no. 198701.

[7] V. Latora and M. Marchiori, "How the science of complex networks can help developing strategies against terrorism", *Chaos, Solitons and Fractals* ***20*** (2004), 69–75.

[8] V. Latora, M. Marchiori, "Vulnerability and protection of infrastructure networks", Phys. Rev. **E71**(2005), art. no. 015103

[9] M. E. J. Newman, "The structure and function of complex networks", *SIAM Review* ***45*** (2003), 167–256.

[10] S. H. Strogatz, "Exploring complex networks", *Nature* ***410*** (2001), 268–276.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# Energy and Discrepancy as Criteria for Designs for Numerical Computation

**Steven B. Damelin**[1] **and Fred J. Hickernell**[2]

[1] *Department of Mathematical Sciences, Georgia Southern University, Post Office Box 8093, Statesboro, GA 30460-8093*

[2] *Department of Applied Mathematics, Illinois Institute of Technology, 10 W. 32nd Street, E1-208, Chicago, IL 60616*

emails: `damelin@georgiasouthern.edu`, `hickernell@iit.edu`

### Abstract

Criteria for optimally placing points on sets in Euclidean space for numerical computation is a difficult and old problem. Placing points to minimize energy provides a physical model and intuition. Placing points to minimize discrepancy appeals to ideas from goodness-of-fit for distributions. This talk shows that energy and discrepancy are essentially the same concept. The relationship to of energy and discrepancy

*Key words: capacity, cubature, discrepancy, distribution, energy, equilibrium measure, inner product, kernels, invariance, minimizer, norm, numerical integration, positive definite, potential field, Riesz kernel, reproducing Hilbert space, signed measure*

## 1 Introduction

The problem of uniformly distributing points on some compact set in $d \geq 1$ dimensional Euclidean space with positive $d$ dimensional Hausdorf measure, is an interesting and difficult problem. A physically motivated solution is to treat the points as electrostatic charges and place them so that an electrostatic *energy* is minimized. Another approach to spreading points uniformly, developed initially for the $d$-dimensional unit cube, is the *discrepancy* defined by Weyl [15]. The discrepancy measures the sup-norm of the difference between the uniform distribution and the empirical distribution of the points, and is known in the statistics literature as a Kolmogorov-Smirnov statistic [1]. Besides energy and discrepancy, other distance-based measures of even spread include the fill distance (mesh norm, sphere covering radius) and the separation distance (sphere packing distance). See [6, 11, 14] and the references cited therein, for general discussions of these concepts.

Measures of quality placement of points arise from the numerical analysis and statistics literatures, where this is known as the design problem. The JMP statistical package [12] offers minimum energy, minimum discrepancy, and sphere packing designs among its options for space-filling designs. In some cases, these quality measures can be related to energy, discrepancy and the other measures of even spread. For example, it is known that certain minimum energy type points on the unit interval are good for polynomial interpolation, (see [2, 13] and the references cited therein) and that discrepancy provides a tight upper bound on the numerical integration error [10].

The literature surrounding energy, discrepancy and other measures of even spread have developed mostly independently of each other. The purpose of this talk is to make the connection between the two. Specifically, we show that *energy and discrepancy are equivalent* under rather broad conditions. This implies that numerical integration error has a tight bound in terms of energy.

The work presented here arose out of the authors' work on different aspects of energy, discrepancy and numerical integration error [3, 4, 5, 7, 8, 9]. We have observed similar ideas arising in the energy and discrepancy literatures, often using different terminology and notations, and we believe that having these connections made explicit would facilitate a deeper understanding of these concepts and future research.

## 2 Summary of Results

The main points of this talk are summarized in Tables 1-2. These tables list key concepts, and their interpretations in the energy, discrepancy and numerical integration error literatures. In this article, we demonstrate connections between these concepts.

## References

[1] R. B. D'Agostino, M. A. Stephens (eds.), Goodness-of-Fit Techniques, Marcel Dekker, New York, 1986.

[2] S. B. Damelin, Marcinkiewicz-Zygmund inequalities and the numerical approximation of singular integrals for exponential weights: methods,results and open problems, some new, some old, J. Complexity 19 (2003) 406–415.

[3] S. B. Damelin, P. Grabner, Numerical integration, energy and asymptotic equidistributon on the sphere, J. Complexity 19 (231–246).

[4] S. B. Damelin, J. Levesley, X. Sun, Energies, group invariant kernels and numerical integration on compact manifolds, submitted for publication (2006).

[5] S. B. Damelin, J. Levesley, X. Sun, Energy estimates and the Weyl criterion on compact homogeneous manifolds, in: Algorithms and Approximation V, Springer-Verlag, 2007.

Table 1: Outline of the Equivalence Between Energy, Discrepancy, and Numerical Integration Error
$\mathcal{X} \in \mathbb{R}^d$ belongs to a suitable (large) class of measurable sets

| Concept | Energy | Discrepancy | Numerical Integration Error |
|---|---|---|---|
| Symmetric, positive definite *kernel function* $K$ defined on $\mathcal{X}^2$ | $K(\boldsymbol{x},\boldsymbol{y})$ defines energy between two unit charges located at $\boldsymbol{x},\boldsymbol{y} \in \mathcal{X}$ | $K(\boldsymbol{x},\boldsymbol{y})$ defines an inner product between two probability measures with unit probabilities at $\boldsymbol{x}$ and $\boldsymbol{y}$ respectively | $K(\boldsymbol{x},\boldsymbol{y})$ is the *reproducing kernel* for a Hilbert space, $\mathcal{H}(K)$, of integrands, i.e., $K(\cdot,\boldsymbol{y})$ is the representer for function evaluation at $\boldsymbol{y}$ |
| Signed *measure* $\mu : \mathcal{X} \to \mathbb{R}$, $Q(\mu) = \int_{\mathcal{X}} \mathrm{d}\mu(\boldsymbol{x})$; $\mathcal{M}(K)$ a linear space of signed measures | $\mu$ represents a *charge distribution* on $\mathcal{X}$, $Q(\mu)$ is the *total charge* | $\langle \mu, \nu \rangle_{\mathcal{M}(K)}$ $= \int_{\mathcal{X}^2} K(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\mu(\boldsymbol{x})\mathrm{d}\nu(\boldsymbol{y})$, $Q(\mu)$ is the total measure of the domain $\mathcal{X}$ | $\int_{\mathcal{X}} f(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x})$ for $f \in \mathcal{H}(K)$ is a continuous linear functional on $\mathcal{H}(K)$ |
| *Function* $f_\mu : \mathcal{X} \to \mathbb{R}$ given by $f_\mu = \int_{\mathcal{X}} K(\cdot,\boldsymbol{y})\,\mathrm{d}\mu(\boldsymbol{y})$;<br>$\mathcal{H}(K)$ is a linear space of functions | $f_\mu$ represents the *potential field* induced by $\mu$ | | $\langle f_\mu, f_\nu \rangle_{\mathcal{H}(K)}$ $= \int_{\mathcal{X}^2} K(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\mu(\boldsymbol{x})\mathrm{d}\nu(\boldsymbol{y})$ |
| $E(\mu)$ $= \int_{\mathcal{X}^2} K(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\mu(\boldsymbol{x})\mathrm{d}\mu(\boldsymbol{y})$ | $E(\mu)$ represents the *energy* of the charge distribution $\mu$ | $E(\mu) = \Vert\mu\Vert^2_{\mathcal{M}(K)}$ | $E(\mu) = \Vert f_\mu\Vert^2_{\mathcal{H}(K)}$ |
| $D(\nu;\mu) = \int_{\mathcal{X}^2} K(\boldsymbol{x},\boldsymbol{y})$ $\times\, \mathrm{d}(\mu-\nu)(\boldsymbol{x})\mathrm{d}(\mu-\nu)(\boldsymbol{y})$ | $D(\nu;\mu) = \sqrt{E(\mu-\nu)}$ | $D(\nu;\mu)$ is the *discrepancy* of the measure $\nu$ compared to the measure $\mu$ | $D(\nu;\mu) = \sup_{\Vert f\Vert_{\mathcal{H}(K)} \le 1}$ $\big\vert \int_{\mathcal{X}} f(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x})$ $- \int_{\mathcal{X}} f(\boldsymbol{x})\,\mathrm{d}\nu(\boldsymbol{x})\big\vert$ |

Table 2: Outline of the Equivalence Between Energy, Discrepancy, and Numerical Integration Error, continued
$\mathcal{X} \in \mathbb{R}^d$ belongs to a suitable (large) class of measurable sets

| Concept | Energy | Discrepancy | Numerical Integration Error |
|---|---|---|---|
| $\mu_{\mathrm{e},K}$, with normalization $Q(\mu_{\mathrm{e},K}) = 1$, defined by $\int_{\mathcal{X}} K(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\mu_{\mathrm{e},K}(\boldsymbol{y}) = 1/C_K$, where $C_K$ is the *capacity* of $\mathcal{X}$ | $\mu_{\mathrm{e},K}$ is the *equilibrium measure*, which produces a constant potential field over the domain; $Q(\mu) = 1$ implies $E(\mu) = E(\mu_{\mathrm{e},K}) + E(\mu - \mu_{\mathrm{e},K})$ | $\mu_{\mathrm{e},K} \perp \nu$ for any measure $\nu$ with $Q(\nu) = 0$; $C_K \mu_{\mathrm{e},K}$ is the representer of the total charge linear functional $Q(\cdot)$ | If $Q(\nu) = 1$, then $\sup_{\Vert f \Vert_{\mathcal{H}(K)} \le 1} \left\lvert \int_{\mathcal{X}} f(\boldsymbol{x})\,\mathrm{d}\mu_{\mathrm{e},K}(\boldsymbol{x}) - \int_{\mathcal{X}} f(\boldsymbol{x})\,\mathrm{d}\nu(\boldsymbol{x}) \right\rvert = \sqrt{E(\nu) - E(\mu_{\mathrm{e},K})}$ |
| $\mu_{\min,K}$, with normalization $Q(\mu_{\min,K}) = 1$, defined by $\mu_{\min,K} = \operatorname{argmin}_{Q(\mu)=1} E(\mu)$ | $\mu_{\min,K}$ is the energy *minimizer*; when $\mu_{\mathrm{e},K}$ exists, then $\mu_{\min,K} = \mu_{\mathrm{e},K}$ | | [3, 4, 5] establish upper bounds on numerical integration errors (in terms of minimizers) by energies. |
| $\mu_{\mathcal{P}} = \frac{1}{n} \sum_{\boldsymbol{z} \in \mathcal{P}} \delta_{\boldsymbol{z}}$, where $\mathcal{P}$ is a set of $n$ points in $\mathcal{X}$ | $E(\mu_{\mathcal{P}})$ is the energy of a set of point charges with equal charge at each point and total charge of unity | $\mu_{\mathcal{P}}$ is the *empirical distribution*, and $D(\mu_{\mathcal{P}};\mu)$ measures how well $\mu_{\mathcal{P}}$ approximates $\mu$ | $\sup_{\Vert f \Vert_{\mathcal{H}(K)} \le 1} \left\lvert \int_{\mathcal{X}} f(\boldsymbol{x})\,\mathrm{d}\mu(\boldsymbol{x}) - \frac{1}{n} \sum_{\boldsymbol{z} \in \mathcal{P}} f(\boldsymbol{z}) \right\rvert = D(\mu_{\mathcal{P}};\mu)$; $\sup_{\Vert f \Vert_{\mathcal{H}(K)} \le 1} \left\lvert \int_{\mathcal{X}} f(\boldsymbol{x})\,\mathrm{d}\mu_{\mathrm{e},K}(\boldsymbol{x}) - \frac{1}{n} \sum_{\boldsymbol{z} \in \mathcal{P}} f(\boldsymbol{z}) \right\rvert = \sqrt{E(\mu_{\mathcal{P}}) - E(\mu_{\mathrm{e},K})}$ |

[6] Q. Du, V. Faber, M. Gunzburger, Centroidal Voronoi tessellations: applications and algorithms, SIAM Rev. 41 (4) (1999) 637–676.

[7] F. J. Hickernell, A generalized discrepancy and quadrature error bound, Math. Comp. 67 (1998) 299–322.

[8] F. J. Hickernell, Goodness-of-fit statistics, discrepancies and robust designs, Statist. Probab. Lett. 44 (1999) 73–78.

[9] F. J. Hickernell, What affects the accuracy of quasi-Monte Carlo quadrature?, in: H. Niederreiter, J. Spanier (eds.), Monte Carlo and Quasi-Monte Carlo Methods 1998, Springer-Verlag, Berlin, 2000.

[10] H. Niederreiter, Random Number Generation and Quasi-Monte Carlo Methods, CBMS-NSF Regional Conference Series in Applied Mathematics, SIAM, Philadelphia, 1992.

[11] T. J. Santner, B. J. Williams, W. I. Notz, The Design & Analysis of Computer Experiments, Springer-Verlag, New York, 2003.

[12] SAS Institute, JMP 6.0 (2005).

[13] L. N. Trefethen, Spectral Methods in MATLAB, Software, Environments, Tools, SIAM, Philadelphia, 2000.

[14] H. Wendland, Scattered Data Approximation, No. 17 in Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge, 2005.

[15] H. Weyl, Über die Gleichverteilung der Zahlen mod Eins, Math. Ann. 77 (1916) 313–352.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE2007*
*Chicago, 20-23 June 2007*

# Investigation about Fractional Calculus in Financial Markets

**Sergio Adriani David**
*Faculty of  Mathematics,  Pontiff Catholic University of Campinas*
*São Paulo State - Brazil*

email: sergioadriani_david@yahoo.com.br

## Abstract

In the last decade the progress in the areas of chaos and fractals revealed subtle relationships with the fractional calculus. Nonetheless, it is generally excluded from standard courses in mathematics, partly because many mathematicians are unfamiliar with its nature and its applications. One particular purpose of this paper is to discuss the usefulness of fractional-order calculus in financial markets. Even there is an increasing interest in the development of the new paradigm another purpose is to encourage the usage of this mathematical idea in other scientific areas by means of a historical apologia for development of the fractional calculus.

*Key words: Fractional Calculus, Finance*

## 1. Introduction

The theory of fractional calculus goes back to the beginning of the theory of differential calculus but its inherent complexity postponed the application of the associated concepts.

In fact, fractional calculus is a natural extension of the classical mathematics. Since the beginning of the theory of differential and integral calculus, mathematicians such as Euler and Liouville investigated their ideas on the calculation of non-integer order derivatives and integrals. Perhaps, the subject would better be called integration and differentiation of arbitrary order.

In spite of the work that has been done in the area, the application of fractional derivatives and integrals has been scarce until recently. However, in the last years, the advances in the theory of chaos revealed relations with fractional derivatives and integrals, motivating a renewed interest in this field.

The basic aspects of the fractional calculus theory can be addressed in references [22]. In what concerns the application of the concepts can be mentioned research about damping [5,18] , chaos and fractals [19,21].

Regarding the adoption of this concept in other scientific areas I outline that several researchers has been inspired to paying attention to the new possibility.

In special case of the applications involving financial markets only some researchers around the world has been interested by this tool [29].

Therefore, although this work is still giving its first steps and, consequently, many aspects remain to be investigated, I consider that it will be able possible a greater interest in this field.

This paper is divided as follows. In section 2, I outline the origins of fractional calculus to doing permissible an historical apologia for the development. In the section 3 will be presented several approaches of mathematical formulation. Section 4 is devoted to the financial markets application possibility. Finally, in section 5 will be presented the conclusions and outlook

## 2. Origins and Historical Apologia

Leibniz when asked about what if n be ½ in $\frac{d^n y}{d\,x^n}$ said [28]: "Some day it would lead to useful consequences".

In 1730 Euler mentioned interpolating between integral orders of a derivative. In 1812 Laplace defined a fractional derivative by means of an integral and in 1819 there appeared the first discussion of a derivative of fractional order in a calculus written by Lacroix.

Lacroix expressed its nth derivative (for n ≤ m) in terms of Legrende's symbol Γ for the generalized factorial as follow:

$$\frac{d^n y}{d\,x^n} = \frac{m!}{(m-n)!}\,x^{m-n} = \frac{\Gamma(m+1)}{\Gamma(m-n+1)}\,x^{m-n}$$

Thus, starting with the function y = $x^m$ replaced n with ½ and let m = 1, obtaining the derivative or order ½ of the function x :

$$\frac{d^{1/2} y}{d\,x^{1/2}} = \frac{\Gamma(2)}{\Gamma(3/2)}\,x^{1/2} = \frac{2}{\sqrt{\pi}}\sqrt{x}$$

It was Liouville who made the first major study of fractional calculus. The Liouville's first definition of a derivative of arbitrary order ν involved an infinite series. Here the series need to be convergent for some ν. The Liouville's second definition was able to give a fractional derivative of x $^{-a}$ whenever both x and a are positive. Based on the definite integral related to the gamma integral of Euler can be to calculate the integral formula to x$^{-a}$. Note that in the integral

$$\int_0^\infty u^{a-1}\,e^{-xu}\,du$$

if we change the variables t = x u , then

$$\int_0^\infty u^{a-1} e^{-xu}\, du = \int \left(\frac{t}{x}\right)^{a-1} e^{-t}\,\frac{1}{x} dt = \int \frac{t^{a-1}}{x^a\, x^{-1}}\, e^{-t}\,\frac{1}{x} dt = \frac{1}{x^a}\int_0^\infty t^{a-1} e^{-t}\, dt$$

Thus,

$$\int_0^\infty u^{a-1} e^{-xu}\, du = \frac{1}{x^a}\int_0^\infty t^{a-1} e^{-t}\, dt$$

But,

$$\Gamma(a) = \int_0^\infty t^{a-1} e^{-t}\, dt$$

Therefore, this yields the integral formula

$$x^{-a} = \frac{1}{\Gamma(a)} \int_0^\infty u^{a-1} e^{-xu}\, du$$

Consequently, by analogy to equation (1),

$$\frac{d^\nu x^{-a}}{d x^\nu} = \frac{\Gamma(a+\nu)}{\Gamma(a)} x^{-a-\nu} = (-1)^\nu \frac{\Gamma(a+\nu)}{\Gamma(a)} x^{-a-\nu}$$

The $(-1)^\nu$ term in the last equation suggests the need to broaden the theory to include complex numbers.

Indeed in terms of contemporary definitions, the modern theory of fractional calculus is intimately connected with the theory of operators. In classical calculus the symbol $D_x^n$ is often used fr the nth derivative operador ( for n ≥ 0) while, less commonly, $D_x^{-1}$ is used for the antiderivative ( or integral ) operador.

A convenient notation described by Davis [28] was the following: if υ is a positive real number, ${}_cD_x^\nu\ f(x)$ denotes differentiation of order υ of the function f along the x-axis. Similarly, the operator ${}_cD_x^{-\nu}\ f(x)$ will denote integration of order υ of the function f along the x-axis.

In fractional calculus there is no geometric interpretation of integration or differentiation of arbitrary order. Because this, the subscripts c and x are here called terminals of integration instead of limits of integration. This avoid unnecessary confusion.

In 1884 Laurent published what is now recognized as the definitive paper in the foundation of fractional calculus. By means the Cauchy’s integral formula for complex valued analytic functions and a simple change of notation to employ a positive υ rather than a negative υ will now yield Laurent's definition of integration of arbitrary order υ>0 :

$$ {}_cD_x^{-\nu} \ f(x) = \frac{1}{\Gamma(\nu)} \int_c^x (x-t)^{\nu-1} \ f(t)dt $$

The appropriatte definition of differentiation of arbitrary order is to integrate up to a point from which the desired result can be obtained by convencional differentiation.
Let be $\nu = m - \rho$ where for conveniente, m is taken to be the least integer greater than υ and 0 < ρ ≤ 1.
Observe that,

$$ {}_cD_x^{\nu} \ f(x) = {}_cD_x^{m-\rho} \ f(x) $$

Thus,

$$ {}_cD_x^{m-\rho} \ f(x) = \frac{d^m}{dx^m}\left[{}_cD_x^{-\rho} \ f(x)\right] = \frac{d^m}{dx^m}\left[\frac{1}{\Gamma(\rho)} \int_0^x (x-t)^{\rho-1} \ f(t)dt\right] $$

Indeed, since the foundation of the differential and integral calculus the generalization of the concept of derivative and integral to a non-integer order has been the subject of several approaches. Due to this reason there are various definitions which are proved to be equivalent and their use should be encouraged by the researchers in different scientific areas. They will be presented in the following section.
Besides, to bear in mind this purpose I suggest in this paper an investigation involving an application in financial market using the fraccional calculus idea. The suggestion will be presented in the section 4.

## 3. Mathematical Formulation - Different Approaches

It is clear that many mathematicians contributed to the history of fractional calculus and were trying to solve a fundamental problem as well as they understood.
Each researcher due to this reason looking for a definition and therefore different approaches to lead to various definitions of differentiation and antidifferentiation of non-integer orders which are proved to be equivalent. Some definitions of derivatives can be summarized and will be shown in Table 1.
Equally a table can be written for different and equivalent integrals definition.

**Table 1:** Definition of derivatives of arbitrary order

| | |
|---|---|
| Lacroix | $\frac{d^n y}{d x^n} = \frac{\Gamma(m+1)}{\Gamma(m-n+1)} x^{m-n}$ |
| Liouville | $\frac{d^\nu x^{-a}}{d x^\nu} = (-1)^\nu \frac{\Gamma(a+\nu)}{\Gamma(a)} x^{-a-\nu}$ |
| Laurent | ${}_cD_x^\nu\ f(x) = {}_cD_x^{m-\rho}\ f(x) = \frac{d^m}{dx^m}\left[\frac{1}{\Gamma(\rho)}\int_0^x (x-t)^{\rho-1}\ f(t)dt\right]$ |
| Hadamard | $D_x^\nu\ f(x) = \left[\frac{\nu}{\Gamma(1-\nu)}\int_0^x \frac{f(x)-f(t)}{t(\ln(x/t))^{1+\nu}}dt\right]$ |
| Chen | $D_x^\nu\ f(x) = \frac{1}{\Gamma(1-\nu)}\frac{d^m}{dx^m}\left[\int_0^x (x-t)^{-\nu}\ f(t)dt\right]$ |
| Marchaud | $D_x^\nu\ f(x) = \left[\frac{\nu}{\Gamma(1-\nu)}\int_{-\infty}^x \frac{f(x)-f(t)}{(x-t)^{1+\nu}}dt\right]$ |

Likewise other researchers have been important and contributed with owner approach and definition as well as Fourier, Laplace, Riemann, Grunwald-Letnikov.
Although all this can be equivalent definitions , from the one specific point of view, that is to say for a specific application some definitions seem more attractive.
The problem of to investigate an application in finance will be the matter of the next section.

## 4. Application in Finance

The problem of estimating capital asset price volatility is good for risk management. The dynamics in financial markets demand full time complete and more accurate modelling.
Although a good number of works has been published about financial markets, most of do not use the fractional calculus like tool. [1,3,4,6,7,14,15,16,32]. Nevertheless, some investigation in finance using fractional diffusion equation has been studied by Scalas et al [23].
In this section , I propose a simple model based on the following discussion: It is well-know that commonly occur a significant change in capital flow when the investors has at any moment a minimum risk perception changes.
In this way it seems to me that is reasonable to imagine that the capital flow invested, denoted $\frac{d\lambda}{dt}$ ,can be proportional to this risk perception changes denoted $(y-y_0)$.

Mathematically,

$$\left(\frac{d\lambda}{dt}\right)^2 \propto (y - y_0)$$

Can be written,

$$\left(\frac{d\lambda}{dt}\right)^2 = -C\,(y - y_0)$$

Note that the an increasing in the risk perception can to stimulate a reduction in the capital injection. This is the meaning of minus signal in above equation. Thus,

$$\frac{d\lambda}{dt} = \sqrt{C}\,\sqrt{(y_0 - y)}$$

and

$$\frac{d\lambda}{\sqrt{(y_0 - y)}} = \sqrt{C}\,dt$$

Now, integrating both sides,

$$\sqrt{C}\int_0^T dt = \int_0^{y_0} (y_0 - y)^{-1/2}\, d\lambda$$

or,

$$K = \int_0^{y_0} (y_0 - y)^{-1/2}\, d\lambda$$

where $K = \sqrt{C}\ T$

Here we can to observe that $\lambda = F(y)$ where $\lambda$ is the amount of capital of return and $y$ represent the risk perception.
To bear in mind this fact, we can notice $d\lambda = F^{'}(y)\,dy$
If we change variables $y_0$ and $y$ to $x$ and t, and replace $F^{'}$ by $f$ the integral equation becomes

$$K = \int_0^x (x - t)^{-1/2}\ f(t)dt$$

From now the problem is to determine the function $f$. This can be done multiplying the last equation by $1/\Gamma(1/2)$ in order to obtain

$$\frac{K}{\Gamma(1/2)} = \frac{1}{\Gamma(1/2)}\int_0^x (x - t)^{-1/2}\ f(t)dt = {}_0D_x^{-1/2}\ f(x)$$

Consequently,

$${}_0D_x^{1/2}\,K = \sqrt{\pi}\;\; f(x) \qquad (*)$$

I outline that of the general Laurent’s definition of derivative let us now consider the derivative of order $1/2$ of the constant $K$. Using $D^{\nu} = D^{m-\rho}$ we can notice

$${}_0D_x^{\nu}\,K = \frac{d^m}{dx^m}\left[\frac{1}{\Gamma(\rho)}\int_0^x (x-t)^{\rho-1}\,K\,dt\right] = \frac{d^m}{dx^m}\left[\frac{K}{\Gamma(\rho)}\frac{x^{\rho}}{\rho}\right] = \frac{K\;p!\,x^{\rho-m}}{(\rho-m)!\,\rho\,\Gamma(\rho)} = \frac{K}{\Gamma(1-\nu)}\,x^{-\nu}$$

In particular case where $\nu = 1/2$ we haven,

$${}_0D_x^{1/2}\,K = \frac{K}{\Gamma(1/2)}\,x^{-1/2} = \frac{K}{\sqrt{\pi}}\,x^{-1/2} \qquad (**)$$

Therefore, from (*) and (**) can be concluded

$$f(x) = \frac{K}{\pi\sqrt{x}}$$

This curious results show similarity with the solution to the tautochrone problem solved by Abel [28].
Of course , there are limitations to the model proposed. For instance, I am essentially considering the single asset in isolation.
The estimation of other unknown parameters is not treated here
Perhaps the quadratic order for $\frac{d\,\lambda}{dt}$ could be better adjusted.
This work is still living its first steps and consequently, many aspects remain to be investigated.

## 5. Conclusions

The recent progress in the area of chaos reveals promising aspects for future developments and application of the theory of fractional calculus in various scientific areas. In this paper the treatment of fractional calculus has been suggestive rather than rigorous in order to rescue the interest to the reader and at the same time to provide a hint of its potential in many scientific areas. In special case of the financial markets some preliminary works has been proposed. A simple model system involving risk and capital return based on the fractional order concepts are simple to implement and reveal a good alternative way of prediction in financial markets.

Besides, in accordance to the nature of this conference this paper have also contributed to the interdisciplinary feature involving fractional calculus applied in finance systems by means of modeling, presentation and discussion about capital flow when the investors has at any moment a minimum risk perception changes.

## References

[1] ] Akdeniz, L., Dechert, W. D., "Risk and Return in a Dynamic Asset Pricing Model", In Proc: Second International Conference on Computing in Economics and Finance. Geneva, Switzerland, 1996

[2] Anastasio, T. J. , “The Fractional-Order Dynamics of Brainstem Vestibulo-oculomotor Neurons”, Biological Cybernetics, vol. 72, pp. 69-79 , 1994.

[3] Anderson, E. W., Hansen, L. P., "Perturbation Methods for Risk-Sensitive Economies", Second International Conference on Computing in Economics and Finance. Geneva, Switzerland, 1996.

[4 ] Arcoverde, G. Lins , "Uma nota sobre o procedimento de Mapeamentos em Vértices nos modelos de cálculo do VaR de instrumentos de renda fixa”, 1999. Artigo downloadable de : http://www.risktech.com.br/

[5] Bagley, R.L , Torvik, P.J. , “On the Fractional Calculus Model of Viscoelastic Behaviour” Journal of Rheology, vol. 3-, no. 1, pp. 133-135, 1986.

[6] Clark, P. K. , “A subordinated stochastic process model with finite variance for speculative prices”, Econometrica, vol 41, pp. 135-156, 1973.

[7] Correia, M.M.R.L., “Memória longa, agrupamento de valores extremos e assimetria em séries financeiras”. Dissertação de mestrado. Universidade de São Paulo – USP – 1997.

[8] Correia, M.M.R.L, Pereira, P.L.V. “Modelos não lineares em finanças: previsibilidade em mercados financeiros e aplicações à gestão de risco”. Annals of the XXI Brazilian Econometric Meeting, Vol 1, p. 240-59, Brazilian Econometric Society: Vitória, Espírito Santo. 1998.

[9] David, S.A., Rosário, J.M., Machado, J. “Chaotic Vibrations of a beam induced by magnetoelastic interactions”. VII International Conference on Computational Plasticity – Barcelona – Espanha ( 07/04/2003 a 10/04/2003).

[10] David, S.A, Rosário, J. M., Machado, J.: “On Numerical Simulations of a Magnetoelastic Dynamical Systems”, Nonlinear Dynamics, Chaos, Control and their Applications to Engineering Sciences, Chapter 2: Nonlinear Dynamics, Vol 5. Chaos Control and Times Series, Published by ABCM, AAM, SBMAC and SIMAI, pp. 157-166,ISBN: 85-900351-5-8, (2002).

[11] David, S.A., Rosário, J.M.: “Investigation about Nolinearities in a Robot with Elastic Members”, Computational Fluid and Solid Mechanics, Solids and Structures, vol.1, K.J.Bathe editor, pp. 137-139, Elsevier Science Ltd., England, ISBN 0-08-043944-6. (2001)

[12] David, S.A., Rosário, J.M., Machado, J.: “Investigation about Chaos in a Magneto-elastic Dynamical System”, Computational Fluid and Solid Mechanics, Multi-Physics, vol.2, K.J.Bathe editor, pp. 1120-1123, Elsevier Science Ltd., England, ISBN 0-08-043944-6. (2001)

[13] David, S.A.; Rosário, J.M. (1999): “Modeling, Simulation and Control of Flexible Robots” Nonlinear Dynamics, Chaos, Control and Their Applications to Engineering Sciences, Chapter 3: Control, Robotics, Neural Networks and Optimization Engineering, vol. 1, Published by ABCM and AAM, pp. 353-358, ISBN: 85-900351-2-3, (1999).

[14] Daykin, C. D. , "Practical Risk Theory for Actuaries", (1994).

[15] Garman, M. B., "Descomposición de los Elementos Componentes del VeR Dentro de una Cartera de Inversión", Financial Engineering Associates, Inc., (1997).

[16] Garman, M. B., Klass, M. J. , “On the Estimation of Security Price Volatility from Historical Data”, Financial Engineering Associates, Inc., (2003).

[17] Gaul, L. , Chen, C. M., “Modeling of Viscoelastic Elastomer Mounts in Multibody Systems”, Advanced Multibody System Dynamics, W. Schielen (ed.) , pp. 257-276, Kluwe Academic Publishers, (1993).

[18] Koeller, R. C. , “Applications of Fractional Calculus to the Theory of Viscoelasticity”, ASME , Journal of Applied Mechanics”, vol. 51, pp. 299 – 307 , June (1984).

[19] Liu, S. H. , “Fractal Model for the ac Response of a Rough Interface”, Physical Review Letters. Vol.55 no. 5, pp. 529-532, July (1985).

[20] Machado, J. A. T. , “System Modeling and Control Trough Fractional-Order Algorithms”, Nonlinear Dynamics, Chaos, Control and Their Applications to Engineering Sciences, vol. 4, pp. 99-115, (2002).

[21] Machado, J. A. T. , “Analysis and Design of Fractional-Order Digital Control systems”, SAMS – Journal Systems Analysis, Modelling, Simulation, vol. 27 , pp. 107-122, (1997).

[22] Machado, J. A. T. , Azenha, A. “Position/Force Fractional Control of Mechanical Manipulators”, IEEE Int. Workshop on Advanced Motion Control, Coimbra, Portugal, (1998).

[23] Mainardi, F. “ Fractional Relaxation-Oscillation and Fractional Diffusion – Wave Phenomena”. Chaos, Solitons & Fractals, vol. 7, no. 9, pp. 1461-1477, (1996).

[24] Mathieu, B., Lay, L. L. , Oustaloup, A. “Identifications of Non-Integer Order Systems in Time Domain”, IEEE-SMC/IMACS Symposium on Control, Optimization and Supervision, pp. 952-956, Lille, France, (1996).

[25] Méhauté, A. L. , “From Dissipative and Non-Dissipative Processes in Fractal Geometry: The Janal”, New Journal of Chemistry, vpl.14, no. 3, pp. 207-215, (1990).

[26] Oustaloup, A. ,”Fractional Order Sinusoidal Oscillators: Optimization and Their Use in Highly Linear FM Modulation”, IEEE Trans. On Cicuits and systems, vol. 28, no. 10, pp. 1007-1009, October (1981).

[27] Ozaktas, H. M. , Arikan, O. , Kutay, M. A. , Bozdagi, G. , “Digital Computation of the Fractional Fourier Transform”, IEEE Trans. On signal Processing, vol. 44, no. 9, pp. 2141-2150, Sept. (1996).

[28] Ross Bertram , “Fractional Calculus”, Mathematics Magazine, vol. 50, no. 3, pp. 115-122, Maio (1977).

[29] Scalas, E. , Gorenflo, R. , Mainardi, F. , “Fractional Calculus and continuous time finance “, Physica A, vol. 284, pp. 376-384 , (2000).

[30] Schor, A ; Bonomo, M & Valls Pereira, P. L. "Arbitrage Pricing Theory (APT) and Macroeconomics Variables: a comparative study for the brazilian stock market". Annals of the XXI Brazilian Econometric Meeting, vol 1, p. 181-200, Brazilian Econometric Society: Vitória, Espírito Santo, (1998).

[31] Souza, L.A.R. & Da Silva, M. E., "Teoria de Valores Extremos para Cálculo de VaR",(1999). Artigo downloadable de : http://www.risktech.com.br/

[32] Ursini, E. L. Girolami, A. David, S.A "Optimal Dynamic Investment Path of the Firm: Planned Rate of Return, Prices (Tariffs) and Impairments due to Technological Obsolescence. In Proc: International Conference on Computational and Mathematical Methods in Science and Engineering - Alicante - Espanha - (20/09/2002 a 25/09/2002)

[33] Ziegelmann, F. A. , Pereira, P.L.V., "Modelos de Volatilidade estocástica com deformação temporal: um estudo empírico para o índice Bovespa", Pesquisa e Planejamento Econômico, 27, 2, p.323-343. (1997).

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# On the optimal approximation for the symmetric Procrustes problems of the matrix equation $AXB = C$

**Yuanbei Deng** [1] **and Daniel Boley**[2]

[1] *College of Mathematics and Econometrics, Hunan University, Changsha,Hunan 410082, P.R.China*

[2] *Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN55455,U.S.A.*

emails: `ybdeng@hnu.cn`, `boley@cs.umn.edu`

**Abstract**

The explicit analytical expressions of the optimal approximation solutions for the symmetric Procrustes problems of the linear matrix equation $AXB = C$ are derived, with the projection theorem in Hilbert space , the quotient singular value decomposition (QSVD) and the canonical correlation decomposition (CCD) being used.

*Key words: Linear matrix equation, least squares problem, optimal approximation , QSVD , CCD*
*MSC 2000: 65F15,65F20*

## 1 Introduction

The least-squares problems of linear matrix equations are called Procrustes problems(cf. Higham,1988 and Andersson and Elfving, 1997). The unconstrained and constrained least squares problems have been of interest for many applications, including particle physics and geology , inverse Sturm-Liouville problem [11], inverse problems of vibration theory [6], control theory, digital image and signal processing, photogrammetry, finite elements, and multidimensional approximation [8]. Penrose(cf. [2], [13] ) first considered the linear matrix equation

$$AX = B \tag{1.1}$$

and obtained its general solution and least-squares solution by making use of the Moore-Penrose generalized inverse, then Sun[14] obtained the least-squares solution and the related optimal approximation solution of Eq. (1.1) when $X$ is a real matrix. When $X$

is constrained to be a real symmetric matrix , the least-squares solution of (1.1) was derived by Higham and Sun respectively in 1988([12] and [15]) , and Sun also obtained the related symmetric optimal approximation solution of Eq. (1.1) in [15].

In this paper , the following linear matrix equation

$$AXB = C \tag{1.2}$$

are considered . Fausett and Fulton[8] and Zha[18] considered the unconstrained least-squares problems of Eq. (1.2), Eric Chu[4] and Dai Hua[5] obtained the general expressions for the symmetric solution of Eq. (1.2) by using the general singular value decomposition of matrices (GSVD), and the symmetric and skew-symmetric least-squares solutions of Eq. (1.2) have been derived by Deng, Hu and Zhang[7]. But it remains unsolved about the optimal approximation solutions for the symmetric and skew-symmetric Procrustes problems of this equation. Therefore in the following, we will consider the optimal approximation solutions of the symmetric least squares problems of Eq. (1.2). We always suppose that $R^{m\times n}$ is the set of all $m \times n$ real matrices, $SR^{n\times n}$ and $OR^{n\times n}$ are the sets of all symmetric and orthogonal matrices in $R^{n\times n}$, respectively, $A * B$ represent the Hadamard product of $A$ and $B$, and $\|Y\|_F$ denotes the Frobenius norm of a real matrix $Y$, defined as

$$\|Y\|_F^2 =< Y, Y >= \sum_{i,j} y_{ij}^2,$$

here the inner product is given by $< A, B >= trace(A^T B)$, and $R^{m\times n}$ become a Hilbert space with the inner product.

**Problem I.** Given matrices $A \in R^{m\times n}$,$B \in R^{n\times p}$,$C \in R^{m\times p}$ and $X_f \in R^{n\times n}$, let

$$S_E = \{X|X \in SR^{n\times n}, \|AXB - C\|_F = min\}. \tag{1.3}$$

Then find $X_e \in S_E$, such that

$$\|X_e - X_f\|_F = \min_{X\in S_E} \|X - X_f\|_F. \tag{1.4}$$

We first introduce some results about the quotient singular value decomposition (QSVD) and the canonical correlation decomposition (CCD)of matrices , as soon as the projection theorem on Hilbert space, which are essential tools for the Problem , see [3],[9], [10] and [16] for details.

The QSVD is a simple form of the GSVD. The QSVD of a pair of matrices $(A, B^T)$ is as follows.

**QSVD** Theorem. *Let $A \in R^{m\times n}, B \in R^{n\times p}$. Then there exist orthogonal matrices $U \in OR^{m\times m}, V \in OR^{p\times p}$ and a nonsingular matrix $Y \in R^{n\times n}$ such that*

$$A = U\Sigma_1 Y^{-1}, \quad B^T = V\Sigma_2 Y^{-1}, \tag{1.5}$$

where

$$\Sigma_1 = \begin{array}{cc} \left(\begin{array}{cccc} I_{r'} & 0 & 0 & 0 \\ 0 & S & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right) & \begin{array}{l} r' \\ s' \\ m-r'-s' \end{array} \\ \begin{array}{cccc} r' & s' & t' & n-k' \end{array} & \end{array}, \tag{1.6}$$

$$\Sigma_2 = \begin{array}{c} \left(\begin{array}{cccc} 0 & 0 & 0 & 0 \\ 0 & I_{s'} & 0 & 0 \\ 0 & 0 & I_{t'} & 0 \end{array}\right) \\ \begin{array}{cccc} r' & s' & t' & n-k' \end{array} \end{array} \begin{array}{l} p+r'-k' \\ s' \\ t' \\ \phantom{x} \end{array} , \tag{1.7}$$

$$k' = rank(A^T, B), r' = k' - rank(B),$$
$$s' = rank(A) + rank(B) - k', S = diag(\sigma_1, \cdots, \sigma_{s'}),$$
$$\sigma_i > 0(i = 1, \cdots, s'), t' = k' - r' - s'.$$

When $A$ and $B^T$ are of full column rank, i.e. $r(B) = r(A) = n$, then $r' = 0, s' = n, k' = n$, and

$$\Sigma_1 = \begin{array}{c} \left(\begin{array}{c} S \\ 0 \end{array}\right) \\ n \end{array} \begin{array}{c} n \\ m-n \\ \phantom{x} \end{array}, \qquad \Sigma_2 = \begin{array}{c} \left(\begin{array}{c} 0 \\ I_{s'} \end{array}\right) \\ n \end{array} \begin{array}{c} p-n \\ n \\ \phantom{x} \end{array}. \tag{1.8}$$

The canonical correlations decomposition of the matrix pair $(A^T, B)$ is given by the following theorem.

**CCD** Theorem. Let$A \in R^{m\times n}, B \in R^{n\times p}$, and assume that $g = rank(A), h = rank(B)$, $g \geq h$. Then there exist a orthogonal matrix $Q \in OR^{n\times n}$ and nonsingular matrices $X_A \in R^{m\times m}, X_B \in R^{p\times p}$ such that

$$A^T = Q[\Sigma_A, 0]X_A^{-1}, \qquad B = Q[\Sigma_B, 0]X_B^{-1}, \tag{1.9}$$

where $\Sigma_A \in R^{n\times g}$ and $\Sigma_B \in R^{n\times h}$ are of the forms:

$$\Sigma_A = \begin{pmatrix} I_i & 0 & 0 \\ 0 & \Lambda_j & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \Delta_j & 0 \\ 0 & 0 & I_t \end{pmatrix}, \qquad \Sigma_B = \begin{pmatrix} I_h \\ 0 \end{pmatrix}, \tag{1.10}$$

with the same row partitioning, and

$$\begin{array}{ll} \Lambda_j = diag(\lambda_{i+1}, \ldots, \lambda_{i+j}), & 1 > \lambda_{i+1} \geq \ldots \geq \lambda_{i+j} > 0, \\ \Delta_j = diag(\delta_{i+1}, \ldots, \delta_{i+j}), & 0 < \delta_{i+1} \leq \ldots \leq \delta_{i+j} < 1, \\ \lambda_{i+1}^2 + \delta_{i+1}^2 = 1, \ldots, \lambda_{i+j}^2 + \delta_{i+j}^2 = 1, & i.e., \Lambda_j^2 + \Delta_j^2 = I, \end{array}$$

Here,

$$\begin{array}{l} i = rank(A) + rank(B) - rank[A^T, B], \\ j = rank[A^T, B] + rank(AB) - rank(A) - rank(B), \\ t = rank(A) - rank(AB), \quad g = i + j + t. \end{array}$$

Following is the projection theorem (cf. [16]).

**Lemma 1.1** *Let $\mathcal{H}$ be a Hilbert space, $\mathcal{M}$ be a subspace of $\mathcal{H}$, and $\mathcal{M}^\perp$ be the orthogonal complement subspace of $\mathcal{M}$. For a given $H \in \mathcal{H}$, if there exists an $M_0 \in \mathcal{M}$ such that $\|H - M_0\| \leq \|H - M\|$ holds for any $M \in \mathcal{M}$, then $M_0$ is unique and $M_0 \in \mathcal{M}$ is the unique minimization vector in $\mathcal{M}$ if and only if $(H - M_0)\perp\mathcal{M}$, i.e.,$(H - M_0) \in \mathcal{M}^\perp$.*

## 2 The main results

In this section, the explicit expression for the solution of Problem I is derived. Without loss of generality, we suppose that $rank(A) \geq rank(B)$.

Instead of considering the solution of Problem I, we will find a matrix $C_0$, and then transform Problem I to the following equivalent problem.

**Problem $I_0$.** Given matrices $A \in R^{m\times n}$,$B \in R^{n\times p}$,$C_0 \in R^{m\times p}$ and $X_f \in R^{n\times n}$, let

$$S_{E_0} = \{X|X \in SR^{n\times n}, AXB = C_0\}. \tag{2.11}$$

Then find $X_e \in S_{E_0}$, such that

$$\|X_e - X_f\|_F = \min_{X\in S_{E_0}} \|X - X_f\|_F. \tag{2.12}$$

First we use the projection theorem on $R^{m\times p}$.

**Theorem 2.1** *Given $A \in R^{m\times n}, B \in R^{n\times p}, C \in R^{m\times p}$, let $X_0$ be one of the symmetric least-squares solutions of the matrix equation (1.2) and define*

$$C_0 = AX_0B, \tag{2.13}$$

*then the matrix equation*

$$AXB = C_0, \tag{2.14}$$

*is consistent in $SR^{n\times n}$, and the symmetric solution set $S_{E_0}$ of the matrix equation (2.13) is the same as the symmetric least-squares solution set $S_E$ of the matrix equation (1.2).*

*Proof.* Let

$$\mathcal{L} = \{Z|Z = AXB, X \in SR^{n\times n}\}. \tag{2.15}$$

Then $\mathcal{L}$ is obviously a linear subspace of $R^{m\times p}$.Because $X_0$ is the symmetric least-squares solutions of the matrix equation (1.2), from (2.13) we see that $C_0 \in \mathcal{L}$ and

$$\|C_0 - C\|_F = \|AX_0B - C\|_F$$

$$= \min_{X\in SR^{n\times n}} \|AXB - C\|_F$$

$$= \min_{Z\in\mathcal{L}} \|Z - C\|_F.$$

Then by Lemma 1.1 we have

$$(C_0 - C)\perp\mathcal{L} \ \ or \ \ (C_0 - C) \in \mathcal{L}^{\perp}.$$

Next for all $X \in SR^{n\times n}$, $AXB - C_0 \in \mathcal{L}$, it then follows that

$$\begin{aligned}&\|AXB - C\|_F^2\\ &= \|(AXB - C_0) + (C_0 - C)\|_F^2\\ &= \|AXB - C_0\|_F^2 + \|C_0 - C\|_F^2.\end{aligned}$$

Hence, $S_E = S_{E_0}$, and the conclusion of the theorem is true. □

Now suppose $A \in R^{m\times n}, B \in R^{n\times p}$ and the matrix pair $(A, B^T)$ has the QSVD (1.5), and partition $U^TCV$ into the following blocks matrix.

$$U^TCV = \begin{array}{cc}\left(\begin{array}{ccc} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33}\end{array}\right) & \begin{array}{l} r' \\ s' \\ m-r'-s' \end{array} \\ \begin{array}{ccc} p+r'-k' & s' & t' \end{array} & \end{array}, \tag{2.16}$$

then the expression of $C_0$ will be shown in the following theorem.

**Theorem 2.2** *Let $A, B, C$ be given in Problem I, the matrix pair $(A, B^T)$ have the QSVD (1.5), and $U^TCV$ be partitioned by (2.16), then for any symmetric least-squares solution $X_0$ of the matrix equation (1.2) the matrix $C_0$ defined by (2.13) can be determined by the following form.*

$$C_0 = UC^*V^T, \quad C^* = \begin{array}{cc}\left(\begin{array}{ccc} 0 & C_{12} & C_{13} \\ 0 & S\hat{X}_{22} & C_{23} \\ 0 & 0 & 0\end{array}\right) & \begin{array}{l} r' \\ s' \\ m-r'-s' \end{array} \\ \begin{array}{ccc} p+r'-k' & s' & t' \end{array} & \end{array}, \tag{2.17}$$

*where*

$$\begin{aligned}&\hat{X}_{22} = \phi * (C_{22}^T S + SC_{22}),\\ &\phi = (\varphi_{kl}) \in SR^{s'\times s'}, \varphi_{kl} = \frac{1}{\sigma_k^2+\sigma_l^2}, 1 \le k, l \le s'.\end{aligned} \tag{2.18}$$

*Proof.* From Theorem 2.1 in [7] we know that the symmetric least-squares solution of the matrix equation (1.2) can be obtained using of the QSVD of matrix pair $(A, B^T)$ and the general form of the solution is

$$X_0 = Y \begin{bmatrix} X'_{11} & C_{12} & C_{13} & X'_{14} \\ C_{12}^T & \hat{X}_{22} & S^{-1}C_{23} & X'_{24} \\ C_{13}^T & (S^{-1}C_{23})^T & X'_{33} & X'_{34} \\ X'^T_{14} & X'^T_{24} & X'^T_{34} & X'_{44}\end{bmatrix} Y^T, \tag{2.19}$$

where $\hat{X}_{22}$ is given by (2.18) and $X'_{11} \in SR^{r'\times r'}, X'_{33} \in SR^{t'\times t'}, X'_{44} \in SR^{(n-k')\times(n-k')}$,

$X'_{14} \in R^{r' \times (n-k')}$, $X'_{24} \in R^{s' \times (n-k')}$, $X'_{34} \in R^{t' \times (n-k')}$ are arbitrary matrix blocks. Substituting (1.5),(2.19) into (2.13), we can easily obtain (2.17). □

Evidently, (2.17) shows that the matrix $C_0$ in theorem 2.2 is dependent only on the matrices $A, B$ and $C$, but is independent on the symmetric least-squares solution $X$ of the matrix equation (1.2). Since $C_0$ is known, from Theorem 2.1 we know that Problem I is equivalent to Problem $I_0$. In Problem $I_0$, since $S_{E_0} \neq \emptyset$, we can derive the general expression of of the elements of $S_{E_0}$ in the following theorem. In this theorem, given $A \in R^{m \times n}, B \in R^{n \times p}$ ,while $C_0$ is given by (2.17),and assume that $g = rank(A), h = rank(B)$, the matrix pair $(A^T, B)$ has CCD (1.9).Notice that we only state the result with $g = h$, because in the case $g > h$, the results of the theorem and process of the proof are similar, only the partitions of the related matrices are more complex.

Suppose $X \in S_{E_0}$, then partition the symmetric matrix $X^* \equiv Q^T X Q$ into blocks matrix,

$$X^* = (X_{kl})_{6 \times 6}, \tag{2.20}$$

with the row numbers (and the related column numbers) of blocks are $i, j, t, n-g-j-t, j, t$ respectively, and $X_{kl} = X_{lk}^T, k, l = 1, 2, \ldots, 6$. Let $E = X_A^T C_0 X_B$ and also partition $E$ into blocks matrix,

$$E = (E_{kl})_{4 \times 4}, \tag{2.21}$$

with the row numbers of blocks are $i, j, t, m-g$ and the column numbers of blocks are $i, j, t, p-g$ respectively.

**Theorem 2.3** *In Problem $I_0$, the general form of the elements of $S_{E_0}$ can be expressed as $X = QX^*Q^T$, where $X^*$ has the form*

$$\begin{pmatrix} E_{11} & E_{12} & E_{13} & X_{14} & X_{51}^{*T} & E_{31}^T \\ E_{12}^T & X_{22} & X_{23} & X_{24} & X_{52}^{*T} & E_{32}^T \\ E_{13}^T & X_{23}^T & X_{33} & X_{34} & X_{53}^{*T} & E_{33}^T \\ X_{14}^T & X_{24}^T & X_{34}^T & X_{44} & X_{45} & X_{46} \\ X_{51}^* & X_{52}^* & X_{53}^* & X_{45}^T & X_{55} & X_{56} \\ E_{31} & E_{32} & E_{33} & X_{46}^T & X_{56}^T & X_{66} \end{pmatrix} \tag{2.22}$$

*where* $X_{51}^* = \Delta_j^{-1}(E_{21} - \Lambda_j E_{12}^T), X_{52}^* = \Delta_j^{-1}(E_{22} - \Lambda_j X_{22}), X_{53}^* = \Delta_j^{-1}(E_{23} - \Lambda_j X_{23})$,*while* $X_{kk} = X_{kk}^T, 2 \leq k \leq 6$, $X_{14}, X_{23}, X_{24}, X_{34}, X_{45}, X_{46}$ *and* $X_{56}$ *are arbitrary matrices with the associated sizes.*

*Proof.* Suppose $X \in S_{E_0}$, then

$$AXB = C_0. \tag{2.23}$$

Substitute (1.9) into (2.23),we have

$$\begin{pmatrix} \Sigma_A^T \\ 0 \end{pmatrix} X^* (\Sigma_B, 0) = E, \tag{2.24}$$

then substitute (1.10),(2.20) and (2.21) into (2.24), it holds

$$\begin{pmatrix} X_{11} & X_{12} & X_{13} & 0 \\ \Lambda_j X_{21} + \Delta_j X_{51} & \Lambda_j X_{22} + \Delta_j X_{52} & \Lambda_j X_{23} + \Delta_j X_{53} & 0 \\ X_{61} & X_{62} & X_{63} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} E_{11} & E_{12} & E_{13} & E_{14} \\ E_{21} & E_{22} & E_{23} & E_{24} \\ E_{31} & E_{32} & E_{33} & E_{34} \\ E_{41} & E_{42} & E_{43} & E_{44} \end{pmatrix} \tag{2.25}$$

Because the matrix equation (2.23) is consistent, therefore we can obtain some $X_{ij}$ from (2.25) directly. Comparing with both sides of (2.25) , the expression (2.22) of $X^*$ can be derived according to the symmetric property of $X^*$. □

The following lemmas are needed for the main results.

**Lemma 2.1** [17] *For given $J_1, J_2, J_3$ and $J_4 \in R^{m\times n}$,*

$$S_a = diag(a_1, \ldots, a_m) > 0, \quad S_b = diag(b_1, \ldots, b_m) > 0,$$
$$S_c = diag(c_1, \ldots, c_m) > 0, \quad S_d = diag(d_1, \ldots, d_m) > 0,$$

*there exists a unique $W \in R^{m\times n}$, such that*

$$\|S_aW - J_1\|_F^2 + \|S_bW - J_2\|_F^2 + \|S_cW - J_3\|_F^2 + \|S_dW - J_4\|_F^2 = min$$

*and $W$ can be expressed as*

$$W = P * (S_aJ_1 + S_bJ_2 + S_cJ_3 + S_dJ_4),$$

*where*

$$P = (p_{kl}) \in R^{m\times n}, p_{kl} = 1/(a_k^2 + b_k^2 + c_k^2 + d_k^2), 1 \le k \le m, 1 \le l \le n.$$

**Lemma 2.2** *For given $J_1, J_2$ and $J_3 \in R^{s\times s}$, $S_a = diag(a_1, \ldots, a_s) > 0$,$S_b = diag(b_1, \ldots,$ $b_s) > 0$, $S_c = diag(c_1, \ldots, c_s) > 0$, there exists a unique symmetric matrix $W \in SR^{s\times s}$, such that*

$$\mu \equiv \|S_aW - J_1\|_F^2 + \|S_bW - J_2\|_F^2 + \|S_cW - J_3\|_F^2 = min,$$

*and $W$ can be expressed as*

$$W = \Phi * (S_aJ_1 + J_1^TS_a + S_bJ_2 + J_2^TS_b + S_cJ_3 + J_3^TS_c), \tag{2.26}$$

*where*

$$\Phi = (\phi_{kl}) \in R^{s\times s}, \phi_{kl} = 1/(a_k^2 + a_l^2 + b_k^2 + b_l^2 + c_k^2 + c_l^2), 1 \le k, l \le s.$$

*Proof.* For $W \in SR^{s\times s}$, it holds $w_{kl} = w_{lk}$ $(1 \le k, l \le s)$, and

$$\mu = \textstyle\sum_{k=1}^s[(a_kw_{kk} - J_{1kk})^2 + (b_kw_{kk} - J_{2kk})^2 + (c_kw_{kk} - J_{3kk})^2]$$
$$+ \textstyle\sum_{1\le k<l\le s}[(a_kw_{kl} - J_{1kl})^2 + (a_lw_{kl} - J_{1lk})^2 + (b_kw_{kl} - J_{2kl})^2$$
$$+(b_lw_{kl} - J_{2lk})^2 + (c_kw_{kl} - J_{3kl})^2 + (c_lw_{kl} - J_{3lk})^2].$$

Since the function $\mu$ is a continuous and differentiable function of $\frac{1}{2}s(s+1)$ variables $w_{kl}$, hence $\mu$ obtains its minimum value at $\{w_{kl}\}$ when $\frac{\partial \mu}{\partial w_{kl}} = 0$, *i.e.*,

$$w_{kl} = \frac{a_k J_{1kl} + a_l J_{1lk} + b_k J_{2kl} + b_l J_{2lk} + c_k J_{3kl} + c_l J_{3lk}}{a_k^2 + a_l^2 + b_k^2 + b_l^2 + c_k^2 + c_l^2}, \quad 1 \le k \le l \le s.$$

Therefore $W$ can be expressed by (2.26). $\square$

Finally we give the the optimal approximation solutions for the symmetric least-squares problems of the linear matrix equation $AXB = C$, and we still suppose that $rank(A) = rank(B)$.

**Theorem 2.4** *Let matrices $A, B, C$ and $X_f$ be given in Problem I, suppose $rank(A) = rank(B)$, partition the matrix $Q^T X_f Q$ into blocks matrix*

$$Q^T X_f Q = (X_{kl}^{(f)})_{6\times 6}, \tag{2.27}$$

*with the same row and column numbers as $X^*$ of (2.20). Then the unique solution $X_e$ of Problem I can be expressed as $X_e = QX_*Q^T$, and $X_*$ is equal to*

$$\begin{pmatrix} E_{11} & E_{12} & E_{13} & \{X_{14}^{(f)}\} & \bar{X}_{51}^T & E_{31}^T \\ E_{12}^T & \bar{X}_{22} & \bar{X}_{23} & \{X_{24}^{(f)}\} & \bar{X}_{52}^T & E_{32}^T \\ E_{13}^T & \bar{X}_{23}^T & \{X_{33}^{(f)}\} & \{X_{34}^{(f)}\} & \bar{X}_{53}^T & E_{33}^T \\ \{X_{41}^{(f)}\} & \{X_{42}^{(f)}\} & \{X_{43}^{(f)}\} & \{X_{44}^{(f)}\} & \{X_{45}^{(f)}\} & \{X_{46}^{(f)}\} \\ \bar{X}_{51} & \bar{X}_{52} & \bar{X}_{53} & \{X_{54}^{(f)}\} & \{X_{55}^{(f)}\} & \{X_{56}^{(f)}\} \\ E_{31} & E_{32} & E_{33} & \{X_{64}^{(f)}\} & \{X_{65}^{(f)}\} & \{X_{66}^{(f)}\} \end{pmatrix} \tag{2.28}$$

*where* $\bar{X}_{51} = \Delta_j^{-1}(E_{21} - \Lambda_j E_{12}^T), \bar{X}_{52} = \Delta_j^{-1}(E_{22} - \Lambda_j \bar{X}_{22}), \bar{X}_{53} = \Delta_j^{-1}(E_{23} - \Lambda_j \bar{X}_{23}),$

$$\{X_{kl}^{(f)}\} = \frac{1}{2}(X_{kl}^{(f)} + X_{lk}^{(f)T}) = \{X_{lk}^{(f)}\}^T,$$

$$\bar{X}_{22} = \Psi * [X_{22}^{(f)} + X_{22}^{(f)T} + \Delta_j^{-1}\Lambda_j(\Delta_j^{-1}E_{22} - X_{25}^{(f)T}) + (\Delta_j^{-1}E_{22} - X_{25}^{(f)T})^T \Lambda_j \Delta_j^{-1}$$

$$+\Delta_j^{-1}\Lambda_j(\Delta_j^{-1}E_{22} - X_{52}^{(f)}) + (\Delta_j^{-1}E_{22} - X_{52}^{(f)})^T \Lambda_j \Delta_j^{-1}],$$

$$\Psi = (\psi_{kl}) \in R^{j\times j}, \quad \psi_{kl} = \frac{1}{2(1 + (\frac{\delta_{i+k}}{\lambda_{i+k}})^2) + (\frac{\delta_{i+l}}{\lambda_{i+l}})^2)}, 1 \le k, l \le j.$$

*and*

$$\bar{X}_{23} = G * [X_{23}^{(f)} + X_{32}^{(f)T} + \Delta_j^{-1}\Lambda_j(\Delta_j^{-1}E_{23} - X_{35}^{(f)T}) + \Delta_j^{-1}\Lambda_j(\Delta_j^{-1}E_{23} - X_{53}^{(f)}),$$

$$G = (g_{kl}) \in R^{i\times t}, \quad g_{kl} = \frac{1}{2}\lambda_{i+k}, 1 \le k, \le i, 1 \le l \le t.$$

*P*roof. Suppose $X \in S_E = S_{E_0}$, by using (2.22) and (2.27), we have

$$\begin{aligned}
&\|X - X_f\|_F^2 = \|X^* - Q^T X_f Q\|_F^2 \\
&= (\|X_{33} - X_{33}^{(f)}\|_F^2) + (\|X_{44} - X_{44}^{(f)}\|_F^2) + (\|X_{55} - X_{55}^{(f)}\|_F^2) + (\|X_{66} - X_{66}^{(f)}\|_F^2) \\
&+(\|X_{14} - X_{14}^{(f)}\|_F^2 + \|X_{14}^T - X_{41}^{(f)}\|_F^2) + (\|X_{24} - X_{24}^{(f)}\|_F^2 + \|X_{24}^T - X_{42}^{(f)}\|_F^2) \\
&+(\|X_{34} - X_{34}^{(f)}\|_F^2 + \|X_{34}^T - X_{43}^{(f)}\|_F^2) + (\|X_{45} - X_{45}^{(f)}\|_F^2 + \|X_{45}^T - X_{54}^{(f)}\|_F^2) \\
&+(\|X_{46} - X_{46}^{(f)}\|_F^2 + \|X_{46}^T - X_{64}^{(f)}\|_F^2) + (\|X_{56} - X_{56}^{(f)}\|_F^2 + \|X_{56}^T - X_{65}^{(f)}\|_F^2) \\
&+(\|X_{22} - X_{22}^{(f)}\|_F^2 + \|(\Delta_j^{-1}(E_{22} - \Lambda_j X_{22}))^T - X_{25}^{(f)}\|_F^2 + \\
&\|\Delta_j^{-1}(E_{22} - \Lambda_j X_{22}) - X_{52}^{(f)}\|_F^2) + (\|X_{23} - X_{23}^{(f)}\|_F^2 + \|X_{23}^T - X_{32}^{(f)}\|_F^2 + \\
&\|(\Delta_j^{-1}(E_{23} - \Lambda_j X_{23}))^T - X_{35}^{(f)}\|_F^2 + \|\Delta_j^{-1}(E_{23} - \Lambda_j X_{23}) - X_{53}^{(f)}\|_F^2) + \alpha_0,
\end{aligned} \tag{2.29}$$

where $\alpha_0$ is a constant.

According to (2.29), $\|X - X_f\|_F^2 = min$ if and only if each of the brackets in (2.29) takes minimum. Notice that $X_{kk} = X_{kk}^T, k = 3, 4, 5, 6$ and by making use of Lemma 2.1 and Lemma 2.2, the results of this theorem can be derived easily. □

**Conclusions**. Using the projection theorem in Hilbert space , the quotient singular value decomposition and the canonical correlation decomposition , we have obtained the explicit analytical expressions of the optimal approximation solutions for the symmetric least-squares problems of the linear matrix equation $AXB = C$. In fact, we have also obtained the explicit analytical expressions of the optimal approximation solutions for the skew-symmetric least-squares problems of the linear matrix equation $AXB = C$ , because of the limitation of the pages, we omit the content here, and we can design new algorithms to solve the large scale least-square problems of linear matrix equation $AXB = C$. These new results have generalized the work of Eric Chu [4], Dai Hua [5], Higham [12] and Sun [15] in some aspects.

## References

[1] L.Andersson and T. Elfving, A constrained Procrustes problem, *SIAM J. Matrix Anal. Appl.* , 18-1(1997), 124-139.

[2] A.Ben-Israel and T.N.E. Greville, Generalized Inverses: Theory and Applications , *Wiley*, New York, 1974.

[3] Delin Chu, Bart De Moor, On a variational formulation of the QSVD and the RSVD, *Linear Algebra Appl.*, 311(2000), 61–78.

[4] King-wah Eric Chu, Symmetric solutions of linear matrix equations by matrix decompositions, *Linear Algebra Appl.*, 119(1989), 35–50.

[5] Dai Hua, On the symmetric solutions of linear matrix equations, *Linear Algebra Appl.* , 131(1990),1-7.

[6] Dai Hua,P.Lancaster, Linear matrix equations from an inverse problem of vibration theory, *Linear Algebra Appl.*, 246(1996),31–47.

[7] Yuanbei Deng, Xiyan Hu and Lei Zhang, Least squares solution of BXA'=T over symmetric, skew-symmetric and positive semi-definite X, *SIAM J. Matrix Anal. Appl.* , 25-2(2003),486-494.

[8] D.W.Fausett and C.T.Fulton,Large least squares problems involving Kronecker products, *SIAM J. Matrix Anal. Appl.*, 15(1994),219–227.

[9] G.H. Golub and C.F. Van Loan, Matrix Computations, 2nd Edition, *Johns Hopkins University Press*, Baltimore, 1989.

[10] G. H. Golub and H. Zha, Perturbation analysis of the canonical correlations of matrix pairs, *Linear Algebra Appl.* , 210 (1994), 3-28.

[11] O.Hald, On discrete and numerical Sturm-Liouville problems, *Ph.D. dissertation* , Dept. Mathematics, New York Univ., New York,1972.

[12] N. J. Higham, The symmetric Procrustes problem, *BIT* , 28(1988), 133-143.

[13] R. A. Penrose, A generalized inverse for matrices, *Proceedings of the Cambridge Philo- sophical Society*, 51(1955), 406-13.

[14] Jiguang Sun, The least-squares solution of a kind of inverse eigenvalue problems, *Math. Numer. Sinica*, 2(1987), 206-216.(in Chinese)

[15] Jiguang Sun, Two kinds of inverse eigenvalue problems for real symmetric matrices, *Math. Numer. Sinica*, 3(1988), 282–290.(in Chinese)

[16] Rishuang Wang, Functional Analysis and Optimization Theory, *Beijing University of Aeronautics and Astronautics Press* , Beijing, 2003 . (in Chinese).

[17] Guiping Xu, Musheng Wei, Daosheng Zheng, On solutions of matrix equation , *Linear Algebra Appl.* , 279(1998),93-109.

[18] Hongyuan Zha, Comments on large least squares problems involving Kronecker products, *SIAM J. Matrix Anal.Appl.* , 16–4(1995),1172.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# A CFL-like Constraint for the Fast Marching Method in Nonhomogeneous Chemical Kinetics

**Ramón Escobedo**[1]

[1] *Departamento de Matemática Aplicada y Ciencias de la Computación,*
*E.T.S.I. de Industriales y de Telecomunicación*
*Universidad de Cantabria*
*Av. de Los Castros s/n, 39005 Santander - Spain*

email: `escobedo@unican.es`

### Abstract

Level sets and fast marching methods are a widely used technique for problems with moving interfaces. Chemical kinetics has been recently added to this family, for the description of reaction paths and chemical waves in homogeneous media, in which the velocity of the interface is described by a given field. A more general framework must consider variable velocities due to inhomogeneities induced by chemical changes. In this case, a constraint must be satisfied for the correct use of fast marching method. We deduce an analytical expression of this constraint when the Godunov scheme is used to solve the Eikonal equation, and we present numerical simulations of a case which must be enforced to obey the constraint.

*Key words: Chemical kinetics, Fast marching method, Godunov scheme*
*MSC 2000: AMS codes (optional)*

## 1 Introduction

Recently the fast marching level set method of J. A. Sethian [1, 2] has been used in chemical kinetics to solve fundamental problems such as simulating chemical waves [3], finding reaction paths [4], calculating reaction trajectories [5] and computing tunneling paths [6], among others, all of them recognized as fundamental challenges in Chemistry.

Fast Marching Methods (FMM) are especially suitable for tracking chemical interfaces whose velocity is defined by a funcion $F(\mathbf{x},t)$, $(\mathbf{x},t) \in I\!R^n \times [0,+\infty)$, which does not change sign, that is, $F(\mathbf{x},t) > 0$ (or $F(\mathbf{x},t) < 0$) for all $(\mathbf{x},t)$. In the cited works, the function $F$ is a given field which remains constant along the problem, assuming an homogeneous or chemically isotropic medium. Unfortunatelly, this is not the general case and the velocity field often depends on the position of the reaction interface itself. Then, $F$ must be calculated by solving a (system of) partial differential equation(s) simultaneously with the interface evolution problem. In this more general framework, the FMM must be coupled to a static chemical problem which gives the instantaneous velocity field and a CFL-like constraint for $F$ must be taken into account.

In inhomogeneous or anisotropic chemical media, the mathematical model usually consists in describing the evolution of a chemical magnitude $u(x, y, t)$ in a 2D or 3D domain $\Omega$. To do this, one must be able to calculate the magnitude $u(x, y, t)$ for a given position of the interface $\Gamma$ at a given time $t^n$, and the velocity of (at least) all the points of $\Gamma$ at time $t^n$. Then the FMM can obtain the new position of the interface after a short interval of time $\Delta t$, and start again for $t^{n+1} = t^n + \Delta t$.

The algorithm consits of a *chemical* part, where the solution $u^n$ and the velocity $F^n$ are calculated in $\Omega^n$ (characterized by $\Gamma^n$, i.e. changing in time) and a *geometric* part, where the geometry is changed by moving the interface to its new position $\Gamma^{n+1}$.

In inhomogeneous problems, the velocity is renewed after each interval of time $\Delta t$ by solving the chemical problem with a fixed interface. The consistency of the algorithm is based on the assumption that $\Delta t$ is short enough to ensure that the velocity field remains almost constant during this interval of time.

Here, we derive an explicit expression which acts as a CFL contraint from the renewal of $F$ and imposes a restriction for the FMM, concretely in the Godunov method, which is the one used in [4–6].

## 2 The Fast Marching Method

### 2.1 Setup of the algorithm

Let $\{(x_i, y_j)\}_{i=1,j=1}^{N_x,N_y}$ be a rectangular space discretization of a two-dimensional domain $\Omega$. At time $t^n$, the chemical interface $\Gamma^n$ is given by a set of $N_l^n$ nodes of $\Omega$:

$$\Gamma^n = \{(x_l^n, y_l^n)\}_{l=1}^{N_l^n}. \tag{1}$$

Assume that $F \geq 0$ (recall that the velocity does not change sign). The idea of the FMM consists in describing the evolution of the interface –the front– by means of an *arrival time* function $\phi(x, y)$ defined as *the time it takes to the interface to arrive to the point* $(x, y) \in \Omega$. At time $t^n$, the interface is given by the points such that

$$\phi^n(x, y) = t^n. \tag{2}$$

Starting from these points, the FMM provides the arrival time for $t \geq t^n$. At the end of the FMM step, the arrival time is $t^{n+1}$ and the position of the interface is given by the points such that $\phi^n(x, y) = t^{n+1}$. In principle, $\phi^n(x, y)$ can be constructed beyond $t^{n+1}$, but its validity is subject to restrictions imposed by $F^n$, especially in inhomogeneous media; in this case, $\phi^n(x, y)$ must be often reconstructed after each $\Delta t$.

Three sets can be defined using the classical notation: (we omit the index $n$)

A $= \{(x, y) \in \Omega / \ \phi(x, y) \leq 0\}$: the front is or has been here;

C $= \{(x, y) \in \Omega / \ \phi(x, y) > 0$ and **at least one** of its neighbors is in A$\}$;

$$\mathrm{F} = \{(x,y) \in \Omega /\ \phi(x,y) > 0 \text{ and } \textbf{none} \text{ of its neighborsis in A}\},$$

where A= Accepted points, C= Close points and F= Far points.

Numerically, the interface at time $t^n + \Delta t$ is given by the points of $\Omega$ such that

$$|\phi^n(x,y) - \Delta t| < \epsilon, \tag{3}$$

where $\epsilon$ is a small tolerance of the width of the front that can be tuned to obtain a "one-point width" front. Then, $\Gamma^n$ is approximated by the set C; see Fig. 1.

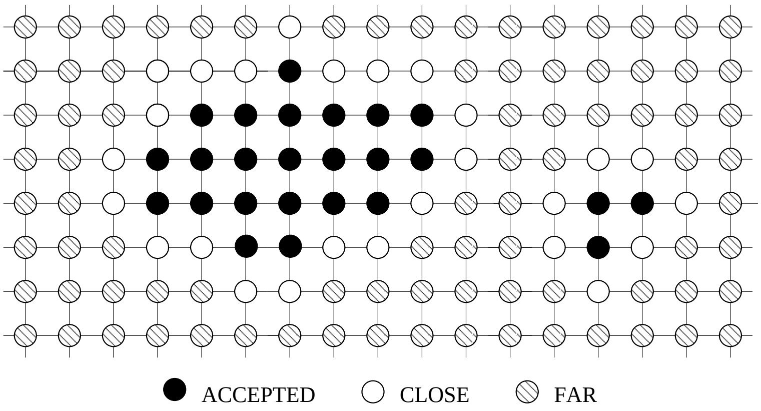


Figure 1: Accepted, Close and Far points for a nonconnected interface.

The level sets formulation consits in identifying the front with the zero level set of a 3D surface $W(x,y,t)$ whose evolution is described by the Hamilton-Jacobi equation

$$\frac{\partial W}{\partial t} + F\,|\nabla W| = 0. \tag{4}$$

By definition, $W(x,y,\phi(x,y)) = 0$. Taking the gradient of this expression yields $\nabla W + W_t|\nabla W| = 0$, and, using (4), we obtain the following equation for $\phi$:

$$\nabla W - F|\nabla W|\nabla\phi = 0. \tag{5}$$

Eq. (5) shows that $\nabla W$ and $\nabla\phi$ are colinear vectors, and that their modules are such that $|\nabla W| = F|\nabla W|\,|\nabla\phi|$. The arrival time function $\phi(x,y)$ is then given by the eikonal equation

$$|\nabla\phi| = \frac{1}{F}. \tag{6}$$

The effectiveness of the FMM lies in the fact that $\phi$ is constructed in the upwind direction (i.e. from low to high values of $\phi$), in order to guarantee the increasing evolution of $\phi$, according to that the time it takes to the front to arrive to a point depends only on the history. The FMM step finishes as soon as a point $(x,y)$ is found such that $\phi(x,y) > \Delta t + \epsilon$. The new front is then given by the new set C, and we return to the chemical problem to obtain the solution and the new velocity at these points.

## 2.2 FMM step: from $t^n$ to $t^{n+1}$:

The FMM first initializes $\phi(x, y)$ and then corrects its values by solving (6) with an iterative process. The initialization must be done at least for the first time step, and can be omitted in successive time steps if no reinitialization of $\phi$ is needed. The constraint we present here is in fact a test which can enforce this reinitialization.

**FMM algorithm**

1.1 At all points of A, set $\phi = 0$.

1.2 At all points of C (the interface), assign to $\phi$ the value of the time obtained by dividing the distance from the point to the set A by the velocity of the interface at this point: $\phi = d/F$ (see later a better way to do this step).

1.3 At all points of F, assign to $\phi$ the value $+\infty$. These points are far and they don't have any influence in the correction of the points of C (*upwind*).

Once $\phi$ is initialized, the *fast marching* starts. The interface evolves point by point by correcting the initial estimation by means of the following iterative process:

2.1 Obtain the point TRIAL $(x_T, y_T)$ from set C which has the smallest value of $\phi$: $(x_T, y_T) \in C$ and $\phi(x, y) \geq \phi(x_T, y_T)$, $\forall (x, y) \in C$. See Fig. 2(a).

2.2 Test:

   i. If $\phi(x_T, y_T) > \Delta t + \epsilon$, $\phi$ has been constructed for all the points verifying (3). Then the FMM step is finished and we return to the chemical problem.

   ii. If not, there still exists points at which the interface will arrive at a time lesser or equal to $\Delta t + \epsilon$ and we have to continue.

2.3 Move $(x_T, y_T)$ to A (and delete it from C); its value is the definitive one because it cannot be improved with this algorithm. See Fig. 2(b).

2.4 Move to C the neighbors of $(x_T, y_T)$ that are in F, because these points have now a neighbor in A. See Fig. 2(c).

2.5 Actualize the value of $\phi$ at all the neighbors of $(x_T, y_T)$ that are in C, by solving Eq. (6). See Fig. 3. This is the *gordian knot* of the FMM; see next section.

2.6 The interface has been moved one point; see Fig. 4. `Goto 2.1`.

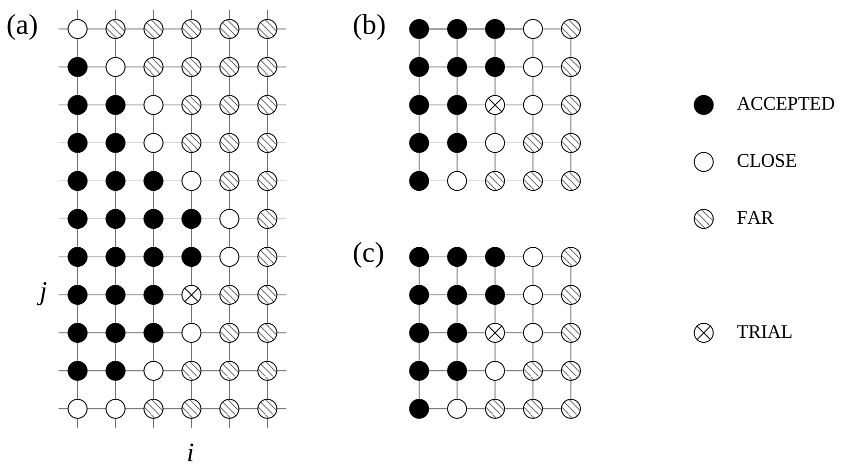


Figure 2: The interface is moved one point: (a) Find the point $(x_T, y_T)$. (b) Move $(x_T, y_T)$ to A. (c) Move the neighbors of $(x_T, y_T)$ from F to C.

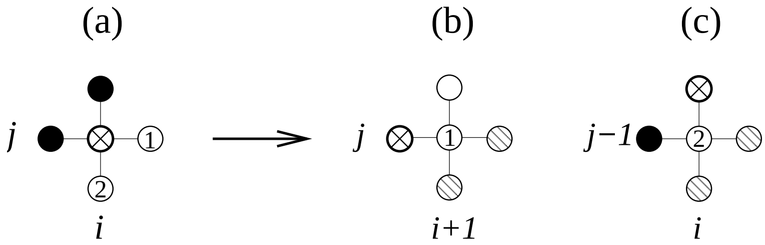


Figure 3: Actualization of $\phi$ with the Godunov scheme in the neighbors of $(x_T, y_T)$ that are in C: (a) the two neighbors; (b) First neighbor of $(x_T, y_T)$ and its respective neighbors: the information comes only from the left. (c) Second neighbor: the information comes in both directions, from the left and from above.

### 2.3 Godunov's method

A suitable way to solve the eikonal equation in the context of the fast marching algortihm is the Godunov method [1], which is precisely the method used in [4–6]. The Godunov method makes use of the following approximation of the gradient:

$$\left[\max\left(D^{-x}_{i,j}\phi,\ -D^{+x}_{i,j}\phi,\ 0\right)^2 + \max\left(D^{-y}_{i,j}\phi,\ -D^{+y}_{i,j}\phi,\ 0\right)^2\right]^{1/2} = \frac{1}{F_{i,j}}, \tag{7}$$

where $D^{-x}_{i,j}\phi = (\phi_{i,j} - \phi_{i-1,j}/\Delta x$ and $D^{+x}_{i,j}\phi = (\phi_{i+1,j} - \phi_{i,j})/\Delta x$. The numerical resolution of Eq. (7) is not trivial because unknown values must be compared *a priori*. Typically, expensive iterative methods are used (Rouy-Tourin, 1992). Fortunately, we can take advance from the upwind character of the Godunov approximation.

Before to explain the step 2.5 of the FMM algorithm, note that the step 1.2 is equivalent to solve (7) with $\phi = 0$ in the points of A and $\phi = +\infty$ in the points of F.

**Step 2.5:** The actualization of $\phi$ at the neighbors of $(x_T, y_T)$ is done by solving Eq. (7) by assigning to $\phi$ the value $+\infty$ at the neighbors [of the neighbor of $(x_T, y_T)$ which is being actualized] that are not in A. Once the calculation is done, the previous values of $\phi$ at the neighbors of the neighbor of $(x_T, y_T)$ must by restituted. The neighbors that are in A are used with their value and they remain unchanged.

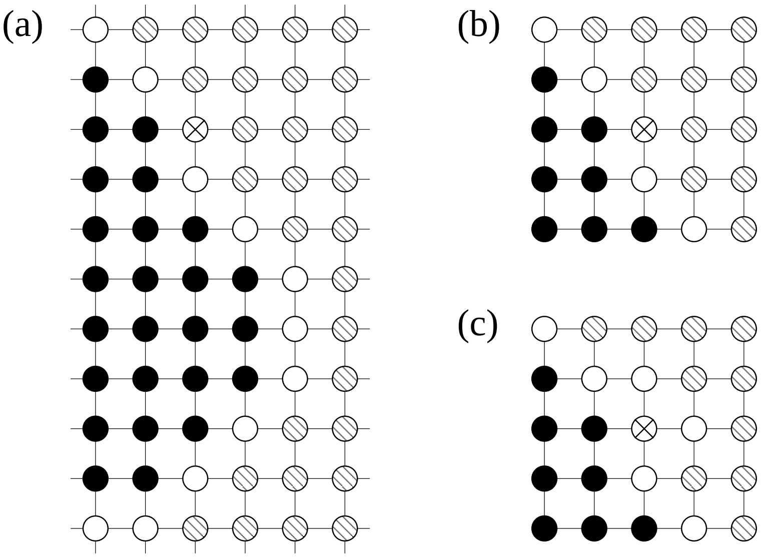


Figure 4: Next FMM step: (a) new position of the interface and localization of the new point $(x_T, y_T)$; (b) and (c): steps 2.3 to 2.5 in this new situation.

How we do this? Assume a uniforme grid: $h = \Delta x = \Delta y$. Then Eq. (7) becomes

$$\max\left(\phi_{i,j} - \phi_{i-1,j},\ \phi_{i,j} - \phi_{i+1,j},\ 0\right)^2 + \max\left(\phi_{i,j} - \phi_{i,j-1},\ \phi_{i,j} - \phi_{i,j+1},\ 0\right)^2 = \frac{h^2}{F_{i,j}^2}. \quad (8)$$

These maxima cannot be calculated without knowing $\phi_{i,j}$, which is precisely the unknown of the equation. Let us replace maxima by minima,

$$\max(\phi_{i,j} - \phi_{i-1,j},\ \phi_{i,j} - \phi_{i+1,j},\ 0) = \phi_{i,j} - \min(\phi_{i-1,j},\ \phi_{i,j},\ \phi_{i+1,j}), \quad (9)$$

$$\max\left(\phi_{i,j} - \phi_{i,j-1},\ \phi_{i,j} - \phi_{i,j+1},\ 0\right) = \phi_{i,j} - \min(\phi_{i,j-1},\ \phi_{i,j},\ \phi_{i,j+1}), \quad (10)$$

and define the values

$$\alpha_1 = \min(\phi_{i-1,j},\ \phi_{i+1,j}) \quad \text{and} \quad \alpha_2 = \min(\phi_{i,j-1},\ \phi_{i,j+1}), \quad (11)$$

which are known values, because they correspond to nodes of A or they have been assigned to $+\infty$ for this calculation. Then Eq. (7) is equivalent to

$$\left(\phi_{i,j} - \min(\phi_{i,j},\ \alpha_1)\right)^2 + \left(\phi_{i,j} - \min(\phi_{i,j},\ \alpha_2)\right)^2 = \frac{h^2}{F_{i,j}^2}. \quad (12)$$

As we are actualizing $\phi_{i,j}$, at least one of the neighbors of $(x_i, y_j)$ must be in A, that is, the information comes from at least one direction $x$ or $y$, or from both directions:

- If the information comes only from direction $x$ (respectively $y$), then we can calculate $\min(\phi_{i,j}, \alpha_2) = \phi_{i,j}$, because $(x_i, y_{j-1})$ and $(x_i, y_{j+1})$ are not in A, so $\phi$ must be at $+\infty$ in these points. The equation to solve is simply

$$\phi_{i,j} = \alpha_1 + \frac{h}{F_{i,j}} \quad \Big(\text{Respectively,}\ \ \phi_{i,j} = \alpha_2 + \frac{h}{F_{i,j}}\Big). \tag{13}$$

- If the information comes fromboth directions, then there exists at least one neighbor in each direction where the value of $\phi$ is lower than $\phi_{i,j}$: $\alpha_{1,2} < +\infty$. This way, $\phi_{i,j}$ can be dropped from minima calculations in (12) and minima calculations are redundant. The Godunov equation becomes $(\phi_{i,j} - \alpha_1)^2 + (\phi_{i,j} - \alpha_2)^2 = h^2/F_{i,j}^2$. This can be rewritten as

$$\phi_{i,j}^2 - (\alpha_1 + \alpha_2)\phi_{i,j} + \frac{1}{2}\left(\alpha_1^2 + \alpha_2^2 - \frac{h^2}{F_{i,j}^2}\right) = 0. \tag{14}$$

  Here $\Delta = -(\alpha_1 - \alpha_2)^2 + 2h^2/F_{i,j}^2$ cannot be negative, so the velocity must be such that $|\alpha_1 - \alpha_2| F_{i,j} \le \sqrt{2}h$. This is called a “classically allowed point” in [3]. When $\Delta > 0$, the higher root must be used to preserve the upwind.

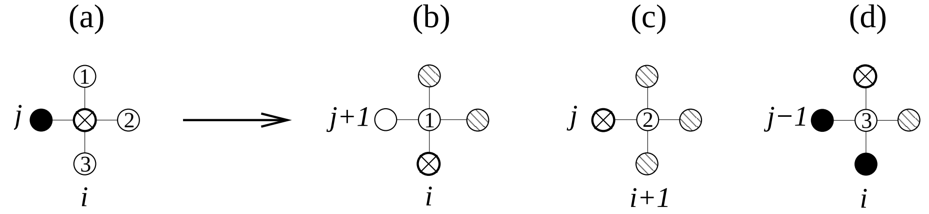


Figure 5: Actualization of the neighbors of the new point $(x_T, y_T)$. In this case there are three neighbors to be actualized, each one with a different kind of neighborhood.

For the consistency of the FMM algorithm, a second upwind condition must be verified by Eq. (14), which is that $\phi_{i,j}$ must be greater or equal than $\alpha_1$ and $\alpha_2$:

$$\phi_{i,j} = \frac{\alpha_1 + \alpha_2}{2} + \frac{1}{2}\sqrt{\Delta} \ > \ \alpha_{1,2}, \quad \text{i.e.} \quad \Delta \ > \ 4\left[\alpha_{1,2} - \frac{\alpha_1 + \alpha_2}{2}\right]^2 = (\alpha_1 - \alpha_2)^2,$$

and then

$$F_{i,j} \ < \ \frac{h}{|\alpha_1 - \alpha_2|}. \tag{15}$$

This condition is more restrictive than the one deduced for “classically allowed points”, and is consequently the one which must be used.

## 2.4 The Condition (15) is a constraint

By construction, the FMM is such that expression (15) is always verified during the geometric part of the problem. Let us give a qualitative description of this fact.

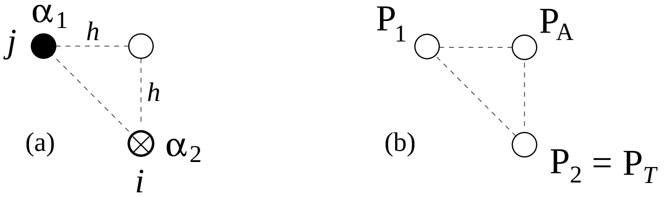


Figure 6: Numerical cell in the actualization of $P_A = (x_i, y_j)$.

Fig. 6 shows the numerical cell of a point $P_T = (x_T, y_T)$ whose upper neighbor $P_A = (x_i, y_j)$ is being updated. Assume that information comes in both directions, from the left from $P_1 = (x_{i-1}, y_j)$ and from below from $P_2 = (x_i, y_{j-1})$. The value of $\phi$ at $P_A$ will depend on $\phi(P_1)$ and $\phi(P_2)$. The trial point $P_T$ must be $P_1$ or $P_2$; assume $P_T = P_2$.

In this case, $\alpha_1 = \phi_{i-1,j}$ and $\alpha_2 = \phi_{i,j-1}$. The value $\alpha_2 - \alpha_1$ (which is positive, because the front arrives first to $P_1$) is the time elapsed between the arrivals of the front to $P_1$ and to $P_2$: $\alpha_2 - \alpha_1$ *is the time spent by the front in going from* $P_1$ *to* $P_2$. Note that the interface can be a not connected set, in such a way that it can arrive to $P_2$ from far from $P_1$. For simplicity, assume that it is a connected set.

In terms of velocities, the value $h/(\alpha_2 - \alpha_1)$ is the velocity at which the interface covers the distance between $P_{1,2}$ and $P_A$ in a time $\alpha_2 - \alpha_1$. The restriction (15) means that $F_{i,j}$ must be lower than this value. This, of course, makes sense, because $F_{i,j}$ is the value of the velocity of the interface at $P_A$. Numerically, $F_{i,j}$ is the value used to actualize $\phi_{i,j}$, that is, the value of $F$ in the numerical cell of $P_A$. It is then necessary to $F_{i,j}$ to be lower than the velocity at which the interface goes from $P_1$ to $P_2$; if not, once in $P_1$, the interface will arrive first to $P_A$ and then to $P_2$, and this is not possible.

In terms of times, the time it takes to the interface to arrive to $(x_i, y_j)$ coming from its neighbors (located at a distance $h$) at a velocity $F_{i,j}$ must be greater or equal than the time spent in going from one neighbor to the other, $|\alpha_1 - \alpha_2|$. If this is not so, the interface will arrive first to the point that is being actualized than to a neighbor which is being used for this actualization, and this is an absurdity.

As we have seen, the FMM preserves the upwind during the geometric part of the numerical resolution. But, during the chemical part of the problem, the velocity is renewed and the expression (15) can be no longer satisfied. Let us show how.

The function $F^n$ is defined as the velocity at the points of $\Gamma^n$ at time $t^n$; nothing is said about the value of $F^n$ outside the interface, and it may happen (especially in inhomogeneous problems) that a *velocity* does not make sense outside the interface. However, these values are used in the FMM, by extending $F^n$ to a narrow band around $\Gamma^n$ (several methods can be used [7], e.g. the velocity at the nearest point of $\Gamma^n$). When the FMM step has been done, and before the renewal of $F^n$, the interface is given by a new set of points

$$\Gamma^{n+1} = \{(x_l^{n+1}, y_l^{n+1})\}_{l=1}^{N_l^{n+1}}.$$

The velocity at these points at time $t^n$ is called, actually, $F_{\text{ext}}^n$, an extention of $F^n$. The condition (15) is still preserved for $F_{\text{ext}}^n$, but this can be false for the renewed velocity $F^{n+1}(\Gamma^{n+1})$. In fact, the velocity at $\Gamma^{n+1}$ can be very different, depending on the smoothness of the chemical processes: the velocity is not constant between time-steps of size $\Delta t$, and important changes may happen from one interval of time to the other which can affect considerably the velocity:

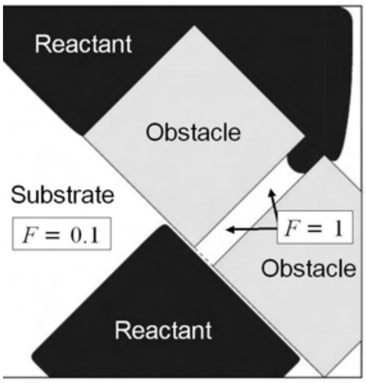
$$F^{n+1}(\Gamma^{n+1}) \neq F_{\text{ext}}^n(\Gamma^{n+1}).$$

From the chemical point of view, the velocity of the interface at a given point is unique (recall that $F$ does not change sign), but the numerical resolution makes use of two values of the velocity at the points at which the arrival of the interface coincides with the change of time-step: the extended value $F_{\text{ext}}^n$ and the renewed value $F^{n+1}$.

The condition (15) can be not satisfied by $F^{n+1}$; there is no reason why $F^{n+1}$ could be enforced to verify (15). In the cited works it is mentioned that "nonclassical points" are simply deleted from the close or trial sets [4]. This may affect to the fidelity of the simulation to the real chemical problem. Both the existence of real solutions of Eq. (7) and the upwind condition (15) must be ensured by the algorithm.

It is in this way that the condition (15) acts as a constraint for $\phi$. In fact, in the algorithm we have proposed here, this condition must be introduced as a test for the validity of $\phi$ for the next FMM step. If the condition is not satisfied, then the reinitialization of $\phi$ is needed, or, in a more complicated way, an adaptive time-step algorithm must be implemented. The refinement of the spatial mesh makes things smoother, but it also contributes to render more demanding the condition (15).

## 3 A numerical example



We have considered a general situation in which a chemical reaction takes place in a domain characterized by the geometry described in the contiguous figure.

Assume that the chemical reactant interface arrives to a region in which the substrate is chemically inhomogeneous. Inhomogeneities can be due to different properties of the substrate, but also to dynamical effects induced by the reaction (electromagnetic effects, temperature...). The reactant can then evolve with a different speed in different parts of the substrate. Assume that the channel (with two entrances, A and B) described by two nonreactive and noncatalytic obstacles, is a fast reactant region ($F_{\text{ch}} = 1$), compared to the rest of the substrate ($F_{\text{s}} = 0.1$). This figure shows the (frequent) case in which the interface arrives first to one of the entrances (e.g., A).

Inside the channel, the advance of the interface is faster than outside. Fig. 7 shows that the arrival time at the most advanced point of the interface inside the channel is lower than at the entrance B; see the two isolated points at (15,4.5,0.4).

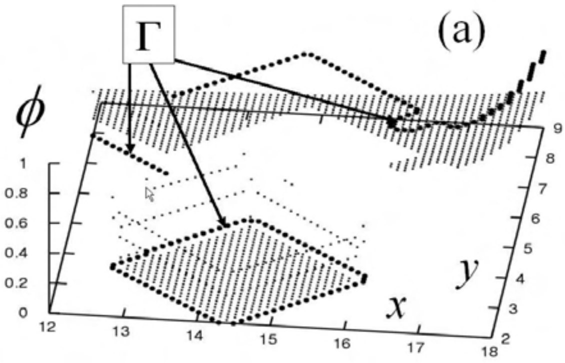


Figure 7: The not-connected interface Γ when just entering into the channel ($F_{\mathrm{ch}} = 1$), and at the other end, outside the channel ($F_{\mathrm{s}} = 0.1$). Depicted is arrival time $\phi(x, y)$. Narrow points are *accepted* points, boldface points are *close points* (which define the interface, and in which the value of $\phi$ is provisional at this stage of the algorithm).

During the next FMM step, the interface will arrive approximately at the same time to the entrance B from inside and outside of the channel. Level sets and fast marching methods are especially suitable for this kind of situation in which topologycal changes may appear during the evolution of the interface. This is a typical situation which can be produced e.g. by four-well potentials as in [4] (see Fig. 1 therein).

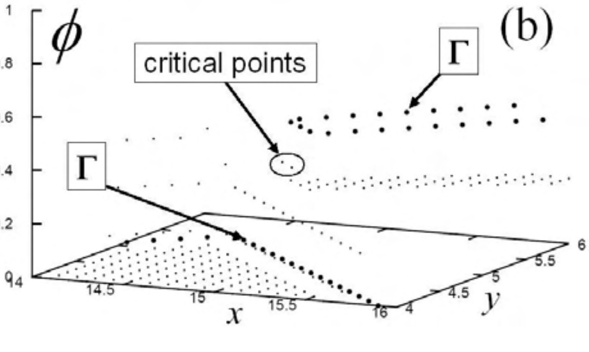


Figure 8: Same as before, after a FMM step. The (definitive) arrival time at critical points near (15,4.5) is larger than inside the channel, producing a large value of $|\alpha_1 - \alpha_2|$.

Fig. 8 shows the precise instant in which both fronts of the interface encounter each other. The interface Γ is given by two nonconnected sets of points (at least in the region we are focusing on). The actualization of $\phi$ (step 2.5 of the FMM) may require the use of close points located at different regions of the substrate, i.e. in regions of different velocities: outside the channel, the interface is just arriving (high arrival time), whereas inside the channel the interface has a low arrival time. Then $|\alpha_1 - \alpha_2|$ becomes large with respect to its typical value when points are all in the same region. When the velocity is renewed, chemical processes can be such that $F$ can adopt a large value; the combination of these two effects can lead to a violation of the constraint.

It is important to note that even if the constraint (15) can be seen as a CFL-like condition, where the value of a parameter ($F$) is restricted by a *space over time* fraction, it is not a CFL condition. Even if the space value is $h$ (the space step), the time value $|\alpha_1 - \alpha_2|$ is absolutely not related to the time step of the algorithm.

## Conclusion

In conclusion, we have derived a constraint for the Godunov scheme when it is used to solve the Eikonal equation during the FMM, when this later is coupled to a chemical problem in inhomogeneous media producing nonconstant velocity fields.

## Acknowledgements

We thank F. Bernal for fruitful discussions. This work has been carried out under a Ramón y Cajal contract at the University of Cantabria.

## References

[1] J. A. Sethian, *A fast marching level set method for monotonically advancing fronts*, Proc. Natl. Sci. Sci. USA **93** (1996), 1591–1595.
*Fast Marching Methods*, SIAM Review **41**, No. 2 (1999), 199–235.

[2] D. L. Chopp, *Some Improvements of the Fast Marching Method*, SIAM J. Sci. Comput. **23**, No. 1 (2001), 230–244.

[3] K. Kály-Kullai, *A fast method to simulate travelling waves in nonhomogeneous chemical or biological media*, J. Math. Chem. **34** (2003), 163–176.

[4] B. K. Dey, M. R. Janicki, P. W. Ayers, *Hamilton-Jacobi equation for the least-action/least-time dynamical path based on fast marching method*, J. Chem. Phys. **121** (2004), 6667–6679.

[5] B. K. Dey, S. Bothwell, P. W. Ayers, *Fast marching method for calculating reactive trajectories for chemical reactions*, J. Math. Chem. **41** (2007), 1–25.

[6] B. K. Dey, P. W. Ayers, *Computing tunneling paths with the Hamilton-Jacobi equation and the fast marching method*, Molec. Phys. **105** (2007), 71–83.

[7] D. Adalsteinsson, J. A. Sethian, *The Fast Construction of Extension Velocities in Level Set Methods*, J. Comput. Phys. **148**, No. 2 (1998), 2–22.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE2007*
*Chicago, 20-23 June 2007*

# A Transshipment Problem with Random Demands

**Dr. S.N. Gupta**

School of Computing, Information & Mathematical Sciences,
University of the South Pacific,Laucala Campus,
Private Mail Bag, Suva, **FIJI.**
Email: gupta_s@usp.ac.fj ; guptasnath@yahoo.com

## Abstract

This paper presents a study of a stochastic transshipment problem with random demands. In our model the random demand has not been replaced by its expectation but the probabilistic nature of the problem has been built into the problem formulation itself so that the system has the opportunity to take maximum advantage of the probability distribution of demand. Deterministic equivalent of the problem is obtained and an algorithm is developed for solving the same.

*Keywords: transshipment, random demand,*
*global optimum, basic feasible solution.*

## 1. Introduction

The standard transportation problem and its several variants including the stochastic transportation problems have been extensively studied and special methods developed for solving them. However, the transshipment problem with random demands seems to have remained unexplored so far. Interest in transshipment problem arises because transshipments are often required for effective supply chain management and better replenishment strategies [2,4,6]. Transshipments also occur in the military logistics where direct transportation of goods to destination may not be advisable for security reasons. Moreover, the demands in real life are usually uncertain and have to be treated as random variables. This creates considerable complications. The purpose of this paper is to study a stochastic transshipment problem in which the demands at various destinations are assumed to be independent discrete random variables with known probability distributions. In this study, the random demand has not been replaced by its expectation but, following the technique of Dantzig[1] the probabilistic nature of the problem has been built into the problem formulation itself so that the system has the opportunity to take maximum advantage of the probability distribution of demand. The stochastic transshipment problem is reduced to an equivalent deterministic transportation type linear programming problem and an algorithm is developed to solve the same.

## 2. Problem Formulation

Consider a transshipment problem with m sources numbered 1, 2,………, m and n sinks numbered m+1, m+2,..., m+n. The sequential numbering of sources and sinks is found convenient because in a transshipment problem every source and sink acts both as a shipping point as well as a receiving point of goods.

Let, $a_i$ = the quantity available at source i= 1,2,……..m,

$b_j$ = the quantity demanded at sink j= m+1,m+2,………….,m+n,

$x_{ij}$ = the quantity shipped from station i to j (i, j= 1,2,……,m+n),

$c_{ij}$ = the per unit shipment cost from station i to j (i, j= 1,2,……,m+n),

$t_i$ = quantity transshipped at the station i (i= 1,2,……,m+n),

$l_i$ = per unit transshipment cost (including unloading, reloading, and storage etc.) at the station i (i= 1,2,……,m+n),

$s_j$ = revenue received (sale proceeds minus the handling costs like seller's commission etc.) per unit of demand satisfied at sink $D_j$

In the standard transshipment problem the objective is usually to minimize the total of transshipment and transportation costs. However, in our case we have no precise information concerning the demand $b_j$, but that we know its probability distribution for each j. So, in order to take care of the randomness of demands, instead of minimizing the total cost, we take our objective as the maximization of the net expected revenue (i.e., total expected revenue minus the total costs of procurement, transshipment and transportation). This would cause a tug of war between maximizing the expected revenue and minimizing the costs of transportation and transshipment.

Mathematically, the problem may be stated as under:

**Problem $P_1$:** Find $x_{ij}$ so as to

maximize $$F = \sum_{j=m+1}^{m+n} f_j(s_j, Y_j) - \sum_{i=1}^{m+n}{}^{*} \sum_{j=1}^{m+n}{}^{*} c_{ij} x_{ij} - \sum_{i=1}^{m+n} l_i t_i \qquad \ldots(2.1)$$

subject to $$\sum_{j=1}^{m+n}{}^{*} x_{ij} = \begin{cases} a_i + t_i & i = 1,2,\ldots\ldots, m \qquad \ldots(2.2a) \\ t_i & i = m+1, \ldots m+n \qquad \ldots(2.2b) \end{cases} \qquad \ldots(2.2)$$

$$\sum_{i=1}^{m+n}{}^{*} x_{ij} = \begin{cases} t_j & j = 1,2,\ldots\ldots, m \qquad \ldots(2.3a) \\ b_j + t_j & j = m+1, \ldots m+n \qquad \ldots(2.3b) \end{cases} \qquad \ldots(2.3)$$

$$x_{ij} \geq 0 \qquad \ldots \qquad \ldots \qquad \ldots(2.4)$$

$\sum_{j=1}^{m+n}{}^{*}$ indicates that the term j = i is excluded from the sum.

Here $f_j(s_j, Y_j)$ is a yet unknown function that describes the expected revenue from the destination j if a net total of $Y_j$ units are shipped to that destination. So the function F is a measure of the net expected revenue.

Constraints (2.2a) imply that the total quantity that leaves the source i ( = 1, 2, …, m) is equal to the quantity available plus the quantity transshipped and (2.2b) imply that the total quantity that leaves the sink i ( = m+1, m+2, …, m+n) is equal to the quantity transshipped. Similarly Constraints (2.3a) imply that the total quantity that arrives at the source j ( = 1, 2, …, m) is equal to the quantity that source transships and (2.3b) imply that the total arriving at the sink j (= m+1, m+2, …, m+n) is equal to the demand at that sink plus the quantity that the sink transships.

The transshipped quantities $t_i$ are unknown and generate -1 coefficients when brought over to left-hand side. So we impose an upper bound $t_o$ (say) on the amount that can be transshipped at any point such that

$$t_i \le t_o \quad \text{or} \quad t_i = t_o - x_{ii} \qquad i = 1, 2, 3,\ldots.., m+n, \qquad \ldots(2.5)$$

where $x_{ii}$ is a nonnegative slack. After substituting (2.5) in (2.1) to (2.3) and on simplifying the original transshipment problem $P_1$ is reduced to the following genuine transportation type linear programming problem.

**Problem $P_2$:**

$$\text{Maximize} \qquad F = \sum_{j=m+1}^{m+n} f_j(s_j, Y_j) - \sum_{i=1}^{m+n}\sum_{j=1}^{m+n} c_{ij}x_{ij} - \sum_{i=1}^{m+n} l_i t_o \qquad \ldots(2.6)$$

$$\text{Subject to } \sum_{j=1}^{m+n} x_{ij} = \begin{cases} a_i + t_o & i = 1,2,\ldots..,m \quad \ldots(2.7a) \\ t_o & i = m+1,\ldots m+n \quad \ldots(2.7b) \end{cases} \qquad \ldots(2.7)$$

$$\sum_{i=1}^{m+n} x_{ij} = \begin{cases} t_o & j = 1,2,\ldots..,m \quad \ldots(2.8a) \\ b_j + t_o & j = m+1,\ldots m+n \quad \ldots(2.8b) \end{cases} \qquad \ldots(2.8)$$

$$x_{ij} \ge 0 \qquad i, j = 1, 2, 3,\ldots.., m+n, \qquad \ldots(2.9)$$

where $c_{ii} = -l_i$ and note that the * (asterisk) on the summations have disappeared due to the inclusion of $x_{ii}$.

The upper bound $t_o$ can be interpreted as the size of a fictitious stockpile at each of the sources and sinks which is large enough to take care of all transshipments. Obviously, $t_o$ need not be larger than the total quantity required to be shipped. So, we take $t_o = \sum_{i=1}^{m} a_i$ which ensures that $t_o$ is not limiting. The unused stockpile at the station i=1, 2, ……..m+n, if any, will be absorbed in the slack $x_{ii}$.

Thus, the $m \times n$ order transshipment problem $P_1$ has been converted into a direct shipment transportation problem $P_2$ of order $(m+n) \times (m+n)$ which can have no more than 2(m+n)-1 variables different from zero. However, (m+n) of these non-zero variables are the slack variables $x_{ii}$ representing the unused stockpile and hence there are in fact no more than (m+n-1) variables of interest which are different from zero.

As indicated earlier, the demands $b_j$ (j =m+1, m+2, …,m+n) are independently distributed discrete random variables. Let the probability distribution of $b_j$ be as below:

| Demand $b_j$ | $b_{1j}$ < | $b_{2j}$ < | ………< | $b_{H_j j}$ |
|---|---|---|---|---|
| Prob $(b_j=b_{hj}) = p_{hj}$ | $p_{1j}$ | $p_{2j}$ | ……….. | $p_{H_j j}$ |
| Prob $(b_j \geq b_{hj}) = \pi_{hj}$ | $\pi_{1j} = \sum_{h=1}^{H_j} p_{hj}$ (= 1) | $\pi_{2j} = \sum_{h=2}^{H_j} p_{hj}$ | ………… | $\pi_{H_j j} = p_{H_j j}$ |

However, the moment we treat $b_j$ as random variable, a new problem begins to rear its head. The constraints (2.8*b*) fail to make sense. To make the problem meaningful it has to be reformulated into an equivalent deterministic problem.

**3. Equivalent Deterministic Problem**

In the function $f_j(s_j, Y_j)$, note that $Y_j$, the net quantity shipped to sink j, can be any amount between the lowest value $b_{1j}$ and the highest value $b_{H_j j}$ in the probability distribution of the demand $b_j$ (j = m+1, m+ 2, …, m+n).

If $0 \leq Y_j \leq b_{1j}$, then each of the $Y_j$ units shall be absorbed with probability $\pi_{1j}$ (= 1). Hence, the expected revenue is = $s_j \pi_{1j} Y_j$.

If $b_{1j} \leq Y_j \leq b_{2j}$, then each unit upto $b_{1j}$ shall be absorbed with probability $\pi_{1j}$ and each of the additional units $(Y_j - b_{1j})$ shall be absorbed with probability $\pi_{2j}$. Hence, the expected revenue is = $s_j \pi_{1j} b_{1j} + s_j \pi_{2j} (Y_j - b_{1j})$.

In general, if $b_{hj} \leq Y_j \leq b_{h+1,j}$, then the expected revenue is
$= s_j\{ \pi_{1j} b_{1j} + \pi_{2j} (b_{2j} - b_{1j}) + \ldots\ldots. \pi_{2j} (b_{hj} - b_{h-1,j}) + \pi_{h+1,j} (Y_j - b_{hj})\}$.

Let us now break $Y_j$ into incremental units $y_{hj}$ (h = 1, 2,…$H_j$) as:

$$Y_j = y_{1j} + y_{2j} + y_{3j} + \ldots. + y_{hj} + \ldots. y_{H_j j} \qquad \ldots(3.1)$$

where
$$\left.\begin{array}{ll} 0 \le y_{1j} \le b_{1j} & = R_{1j}\ (say) \\ 0 \le y_{2j} \le b_{2j} - b_{1j} & = R_{2j} \\ : \quad : \quad : & : \\ 0 \le y_{H_j j} \le b_{H_j j} - b_{H_j -1, j} & = R_{H_j j} \end{array}\right\} \qquad \ldots(3.2)$$

Relation (3.1) makes physical sense only if there exists some h = $h_j$ (say) such that all intervals below the $h_j^{th}$ interval are filled to capacity and all intervals above it are empty i.e.

$$\left.\begin{array}{ll} y_{hj} = R_{hj} & (h = 1,2,...,h_j - 1) \\ y_{hj} \le R_{hj} & (h = h_j) \\ y_{hj} = 0 & (h = h_j + 1,...,H_j) \end{array}\right\} \qquad \ldots(3.3)$$

Assuming for the time being that the conditions (3.3) hold, the total expected revenue from sink j is: $f_j(s_j, y_j) = \sum_{h=1}^{H_j} s_j \pi_{hj} y_{hj}$

Substituting the value of $f_j(s_j, y_j)$ in (2.6), the net expected revenue is:

$$F = \sum_{j=m+1}^{m+n} \sum_{h=1}^{H_j} s_j \pi_{hj} y_{hj} - \sum_{i=1}^{m+n} \sum_{j=1}^{m+n} c_{ij} x_{ij} - \sum_{i=1}^{m+n} l_i t_o$$

The third term on the right hand side is a constant that can now be ignored but adjusted in the end.

Since we have to maximize F (or minimize –F), so, ignoring the constant term and treating both $x_{ij}$ and $y_{hj}$ as decision variables, the deterministic equivalent of the Problem $P_2$ is:

**Problem P$_3$:** Minimize $Z = \sum_{i=1}^{m+n} \sum_{j=1}^{m+n} c_{ij} x_{ij} + \sum_{i=1}^{m+n} \sum_{h=1}^{H_j} d_{hj} y_{hj}$ …(3.4)

where $d_{hj}$= -$s_j \pi_{hj}$

Subject to $$\sum_{j=1}^{m+n} x_{ij} = \begin{cases} a_i + t_o & i = 1,2,.....,m \quad \ldots(3.5a) \\ t_o & i = m+1,...m+n \quad \ldots(3.5b) \end{cases} \qquad \ldots(3.5)$$

$$\left.\begin{array}{lll} \sum_{i=1}^{m+n} x_{ij} = t_o, & j=1,2,.....,m & \ldots(3.6a) \\ \sum_{i=1}^{m+n} x_{ij} - \sum_{h=1}^{H_j} y_{hj} = t_o, & j= m+1,..., m+n & \ldots(3.6b) \end{array}\right] \qquad \ldots(3.6)$$

$$x_{ij}, y_{hj} \ge 0 \qquad (\forall i, j, h) \qquad \ldots(3.7)$$

$$y_{hj} \le R_{hj} \qquad (\forall h, j) \qquad \ldots(3.8)$$

and subject to the additional stipulation that the constraints (3.3) are also satisfied. Fortunately, it turns out that (3.3) do not restrict our choice of optimum solution in any way. This we prove in the following theorem.

**Theorem 1**

*A feasible solution to Problem $P_3$ can always be improved if it violates any of the constraints (3.3).*

**Proof**

Let ($x^*_{ij}$, $y^*_{hj}$) be a feasible solution to Problem $P_3$ obtained on ignoring (3.3). The value of Z at this solution is:

$$Z^* = \sum_{i=1}^{m+n}\sum_{j=1}^{m+n} c_{ij} x^*_{ij} + \sum_{j=m+1}^{m+n}\sum_{h=1}^{H_j} d_{hj} y^*_{hj}$$

Suppose that there exists some h = $h^o$ & j = $j^o$ such that $y^*_{h^o j^o} < R_{h^o j^o}$ & $y^*_{h^o+1,j^o} > 0$

It is clearly a violation of the constraints (3.3).

Now, we increase $y^*_{h^o j^o}$ and decrease $y^*_{h^o+1,j^o}$ by equal amounts $\theta\,(> 0)$ such that the feasibility of the solution is not disturbed. The new value of the objective function becomes: $Z^o = Z^* + \theta\ (d^*_{h^o j^o} - d^*_{h^o+1,j^o})$

But $(d^*_{h^o j^o} - d^*_{h^o+1,j^o}) \leq 0$, as $\pi_{hj} \geq \pi_{h+1,j}$ for all h and j.

Hence it follows that $Z^o \leq Z^*$. **Q.E.D.**

This result shows that if an optimum solution to Problem $P_3$ is obtained after ignoring (3.3), it shall suo moto satisfy (3.3). Thus, to solve problem $P_3$, we may simply ignore the constraints (3.3).

## 4. Preliminaries to the Solution of Problem $P_3$

1. It is assumed that the set of all feasible solutions of Problem $P_3$ is regular (i.e. non-empty and bounded).
2. Problem $P_3$ is a transportation type linear programming problem with upper bound restrictions on some variables. So, its global minimum exists at a basic feasible solution of its constraints.
3. We shall, hereinafter, call the constraints (3.5) through (3.7) as the original system and the constraints (3.5) through (3.8) as the bounded system. As none of the constraints in the original system is redundant, a basic feasible solution to the original system shall contain 2(m+n) basic variables. For the capacitated system also, a basic feasible solution shall contain 2(m+n) basic variables and the same may be found by working on the original system provided that some of the non basic variables are

allowed to take their upper bound values[3].

4. The special structure of Problem $P_3$, permits us to arrange it into an array as shown in table 1:

**Table 1**

| | | | | | | |
|---|---|---|---|---|---|---|
| $x_{11}$<br>$c_{11}$ | … | $x_{1m}$<br>$c_{1m}$ | $x_{1\ m+1}$<br>$c_{1\ m+1}$ | … | $x_{1\ m+n}$<br>$c_{1\ \ m+n}$ | $\mathbf{a_1+t_o}$ |
| …….<br>……. | …<br>… | …….<br>……. | …….<br>……. | …<br>… | …….<br>……. | **…….<br>…….** |
| $x_{m1}$<br>$c_{m1}$ | … | $x_{mm}$<br>$c_{mm}$ | $x_{m,m+1}$<br>$c_{m,m+1}$ | … | $x_{m\ m+n}$<br>$c_{m\ m+n}$ | $\mathbf{a_m+t_o}$ |
| $x_{m+1,1}$<br>$c_{m+1,1}$ | … | $x_{m+1m}$<br>$c_{m+1m}$ | $x_{m+1,m+1}$<br>$c_{m+1,m+1}$ | … | $x_{m+1\ m+n}$<br>$c_{m+1\ m+n}$ | $\mathbf{t_o}$ |
| …….<br>……. | …<br>… | …….<br>……. | …….<br>……. | …<br>… | …….<br>……. | **…….<br>…….** |
| $x_{m+n,1}$<br>$c_{m+n\ 1}$ | … | $x_{m+n,m}$<br>$c_{m+n\ m}$ | $x_{m+n,m+1}$<br>$c_{m+n\ m+1}$ | … | $x_{m+n\ m+n}$<br>$c_{m+n\ m+n}$ | $\mathbf{t_o}$ |
| $\mathbf{t_o}$ | … | $\mathbf{t_o}$ | $y_{1\ m+1}$ $R_{1\ m+1}$<br>$d_{1\ m+1}$ | … | $y_{1,m+n}$ $R_{1,m+n}$<br>$d_{1,m+n}$ | |
| | | | ……. | … | ……. | |
| | | | ……. | … | ……. | |
| | | | $Y_{H,m+1}$ $R_{H,m+1}$<br>$d_{H,m+1}$ | … | $Y_{H,m+n}$ $R_{H,m+n}$<br>$d_{H,m+n}$ | |
| | | | $\mathbf{t_o}$ | **…** | $\mathbf{t_o}$ | |

In the above table, there are (m+n) rows in columns j=1,2,….m and (m+n+H) rows in columns j=m+1,m+2,….m+n. Here H = max $H_j$, so that there shall be some empty boxes near the bottom of the table in columns j=m+1,m+2,….m+n. These empty boxes shall be crossed out.

Absence of the row totals for $y_{hj}$'s in the table indicates that there are no row equations for $y_{hj}$ variables. Besides, to obtain the column equations (3.6b), each $y_{hj}$ has to be multiplied by (-1). We have omitted (-1) from $y_{hj}$ boxes for convenience.

## 5. Initial Basic Feasible Solution

To start with, we fix the demands $b_j$'s approximately equal to their expected values such that $\sum_{j=m+1}^{m+n} b_j = \sum_{i=1}^{m} a_i$ and also such that for all j except $j = j^*$, each $b_j$ falls at the upper end of one of the intervals $y_{hj}$ into which the $b_j$ has been divided i.e. $b_j = \sum_{h=1}^{h'_j} R_{hj}$ for some $h'_j \leq H_j$ and for all j except $j = j^*$ (the $b_j$ can always be so chosen that it is done).

With these fixed demands the upper portion (above the double line) of the Table 1 resembles a (m+n)×(m+n) standard transportation problem for which an initial basic feasible solution with {2(m+n)-1} basic variables is obtained by any of the several available methods. Now, in each of the columns j=m+1,m+2,….m+n the values of the non basic $y_{hj}$'s are entered at their upper bounds $R_{hj}$ in turn h =1, 2,…. until we have entered enough non basic $y_{hj}$'s so that their sum over h is equal to $b_j$. Obviously, we shall never have to enter $y_{hj}$ below its upper bound except in column $j = j^*$, where the last nonzero entry will be $y_{hj*} \leq R_{hj*}$. This last entry and the {2(m+n)-1} basic $x_{ij}$'s found earlier, constitute the required initial basic feasible solution with 2(m+n) basic variables. In case the last non zero entry in column j* is also at its upper bound, then we take the last $y_{hj}$ entry of any column as our 2(m+n)th basic variable.

## 6. Optimality criteria

Let the simplex multipliers corresponding to the objective function Z (Problem $P_3$) be $u_i$ and $v_j$ $(\forall i, j = 1,2,...,m+n)$. These are determined by solving the following equations.

$$\left.\begin{aligned} c_{ij} + u_i + v_j &= 0 \quad \text{for basic } x_{ij} \\ d_{hj} - v_j &= 0 \quad \text{for basic } y_{hj} \end{aligned}\right\} \qquad \ldots(6.1)$$

These are 2(m+n) linear equations in as many unknowns $u_i$ and $v_j$ and can be easily solved. Let the relative cost coefficients corresponding to the variables $x_{ij}$ and $y_{hj}$ be $\delta_{ij}$ and $\lambda_{hj}$. These are determined by solving the following equations:

$$\left.\begin{aligned} \delta_{ij} &= c_{ij} + u_i + v_j \quad \text{for non basic } x_{ij} \\ \lambda_{hj} &= d_{hj} - v_j \qquad \text{for non basic } y_{hj} \end{aligned}\right\} \qquad \ldots(6.2)$$

It can be easily shown that for a given basic feasible solution ($x_{ij}$, $y_{hj}$) of the Problem $P_3$, the value of the objective function Z is:

$$Z=\sum_{i=1}^{m+n}\sum_{j=1}^{m+n}\delta_{ij}x_{ij}+\sum_{j=m+1}^{m+n}\sum_{h=1}^{H_j}\lambda_{hj}y_{hj}-\left\{\sum_{i=1}^{m}u_i\left(a_i+t_o\right)+\sum_{i=m+1}^{m+n}u_it_o+\sum_{j=1}^{m+n}v_jt_o\right\}$$

$$=\sum_{i=1}^{m+n}\sum_{j=1}^{m+n}\delta_{ij}x_{ij}+\sum_{j=m+1}^{m+n}\sum_{h=1}^{H_j}\lambda_{hj}y_{hj}-\left\{\sum_{i=1}^{m}u_ia_i+t_o(\sum_{i=m+1}^{m+n}u_i+\sum_{j=1}^{m+n}v_j)\right\}\quad \ldots(6.3)$$

Here, $\delta_{ij}=0$ for all basic $x_{ij}$ and also the values of the non basic $x_{ij}$ are zero. So, the first term on the r.h.s. of (6.3) vanishes. Similarly $\lambda_{hj}=0$ for the basic $y_{hj}$, but as regards the values of non basic $y_{hj}$'s - some are zero and the others are at their upper bounds. Hence,

$$Z=\sum_{j=m+1}^{m+n}{}^{*}\sum_{h=1}^{H_j}{}^{*}\lambda_{hj}R_{hj}-\left\{\sum_{i=1}^{m}u_ia_i+t_o(\sum_{i=m+1}^{m+n}u_i+\sum_{j=1}^{m+n}v_j)\right\}\quad \ldots(6.4)$$

Where $\sum^{*}$ indicates the sum over those non basic $y_{hj}$ which are at their upper bounds. Now if the value of any one of the non basic variables $x_{st}$ or $y_{rt}$ is changed to:

$$\hat{x}_{st}=(x_{st}+\theta)\quad \text{or}\quad \hat{y}_{rt}=(y_{rt}\pm\theta),$$

with the other non basic variables remaining unaltered and the basic variables adjusted to maintain feasibility of the solution, then the improved value of Z shall be:

$$\hat{Z}=Z+\theta\,\delta_{ij}\quad \text{or}\quad \hat{Z}=Z\pm\theta\,\lambda_{hj},\quad \text{as the case may be.}$$

Note that we take plus sign if $y_{rt}=0$ and minus sign if $y_{rt}=R_{rt}$.

The objective function will improve iff $\hat{Z}-Z<0$ i.e.

$$(Z+\theta\,\delta_{ij})-Z<0\quad \text{or}\quad (Z\pm\theta\,\lambda_{hj})-Z<0$$

$$\Rightarrow\ \theta\,\delta_{ij}<0\quad \text{or}\quad \pm\theta\,\lambda_{hj}<0$$

$$\Rightarrow\ \delta_{ij}<0\quad \text{or}\quad \pm\lambda_{hj}<0$$

(Since in non degenerate case $\theta>0$ and in degenerate case $\theta=0\Rightarrow\hat{Z}=Z$).

Thus, the current solution is optimum iff

$$\left.\begin{array}{ll}\delta_{ij}\geq 0 & (\forall\ \text{non basic } x_{ij})\\ \lambda_{hj}\geq 0 & (\forall\ \text{non basic } y_{hj} \text{ at zero level})\\ \lambda_{hj}\leq 0 & (\forall\ \text{non basic } y_{hj} \text{ at upper bound})\end{array}\right\}\quad \ldots(6.5)$$

If any of the optimality criteria (6.5) is violated, the current solution can be improved. The non basic variable which violates (6.5) most severely is selected to enter the basis. The values of the new basic variables are found by applying the usual $\theta$-adjustments. It should, however, be kept in mind that the coefficient of each $y_{hj}$ in the column equations (3.7b) is (-1).

The variable to leave the basis is the one that becomes either zero or equal to its upper bound. If two or more basic variables reach zero or their upper bounds simultaneously then only one of them becomes non basic. Should it happen that the entering variable itself attains upper or lower bound (zero) without simultaneously making any of the basic variables zero or equal to its upper bounds, the set of basic variables remains unaltered; only their values are changed to allow the so-called entering variable to be fixed at its upper or lower bound.

### Termination of the process

The process is bound to terminate with a finite number of iterations as it involves movement from one basic feasible solution to another basic feasible solution, and the number of basic feasible solutions is always finite. The author has tested the algorithm on several numerical examples.

**Acknowledgement**
I am indebted to the authorities of the USP, Fiji for facilitating this research.

### References

[1] Dantzig G.B. (1955), “Linear Programming under Uncertainty”, Management Science 1, 197-206.

[2] Garg, R. and Prakash, S. (1985), “Time minimizing Transshipment Problem”, Indian J. of Pure Applied Mathematics 16, 449-460.

[3] Garvin W.W. (1963), “Introduction to Linear Programming”, McGraw Hill Book Company, New York.

[4] Herer, Y.T. and Tzur, M. (2001), “The Dynamic Transshipment Problem”, Naval Research Logistic Quarterly 48, 386-408.

[5] Orden, A. (1956), “The Transshipment Problem” Management Science 2, 276-285.

[6] Zhao, H., Despande, V. and Rayan, J.K. (2006), “Emergency transshipment in decentralized dealer network: when to send and accept transshipment requests” Naval Research Logistic Quarterly 53, 547-567.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# Third Order Analysis of Efficiency and Improvement for Barabási-Albert Networks

**B.Hernández-Bermejo[1], J.Marco-Blanco[1] and M.Romance[1]**

[1] *Departamento de Matemática Aplicada,ESCET, Universidad Rey Juan Carlos,*
*c/Tulipán s.n., 28933 Móstoles, Madrid, Spain*

emails: `benito.hernandez@urjc.es`, `j.marcob@alumnos.urjc.es`,
`miguel.romance@urjc.es`

### Abstract

Third order analytical expressions for Barábasi-Albert networks efficiency are obtained. They are used to study the improvement of such kind of networks.

*Key words: Network efficiency; Network vulnerability; Scale-free networks; Network topology; Network design*
*MSC 2000: 02.10*

## 1 Introduction

Performance and stability are very important properties to characterise complex networks which have been widely studied [2,8,9,10,11,12,13,14,15,16,17,18]. Magnitudes as *efficiency* [17], *vulnerability* [11] *improvement* [18] have been introduced and studied in order to quantify these properties. Real networks, whose understanding is a primary goal of the field of complex networks, may be formed for a huge number of nodes and connections (of the order of hundred-thousand or even million)[3,5,19]. Therefore, calculations in the modelled networks necessary to measure these or other magnitudes may be computationally very expensive. For this reason, we consider of a great importance the development of alternative “cheaper” approximations which lead to the calculation estimation of relevant magnitudes. In this work we follow this guideline in networks constructed following the Bárabasi and Albert (BA) network model [1], some of which properties reproduce those of real networks. The special properties of the tree structure allow obtaining analytic estimates of the *efficiency* and the *improvement* of Bárabasi and Albert Networks. These estimates are used to obtain important conclusions which show the advantages of the preferential attachment way of growth.

We focus on the study of *efficiency* and *improvement* in BA networks. *Efficiency* quantifies the performance of the network. If $G = (V, E)$ is a complex network, its *efficiency* is defined [17] as

$$E(G) = \frac{1}{n(n-1)} \sum_{i \neq j \in G} \frac{1}{d_{ij}}, \tag{1}$$

where $d_{i,j}$ is the distance between the nodes $i$ and $j$. *Improvement* measures the relative variation of the efficiency of a network when a new element is added. It is defined [18] as:

$$I(G, G^*) = \frac{E(G^*) - E(G)}{E(G)}, \tag{2}$$

where $G^*$ is the network $G$ with the addition of the new elements.

First of all we shall briefly introduce the BA model, whose most important case leads to tree structure networks. Then we present some general properties of this kind of networks, and making use of these results, we will finally obtain analytic expressions for efficiency and improvement.

## 2 Barabási and Albert Model

Most real networks are created by successive addition of nodes. It is reasonable to consider this addition to depend upon the network properties at the moment that a new connection is created. The Barabási and Albert model (BA model) combines both characteristics in the following way:

1. There is a periodic addition of nodes that are connected to the already present ones: we begin with a small graph of $m_0$ nodes. In every time step a new node with $m$ edges is introduced and connected.

2. The higher is the degree of an existing node, the higher is the probability of being connected to the new node: the probability that the new node is linked to the existing node $i$ is

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j} \tag{3}$$

   where $k_i$ is the degree of node $i$. After $t$ time steps a network with $n = m_0 + t$ nodes and $mt$ edges is obtained.

Networks constructed according to this model reproduce the aforementioned self-similarity present in many real-world networks: the degree distribution of the system follows a power law $p(k) \sim k^{-\gamma}$ independently of the size of the network. Due to this self-similar character, these networks are also called scale-free networks. The case $m = 1$, in which new nodes connect to the network with one edge is one of the most common. The resulting networks constructed this way have a tree structure.

## 3 Some properties of networks with tree structure

As mentioned, due to the way of construction, BA networks with $m = 1$ have a tree structure. In a connected network with such structure, there exist univocal relations between the nodes degrees and the distances among them as it is explained. In figure 1.a) it is represented the node $i$ of a tree type graph. The distance among first neighbours of the same node is always 2 (it cannot be 1 because in that case there would be a cycle and the network could not be a tree). Then, there are $k_i(k_i - 1)$ length-two distances which pass through node $i$. The sum over all nodes $\sum_{i=1}^{n} k_i(k_i - 1)$ gives the total number of distances 2 in the network.

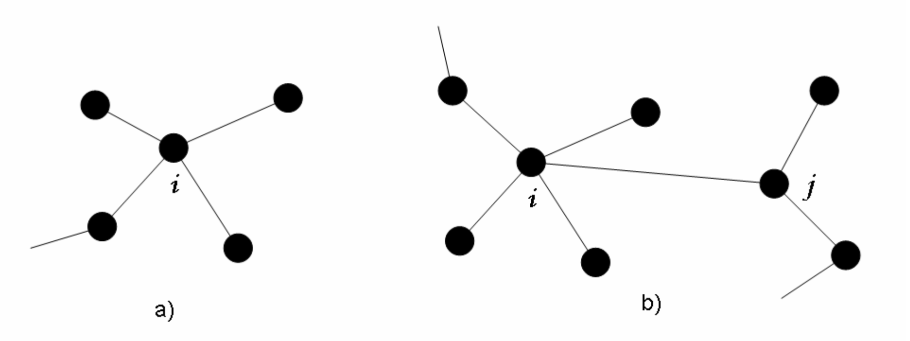


Figure 1: *a)Node i with degree $k_i = 4$ of a tree type network. 12 length 2 distances pass through it. b) The distance among first neighbours of i and j is 3.*

In figure 1.b) it can be seen that each first neighbour of $i$ (except node $j$) is placed at a distance of 3 from every first neighbour of $j$ and vice versa. Then, there are $(k_i - 1)(k_j - 1)$ length 3 distances which pass through $i$ and $j$. If we sum over all connected pair of nodes in the network, we obtain the total number of length 3 distances.

But the efficiency of a network is precisely the weighted sum of the distances among its nodes, then if we know the number of distances until $i$-th order, we may obtain an $i$-th order approximation of the efficiency. To make use of these properties we need information about the network nodes degree.

## 4 Mean Field Theory

Barabási and Albert have also developed a method [4] that describes analytically the evolution of BA Networks. In the mean-field theory it is assumed that the degree of each node of the network evolves as a continuous variable with a rate of change

$$\frac{\partial k_i}{\partial t} = m\Pi(k_i). \tag{4}$$

With the initial condition that node $i$ was introduced at time $t_i$, we obtain an analytical expression for the time dependence of the degree of node $i$ given by

$$k_i(t) = m\left(\frac{t}{t_i}\right)^{0.5}, \tag{5}$$

where $\{k_1, k_2, ..., k_n\}$ is termed degree sequence. With this information of the degree sequence and the mentioned tree structure properties we develop interesting results on BA networks efficiency and vulnerability.

## 5 Upper and lower bounds for efficiency in BA networks

Combining the degree sequence with the properties of the tree structure showed in section 2, third order lower and upper estimates for BA graphs are obtained. The results are compared with empirical values obtained by computer simulation of the same networks. That is represented in figure 2.

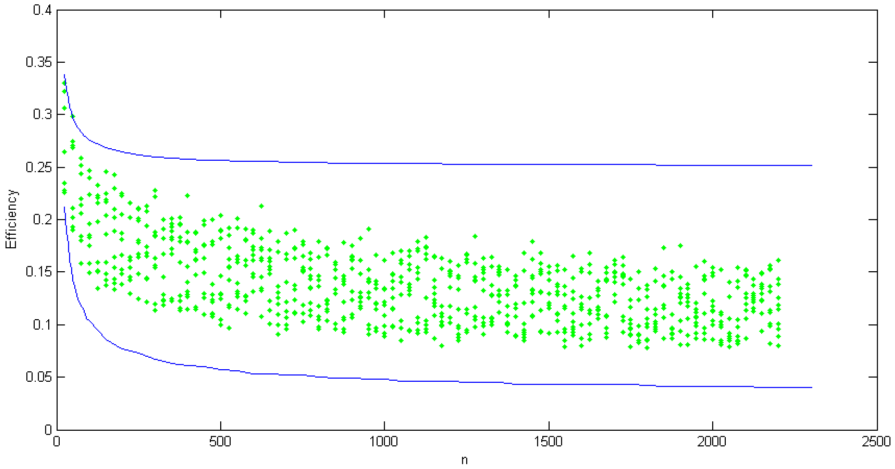


Figure 2: *Efficiency of BA networks with m=1 vs number of nodes, n (dots). Upper and lower estimates (lines)*

BA networks with $m = 1$ with number of nodes ranging from 3 to 2300 are plotted. Ten realisations are made for each size of network. They are compared with the upper and lower analytic estimates $(E^+, E^-)$ given by

$$E^+(G) = \frac{1}{n(n-1)}\left[\sum_{i=1}^{3}\frac{1}{i}d_i - (n(n-1) - d_1 - d_2 - d_3)\frac{1}{4}\right], \tag{6}$$

$$E^-(G) = \frac{1}{n(n-1)}\left[\sum_{i=1}^{3}\frac{1}{i}d_i - (n(n-1) - d_1 - d_2 - d_3)\frac{1}{D}\right], \tag{7}$$

where $d_i$ is the number of length $i$ distances in the graph and $D$ the diameter of the graph. With these estimates we may approximate the values for the efficiency for larger networks, whose size may not allow direct calculations. But more important is the trend of the efficiency showed in the figure, which reveals new features of BA Networks.

## 6 Improvement on BA networks

In this section we study the *improvement* (2) of a network which growths following the BA model with $m = 1$. The empirical values of the improvement and the analytic ones are compared. In figure 3 empirical values of $I(G, G^*)$ are shown, where $G*$ is the the graph which results from adding $G$ a new node with a new edge. In figure 4 it is represented the corresponding quantities $\frac{\partial E^+(G)}{\partial t}$ and $\frac{\partial E^-(G)}{\partial t}$.

We observe in figures 3 and 4 that for large numbers of nodes the preferential attachment leads to a conservation of the efficiency. As in the degree distribution, the scale-free character arise.

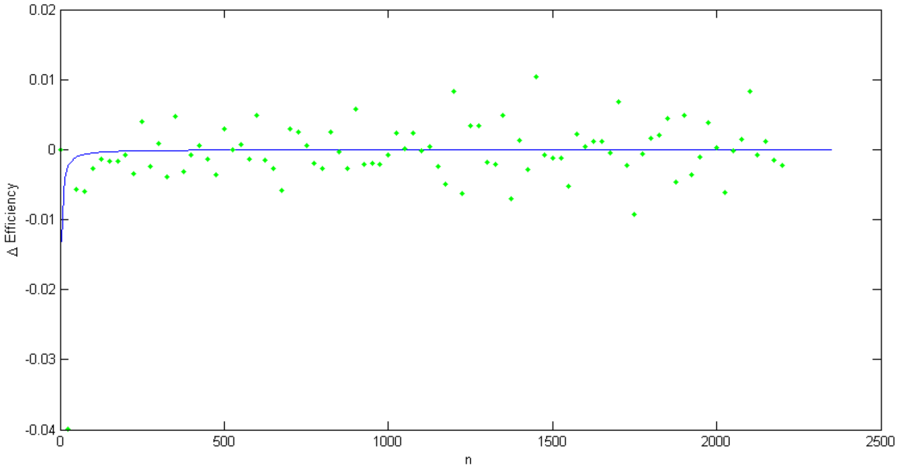


Figure 3: *Empirical improvement* $\frac{E(G*)-E(G)}{E(G)}$ *against* $n$ *(dots). Numerical fitting (line).*

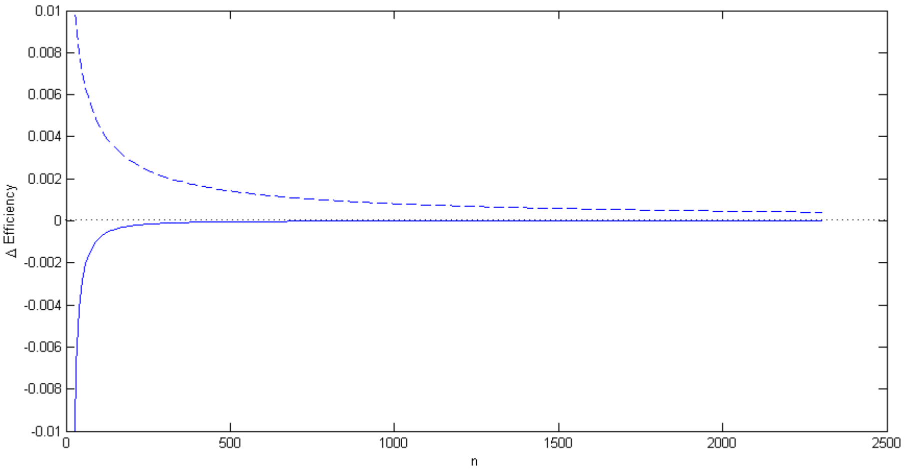


Figure 4: $\frac{\partial E^{+}(G)}{\partial t}$ *(dashed line) and* $\frac{\partial E^{-}(G)}{\partial t}$ *(continuous line) against* $n$.

When a new node is added to a network, the highest increase of the efficiency would occur if the new node connects to the most connected existing node. On the other hand, the network would become very vulnerable to a deliberate removal of that node, that is, more sensible to an intentional attack. Therefore, the preferential attachment seems to lead to a growth that balances the efficiency and the robustness against intentional attacks to the network.

ACKNOWLEDGEMENTS: The authors acknowledge the grant from the Spanish MEC, Project Ref. MTM2006-10053.

## References

[1] A. L. Barabasi and R. Albert, “Emergence of scaling in random networks”, *Science* **286** (1999)509.

[2] R. Albert, H. Jeong and A. L. Barabasi, “Error and attack tolerance of complex networks”, *Nature* **406** (2000)378.

[3] R. Albert and A. L. Barabasi, “Statistical Mechanics of complex networks”, *Rev. Mod. Phys.* **74** (2002) 47–97.

[4] . L. Barabasi, R. Albert and H. Jeong, “Mean-field theory for scale-free random networks”, *Physica A* **272** (1999) 173-187.

[5] . L. Barabasi, R. Albert and H. Jeong, “Scale-free characteristics of random networks: The topology of the World Wide Web”, *Physica A* **281** (2000) 69.

[6] S. Bocaletti, V. Latora, Y. Moreno, M. Chavez and D.-U. Hwang, “Complex Networks: Structure and dynamics”, *Phys. Rep.* **424** (2006) 175-308.

[7] B. Bollobas and O. Riordan, “The diameter of a scale-free random graph”, *Combinatorica* **24** (2004) 5-34.

[8] R. Cohen, K. Erez, D. ben-Avraham and S. Havlin, “Resilience of the Internet to Random Breakdowns”, *Phys Rev Lett* **85** (2000) 4626.

[9] R. Cohen, K. Erez, D. ben-Avraham and S. Havlin, “Breakdown of the Internet under Intentional Attack”, *Phys Rev Lett* **86** (2001) 33682.

[10] R. Criado, B. Hernández-Bermejo and M. .Romance, “On the correlations among efficiency, vulnerability and cost in random networks”, *CMMSE-2005* (2005) 1-8.

[11] R. Criado, J. Flores, B. Hernández-Bermejo, J. Pello and M. Romance, “Effective measurement of network vulnerability under random and intentional attacks”, *Journal of Mathematical Modelling and Algorithms* **4** (2005) 307-316.

[12] R. Criado, A. García del Amo, B. Hernández-Bermejo and M. Romance, “New Results in computable efficiency and its stability for complex networks”, *J. Comput. Appl. Math.* **196** (2006) 59-74.

[13] R. Criado, B. Hernández-Bermejo, J. Marco-Blanco and M. Romance, “Asymptotic estimate for efficiency, vulnerability and cost for random networks”, to appear at *Journal of Computational and Applied Mathematics.*

[14] P. Crucitti, V. Latora, M. Marchiori and A. Rapisarda, “Efficiency of scale-free networks: error and attack tolerance”, *Physica A* **320** (2003) 622–642.

[15] A. H. Dekker and B. D. Colbert, “Network robustness and graph theory”, *Proceedings of ACSC04, the 27th Australasian Computer Science* (2004) .

[16] P. Holme, B. Kim, C. Yoon and S. Han, “Attack vulnerability of complex networks”, *Phys Rev E* **65** (2003) 056109.

[17] V. Latora and M. Marchiori, “Efficient Behavior of Small-World Networks”, *Phys Rev Lett* **87** (2001) 198701.

[18] V. Latora and M. Marchiori, “Vulnerability and Protection of Network Infrastructures”, *Phys. Rev. E* **71** (2005) 015103(R).

[19] M. E. J. Newman, “The structure and function of complex networks”, *SIAM review* **45** (2003) 167–256.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# A New Data Structure for Multiplying a Sparse Matrix with a Dense Vector

**Shahadat Hossain**[1]

[1] *Department of Mathematics and Computer Science, University of Lethbridge, Alberta, Canada*

emails: `shahadat.hossain@uleth.ca`

**Abstract**

We propose an efficient storage scheme for sparse matrices with general sparsity pattern. The new storage scheme utilizes a compression procedure to improve the temporal and spatial locality of the data for sparse linear algebra operations e.g., matrix-vector product calculation. By accepting overhead for explicitly storing a few zero entries, indirect access to the dense input vector is avoided. Furthermore, it uses column reordering to minimize the auxiliary storage needed. Our proposal is suitable for vector- as well as modern cache-based super-scalar architectures. Preliminary numerical testing demonstrate the effectiveness of the new storage scheme.

*Key words: column compression, structural orthogonality, consecutive elements*

# 1 Introduction

Computations involving sparse matrices arise frequently in problems in scientific and engineering applications. For example, discretization of partial differential equations using finite-difference or finite-element schemes give rise to matrix problems that are large and a significant proportion of their entries are zero[5]. Solving such large-scale problems satisfactorily on even the most advanced present day computers poses many algorithmic and data management challenges. In this paper we focus on one of the fundamental computational kernels of sparse linear algebra, namely computing the product of a sparse matrix $A \in R^{m \times n}$ with a dense vector $x \in R^n$. For example, sparse eigen value computation methods such as the Jacobi-Davidson algorithm [2] require repeated calculation of matrix-vector products (MVP). Efficient execution of MVP calculation depends on the effective utilization of a priori known sparsity information by avoiding operations on zero entries. As such, only the nonzero entries of the sparse matrix are stored in contiguous locations of computer memory. The data storage schemes for sparse matrices are quite varied and some of the more specialized ones are naturally derived from specific computational problems. A survey of more common storage schemes can be found in [1]. The data structures for sparse matrices with arbitrary sparsity pattern usually have two components – an array to store the nonzero data values and auxiliary arrays to identify the row and column indices of those data values. The Compressed Row Storage (CRS) is one of the popular storage schemes for representing sparse matrices with no special structure (e.g. banded). The shorter length of the inner loop of MVP computation, however, causes performance degradation with the CRS scheme on vector processors. Furthermore, most of these storage schemes also involve indirect access to data elements. The main contribution of this paper is a new proposal for storing sparse matrices. The important features of the new storage scheme are as follows:

1. it uses structural orthogonality to pack the nonzero entries from different matrix rows into a small number of dense vectors (that may store some zero) each of dimension $n$,

2. there are no indirect access to the vector $x$ in the calculation of the product $Ax$ and,

3. it uses column ordering to place the nonzero entries in each row of $A$ into contiguous locations and thereby reducing the auxiliary storage.

The paper is organized in 4 sections. After a brief introduction to the CRS we describe the new storage scheme in Section 2. We provide algorithms for performing matrix-vector multiplication using the storage schemes discussed in this section. Section 3 reports some preliminary experimental results for performing MVP calculation. Section 4 concludes the paper with notes on further research.

## 2 Representing Sparse Matrices

Efficient sparse matrix computations critically rely on effective utilization of sparsity and other structural information. Known regularity of the sparsity pattern can be taken advantage of when representing a sparse matrix on modern computers. Significant savings in storage and computation can be realized, for example, for a banded sparse matrix by storing the dense diagonals as long vectors. However, matrices with irregular sparsity pattern need auxiliary storage to store the indices of the nonzero entries and necessitate indirect access to the data. Indirect access implies additional data movement which is order of magnitude slower than floating point operations. Also the irregular access pattern to the elements often results in enough cache misses (in cache-based machines) such that the overall performance of the sparse matrix algorithms on such machines can become unacceptably low.

Throughout the paper we use the following example to illustrate the data structures and algorithms for sparse matrix operations.

$$A = \begin{pmatrix} a_{11} & 0 & a_{13} & a_{14} & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 0 & a_{26} \\ a_{31} & 0 & a_{33} & 0 & a_{35} & 0 \\ 0 & a_{42} & 0 & a_{44} & 0 & 0 \\ a_{51} & 0 & a_{53} & 0 & a_{55} & 0 \\ 0 & a_{62} & 0 & a_{64} & 0 & a_{66} \end{pmatrix}$$

Figure 1: Sparsity pattern of a 6 × 6 sparse matrix.

### 2.1 Compressed Row Storage

The CRS scheme is one of the most popular data structures for representing matrices whose sparsity pattern have no known regular structure. This storage scheme can be implemented using three arrays: `value` to store the nonzero entries, `colind` that stores the column indices of the nonzero entries row-by-row, and `rowptr` that contains the index of the first nonzero element of each row of the sparse matrix stored in `colind` array.

In the CRS storage scheme the sparse matrix is compressed by moving the nonzero entries in each row to the left as shown below. The data structures to store the example matrix $A$ under the CRS scheme is shown in Figure 2.1. It consists of 3 arrays as mentioned before. Array `value` stores the nonzero entries in each row contiguously, array `colind` stores the column indices of the nonzero entries of `value`, and array `rowptr` indexes into `colind` array and stores the location of the first nonzero entry in each row of $A$. For example, the nonzero entries in row $i$ can be accessed by `value(rowptr(i)) ... value(rowptr(i+1)-1)`. Note that the CRS scheme stores only the nonzero entries of $A$. Let $nnz(A)$ denote the number of nonzero entries in matrix $A$. The storage requirement in the CRS scheme for $A \in R^{m\times n}$ is therefore

$$A = \begin{pmatrix} a_{11} & a_{13} & a_{14} & 0 & 0 & 0 \\ a_{22} & a_{26} & 0 & 0 & 0 & 0 \\ a_{31} & a_{33} & a_{35} & 0 & 0 & 0 \\ a_{42} & a_{44} & 0 & 0 & 0 & 0 \\ a_{51} & a_{53} & a_{55} & 0 & 0 & 0 \\ a_{62} & a_{64} & a_{66} & 0 & 0 & 0 \end{pmatrix}$$

Figure 2: The sparse matrix after CRS compression.

$$memory_{CRS} = 2nnz(A) + m + 1.$$

Access to the nonzero entries of $A$ is provided row-wise.

Computers that employ high-speed cache memory to improve the speed of data access relies on *reuse* of data that are brought into the high-speed memory. *Temporal*

| | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| value | $a_{11}$ | $a_{13}$ | $a_{14}$ | $a_{22}$ | $a_{26}$ | $a_{31}$ | $a_{33}$ | $a_{35}$ | $a_{42}$ | $a_{44}$ | $a_{51}$ | $a_{53}$ | $a_{55}$ | $a_{62}$ | $a_{64}$ | $a_{66}$ |

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| colind | 1 | 3 | 4 | 2 | 6 | 1 | 3 | 5 | 2 | 4 | 1 | 3 | 5 2 | 4 | 6 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| rowptr | 1 | 4 | 6 | 9 | 11 | 14 | 17 |

Figure 3: Compressed Row Storage (CRS) data structure

```
for i=1:n
    for k=rowptr(i):rowptr(i+1)-1;
        j = colind(k);
        y(i) = y(i) + val(k)*x(j);
    end
  end
```

Figure 4: Matrix-vector multiplication $y = Ax$ in CRS scheme.

*locality* is the property whereby a recently referenced data item is most likely to be referenced again in the near future while *spatial locality* ensures that the data that are stored close to each other in computer memory are most likely to be referenced together. Both of these locality properties imply reuse. The code for MVP with the CRS scheme as shown in Figure 4 depicts that the access to the elements of vector $x$ can be highly irregular resulting in poor spatial locality. A second difficulty for MVP calculation using the CRS scheme is concerned with the need for indirect access to the elements of vector $x$. To address the problem with indirect access Pinar and Heath [4] propose column reordering to make the nonzero entries in each row contiguous. Then each such block of nonzero entries can be stored as a unit such that only one indirect access is needed per block compared with one indirect access per nonzero in the CRS scheme. However, reordering of columns for arranging the nonzero entries in contiguous location is NP-hard [4]. For our example matrix the permutation vector $\begin{pmatrix} 4 & 2 & 6 & 5 & 3 & 1 \end{pmatrix}$ rearranges the columns so that most of the nonzero entries in each row become contiguous. The column reordered matrix is shown in Figure 5.

## 2.2 Compressed Column Block Storage (CCBS)

The blocking procedure described above, when implemented in conjunction with the CRS scheme, renders reduced indirect access. The implementation of this blocking scheme, however, requires an additional array (of length the number of blocks of consecutive nonzero entries in the matrix) in addition to the auxiliary storage required in the CRS scheme. The new storage scheme *Compressed Column Block Storage (CCBS)*

$$A = \begin{pmatrix} a_{14} & 0 & 0 & 0 & a_{13} & a_{11} \\ 0 & a_{22} & a_{26} & 0 & 0 & 0 \\ 0 & 0 & 0a_{35} & a_{33} & a_{31} & \\ a_{44} & a_{42} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{55} & a_{53} & a_{51} \\ 9a_{64} & a_{62} & a_{66} & 0 & 0 & 0 \end{pmatrix}.$$

Figure 5: Columns are permuted to align the nonzero entries in each row.

that we propose here for representing sparse matrices does not need to store column indices. Instead it stores the row index of each block of contiguous nonzero entries.

Two vectors (or rows) are said to be *structurally orthogonal* if they do not both contain nonzero entries in the same column position. The central idea behind CCBS scheme is to partition the rows of the sparse matrix $A$ into $p$ groups such that the rows in each group are structurally orthogonal. A group of structurally orthogonal rows can be "packed" into one dense row. Ideally, we would like to have the dense rows fully packed i.e., with no zero entries. However, in practice this is unlikely to happen so that we allow few zero entries explicitly stored. To illustrate this data structure we refer to our example matrix shown in Figure 5. A structurally orthogonal partition of the rows can be obtained as

$$\mathcal{P}_r(A) = \{\{1, 2\}, \{3, 4\}, \{5, 6\}\}$$

where the numbers denote row indices. As can be easily verified the rows in each group are structurally orthogonal and therefore can be packed into a dense row vector. The resulting compressed matrix is shown in Figure 6 where matrix $S$ represents the column

$$B = S^T A = \begin{pmatrix} a_{14} & a_{22} & a_{26} & 0 & a_{13} & a_{11} \\ a_{44} & a_{42} & 0 & a_{35} & a_{33} & a_{31} \\ a_{64} & a_{62} & a_{66} & a_{55} & a_{53} & a_{51} \end{pmatrix} \text{ where } S^T = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Figure 6: Matrix compressed via structurally orthogonal row partition.

compression defined by the row partition $\mathcal{P}_r(A)$. To represent the compressed matrix we need two auxiliary arrays: one to store the row index of each block of contiguous elements and the other to index into array `value` for the stored elements of $A$. In Figure 7 `value` contains the actual elements of $A$, `brind` holds the row index of the first entry of each "element block", and `brptr` holds the index of the first entry of the "element block" as located in `value`. Note that compressed matrix rows 1 and 2 contain two zero entries which are explicitly stored in the data structure array `value`. Sample code for computing the matrix-vector product in the new CCBS scheme is shown in Figure 8. Variable `ncolr` represents the number of row vectors in the compressed matrix.

| | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| value | $a_{14}$ | $a_{22}$ | $a_{26}$ | 0 | $a_{13}$ | $a_{11}$ | $a_{44}$ | $a_{42}$ | 0 | $a_{35}$ | $a_{33}$ | $a_{31}$ | $a_{64}$ | $a_{62}$ | $a_{66}$ | $a_{55}$ | $a_{53}$ | $a_{51}$ |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| brind | 1 | 2 | 1 | 4 | 3 | 6 | 5 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| brptr | 1 | 2 | 5 | 7 | 10 | 13 | 16 | 19 |

Figure 7: Compressed Column Block Storage (CCBS) data structure

```
blk = 1;
for l=1:ncolr;
  j=1;
  while j <= n
       i=brind(blk);
       for k=bptr(blk):bptr(blk+1)-1;
            y(i) = y(i) + val(k)*x(j);
            j = j+1;
       end
   blk = blk+1;
 end;
end
```

Figure 8: Matrix-vector multiplication $y = Ax$ in CCBS scheme.

The effectiveness of the CCBS storage scheme depends on the two preprocessing steps: ordering of the columns so that the nonzero entries in each row are contiguous and compressing the columns by computing a structurally orthogonal row partition. For each block of elements in a compressed matrix row we need to store its row index. Therefore, it is important that the column reordering step minimizes the number of blocks of contiguous nonzero entries. As noted earlier this is a computationally hard problem. The goal of the column compression step is to partition the rows into structurally orthogonal groups that minimizes the number of zero entries in the rows of the compressed matrix. Unfortunately, the general problem of structurally orthogonal partitioning of rows is also NP-hard [3]. However, for both the preprocessing steps good heuristics are available [3, 4].

# 3 Numerical Experiments

In this section we report very preliminary experimental results using the CCBS scheme for computing MVP. The experiments have been performed with MATLAB 7.0.X on Apple PowerBook G4 running OS-X operating system.

In our numerical testing we have used square matrices of dimension 10000 with random sparsity pattern. The number of row groups in the structurally orthogonal partition have been fixed at 10. The number of blocks per compressed row are varied between 500 to 8000. Table 1 displays timing results for the two storage schemes. In the table the column labeled "Blocks/Row" denotes the number of blocks of consecutive nonzero entries per compressed matrix row and the column labeled "CCBS/CRS" displays the ratio of the running times of CCBS and CRS schemes.

Table 1: Running Time For MVP calculation in CRS and CCBS Schemes.

| *Blocks/Row* | CCBS/CRS |
|---|---|
| 500 | .71 |
| 1000 | .62 |
| 3000 | .66 |
| 5000 | .82 |
| 8000 | .81 |

The running times reported are the averages over 5 random sparsity patterns at each block size. It is clear that the CCBS scheme outperforms the CRS scheme. We also emphasize that the CRS scheme used in the experiments are actually blocked CRS scheme. Therefore, CCBS is expected to be much more efficient compared with ordinary CRS storage.

# 4 Concluding Remarks

We have presented a new storage scheme for sparse matrices that exploits the sparsity information to pre-compress the matrix before generating the data structure for storing the nonzero entries and the sparsity pattern. Although experimental testing on practical test problems has not been realized, the limited test results do indicate that this storage scheme is very promising. The CCBS scheme exhibits superior temporal and spatial locality and avoids indirect referencing of the vectors. However, more thorough theoretical analysis and extensive numerical testing are needed before full potential of this new sparse matrix representation can be appreciated. Furthermore, investigation into appropriate column ordering and row partitioning heuristics suitable for problems encountered in practice constitutes another important research direction. All of the above mentioned research questions are currently being studied.

## Acknowledgements

This research was partially supported by NSERC (discovery grant) and the University of Lethbridge.

## References

[1] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition.* SIAM, Philadelphia, PA, 1994.

[2] P. N. Brown and Y. Saad. Hybrid Krylov methods for nonlinear systems of equations. *SIAM J. Sci. Stat. Comput.*, 11(3):450–481, 1990.

[3] T. F. Coleman and J. J. Moré. Estimation of sparse Jacobian matrices and graph coloring problems. *SIAM J. Numer. Anal.*, 20(1):187–209, 1983.

[4] Ali Pinar and Michael T. Heath. *Improving performance of sparse matrix-vector multiplication. Proceedings of the 1999 ACM/IEEE conference on Supercomputing (CDROM)*, 1999

[5] Y. Saad. *Iterative Methods for Sparse Linear Systems, 2nd Edition.* SIAM, Philadelphia, PA, 2003.

# An asymptotic expansions for numerical solution of linear Differential-Algebraic Equations

**M. M. Hosseini and Y. Taherinasab**

Department of Mathematics, Yazd University, Yazd, Iran.
Emails: Hosse_m@yazduni.accuracy.ir, Yasser_t_n@yahoo.com

**Abstract**

Here, it is attempt to present an efficient analytical and numerical method for solving linear differential algebraic equations(DAEs). The modified form of asymptotic expansion is found to be fast and accurate. First we calculate matrix $D_K$ such that if $\det D_K \neq 0$ for every integer $K$ then we can achieve asymptotic expansion of linear DAEs, which gives an arbitrary order for solving linear DAEs. The analysis is accompanied by numerical example.

Keywords: Asymptotic expansions, Differential algebraic equations.

## 1. Introduction

Consider the linear DAEs with variable matrix coefficients in the $C^m$ space

$$A(t)\dot{X}(t) + B(t)X(t) = f(t), \qquad (1)$$

with an initial condition

$$\lim_{t\to 0} X(t) = X_0 \qquad t \in S, \qquad (2)$$

where $S = \{t \in C, 0 < |t| < t_0, \alpha < \arg t < \beta, -\pi/2 \le \alpha \le \beta \le \pi/2\}$ is a sector of variable in complex plane with a corner in zero ( $t_0$-some positive constant ) and $A, B \in R^{m\times m}$ and $f \in R^m$ are functions of $t$. Suppose that $A(t), B(t)$ and $f(t)$ have the following asymptotic expansion on $S$ [5]

$$f(t) \approx \sum_{r=0}^{\infty} f_r t^r \qquad t \to 0 \qquad t \in S,$$

$$\mathrm{A(t)} \approx \sum_{\mathrm{r=0}}^{\infty} \mathrm{A_r t^r} \qquad \mathrm{t} \to 0 \qquad t \in S\,, \tag{3}$$

$$B(t) \approx \sum_{r=0}^{\infty} B_r t^r \qquad \mathrm{t} \to 0 \qquad t \in S\,,$$

and holomorphic in sector $S$ . Let`s find the solution in the such form

$$X(t) = \sum_{r=0}^{\infty} X_r t^r \qquad \mathrm{t} \to 0 \qquad t \in S\,, \tag{4}$$

where power series $\sum_{r=0}^{\infty} X_r t^r$ satisfied the equation formally. It means that the power series in (3) after inserting instead of $X(t)$ into equation (1) leads to the linear system of the algebraic equation

$$D_K X_K = E_K\,, \tag{5}$$

where $D_K$ and $E_K$ are constant matrices ( $\det D_K \neq 0$ ) solving the equation of (5) repeating the above procedure for higher order terms ( $K = 1,2,3,\cdots$ ). We can get the arbitrary order power series of the solution for (1). This method will be very efficient when the coefficient matrices be analytic. In this article, the focus of study is on problems that are analytic on $S$ .

## 2. Analysis of the method

Consider the linear DAEs of the form (1) that $A(t)$ is singular ( $\det A = 0$ ) and $B(t)$ is arbitrary matrix. At first we rewrite $A(t)$ such that all of the nonzero rows be up and zero rows be down such that $\det A_1 \neq 0$ ,

$$\begin{array}{r} p\{ \\ m-p\{ \end{array} \begin{bmatrix} A_1 & A_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} + \begin{bmatrix} B_1 & B_2 \\ B_3 & B_4 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} f_1(t) \\ f_2(t) \end{bmatrix}. \tag{6}$$

Here, if $\det A_1 = 0$ then by using elementary operators on $A(t)$, we can obtain $\mathrm{A_1(t)}$ such that $\det A_1 \neq 0$ .

In (6), $A_1$ and $B_1$ are $p \times p$ -matrices, $A_2$ and $B_2$ are $p \times (m-p)$ -matrices, $B_3$ is $(m-p) \times p$ -matrix, $B_4$ is $(m-p) \times (m-p)$ -matrix, $f_1(t)$ is $p \times 1$-matrix and $f_2(t)$ is $(m-p) \times 1$ -matrix. Here from (6), we have

$$A_1 \dot{X}_1 + A_2 \dot{X}_2 + B_1 X_1 + B_2 X_2 = f_1 \tag{7.a}$$

$$B_3 X_1 + B_4 X_2 = f_2 \tag{7.b}$$

and with substituting (3) and (4) into (7), we have

$$\left(\sum_{r=0}^{\infty} A_{1r} t^r\right)\left(\sum_{r=1}^{\infty} r X_{1r} t^{r-1}\right) + \left(\sum_{r=0}^{\infty} A_{2r} t^r\right)\left(\sum_{r=1}^{\infty} r X_{2r} t^{r-1}\right) +$$
$$\left(\sum_{r=0}^{\infty} B_{1r} t^r\right)\left(\sum_{r=0}^{\infty} r X_{1r} t^{r}\right) + \left(\sum_{r=0}^{\infty} B_{2r} t^r\right)\left(\sum_{r=0}^{\infty} X_{2r} t^{r}\right) = \left(\sum_{r=0}^{\infty} f_{1r} t^r\right) \qquad (8.a)$$

and

$$\left(\sum_{r=0}^{\infty} B_{3r} t^r\right)\left(\sum_{r=0}^{\infty} X_{1r} t^{r}\right) + \left(\sum_{r=0}^{\infty} B_{4r} t^r\right)\left(\sum_{r=0}^{\infty} X_{2r} t^{r}\right) = \left(\sum_{r=0}^{\infty} f_{2r} t^r\right). \qquad (8.b)$$

Because the coefficients of $1, t, t^2, \ldots$ are equal in two side of equations (8.a) and (8.b), so,

$$A_{10} X_{11} + A_{20} X_{21} + B_{10} X_{10} + B_{20} X_{20} = f_{10}.$$
$$2A_{10} X_{12} + A_{11} X_{11} + 2A_{20} X_{22} + A_{21} X_{21} + B_{10} X_{11} + B_{11} X_{10} + B_{20} X_{21} + B_{21} X_{20} = f_{11}.$$
$$3A_{10} X_{13} + 2A_{11} X_{12} + A_{12} X_{11} + 3A_{20} X_{23} + 2A_{21} X_{22} + A_{22} X_{21} + B_{10} X_{12} + B_{11} X_{11} + B_{12} X_{10}$$
$$+ B_{20} X_{22} + B_{21} X_{21} + B_{22} X_{20} = f_{12}$$

. (9.a)

.

.

and

$$B_{30} X_{10} + B_{40} X_{20} = f_{20}.$$
$$B_{30} X_{11} + B_{31} X_{10} + B_{40} X_{21} + B_{41} X_{20} = f_{21}. \qquad (9.b)$$
$$B_{30} X_{12} + B_{31} X_{11} + B_{32} X_{10} + B_{40} X_{22} + B_{41} X_{21} + B_{42} X_{20} = f_{22}.$$

.

.

.

Now we can obtain the coefficient $t^{K-1}$ in (8.a) as below

$$\sum_{i=0}^{K} (K - i + 1)\left[A_{1i} X_{1(K-i+1)} + A_{2i} X_{2(K-i+1)}\right] + \left(B_{1i} X_{1(K-i)} + B_{2i} X_{2(K-i)}\right) = f_{1(K-1)}. \qquad (10.a)$$

In the same way, from (8.b) the coefficient of $t^K$ is

$$\sum_{i=0}^{K} \left(B_{3i} X_{1(K-i)} + B_{4i} X_{2(K-i)}\right) = f_{2K} \qquad (10.b)$$

We can write two equations (10.a) and (10.b) in the matrix form as below

$$\begin{bmatrix} KA_{10} & KA_{20} \\ B_{30} & B_{40} \end{bmatrix}\begin{bmatrix} X_{1K} \\ X_{2K} \end{bmatrix} = \begin{bmatrix} f_{1(K-1)} \\ f_{2K} \end{bmatrix} - \sum_{i=1}^{K}\begin{pmatrix} (K-i)A_{1i} + B_{1(i-1)} & (K-i)A_{3i} + B_{2(i-1)} \\ B_{3i} & B_{4i} \end{pmatrix}\begin{bmatrix} X_{1(K-i)} \\ X_{2(K-i)} \end{bmatrix}, \qquad k \geq 1 \tag{11}$$

Now consider the two matrices of the form

$$D_K = \begin{bmatrix} KA_{10} & KA_{20} \\ B_{30} & B_{40} \end{bmatrix}$$

and

$$E_K = \begin{bmatrix} f_{1(K-1)} \\ f_{2K} \end{bmatrix} - \sum_{i=1}^{K}\begin{pmatrix} (K-i)A_{1i} + B_{1(i-1)} & (K-i)A_{3i} + B_{2(i-1)} \\ B_{3i} & B_{4i} \end{pmatrix}\begin{bmatrix} X_{1(K-i)} \\ X_{2(K-i)} \end{bmatrix}.$$

By considering (11), we have

$$D_K X_K = E_K . \tag{12}$$

Here if $D_K$ be a nonsingular for every integer $K$ then the Tylor series of $X_K$ is exist for every integer $K$.

**Example:** Consider for $0 \leq t \leq 1$, the DAEs of the form

$$\begin{pmatrix} 1 & -t & t^2 \\ 0 & 1 & -t \\ 0 & 0 & 0 \end{pmatrix}\begin{pmatrix} v_1' \\ v_2' \\ v_3' \end{pmatrix} + \begin{pmatrix} 1 & -(1+x) & x^2+2x \\ 0 & -1 & x-1 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \sin(t) \end{pmatrix} \tag{13}$$

with initial condition

$v(0) = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$. The exact solution is

$$\begin{aligned} v_1(t) &= e^{-t} + te^{t} \\ v_2(t) &= e^{t} + t\sin(t) \\ v_3(t) &= \sin(t) \end{aligned} \tag{14}$$

By using the Taylor expansion of $A(t)$, $B(t)$ and $f(t)$, we have

$$\left(\begin{array}{cc} \overbrace{\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}}^{A_{10}} & \overbrace{\begin{pmatrix}0\\ 0\end{pmatrix}}^{A_{20}} \\ 0 \quad 0 & 0 \end{array}\right), \left(\begin{array}{cc} \overbrace{\begin{pmatrix}0 & -1\\ 0 & 0\end{pmatrix}}^{A_{11}} & \overbrace{\begin{pmatrix}0\\ -1\end{pmatrix}}^{A_{21}} \\ 0 \quad 0 & 0 \end{array}\right), \left(\begin{array}{cc} \overbrace{\begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix}}^{A_{12}} & \overbrace{\begin{pmatrix}1\\ 0\end{pmatrix}}^{A_{22}} \\ 0 \quad 0 & 0 \end{array}\right) \qquad (15)$$

which for every $i \geq 3$, $A_{1i} = 0$ , $A_{2i} = 0$ . In addition,

$$\left(\begin{array}{cc} \overbrace{\begin{pmatrix}1 & -1\\ 0 & -1\end{pmatrix}}^{B_{10}} & \overbrace{\begin{pmatrix}0\\ 1\end{pmatrix}}^{B_{20}} \\ \underbrace{\begin{pmatrix}0 & 0\end{pmatrix}}_{B_{30}} & \underbrace{\begin{pmatrix}1\end{pmatrix}}_{B_{40}} \end{array}\right), \left(\begin{array}{cc} \overbrace{\begin{pmatrix}0 & -1\\ 0 & 0\end{pmatrix}}^{B_{11}} & \overbrace{\begin{pmatrix}2\\ 1\end{pmatrix}}^{B_{21}} \\ \underbrace{\begin{pmatrix}0 & 0\end{pmatrix}}_{B_{31}} & \underbrace{\begin{pmatrix}0\end{pmatrix}}_{B_{41}} \end{array}\right), \left(\begin{array}{cc} \overbrace{\begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix}}^{B_{12}} & \overbrace{\begin{pmatrix}1\\ 0\end{pmatrix}}^{B_{22}} \\ \underbrace{\begin{pmatrix}0 & 0\end{pmatrix}}_{B_{32}} & \underbrace{\begin{pmatrix}1\end{pmatrix}}_{B_{42}} \end{array}\right) \qquad (16)$$

which for every $i \geq 3$, $B_{1i} = 0$ , $B_{2i} = 0$ , $B_{3i} = 0$, $B_{4i} = 0$ .
In the same way for $f(t)$,

$$\left(\begin{array}{c} \overbrace{\begin{pmatrix}0\\ 0\end{pmatrix}}^{f_{10}} \\ \underbrace{\begin{pmatrix}0\end{pmatrix}}_{f_{20}} \end{array}\right), \left(\begin{array}{c} \overbrace{\begin{pmatrix}0\\ 0\end{pmatrix}}^{f_{11}} \\ \underbrace{\begin{pmatrix}1\end{pmatrix}}_{f_{21}} \end{array}\right), \left(\begin{array}{c} \overbrace{\begin{pmatrix}0\\ 0\end{pmatrix}}^{f_{12}} \\ \underbrace{\begin{pmatrix}0\end{pmatrix}}_{f_{22}} \end{array}\right), \left(\begin{array}{c} \overbrace{\begin{pmatrix}0\\ 0\end{pmatrix}}^{f_{13}} \\ \underbrace{\begin{pmatrix}-1/6\end{pmatrix}}_{f_{23}} \end{array}\right), \qquad (17)$$

such that for every $i \geq 4$, $f_{1i} = 0$, $f_{2i} = 0$ . From (12),(15),(16) and (17) we have

$D_K = \begin{pmatrix} K & 0 & 0\\ 0 & K & 0\\ 0 & 0 & 1 \end{pmatrix}$ for every $K \geq 1$ , such that $\det D_K \neq 0$ .

Now from (12) and $v(0)$ for $K = 1$,

$$\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}\begin{pmatrix}x\\ y\\ z\end{pmatrix} = \begin{pmatrix}0\\ 0\\ 1\end{pmatrix} - \begin{pmatrix}1 & -1 & 0\\ 0 & -1 & -1\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}1\\ 1\\ 0\end{pmatrix}$$

By Solving this equation, we obtain $v(1) = \begin{pmatrix}0\\ 1\\ 1\end{pmatrix}$.

By repeating above procedure for $K = 2$ with $v(0), v(1)$, we have

$$\begin{pmatrix}2 & 0 & 0\\ 0 & 2 & 0\\ 0 & 0 & 2\end{pmatrix}\begin{pmatrix}x\\ y\\ z\end{pmatrix} = \begin{pmatrix}0\\ 0\\ 0\end{pmatrix} - \begin{pmatrix}1 & -2 & 0\\ 0 & -1 & -2\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}0\\ 1\\ 1\end{pmatrix} - \begin{pmatrix}0 & -1 & 2\\ 0 & 0 & 1\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}1\\ 1\\ 0\end{pmatrix}$$

Therefore $v(2)$ is achieved as $v(2)=\begin{pmatrix}3/2\\3/2\\0\end{pmatrix}$. In the same way we obtain $v(3)=\begin{pmatrix}1/3\\1/6\\-1/6\end{pmatrix}$.

It must be noted that we can calculate every power of series for every $K$.
For instance with $\mathrm{K}=3$, we have

$$V(t)=\begin{pmatrix}1\\1\\0\end{pmatrix}+\begin{pmatrix}0\\1\\1\end{pmatrix}t+\begin{pmatrix}3/2\\3/2\\0\end{pmatrix}t^2+\begin{pmatrix}1/3\\1/6\\-1/6\end{pmatrix}t^3$$

which it exactly is the first three term of Taylor series of solution.

**Remark:** In this method if matrix $D$ is ill condition then the Taylor series may be not convergent to exact solution. In this case it is prefer to use partial or complete pivoting to get a better solution.

## References

1. E. Babolian, M. M. Hosseini, A modified spectral method for numerical solution of ordinary differential equations with non-analytic solution, , J. Comput. Appl. Math. 137 ,2003, **pp.**151-160.

2. P. Henrici, Applied computational complex analysis, Vol. 1, Wiley, New York, 1974 (chapter 1).

3. H.Hirayama, Arbitrary order and A-stable numerical method for solving algebraic ordinary differential equation by power series, in: International Cofference on Mathemathics and Computers in Physics, Vouliagmeni, Athena, Greece, July 9-16, 2000.

4. Sergiy Nesenenko, Asymptotic expansions for solutions of Linear Differential-Algebraic Equations.

5. W. Wasow, Asymptotic expansions for Ordinary Differential Equations, J. Wiley and Sons, New York-London-Sidney, 1965, **pp.**462.

6. K. E. Brenan, S. L. Campbell, L. R. Petzold, Numerical solution of initial-value problem in differential-algebraic equations, North-Holland,Amsterdam,1989.

7. Ercan Celik, Mustafa Bayram, On the numerical solution of differential-algebraic equations by Pade series, J. Comput. Appl. Math. 137 ,2003, **pp.**151-160.

8. Y. E. Boyarintsev. Regular and singular of Linear Ordinary Differential Equations, Novosibirsk, 1980, (Russian).

9. G. Corliss, Y. F. Chang, Solving Ordinary Differential Equations using Taylor series, ACM Trans. Math. Soft. 8 ,1982, **pp.**114-144.

10. Yasutaka Sibuya, Some Global Properties of Matrices of Functions of one Variable, Math. Annalen 161, Hf. 1, 1965, **pp.** 67.

11. F. R. Gantmacher. The Theory of Matrices. Vol. II, Chelsea, New York, 1964.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# Linearly-Implicit methods applied to a chemotaxis model

**Britta Janssen**[1]

[1] *Department of Mathematical Science, University of Wisconsin-Milwaukee,*
*Milwaukee, WI, 53212, USA*

emails: `jansse24@uwm.edu`

**Abstract**

We consider a mathematical model describing the bacterial growth of *E. coli* and *S. typhiurium* when abandoned to chemicals. These bacteria form very interesting patterns which can be simulated by using a system of three coupled differential equations. This non-linear system of partial differential equations presents a challenge to accurate computation due to the presence of advection and widely varying diffusion coefficients, as well as initial data with jumps.

We have used a straightforward idea, using in space finite differences and in time a linearly-implicit approach to solve the non-linear coupled system. The diffusion term is considered implicitly, whereas the nonlinear hyperbolic and reaction parts are treated explicitly.

*Key words: Linearly-Implicit – Bacterial growth – Chemotaxis – Pattern formation*

## 1 Introduction

Many organisms (bacteria for example) show a random walk if a chemical is present. That is the density at each point changes slightly due to motion. This is called chemotaxis. Often this movement results in interesting spatial patterns. To explore these patterns mathematical models of the biological process may be considered and investigated through computational simulations. Chemotactic behavior is determined by a density-dependent diffusion term and is highly nonlinear. Hence, realistic models are too difficult to be solved analytically, and we turn to numerical methods.

The models for this chemotaxis are time-dependent systems of partial differential equations. Here we consider a system in two dimensions which contains three distinct processes:

- reaction terms (for example: growth and death of cells)

- diffusion terms (random movement under some physical influence)

- chemotaxis terms (directed motion in response to chemical concentration gradient)

The reaction term models the interaction between different components, e.g. consumption of nutrients, growth or death of cells, release of chemoattractant, etc. The diffusion term models a random walk each component performs. The chemotaxis term models the directed motion of a component in response to the concentration gradient of another component.

Additionally the initial data may be nonsmooth. In particular, the initial data has jump discontinuities. This causes problems for many numerical methods which generate oscillations on points where the data is discontinuous [6, 11, 15, 17].

We consider a mathematical model of a bacterial growth process. Here the initial data are discontinuous. Numerical schemes often develop inaccuracies for solving parabolic partial differential equations with nonsmooth data, e.g. high frequency components. A well known property of parabolic partial differential equations is the smoothing property. That is, for any positive time their solution is infinitely differentiable even if the initial data is nonsmooth. Numerical methods are also expected to have an analogous property. But numerical methods often develop inaccuracies when the initial data is discontinuous. For more details, refer to [7, 8, 9, 10].

## 2 Bacteria and Model

Escherichia coli (*E. coli*) and Salmonella typhimurium (*S. typhiurium*) are the bacteria considered in this paper. *E. coli* can, for example, be found in the human intestine and *S. typhimurium* in poultry if it is incompletely cooked. Both physical mechanisms of movement are essentially the same, as was first studied by H.C. Berg in 1983 [3]. These bacteria put themselves in motion by means of long hairlike appendages known as flagella. If these appendages all rotate counter-clockwise, they join together, moving the bacteria forward  known as "run." If the rotation is clockwise, instead of a forward movement it turns around irregularly in one spot  known as "tumbling."

If a chemical is present the motion of the bacteria may be directed preferentially towards lower or higher concentration of the chemical. If the bacteria moves preferentially towards the lower concentration of the chemical, the chemical is called a chemorepellent, otherwise a chemoattractant. The only difference between a chemoattractant and a chemorepellent is the direction of motion. Therefore, for simplicity, only the case of chemoattractant is considered in this paper.

In 1966 J. Adler [1, 2] performed experiments with *E. coli.* In laboratories, both *E. coli* and *S. typhimurium* have been observed to form interesting one-dimensional and two-dimensional patterns.

In the two-dimensional experiment a petri dish was equally covered with the same chemoattracting "food." The bacteria was set in the middle of the petri dish as a high density inoculum. Here the bacteria split into two parts, one which remained at the center of the petri dish and another one which moved outwards, forming an expanding ring.

If there was more than one kind of food, all chemoattracting to the bacteria, the bacteria split into more than two parts. Here, one remained at the center of the petri dish and the others formed expanding rings, each consuming one kind of food. The number of rings matches the number of foods on the petri dish.

In 1991 Budrene and Berg [4] showed that a colony of *E. coli* or *S. typhimurium* form interesting and regular patterns when abandoned to, or feeding on, intermediates of the tricarboxylic acid (TCA) cycle. They discovered that succinate and fumarate have the strongest effects. This substance we call the "stimulant" since it is the initiator of the pattern. If the bacteria is exposed to or feeds on TCA it secretes aspartate, which is a potent chemoattractant.

Budrene and Berg performed two different kinds of experiments, one where the bacteria were placed in a liquid medium and the other where the bacteria were placed on a semi-solid substrate (0.24% water agar/nutrient). Here we will only consider the pattern formed in the semi-solid experiments. In these experiments a high density inoculum of bacteria was placed on a petri dish. The petri dish contained a uniform distribution of the stimulant in 0.24% water agar. The stimulant is in this case also the main food for the bacteria. After a few days the bacteria had been through 25 to 40 generations. During this time the bacteria spread out from the inoculum and covered then the whole petri dish with a stationary pattern. This pattern consists of high density aggregates with small regions of nearly zero cell density. The patterns formed by *E. coli* are more complex than the pattern of *S. typhimurium* which forms concentric rings. These rings are either continuous or spotted. *E. coli* patterns include sunflower spirals, radial spots, radial stripes and chevrons.

For both experiments (*E. coli* and *S. typhimurium*), initially succinate is distributed uniformly throughout the medium and an inoculum of bacteria is put at the center of the medium. For *E. coli* a very low density bacterial population forms, which then spreads outward from the initial inoculum. Within this bacterial population high density rings of bacteria are seen. For *S. typhimurium* a swarm ring (high density ring of energetically agile bacteria) forms and disperses outwards from the initial inoculum. The bacterial density in this swarm ring increases until a special point, when it becomes unstable and a percentage of the bacteria are left behind as aggregates which remain full of energetically agile bacteria for a short period of time and then disband as the bacteria combines again with the swarm ring. A clump of bacteria is left behind in the aggregates original location; this is non-motile.

All patterns have the same building blocks of bacteria, aspartate, i.e. chemoattractant and succinate, i.e. stimulant. An important role have the following biological processes:

- diffusion of bacteria, aspartate and succinate
- proliferation of bacteria (includes reproduction and death of cells)
- secretion and uptake of aspartate by the bacteria
- consumption of succinate

- chemotaxis of the cells up gradients of aspartate

There is no actual death of cells, but some cells become non-motile and stop participating in forming the pattern.

If we combine all these processes into a mathematical model, we derive a system of three partial differential equations of the form

$$\boxed{\begin{array}{l}\text{rate of change}\\\text{of cell density}\\\text{n}\end{array}} = \boxed{\begin{array}{l}\text{diffusion}\\\text{of cells}\end{array}} + \boxed{\begin{array}{l}\text{chemotaxis}\\\text{of cells to}\\\text{aspartate}\end{array}} + \boxed{\begin{array}{l}\text{growth and}\\\text{death of cells}\end{array}}$$

$$\boxed{\begin{array}{l}\text{rate of change}\\\text{of aspartate}\\\text{concentration c}\end{array}} = \boxed{\begin{array}{l}\text{diffusion}\\\text{of}\\\text{aspartate}\end{array}} + \boxed{\begin{array}{l}\text{production}\\\text{of aspartate}\\\text{by cells}\end{array}} - \boxed{\begin{array}{l}\text{uptake of}\\\text{aspartate by}\\\text{cells}\end{array}}$$

$$\boxed{\begin{array}{l}\text{rate of change}\\\text{of succinate}\\\text{concentration s}\end{array}} = \boxed{\begin{array}{l}\text{diffusion}\\\text{of}\\\text{succinate}\end{array}} - \boxed{\begin{array}{l}\text{uptake of}\\\text{succinate by}\\\text{cells}\end{array}}$$

Here $n$ denotes the bacterial cell density, $c$ the aspartate concentration and $s$ the succinate concentration.

Using functional forms for the box terms and expressions special for *E. coli* and *S. typhimurium* which were also determined by experiments, we arrive at the following mathematical model

$$\begin{aligned}\frac{\partial n}{\partial t} &= D_n \nabla^2 n - \alpha \nabla \left[ \frac{n}{(1+c)^2} \nabla c \right] + \rho n \left( \delta \frac{s^2}{1+s^2} - n \right) \\ \frac{\partial c}{\partial t} &= D_c \nabla^2 c + \beta s \frac{n^2}{\gamma + n^2} - nc \\ \frac{\partial s}{\partial t} &= D_s \nabla^2 s - \kappa n \frac{s^2}{1+s^2} \qquad (1)\end{aligned}$$

where $\alpha, \beta, \gamma, \delta, \rho$ and $\kappa$ are experimentally determined parameters. The domain $\Omega$ is assumed to be compact. For more details, refer to [12, 14].

The initial condition is given as

$$\begin{aligned} n(x,y,0) &= \begin{cases} u_0 & \text{if } (x-x_0)^2 + (y-y_0)^2 \le r \\ 0 & \text{otherwise} \end{cases} \\ c(x,y,0) &= 0 \\ s(x,y,0) &= s_0 \end{aligned}$$

where $r$ denotes the width of the initial inoculum and $(x_0, y_0)$ its center.

The boundary conditions are assumed to be

$$\begin{aligned} n(x,y,t) &= 0 \\ c(x,y,t) &= 0 \\ s(x,y,t) &= s_0 \end{aligned}$$

on the boundary $\delta\Omega$. The domain $\Omega$ is assumed to be large enough such that the bacteria never reaches the boundary.

## 3 Linearly-Implicit Approach

Mixed implicit-explicit methods are useful in applications as chemical reactions, population dynamics, biological environments, and enzyme reactions. There exist already software for Linearly-implicit one-step methods, e.g. RODAS with LU-decomposition [5] and ROWMAP [16], which uses Krylov-techniques for the solution of huge problems which arise from semi-discretization of parabolic equations.

Various numerical methods are available to solve reaction-diffusion systems. One could use, for instance, an implicit method that is unconditionally stable. This would allow us to choose a larger time step for solving the system compared to an explicit method. If one of the diffusion coefficients is very small or one of the reaction kinetics is highly non-linear these methods may be very slow. If we use explicit methods and one of the diffusion coefficients is huge, or one of the reaction parts is very stiff, this may lead to spurious solutions and we may have a severe restriction on the time step. That means we receive solutions which are dramatically different from the true solution and if we choose a time step which does not satisfy the restriction we even have an instable numerical method. Another idea could be a predictor-corrector approach where the predictor consists of an explicit method to get an initial guess for the corrector which is an implicit method. But also this approach is in our case very slow.

Our model is given by (1)

$$\begin{aligned}
\frac{\partial n}{\partial t} &= D_n\nabla^2 n - \alpha\nabla\left[\frac{n}{(1+c)^2}\nabla c\right] + \rho n\left(\delta\frac{s^2}{1+s^2} - n\right)\\
\frac{\partial c}{\partial t} &= D_c\nabla^2 c + \beta s\frac{n^2}{\gamma + n^2} - nc\\
\frac{\partial s}{\partial t} &= D_s\nabla^2 s - \kappa n\frac{s^2}{1+s^2}
\end{aligned}$$

with $x \in (0, X), y \in (0, Y)$ and $t \geq 0$ where $\nabla^2$ represents the Laplacian in two dimensions. The grid spacing in x- and y-direction will be denoted by $\Delta x$ and $\Delta y$, respectively. We divide the length of the x-domain and y-domain into $N_x$ and $N_y$ partitions, respectively, with $X = N_x\Delta x$ and $Y = N_y\Delta y$. We denote the time by $\Delta t$.

Discretizing this model with the approach of a linearly-implicit method results in the following if we assume $\Delta x = \Delta y$ and $N_x = N_y$, so also $X = Y$ :

$$\begin{aligned}
\frac{u_{i,j}^{k+1} - u_{i,j}^{k}}{\Delta t} &= D_n \frac{u_{i,j+1}^{k+1} + u_{i,j-1}^{k+1} + u_{i+1,j}^{k+1} + u_{i-1,j}^{k+1} - 4u_{i,j}^{k+1}}{\Delta x^2} \\
&\quad - \frac{\alpha}{\Delta x^2} \left[ \left( \frac{u_{i,j+1}^{k} - u_{i,j}^{k}}{(1+v_{i,j}^{k})^2} - 2u_{i,j}^{k} \frac{v_{i,j+1}^{k} - v_{i,j}^{k}}{(1+v_{i,j}^{k})^3} \right) \left( v_{i,j+1}^{k} - v_{i,j}^{k} \right) \right. \\
&\qquad + \left( \frac{u_{i+1,j}^{k} - u_{i,j}^{k}}{(1+v_{i,j}^{k})^2} - 2u_{i,j}^{k} \frac{v_{i+1,j}^{k} - v_{i,j}^{k}}{(1+v_{i,j}^{k})^3} \right) \left( v_{i+1,j}^{k} - v_{i,j}^{k} \right) \\
&\qquad \left. + \frac{u_{i,j}^{k}}{(1+v_{i,j}^{k})^2} \left( v_{i,j+1}^{k} + v_{i,j-1}^{k} + v_{i+1,j}^{k} + v_{i-1,j}^{k} - 4v_{i,j}^{k} \right) \right] \\
&\quad + \rho\, u_{i,j}^{k} \left( \delta \frac{(w_{i,j}^{k})^2}{1+(w_{i,j}^{k})^2} - u_{i,j}^{k} \right) \\
\frac{v_{i,j}^{k+1} - v_{i,j}^{k}}{\Delta t} &= D_c \frac{v_{i,j+1}^{k+1} + v_{i,j-1}^{k+1} + v_{i+1,j}^{k+1} + v_{i-1,j}^{k+1} - 4v_{i,j}^{k+1}}{\Delta x^2} \\
&\quad + \beta w_{i,j}^{k} \frac{(u_{i,j}^{k})^2}{\gamma + (u_{i,j}^{k})^2} - u_{i,j}^{k} v_{i,j}^{k} \\
\frac{w_{i,j}k + 1 - w_{i,j}^{k}}{\Delta t} &= D_s \frac{w_{i,j+1}^{k+1} + w_{i,j-1}^{k+1} + w_{i+1,j}^{k+1} + w_{i-1,j}^{k+1} - 4w_{i,j}^{k+1}}{\Delta x^2} \\
&\quad - \kappa\, u_{i,j}^{k} \frac{(w_{i,j}^{k})^2}{1+(w_{i,j}^{k})^2}
\end{aligned}$$

where the linear diffusion terms are discretized implicitly and the non-linear hyperbolic part as well as the non-linear reaction terms are discretized explicitly. Here $u_{i,j}^{k}, v_{i,j}^{k}$, and $w_{i,j}^{k}$ are numerical solutions of $n$, $c$, and $s$ at $x = i\Delta x, i = 1, \dots, N_x - 1, y = j\Delta y, j = 1, \dots, N_y - 1$ and $t = k\Delta t, k > 0$, respectively.

This can also be rewritten in the general form

$$(I - r_l A_l) v^{k+1} = v^k + \Delta t F_l(t_k, x, u^k, v^k) + \Delta t b$$

with

$$r_1 = D_n \frac{\Delta t}{\Delta x^2} \qquad r_2 = D_c \frac{\Delta t}{\Delta x^2} \qquad r_3 = D_s \frac{\Delta t}{\Delta x^2}.$$

The matrix $A_l \in \mathbb{R}^{(N-1)\times(N-1)}$ with $N - 1 = (N_x - 1)(N_y - 1)$ and $N_x = N_y$ is pentagonal

$$A_l = \begin{pmatrix} D_l & -r_l I & & & \\ -r_l I & Dl & -r_l I & & \\ & \ddots & \ddots & \ddots & \\ & & -r_l I & D_l & -r_l I \\ & & & -r_l I & D_l \end{pmatrix}$$

with $D_l \in \mathbb{R}^{(N_x-1)\times(N_y-1)}$ given as

$$D_l = \begin{pmatrix} 1+4r_l & -r_l & & & \\ -r_l & 1+4r_l & -r_l & & \\ & \ddots & \ddots & \ddots & \\ & & -r_l & 1+4r_l & -r_l \\ & & & -r_l & 1+4r_l \end{pmatrix}$$

and $I \in \mathbb{R}^{(N_x-1)\times(N_y-1)}$ is the identity matrix.

We do not have variations in $a_{1,2}$ and $a_{N+1,N}$ since we have no derivative boundary conditions. The vector $b$ contains the boundary conditions.

We get three independent systems. Since the matrices are not time dependent, we can calculate the LU-decomposition of the matrices once and use the decomposition to solve the linear system we get in each time step.

To calculate the LU-decomposition of our matrices we used the UMFPACK library (Version 2.2), which stands for "Unsymmetric-pattern MultiFrontal PACKage." This software is freely available at the webpage

**http://www.cise.ufl.edu/research/sparse/umfpack/**

for educational and non-commercial purposes. It is a package for solving sparse linear systems, $Ax = b$, where $A$ is sparse and can be unsymmetric. It is written in ANSI Fortran 77. There are options for choosing a good pivot order, factorizing a subsequent matrix with the same pivot order and nonzero pattern as a previously factorized matrix, and solving systems of linear equations with the factors. Iterative refinement, with sparse backward error estimates, can also be performed. Single and double precision, complex, and complex double precision (complex*16) routines are available.

For using the routines only the non-zero elements of the matrix must be stored which is a huge difference than storing all values of the matrix. In our case, you only need to store $(4 + 3(N_x - 2))$ for each matrix $D \in \mathbb{R}^{(N_x-1)\times(N_y-1)}$ and $2N_x(N_x - 1)$ for all identity-matrices, that is in total $N_x(4 + 3(N_x - 2)) + 2N_x(N_x - 1)$ instead of $(N-1)^2 = (N_x - 1)^2(N_y - 1)^2$ elements.

# 4 Results

We did numerical simulations of the model (1). Once the simulations were done by using a program written by Rebecca Tyson, who kind supplied the source code. She considered exactly the same model in her PhD-thesis [12]. She used the CLAWPACK-software which is freely available at the webpage

**http://www.amath.washington.edu/~rjl/clawpack.html** .

Afterwards, we carried out the same simulations, on the same computational platform, with our program. The input parameters were exactly the same.

The bacteria is assumed to be in the middle of the petri dish. The coordinates $(x_0, y_0)$ are given in the input file together with the length of the domain in x- and y-direction. The length is given as 30 cm in both direction. The center for the bacteria was given as $(x_0, y_0) = (10, 10)$.

The cell density $u$ was set to the initial concentration $u_0 = 2$ within a radius of width $= 2$ cm from the center. The concentration of the chemoattractant (the food of the bacteria) $v$ was set to zero and the succinate concentration $w$ was set to $w_0 = 1$. After some time there will be chemoattractant since the cells produce the chemoattractant. The experiment was simulated until the end time $t_{end} = 5s$ was reached. The maximal time was taken not so large that the bacteria never reaches the domain boundary. This has the advantage of allowing simple zero boundary conditions in the program. Together with the initial data $u$, $v$, and $w$ were printed into file eleven times. The coefficient were set as seen in Table 1. We used the same coefficient as Rebecca Tyson does in her paper [13].

| | | |
|---|---|---|
| diffusion coefficient for cells | $D_u$ | 0.25 |
| diffusion coefficient for chemoattractant | $D_v$ | 1.0 |
| diffusion coefficient for succinate | $D_w$ | 1.0 |
| chemotaxis coefficient | $\alpha$ | 2.25 |
| production of chemoattractant | $\beta$ | 0.2 |
| saturation of production of chemoattractant | $\gamma$ | 1.0 |
| carrying capacity or yield coefficient | $\delta$ | 20.0 |
| consumption of food | $\kappa$ | 0.0 |
| growth rate for cells | $\rho$ | 0.01 |

Table 1: Input parameters

Since no analytical solution is known, we ran Rebecca Tyson's program on a very fine grid and considered this as the exact solution. We ran her program as well as ours with different values for $\Delta x$ and $\Delta t = \frac{1}{2}\Delta x$. Afterwards, we measured the errors in the infinity-norm/$L_2$-norm and calculated the rates as

$$r = \frac{\ln \frac{e_i}{e_{i+1}}}{\ln \frac{h_i}{h_{i+1}}}$$

where $e_i$ is the error in the $i^{th}$ iteration and $h_i$ the step is the chosen spatial step width $\Delta x$ in the $i^{th}$ iteration.

In Table 2 we list the results from our program, and in Table 3 the results calculated with the Tyson program.

| $\Delta x$ | $\Delta t = \frac{1}{2}\Delta x$ | $\lVert\cdot\rVert_1$ | rates | $\lVert\cdot\rVert_{L_2}$ | rates |
|---|---|---|---|---|---|
| 1 | 0.5 | 10.4279 | | 11.1848 | |
| 0.5 | 0.25 | 2.5598 | 2.0264 | 2.0361 | 2.4577 |
| 0.25 | 0.125 | 2.3785 | 0.1060 | 1.3058 | 0.6409 |
| 0.125 | 0.0625 | 2.2985 | 0.0494 | 0.8958 | 0.5437 |

Table 2: Errors and rates calculated with our program depending on the chosen values for $\Delta x$ and $\Delta t$

| $\Delta x$ | $\Delta t = \frac{1}{2}\Delta x$ | $\lVert\cdot\rVert_1$ | rates | $\lVert\cdot\rVert_{L_2}$ | rates |
|---|---|---|---|---|---|
| 1 | 0.5 | 10.4097 | | 11.2187 | |
| 0.5 | 0.25 | 2.4958 | 2.0604 | 1.9682 | 2.5110 |
| 0.25 | 0.125 | 2.0315 | 0.2970 | 1.0685 | 0.8813 |
| 0.125 | 0.0625 | 1.7279 | 0.2335 | 0.6307 | 0.7606 |

Table 3: Errors and rates calculated with Rebecca Tyson's program depending on the chosen values for $\Delta x$ and $\Delta t$

The rates are similar. They oscillate, which is common for the nonsmooth data case. We used only a first-order accurate method for the discretization in time and space whereas they used a second-order accurate method. Therefore, our rates are slightly lower. The error in the $L_2$-norm is better than the error in the $\infty$-norm which is due to the oscillations. Therefore the rates in the $L_2$-norm are better. Rebecca Tyson claims in her PhD-Thesis [12] that “the convergence rate obtained for these solutions is a satisfactory 2.22 for the cell density.” Here she used different parameters then those listed in Table 1.

We also compared the CPU time needed for each program with the specified parameters. The simulations were carried out on a Gateway notebook with an Intel Pentium Dual-Core processor with 1024 MB memory.

| $\Delta x$ | $\Delta t = \frac{1}{2}\Delta x$ | CPU time |
|---|---|---|
| 1 | 0.5 | 0.140625 |
| 0.5 | 0.25 | 0.359375 |
| 0.25 | 0.125 | 2.406250 |
| 0.125 | 0.0625 | 21.484375 |

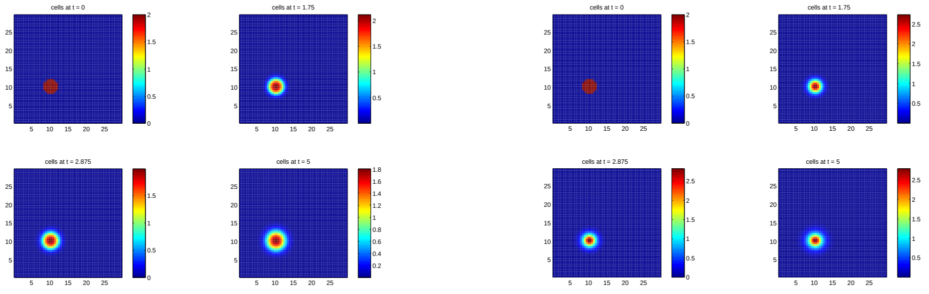
Table 4: CPU time needed by our program depending on the chosen values for $\Delta x$ and $\Delta t$

| $\Delta x$ | $\Delta t = \frac{1}{2}\Delta x$ | CPU time |
|---|---|---|
| 1 | 0.5 | 0.234375 |
| 0.5 | 0.25 | 0.375000 |
| 0.25 | 0.125 | 2.078125 |
| 0.125 | 0.0625 | 16.875000 |

Table 5: CPU time needed by Rebecca Tyson's program depending on the values for $\Delta x$ and $\Delta t$

Her program is faster, but ours is much less complicated to understand and to implement. Additionally our program is easier to generalize to a similar model or to extend if the model changes. This has been the purpose of the present investigation.

If we plot our results for the cell density $u$, we get the pictures in Figure 1(a) with our program and the pictures in Figure 1(b) with her program. For these plots we used 60 output points in each direction, that is in total 360,000 points.



(a) Results for the cell density $u$ at time $t = 0, t = 1.75, t = 2.875$, and $t = 5$ with our program

(b) Results for the cell density $u$ at time $t = 0, t = 1.75, t = 2.875$, and $t = 5$ with Rebecca Tyson's program

Figure 1: Results comparing our program and the Tyson program

## References

[1] J. Adler, *Chemotaxis in bacteria* , Science **153** (1966) 708-716.

[2] J. Adler, *Chemotaxis in bacteria* , Ann. Rev. Biochem. **44** (1975) 341-356.

[3] S.M. Block, J.E. Segall, and H.C. Berg, *Adaptation kinetics in bacterial chemotaxis J Bacteriol.* , J. Bacteriol. **154(1)** (1983) 312-323.

[4] E.O. Budrene and H.C. Berg, *Complex patterns formed by motile cells of escherichia coli* , Nature **349(6310)** (1991) 630-633.

[5] E. Hairer, G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems* , Springer-Verlag, Berlin, 1991.

[6] R.J. LeVeque, *Wave propagation algorithms for multi-dimensional hyperbolic systems* , J. Comput. Phys. **131** (1997) 327–353.

[7] M. Luskin and R. Rannacher, *On the Smoothing Property of the Crank-Nicholson Scheme* , Appl. Anal. **14** (1982) 117–135.

[8] M. Luskin and R. Rannacher, *On the Smoothing Property of the Galerkin Method for Parabolic Equations* , SIAM J. Numer. Anal. **19** (1982) 93–113.

[9] R. Rannacher, *Finite Element Solution of Diffusion Problems with Irregular Data* , Numer. Math. **43** (1984) 309–327.

[10] R. Rannacher, *Discretization of the Heat Equation with Singular Initial Data* , Zeit. Ang. Math. Meth. (ZAMM) **62** (1982) 346–348.

[11] G.D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods* , Oxford University Press, New York, 1985.

[12] R.C. Tyson, *Pattern formation by E. coli - mathematical and numerical investigation of a biological phenomenon* , PhD thesis, University of Washington, 1996.

[13] R.C. Tyson, L.G. Stern and R.J. LeVeque, *Fractional step methods applied to a chemotaxis model* , J. Math. Biol. **41** (2000) 455–475.

[14] R.C. Tyson, S.R. Lubkin, and J.D. Murray, *A minimal mechanis for bacterial pattern formation* , Proceedings of the Royal Society B: Biological Sciences **266(1416)** (1999) 299-304.

[15] B.A. Wade, A.Q.M. Khaliq, M. Siddique, and M. Yousuf, *Smoothing with Positivety-Preserving Padé Schemes for Parabolic Problems with Nonsmooth data Numerical Methods for Partial Differential Equations* , Wiley Interscience **21(3)** (2005) 553-573.

[16] R. Weiner, B.A. Schmitt, H. Podhaisky, *ROWMAP - a ROW-code with Krylov techniques for large stiff ODEs* , Appl. Num. Math. **25** (1997) 303–319.

[17] M. Yousuf, *Smoothing schemes for the inhomogeneous linear and semilinear parabolic problems with nonsmooth data* , PhD thesis, University of Wisconsin - Milwaukee, 2005.

# Solving stiff second order initial value problems directly by Backward Differentiation Formulas

Samuel N. Jator
Department of Mathematics
Austin Peay State University
Clarksville, TN 37044, U. S. A
Email: Jators@apsu.edu

May 17, 2007

### Abstract

In this paper, we propose a family of Backward Differentiation Formulas (BDFs)for the direct solution of the general stiff second order initial value problems (IVPs) of the form $y'' = f(x, y, y')$. The method is derived by the interpolation and collocation of the assumed approximate solution and its second derivative at $x = x_{n+j}$ ,$j = 1, 2, ..., k - 1$ and $x = x_{n+k}$ respectively, where k is the step number of the method. The interpolation and collocation procedures lead to a system of (k+1) equations, which are solved to to determine the unknown coefficients. The resulting coefficients are used to construct the approximate continuous solution from which Multiple Finite Difference Methods (MFDMs) are obtained and simultaneously applied to provide a direct solution to IVPs. Two specific methods for $k = 2$ and $k = 3$ are used to illustrate the process. Numerical examples are given to show the efficiency of the method.

**AMS Subject Classification : 65L05, 65L06, 65L12**

**Key Words:** Backward Differentiation Formulas, Collocation, Interpolation, Second Order, Initial Value Problem, Multiple Finite Difference Methods

# 1 Introduction

The general second-order ordinary differential equation of the form

$$y'' = f(x, y, y'), \tag{1}$$

$$y(a) = y_0, \; y'(a) = \delta_0$$

is encountered in several areas of engineering and science such as circuit theory, control theory, chemical kinetics, and biology. In practice, problem (1) is solved by first reducing it to a system of first-order differential equations and then applying the various methods available for solving systems of first order IVPs. This approach is extensively discussed in the literature and in this paper we cite just a few notable ones such as Lambert[8], [9], Brugnano and Trigiante[2], Onumanyi et al [11], [10], Fatunla[4]. In particular, BDFs are well known for their effectiveness in solving stiff IVPs Gear[5] . Although there has been tremendous success with this approach, it has certain draw backs. For instance, computer programs associated with the methods are often complicated especially when incorporating subroutines to supply the starting values for the methods resulting in longer computer time and more human effort.

In the past decades, considerable attention has been devoted to solving (1) directly without first reducing it to a system of first order differential equations. In particular, the focus has mostly been concentrated on solving directly the special form of (1) given by $y'' = f(x, y)$ since it occurs frequently in mechanical systems without dissipation. In this area, Twizell and Khaliq[13], proposed a class of p-stable two-step higher-derivative formulas for the special second-order initial-value problems. Yusuph and Onumanyi[14] proposed two new LMMs of order 4 associated with the standard Numerov method which were applied to solve the special form of (1) directly. We also cite the works of Simos[12], Fatunla[4], and Lambert[8].

Several methods have also been proposed in the literature for solving the general form (1) directly without first reducing it to an equivalent first-order system. For instance, Hairer and Wanner[6] proposed Nystrom type methods for (1) and stated order conditions for determining the parameters of the methods. Other methods of the Runge-Kutta type considered in the literature for solving (1) are due to Chawla and Sharma[3] and Henrinci[7]. Awoyemi[1] considered multiderivative methods of the LMM type and stated that the direct application of those methods to non-stiff problems of the form (1) saves computer time and human effort. In Awoyemi[1] the methods were implemented as predictor - corrector methods and the Taylor Series algorithm was used to supply the starting values. While the methods yielded good results, the disadvantage in this approach is that the Taylor series algorithm involves higher order partial derivatives that are tedious to obtain.

In this paper, we propose a family of BDFs of steps $k = 2$ and $k = 3$ which are implemented on stiff IVPs of the form (1) without the need for either predictors or starting values from other methods.

The paper is organized as follows. In section two, we derive an approximation for $y(x)$ which is continuous. Section three is devoted to the specification of the methods and how the MFDMs are obtained and simultaneously applied to (1). Numerical examples are given in section four to show the efficiency of the method. Finally, the conclusion of the paper is discussed in section five.

## 2 The derivation method

We seek an approximation of the form

$$y(x) \approx \overline{y}(x) = \sum_{j=0}^{k} \iota_j \Upsilon_j(x) \tag{2}$$

where $\Upsilon_j(x)$ 's are assumed polynomial basis functions and $\iota_j$ 's are unknown coefficients to be determined by imposing the following conditions.

$$\overline{y}\left(x_{n+j}\right) = y_{n+j}, j = 0, \cdots, k-1 \tag{3}$$

$$\overline{y}''\left(x_{n+k}\right) = f_{n+k} \tag{4}$$

Hence, equations (3) and (4) lead to a system of (k+1) equations which is solved to obtain $\iota_j$ 's .Our k-step continuous BDF is constructed by substituting the the values of $\iota_j$ 's into equation (2). After some manipulation, our method is expressed in the form

$$\overline{y}(x) = \sum_{j=0}^{k-1} \alpha_j(x) y_{n+j} + h^2 \beta_k(x) f_{n+k} \tag{5}$$

which is applied directly to provide the solution to (1). In particular, we seek a solution on

$\pi_N : a = x_0 < x_1 < x_2 < ... < x_n < x_{n+1} < ... < x_N = b$

$h = x_{n+1} - x_n$ , n = 0, 1, . . . N

where $\pi_N$ is a partition of $[a, b]$ and $h$ is the constant step-size of the partition of $\pi_N$.

In the next section, we specify the methods.

# 3 Specification of the Methods

We obtain 2 methods for $k = 2$ and $k = 3$ with the following specifications:

**Case: k=2**

$\Upsilon_j(x) = x^j$ , $i = 0, \cdots, 2$.

We also express $\alpha_j(x)$ and $\beta_j(x)$ as functions of t where $t = (x - x_{n+1})/h$ in what follows:

$\alpha_0(t) = -t$ ; $\alpha_1(t) = (1 + t)$

$\beta_2(t) = \frac{1}{2}(t + t^2)$

The following 2-step BDF is obtained by evaluating (5) at $x = x_{n+2}$

$y_{n+2} - 2y_{n+1} + y_n = h^2 f_{n+2}$

An additional Equation and derivatives are obtained by imposing that

$\overline{y'}(x) = \delta(x)$ , $\overline{y'}(a) = \delta_0$

In particular, to start the initial value problem for $n = 0$, we obtain the following equation from $\overline{y'}(a) = \delta_0$.

$h\delta_0 = y_{n+1} - y_n - \frac{h^2}{2} f_{n+2}$

It is worth noting that the derivatives are provided by $\delta(x_{n+\tau}), \tau = 1, 2$ as follows:

$h\delta_{n+1} = -y_n + y_{n+1} + \frac{h^2}{2} f_{n+2}$

$h\delta_{n+2} = -y_n + y_{n+1} + \frac{3h^2}{2} f_{n+2}$

**Case: k=3**

$\Upsilon_j(x) = x^j$ , $i = 0, \cdots, 3.$

We also express $\alpha_j(x)$ and $\beta_j(x)$ as functions of t where $t = (x - x_{n+2})/h$ in what follows:

$\alpha_0(t) = \frac{1}{12}(4t + 3t^2 - t^3)$ ; $\alpha_1(t) = \frac{1}{6}(-10t - 3t^2 + t^3)$

$\alpha_2(t) = \frac{1}{12}(12 + 16t + 3t^2 - t^3)$ ; $\beta_3(t) = \frac{h^2}{12}(2t + 3t^2 + t^3)$

The following 3-step BDF is obtained by evaluating (5) at $x = x_{n+3}$

$$2y_{n+3} - 5y_{n+2} + 4y_{n+1} - y_n = h^2 f_{n+3} \qquad (6)$$

Equation (6) can be used with the following additional methods:

$-2y_n + 5y_{n+1} - 4y_{n+2} + y_{n+3} = -h^2 f_n$

which is obtained from (5) by numbering the grid points from the right to the left with a negative step-size and

$h\delta_0 = -\frac{5}{3}y_n + \frac{7}{3}y_{n+1} - \frac{2}{3}y_{n+2} + \frac{h^2}{6}f_{n+3}$

is obtained from $\overline{y'}(a) = \delta_0$.

It is worth noting that the derivatives are provided by $\delta(x_{n+\tau}), \tau = 1, \cdots, 3$ as follows:

$h\delta_{n+1} = -\frac{5}{12}y_n - \frac{1}{6}y_{n+1} + \frac{7}{12}y_{n+2} - \frac{h^2}{12}f_{n+2}$

$h\delta_{n+2} = \frac{1}{3}y_n - \frac{5}{3}y_{n+1} + \frac{5}{3}y_{n+2} + \frac{h^2}{6}f_{n+3}$

$h\delta_{n+3} = \frac{7}{12}y_n - \frac{13}{6}y_{n+1} + \frac{19}{12}y_{n+2} + \frac{11h^2}{12}f_{n+3}$

It is vital to note that the resulting specific case of $\overline{y}(x)$ evaluated at $x_{n+k}$ can be used as a continuous numerical integrator directly and singly in the conventional way on overlapping sub-intervals. However, a better approach is to derive MFDMs from the evaluation of $\overline{y}(x)$ and the first derivative function $\delta(x)$ at specified points. The MFDMs obtained are applied to simultaneously provide values for $y_1, \cdots, y_k$ without looking for any other methods to provide the starting values. Hence, this is an improvement over the use of (5) evaluated at $x_{n+k}$ singly for IVPs. We proceed by explicitly obtaining initial conditions at $x_{n+k}, n = 0, k, \cdots, N-k$ using the computed values $\overline{y}(x_{n+k}) = y_{n+k}$ and $\delta(x_{n+k}) = \delta_{n+k}$ over sub-intervals $[x_0, x_k], \cdots, [x_{N-k}, x_N]$ which do not overlap (see [14]). In addition to providing the starting values, the method also provides an accurate approximation to $y'(x)$ using $\delta(x)$.

# 4 Numerical examples

In this section, we give 2 examples to illustrate the efficiency of the methods. We find absolute errors of the approximate solution in $\pi_N$. The computations were carried out using our written Mathematica code in Mathematica 5.2 (see tables 1 to 4) .

**Example 4.1.**

$$y'' + 1001y' + 1000y = 0, y(0) = 1, \ y'(0) = -1$$

$$Exact : y(x) = e^{-x}$$

**Example 4.2.**

$$y'' + 102y' + 200y = 0, y(0) = 1, \ y'(0) = -2$$

$$Exact : y(x) = e^{-2x}$$

| $x$ | $y$ | $\overline{y}$ | $Error$ |
|---|---|---|---|
| 0.0 | 1.0000000000 | 1.0000000000 | 0.000000 |
| 0.1 | 0.904837418035959473 | 0.904543399638336964 | $2.94018 \times 10^{-4}$ |
| 0.2 | 0.818730753077981887 | 0.818173598553347147 | $5.57155 \times 10^{-4}$ |
| 0.3 | 0.740818220681717853 | 0.740066941293423763 | $7.51279 \times 10^{-4}$ |
| 0.4 | 0.670320046035639371 | 0.669399772406962778 | $9.20274 \times 10^{-4}$ |
| 0.5 | 0.606530659712633379 | 0.605501145850122934 | $10.29514 \times 10^{-4}$ |
| 0.6 | 0.5488116360940265 | 0.547685220660996208 | $11.26415 \times 10^{-4}$ |
| 0.7 | 0.496585303791409593 | 0.495405051428370057 | $11.80252 \times 10^{-4}$ |
| 0.8 | 0.449328964117221563 | 0.448101587862691275 | $12.27376 \times 10^{-4}$ |
| 0.9 | 0.406569659740599131 | 0.405327333668655853 | $12.42326 \times 10^{-4}$ |
| 1.0 | 0.367879441171442334 | 0.366624888659087 | $12.54553 \times 10^{-4}$ |

Table 1: Absolute Errors, $\|y - \overline{y}\|$, for Example 4.1, Case $k = 2$, where $y(x) = e^{-x}$

| $x$ | $y$ | $\overline{y}$ | $Error$ |
|---|---|---|---|
| 0.0 | 1.0000000000 | 1.0000000000 | 0.000000 |
| 0.1 | 0.818730753077981887 | 0.816535433070866112 | $2.19532 \times 10^{-3}$ |
| 0.2 | 0.670320046035639371 | 0.666141732283464538 | $4.17831 \times 10^{-3}$ |
| 0.3 | 0.5488116360940265 | 0.543474486948973822 | $5.33715 \times 10^{-3}$ |
| 0.4 | 0.449328964117221563 | 0.443189286378572688 | $6.13968 \times 10^{-3}$ |
| 0.5 | 0.367879441171442334 | 0.361550989243710718 | $6.32845 \times 10^{-3}$ |
| 0.6 | 0.301194211912202147 | 0.294824454215837548 | $6.36976 \times 10^{-3}$ |
| 0.7 | 0.246596963941606528 | 0.240514319493515893 | $6.08264 \times 10^{-3}$ |
| 0.8 | 0.201896517994655377 | 0.196125223275308577 | $5.77129 \times 10^{-3}$ |
| 0.9 | 0.165298888221586555 | 0.159996554702243365 | $5.30233 \times 10^{-3}$ |
| 1.0 | 0.135335283236612702 | 0.130467702915348038 | $4.86758 \times 10^{-3}$ |

Table 2: Absolute Errors, $\|y - \overline{y}\|$, for Example 4.2, Case $k = 2$, where $y(x) = e^{-2x}$

| $x$ | $y$ | $\overline{y}$ | $Error$ |
|---|---|---|---|
| 0.0 | 1.0000000000 | 1.0000000000 | 0.000000 |
| 0.1 | 0.904837418035959473 | 0.904848530444377274 | $1.11124 \times 10^{-5}$ |
| 0.2 | 0.818730753077981887 | 0.818788243555019512 | $5.74905 \times 10^{-5}$ |
| 0.3 | 0.740818220681717853 | 0.740910321998192245 | $9.21013 \times 10^{-5}$ |
| 0.4 | 0.670320046035639371 | 0.670360829978287053 | $4.07839 \times 10^{-5}$ |
| 0.5 | 0.606530659712633379 | 0.606505357852192261 | $2.53019 \times 10^{-5}$ |
| 0.6 | 0.5488116360940265 | 0.548764377514921797 | $4.72586 \times 10^{-5}$ |
| 0.7 | 0.496585303791409593 | 0.496566369066338086 | $1.89347 \times 10^{-5}$ |
| 0.8 | 0.449328964117221563 | 0.449371845425690263 | $4.28813 \times 10^{-5}$ |
| 0.9 | 0.406569659740599131 | 0.406649327717074182 | $7.96680 \times 10^{-5}$ |
| 1.0 | 0.367879441171442334 | 0.367908853058919316 | $2.94119 \times 10^{-5}$ |
| 1.1 | 0.332871083698079539 | 0.332826522546983571 | $4.45612 \times 10^{-5}$ |
| 1.2 | 0.301194211912202147 | 0.301119953271358253 | $7.42586 \times 10^{-5}$ |

Table 3: Absolute Errors, $\|y - \overline{y}\|$, for Example 4.1, Case $k = 3$, where $y(x) = e^{-x}$

| $x$ | $y$ | $\overline{y}$ | $Error$ |
|---|---|---|---|
| 0.0 | 1.0000000000 | 1.0000000000 | 0.000000 |
| 0.1 | 0.818730753077981887 | 0.818894645941277943 | $1.63893 \times 10^{-4}$ |
| 0.2 | 0.670320046035639371 | 0.671157167530224008 | $8.37121 \times 10^{-4}$ |
| 0.3 | 0.5488116360940265 | 0.550155440414506724 | $13.43804 \times 10^{-4}$ |
| 0.4 | 0.449328964117221563 | 0.449968231809354257 | $6.39268 \times 10^{-4}$ |
| 0.5 | 0.367879441171442334 | 0.367517875200228694 | $3.61566 \times 10^{-4}$ |
| 0.6 | 0.301194211912202147 | 0.300437595640151045 | $7.56616 \times 10^{-4}$ |
| 0.7 | 0.246596963941606528 | 0.246312289173206311 | $2.84675 \times 10^{-4}$ |
| 0.8 | 0.201896517994655377 | 0.202533535807732123 | $6.37018 \times 10^{-4}$ |
| 0.9 | 0.165298888221586555 | 0.166444586543129613 | $11.45698 \times 10^{-4}$ |
| 1.0 | 0.135335283236612702 | 0.135829954803814967 | $4.94672 \times 10^{-4}$ |
| 1.1 | 0.110803158362333875 | 0.11023920371426179 | $5.63955 \times 10^{-4}$ |
| 1.2 | 0.0907179532894125273 | 0.0896631588239585397 | $10.54795 \times 10^{-4}$ |

Table 4: Absolute Errors, $\|y - \overline{y}\|$, for Example 4.2, Case $k = 3$, where $y(x) = e^{-2x}$

# 5 Conclusions

We have proposed Two specific methods for $k = 2$ and $k = 3$ for solving stiff second-order IVPs directly without first adapting the second order IVP to an equivalent first order system. An essential ingredient in the method involves the way in which it is applied. For instance, we proceed by explicitly obtaining initial conditions at $x_{n+k}, n = 0, k, ..., N - k$ using the computed values $\overline{y}(x_{n+k}) = y_{n+k}$ and $\delta(x_{n+k}) = \delta_{n+k}$ over sub-intervals $[x_0, x_k], \cdots, [x_{N-k}, x_N]$ which do not overlap. We have demonstrated the efficiency of the methods on two stiff problems and the results are given in tables 1 and 2 (case k = 2) and tables 3 and 4 (case k = 3). Our future research will be focused on studying the stability properties of these methods and applying them as Boundary value methods to solve both initial and boundary value problems.

# References

[1] Awoyemi, D. O., 1999, A class of continuous methods for general second order initial value problems in ordinary differential equations,*Inter. J. Compt. Maths*, 72, 1-9, 29-37.

[2] Brugnano L. and D. Trigiante, D., 1998, Solving Differential Problems by Multitep Initial and Boundary Value Methods, Gordon and Breach Science Publishers, Amsterdam, 1998, 280-299.

[3] Chawla, M. M. and Sharma S. R., 1985, Families of Three-Stage Third Order Runge-Kutta-Nystrom Methods for $y'' = f(x, y, y')$ , Journal of the Australian Mathematical Society 26, 375-386.

[4] Fatunla, S.O., 1988, Numerical Methods for InitialValue Problems in Ordinary Differential Equation (NewYork: Academic Press Inc.).

[5] Gear, C.W., (1971a) Algorithm 407, DIFSUB for solution of Ordinary Differential Equations. Comm. ACM., vol 14, pp. 185  190.

[6] Hairer E. and Wanner G., 1976, A Theory for Nystrom methods, Numerische Mathematik 25, 383 - 400.

[7] Henrici, P.,1962, Discrete Variable Methods for ODEs, John Wiley, New York, USA.

[8] Lambert, J. D.,1991, Numerical methods for ordinary differential systems, John Wiley, New York.

[9] Lambert, J. D,1973 Computational methods in ordinary differential equations, John Wiley, New York.

[10] Onumanyi, P., Sirisena, U. W. and Jator, S.N., 1999, Continuous finite difference approximations for solving differential equations,*Inter. J. Compt. Maths. 72*, 1, 15-27( 80).

[11] Onumanyi,P., Awoyemi, D. O., Jator, S. N. and Sirisena,U. W., 1994, New linear mutlistep methods with continuous coefficients for first order initial value problems, *J. Nig. Math. Soc. 13*, 37-51(7).

[12] Simos, T. E., 2002, Dissipative Trigonometrically-Fitted Methods for Second Order IVPs with oscillating Solution, Int. J. Mod. Phys. 13, (10), 1333-1345.

[13] Twizell, E.H. and Khaliq, A.Q.M., 1984, Multiderivative methods for periodic IVPs. SIAM Journal of Numerical Analysis, 21, 111121.

[14] Yusuph, Y. and Onumanyi, P., 2005, New Multiple FDMs through multistep collocation for $y'' = f(x, y)$, Proceedings of the conference organized by the National Mathematical Center, Abuja Nigeria.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# A uniformly convergent B-spline collocation technique on a non-uniform mesh for solving singularly perturbed turning point problem exhibiting two boundary layers

**Mohan K. Kadalbajoo**[1] **and Vikas Gupta**[1]

[1] *Department of Mathematics & Statistics, Indian Institute of Technology Kanpur, India.*

emails: `kadal@iitk.ac.in`, `vicky@iitk.ac.in`

### Abstract

A numerical method is devised for solving singularly perturbed turning point boundary value problems having two boundary layers. The proposed method is composed of B-spline collocation on piecewise uniform mesh of Shishkin type. Some theoretical bounds are given for the derivative of analytical solution. The method is shown to be unconditionally stable and accurate of order $O(\Delta x)^2$. An extensive amount of analysis has been carried out to prove the uniform convergence with respect to singular perturbation parameter. Several numerical experiments have been included to support the theoretical results and to demonstrate the efficiency of the method.

*Key words: singular perturbation, turning point, B-spline collocation, Shishkin mesh, boundary layer, uniform convergence*
*MSC 2000: 65L10, 65L20*

## 1 Introduction

Singularly perturbed problems arises in various field of Physics and Engineering such as fluid flows at high Reynolds numbers, heat and mass transfer with high peclet numbers [10][12], drift-diffusion equation of semiconductor device modeling [17] and magneto-hydrodynamics duct problems at high Hartman numbers [9]. The solution of singular perturbation problems possesses steep gradients in narrow layer regions of the domain depending upon the nature of the convection coefficient, as the singular perturbation parameter $\varepsilon$ approaches zero. Therefore in order to tackle such oscillations we need to derive a method using a class of special piecewise-uniform mesh, called Shishkin mesh, where half the mesh points are concentrated in the layer regions. For further discussions, reader may refer to Doolan *et. al.* [5] Farrell *et. al.* [6] Miller *et. al.* [14] and Roos *et al.* [19].

In particular, singularly perturbed turning point problems received much attention in the literature due to the complexity involved in finding uniformly valid asymptotic expansions unlike non-turning point problems. In this paper, we consider the following singularly perturbed two-point boundary value problem with a turning point at $x = 0$:

$$Lu(x) \equiv \varepsilon u''(x) + a(x)u'(x) - b(x)u(x) = f(x), \qquad x \in \Omega = (-1, 1), \tag{1.1a}$$

$$u(-1) = A, \qquad u(1) = B, \tag{1.1b}$$

where $\varepsilon$ is a small perturbation parameter satisfying $0 < \varepsilon << 1$, $A$ and $B$ are given constants, $a, b$ and $f$ are sufficiently smooth functions. Moreover,

$$a(0) = 0, \qquad a'(0) \leq 0. \tag{1.2}$$

$$|a(x)| \geq a_0 > 0, \qquad 0 < |x| \leq 1. \tag{1.3}$$

$$b(x) \geq 0, \qquad b(0) > 0. \tag{1.4}$$

$$b(x) \geq b_0 > 0, \qquad x \in \bar{\Omega} = [-1, 1], \tag{1.5}$$

$$|a'(x)| \geq \left|\frac{a'(0)}{2}\right|, \qquad x \in \bar{\Omega} = [-1, 1]. \tag{1.6}$$

Under these assumptions (1.2)- (1.6), the turning point problem (1.1) possesses a unique solution having twin boundary layers at $x = -1$ and $x = 1$ *i.e.*, at both end points [2].

Abrahmsson [1] derived a priori estimates for solution of singular perturbation problems with a turning point. Berger *et. al.*[2], Wasow [23] and O'Malley [16] studied qualitative aspects of these problems. Farrell [7] gave the sufficient conditions for uniform convergence of a difference scheme for these turning point problems. Sun and Stynes used Galerkin finite element methods on various piecewise uniform meshes for such problems [20]. Clavero *et. al.* presented a uniformly convergent finite difference method for such problems with turning points [3], whereas, Surla and Uzelac [22] solved them by taking a linear combination of the two spline difference schemes. Kadalbajoo and Patidar [11] proposed a numerical method based on cubic spline approximation with nonuniform mesh for the singularly perturbed two-point boundary value problems having a turning point. In this paper, we propose a B-spline collocation method to solve problems of type (1.1) with piecewise uniform mesh of Shishkin type.

This paper is organized as follows. In Section 2 we present some analytical results for continuous turning point problem (1.1). In Section 3 we use B-spline collocation method to solve the problem with Shishkin mesh. The derivation of uniform convergence is given in Section 4. Some numerical examples have been solved and the results are presented in Section 5. Finally, discussion and conclusion is given at the end of the paper in Section 6. Throughout the paper we use $C$ as a generic positive constant independent of $\varepsilon$ and mesh parameter.

## 2 Continuous Problem

In this section explicit bounds for the solution of turning point problem (1.1) and its derivative are derived. We divide the interval $\bar{\Omega}$ into three subintervals as $\Omega_1 = [-1, -\tau]$, $\Omega_2 = [-\tau, 1-\tau]$ and $\Omega_3 = [1-\tau, 1]$ such that $\bar{\Omega} = \Omega_1 \cup \Omega_2 \cup \Omega_3$, where $\tau$ is the width of boundary layer defined in next section.
Let us for any given function $g(x) \in C^k(\bar{\Omega})$ ($k$ a nonnegative integer), $||g||_k$ is defined by

$$||g||_k = \sum_{i=0}^{k} \max_{x\in\bar{\Omega}} |g^{(i)}(x)|.$$

**Lemma 2.1. Minimum Principle** *Let $y(x) \in C^2(\bar{\Omega})$, satisfying $y(\pm 1) \geq 0$, such that $Ly(x) \leq 0, \forall x \in \Omega$. Then $y(x) \geq 0, \forall x \in \bar{\Omega}$.*

**Lemma 2.2.** *If $u(x)$ is the solution of the problem (1.1), then $\forall \varepsilon > 0$ we have*

$$||u(x)||_0 \leq \frac{||f||_0}{b_0} + max(|A|, |B|), \qquad \forall x \in \bar{\Omega}.$$

**Proof.** Let us define $\phi(x) = ||f||_0/b_0 + \max(|A|, |B|)$. Now applying the lemma 2.1 to comparison functions $\phi(x) \pm u(x)$ we get the required estimate immediately. □

**Theorem 2.1.** *If u(x) is the solution of turning point problem (1.1) and a, b and f $\in C^m(\bar{\Omega}), m > 0$, then the bounds*

$$|u^i(x)| \leq C[1 + \varepsilon^{-i} exp(-\eta(x+1)/\varepsilon)], \quad i = 1, \cdots, m+1, \quad x \in \Omega_1,$$

$$|u^i(x)| \leq C[1 + \varepsilon^{-i} exp(-\eta(1-x)/\varepsilon)], \quad i = 1, \cdots, m+1, \quad x \in \Omega_3,$$

*are valid for any $\tau > 0$. Here $\eta$ and $C$ are generic positive constants independent of $\varepsilon$ and $x$.*

**Proof.** Following the approach given by Kellog *et. al.*[13] and Miller *et. al.*[14] , the bounds for the derivatives are obtained. □
Further we show that $u(x)$ is smooth near turning point $x = 0$ if $\beta = b(0)/a'(0) < 0$, (see Abrahamsson [1]).

**Theorem 2.2.** *Suppose $\beta < 0$. If $u(x)$ is the solution of ( 1.1) and satisfies all conditions from ( 1.2) to ( 1.6), let $a, b$ and $f \in C^m(\bar{\Omega}), m > 0$. Then we have*

$$|u^{(i)}(x)| \leq C, \quad i = 1 \cdots m, \quad \forall x \in \Omega_2,$$

*for sufficiently small $\tau > 0$.*

**Proof.** For the proof one can see [2]. □
The following theorem provides bounds for the smooth and singular components of the solution $u$ of the turning point problem (1.1).

**Theorem 2.3.** *Suppose the solution $u$ of the turning point problem (1.1) has the decomposition*

$$u = v + w,$$

*where, for all $i, 0 \leq i \leq 3$, the smooth component satisfies*

$$|v^{(i)}(x)| \leq C[1 + \varepsilon^{-(i-2)}(exp(-a_0(1+x)/\varepsilon) + exp(-a_0(1-x)/\varepsilon))], \quad \forall x \in \bar{\Omega},$$

*and the singular component satisfies*

$$|w^{(i)}(x)| \leq C\varepsilon^{-i}(exp(-a_0(1+x)/\varepsilon) + exp(-a_0(1-x)/\varepsilon)), \quad \forall x \in \bar{\Omega}.$$

**Proof.** The proof follows by approach as given in Miller *et. al.*[14] □

# 3 Discrete Problem

In this section, third-degree B-splines are used to construct collocation method to turning point problem (1.1) discussed in section 1 with nonuniform mesh $\bar{\Omega}_N$ of Shishkin type. Shishkin mesh is defined as follows:

## 3.1 Shishkin Mesh

Consider $N = 2^m$ with $m \geq 3$, be a positive integer and let $\tau$ be the width of boundary layer. Therefore, for a given $N$ and $\varepsilon$, the interval $\bar{\Omega} = [-1, 1]$ be divided into three subintervals $\Omega_1 = [-1, -1+\tau], \Omega_2 = [-1+\tau, 1-\tau]$ and $\Omega_3 = [1-\tau, 1]$ such that $\bar{\Omega} = \Omega_1 \cup \Omega_2 \cup \Omega_3$.
The transition parameter $\tau$ is given by

$$\tau \equiv \min\{\frac{1}{4}, K\varepsilon \log\ N\}, \qquad K \geq \frac{1}{\min\{a_0, b_0\}}.$$

Thus transition parameter $\tau$ depends on both $\varepsilon$ and $N$. The value of constant $K$ depends on the particular scheme being used. Define

$$\tilde{h} = \begin{cases} h_1 = h_i = 4\tau/N, & \text{if } i = 1, 2, \cdots N/4, \\ h_2 = h_i = 4(1-\tau)/N, & \text{if } i = N/4+1, \cdots 3N/4, \\ h_3 = h_i = 4\tau/N, & \text{if } i = 3N/4+1, \cdots N. \end{cases} \tag{3.1}$$

where $N$ is the no. of discretization points and the set of mesh points $\bar{\Omega}_N = \{x_i\}_{i=0}^N$ with

$$x_i = \begin{cases} -1 + (4\tau/N)i, & \text{if } i = 0, 1, 2, \cdots N/4, \\ (-1+\tau) + (4(1-\tau)/N)(i - N/4), & \text{if } i = N/4+1, \cdots 3N/4, \\ (1-\tau) + (4\tau/N)(i - 3N/4), & \text{if } i = 3N/4+1, \cdots N, \end{cases} \tag{3.2}$$

*i.e.* the finite interval $[-1, 1]$ is partitioned into $N$ finite elements by the partition $\pi : -1 = x_0 < x_1 < x_2 < \cdots < x_N = 1$, where $\tilde{h}$ is the piecewise uniform mesh spacing.

### 3.2 Methodology of B-spline Collocation

We assume $X$ is a linear subspace of $L_2(\bar{\Omega})$, the space of all square integrable functions defined on $\bar{\Omega}$. For $i = -1, 0, \ldots, N+1$, the cubic B-splines are defined by the following relation[18]

$$\phi_i(x) = \frac{1}{\tilde{h}^3} \begin{cases} (x - x_{i-2})^3, & x_{i-2} \le x \le x_{i-1}, \\ \tilde{h}^3 + 3\tilde{h}^2(x - x_{i-1}) + 3\tilde{h}(x - x_{i-1})^2 - 3(x - x_{i-1})^3, & x_{i-1} \le x \le x_i, \\ \tilde{h}^3 + 3\tilde{h}^2(x_{i+1} - x) + 3\tilde{h}(x_{i+1} - x)^2 - 3(x_{i+1} - x)^3, & x_i \le x \le x_{i+1}, \\ (x_{i+2} - x)^3, & x_{i+1} \le x \le x_{i+2}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

Each $\phi_i(x)$ is a continuously differentiable, piecewise cubic on $\pi$ and $\phi_i(x) \in X$. Reader can find detailed description of B-spline functions in [18], [15] and [21]. Now suppose the approximate solution of Eq. (1.1) is given by

$$U(x) = \sum_{i=-1}^{N+1} c_i \phi_i(x), \tag{3.4}$$

where $c_i$ are unknown real coefficients. Here we have introduced two extra cubic B-splines, $\phi_{-1}$ and $\phi_{N+1}$ to satisfy the boundary conditions. Therefore, we have

$$LU(x_i) = f(x_i), \qquad 0 \le i \le N, \tag{3.5}$$

and

$$U(x_0) = A, \qquad U(x_N) = B. \tag{3.6}$$

On solving the collocation equations (3.5), and putting the values of B-spline functions $\phi_i$ and of derivatives at mesh points $\bar{\Omega}_N$, we obtain a system of $(N+1)$ linear equations in $(N+3)$ unknowns

$$(6\varepsilon - 3a_i\tilde{h} - b_i\tilde{h}^2)c_{i-1} + (-12\varepsilon - 4b_i\tilde{h}^2)c_i + (6\varepsilon + 3a_i\tilde{h} - b_i\tilde{h}^2)c_{i+1} = f_i\tilde{h}^2, \qquad 0 \le i \le N. \tag{3.7}$$

The given boundary conditions become

$$c_{-1} + 4c_0 + c_1 = A, \tag{3.8}$$

and

$$c_{N-1} + 4c_N + c_{N+1} = B. \tag{3.9}$$

Thus by the Eqs. (3.7), (3.8) and (3.9) we obtain a $(N+3) \times (N+3)$ system with (N+3) unknowns $\{c_{-1}, c_0, \ldots, c_{N+1}\}$. Now eliminating $c_{-1}$ from first equation of (3.7) and from equation (3.8), we find

$$(-36\varepsilon + 12a_0\tilde{h})c_0 + 6a_0\tilde{h}c_1 = f_0\tilde{h}^2 - A(6\varepsilon - 3a_0\tilde{h} - b_0\tilde{h}^2). \tag{3.10}$$

Similarly, eliminating $c_{N+1}$ from the last equation of (3.7) and from (3.9), we get

$$6a_N\tilde{h}c_{N-1} + (-36\varepsilon + 12a_N\tilde{h})c_N = f_N\tilde{h}^2 - B(6\varepsilon + 3a_N\tilde{h} - b_N\tilde{h}^2). \tag{3.11}$$

Thus we are lead to a system of $(N+1)$ linear equations in $(N+1)$ unknowns

$$Tx^N = d^N. \tag{3.12}$$

Where $x^N = (c_0, c_1, \ldots, c_N)^T$ are the unknown real coefficients with right hand side $d^N = (f_0\tilde{h}^2 - A(6\varepsilon - 3a_0\tilde{h} - b_0\tilde{h}^2), f_1\tilde{h}^2, \ldots, f_{N-1}\tilde{h}^2, f_N\tilde{h}^2 - B(6\varepsilon + 3a_N\tilde{h} - b_N\tilde{h}^2))^T$. The coefficient matrix $T$ is given by

$$\begin{bmatrix}
-36\varepsilon + 12a_0\tilde{h} & 6a_0\tilde{h} & 0 & 0 & \cdots & 0 \\
6\varepsilon - 3a_1\tilde{h} - b_1\tilde{h}^2 & -12\varepsilon - 4b_1\tilde{h}^2 & 6\varepsilon + 3a_1\tilde{h} - b_1\tilde{h}^2 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\
0 & 0 & 6\varepsilon - 3a_i\tilde{h} - b_i\tilde{h}^2 & -12\varepsilon - 4b_i\tilde{h}^2 & 6\varepsilon + 3a_i\tilde{h} - b_i\tilde{h}^2 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \cdots & 0 & 6\varepsilon - 3a_{N-1}\tilde{h} - b_{N-1}\tilde{h}^2 & -12\varepsilon - 4b_{N-1}\tilde{h}^2 & 6\varepsilon + 3a_{N-1}\tilde{h} - b_{N-1}\tilde{h}^2 \\
0 & \cdots & 0 & 0 & -6a_N\tilde{h} & -36\varepsilon - 12a_N\tilde{h}
\end{bmatrix}$$

It is easily seen that collocation matrix $T$ is strictly diagonally dominant and hence nonsingular. Since $T$ is nonsingular, we can solve (3.12) for $c_0, c_1, \ldots, c_N$ and substitute into the boundary conditions (3.8) and (3.9) to obtain $c_{-1}$ and $c_{N+1}$. Hence the method of collocation using a basis of cubic B-splines applied to problem (1.1) has a unique solution $U(x)$ given by (3.4).

## 4 Derivation for Uniform Convergence

In this section we deduce that the present collocation method has a uniform convergence of order two with Shishkin mesh in maximum norm.

**Theorem 4.1.** *Let $u \in C^4[-1,1]$ be the solution of problem (1.1) and let $U$ be the cubic B-spline collocation approximate on the piecewise uniform mesh. Then for $N$ sufficiently large (independently of $\varepsilon$), we have*

$$\sup_{0<\varepsilon\le 1} ||U - u||_{\bar{\Omega}} \le CN^{-2}(\ln N)^2.$$

**Proof.** The solution $U$ of the discrete problem is decomposed in an analogous manner as that of the continuous solution u. Thus $U = V + W$, where $V$ is the solution of the inhomogeneous problem given by

$$LV = f, \quad V(-1) = v(1), \quad V(1) = v(1),$$

and $W$ is the solution of the homogeneous problem

$$LW = 0, \quad W(-1) = w(1), \quad W(1) = w(1).$$

To show the proposed collocation method is $\varepsilon$-uniform, here we consider only the subinterval $[-1,0]$. In the same way, one can obtain similar estimate in the subinterval $[0,1]$. Here we use de Boor [4] and Hall [8] spline interpolation error estimates to derive $\varepsilon$-uniform error estimate.

By using de Boor-Hall error estimates and doing some simplification we are lead to the following $\varepsilon$-uniform error estimate

$$\sup_{0<\varepsilon\leq 1} ||U - u||_{\bar{\Omega}} \leq Ch_c^2 \max_{\bar{\Omega}} |u''|, \tag{4.1}$$

where $h_c = \max\{h_1, h_2, h_3\}$. Now the $\varepsilon$-uniform convergence estimate is obtained on each subinterval $\Omega_i = (x_{i-1}, x_i), \forall i = 1, 2 \ldots N/2$, separately. Each finite subinterval $\Omega_i$ is covered by four cubic B-splines, therefore the B-spline collocation approximation $U$ of $u$, on $\Omega_i$, is given by

$$U = c_{i-2}\phi_{i-2} + c_{i-1}\phi_{i-1} + c_i\phi_i + c_{i+1}\phi_{i+1},$$

and it is obvious that on $\Omega_i$

$$|U(x)| \leq \max_{\Omega_i} |u(x)|, \tag{4.2}$$

and by above $\varepsilon$-uniform error estimate (4.1), it is easy to see that

$$|U(x) - u(x)| \leq C\tilde{h}^2 \max_{\Omega_i} |u''(x)|. \tag{4.3}$$

Now from Theorems 2.1, 2.2, and Eq. (4.3) on $\Omega_i$, we have

$$|U(x) - u(x)| \leq C\frac{\tilde{h}^2}{\varepsilon^2}. \tag{4.4}$$

Also, using Theorem 2.3, Eqs. (4.2) and (4.3), on $\Omega_i$, we have

$$\begin{aligned} |U(x) - u(x)| &= |V(x) + W(x) - v(x) - w(x)|, \\ &\leq |V(x) - v(x)| + |W(x)| + |w(x)| \\ &\leq C\tilde{h}^2 \max_{\Omega_i} |v''(x)| + 2\max_{\Omega_i} |w(x)|, \\ &\leq C(\tilde{h}^2 + e^{-a_0(1+x_i)/\varepsilon}). \end{aligned} \tag{4.5}$$

Now the required $\varepsilon$- uniform estimate depends on whether $K\varepsilon \log N \geq 1/4$ or $K\varepsilon \log N \leq 1/4$. In the first case $1/\varepsilon \leq C \log N$ and the mesh is uniform with mesh spacing $\tilde{h} = 2/N$. Therefore the $\varepsilon$- uniform estimate easily follows at once from Eq. (4.4).

In the second case $K\varepsilon \log N \leq 1/4$, therefore we have $\tau = K\varepsilon \log N$. Then $\tilde{h} = 4\tau/N$ for $i$ satisfies $1 \leq i \leq N/4$ in the boundary layer region. Therefore

$$\frac{\tilde{h}}{\varepsilon} = \frac{4\tau}{N\varepsilon} = CN^{-1}\log N, \qquad 1 \leq i \leq N/4,$$

thus the results immediately follows combining this with Eq. (4.4). On the other hand, if $i$ satisfies $N/4 < i \leq N/2$ in no boundary layer region then $\tau \leq 1 + x_i$ and so

$$e^{-a_0(1+x_i)/\varepsilon} \leq e^{-a_0\tau/\varepsilon} = e^{-a_0 K \log N} = N^{-a_0 K} = N^{-2},$$

whenever $K = 2/a_0$ in the definition of transition parameter $\tau$. Using this in (4.5), we get the required result. In a similar manner, one can obtain a similar estimate for the subinterval $[0, 1]$. Thus the method is uniformly convergent of order two in the discrete maximum norm. □

# 5 Numerical Experiments and Results

In this section, we give some numerical experiments to validate theoretical results. Both of the following examples [11] exhibits a turning point at $x = 1/2$.

**Example 1.** This example corresponds to the following singularly-perturbed turning point problem:

$$\varepsilon u''(x) - 2(2x-1)u'(x) - 4u(x) = 0, \qquad x \in (0,1), \tag{5.1a}$$

$$u(0) = 1, \qquad u(1) = 1, \tag{5.1b}$$

whose exact solution is given by

$$u(x) = e^{-2x(1-x)/\varepsilon}. \tag{5.2}$$

For every $\varepsilon$ the computed maximum pointwise errors are estimated by

$$E_\varepsilon^N = \max_{x_i \in \bar{\Omega}_N} |u(x_i) - U^N(x_i)|, \tag{5.3}$$

where $U^N$ denotes the numerical solution obtained by using $N$ finite elements. Furthermore, the $\varepsilon$- uniform order of convergence is obtained by

$$p_{\varepsilon,N} = \frac{log(E_\varepsilon^N / E_\varepsilon^{2N})}{log2}. \tag{5.4}$$

The numerical results are presented in Table 1 with piecewise uniform mesh.

**Example 2.** Now we consider the following nonhomogeneous turning point problem :

$$\varepsilon u''(x) - 2(2x-1)u'(x) - 4u(x) = 4(4x-1), \qquad x \in (0,1), \tag{5.5a}$$

$$u(0) = 1, \qquad u(1) = 1, \tag{5.5b}$$

which has the analytical solution given by

$$u(x) = -2x + 2e^{-2x(1-x)/\varepsilon} + e^{-2x(1-x)/\varepsilon} erf((2x-1)/\sqrt{2\varepsilon})/erf(1/\sqrt{2\varepsilon}), \tag{5.6}$$

whereas the maximum pointwise errors and numerical order of convergence are calculated as in Example 1. The numerical results are displayed in Table 2.

# 6 Discussions and Conclusions

We have proposed a B-spline collocation method to solve singularly perturbed two-point boundary value problems with a turning point exhibiting twin boundary layers. Numerical results presented in Table 1 and Table 2 show that for a fixed value of $\varepsilon$, the pointwise errors decrease whereas the order of convergence increases, in general, as the number of mesh points increases. It has been found that pointwise errors are minimum near to the turning point. Also, it is noticed that maximum pointwise errors are observed near the transition point due to the abrupt changes in the mesh size.

It has been seen that exact and numerical solutions with uniform mesh are identical for most of the region of the domain except in the boundary layer regions. Thus the present method is second order accurate and numerical results support the theoretical estimates. This method is shown to be uniformly convergent and independent of mesh parameters. The proposed method gives more accurate results than many of other boundary layer resolving finite difference methods. Also this method produces the solution at any point in the domain, whereas the finite difference methods gives the solution only at the chosen mesh points.

Table 1: Maximum pointwise errors and numerical order of convergence for example 1 with Shishkin mesh

| $\varepsilon \downarrow$ | N=16 | N=32 | N=64 | N=128 | N=256 | N=512 | N=1024 |
|---|---|---|---|---|---|---|---|
| $2^0$ | 3.6801E-3 | 1.0922E-3 | 2.9794E-4 | 7.7844E-5 | 1.9897E-5 | 5.0300E-6 | 1.2645E-6 |
| | 1.7525 | 1.8741 | 1.9364 | 1.9680 | 1.9840 | 1.9920 | |
| $2^{-4}$ | 4.4364E-2 | 2.7484E-2 | 1.3777E-2 | 6.4524E-3 | 2.6386E-3 | 9.0787E-4 | 3.0787E-4 |
| | 0.6908 | 0.9963 | 1.0943 | 1.2901 | 1.5392 | 1.5602 | |
| $2^{-8}$ | 4.2182E-2 | 2.6684E-2 | 1.3506E-2 | 5.9281E-3 | 2.4642E-3 | 8.9815E-4 | 3.1380E-4 |
| | 0.6606 | 0.9824 | 1.1879 | 1.2665 | 1.4561 | 1.5171 | |
| $2^{-12}$ | 4.2334E-2 | 2.6746E-2 | 1.3535E-2 | 5.9425E-3 | 2.3628E-3 | 8.9004E-4 | 3.1297E-4 |
| | 0.6625 | 0.9826 | 1.1875 | 1.3306 | 1.4086 | 1.5078 | |
| $2^{-14}$ | 4.2343E-2 | 2.6752E-2 | 1.3537E-2 | 5.9433E-3 | 2.3632E-3 | 8.9015E-4 | 3.1301E-4 |
| | 0.6625 | 0.9827 | 1.1876 | 1.3306 | 1.4086 | 1.5079 | |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $2^{-28}$ | 4.2347E-2 | 2.6754E-2 | 1.3538E-2 | 5.9437E-3 | 2.3633E-3 | 8.9016E-4 | 3.1302E-4 |
| | 0.6625 | 0.9827 | 1.1876 | 1.3306 | 1.4087 | 1.5078 | |
| $2^{-32}$ | 4.2347E-2 | 2.6753E-2 | 1.3538E-2 | 5.9432E-3 | 2.3629E-3 | 8.9021E-4 | 3.1233E-4 |
| | 0.6625 | 0.9827 | 1.1877 | 1.3307 | 1.4083 | 1.5111 | |

Table 2: Maximum pointwise errors and numerical order of convergence for example 2 with Shishkin mesh

| $\varepsilon \downarrow$ | N=16 | N=32 | N=64 | N=128 | N=256 | N=512 | N=1024 |
|---|---|---|---|---|---|---|---|
| $2^0$ | 3.1243E-3 | 7.7599E-4 | 1.9368E-4 | 4.8401E-5 | 1.2000E-5 | 3.0249E-6 | 7.562E-7 |
| | 2.0094 | 2.0023 | 2.0006 | 2.0000 | 2.0000 | 2.0000 | |
| $2^{-4}$ | 2.9938E-1 | 1.2030E-1 | 2.6798E-2 | 6.5282E-3 | 1.6217E-3 | 4.048E-4 | 1.0118E-4 |
| | 1.3154 | 2.1664 | 2.0374 | 2.0091 | 2.0023 | 2.0003 | |
| $2^{-8}$ | 3.4851E-1 | 1.7793E-1 | 8.2282E-2 | 3.5436E-2 | 1.1722E-2 | 6.7302E-3 | 4.4969E-3 |
| | 0.9699 | 1.1127 | 1.2154 | 1.5960 | 0.8005 | 0.5817 | |
| $2^{-12}$ | 3.5433E-1 | 1.8311E-1 | 8.6111E-2 | 3.9396E-2 | 1.5134E-2 | 5.6777E-3 | 2.1892E-3 |
| | 0.9524 | 1.0884 | 1.1281 | 1.3803 | 1.4144 | 1.3749 | |
| $2^{-14}$ | 3.5463E-1 | 1.8338E-1 | 8.6312E-2 | 3.9607E-2 | 1.5327E-2 | 5.8550E-3 | 2.0149E-3 |
| | 0.9515 | 1.0872 | 1.1238 | 1.3696 | 1.3884 | 1.5390 | |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $2^{-28}$ | 3.5473E-1 | 1.8346E-1 | 8.6379E-2 | 3.9677E-2 | 1.5397E-2 | 5.9144E-3 | 1.9647E-3 |
| | 0.9512 | 1.0868 | 1.1224 | 1.3696 | 1.3884 | 1.5390 | |
| $2^{-32}$ | 3.5472E-1 | 1.8346E-1 | 8.6379E-2 | 3.9678E-2 | 1.5393E-2 | 5.9147E-3 | 1.9702E-3 |
| | 0.9512 | 1.0868 | 1.1249 | 1.3661 | 1.3799 | 1.5860 | |

# References

[1] L. Abrahamsson, *A priori estimates for solutions of singular perturbations with a turning point*, Stud. Appl. Math. **56** (1977) 51–69.

[2] A. Berger, H. Han and R. Kellog, *A priori estimates and analysis of a numerical method for a turning point problem*, Math. Comp. **42** (1984) 465–492.

[3] C. Clavero and F. Lisbona, *Uniformly convergent finite difference methods for singularly perturbed problems with turning points*, Numer. Algo. **4** (1993) 339–359.

[4] C. de Boor, *On the convergence of odd degree spline interpolation*, J. Approx. Theory **1** (1968) 452–463.

[5] E.P. Doolan, J.J.H. Miller, and W.H.A. Schilders, *Uniform Numerical Methods for Problems with Initial and Boundary Layers*, Boole Press, Dublin, 1980.

[6] P.A. Farrell, A.F. Hegarty, J.J.H. Miller, E. O'Riordan, and G.I. Shishkin, *Robust Computational Techniques for Boundary Layers*, Chapman & hall, CRC Press, Boca Raton, Florida, 2000.

[7] P.A. Farrell, *Sufficient conditiond for the uniform convergence of a difference scheme for a singularly perturbed turning point problem*, SIAM J. Numer. Anal. **25** (1988) 618–643.

[8] C.A. Hall, *On error bounds for spline interpolation*, J. Approx. Theory **1** (1968) 209–218.

[9] J.C.R. Hunt and J.A. Shercliff, *Magnetohydrodynamics at high Hartman number*, Ann. Rev. Fluid Mech. **3** (1971) 37-62.

[10] M. Jacob, *Heat transfer.*, Wiley, New York, 1959.

[11] M.K. Kadalbajoo and K.C. Patidar, *Variable mesh spline approximation method for solving singularly perturbed turning point problems having boundary layer(s)*, Comp. Math. Appl. **42** (2001) 1439–1453.

[12] H.O. Kreiss, T.A. Manteuffel, B. Schwartz, B. Wendroff, and A.B. White, *Supra-convergent schemes on irregular grids*, Math. Comp. **47** (1986) 537-554.

[13] R.B. Kellog and A. Tsan, *Analysis of some difference approximations for a singular perturbation problem without turning points*, Math. Comp. **32** (1978) 1025–1039.

[14] J.J.H. Miller, E. O'Riordan and G.I. Shishkin, *Fitted Numerical Methods for Singular Perturbations Problems*, World Scientific, Singapore, 1996.

[15] G. Micula, *Handbook of Splines*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999.

[16] R. O'Malley, *Introduction to Singular Perturbations*, Academic Press, New York, 1974.

[17] S. Polak, C. Den Heiger, W.H. Schilders and P. Markowich, *Semiconductor device modelling from the numerical point of view* , Int. J. Numer. Methods Eng. **24** (1987) 763-838.

[18] P.M. Prenter, *Spline and Variational Methods*, John Willey & Sons, New York, 1975.

[19] H.G. Roos, M. Stynes and L. Tobiska, *Numerical Methods for Singularly Perturbed Differential Equations*, Springer, Berlin, 1996.

[20] G.-F. Sun and M. Stynes, *Finite element methods on piecewise equidistant meshes for interior turning point problems*, Numer. Algo. **8** (1994) 111–129.

[21] L.L. Schumaker, *Spline Functions: Basic Theory* , Krieger Publishing Company, Florida, 1981.

[22] K. Surla and Z. Uzelac, *A difference scheme for boundary value problems with turning point*, Univ. u Novom Sadu Zb. Rad. Prirod. -Mat. Fak. Ser. Mat. **25**(1995) 67-73.

[23] W. Wasow, *Linear turning point theory*, Springer, New York, 1985.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# Numerical Simulation of Turbulent Shear Flows Over a Third–Order Stokes Wave

**Harihar Khanal**[1] **and Shahrdad Sajjadi**[1]

[1] *Center for Geophysics and Planetary Physics, Department of Mathematics,*
*Embry-Riddle Aeronautical University, Daytona Beach, FL, U.S.A.*

emails: Harihar.Khanal@erau.edu, Shahrdad.Sajjadi@erau.edu

### Abstract

The problem of water wave generation and attenuation by wind is proposed by numerically calculating the turbulent air flow over a third order Stokes wave train. Most existing works Townsend [11], Gent & Taylor [4] uses a turbulent flow closer model of one equation type and assumed the flow to be aerodynamically rough to avoid the difficulties arising from the viscous sublayer. However, air flows over water waves are known to be aerodynamically transitional Snyder et al. [10]. Thus, in this investigation we shall adopt Sajjadi's [9] roughness parameter to circumvent this difficulty.

The turbulence model adopted for this investigation is based on the two equation closer scheme proposed by Saffman & Wilcox [7], and is used to simulate turbulent flow within and outside the viscous sublayer over steep nonlinear surface waves.

The linearized turbulent flow equations for small (yet finite) wave slope are solved numerically up, and including, the third order in wave steepness, taking into account the dynamical and kinematical boundary conditions at wave free surface. The resulting PDEs are first decomposed into a system of ODEs and solved numerically using the Multiple Shooting Method as described in Ascher et al. [1].

The main aim of the present investigation is to calculate the vertical structure of wind field, the perturbation pressure as well as the fractional rate of energy input from wind to nonlinear surface waves and hence calculate the energy transfer to a third-order Stokes wave as well as the growth rate due to turbulent shear flow flowing over it. The results show good agreement with computations of Conte & Miles [3] and also supports the recent theoretical investigation of Sajjadi [9].

*Key words: Air-Sea Interactions, Turbulence Model, Third-Order Stokes Waves.*

## 1 Introduction

Despite of the large amount of research conducted over the last sixty years, still the mechanism by which ocean waves are generated by wind is not fully understood. For example, when energy is transferred by wind to the ocean, it is still not known what percentage of energy and momentum go into waves and what percentage go into currents.

Earlier theories Miles [6], in which the interaction of the atmospheric turbulence and wave fields was neglected, have provided some explanation of the physics of the air flow over water waves. Recent works (Belcher & Hunt [2], Sajjadi [8, 9]) have provided a more detailed physical mechanism for this processes, in particular, they have shown that the surface pressure is extremely sensitive to the turbulence closure schemes adopted. However, the equations of motion which govern the turbulent air flow over water waves are not amenable to analytical solutions and therefore must be solved numerically.

In the present contribution, a numerical model is constructed for turbulent air flow over steep water waves. The water wave is represented by a third order Stokes wave and turbulence model adopted for the air flow over such waves is taken from the two equation model originally proposed by Saffman & Wilcox [7].

Since the boundary layer above water waves is not large compared to the wave amplitude, the boundary conditions cannot be prescribed on the mean undisturbed surface. Thus, orthogonal curvilinear coordinates are adopted and the equations of motion and the boundary conditions (BCs) are transformed into this coordinate system. The resulting equations and the BCs, in the new coordinates, are then expressed in perturbation expansions with respect to the wave slope, up to and including the third order in wave steepness.

The perturbation equations consists of a set of coupled ordinary differential equations, with respect to the vertical coordinate, which are solved numerically using the multiple shooting method as described by Ascher et al. [1].

## 2 Formulation of the Problem

We consider motion of an incompressible air flow over water waves, and refer the equations to Cartesian coordinates $(x, y, z)$ in which the $y$-axis is measured vertically upwards from the undisturbed water surface.

If $\tilde{u}_i = (\tilde{u}, \tilde{v}, \tilde{w})$ are the local components of flow velocity in Eulerian frame of reference at a point in Cartesian coordinates $(x, y, z)$, then the Navier-Stokes equations may be cast in the form

$$\frac{\partial \tilde{u}_i}{\partial x_j} = 0, \qquad \rho\left(\frac{\partial \tilde{u}_j}{\partial t} + \tilde{u}_j \frac{\partial \tilde{u}_i}{\partial x_j}\right) = -\frac{\partial \tilde{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left\{\mu\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right)\right\}, \tag{1}$$

where $\rho$, $\tilde{p}$ and $\mu$ are respectively the air density, the pressure and the dynamic viscosity. We will consider a fully developed turbulent flow of a third order Stokes wave

$$y_s = a\cos\kappa x + \tfrac{1}{2}a^2\kappa\cos 2\kappa x + \tfrac{3}{8}a^3\kappa^2\cos 3\kappa x + \mathcal{O}(a^4\kappa^3) \tag{2}$$

in a frame of reference moving with the wave with a speed $c$ in the positive $x$-direction, where $\kappa = 2\pi/\lambda$ is the wave number, $\lambda$ the wavelength and $a$ the amplitude.

Decomposing the instantaneous velocity fields and the pressure into mean and fluctuating components according to $\tilde{u}_i = U_i + u_i$ and $\tilde{p} = P + p$, then upon substitution into (1), followed by time averaging we obtain the Reynolds-averaged Navier-Stokes equations

$$\frac{\partial U_i}{\partial x_i} = 0, \qquad \rho U_j \frac{\partial U_i}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left\{\mu\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) - \rho\overline{u_i u_j}\right\} \tag{3}$$

In equation (3), $\rho\overline{u_i u_j}$ are the unknown Reynolds stresses which must be provided through a closure scheme. Here, we follow the closure model suggested by Saffman & Wilcox [7] and express the Reynolds stresses as

$$-\rho\overline{u_i u_j} = 2\rho\varepsilon S_{ij} - \tfrac{2}{3}\rho E\delta_{ij}, \quad S_{ij} = \tfrac{1}{2}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right), \; E = \tfrac{1}{2}\overline{u_i u_j} \tag{4}$$

where $\varepsilon$ is a scalar eddy viscosity, $S_{ij}$ is the mean rate of strain tensor, and $E$ is the specific turbulent kinetic energy.

Hence, the two-dimensional governing equations for the turbulent flow over Stokes wave may be cast in their final form as

$$\frac{\partial U_i}{\partial x_i} = 0, \; U_j\frac{\partial U_i}{\partial x_j} = -\frac{\partial \mathcal{P}}{\partial x_i} + \frac{\partial}{\partial x_j}\left\{2(\nu+\varepsilon)S_{ij}\right\}, \; \mathcal{P} = \tfrac{1}{2}P + \tfrac{2}{3}E \tag{5a}$$

where $\nu = \mu/\rho$ represents the kinematic viscosity. The turbulent closure (kinetic energy $E$ and pseudo-vorticity $\omega$) equations are

$$U_j\frac{\partial E}{\partial x_j} = E\left(b_1\sqrt{2S_{ij}S_{ij}} - b_2\omega\right) + \frac{\partial}{\partial x_j}\left\{(\nu + b_3\varepsilon)\frac{\partial E}{\partial x_j}\right\} \tag{5b}$$

$$U_j\frac{\partial \omega^2}{\partial x_j} = \omega^2\left(b_4\sqrt{\frac{\partial U_i}{\partial x_j}\frac{\partial U_i}{\partial x_j}} - b_5\omega\right) + \frac{\partial}{\partial x_j}\left\{(\nu + b_3\varepsilon)\frac{\partial \omega^2}{\partial x_j}\right\}, \tag{5c}$$

where $\varepsilon = E/\omega$.

## 3 Equations in Curvilinear Coordinates

In the problem posed here, the boundary layer thickness is small compared with the wave amplitude. Thus, it is not satisfactory to apply the surface boundary conditions at the mean water level ($y = 0$). Hence to circumvent this difficulty a reference frame, moving with the wave, is chosen and use is made of a system of orthogonal curvilinear coordinates in which the wave surface is a coordinate line. Therefore, we define

$$x = \zeta - \mathrm{Re}[iae^{(i\kappa(\zeta+i\eta))}], \quad y = \eta - \mathrm{Re}[ae^{(i\kappa(\zeta+i\eta))}]$$

To third order in $(a\kappa)$, the coordinate $\eta = 0$ corresponds to the free surface $y_s$ given by (2). The Jacobian of the transformation may be expressed as

$$J = \frac{\partial(\zeta,\eta)}{\partial(x,y)} = \left\{1 + 2a\kappa e^{-\kappa\eta}\cos\kappa\zeta + (a\kappa)^2 e^{-2\kappa\eta} + (a\kappa)^3 e^{-3\kappa\eta}\right\}^{-1} \tag{6}$$

correct to the third order.

Non-dimensionalizing the governing equations, given in section §2, such that the velocities are scaled with respect to the friction velocity $U_*$ and the length is scaled with

respect to $\nu/U_*$. Thus the equations of motion can be written in $(\zeta, \eta)$ coordinates system as follows:
Continuity:

$$J\left\{\frac{\partial}{\partial\zeta}\left(J^{-\frac{1}{2}}u\right)+\frac{\partial}{\partial\eta}\left(J^{-\frac{1}{2}}v\right)\right\} = 0 \tag{7}$$

where $u$ and $v$ are the mean mean velocities in the $\zeta$ and $\eta$ directions, respectively.
$u$-Momentum:

$$\begin{aligned}\frac{\partial}{\partial\zeta}\left(J^{-1}uu\right)+\frac{\partial}{\partial\eta}\left(J^{-1}uv\right)+J^{-2}\frac{(u^2+v^2)}{2}\frac{\partial J}{\partial\zeta}+J^{-1}\frac{\partial p}{\partial\zeta} = \\ \frac{\partial}{\partial\zeta}\left\{J^{-1}\hat{\varepsilon}\left[\frac{\partial}{\partial\zeta}\left(J^{\frac{1}{2}}u\right)+\frac{\partial}{\partial\eta}\left(J^{\frac{1}{2}}v\right)\right]\right\}-\frac{\partial}{\partial\eta}\left\{J^{-1}\hat{\varepsilon}\left[\frac{\partial}{\partial\eta}\left(J^{\frac{1}{2}}u\right)-\frac{\partial}{\partial\zeta}\left(J^{\frac{1}{2}}v\right)\right]\right\}\end{aligned} \tag{8}$$

Here $\nu = \mu/\rho$ is the kinematic viscosity, $\varepsilon = E/\omega$ is the turbulent viscosity and $\hat{\varepsilon} = \nu+\varepsilon$.
$v$-Momentum:

$$\begin{aligned}\frac{\partial}{\partial\zeta}\left(J^{-1}uv\right)+\frac{\partial}{\partial\eta}\left(J^{-1}vv\right)+J^{-2}\frac{(u^2+v^2)}{2}\frac{\partial J}{\partial\zeta}+J^{-1}\frac{\partial p}{\partial\eta} = \\ \frac{\partial}{\partial\zeta}\left\{J^{-1}\hat{\varepsilon}\left[\frac{\partial}{\partial\zeta}\left(J^{\frac{1}{2}}u\right)+\frac{\partial}{\partial\eta}\left(J^{\frac{1}{2}}v\right)\right]\right\}+\frac{\partial}{\partial\eta}\left\{J^{-1}\hat{\varepsilon}\left[\frac{\partial}{\partial\zeta}\left(J^{\frac{1}{2}}u\right)-\frac{\partial}{\partial\eta}\left(J^{\frac{1}{2}}v\right)\right]\right\}\end{aligned} \tag{9}$$

Turbulent Kinetic Energy:

$$\begin{aligned}J^{-\frac{1}{2}}u\frac{\partial E}{\partial\zeta}+J^{-\frac{1}{2}}v\frac{\partial E}{\partial\eta}-J^{-1}E\{b_1\sqrt{2S_{ij}S_{ij}}-b_2\omega\} \\ -\frac{\partial}{\partial\zeta}\left\{(\nu+b_3\hat{\varepsilon})\frac{\partial E}{\partial\zeta}\right\}-\frac{\partial}{\partial\eta}\left\{(\nu+b_3\hat{\varepsilon})\frac{\partial E}{\partial\eta}\right\} &= 0\end{aligned} \tag{10}$$

Pseudo-vorticity Equation:

$$\begin{aligned}J^{-\frac{1}{2}}u\frac{\partial\omega^2}{\partial\eta}+J^{-\frac{1}{2}}v\frac{\partial\omega^2}{\partial\zeta}-J^{-1}\omega^2\{b_4\sqrt{S_\omega}-b_5\omega\} \\ -\frac{\partial}{\partial\zeta}\left\{(1+b_6E\omega^{-1})\frac{\partial\omega^2}{\partial\eta}\right\}-\frac{\partial}{\partial\eta}\left\{(1+b_6E\omega^{-1})\frac{\partial\omega^2}{\partial\eta}\right\} = 0\end{aligned} \tag{11}$$

The stress terms in (10) and (12a) are given by

$$2S_{ij}S_{ij} = \mathcal{R}_t^2+\mathcal{R}_n^2 = \left\{\frac{\partial}{\partial\zeta}\left(J^{\frac{1}{2}}u\right)-\frac{\partial}{\partial\eta}\left(J^{\frac{1}{2}}v\right)\right\}^2+\left\{\frac{\partial}{\partial\eta}\left(J^{\frac{1}{2}}u\right)+\frac{\partial}{\partial\zeta}\left(J^{\frac{1}{2}}v\right)\right\}^2 \tag{12a}$$

where

$$\varepsilon\mathcal{R}_t = -\overline{uv} = \varepsilon\left\{\frac{\partial}{\partial\zeta}\left(J^{\frac{1}{2}}u\right)-\frac{\partial}{\partial\eta}\left(J^{\frac{1}{2}}v\right)\right\} \tag{12b}$$

$$\varepsilon\mathcal{R}_n = -\overline{u^2}+\frac{2}{3}E = \overline{v^2}-\frac{2}{3}E = \varepsilon\left\{\frac{\partial}{\partial\zeta}\left(J^{-\frac{1}{2}}u\right)-\frac{\partial}{\partial\eta}\left(J^{-\frac{1}{2}}v\right)\right\} \tag{12c}$$

are the tangential ($\varepsilon\mathcal{R}_t$) and the normal ($\varepsilon\mathcal{R}_n$) components of Reynolds stresses in the orthogonal curvilinear coordinates, and

$$S_\omega = S_{ij}S_{ij}+\frac{1}{2}J^2\left\{\frac{\partial}{\partial\zeta}\left(J^{-\frac{1}{2}}u\right)-\frac{\partial}{\partial\eta}\left(J^{-\frac{1}{2}}v\right)\right\}^2 \tag{12d}$$

The constants $b_1$ to $b_6$ in the transport equations (10) and (12a) are taken from Saffman & Wilcox [7] and their values are

$$b_3 = b_6 = \tfrac{1}{2},\ b_1 = 0.3,\ b_2 = b_1^2,\ b_4 = b_1\left\{\tfrac{b_5}{b_2} - \tfrac{4b6}{b_1}K^2\right\},\ K = 0.41,\ \text{and } \tfrac{5}{3} < \tfrac{b_5}{b_2} < 2.$$

Boundary Conditions: At $\eta = \eta_\infty$, which we typically take to be 10m above the water surface, the boundary conditions are

$$u = \tfrac{1}{K}\ln(1+\eta) + B - c,\ v = P = 0,\ E = \tfrac{1}{b_1},\ \omega = \tfrac{1}{Kb_1\eta} \tag{13}$$

whilst on the water surface ($\eta = 0$) the boundary conditions are

$$u = -cJ^{-\frac{1}{2}}(0),\ v = E = 0,\ \omega = Q\left(\tfrac{U_* z_0}{\nu}\right)J^{-1}(0)/b_1 \tag{14}$$

The constant $B$ in (14) depends on the nature of the surface. $z_0$ is the the roughness length and $U_*$ is the friction velocity. Here, $Q$ is the universal function defined by

$$Q\left(\tfrac{U_* z_0}{\nu}\right) = \begin{cases} \dfrac{6.26}{[\ln\left(\frac{U_* z_0}{\nu}\right) + 2.38]^2} & \text{(Smooth and transitional)} \\ \dfrac{1.44}{[\ln\left(\frac{U_* z_0}{\nu}\right) + 0.68]^2} & \text{(Rough)} \end{cases} \tag{15}$$

### 3.1 Linearized Perturbation Equations

We expand $u$, $v$, $P$, $E$, $\omega$ and $J$ in order of wave steepness $a\kappa$ as

$$\begin{aligned} u(\zeta,\eta) &= U_0(\eta) + a\kappa U_1(\eta)e^{iK\zeta} + (a\kappa)^2U_2e^{2iK\zeta} + (a\kappa)^3U_3e^{3iK\zeta} + \mathcal{O}(a\kappa)^4 \\ v(\zeta,\eta) &= a\kappa V_1(\eta)e^{iK\zeta} + (a\kappa)^2V_2e^{2iK\zeta} + (a\kappa)^3V_3e^{3iK\zeta} + \mathcal{O}(a\kappa)^4 \\ P(\zeta,\eta) &= a\kappa P_1(\eta)e^{iK\zeta} + (a\kappa)^2P_2e^{2iK\zeta} + (a\kappa)^3P_3e^{3iK\zeta} + \mathcal{O}(a\kappa)^4 \\ E(\zeta,\eta) &= E_0(\eta) + a\kappa E_1(\eta)e^{iK\zeta} + (a\kappa)^2E_2e^{2iK\zeta} + (a\kappa)^3E_3e^{3iK\zeta} + \mathcal{O}(a\kappa)^4 \\ \omega(\zeta,\eta) &= \Omega_0(\eta) + a\kappa\Omega_1(\eta)e^{iK\zeta} + (a\kappa)^2\Omega_2e^{2iK\zeta} + (a\kappa)^3\Omega_3e^{3iK\zeta} + \mathcal{O}(a\kappa)^4 \\ J(\zeta,\eta) &= 1 + a\kappa J_1(\eta)e^{iK\zeta} + (a\kappa)^2J_2e^{2iK\zeta} + (a\kappa)^3J_3e^{3iK\zeta} + \mathcal{O}(a\kappa)^4 \end{aligned}$$

Here $\lambda$ is wave length, $\kappa = \frac{2\pi}{\lambda}$ is the wave number, $K = \frac{\nu\kappa}{U_*}$, $J_1(\eta) = 2e^{-K\eta}$, $J_2(\eta) = 2e^{-2K\eta}$ and $J_3(\eta) = 2e^{-3K\eta}$. Substituting these expansions into the partial differential equations (7) – (12a), and the boundary conditions and equating the coefficients of $\mathcal{O}(1)$, $\mathcal{O}(a\kappa)$ and $\mathcal{O}(a\kappa)^2$ terms, we get the following system of ordinary differential equations.

To $\mathcal{O}(1)$ the equations of motion are

$u$-Momentum:

$$(1 + E_0\Omega_0^{-1})\frac{d^2U_0}{d\eta^2} + \left\{\frac{d}{d\eta}(1 + E_0\Omega_0^{-1})\right\}\frac{dU_0}{d\eta} = 0 \tag{16a}$$

Turbulent Kinetic Energy:

$$(1 + b_3E_0\Omega_0^{-1})\frac{d^2E_0}{d\eta^2} + \left\{\frac{d}{d\eta}(1 + b_3E_0\Omega_0^{-1})\right\}\frac{dE_0}{d\eta} + \left(b_3\Omega_0 - b_1\frac{dU_0}{d\eta}\right)E_0 = 0 \tag{16b}$$

Pseudo-vorticity:

$$(1+b_6E_0\Omega_0^{-1})\tfrac{d^2\Omega_0^2}{d\eta^2}+\left\{\tfrac{d}{d\eta}(1+b_6E_0\Omega_0^{-1})\right\}\tfrac{d\Omega_0^2}{d\eta}+\left(b_5\Omega_0-b_4\tfrac{dU_0}{d\eta}\right)\Omega_0^2\;=\;0 \tag{16c}$$

The boundary conditions are given by $U_0=\frac{1}{K}\ln(1+\eta_\infty)+B-c$, $E_0=\frac{1}{b_1}$, $\Omega_0=\frac{1}{Kb_1\eta_\infty}$ (typicallyfew meters above the wave) and $U_0=-c,\;E_0=0,\;\Omega_0=Q\left(\frac{U_*z_0}{\nu}\right)J_1^{-1}(0)/b_1.$ on the water surface ($\eta=0$).
Similarly the $\mathcal{O}(a\kappa)$ equations are given by
Continuity:

$$\tfrac{dV_1}{d\eta}+2iK\left(U_1-\tfrac{1}{2}U_0J_1\right)\;=\;0 \tag{17a}$$

$u$-Momentum:

$$\begin{aligned}&(1+E_0\Omega_0^{-1})\tfrac{d^2U_1}{d\eta^2}+\left\{\tfrac{d}{d\eta}(1+E_0\Omega_0^{-1})\right\}\tfrac{dU_1}{d\eta}-\left\{K^2(1+E_0\Omega_0^{-1})+iKU_0\right\}U_1\\&-iKP_1+\left\{iK\tfrac{d}{d\eta}(1+E_0\Omega_0^{-1})-\tfrac{dU_0}{d\eta}\right\}V_1+\tfrac{d}{d\eta}\left\{(E_0\Omega_0^{-1}-E_0\Omega_0^{-2}\Omega_1)\tfrac{dU_0}{d\eta}\right\}\\&-\tfrac{1}{2}K^2(1+E_0\Omega_0^{-1})U_0J_1-\tfrac{1}{2}J_1\tfrac{d}{d\eta}(1+E_0\Omega_0^{-1})\tfrac{dU_0}{d\eta}-\tfrac{1}{2}J_1(1+E_0\Omega_0^{-1})\tfrac{d^2U_0}{d\eta^2}\\&+\tfrac{1}{2}U_0(1+E_0\Omega_0^{-1})\tfrac{d^2J_1}{d\eta^2}\;=\;0\end{aligned} \tag{17b}$$

$v$-Momentum:

$$\begin{aligned}&(1+E_0\Omega_0^{-1})\tfrac{d^2V_1}{d\eta^2}+2\left\{\tfrac{d}{d\eta}(1+E_0\Omega_0^{-1})\right\}\tfrac{dV_1}{d\eta}-\left\{K^2(1+E_0\Omega_0^{-1})+iKU_0\right\}V_1\\&-\tfrac{dP_1}{d\eta}-\tfrac{1}{2}U_0^2\tfrac{dJ_1}{d\eta}+iK\left(\tfrac{dU_0}{d\eta}\right)\left(E_1\Omega_0^{-1}-E_0\Omega_0^{-2}\Omega_1\right)\\&-iKJ_1\left\{U_0\tfrac{d}{d\eta}(1+E_0\Omega_0^{-1})+(1+E_0\Omega_0^{-1})\tfrac{dU_0}{d\eta}\right\}\;=\;0\end{aligned} \tag{17c}$$

Turbulent Kinetic Energy:

$$\begin{aligned}&(1+b_3E_0\Omega_0^{-1})\tfrac{d^2E_1}{d\eta^2}+\left\{\tfrac{d}{d\eta}(1+b_3E_0\Omega_0^{-1})\right\}\tfrac{dE_1}{d\eta}\\&-\left\{K^2(1+E_0\Omega_0^{-1})+iKU_0-b_1\tfrac{dU_0}{d\eta}+b_2\Omega_0\right\}E_1\\&+b_3(E_1\Omega_0^{-1}-E_0\Omega_0^{-2}\Omega_1))\tfrac{d^2E_0}{d\eta^2}+\left\{b_3\tfrac{d}{d\eta}(E_1\Omega_0^{-1}-E_0\Omega_0^{-2}\Omega_1))-V_1\right\}\tfrac{dE_0}{d\eta}\\&+\left\{b_1\tfrac{dU_1}{d\eta}+\tfrac{1}{2}b_1U_0\tfrac{dJ_1}{d\eta}-\tfrac{1}{2}b_1J_1\tfrac{dU_0}{d\eta}+b_2\Omega_0J_1-b_2\Omega_1+iKV_1\right\}E_0\;=\;0\end{aligned} \tag{17d}$$

Pseudo-vorticity:

$$\begin{aligned}&\Omega_0(1+b_6E_0\Omega_0^{-1})\tfrac{d^2\Omega_1}{d\eta^2}+\left\{\Omega_0\tfrac{d}{d\eta}(1+b_6E_0\Omega_0^{-1})+\right.\\&\left.2(1+b_6E_0\Omega_0^{-1})\frac{d\Omega_0}{d\eta}\right\}\frac{d\Omega_1}{d\eta}-\left[K^2(1+b_6E_0\Omega_0^{-1})\Omega_0+b_4\Omega_0\frac{dU_0}{d\eta}-\frac{3}{2}b_5\Omega_0^2\right.\\&\left.-iKU_0\Omega_0+\tfrac{d}{d\eta}\left\{(1+b_6E_0\Omega_0^{-1})\tfrac{d\Omega_0}{d\eta}\right\}\right]\Omega_1+b_6E_1\Omega_0^{-1}\tfrac{d^2\Omega_0^2}{d\eta^2}+\\&b_4\Omega_0^2\tfrac{dU_0}{d\eta}\tfrac{dU_1}{d\eta}-V_1\tfrac{d\Omega_0^2}{d\eta}+b_5\Omega_0^3J_1-\tfrac{1}{2}b_4\Omega_0^2J_1\tfrac{dU_0}{d\eta}\;=\;0\end{aligned} \tag{17e}$$

The respective boundary conditions at $\eta = \eta_\infty$ are $U_1 = V_1 = P_1 = E_1 = \Omega_1 = 0$ and and on $\eta = 0$, $U_1 = \frac{1}{2}cJ_1(0)$, $V_1 = E_1 = 0$ and $\Omega_1 = Q\left(\frac{U_* z_0}{\nu}\right) J_1(0)/b_1$.
The $\mathcal{O}(a\kappa)^2$ and $\mathcal{O}(a\kappa)^3$ equations may be derived in the same way as the lower order ones. However, these equations are very lengthy and thus are not presented here.

### 3.2 Numerical Methods

We transform the independent variable $\eta$ to $\xi$ using the substitution $\xi = \ln(1+\eta)$. We then seperate the real and imaginary parts and put the system in a normal form. The resulting boundary value problem with a system of 54 first order nonlinear ordinary differential equations are solved numerically using the Multiple Shooting Method as described by Ascher et al. [1].

## 4 Results

The wave-perturbation pressure $p$ is proportional to the wave steepness $a\kappa$ and may be expressed as

$$p = \rho_w c^2 \Big[ \tfrac{a\kappa}{\sigma_1} (\alpha_1 + i\beta_1)\, e^{iK\zeta} + \tfrac{(a\kappa)^2}{\sigma_2} (\alpha_2 + i\beta_2)\, e^{2iK\zeta} + \tfrac{(a\kappa)^3}{\sigma_3} (\alpha_3 + i\beta_3)\, e^{3iK\zeta} \Big] \quad (18)$$

where $c = \sqrt{\frac{g}{\kappa}(1 + (a\kappa)^2}$ is the wave phase velocity, $\sigma_n$ $(n = 1, 2, 3)$ are the frequencies of the first, second and third harmonics and $\rho_w$ is the water density. The total energy transfer parameter from the wind to the wave is given by

$$\beta = \beta_0 + a\kappa\beta_1 + (a\kappa)^2\beta_2 + (a\kappa)^3\beta_3 \quad (19)$$

which is related to fractional growth of wave per radian $\zeta_a$. The energy transfer parameter, $\beta$, is related to the total energy of the water wave, $E$, by

$$(\kappa c E)^{-1} \tfrac{\partial E}{\partial t} = s\beta \left(\tfrac{U_1}{c}\right)^2 \quad (20)$$

where $s = \frac{\rho_a}{\rho_w}$, $\rho_a$ being the air density and $U_*$ the wind friction velocity, and $U_1 = \frac{U_*}{K}$.
The present turbulent closure model adopted here automatically produces a logarithmic mean velocity for the wind, which agrees exactly with the following analytical profile

$$U = U_1 \ln\left(\frac{\eta}{\eta_0}\right), \quad \eta \gg \eta_0 \quad (21)$$

Thus, we adopt the following formulation for the energy transfer rate

$$\beta = -\pi \frac{w_c''}{w_c'^3} \left( \int_{\eta_c}^{\infty} e^{-w} w^2 \, d\eta \right)^2 \quad (22)$$

where $w = U(\eta)/U_1$ and the suffix $c$ indicates evaluation at the critical height at which $U = c$. Hence, after calculating $u(\zeta_f, \eta) = U(\eta)$ at a fixed value of $\zeta$, namely $\zeta = \zeta_f$, we substitute the result in (22) and evaluate the integral numerically using Gauss-Laguerre quadrature. In (22), $\eta_c$ is given by $\eta_c = \Omega\left(\frac{U_1}{c}\right)^2 e^{c/U_1}$, where $\Omega = g\eta_0/U_1^2$

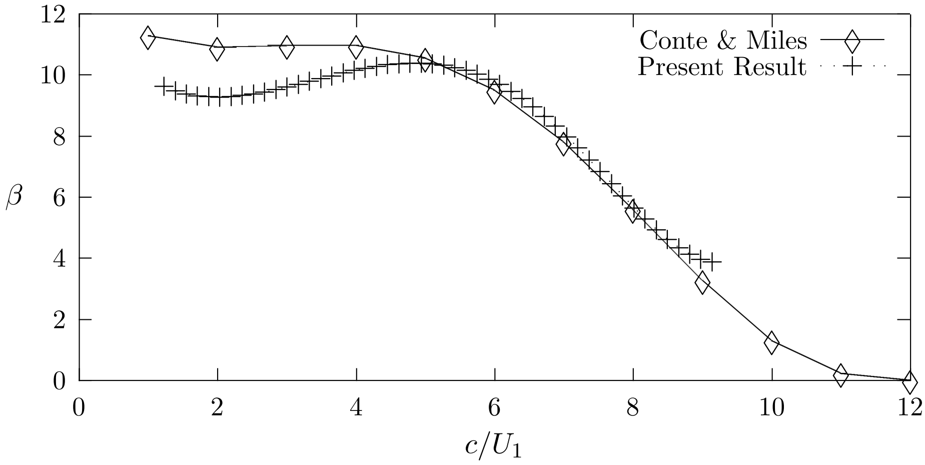


Figure 1: Variation of $\beta$ with $c/U_1$ for $\Omega = 3 \times 10^{-3}$. –◇–, Computation of Conte & Miles [3]; +++, Present computation.

is the dimensionless Charnock's constant, and $g$ is the acceleration due to gravity. In comparing our results with the numerical simulations of Conte & Miles [3], we have considered the transitional flow, for which $U_* \eta_0/\nu = 0.25$ for three values of $\Omega = 3\times10^{-3}$, $10^{-2}$ and $2\times10^{-2}$. Figure 1 shows the plot of $\beta$ against $c/U_1$ for $\Omega = 3\times10^{-3}$. As can be seen from the figure the result of the present computation agrees well with Conte & Miles in the narrow range $6 \leq c/U_1 \leq 9$ where Miles [6] formulation is expected to become important Phillips [5, §4.3]. For $c/U_1 < 6$ however, the present calculated result are smaller compared with Conte & Miles' values. Similarly, for $c/U_1 > 9$ the result is larger than theirs. The reason for this may be attributed to the fact that Miles' critical layer is moving farter (for smaller values) and closer (for the larger values) to the wave surface as $c/U_1$ decreases (for smaller values) and increases (for larger values) from the narrow range. Also, Conte & Miles' model is an inviscid laminar model and hence does not account for the effect of turbulence, particularly as $U_1$ increases and thus $c/U_1$ decreases. In figure 2 we consider the transitional air flow for which $\Omega = 10^{-2}$. As can be seen from this figure the agreement between the present computations and Conte & Miles' is in excellent agreement for $c/U_1 \leq 3$ and also in a narrow region $6 \leq c/U_1 \leq 7$. However, for a fully rough flow, $\Omega = 2 \times 10^{-2}$, where the effect of turbulence becomes more important, we observe the result of the present computation is higher, over the entire range of $c/U_1$, compared with that of Conte & Miles, as shown in figure 3. This is, of course, to be expected as Miles' model neglects the interaction of turbulent air flow with waves. Finally, in figure 4 the fractional growth of the wave per unit radian, $\zeta_a/s$, is plotted against $U_1/c$ for all values of Charnock's constant considered. Also, for comparison, the results of computation by Conte & Miles is also plotted. As can be

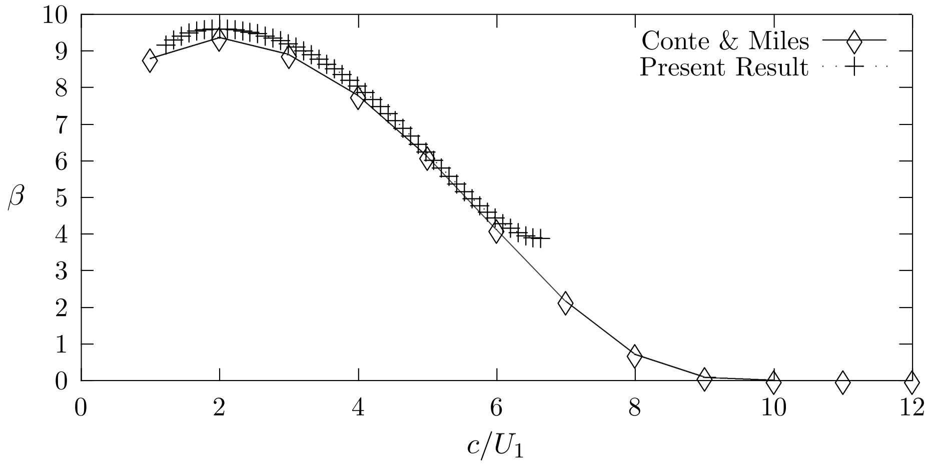


Figure 2: Variation of $\beta$ with $c/U_1$ for $\Omega = 10^{-2}$. –◇–, Computation of Conte & Miles [3]; +++, Present computation.

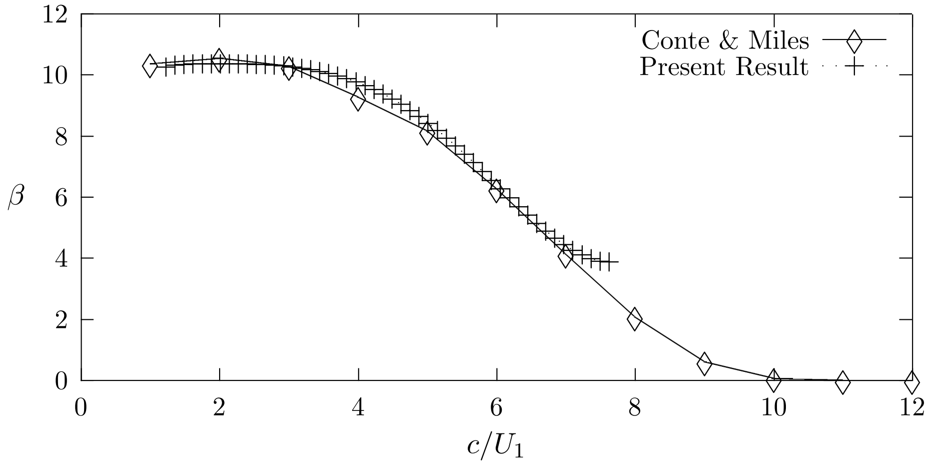


Figure 3: Variation of $\beta$ with $c/U_1$ for $\Omega = 2 \times 10^{-2}$. –◇–, Computation of Conte & Miles [3]; +++, Present computation.

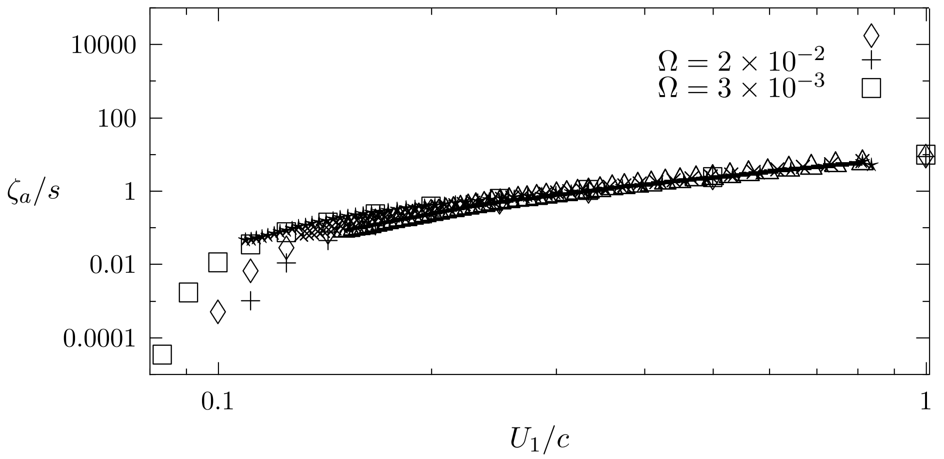


Figure 4: Variation of $\zeta_a/s$ with $U_1/c$ for $\Omega = 3 \times 10^{-3}, \Omega = 10^{-2}$ and $\Omega = 2 \times 10^{-2}$. Symbols, Computation of Conte & Miles [3]; solid lines, Present computation.

seen from this figure these is a very good agreement between the present computations and that of Conte & Miles [3]. We remark that the results indicate that there is very little sensitivity to the choice of Charnock's constant, $\Omega$, used, particularly in the region where turbulence is dominant. This is because in the presence of turbulence, the critical layer moves much closer to the surface and hence there will be insignificant contribution to the wave growth.

## 5 Conclusions

A numerical model for generation and growth of a third-order Stokes wave by a turbulent shear flow is proposed. The turbulence model adopted here is based on two equation model proposed by Saffman & Wilcox [7]. The governing equations for turbulent flow over a third-order Stokes wave are solved numerically up and including the third order in wave steepness.

From the results of computation, the energy transfer parameter from wind to wave as well as the wave growth per unit radian are calculated for a range of nondimensional wind parameters. The present results are compared with the earlier computation of Conte & Miles [3] and favorable agreement is achieved.

**ACKNOWLEDGEMENTS**

This work was supported by the Office of Sponsored Program and Department of Mathematics, Embry-Riddle Aeronautical University.

# References

[1] U. Ascher, R. Matheij and R. Russel, *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*, SIAM, Philadelphia, PA, 1995.

[2] S.E. Belcher and J.C.R. Hunt, *Turbulent shear flow over slowly moving waves*, J. Fluid Mech. **251** (1993) 109–148.

[3] S. Conte and J.W. Miles, *On the numerical investigation of the Orr-Sommerfeld equation*, J. Soc. Indust. Appl. Math. **7** (1959) 361–366.

[4] P.R. Gent and P.A. Taylor, *A Numerical Model of Flow above Water Waves*, J. Fluid Mech. **77** (1976) 105–128.

[5] O.M. Phillips, *The Dynamics of the Upper Ocean*, Cambridge University Press, 1977.

[6] J.W. Miles, *On the generation of waves by shear flows*, J. Fluid Mech. **3** (1957) 185–204.

[7] P.G. Saffman and D.C. Wilcox, *Turbulence model predictions for turbulent boundary layers*, AIAA J. **12** (1974) 541–546.

[8] S.G. Sajjadi, *On the growth of a fully non-linear Stokes wave by turbulent shear flow. Part 2. Rapid distortion theory*, Math. Engng. Ind. **6** (1998) 247–260.

[9] S.G. Sajjadi, *Interaction of Turbulence due to Tropical Cyclones with Surface Waves*, Adv. Appl. Fluid Mech. (2007) In press.

[10] R.L. Snyder, F.W. Dobson, J.A. Elliot and R.B. Long, *Array measurement of atmospheric pressure fluctuations above surface gravity waves*, J. Fluid Mech. **102** (1974) 1–59.

[11] A.A. Townsend, *Flow in a deep turbulent boundary layer over a surface distroted by water waves, J. Fluid Mech.* **55** (1972) 719-735.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# Sampling with prolates

**Tatiana Levitina[1] and Erkki J. Brändas[2]**

[1] *Institut Computational Mathematics, TU Braunschweig, Braunschweig, Germany*

[2] *Department of Quantum Chemistry, Uppsala University, Uppsala, Sweden*

emails: `t.levitina@tu-bs.de`, `Erkki.Brandas@kvac.uu.se`

**Abstract**

The Filter diagonalization technique using exact eigenfunctions of the finite Fourier transform is discussed and improved. A previously developed computational method based on the Walter-Shen sampling formula is advanced and extended.

*Key words: Fourier transform, filter diagonalization, spectral analysis, prolates, sampling theorem.*
*MSC 2000: 94A20, 42A99, 44A20, 33C90, 33F05*

## 1 Introduction

The present work continues a series of publications [1, 2, 3] on a modification of the Filter–diagonalization technique. Invented originally by Neuhauser and coworkers [4, 5, 6], this technique was later significantly developed (see, e.g. [7, 8, 9, 10]) and nowadays is one of the most popular and efficient tools for spectral analysis of complex quantum systems.

## 2 Filter Diagonalization Technique

In spite of the enormous variety of modifications, the basic idea of the Filter Diagonalization Technique remains unchanged. At first the auto–correlation function of the system is formed as $C(t) =< \Psi(0)|\hat{H}|\Psi(t) >$, with $\Psi(t)$ being a wave packet, that evolves in accordance with the time–dependent Schrödinger equation:

$$\frac{\partial\varphi}{\partial t}(\vec{x},t) = -i\hat{H}\varphi(\vec{x},t), \quad \text{i.e.} \quad \varphi(\vec{x},t) = e^{-i\hat{H}t}\varphi(\vec{x},0);$$

above $\hat{H}$ stands for the system Hamiltonian.

The auto–correlation function is assumed to be a sum of sinusoids of unknown frequencies — eigenvalues of the Hamiltonian. In order to detect and compute such frequencies located at a selected interval $(\omega^* - \Omega, \omega^* + \Omega)$, one filters $C(t)$, suppressing the contribution of harmonics that are situated outside this interval. Spectral estimation of the resulting filtered part of the auto–correlation function is then formulated as a linear algebra eigenvalue problem

$$\mathbf{U}\vec{b}_k = \omega_k \mathbf{W}\vec{b}_k, \tag{1}$$

for a pair of small matrixes, which entries are expressed through the short–time $(-T, T)$ segments of the auto–correlation function convolved with a filter. For the detailed and comprehensive survey of the Filter Diagonalization technique we address the reader to reference [11].

## 3 Filtering with Prolates

The present modification differs from the others by the choice of filtering functions, namely the eigenfunctions of the Finite Fourier Transform, i.e. defined by the equation

$$\int\limits_{-\sqrt{c}}^{\sqrt{c}} \exp(\mathrm{i}x\, y)\, \psi_l(c,y)\, dy = \mu_l(c)\, \psi_l(c,x)\,, \quad x \in [-\sqrt{c}, \sqrt{c}].$$

These functions are in several aspects superior in comparison with all other filters (see [1, 2, 3] and the references therein). In what follows we shall also refer to them as *prolates* as is the practice among the signal processing community.

We first scale and squeeze prolates so that they may fit the time- and frequency-intervals of interest:

$$\begin{aligned} \theta_l(\omega) &= \psi_l\left(\sqrt{\frac{T}{\Omega}}\,\omega\right)\Xi_\Omega, \quad \Xi_\Omega = \left\{ \begin{array}{l} 1,\ \omega \in (-\Omega, \Omega), \\ 0,\ \omega \in (-\infty, -\Omega) \cap (\Omega, \infty), \end{array} \right. \\ \Theta_l(t) &= \psi_l\left(\sqrt{\frac{\Omega}{T}}\,t\right), \end{aligned}$$

preserving $\Omega\, T = c$.

Convolved with the Fourier transform of the auto–correlation function, prolates $\theta_l(\omega)$ eliminate completely the contribution of the Hamiltonian $\hat{H}$ spectrum from outside the interval $(\omega^* - \Omega, \omega^* + \Omega)$, while for the spectrum located inside this interval the eigenvalue problem (1) arises, where matrix entrances have the appearance

$$\begin{aligned} \mathbf{W}_{sl} &= (-1)^{s+l}\frac{\Omega\, \mu_s \mu_l}{4\pi^2 T} \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} e^{i\omega^*(t-\tau)}\, C(t-\tau)\, \Theta_s(t)\; \Theta_l(\tau)\, d\tau\, dt \qquad (2) \\ \mathbf{U}_{sl} &= (-1)^{s+l}\frac{\Omega\, \mu_s \mu_l}{4\pi^2 T} \int\limits_{-\infty}^{\infty}\int\limits_{-\infty}^{\infty} e^{i\omega^*(t-\tau)} C(t-\tau)\, \Theta_s(t) \left\{ \omega^* \Theta_l(\tau) - i\, \frac{d\Theta_l\,(\tau)}{d\tau} \right\} d\tau\, dt. \end{aligned}$$

We transform the double integrals in infinite limits to the following expressions:

$$
\begin{aligned}
\mathbf{W}_{sl} &= \frac{(-1)^{l+s}}{4\,\pi^2}\mu_l\,\mu_s\,\frac{\Omega}{T}\int_{-\infty}^{\infty} e^{i\omega^* t}\,c(t)\,\Upsilon_{sl}(t)\,dt,\\
\mathbf{U}_{sl} &= \frac{(-1)^{l+s}}{4\pi^2}\mu_l\,\mu_s\frac{\Omega}{T}\int_{-\infty}^{\infty} e^{i\omega^* t}\,c(t)\,\zeta_{sl}(t)\,dt,
\end{aligned}
$$

where functions $\Upsilon_{sl}(t)$ and $\zeta_{sl}(t)$ are the convolutions of prolates:

$$
\begin{aligned}
\Upsilon_{sl}(t) &= \int_{-\infty}^{\infty} \Theta_s\,(t+\tau)\,\Theta_l(\tau)\,d\tau,\\
\zeta_{sl}(t) &= \int_{-\infty}^{\infty} \Theta_s\,(t+\tau)\left\{\omega^*\Theta_l(\tau) - i\,\frac{d\Theta_l\,(\tau)}{d\tau}\right\}d\tau,
\end{aligned}
$$

which allows us to convert the infinite integrals to the finite ones:

$$
\begin{aligned}
\Upsilon_{sl}(t) &= (-1)^l\,\frac{2\,\pi\,T}{\mu_s\,\mu_l\,\Omega}\int_{-\Omega}^{\Omega} e^{i\,\omega\,t}\theta_s\,(\omega)\,\theta_l(\omega)\,d\omega,\\
\zeta_{sl}(t) &= (-1)^l\,\frac{2\,\pi\,T}{\mu_s\,\mu_l\,\Omega}\int_{-\Omega}^{\Omega} e^{i\,\omega\,t}\theta_s\,(\omega)\,\theta_l(\omega)[\omega^* - \omega]\,d\omega,
\end{aligned}
$$

## 4 Sampling for Integration of Matrix Entries

To avoid the necessity to recompute $\Upsilon_{sl}(t)$ and $\zeta_{sl}(t)$ at a fine integration grid, we offer to interpolate these functions using special sampling formulas obtained recently by Walter and Shen [12], since both $\Upsilon_{sl}(t)$ and $\zeta_{sl}(t)$ are $\Omega$-band limited. In [3] we showed that the sampling formula in Ref. [12] works for these functions perfectly inside the interval $(-T,T)$, despite that they are $(-2T,2T)$–time concentrated; besides only few samples of $\Upsilon_{sl}(t)$ and $\zeta_{sl}(t)$ are required, see estimates in [3]. However our calculations show that at the points of $[-2T,-T]\bigcup[T,2T]$ both functions $\Upsilon_{sl}(t)$ and $\zeta_{sl}(t)$ differ essentially from zero, which means that they can not be efficiently expressed here as a linear combination of squeezed $\psi_l(c,y)$, as the latter are mostly concentrated on the interval $[-T,T]$. Yet one should not give up of sampling with prolates. Indeed, according to our estimates in [3], both $\Upsilon_{sl}(t)$ and $\zeta_{sl}(t)$ are highly concentrated in $[-2T,2T]$, which together with that they are $\Omega$–band limited, makes these functions a perfect object for sampling via $\psi_l(2c,y)$.

Preliminary calculations show the perfect correspondence between the sampling formula and the functions $\Upsilon_{sl}(t)$ and $\zeta_{sl}(t)$ at the whole range $[-2T,2T]$. Since outside this interval both $\Upsilon_{sl}(t)$ and $\zeta_{sl}(t)$ are practically zero, the outer integrals in the right hand side of (2) should be truncated to $[-2T,2T]$. The truncation error $\varepsilon_{tr}$ does not exceed then

$$
\varepsilon_{tr} \le \frac{2\,\|C(t)\|_\infty}{|\mu_p|}\left[\pi\sqrt{c}\left(1-\frac{|\mu_p|^2}{2\pi}\right)\right]^{1/2}, \quad p=\min\{l,s\},
$$

that is an invisibly small value, provided $l,s \ll 2c/\pi$.

Note that the number of required samples is 2 times larger than that for sampling on the interval $[-T, T]$. The numerical procedure used to calculate the prolates as well as all integrals of them has been described in detail in [13]. It allows calculation of prolates in a wide range of parameter "$c$" variation and guaranties the prescribed accuracy of all supplementary calculations.

## Acknowledgements

This work is supported by the Royal Swedish Academy of Sciences, HPC2N/NSC/PDC, the Foundation of Strategic Research, project number 99232, and the Russian Foundation for Basic Research under Grant No. 05-01-00257.

## References

[1] T.V. Levitina and E.J. Brändas, *Int. J. Theor. Phys.*, **Vol. 42**(10), 2531 (2003)

[2] T.V. Levitina and E.J. Brändas, *Journal of Mathematical Chemistry,* **40**(1), 43 (2006).

[3] T.V. Levitina and E.J. Brändas, *Int. J. of Comp. Math.,* (accepted for publishing).

[4] D. Neuhauser, *J. Chem. Phys.*, **Vol. 93**, 2611 (1990).

[5] M.R. Wall and D. Neuhauser, *J. Chem. Phys.*, **Vol. 102**, 8011 (1995).

[6] J.W. Pang and D. Neuhauser, *Chem. Phys. Lett.*, **Vol. 252**, 173 (1996).

[7] V.A. Mandelshtam and H.S. Taylor, *J. Chem. Phys.*, **Vol. 107**, 6756 (1997).

[8] J. Chen, V. A. Mandelstham and A. J. Shaka, *J. Magn. Reson.*, **Vol. 146**, 368 (2000).

[9] V. A. Mandelshtam, *Prog. in Nucl. Magn. Reson. Spec*, **Vol. 38**, 159 (2001).

[10] R. Santra, J. Breidbach, J. Zobeley, and L.S. Cederbaum, *J. Chem. Phys.*, **Vol. 112** (21), 9243 (2000).

[11] V.A. Mandelshtam, *Progress in Nuclear Magnetic Resonance Spectroscopy*, **Vol. 38**, 159 (2001).

[12] G. G. Walter and X. Shen, *Journal of Sampling Theory in Signal and Image Processing* , **Vol. 2**, N. 1, 25 (2003).

[13] T.V. Levitina, E.J. Brändas, *J.Comp.Meth.Sci. & Engrg.* **Vol. 1**, N. 1, 287 (2001).

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE2007*
*Chicago, 20-23 June 2007*

# Quantitative Diffusion Tensor Imaging Tractography Measures along Geodesic Distances in Amnestic Mild Cognitive Impairment

**Xuwei Liang, Ning Cao and Jun Zhang**

*Laboratory of Computational Medical Imaging & Data Analysis,*
*Department of Computer Science, University of Kentucky*

emails: xuwei.liang@uky.edu, ningcao@csr.uky.edu, jzhang@cs.uky.edu

## Abstract

Diffusion tensor imaging (DTI) based tractography enables selective reconstruction of specific white matter pathways in human brain. The various DTI techniques have been used to investigate white matter abnormalities and alternations in amnestic mild cognitive impairment (MCI) and Alzheimer’s disease (AD). Region-of-interest (ROI) and voxel based morphometric (VBM) analyses have been popular approaches so far although there are potential limitations with these methods. In this study, we estimated the white matter abnormalities of the cingulum in participants with MCI compared with age-matched controls using DTI tractography and statistical analysis of diffusivity measures, namely mean diffusivity (MD) and fractional anisotropy (FA) indices, mapped as geodesic pathways to establish the correspondence across individual subjects. The preliminary result illustrates localized micro structural whiter matter changes within the left posterior cingulum which is in agreement with previously proposed clinical studies. This demonstrates taht the proposed method is feasible and may be useful of early AD assessment.

Key words: Mild cognitice impairment, diffusion tensor imaging, tractography, geodesic distances

## References

[1] Kang N, Zhang J, Carlson ES, Gembris D. White matter fiber tractography via anisotropic diffusion simulation in the human brain. *IEEE Trans. Med. Imag.* 2005;24:1127–1137.

[2] Zhang J, Kang N, Rose SE. Approximating anatomical brain connectivity with diffusion tensor MRI using kernel-based diffusion simulations. In G. E. Christensen and M. Sonka, editors, *Proc. of IPMI 2005, LNCS 3565*, pages 64–75, 2005.

[3] Xu D, Mori S, Solaiyappan M, van Zijl PCM, and Davatzikos C, A framework for callosal fiber distribution analysis, *NeuroImage* 17**,** 1131–1143 (2002).

[4] Stephen E Rose, Katie L McMahon, Andrew L Janke, Brona O'Dowd, Greig de Zubicaray, Mark W Strudwick and Jonathan B Chalk. MRI diffusion indices and neuropsychological performance in amnestic mild cogntive impairment. *J. Neurol. Neurosurg. Psychiatry* published online 5 Jun. 2006.

*Proceedings of the International Conference
on Computational and Mathematical Methods
in Science and Engineering, CMMSE 2007
Chicago, 20–23 June 2007.*

# High Order Compact Finite Difference Scheme for Solving Nonlinear Black-Scholes equation with Transaction Costs

**Wenyuan Liao[1] and Abdul Q. M. Khaliq[2]**

[1] *Department of Mathematics and Statistics, McMaster University,
Hamilton, Ontario, L8S 4K1, Canada*

[2] *Department of Mathematical Sciences, Middle Tennessee State University,
Murfreesboro, Tennessee 37132 , USA*

emails: wyliao@math.mcmaster.ca, akhaliq@mtsu.edu

**Abstract**

Recently several nonlinear Black-Scholes equations were widely used to model option price when transaction cost is considered. Due to the complicity the analytical solution to such model is seldom available, so numerical method is fairly important and necessary. In this paper, an unconditionally stable high order compact finite difference scheme is proposed. The compact algorithm is fourth-order accurate in both the temporal and spatial dimensions. Except for price of option, the new algorithm also computes the hedging delta $\frac{\partial V}{\partial S}$ as well. Two numerical examples are presented to demonstrate the accuracy and efficiency of the proposed scheme.

*Key words: High Order Compact, Black Scholes, Option Pricing, Transaction Cost*

## 1 Introduction

In the past several decades, stock option was one of the most popular financial derivatives which was widely and successfully used to hedge risk in financial world. Many types of options are available to buyers, to name a few, such as European Call(Put) option, American Call(Put) option, Exotic option, Bermuda option, etc, and those options are currently traded throughout the world. However accurately pricing an option was not easy until in 1973, Black and Scholes published their famous Black-Scholes model in [3]. In an idealized financial market, the price of an European option can be obtained by analytically solving the Black-Scholes equation. but this is not very useful in practice, as mentioned in [6] and [17] because the Black-Scholes model had been derived under some very restrictive assumptions, such as frictionless, liquid and complete market. In the real financial market, the traders actually work in a different environment: transaction costs arising [1],[2],[4] and [5]; market is incomplete, etc. Leland firstly noticed and proposed a modified nonlinear Black-Scholes model to deal with transaction costs in [8]. In 1973 Boyle and Vorst modified the volatility in the original Black-Scholes model with $\sigma = \sigma_0(1 + c\frac{\mu}{\sigma_0\sqrt{\Delta t}})^{\frac{1}{2}}$ in [4]. Some other modifications to $\sigma$ also had been proposed recently, such as in [12], Parás and Avellaneda replaced $\sigma$ with $\sigma_0(1 + Asign(V_{SS}))^{\frac{1}{2}}$.

A more complicated model has been proposed by Barles and Soner in [2], which will be discussed in more details in the next section.

As one can see, analytical solution to nonlinear Black-Scholes equation is seldom available. So we have to rely on numerical approaches such as binomial approximations, Monter-Carlo methods, finite element method and finite difference method to get accurate option price. Our goal here is to obtain unconditionally stable high order compact finite difference scheme for solving nonlinear Black-Scholes equation.

The rest of the paper is organized as follows. In section 2 a nonlinear Black-Scholes equation is introduced and reformulated, and an explicit treatment for the nonlinear term is introduced so finally a linear convection-diffusion equation is obtained. Some previous high order compact methods for solving linear convection-diffusion equation are summarized in section 3, which is followed by the description of our compact high order finite difference scheme in section 4. Two numerical examples are presented to illustrate the high order and efficiency of the new algorithm in section 5. Some conclusion remarks and possible future works are discussed in the final section.

## 2 Mathematical model

In this section, we consider the model

$$V_\tau + \frac{1}{2}\sigma(V_{SS})^2 S^2 V_{SS} + \rho S V_S - \rho V = 0 \tag{1}$$

where the nonlinear volatility is defined as $\sigma = \sigma_0(1 + \Phi(e^{(\rho(\tau_0-\tau)})a^2S^2V_{SS})$, $\rho$ is the risk-free interest rate, $\tau_0$ is the maturity and $a$ is a constant relating to transaction costs. Note that in (1) the function $\Phi$ is defined as the solution of the following initial-value problem:

$$\Phi'(x) = \frac{\Phi(x)+1}{2\sqrt{x\Phi(x)} - x} \quad for \quad x \neq 0. \tag{2}$$

with initial condition $\Phi(0) = 0$.

The terminal condition for (1) is given by: $V(S,\tau_0) = V_0(S)$ for $S \geq 0$, where $\tau_0$ is the excise time, $E$ is strike price. The boundary conditions for (1) are given as

$$\begin{aligned} V(0,\tau) &= 0, \quad for \quad 0 \leq \tau \leq \tau_0 \\ V(S,\tau) &\sim S - Ee^{\rho(\tau-\tau_0)}, \quad when \quad S \to \infty \end{aligned} \tag{3}$$

To transform the problem (1) into a convection-diffusion problem and to avoid possible degeneration at $S = 0$, a variable transformation is used [6]. Let $x(S) = ln(\frac{S}{E})$,

$t(\tau) = \frac{1}{2}\sigma_0^2(\tau_0 - \tau)$, and $u = e^{-x}\frac{V}{E}$, then (1) can be reformulated into

$$u_t - (1 + \Phi[e^{(Kt+x)}a^2E(u_{xx} + u_x)])(u_{xx} + u_x) - Ku_x = 0 \tag{4}$$

where $x \in (-\infty, \infty)$, $0 \leq t \leq T = \frac{\sigma_0^2\tau_0}{2}$, $K = \frac{2\rho}{\sigma_0^2}$. The reformulated problem is associated with the following initial and boundary conditions:

$$\begin{aligned} u(x,0) &= u_0(x) = \max(1 - e^{-x}, 0), && (5)\\ u(x,t) &= 0 \qquad (x \to -\infty), && (6)\\ u(x,t) &\sim 1 \qquad (x \to +\infty). && (7) \end{aligned}$$

In the next two sections we will focus on the numerical methods for solving a convection-diffusion equation in the form of (4). Once the numerical solution to (4) is obtained, we can do an inverse transformation to compute the option price.

# 3 Review of previous methods

Solving (4) involves the solution of (2), so any implicit algorithm is very complicated and inefficient, since the function $\Phi$ is not explicitly defined. One widely used approach is to treat the term $\Phi$ explicitly then (4) becomes a convection-diffusion equation with constant coefficients, which can be solved efficiently by some implicit algorithm. Numerical tests shown that the explicit treatment of the nonlinear term $\sigma$ will not effect the algorithm's overall stability.

Thus, equivalently we can just focus on the numerical methods for solving the following one dimensional time dependent convection-diffusion equation:

$$u_t = \beta u_{xx} + \lambda u_x \tag{8}$$

where $\beta$ and $\lambda$ are constants and some corresponding boundary and initial conditions are given as well.

Several high order compact scheme had been developed to solve equation (8) in the past several decades, such as [7], [16] and [18]. These schemes approximate spatial differential with high order accuracy and result in compact thus efficient computation, but failed to handle time derivative in an efficient and accurate way. So far the highest order in temporal dimension is only third order. Further more, some stability issues arise because the time derivative has not been carefully handled.

A family of fourth-order finite difference schemes for solving (8) has also been proposed by Rigal in [13],[14] and [15] as well. These schemes are defined as follows:

$$\begin{aligned} (1+C)D_t u_j^n &= (\frac{1}{2} + A_1)\beta D_+D_-u_j^n + (\frac{1}{2} + A_2)\beta D_+D_-u_j^{n+1}\\ &\quad + \lambda(\frac{1}{2} + B_1)D_0u_j^n + \lambda(\frac{1}{2} + B_2)D_0u_j^{n+1} \qquad (9) \end{aligned}$$

where $D_t, D_0, D_+$ and $D_-$ are some basic difference operators while $A_i$,$B_i$ and $C$ are some parameters which will be chosen such that the highest order truncation error can be eliminated thus the resulted scheme is fourth order accurate(basically to eliminate highest order term in Taylor series) and compact. It is obvious that the resulted schemes are higher order and compact but one can also easily see, that the highest possible order in temporal dimension is still third order, and the stable region is considerably small as well.

## 4 Higher order compact scheme

To take advantage of Padé approximation and Richardson extrapolation, and achieve unconditional stability, a new scheme is proposed in this paper. In stead of solving a single convection-diffusion equation, we convert the original problem (8) to a system of two equations by introducing a new unknown function $v(x,t) = u_x(x,t)$, so $u_t = \beta u_{xx} + \lambda v$. Now apply $\frac{\partial}{\partial x}$ to both sides of the new equation, we obtain $u_{xt} = \beta u_{xxx} + \lambda u_{xx}$. Since $u_{xt} = v_t$ and $u_{xxx} = v_{xx}$, the original convection-diffusion equation can be converted to the following equivalent system:

$$u_t = \beta u_{xx} + \lambda v \tag{10}$$

$$v_t = \beta v_{xx} + \lambda u_{xx} \tag{11}$$

If there is a reaction term $f(u)$ in the original equation (8), we can modify (11) and still obtain a similar system

$$\begin{aligned} u_t &= \beta u_{xx} + \lambda v + f(u) \\ v_t &= \beta v_{xx} + \lambda u_{xx} + \frac{\partial f}{\partial u} v \end{aligned} \tag{12}$$

To be complete, (11) and (12) have to be equipped with corresponding boundary and initial conditions. For $u(x,t)$, the initial and boundary conditions are already been give as $u(x,0) = u_0(x)$, $u(0,t) = b_0(t)$ and $u(1,t) = b_1(t)$ respectively. Assume that the $u_0(t)$ is smooth enough, one can derive the initial condition for $v(x,t)$ by taking derivative of $u(x,t)$ with respect to $x$ then letting $t \to 0$, so $v(x,0) = {u_0}'(x)$.

Unlike the initial condition,, it is difficult to obtain an analytical expressions of the boundary conditions for $v(x,t)$. Here we propose a compact fourth order approximation to solve this issue.

Notice that usually we can not derive an analytical boundary conditions for $v(x,t)$, so a compact fourth order numerical approximation is proposed here to approximate $v(0,t)$ and $v(1,t)$. Define the difference operator $\Delta_x$ as $\Delta_x u_i = \frac{u_{i+1} - u_{i-1}}{2h}$. Assume the grid is uniform, i.e., the interval $[0,1]$ is divided into $N$ subintervals and $h = \frac{1.0}{N}$. As defined early, $v(h,t) = \frac{\partial u}{\partial x}(h,t) \approx \frac{u(2h,t)-u(0,t)}{2h} = \frac{\Delta_x}{2h} u(h,t)$ is a second order approximation, which can be improved to fourth oder if $\Delta_x$ is replaced by $\frac{\Delta_x}{1+\frac{1}{6}\delta_x^2}$. Therefore, $v(h,t) = \frac{\Delta_x}{2h(1+\frac{1}{6}\delta_x^2)} u(h,t)$, which implies that $(1 + \frac{1}{6}\delta_x^2) v(h,t) = \frac{\Delta_x}{2h} u(h,t)$, i.e. $\frac{1}{6} v(2h,t) + \frac{2}{3} v(h,t) + \frac{1}{6} v(0,t) = \frac{1}{2h}(u(2h,t) - u(0,t))$, from which we can obtain a fourth order approximation: $v(0,t) = \frac{3}{h}(u(2h,t) - u(0,t)) - 4v(h,t) - v(2h,t)$. Similarly, we can approximate boundary condition $v(1,t)$ as $v(1,t) = \frac{3}{h}(u(1-h,t) - u(1-2h,t)) - 4v(1-h,t) - v(1-2h,t)$.

Many Padé approximation based higher order efficient schemes which can be used to solve the reaction-diffusion system (11) and (12), such as [9],[10] and [11]. Without loss of generality, we combine (11) and (12) by considering a more general equation

$$\begin{aligned} u_t &= \beta u_{xx} + f(u,v) \\ v_t &= \beta v_{xx} + \lambda u_{xx} + g(u,v) \end{aligned} \tag{13}$$

Here the term $\lambda v$ is included in the general function $f(u,v)$ so in the first equation there is only one diffusion term $\beta u_{xx}$.

The new algorithm starts from the standard second order Crank-Nicolson scheme,

$$\begin{aligned} \frac{u_i^{n+1}-u_i^n}{\Delta t} &= \frac{1}{2}(\frac{\beta}{h^2}\delta_x^2 u_i^{n+1} + \frac{\beta}{h^2}\delta_x^2 u_i^n + f_i^{n+1} + f_i^n) \\ \frac{v_i^{n+1}-v_i^n}{\Delta t} &= \frac{1}{2}(+\frac{\lambda}{h^2}\delta_x^2 u_i^{n+1} + \frac{\lambda}{h^2}\delta_x^2 u_i^n + \frac{\beta}{h^2}\delta_x^2 v_i^{n+1} + \frac{\beta}{h^2}\delta_x^2 v_i^n + g_i^{n+1} + g_i^n) \end{aligned} \tag{14}$$

where $f_i^{n+1} = f(u_i^{n+1}, v_i^{n+1})$, $f_i^n = f(u_i^n, v_i^n)$, $g_i^{n+1} = g(u_i^{n+1}, v_i^{n+1})$, and $g_i^n = g(u_i^n, v_i^n)$, the standard central difference operator $\delta_x^2$ defined by $\delta_x^2 u_i = u_{i+1} - 2u_i + u_{i-1}$, however $(u_{xx})_i \approx \frac{1}{h^2}(u_{i+1} - 2u_i + u_{i-1})$ gives only second order approximation to $u_{xx}$. One way to improve the above approximation to fourth order is Padé approximation: $(u_{xx})_i \approx \frac{\delta_x^2}{h^2(1+\frac{1}{12}\delta_x^2)}$.

If we apply Padé approximation in (14), the following fourth order compact scheme is obtained

$$\begin{aligned} \frac{u_i^{n+1}-u_i^n}{\Delta t} &= \frac{1}{2}(\beta\frac{\delta_x^2}{h^2(1+\frac{1}{12}\delta_x^2)}u_i^{n+1} + \beta\frac{\delta_x^2}{h^2(1+\frac{1}{12}\delta_x^2)}u_i^n + f_i^{n+1} + f_i^n) \\ \frac{v_i^{n+1}-v_i^n}{\Delta t} &= \frac{1}{2}(+\lambda\frac{\delta_x^2}{h^2(1+\frac{1}{12}\delta_x^2)}u_i^{n+1} + \lambda\frac{\delta_x^2}{h^2(1+\frac{1}{12}\delta_x^2)}u_i^n \\ &\quad +\beta\frac{\delta_x^2}{h^2(1+\frac{1}{12}\delta_x^2)}v_i^{n+1} + \beta\frac{\delta_x^2}{h^2(1+\frac{1}{12}\delta_x^2)}v_i^n + g_i^{n+1} + g_i^n) \end{aligned} \tag{15}$$

which is second order in time and fourth order in space. Multiply $1 + \frac{1}{12}\delta_x^2$ to both sides, the new scheme can be rewritten as

$$(1+\frac{\delta_x^2}{12} - \frac{\beta r_x}{2}\delta_x^2)u_i^{n+1} = (1+\frac{\delta_x^2}{12} + \frac{\beta r_x}{2}\delta_x^2)u_i^n + \frac{\Delta t}{2}(1+\frac{\delta_x^2}{12})(f_i^{n+1} + f_i^n) \tag{16}$$

$$\begin{aligned} (1+\frac{1}{12}\delta_x^2 - \frac{\beta r_x}{2}\delta_x^2)v_i^{n+1} &= (1+\frac{1}{12}\delta_x^2 + \frac{\beta r_x}{2}\delta_x^2)v_i^n + \lambda\frac{r_x}{2}\delta_x^2(u_i^{n+1} + u_i^n) \\ &\quad +\frac{\Delta t}{2}(1+\frac{1}{12}\delta_x^2)(g_i^{n+1} + g_i^n) \end{aligned} \tag{17}$$

where $r_x = \frac{\Delta t}{h^2}$.

One can easily show that the truncation error of (16) and (17) is in the form of $C_1\Delta t^2 + C_2\Delta t^4 + C_3 h^4$. Therefore the Richardson extrapolation can be used here to improve the method to fourth order in time. Let $w^{\Delta t} = (u^{\Delta t}, v^{\Delta t})$ and $w^{\frac{\Delta t}{2}} = (u^{\frac{\Delta t}{2}}, v^{\frac{\Delta t}{2}})$ represent the solutions obtained by using time step-size $\Delta t$ and $\frac{\Delta t}{2}$, respectively, we can set the final solution as $w = \frac{4w^{\frac{\Delta t}{2}} - w^{\Delta t}}{3}$ to eliminate the term $\Delta t^2$, which is fourth order accurate in both the temporal and spatial dimensions.

Note that Eq.(16) and (17) contain both $u_i^{n+1}$ and $v_i^{n+1}$(explicitly or implicitly in $f_i^{n+1}$ and $g_i^{n+1}$). It is not efficient if both equations are simply linearized and using Newton's method, so an efficient alternating direction iterative algorithm is proposed here.

Denote the solutions to (16) and (17) after $k$ iterations by $u_i^{n+1^{(k)}}$ and $v_i^{n+1^{(k)}}$ respectively. To get $u_i^{n+1^{(k+1)}}$ and $v_i^{n+1^{(k+1)}}$, We first expand $f_i^{n+1} = f(u_i^{n+1}, v_i^{n+1})$ by

$$f(u_i^{n+1}, v_i^{n+1}) = f(u_i^{n+1^{(k)}}, v_i^{n+1^{(k)}}) + \frac{\partial f}{\partial u}(u_i^{n+1^{(k)}}, v_i^{n+1^{(k)}})(u_i^{n+1} - u_i^{n+1^{(k)}}) \qquad (18)$$

and insert it into (16), then solve the following (19) for $u_i^{n+1^{(k+1)}}$

$$\begin{aligned}(1 + \frac{1}{12}\delta_x^2 - \frac{\beta r_x}{2}\delta_x^2 - \frac{\Delta t}{2}(1 + \frac{1}{12}\delta_x^2)\hat{\mathbf{J}}_i^{\ n+1^{(k)}})u_i^{n+1} &= (1 + \frac{1}{12}\delta_x^2 + \frac{\beta r_x}{2}\delta_x^2)u_i^n \\ + \frac{\Delta t}{2}(1 + \frac{1}{12}\delta_x^2)(f(u_i^{n+1^{(k)}}, v_i^{n+1^{(k)}}) &- \hat{\mathbf{J}}_i^{n+1^{(k)}} u_i^{n+1^{(k)}} + f(u_i^n, v_i^n)) \qquad (19)\end{aligned}$$

where $\hat{\mathbf{J}}_i^{n+1^{(k)}} = \frac{\partial f}{\partial u}(u_i^{n+1^{(k)}}, v_i^{n+1^{(k)}})$.

Once $u_i^{n+1^{(k+1)}}$ is available, we proceed to expand $g_i^{n+1} = g(u_i^{n+1}, v_i^{n+1})$ by

$$g(u_i^{n+1}, v_i^{n+1}) = g(u_i^{n+1^{(k+1)}}, v_i^{n+1^{(k)}}) + \frac{\partial g}{\partial v}(u_i^{n+1^{(k+1)}}, v_i^{n+1^{(k)}})(v_i^{n+1} - v_i^{n+1^{(k)}}) \quad (20)$$

then insert it into (17). We thus obtain the following equation

$$\begin{aligned}(1 + \frac{1}{12}\delta_x^2 - \frac{\beta r_x}{2}\delta_x^2 - \frac{\Delta t}{2}(1 + \frac{1}{12}\delta_x^2)\tilde{\mathbf{J}}_i^{n+1^{(k)}})v_i^{n+1} &= (1 + \frac{1}{12}\delta_x^2 + \frac{\beta r_x}{2}\delta_x^2)u_i^n \\ + \frac{\Delta t}{2}(1 + \frac{1}{12}\delta_x^2)(f(u_i^{n+1^{(k)}}, v_i^{n+1^{(k)}}) &- \tilde{\mathbf{J}}_i^{n+1^{(k)}} u_i^{n+1^{(k)}} + f(u_i^n, v_i^n)). \qquad (21)\end{aligned}$$

where $\tilde{\mathbf{J}}_i^{n+1^{(k)}} = \frac{\partial g}{\partial v}(u_i^{n+1^{(k+1)}}, v_i^{n+1^{(k)}})$. Solve it we can get $v_i^{n+1^{(k+1)}}$. The two steps are repeated alternatively till convergent.

# 5 Numerical results

## 5.1 Case 1: Linear Convection-Diffusion equation

We first test our new scheme by solving an 1D convection-diffusion equation. Our goal for this numerical test case is to show that the new algorithm is fourth order accurate in both time and space. In this example, we consider the following equation

$$\begin{aligned}\frac{\partial u}{\partial t} &= \frac{1}{2}\frac{\partial^2 u}{\partial x^2} + \frac{1}{2}\frac{\partial u}{\partial x}, \quad x \in (0,1), \quad t \in (0,1] \\ u(x,0) &= e^x, \quad x \in [0,1] \\ u(0,t) &= e^t, \quad u(1,t) \ = \ e^{1+t}, \quad t \in (0,1] \qquad (22)\end{aligned}$$

Table 1:

| $\Delta t$ | 0.2 | 0.1 | 0.05 | 0.025 | 0.0125 |
|---|---|---|---|---|---|
| $e_1$ | 3.1458E-003 | 7.8855E-004 | 1.9723E-004 | 4.9313E-005 | 1.2329E-005 |
| $Log_r(\frac{e_1(\Delta t)}{e_1(\frac{\Delta t}{2})})$ | - | 1.9961 | 1.9993 | 1.9999 | 2.0000 |
| $e_2$ | 7.2118E-006 | 3.8724E-007 | 2.3334E-008 | 1.4282E-009 | 8.7867E-011 |
| $Log_r(\frac{e_2(\Delta t)}{e_1(\frac{\Delta t}{2})})$ | - | 4.2195 | 4.0527 | 4.0301 | 4.0228 |

Table 2:

| $h$ | 0.1 | 0.05 | 0.025 | 0.0125 | 0.00625 |
|---|---|---|---|---|---|
| $e_3$ | 4.4151E-007 | 2.7251E-008 | 1.6907E-009 | 1.0831E-010 | 2.6539E-012 |
| $Log_r(\frac{e_3(h)}{e_3(\frac{h}{2})})$ | - | 4.0181 | 4.0106 | 3.9644 | 5.3509 |

for which the exact solution is known as $u(x,t) = e^{t+x}$.

First, let $v(x,t) = \frac{\partial u}{\partial x}$, then the equivalent system of two reaction-diffusion equations is

$$\begin{aligned} u_t &= \frac{1}{2}u_{xx} + \frac{1}{2}v \\ v_t &= \frac{1}{2}v_{xx} + \frac{1}{2}u_{xx} \end{aligned} \tag{23}$$

with the following initial and boundary conditions:

$$u(x,0) = e^x \quad ,v(x,0) = e^x \qquad for \quad x \in (0,1) \tag{24}$$

$$u(0,t) = e^t, \quad u(1,t) = e^{1+t}, \qquad for \quad t \in (0,1). \tag{25}$$

Note the boundary conditions for $v(x,t)$ are derived in the previous section.

The data in Table 1 shows the maximum errors between the calculated and exact solutions at $T = 1$. $e_1$ is the maximum error for a small and fixed $h = 0.0001$ while $\Delta t$ is varying. It clearly shows that when $\Delta t$ is reduced by a factor of $r$, $e_1$ is reduced by a factor about $r^2$, i. e., the algorithm is second oder accurate in time. $e_2$ is the same as $e_1$ except that Richardson extrapolation is used. One can easily see that when $\Delta t$ is reduced by a factor of $r$, $e_2$ is reduced by a factor about $r^4$, i.e., the algorithm with Richardson's extrapolation is fourth oder accurate in time.

The data in Table 2 shows the maximum errors between the calculated and exact solutions at $T = 1$. $e_3$ is the maximum error for a small and fixed $\Delta t = 0.0001$ while $h$ is different. Again, the reason we use a very small value of $\Delta t$ is to make sure that the dominated error is from $h$. It clearly shows that when $h$ is reduced by a factor of $r$, $e_3$ is reduced by a factor about $r^4$, i. e., the algorithm is fourth oder accurate in space.

The data in Table 3 shows that the algorithm with Richardson's extrapolation is fourth order accurate in both time and space. $e_4$ is the maximum error with $\Delta t = h$ both been reduced by a factor of $r$, and we can see that $e_4$ is reduced by a factor about $r^4$.

Table 3:

| $\Delta t = h$ | 0.1 | 0.05 | 0.025 | 0.0125 | 0.01 |
|---|---|---|---|---|---|
| $e_4$ | 3.1866E-007 | 1.9885E-008 | 1.2336E-009 | 7.6730E-011 | 3.1560E-011 |
| $Log_r(\frac{e_4(h)}{e_4(\frac{h}{2})})$ | - | 4.0023 | 4.0107 | 4.0069 | 3.9814 |

### 5.2 Case 2: Nonlinear Black-Scholes equation

We solve the reformulated nonlinear Black-Scholes equation (4) with initial condition (5) and boundary conditions (6)-(7). The derivation and reformulation of the model were briefly discussed early in section 1 and section 2. The nonlinearity is treated explicitly, see [6]. The ODE (2) is solved for future use by high order numerical method and the solution is plotted in Fig. (1). More details can be found in both [2] and [6]. Note that the initial data for the nonlinear Black-Scholes equation (4) is continuous but not differentiable at $x = 0$, so cubic spline interpolation was used to smooth the initial data, and the original data and smoothed initial data are plotted in Fig. (2).

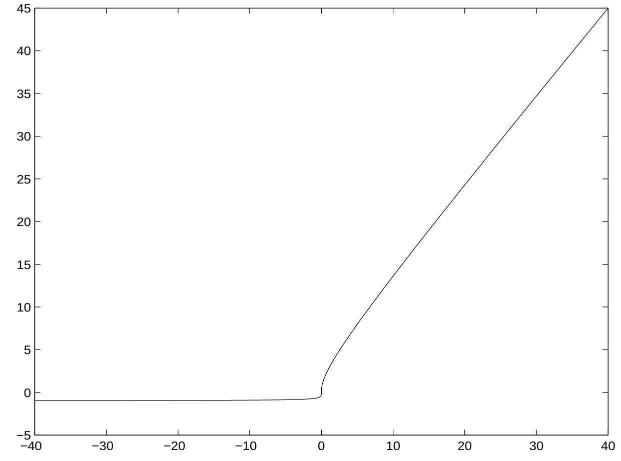
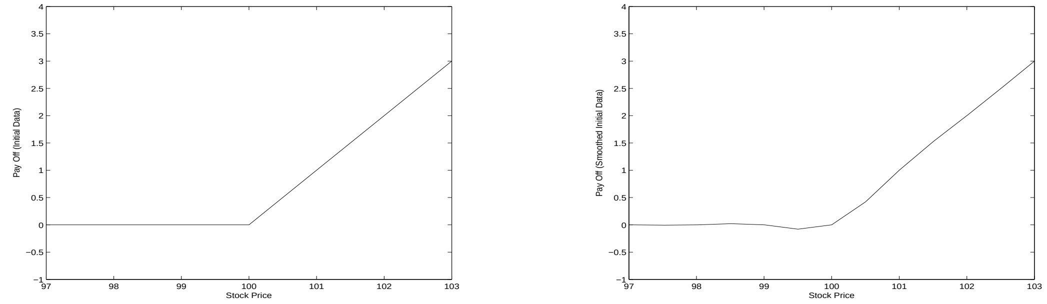


Figure 1: Numerical solution to the ODE (2)

Fig.(3) shows the influence of the transaction costs on the price of the European Call option. The nonlinear Black-Scholes equation (4) is solved by the proposed high order compact scheme for different transaction costs: $a = 0.0, 0.01, 0.02$ and $0.03$, while other model parameters are fixed as: $\sigma_0 = 0.2$, $\rho = 0.1$, $E = 100$ and $T = 0.02$. One can see that for the same pay-off function, when transaction costs $a$ increase, the corresponding option prices also increase. Not surprisingly, the lowest price curve is the case when $a = 0.0$, .i.e, no transaction costs charged.



(a) (b)

Figure 2: Comparison between original and smoothed initial conditions. (a) Original initial data which is non-differentiable at $x = 0$, (b) Smoothed initial data which is differentiable at $x = 0$

# 6 Conclusions

An efficient fourth-order numerical algorithm based on the Padé approximation and the Richardson's extrapolation had been derived in this paper. The algorithm was mainly derived to provide fast, accurate and robust option pricing with transaction cost is taken into account, however it can also be used to solve any convection-diffusion-reaction problem, especially for the problem with nonlinear reaction term. One possible argument to the algorithm is that, instead of solving a single convection-diffusion-reaction equation, this algorithm actually solves a system of two reaction-diffusion equations. This is not true, at least if the problem is Black-Scholes equation. It is well known that $v(s,t) = \frac{\partial u}{\partial s}$ is nothing but the Greeks Delta, which means the number of shares one should hold and needs to be calculated during trading. Therefore nothing is wasted by solving an additional equation. Numerical results also show that the new algorithm performs significantly better than many other classical schemes such as Crank-Nicolson, forward time central space, and R3A, R3B, R3C in [15]. More precisely,

1. High order accuracy: It is fourth order in both time and space;

2. Compact Scheme: In each time step, each iteration involves solving a tridiagonal system, and boundary condition approximation(fourth order) only involves solutions on 3 grid points;

3. Strong stability and non-oscillatory condition therefore no restrict on time step. This is extremely useful when the problem is solved on a long time interval;

4. Greeks Delta is automatically calculated with higher order accuracy, no additional work is needed.

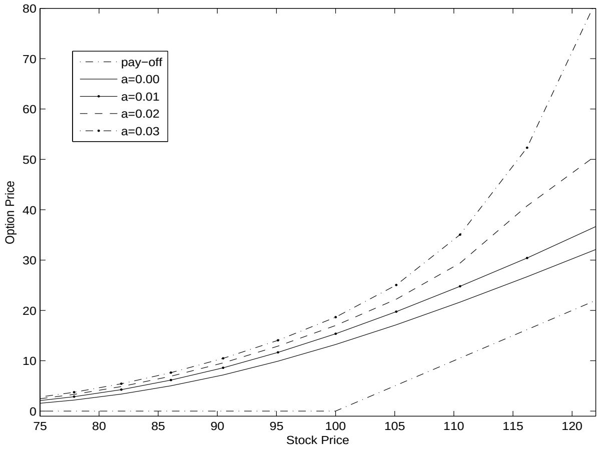


Figure 3: Option prices for different transaction cost

In the future, the authors plan to extend the algorithm to deal with higher dimensional nonlinear Black-Scholes equations, with different parameters, and conduct stability analysis.

## Acknowledgment

The research of second author( AQMK) is supported by FRCP grant, summer 2007, Middle Tennessee State University.

## References

[1] G. Barles, Convergence of numerical schemes for degenerate parabolic equations arising in finance theory, in *Numerical Methods in Finance*, L. C. G. Rogers and D. Talay, etds., Cambridge U. Press, 1997.

[2] G. Barles, H. M. Soner. Option pricing with transaction costs and a nonlinear Black-Scholes equation. *Finance Stochast.* **2**, 369 - 397(1998).

[3] F. Black, M. S. Scholes, The pricing of options and corporate liabilities, *J. Political Economy*, **81**, 637 - 654(1973).

[4] P. Boyle, T. Vorst. Option replication in discrete time with transaction costs. *J. Finance* **47**, 271 - 293(1973).

[5] M. Davis, V. Panis, T. Zariphopoulou. European option pricing with transaction fees. *SIAM J. Contr. Optim.* **31**, 470 - 493(1993).

[6] B. During, M. Fournie and A. Jungel, High Order Compact Finite Difference Schemes for a Nonlinear Black-Scholes Equation,*Intern. J. Theor. Appl. Finance* **6**, 767 - 789(2003).

[7] R. S. Hirsh, Higher order accurate difference solution of fluid mechanics problems by a compact difference technique. *Journal of Computational Physics,* **9**:90 - 109(1975).

[8] H. Leland, Option pricing and replication with Transaction Costs, *Journal of Finance*, **5**, 1283 - 1301(1985).

[9] W. Liao, J. Zhu, and A. Q. M. Khaliq, An efficient high order algorithm for solving systems of reaction-diffusion equations, *Journal of Numerical Methods for Partial Differential Equations*, **18**, 340 - 354 (2002).

[10] Y. Gu, W. Liao, and J. Zhu, An efficient high order algorithm for solving systems of 3-D reaction-diffusion equations, *Journal of Computational and Applied Mathematics*, **155**, 1 - 17 (2003).

[11] W. Liao, J. Zhu and A.Q.M. Khaliq, A fourth-Order Compact Algorithm for Nonlinear Reaction-Diffusion Equations with Neumann Boundary Conditions, *Numer Methods Partial Differential Equations* , **22**,600 - 616(2006).

[12] A. Parás, M. Avellaneda. Dynamic hedging portfolios for derivative securities in the presence of large transaction costs. *Appl. Math. Finance* **1**, 165 - 193, 1994. 2,3

[13] A. Rigal, Numerical Analysis of 2-level finite-difference schemes for unsteady diffusion-convection problems. *International Journal for numerical methods in Engineering* **28**(5): 1001 - 1021, 1989.

[14] A. Rigal, Numerical Analysis of 3 -time-level finite-difference schemes for unsteady diffusion-convection problems. *International Journal for numerical methods in Engineering* **30**(2): 307 - 330, 1990.

[15] A. Rigal, High order difference scheme for unsteady one-dimensional diffusion-convection problems. *Journal of Computational Physics*,**114**, 59 - 76 (1994).

[16] W. F. Spotz and G. F. Carey, Extension of high-order compact schemes to time dependent problems, *Numer. Methods Partial Differential Equations*, **17**, 657 - 672(2001).

[17] V. Zakamouline, Hedging of Option Portfolios and Options on Several Assets With Transaction Costs and Nonlinear Partial Differential Equations" (January 18, 2007). Available at SSRN: http://ssrn.com/abstract=938933

[18] H. Sun and J. Zhang, A high order finite difference discretization strategy based on extrapolation for convection diffusion equations, *Numerical Methods for Partial Differential Equations*, **20**, no.1, 18-32(2004).

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# Asian Options as Ultradiffusion Processes

**Michael D. Marcozzi**[1]

[1] *Department of Mathematical Sciences, University of Nevada Las Vegas*

emails: `marcozzi@unlv.nevada.edu`

**Abstract**

An ultradiffusion is a process which is isomorphic to a parameterized diffusion along a characteristic temporal trajectory. They are motivated by the realization that in many systems exogenous sources of uncertainty enter only certain components of the dynamics. The value function associated with ultradiffusion processes are characterized as the solution to ultraparabolic equations. We consider the arithmetic-average Asian option as a prototypical example of an ultradiffusion processes.

*Key words: Ultradiffusion, Ultraparabolic, Option Pricing*
*MSC 2000: 35K70, 91B28*

## 1 Introduction

An ultradiffusion is a processes which is isomorphic to a parameterized diffusion along a characteristic temporal trajectory; they are motivated by the realization that in many systems exogenous sources of uncertainty enter only certain components of the dynamics (cf. [7], [11], [17]). The value function of an ultradiffusion process is characterized as the solution to an ultraparabolic equation. Ultraparabolic equations evidence multiple temporal variables and are parabolic along characteristic directions. In this paper, we consider as a prototypical example the valuation problem for the arithmetic-average Asian option.

Historically, an interest in ultradiffusion processes and ultraparabolic equations arose relative to the works of Kolmogorov [12], [13] and Uhlenbeck and Ornstein [21] in connection with Brownian motion in phase space and Chandrasekhar [3] with respect to the theory of boundary layers. Unlike parabolic equations, however, neither the strong maximum principle nor interior *a priori* estimates, for example, hold for ultraparabolic equations (cf. [8], [6], [19], [20], [22], [14], [18]).

The outline of this paper is as follows. In section 2, we define the valuation problem in the Black-Scholes framework; this example serves as a basis for subsequent comparison. In section 3, the Asian option is introduced as the expectation of an ultradiffusion process and its value function is seen to satisfy an ultraparabolic equation. The representation of the ultradiffusion as a parameterized diffusion is considered in section 4. In section 5, we consider a fully stochastic approximation of the ultradiffusion as the limit of vanishing viscosity diffusions in the context of the so-called viscosity formulation. Finally, in section 6, the viability framework is briefly introduced which approximates the ultradiffusion in a deterministic context.

## 2 Black-Scholes Framework

As a benchmark, we consider the Black-Scholes option pricing framework in which we model the motion of a stock price $S_t$ by geometric Brownian motion

$$\frac{d\,S_t}{S_t} = r\,ds + \sigma\;dB_t\,, \tag{2.1}$$

for all $t > 0$, such that $S_0 = S$, volatility $\sigma > 0$, and risk-free rate of return $r > 0$. We associate with (2.1) the infinitesimal generator $\mathcal{A}$ given by

$$\mathcal{A}(t)\,u(t,S) = \frac{1}{2}\sigma^2\,S^2\frac{\partial^2 u}{\partial S^2}(t,S) + r\,S\,\frac{\partial u}{\partial S}(t,S) \tag{2.2a}$$

which applies along the (trivial) characteristic direction of the evolutionary operator

$$\mathcal{H}(S)\,u(t,S) = \frac{\partial u}{\partial t}(t,S)\,. \tag{2.2b}$$

The value function associated with (2.1), payoff $\psi(S)$, and discounting factor $r$, is then

$$u(t,S) = \mathbb{E}\left\{\exp\left[-r\cdot(T-t)\right]\cdot\psi(S_T)\right\}\,. \tag{2.3}$$

In the context of option pricing, the valuation (2.3) represents the present value or price of an European-style option exercised at $T > 0$; upon exercise, the contract delivers to the option holder a payoff amount $\psi(S)$. In particular, the value function can be characterized as the unique solution to the terminal-value parabolic equation

$$\mathcal{H}(S)\,u(t,S) + \mathcal{A}(t)\,u(t,S) - ru(t,S) = 0 \text{ on } [0,T)\times(0,\infty)\,, \tag{2.4a}$$

such that

$$u(T,S) = \psi(S) \text{ on } [0,\infty)\,. \tag{2.4b}$$

Approximation techniques dealing with valuations problems relative to diffusion processes are found in Marcozzi [16].

## 3 Asian Options

In contrast to the preceding section, Asian options evidence dependence on the path-history of the underlying stochastic process. To this end, we suppose that the payoff for the Asian option depends on the arithmetic average, or

$$\varsigma(t) = \int_0^t S_\tau\,d\tau\,,$$

in which case we consider the ultradiffusion process

$$d\,\varsigma_t = S_t\,dt\,, \tag{3.1a}$$

$$\frac{d\,S_t}{S_t} = r\,dt + \sigma\;dB_t\,, \tag{3.1b}$$

for all $t > 0$. The prefix “ultra” refers to the presence of multiple time variables and will be explained subsequently.

We associated with (3.1) the infinitesimal generator, depending now upon the temporal pair $(t, \varsigma)$, such that

$$\mathcal{A}(t,\varsigma)\,u(t,\varsigma,S) = \frac{1}{2}\sigma^2\,S^2\frac{\partial^2 u}{\partial S^2}(t,\varsigma,S) + r\,S\,\frac{\partial u}{\partial S}(t,\varsigma,S)\,, \tag{3.2a}$$

which applies along the characteristic direction of the evolutionary operator, which is now

$$\mathcal{H}(S)\,u(t,\varsigma,S) = \frac{\partial u}{\partial t}(t,\varsigma,S) + S\,\frac{\partial u}{\partial \varsigma}(t,\varsigma,S)\,. \tag{3.2b}$$

Indeed, the key idea is that in both (2.2) and (3.2) the temporal operator $\mathcal{H}$ is hyperbolic.

Along with the ultradiffusion process (3.1), payoff $\psi(\varsigma, S)$, and discounting factor $r$, we define the value function as

$$u(t,\varsigma,S) = \mathbb{E}\left\{\exp\left[-r\cdot(T-t)\right]\cdot\psi(\varsigma_T, S_T)\right\}\,. \tag{3.3}$$

That is, the valuation (3.3) represents the present value or price of an arithmetic-average Asian option exercised at $T > 0$, which delivers to the option holder a payoff amount $\psi(\varsigma_T, S_T)$ dependent upon the path-history $\varsigma_T$ of the asset $S_t$. In particular, the value function (3.3) can be characterized as the unique solution to the terminal-value ultraparabolic equation

$$\mathcal{H}(S)\,u(t,\varsigma,S) + \mathcal{A}(t,\varsigma)\,u(t,\varsigma,S) - r\,u(t,\varsigma,S) = 0 \text{ on } [0,T)\times(0,\infty)\times(0,\infty)\,, \tag{3.4a}$$

subject to the terminal condition

$$u(T,\varsigma,S) = \psi(\varsigma,S) \text{ on } [0,\infty)\times(0,\infty)\,. \tag{3.4b}$$

Approximation techniques dealing with valuations problems relative to ultradiffusion processes are found in Marcozzi [17].

# 4 Parametric Representation

We may also use the defining characteristic of ultradiffusion processes, namely their equivalence to a parametric family of diffusion processes, in order to value an Asian option. To this end, we effect a change to the characteristic time $\xi$ such that the ultradiffusion (3.1) becomes a $\varsigma$-indexed family of diffusion processes along the characteristic direction $(1, S)$ in the $(t, \varsigma)$-temporal plane. To this end, for each value of the state variable $S$, we define the so-called characteristic time $\xi$ such that

$$\xi = t + \frac{\varsigma}{S}\,, \tag{4.1}$$

in which case it follows from the scaling property of Brownian motion that

$$d\,S_\xi = r\,S_\xi\,d\xi + \sigma\,S_\xi\,dB_\xi\,, \tag{4.2}$$

for all $\xi > 0$ (cf. [10, Lemma II.9.4]). Note that by (4.1), it follows that

$$d\,\varsigma = S\,d\,\xi = S\,dt\,,$$

in which case we implicitly recover (3.1a). Significantly, the infinitesimal generator associated with the parameterized diffusion (4.2) with respect to the state variable $S$ does not change and is

$$\mathcal{A}_\varsigma(\xi)\,u(\xi, S) = \frac{1}{2}\sigma^2\,S^2\frac{\partial^2 u}{\partial S^2}(\xi, S) + r\,S\,\frac{\partial u}{\partial S}(\xi, S)\,, \tag{4.3a}$$

and operates along the characteristic time direction $\xi$ subject to the evolutionary operator behavior generated by

$$\mathcal{H}(S)\,u(\xi, S) = \frac{\partial u}{\partial \xi}(\xi, S)\,. \tag{4.3b}$$

With respect to $\xi$ and (4.1), the $r$-discounted value function becomes

$$u^\varsigma(\xi, S) = \mathbb{E}\left\{\exp\left[-r\cdot(\Gamma - \xi)\right]\cdot\psi^\varsigma(S_\Gamma)\right\}\,, \tag{4.4}$$

where $\psi^\varsigma(S) = \psi(\varsigma, S)$ and $\Gamma = T + \varsigma/S$. In particular, we characterize the value function $u^\varsigma$ as the unique solution to the parameterized terminal-value parabolic equation

$$\mathcal{H}(S)\,u^\varsigma(\xi, S) + \mathcal{A}_\varsigma(\xi)\,u^\varsigma(\xi, S) - r\,u^\varsigma = 0 \;\; a.e. \text{ on } [0, \Gamma)\times(0, \infty)\,, \tag{4.5a}$$

such that

$$u^\varsigma(\Gamma, S) = \psi^\varsigma(S) \text{ on } (0, \infty)\,, \tag{4.5b}$$

(cf. [2, §III.2.15]). We remark that for a given $\varsigma$, solving (4.5), is significantly simplier than solving (3.4) and lies within the Black-Scholes framework of section 2.

In the context of ultradiffusion processes (3.1), the variable $t$ is referred to as (standard) time, $\varsigma$ as parametric time, $\xi$ as the characteristic time, $(t, \varsigma)$ as the temporal pair, and $S$ as the state variable. That is, the state variable $S$ evolves within the $(t, \varsigma)$-plane equivalently: (i) according to the ultradiffusion (3.1) or (ii) according to the $\varsigma$-indexed diffusion (4.2) along the characteristic curve (4.1). As an ultradiffusion, the value function (3.3) satisfies an ultraparabolic equation (3.4), whereas the value function (4.4) relative to the parametric diffusion representation (4.2) is characterized as the unique solution to the parameterized parabolic equation (4.5). The "ultra" prefix then refers, in both the diffusion process and parabolic operator, to evolution in the temporal $(t, \varsigma)$-plane.

## 5 Viscosity Approximation

One may consider the approximation of ultradiffusions as the limit of vanishing viscosity diffusion processes. That is, we considering the fully stochastic approximation of (3.1) by an $\varepsilon$-perturbed diffusion

$$d\,\varsigma(t) = S_t\,dt + \varepsilon\,dB_t^{(1)}\,, \tag{5.1a}$$

$$\frac{d\,S_t}{S_t} = r\,dt + \sigma\,dB_t^{(2)}\,, \tag{5.1b}$$

for all $t > 0$, as $\varepsilon \to 0^+$, where $(B^{(1)}, B^{(2)})$ is an $\mathbb{R}^2$-valued Brownian motion (cf. [5, §VI.8]). In this case, the infinitesimal generator of the state variables $(\varsigma, S)$ takes the form

$$\mathcal{A}_\varepsilon(t)\,u(t,\varsigma,S) = \frac{1}{2}\sigma^2\,S^2\frac{\partial^2 u}{\partial S^2}(t,\varsigma,S) + \frac{1}{2}\varepsilon^2\frac{\partial^2 u}{\partial \varsigma^2}(t,\varsigma,S) \tag{5.2a}$$

$$+r\,S\,\frac{\partial u}{\partial S}(t,\varsigma,S) + S\,\frac{\partial u}{\partial \varsigma}(t,\varsigma,S)$$

and evolves along the (trivial) characteristic direction of the evolutionary operator

$$\mathcal{H}(\varsigma,S)\,u(t,\varsigma,S) = \frac{\partial u}{\partial t}(t,\varsigma,S)\,. \tag{5.2b}$$

Along with the approximation process (5.1), payoff $\psi(\varsigma, S)$, and discounting factor $r$, we define the value function as

$$u_\varepsilon(t,\varsigma,S) = \mathbb{E}\left\{\exp\left[-r\cdot(T-t)\right]\cdot\psi(\varsigma,S)\right\}\,. \tag{5.3}$$

In particular, the value function (5.3) relative to the diffusion (5.2) can be characterized as the unique solution to the terminal-value perturbation parabolic equation

$$\mathcal{H}(\varsigma,S)\,u_\varepsilon(t,\varsigma,S) + \mathcal{A}_\varepsilon(t)\,u_\varepsilon(t,\varsigma,S) - r\,u_\varepsilon(t,\varsigma,S) = 0 \text{ on } [0,T)\times(0,\infty)\times(0,\infty)\,, \tag{5.4a}$$

subject to the terminal condition

$$u_\varepsilon(T,\varsigma,S) = \psi(\varsigma,s) \text{ on } [0,\infty)\times(0,\infty)\,, \tag{5.4b}$$

for all $\varepsilon \to 0^+$. We note, however, that while so-called viscosity solutions formalize this approximation of the ultradiffusion, in general viscosity solutions lack the requisite regularity for the construction of robust numerical procedures (cf. [9], [4]). Indeed, in viscosity formulations, one typically needs to numerically solve the advection dominated equation (5.4) in order to value (3.3) via (5.3) as $\varepsilon \to 0^+$, which is a much more significant challenge than solving (3.4) or (4.4) directly (cf. [16]).

# 6 Viability Solutions

Conversely, one may also approximate the ultradiffusion within a fully deterministic framework of viability solutions. Viability techniques formulate the tychastic optimal control problem as a deterministic dynamical game such that

$$d\,\varsigma(t) = S_t\,dt \tag{6.1a}$$

$$\frac{d\,S_t}{S_t} = r\,dt + \varepsilon_i\,\sigma\sqrt{\Delta}\,, \tag{6.1b}$$

where $\Delta > 0$ and $\varepsilon_i \in [-1, 1]$, for $i \in V$ and $V \subset \mathbb{N}$ sufficiently large. In this case, the process is purely deterministic with evolutionary component

$$\mathcal{H}\,u(t,\varsigma,S) = \frac{\partial u}{\partial t}(t,\varsigma,S) + rS\frac{\partial u}{\partial \varsigma}(t,\varsigma,S) + (\varepsilon_i\,\sigma\sqrt{\Delta} + r)S\frac{\partial u}{\partial S}(t,\varsigma,S)\,. \tag{6.2a}$$

In order to obtain the value function, one would optimize the deterministic control problem over a set of perturbations $\varepsilon_i$ subject to certain viability constraints (cf. [1]). Significantly, the tychastic formulation does not converge to the stochastic problem as $\mathrm{card}(V) \to \infty$ and $\Delta \to 0^+$.

## Acknowledgements

This research has been supported by the U.S. National Science Foundation, award number DMI-0422985.

## References

[1] J.-P. Aubin, *Viability Theory*, Birkhäuser, Boston, 1991.

[2] A. Bensoussan and J. L. Lions, *Applications of variational Inequalities in Stochastic Control*, North Holland, Amsterdam, 1982.

[3] S. Chandrasekhar, *Stochastic Problems in Physics and Astronomy*, Rev. Mod. Phys. **15** (1943) 1–89.

[4] M. G. Crandell, H. Ishii and P.-L. Lions, *User's guide to viscosity solutions of Hamilton-Jacobi equations*, Bull. Amer. Math. Soc. **27** (1992) 1–67.

[5] W. H. Fleming and R. W. Rischel *Deterministic and Stochastic Optimal Control*, Springer-Verlag, New York, 1975.

[6] T. G. Genčev, *Ultraparabolic equations*, Dokl. Akad Nauk SSSR **151** (1963) 265–268.

[7] B. A. Huberman and M. Kerszberg, *Ultradiffusions: The relaxation on hierarchical systems*, J. Phys. A: Math Gen. **18** (1985) L331–L336.

[8] A. M. Il'in, *On a class of ultraparabolic equations*, Dokl. Akad Nauk SSSR **159** (1964) 1214–1217.

[9] H. Ishii, *On uniqueness and existence of viscosity solutions for fully nonlinear second order elliptic partial differential equations*, Comm. Pure Appl. Math. **42** (1989) 15–45.

[10] I. Karatzas and S. E. Shreve *Brownian Motion and Stochastic Calculus*, Second edition, Springer-Verlag, New York, 1991.

[11] A. N. Kochubei, *Pseudo-Differential Equations and Stochastic over Non-Archimedian Fields*, Marcel Dekker, New York, 2001.

[12] A. N. Kolmogorov, *Sur theorie der stetigen zufalligen progresse*, Math. Annal. **108** (1933) 149–160.

[13] A. N. Kolmogorov, *Zufällige Bewegungen*, Ann. of Math. **35** (1934) 116–117.

[14] M. Manfredini and S. Polidoro, *Interior regularity for weak solutions of ultraparabolic equations in divergence form with discontinuous coefficients*, Bollettino **8** (1998) 651–675.

[15] M. D. Marcozzi, *On the approximation of optimal stopping problems with applications to financial mathematics*, SIAM J. Sci. Comput. **22** (2001) 1865–1884.

[16] M. D. Marcozzi, *On the valuation of Asian options by variational methods*, SIAM J. Sci. Comput. **24** (2003) 1124–1140.

[17] A. Maritan and A. L. Stella, *Exact renormalization group approach to ultradiffusion in a hierarchical structure*, J. Phys. A: Math Gen. **19** (1986) L269–L274.

[18] S. Polidoro and M. A. Ragusa, *Sobolev-Morrey spaces related to an ultraparabolic equation*, Manuscripta Math. **96** (1986) 371–392.

[19] S. A. Tersenov, *On boundary value problems for a class of ultraparabolic equations, and their applications*, Math. Sbornik **133** (1987) 539–555.

[20] S. A. Tersenov, *Basic boundary value problem for one ultraparabolic equation*, Siberian Mathematical Journal **42** (2001) 1173–1189.

[21] G. E. Uhlenbeck and L. S. Ornstein, *On the theory of the Brownian motion*, Phys. Rev. **36** (1930) 823–841.

[22] V. S. Vladimirov and Ju. N. Drožžinov, *Generalized Cauchy problem for an ultraparabolic equation*, Izv. Akad. Nauk. SSSR Ser. Mat. **31** (1967) 1341–1360.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE2007*
*Chicago, 20-23 June 2007*

# Undergraduate Computational Physics Education: Coming of Age?

**R. F. Martin, Jr.**
*Department of Physics, Illinois State University*

email: rfm@phy.ilstu.edu

## Abstract

While some physics educators have included computing in courses and have developed specialized courses for over 40 years, computational physics education has only slowly made inroads into the broader physics education community. There has been a recent surge in interest in a more global approach to computational physics education that offers promise for computing to finally take an important role in the education of undergraduate physicists. In this presentation I will review some developments in computational physics education and present examples from the program at Illinois State University.

*Key words: undergraduate education, computational physics*

## 1. Introduction

The use of computing in physics teaching has been practiced at least since the 1960's, with, for example, the pioneering work of Alfred Bork[1]. My department joined the fray in the mid-1970's[2] when a wave of new specialized computing-based physics courses were developed at many institutions across the nation. The 1980's saw an expansion of computational physics course offerings, the emergence of several influential textbooks such as those by Koonin[3] and Gould and Tobochnik[4], as well as a groundbreaking conference, the *Conference on Computers in Physics Instruction* in 1988. This conference served to bring together a wide variety of proponents of computational methods in physics education covering uses in introductory courses, laboratories, computer-aided-instruction, courses devoted to computational methods, as well as computation-based curricular modifications. The proceedings[5] from this conference can still serve as a valuable reference for those interested in why and how to develop computational physics modules, classes, and curricula. I count myself as one attendee strongly influenced by this meeting such that, with several enthusiastic departmental colleagues, our department developed a computational physics curriculum[6,7] in the 1990's - in parallel with similar developments at other institutions in that decade and into the present one.

Although many individuals have developed course materials and advocated for computational physics education over these four decades, and journals such as the American Journal of Physics, Computers in Physics, and Computing in Science and Engineering (CiSE) have provided outlets for new developments, the field has never really had a professional “home”. In the current decade, this situation appears to be changing. With special sessions at the American Physical Society (through the Forum on Education) March Meeting in 2004, the American Association of Physics Teachers (AAPT) summer meeting in 2006, a topical conference sponsored by AAPT to be held in July 2007, and a Gordon conference planned for 2008, there is new momentum in the field.

In this presentation I will give a brief overview of some developments in computational physics education and then present one model for a successful program developed at Illinois State.

## 2. Computational Physics Education

First, it is worth noting what is meant by computational physics education (CPE). CPE differs from computer science and computer engineering education in that CPE focuses on only those applications of computing that are relevant to the solution of physics problems. For the purposes of this presentation, we exclude applications of computing in laboratory instrumentation and in computer-aided-instruction (in which the teacher authors software to aid student learning), leaving primarily the use of numerical methods to solve intractable equations in physics, the broad domain of computational modeling and simulation, and scientific visualization. Perhaps the key point is that students learn to apply the tools of CPE to learn physics and to model and solve problems in physics.

The tools of CPE have changed significantly from the early days of Fortran programs on a mainframe computer. Today one has access to wide array of programming languages, graphics/visualization suites, web-based tools, and comprehensive analysis packages, to be utilized on an equally broad array of hardware from laptops to workstations to grid networks to massively parallel supercomputers. Based on the discussion at the AAPT special session in 2006, the main languages used are C/C++, Fortran, Python, and Java. Participants appeared to split roughly equally between the Mathematica and Matlab/Maple comprehensive packages that provide programming, symbolic math, and visualization capabilities. A wealth of new ideas of how to use these tools in educating physics students is evident in the articles and web-extras published in a special issue[8] of CiSE.

While the tools have improved markedly, the basic function of CPE has changed remarkably little over the decades. In a 1963 article on the subject, Bork[1] presents several reasons for using computation in physics courses, including (1) “The modern computer makes workable problems which were for all practical purposes impossible only a few years ago”, (2) “The 'feel' of numbers promotes understanding of analytic relations", and (3) “using numerical analysis permits the consideration of material beyond the students' analytic ability”. Similarly, Martin *et al*[6] articulated in 1991 three guiding principles for their program: (1) “Students should be in command of the technology, not the other way around”, (2) “The computer should enhance the student's acquisition of broad problem-solving skills, and (3) “The computer should stimulate: (a) a re-ordering and broadening of the subject matter taught; (b) build physical intuition; and (c) enable the student to pursue independent study.” All these ideas, and more, are still part of the argument for CPE. In a recent article, Landau[9] goes even further and argues that CPE presents a superior model for teaching physics generally: “presenting physics within a scientific problem-solving paradigm is a more effective and efficient way to teach physics than the traditional approach.” In his model computational science provides both elements of its own and a bridge between the three disciplines of computational physics: application physics, mathematics, and computer science. In the article he presents a detailed concept map of the complex interactions between these sub-areas.

All of the above cited articles, and many more, point out one other justification for CPE: practicality. Bork[1] cites a comment by Courant: “the vital task is ... to inject the awareness of the potentialities of modern computing into the general education of talented and open-minded scientists.” Even in 1963 it was clear to those with vision that computational science would be a critical aspect of *doing science* in the future - and that future is certainly here. Study after study (see for example the American Institute of Physics Statistical Research Center website[10]) have shown the importance of computational modeling and programming skills both in physics research careers and in the variety of careers available to those with one or more physics degrees. As Chonacky[11] comments, “Contemporary sciences and engineering practices are cross-disciplinary; these people not only compute but also use computation as a common interface between their respective contributions to cross-disciplinary projects.” Computational science and engineering is simply a part of the scientific and engineering landscape in the 21st century.

One might reasonably ask why it should be necessary to provide justification for CPE at all. To many, the above arguments may appear nearly self-evident. Yet, during the discussion at the 2006 AAPT conference, and reflected in some of the articles in the special edition of CiSE, it is clear that there is still something of a divide between those seeking to encourage CPE and those, for lack of a better phrase, who are proponents of a “traditional analytic-plus-laboratory” mode of physics education (note that even strong advocates of contemporary active learning approaches to physics education can still be “traditionalists” regarding CPE). After presenting results of a significant survey of physics department Chairs and faculty indicating strong agreement about the role of computational physics in practicing physicists' lives and in course preparation, Fuller[12] reports responses to one telling question that suggest that physics educators have not yet translated this idea into their courses. In response to a question asking what percentage of faculty utilize computational assignments in their courses, “fewer than 20 percent of the physics faculty ... include computations in their grading.” Thus, CPE supporters still have their work cut out for them.

Such constraints aside, a considerable amount has been achieved. Landau[9] lists five US universities and colleges with undergraduate degree programs in computational physics, and another four with non-degree specializations (minor, concentration, option, etc.) offered. An additional four institutions offer computational science degrees, with another 11 providing some sort of specialization in that interdisciplinary area. Other departments are adding computation to the physics curriculum through focused courses or by integrating computing into existing courses, as the lists of presentations at the recent conferences demonstrate. It is this momentum that provides evidence that CPE is finally coming-of-age.

## 3. One model: Illinois State University

The physics department at Illinois State has gone through several phases of development of CPE. While each department will have its unique set of strengths and educational goals, and will therefore develop in its own way, it may be of value to know how others have solved common problems during their evolution. Other examples and models can be found in the literature, including the references cited in this article.

The first phase in our development was the creation in the mid-1970's of a single course, *Computers in Physics*, designed for junior/senior physics majors. As a single-instructor course offered mainly to interested students, there were no issues of acceptance by colleagues.

The next phase began in the late 1980's when a core of three faculty who utilized computational methods in their research began to discuss the possibility of integrating computing into various physics classes[6]. The original motivation was two-fold: the recognition that practicing scientists need computational skills and the fact that our fledgling undergraduate research program often involved computational tasks for students. Initially limited to courses taught by the three originators, by the mid-1990's new faculty hires had brought several more computationally-oriented physicists to the department and a full integration of computing into post-introductory physics majors courses was achieved. This integration was accomplished using a consensus-determined “skills matrix”, which outlined the computational skills to be introduced at various levels in the curriculum. This project was partially supported by an NSF grant. While there was some departmental resistance to this process, open discussion and consensus-building, combined with the fortunate situation of a preponderance of computational researchers in the department, allowed the process to move relatively smoothly.

The third phase in CPE development was spurred by input from graduates of the computationally integrated curriculum, who were quite enthusiastic about their abilities to build and solve computational models, expertise that served them well in both graduate school and in technical jobs. We also felt that the new program would help with major recruitment, a significant problem in the late 1990's in physics departments across the country. Funding from the NSF ILI/LLD program allowed us to construct a degree sequence to give interested students even deeper experience with computational physics. After surveying high school students on their interest in such a program we selected the title “Computer Physics” for the degree, since these pre-college students identified the word 'computational' with performing complicated mathematical computations and not with computers.

### 3.1 *Computational integration in physics courses*

The Illinois State University physics department offers four degree sequences: physics, computer physics (CP), physics teacher education (PTE), and engineering physics (a 3/2 program with partner engineering universities). Most courses taken primarily by majors in these four sequences have a computational component. Only our first two introductory physics classes do not, mainly because they also act as service courses for other departments. Majors are first introduced to graphical data analysis with scientific graphics software and to Mathematica in the first-semester freshman “introduction to the discipline” class. This is followed by simple programming assignments in a lab format in the third-semester introductory course, in which the students write simple finite-difference codes (both Euler and a second order leapfrog method) to solve some mechanics problems and a basic Monte Carlo simulation for an optics problem. As the students progress further in their physics courses, more sophisticated computational methods are introduced, following our original idea of a “skills matrix”, as mentioned above. For example, in the intermediate electromagnetism course, before students have taken a differential equations class, they solve Laplace's equation using an over-relaxation algorithm. Solving Laplace's equation on a grid in this way helps students understand the mathematical and physical properties of electrostatic potentials, and visualizing the results as contour, raster, and surface plots further develops physical intuition. In the senior-level quantum physics class, students compute matrix elements for a double well potential and use the QL algorithm to compute eigenvalues and eigenfunctions, a problem that would be beyond their mathematical expertise to work out analytically.

Our philosophy of computational integration lies somewhere in-between Landau's[9] “computational physics--education”, in which students use black-box computational methods to help them learn physics and his “computational--physics education”, which implies that computation and physics are more fully merged. Our students learn the algorithms in enough depth that they can understand their relation to the physics, but not at a deep level. Students do write the code, bust most instructors prefer to provide strategic chunks of code for students to construct their programs around. In this way graduates in all our degree sequences leave our department with some expertise in computing and, according to student evaluation comments, a better understanding of the physics.

### 3.1 *The computer physics degree sequence*

Realizing that students are learning some computation in their physics courses, we are able to provide a degree sequence with the addition of several specialty courses . CP majors take a *Programming for scientists* course (currently C and Fortran) and a *Hardware and software* course (an introduction to computer architecture) from the Information Technology department. From the physics department, they begin with PHY 318 *Methods of computational science*, covering basic algorithm development, analysis, and implementation, PHY 388 *Advanced computational physics,* a projects course team-taught by three faculty each presenting a specific advanced project, and PHY 390 *Computational research in physics,* a semester capstone computational research project. In addition, one senior elective is required and computation-based courses *Nonlinear dynamics* and *Molecular dynamics* are available to meet that need. CP majors therefore take five courses not required of regular physics majors, and we are frequently asked “what physics do they lose?” Our usual response is that they gain as much physics as they “lose”, but the following courses required for physics majors are not required for CP majors: second semester chemistry, senior level electromagnetism, senior advanced lab, and a second senior elective. While our experimental colleagues may decry the loss of the senior lab, in point of fact the analysis techniques for simulation data often coincide with those for laboratory data so that deficit is not as great as it may appear.

Example topics from the PHY 318 methods course include ODE solution of triatomic molecule dynamics using a predictor-corrector algorithm, Monte Carlo simulation of the Ising model, and discrete Fourier transform analysis of NMR data. All algorithms are presented both theoretically and via physics or physical chemistry example systems. The projects course PHY 388 involves three one-third semester projects, the topics selected by the participating faculty each semester. Examples in recent semesters include finite element analysis of thermal conduction, neural network predictors for both physics (the geomagnetic AE index) and financial time series, application of the split-operator method to time dependent wave functions, and a Monte Carlo simulation of photon scattering from a turbid medium. The capstone project course PHY 390 allows students wide latitude in selecting topics: all they need is a faculty member willing to advise them on the project.

These projects have ranged from specific investigations based on faculty research to completely independent projects developed by students. In the latter category is included a study of the fractal dimension of congressional district boundaries, implementation of a grid computer using MPI in order to perform a simulation of magnetospheric plasma dynamics, design and implementation of a cellular automaton to model flocking and schooling vortices observed in bird and insect populations, and a project on nonlinear optimization of a stock portfolio by a CP major with some background in finance courses.

The new degree program, initiated in 1999, rapidly grew to serve a similar number of majors as our other degree sequences, as shown in Figure 1.

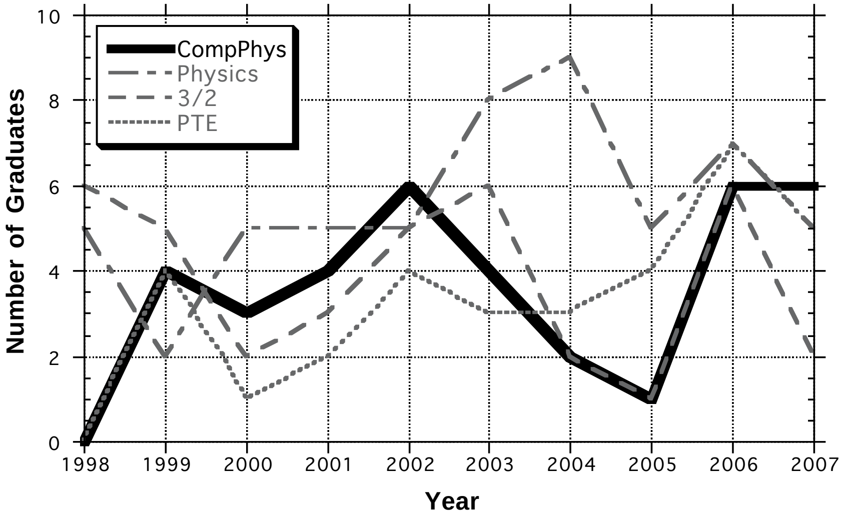


Figure 1. Numbers of graduates in four degree sequences indicating the rapid growth of the Computer Physics degree program.

Despite the statistics of small numbers, this nonetheless shows strong performance and indicates that we are satisfying a student need. However, our original thought that the Computer Physics (CP) degree might be a recruitment tool did not pan out. Figure 2 indicates that few freshmen enter as CP majors, yet a significant fraction graduate with that degree. Thus, incoming majors in other sequences (primarily the physics/engineering 3/2 program) are transferring into the CP program. In this sense, the CP program adds flexibility for our majors and essentially acts as a *retention* tool, helping the department maintain a healthy stream of graduates.

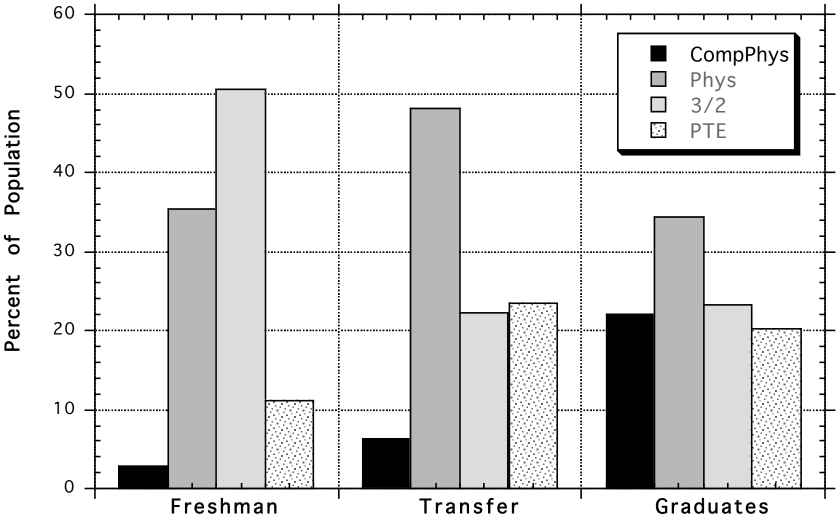


Figure 2. Distribution of declared physics major sequence for incoming freshmen, incoming transfer students, and outgoing graduates, 2000-2007.

In a five-year assessment of the CP sequence performed in 2005, surveyed students reported high satisfaction with the specialized CP courses in the physics department, with somewhat lower satisfaction with the two information technology classes particularly the computer architecture course, which students reported as not being very challenging (we continue to pursue other options with our colleagues in that department). When asked about the utility of the various courses to their current job or graduate student position, the PHY 390 capstone research course earned highest marks from CP alumni. Written comments suggesting that the capstone course's open structure, which essentially asks students to generate an interesting problem, design a computational model to solve it, and bring the project to some sort of closure, was useful for both graduate-school bound students and those moving directly into the job market -- perhaps an *a posteriori* justification for our programmatic guiding principles of putting the students “in command of the technology” and that computing should “enable the student to pursue independent study.”

About 35% of CP majors, in the first five years of the program, continued on to graduate school, with 50% in physics, the remaining 50% in other technical fields including electrical and computer engineering, materials engineering, and environmental science. This percentage is somewhat lower than the graduate school-bound students in the physics major sequence, which has been near 40% in recent years. For those students who took jobs directly after graduation, more than 35% went to computing-related positions, 40% to other engineering jobs, and the remainder to a mix of technical and nontechnical business positions. Of the CP graduates employed in the computer area, a handful indicated that they had started their own businesses that involved computing. Statistically, the CP graduates do not look significantly different than their physics degree counterparts except for a somewhat higher percentage going into non-physics graduate study and more of the directly employed working in computing-related positions.

## 4. Conclusions

After four decades of computational physics education development by a relatively small group of committed educators, there appears to be a sort of “critical mass” emerging. Whether the recent increase in collaboration, conferences, and publications will result in permanent changes in the education of future physicists remains to be seen, but given the rapid growth of computational physics in the research arena, and the growing number of young faculty with expertise in this area, perhaps this is one educational innovation that will, in fact, become an intrinsic part of future physics curricula.

At Illinois State we count our experiment with CPE as a success, both the integration of computational methods and assignments into physics courses and the separate degree sequence. Physics, engineering physics, and even physics teacher education alumni have reported that they are ahead of their graduate student and workplace peers in the computational area, allowing them to seize opportunities not as easily available to their colleagues, or simply to feel more secure in their knowledge. The CP degree appears to be an attractive alternative to the engineering physics 3/2 program and to the physics degree to those students with the interest and the knack for computational science, and the added flexibility acts as a retention tool for some majors not satisfied with those more traditional programs. Finally, our research rationale for CPE has been a success since the CP program meshes synergistically with our undergraduate research effort.

A department considering implementation of CPE in some form currently has a wealth of resources available to assist in the process. I have hinted in this article at some of the potential pitfalls and recommend learning the experiences of other departments beginning, for example, with the references herein cited. Our department's experience was relatively smooth partly due to the significant proportion of computational physicists on the faculty, but also because of our departmental culture of open discussion (and argument) of proposed changes -- a useful sieve for separating feasible goals from those that will not work.

*Acknowledgements*

The author would like to thank his colleagues in curriculum development at Illinois State University: George Skadron, Robert Young, Hiroshi Matsuoka, Rainer Grobe, Q. Charles Su, and Jean Standard for many years of fresh ideas and practical implementations.

## References

[1] See, e.g., ALFRED M. BORK, *A physics independent study course with computers*, Am. J. Physics **31** (1963) 364-368.

[2] CHARLES P. FRAHM AND ROBERT D. YOUNG, *PSI for low-enrollment junior-senior physics courses*, Am. J. Physics **44** (1976) 524-526.

[3] STEVEN KOONIN, *Computational physics*, Benjamin/Cummings, 1985.

[4] HARVEY GOULD AND JAN TOBOCHNIK, *An introduction to computer simulation methods*, Addison-Wesley, Reading MA, 1987.

[5] EDWARD F. REDISH AND JOHN S. RISLEY (EDS.), *The conference on computers in physics instruction: proceedings*, Addison-Wesley, Redwood City CA, 1990.

[6] RICHARD F. MARTIN JR., GEORGE SKADRON AND ROBERT D. YOUNG, *Computers, physics and the undergraduate experience*, Computers in Physics **5** (1991) 302-310.

[7] RICHARD F. MARTIN JR. AND SHANG-FEN REN, *Broadening the Physics Degree: A New Bachelor's Degree in Computational Physics at Illinois State University*, Forum on Education of the American Physical Society, Spring Newsletter (1988) 12.

[8] *Computation in physics courses*, special issue of Computing in Science and Engineering **8** (2006) 11-58; associated *web extras* are available at the CiSE website http://www.computer.org/portal/site/cise/.

[9] RUBIN LANDAU, *Computational physics: a better model for physics education?*, Computing in Science and Engineering **8** (2006) 22-30.

[10] *American Institute of Physics Statistical Research Center*, http://www.aip.org/statistics/

[11] NORMAN CHONACKY, *Has computing changed physics courses?*, Computing in Science and Engineering **8** (2006) 4-5.

[12] ROBERT G. FULLER, *Numerical computation in US undergraduate physics courses,* Computing in Science and Engineering **8** (2006) 16-21.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# Exponential fitted Runge-Kutta methods of collocation type based on Gauss, Radau and Lobatto traditional methods.

**J. Martín Vaquero**[1] **and J. Vigo-Aguiar**[1]

[1] *Departamento de Matematica Aplicada, Universidad de Salamanca, 37008, Salamanca, Spain*

emails: `jesmarva@usal.es`, `jvigo@usal.es`

**Abstract**

Several exponential fitting Runge-Kutta methods of collocation type are derived as a generalization of the Gauss, Radau and Lobatto traditional methods of two steps. The new methods are capable of the exact integration (with only round-off errors) of differential equations whose solutions are linear combinations of an exponential and ordinary polynomials. Theorems of the truncation error reveal the good behavior of the new methods for stiff problems. Numerical examples underscore the efficiency of the proposed codes, especially when they are integrating stiff problems.

*Key words: Runge-Kutta methods, collocation type, exponential fitting, stiff problems*
*MSC 2000: 34A45, 65L06.*

## 1 Introduction

The numerical integration of ordinary differential equations has been one of the principal concerns of numerical analysis. In the early 1950s, after the pioneering work of Curtiss and Hirschfelder [1], it was realized that there was an important class of ordinary differential equations which presented a severe challenge to numerical methods available at that time. These problems have become known as stiff systems. Stiff problems (and highly oscillatory problems) are very common problems in many fields of the applied sciences (see [2], for example): atmosphere, biology, combustion, control theory, dynamics of missile guidance, dispersed phases, electronic circuit theory, fluids, heat transfer, chemical kinetics, lasers, mechanics, molecular dynamics, nuclear, process industries, process vessels, reactor kinetics, ...

Although there has been much controversy about the mathematical definition (see [3]), and in fact, there is no good mathematical definition of the concept of stiffness, we can say that a problem

$$y'(x) = g(x, y(x)), \qquad y(x_0) = y_0 \,, \tag{1}$$

(where $y = [y^1, \dots, y^m]$, and $g = [g^1, \dots, g^m]$, $y_0 = [y_0^1, \dots, y_0^m]$, $x \in R$) is stiff if its Jacobian (in a neighborhood of the solution) has eigenvalues $\lambda_i$ that verify $\frac{\max|Re\lambda_i|}{\min|Re\lambda_i|} \gg 1$ (usually, it is considered that $\max Re\lambda_i < 0$). Stiff systems are considered difficult because explicit numerical methods designed for non-stiff problems are forced to use very small step sizes increasing in this way the computational work. Looking for better methods for solving these systems, Curtiss and Hirschfelder [1] discovered the Backward Differentiation Formulae (BDF). Since then, a great effort has been made in order to obtain new numerical integration methods with strong stability properties desirable for solving stiff systems. For a survey on stiffness of ODE's see [4] or [5].

A great number of schemes based on modifications of the classical BDF formulae have appeared. Among them, we may mention DIFSUB [6] or LSODE [7], VODE [8], which uses the so-called Fixed Leading Coefficient BDF methods, DASSL [9], which is also indicated for solving differential algebraic equations, MEBDF (see [10]), which considers two predicted values to compute a new corrected approximation to the solution using a modified multistep formula, A-BDF [11], which is a one-parameter family that is a generalization of the classical BDF codes, and exponential fitting BDF schemes (EF-BDF) as in [12], [13], [14] or [15].

Implicit Runge-Kutta methods are another kind of formulae very common with stiff problems. Radau [4], STRIDE [16] or [17], DIRK [16], SDIRK, Gauss, Lobatto, Rosenbrock, modified schemes [18], ..., have frequently been used with those kind of numerical problems.

In recent years, another kind of schemes has appeared with good results. Such methods are called exponential fitting and some examples could be [19] (in that paper exponential fitting methods are applied for the first time to stiff problems) or [20].

In this paper, we are going to derive exponential fitting Runge-Kutta methods of collocation type through the Gauss, Radau and Lobatto traditional integrators. This is, we will impose both kind of conditions: the exact integration of differential equations whose solutions are linear combinations of an exponential with parameter $A$ and ordinary polynomials and the order conditions imposed to the traditional Runge-Kutta methods.

The paper is organized as follows. In section 2, we construct several exponential fitted versions of the well-known classical collocation methods. In section 3, an analysis of the converge of these new methods is made. Finally, in section 4 we show, with different test numerical examples, the efficiency of the proposed codes, especially when they are integrating stiff problems.

## 2 Derivation of the methods

Let us consider, first, the scalar initial-value problem of the form

$$y'(x) = f(x, y(x)), \quad x \in [x_0, x_f], \quad y(x_0) = y_0, \tag{2}$$

and assume that the function $f : [x_0, x_f] \times R \to R$ satisfies all the necessary requirements for the existence of a unique solution.

For the description of EFRK methods we use the classical Butcher notation [22]

$$y_{n+1} = y_n + h \sum_{i=1}^{s} b_i f(x_n + c_i h, u_i), \tag{3}$$

$$u_i = y_n + h \sum_{j=1}^{s} a_{ij} f(x_n + c_j h, u_j),$$

with $i = 1, \ldots, s$ and the coefficients are displayed as a Butcher array:

$$\begin{array}{c|ccc} c_1 & a_{11} & \ldots & a_{1s} \\ c_2 & a_{21} & \ldots & a_{2s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \ldots & a_{ss} \\ \hline & b_1 & \ldots & b_s \end{array}$$

Then, we will choose $c_i$ the values of the classical Runge-Kutta methods, but, now, the new coefficients $a_{i,j}$, $b_i$ are those such that

$$y(x_n + c_i h) = y(x_n) + h \sum_{j=1}^{s} a_{ij} f(x_n + c_j h, y(x_n + c_j h)), \tag{4}$$

$$y(x_n + h) = y(x_n) + h \sum_{i=1}^{s} b_i f(x_n + c_i h, y(x_n + c_i h) \tag{5}$$

when $y(x)$ belongs to the space $< 1, x, \ldots, x^{s-1}, e^{\lambda x} >$.

**Case A:** Derivation of the new exponential fitting Gauss method of 2-stages.

The weights $c_i$ of the new Gauss method are the same as in the traditional method: $c_1 = \frac{1}{2} - \frac{\sqrt{3}}{6}$ and $c_2 = \frac{1}{2} + \frac{\sqrt{3}}{6}$. But, $a_{i,j}$, $b_i$ are those such that (4) and (5) when $y(x) = 1$, $y(x) = x$, $y(x) = e^{\lambda x}$.

In this way the new exponential fitting method can be written as

$$\begin{array}{c|cc} \frac{1}{2} - \frac{\sqrt{3}}{6} & -\frac{6-6e^{(-3+\sqrt{3})\widehat{\lambda}/6}+(-3+\sqrt{3})\widehat{\lambda}e^{\sqrt{3}\widehat{\lambda}/3}}{6\widehat{\lambda}(-1+e^{\sqrt{3}\widehat{\lambda}/3})} & c_1 - a_{11} \\ \frac{1}{2} + \frac{\sqrt{3}}{6} & \frac{e^{(-3+\sqrt{3})\widehat{\lambda}/6}(6+e^{(3+\sqrt{3})\widehat{\lambda}/6}(-6+(3+\sqrt{3})\widehat{\lambda}))}{6\widehat{\lambda}(-1+e^{\sqrt{3}\widehat{\lambda}/3})} & c_2 - a_{21} \\ \hline & \frac{e^{(-3+\sqrt{3})\widehat{\lambda}/6}(1-e^{\widehat{\lambda}}+e^{(3+\sqrt{3})\widehat{\lambda}/6}\widehat{\lambda})}{\widehat{\lambda}(-1+e^{\sqrt{3}\widehat{\lambda}/3})} & 1 - b_1 \end{array}$$

$\widehat{\lambda}$ being the parameter $\lambda h$ in the method.

**Case B:** Derivation of the new exponential fitting RadauIIA method of 2-stages.

The weights $c_i$ of the new RadauIIA method are, then, the same as in the traditional method: $c_1 = \frac{1}{3}$ and $c_2 = 1$. The $a_{i,j}$ are those such that (4) and when $y(x) = 1$, $y(x) = x$, $y(x) = e^{\lambda x}$. And, in this case, $b_1 = a_{21}$, $b_2 = a_{22}$

So, in this way the new exponential fitting method can be written as

$$\begin{array}{c|cc} \frac{1}{3} & \frac{3-3e^{-\widehat{\lambda}/3}-\widehat{\lambda}e^{2\widehat{\lambda}/3}}{3\widehat{\lambda}-3\widehat{\lambda}e^{2\widehat{\lambda}/3}} & \frac{1}{3} - a_{11} \\ 1 & \frac{e^{-\widehat{\lambda}/3}(1+e^{\widehat{\lambda}}(-1+\widehat{\lambda}))}{\widehat{\lambda}(-1+e^{2\widehat{\lambda}/3})} & 1 - a_{21} \\ \hline & a_{21} & a_{22} \end{array}$$

**Case C:** Derivation of the new exponential fitting LobattoIIIA method of 2-stages.

The weights $c_i$ of the new LobattoIIIA method are $c_1 = 0$ and $c_2 = 1$. The $a_{i,j}$ are those such that (4) and when $y(x) = 1$, $y(x) = x$, $y(x) = e^{\lambda x}$. And, as with RadauIIA, $b_1 = a_{21}$, $b_2 = a_{22}$.

So, in this way the new exponential fitting method can be written as

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1+e^{\widehat{\lambda}}(-1+\widehat{\lambda})}{\widehat{\lambda}(-1+e^{\widehat{\lambda}})} & 1 - a_{21} \\ \hline & a_{21} & a_{22} \end{array}$$

Vectorial examples are more interesting than scalar ones. With vectorial examples we only have to change $\lambda$ by a matrix $A$, 1 by the identity matrix and we need to consider $\frac{B}{C} = BC^{-1}$. In that case the eigenvalues of the parameter should have a negative real part since positive exponentials give inaccuracies.

# 3 Convergence of the exponential fitting Runge-Kutta methods

In this section we will study the consistency and stability properties of the new methods. Since they can be written as Runge-Kutta algorithms they are zero-stable and we only need to study consistency and absolute stability of these formulas.

## 3.1 Consistency of the exponential fitting BDF-Runge-Kutta methods

If we want to know the local truncation error of the methods in a classical way, we need to consider the Runge-Kutta algorithms and study the order conditions following the theory of elementary differentials (the Fréchet derivatives) and rooted trees (see [22] chapter five, for example).

The local truncation error of a Runge-Kutta method with constant coefficients is given by (formula (5.47) in [22])

$$LTE = \frac{h^{p+1}}{(p+1)!} \sum_{r(t)=p+1} \alpha(t)[1 - \gamma(t)\psi(t)]F(t) + O(h^{p+2}) \qquad (6)$$

being $\alpha(t)$ the number of essentially different ways of labelling the nodes of the tree $t$ with the integers $1, 2, \dots, r(t)$. An easy way of computing $\alpha(t)$ is

$$\alpha(t) = \frac{r(t)!}{\sigma(t)\gamma(t)},$$

where the order $r(t)$, symmetry $\sigma(t)$ and density $\gamma(t)$ of a tree $t$ are defined as in [22], p. 164.

The function $F$ is defined on the set $T$ of all trees as in (5.39) in [22] (see Table 5.2 and the definition of the Mth. Fréchet derivative, p. 158, too) and the relations between $y^{(q)}$ and all elementary differentials of order $q$ is the following theorem (see Butcher [23]):

**Theorem 3.1** *Let $y' = f(y)$, $f$: $R^m \to R^m$. Then*

$$y^{(q)} = \sum_{r(t)=q} \alpha(t) F(t).$$

Finally $\psi(t)$ depends on the elements of the Butcher array as in [22], p. 167: for $i = 1, 2, \dots, s, s+1$ define on the set $T$ of all trees the functions $\psi_i$ by

$$\psi_i(\tau) = \sum_{j=1}^{s} a_{ij}$$

$$\psi_i([t_1 t_2 \dots t_M]) = \sum_{j=1}^{s} a_{ij} \psi_j(t_1) \psi_j(t_2) \dots \psi_j(t_M),$$

then, $\psi(t) := \psi_{s+1}(t)$.

Now, we can study the local truncation error of the exponential fitting Runge-Kutta methods in a similar way to [24].

**Theorem 3.2** *The leading term of the local truncation error of the new exponential fitting Gauss-2s is*

$$\frac{h^5(C_{H24} + C_{H5})}{4320},$$

*where*

$$C_{H24} = -\lambda^3 y'' + 5\lambda^2 f_y^2 y' + 10\lambda(f_{yy} f f_y - f_y^3) y',$$

$$C_{H5} = (f_{yyyy} f^3 + 2 f_{yyy} f_y f^2 - 6 f_{yy}^2 f^2 + 4 f_{yy} f_y^2 f + 6 f_y^4) y'.$$

**Proof 3.3** *i) The only one condition of a Runge-Kutta method to be consistent (at least) is $\sum_{i=1}^{s} b_i = 1 + O(\widehat{\lambda})$ (again $\widehat{\lambda}$ is the parameter $\lambda h$ in the method). In this case $\sum_{i=1}^{s} b_i = 1$.*

*ii) One method with order greater than one has to verify $2\sum_{i=1}^{s} b_i c_i = 1$, in this case*

$$2\sum_{i=1}^{2} b_i c_i = -\frac{2\sqrt{3}e^{(-3+\sqrt{3})\widehat{\lambda}/6} - 2\sqrt{3}e^{(3+\sqrt{3})\widehat{\lambda}/6} + (3+\sqrt{3})\widehat{\lambda} + (-3+\sqrt{3})e^{\sqrt{3}\widehat{\lambda}/3}}{3\widehat{\lambda}(-1+e^{\sqrt{3}\widehat{\lambda}/3})},$$

*whose Taylor series are* $1 + \frac{\widehat{\lambda}^3}{2160} + O(h^5)$. *Then, one part of the local truncation error is* $\frac{h^2}{2} y^{(2)}(x_n)(-\frac{\widehat{\lambda}^3}{2160} + O(h^5))$.

*iii) The two conditions of a third-order method are*

$$3 \sum_{i=1}^{s} b_i c_i^2 = 1 + O(h)$$

*and*

$$6 \sum_{i=1,j=1}^{s} b_i a_{ij} c_j = 1 + O(h).$$

*In this case*

$$3 \sum_{i=1}^{s} b_i c_i^2 = 1 + \frac{\widehat{\lambda}^3}{1440} + O(h^5)$$

*and*

$$6 \sum_{i=1,j=1}^{s} b_i a_{ij} c_j = 1 - \frac{\widehat{\lambda}^2}{144} + \frac{\widehat{\lambda}^3}{720} + O(h^4).$$

*Then,*

$$\frac{h^3}{3!} \sum_{r(t)=3} \alpha(t)[1 - \gamma(t)\psi(t)]F(t) = \frac{h^3}{3!} \frac{\widehat{\lambda}^2}{144} f_y^2 f + O(h^6),$$

*we are considering the scalar problem and the notation as in [22].*

*iv) We study the four conditions to be a fourth-order method and we got that*

$$\frac{h^4}{4!} \sum_{r(t)=4} \alpha(t)[1 - \gamma(t)\psi(t)]F(t) = \frac{h^4}{4!} \left( 3 \frac{\widehat{\lambda}}{54} f_{yy} f f_y f - \frac{\widehat{\lambda}}{18} f_y^3 f \right) + O(h^6).$$

*v) Finally, when we studied the conditions to be a fifth-order method and we got that they are not satisfied*

$$\frac{h^5}{5!} \sum_{r(t)=5} \alpha(t)[1 - \gamma(t)\psi(t)]F(t) = \frac{h^5}{5!} C_5 + O(h^6)$$

*where*

$$C_5 = \frac{1}{36}(f_{yyyy} f^4 + 6 f_{yyy} f_y f^3) - \frac{1}{24}(4 f_{yy}^2 f^3 + 4 f_{yy} f_y^2 f^2) - \\ -\frac{1}{9}(f_{yyy} f_y f^3 + 3 f_{yy} f_y^2 f^2) + \frac{1}{6}(f_{yy} f_y^2 f^2 + f_y^4 f).$$

*If we add the leading terms of local truncation error that we got in i) to v) and simplify we get total expression of the local truncation error.*

**Theorem 3.4** *The leading term of the local truncation error of the new exponential fitting Radau-2s is*

$$\frac{h^4(\lambda^2 y'' + (-4\lambda f_y^2 - f_{yyy} f^2 + f_{yy} f f_y + 3(f_y f_{yy} f + f_y^3)) y')}{216}.$$

**Proof 3.5** *i) The condition to be a consistent method is satisfied since* $\sum_{i=1}^{s} b_i = 1$.

*ii) The method has at least first-order because* $2\sum_{i=1}^{s} b_i c_i = 1 - \frac{\widehat{\lambda}^2}{108} + O(h^3)$. *Then, one part of the local truncation error is* $\frac{h^2}{2} y^{(2)}(x_n)(\frac{\widehat{\lambda}^2}{108} + O(h^3))$.

*iii) The method has order bigger than two because*

$$\frac{h^3}{3!} \sum_{r(t)=3} \alpha(t)[1-\gamma(t)\psi(t)]F(t) = \frac{h^3}{3!} \frac{-\widehat{\lambda}}{9} f_y^2 f + O(h^5).$$

*iv) We studied the four conditions to be a forth-order method but they are not satisfied*

$$\frac{h^4}{4!} \sum_{r(t)=4} \alpha(t)[1-\gamma(t)\psi(t)]F(t) = \frac{h^4}{4!} \left( \frac{-1}{9}(f_{yyy}f^3 + f_{yy}ff_yf) + \frac{1}{3}(f_y f_{yy} f^2 + f_y^3 f) \right) + O(h^5).$$

**Theorem 3.6** *The leading term of the local truncation error of the new exponential fitting Lobatto-2s is*

$$\frac{h^3(\lambda y'' - y''')}{12}.$$

**Proof 3.7** *i) The condition to be a consistent method is satisfied since* $\sum_{i=1}^{s} b_i = 1$.

*ii) While in the condition of a first-order method, we got* $2\sum_{i=1}^{s} b_i c_i = 1 - \frac{\widehat{\lambda}}{6} + O(h^3)$. *Then, one part of the local truncation error is* $\frac{h^2}{2} y^{(2)}(x_n)(\frac{\widehat{\lambda}}{6} + O(h^3))$.

*iii) And we get that the method has order two because*

$$\frac{h^3}{3!} \sum_{r(t)=3} \alpha(t)[1-\gamma(t)\psi(t)]F(t) = \frac{h^3}{3!} \frac{-1}{2}(f_{yy}f^2 + f_y^2 f) + O(h^5).$$

## 3.2 Absolute stability of the exponential fitting BDF-Runge-Kutta methods

The classical definitions of absolute stability regions and A-stability were stated for linear multistep methods with constant coefficients. The stability properties of the proposed methods are analyzed to demonstrate their relevance especially in the integration of stiff oscillatory problems. In this section the definitions are extended to exponential fitting methods. The way is very similar to that used in [25] to extend those definitions.

In [25] Coleman and Ixaru studied the stability properties of existing exponential fitting methods that integrate exactly the problem

$$y''(x) = g(x, y(x)), \tag{7}$$

when $y(x) = exp(\pm ikx)$, but when they want to study their stability properties, they apply the method to the test equation

$$y''(x) = -w^2 y(x), \tag{8}$$

and then, they plot their regions of stability on the $\mu - \theta$ plane (being $\mu = wh$ and $\theta = kh$).

In our case, we shall consider, first the test problem

$$y'(x) = Ay(x), \tag{9}$$

while we introduce in the method the estimated parameter $A^*$, so the weights are $b_i(A^*h)$ and $a_{ij}(A^*h)$.

We are going to this analysis considering that there exist a nonsingular matrix $Q$ such that

$$Q^{-1}AQ = \Lambda = diag[\lambda_1, ..., \lambda_m],$$

and

$$Q^{-1}A^*Q = \Lambda^* = diag[\lambda_1^*, ..., \lambda_m^*].$$

If $Q^{-1}A^*hQ = \Lambda^*h$, $\Lambda^* = diag[\lambda_1^*, ..., \lambda_m^*]$, as $b_i(A^*h)$ and $a_{ij}(A^*h)$ depend only on $e^{A^*h}$, $Id$ and $A^*h$, then we have that $Q^{-1}b_i(A^*h)Q = b_i(\Lambda^*h)$ and $Q^{-1}a_{ij}(A^*h)Q = a_{ij}(\Lambda^*h)$ and the system can be coupled.

So we can reduce to consider as a test problem the very famous Dahlquist's equation

$$y'(x) = \mu y(x), \quad y_0 = 1, \quad z = h\mu, \tag{10}$$

where $Re(\mu) < 0$, with $\mu = \lambda + \nu$. That is, we have introduced the value $\lambda$ in the method while the true solution depends on the exponential of $\mu$. And we are going to calculate the set

$$S = \{z \in C; \mid R(z) \mid \leq 1\},$$

called the stability domain of the method, where $R(z)$, the stability function of the method, is that proposed by Hairer and Wanner in [4]:

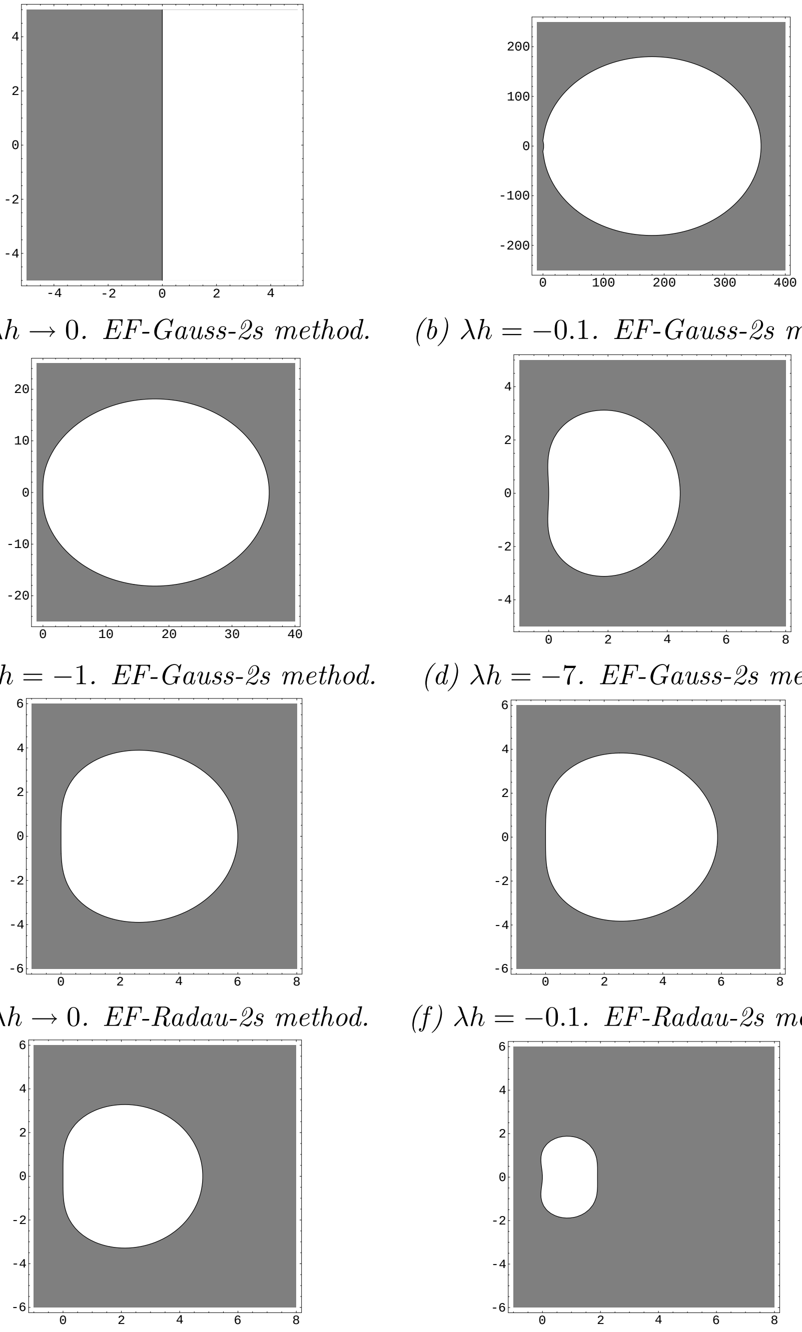
$$R(z) = 1 + zb^t(Id - zA)^{-1}\mathbf{1},$$

being $b^t = (b_1, \ldots, b_s)$, $A = (a_{ij})_{i,j=1}^s$ and $\mathbf{1} = (1, \ldots, 1)^t$, or (see [4], [26] or [27]), they are both the same (see [4], Proposition 3.2, p. 41)

$$R(z) = \frac{det(Id - zA + z\mathbf{1}b^t)}{det(Id - zA)}.$$

Again, as we mentioned in [28], [15], [29] or [30], it is impossible to plot regions of absolute stability. However we can fix $u = h\lambda$ with different real and complex values and plot in the complex plane the values of $\nu h$ that makes the method absolute stable.

We can begin showing some regions of absolute stability of the Gauss-2s and Radau-2s method when $\lambda h \in R^-$. Since the methods approach to the classical methods when $\lambda h \to 0$, then both methods are A-stable in this case and when $\lambda h \to -\infty$, the regions of absolute instability are smaller and smaller, this means that the error (when we calculate the parameter on the method) can be bigger and the method continues being stable for the problem. We have shown some of these regions in Figure 1.



*(a)* $\lambda h \to 0$. *EF-Gauss-2s method.* *(b)* $\lambda h = -0.1$. *EF-Gauss-2s method.*

*(c)* $\lambda h = -1$. *EF-Gauss-2s method.* *(d)* $\lambda h = -7$. *EF-Gauss-2s method.*

*(e)* $\lambda h \to 0$. *EF-Radau-2s method.* *(f)* $\lambda h = -0.1$. *EF-Radau-2s method.*

*(g)* $\lambda h = -1$. *EF-Radau-2s method.* *(h)* $\lambda h = -5$. *EF-Radau-2s method.*

Figure 1: Absolute stability regions (in grey) of the exponential fitting Gauss-2s and Radau-2s methods. The parameters in the method are real. Horizontal and vertical axes represent $Re(\mu h)$ and $Im(\mu h)$.

The behavior of the regions of these methods when $\lambda h \in C^-$ is very similar as we can see in Figure 2, where we have shown some stability regions of the Radau-2s and Lobatto methods. Again, when $ah \to -\infty$ ($a$ being the real part of $\lambda$), the regions of absolute instability are smaller and smaller. We can check in this figure that if we choose $\lambda_1 h = a + ib \in \mathbb{C}^-$ $a, b \in R$ and $\lambda_2 h = a - ib$, then the regions of absolute stability were symmetric.

# 4 Numerical examples

In other papers (see [31], [32], [33], [34], [35] or [30], for example), the chosen parameter has been a matrix, so two big open questions appeared in this field: which is the best way to calculate the exponential matrix and which is the best procedure to choose the parameter for the methods.

In this case, we have used scalar parameters $\lambda h \in R^-$ in the method.

We are going to suppose that the IVP is

$$y'(x) = g(x, y(x)), \qquad y(x_0) = y_0\,, \tag{11}$$

(where $y = [y^1, \ldots, y^m]$, and $g = [g^1, \ldots, g^m]$, $y_0 = [y_0^1, \ldots, y_0^m]$, $x \in R$). The steps to calculate the parameter $\lambda h \in R^-$ are the following:

1) In the first three steps we take the coefficients of the classical methods.

2) We are going to suppose, now, that we want to calculate $y_{n+1}$ and we have calculated $y_n$, $y_{n-1}$, $y_{n-2}$ and $y_{n-3}$.

Since $\frac{y_n^i-3y_{n-1}^i+3y_{n-2}^i-y_{n-3}^i}{h^3} \approx (y_n^i)'''$ and $\frac{2y_n^i-5y_{n-1}^i+4y_{n-2}^i-y_{n-3}^i}{h^3} \approx (y_n^i)''$, then $\lambda_i h = \frac{y_n^i-3y_{n-1}^i+3y_{n-2}^i-y_{n-3}^i}{2y_n^i-5y_{n-1}^i+4y_{n-2}^i-y_{n-3}^i} \approx \frac{(y_n^i)'''}{(y_n^i)'''}$. Then, we choose $\lambda h = \max \lambda_i h$.

3) Since positive parameters or very negative values could give inaccuracies, if $\lambda h \geq 0$, then we have taken the weights $a_{ij}$, $b_i$ and $c_i$ of the classical methods. Finally, if $\lambda \leq -100$, then we have taken $\lambda = -100$.

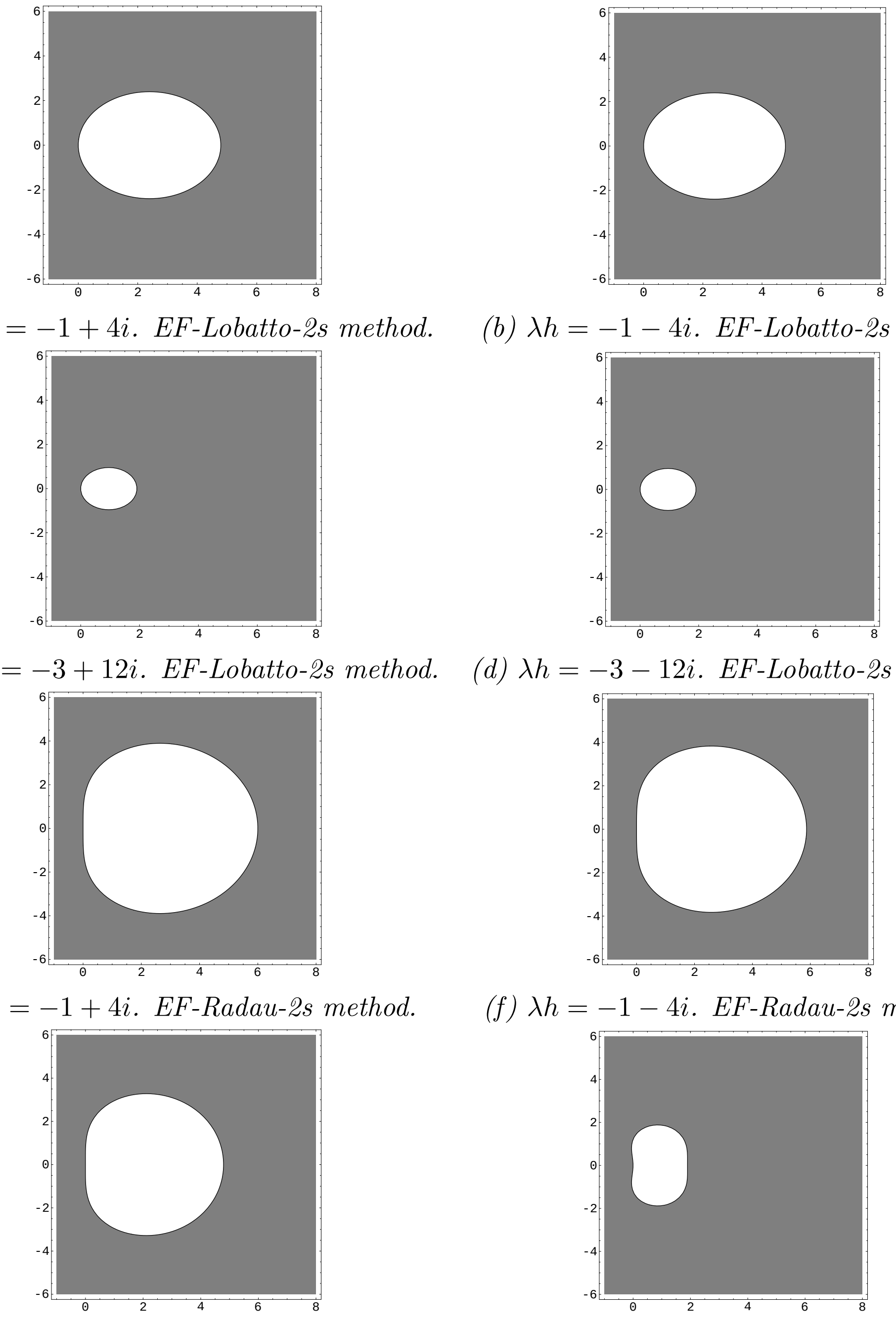
**Problem 1**, the first stiff problem is known as Robertson equation (see, for example, [4]),

$$\begin{array}{l} y_1'(x) = -0.04y_1(x) + 10^4 y_2(x) y_3(x), \\ y_2'(x) = 0.04y_1(x) - 10^4 y_2(x) y_3(x) - 310^7 y_2^2(x), \\ y_3'(x) = 310^7 y_2^2(x), \\ y_1(0) = 1, \quad y_2(0) = 0, \quad y_3(0) = 0, \quad 0 \leq x \leq 40. \end{array} \tag{12}$$

We have compared the numerical results of the traditional Radau-2s (of two steps) with constant step length and the EF-Radau-2s in Table 1.

The methods are in Mathematica and we used an Intel Pentium 4 with 1.40 GHz. We can observe that the error is smaller with the new algorithm, but te CPU Time (in seconds) is smaller, too. The reason is that the new scheme needed less iterations of the Newton's method to solve the nonlinear equation.

**Problem 2**, the second stiff problem is known as Oregonator (see, for example,



*(a)* $\lambda h = -1 + 4i$. *EF-Lobatto-2s method.*

*(b)* $\lambda h = -1 - 4i$. *EF-Lobatto-2s method.*

*(c)* $\lambda h = -3 + 12i$. *EF-Lobatto-2s method.*

*(d)* $\lambda h = -3 - 12i$. *EF-Lobatto-2s method.*

*(e)* $\lambda = -1 + 4i$. *EF-Radau-2s method.*

*(f)* $\lambda h = -1 - 4i$. *EF-Radau-2s method.*

*(g)* $\lambda h = -3 + 12i$. *EF-Radau-2s method.*

*(h)* $\lambda h = -3 - 12i$. *EF-Radau-2s method.*

Figure 2: Absolute stability regions (in grey) of the exponential fitting Radau-2s and Lobatto-2s methods. The parameters in the method are complex. Horizontal and vertical axes represent $Re(\mu h)$ and $Im(\mu h)$.

| Step length | Method | Error | CPU Time (sec) |
|---|---|---|---|
| $h = 0.1$ | Radau-2s | $2.745 \times 10^{-9}$ | 11.677 |
| | EF-Radau-2s | $7.8288 \times 10^{-10}$ | 8.972 |
| $h = 0.05$ | Radau-2s | $3.6865 \times 10^{-10}$ | 23.574 |
| | EF-Radau-2s | $1.8237 \times 10^{-10}$ | 18.226 |
| $h = 0.025$ | Radau-2s | $9.778 \times 10^{-11}$ | 46.147 |
| | EF-Radau-2s | $2.1101 \times 10^{-11}$ | 35.911 |

Table 1: Numerical errors in the integration of problem 1.

[4]),

$$\begin{aligned}
&y_1'(x) = 77.27(y_1(x)(1 - 8.375 \times 10^{-6} y_1(x) - y_2(x)) + y_2(x)),\\
&y_2'(x) = \tfrac{1}{77.27}\left(-(1 + y_1(x))y_2(x) + y_3(x)\right),\\
&y_3'(x) = 0.161(y_1(x) - y_3(x)),\\
&y_1(0) = 1, \quad y_2(0) = 2, \quad y_3(0) = 3, \quad 0 \le x \le 30.
\end{aligned} \tag{13}$$

We have compared the numerical results of the traditional Radau-2s with constant step length and the EF-Radau-2s in Table 2.

| Step length | Method | Error | CPU T | Method | Error | CPU T |
|---|---|---|---|---|---|---|
| $h = 0.01$ | Radau-2s | 0.46543 | 238.624 | EF-Radau-2s | 0.46509 | 195.701 |
| $h = 0.005$ | Radau-2s | 0.054777 | 437.018 | EF-Radau-2s | 0.054663 | 364.755 |
| $h = 0.0025$ | Radau-2s | 0.006649 | 796.856 | EF-Radau-2s | 0.006618 | 667.239 |

Table 2: Numerical errors in the integration of problem 2.

**Problem 3,** we integrate the nonlinear IVP proposed by Frank and van der Houwen (from CWI) [36] or Kaps [37]

$$\left\{ \begin{aligned}
&y_1'(x) = -1002 y_1(x) + 1000 y_2^2(x),\\
&y_2'(x) = y_1(x) - y_2(x)(1 + y_2(x)),\\
&y_1(0) = 1, \qquad y_2(0) = 1,
\end{aligned} \right. \tag{14}$$

with solution

$$y_1(x) = e^{-2x}, \qquad y_2(x) = e^{-x},$$

the Jacobian of the right-hand side of this problem at the initial point has the eigenvalues $\lambda_1 = -1004$, $\lambda_2 = -1.00199$, then we can consider that this problem is stiff.

In table 3 we have compared the results obtained at point $x = 5$ with Gauss-2s and EF-Gauss-2s, using different step lengths.

# References

[1] C.F. Curtiss and J.O. Hirschfelder, Integration of stiff equations, Proc. Nat. Acd. Sci. 38 (1952), pp. 235-243.

[2] R.C. Aitken, Stiff computation, Oxford University Press (1985), New York.

| Step length | Method | Error | CPU Time (sec) |
|---|---|---|---|
| $h = 0.1$ | Gauss-2s | $5.1207 \times 10^{-6}$ | 0.691 |
| | EF-Gauss-2s | $8.3555 \times 10^{-7}$ | 0.58 |
| $h = 0.01$ | Gauss-2s | $6.4579 \times 10^{-12}$ | 6.92 |
| | EF-Gauss-2s | $4.2408 \times 10^{-12}$ | 5.648 |

Table 3: Error in the numerical integration of problem 3.

[3] M.N. Spijker, Stiffness in the numerical initial-value problems, J. Comput. Appl. Math. 72 (1996), pp. 393-406.

[4] E. Hairer and G. Wanner, Solving Ordinary Differential Equations II, Springer, Berlin (1993).

[5] D. J. Higham, Stiffness of ODEs, BIT 33 (1993), pp. 285-303.

[6] C.W. Gear, Algorithm 407-DIFSUB for solution of ordinary differential equations, Comm. ACM 14 (1971), pp. 185-190.

[7] A.C. Hindmarsh, LSODE and LSODI, two new initial value ordinary differential equation solvers, ACM-Signum Newsletter 15 (1980), pp. 10-11.

[8] P.N. Brown, G.D. Byrne and A.C. Hidmarsh, VODE: A variable-coefficient ODE solver, SIAM J. Sci. Statist. Comput. 10 (1989), pp. 1038-1051.

[9] L.R. Petzold, A description of DASSL: a differential-algebraic system solver, IMACS Trans. Scientific Computing, eds. R.S. Stepleman et al., North-Holland, Amsterdam (1993), pp. 65-68.

[10] J.R. Cash, The integration of stiff initial value problems in ODE's using modified extended backward differentiation formulae, Comput. Math. Appl. 9 (1983), pp. 645-657.

[11] C. Fredebul, A-BDF: A generalization of the backward differentiation formulae, SIAM J. Numer. Anal. 35 (1998), pp. 1917-1938.

[12] L.Gr. Ixaru, M. Rizea, G. Vanden Berghe and H. De Meyer, Weights of the exponential fitting multistep algorithms for ODEs, J. of Comput. and Appl. Math. 132 (2001), pp. 83-93.

[13] L.Gr. Ixaru, G. Vanden Berghe and H. De Meyer, Frequency evaluation in exponential fitting multistep algorithms for ODEs, J. of Comput. and Appl. Math. 140 (2002), pp. 423-434.

[14] L.Gr. Ixaru, G. Vanden Berghe and H. De Meyer, Exponentially fitted variable two-step BDF algorithms for first order ODEs, Comput. Phys. Comm. 100 (2003), pp. 56-70.

[15] J. Martin-Vaquero and J. Vigo-Aguiar, Adapted BDF Algorithms: higher-order methods and their stability, J. Scientific Comput. (accepted).

[16] R. Alexander, Diagonally implicit Runge-Kutta methods for stiff ODEs, SIAM J. Numer. Anal. 14 (1977), pp. 1006-1021

[17] S.P. Nørsett, Semi explicit Runge-Kutta methods, Mathematics and Computing Rpt. N. 6/74 (1974), University of Trodheim.

[18] T.E. Simos and J. Vigo-Aguiar, A modified Runge-Kutta method with phase-lag of order infinity for the numerical solution of the Schrodinger equation and related problems, Computers and Chemistry 25 (2001), pp. 275-281.

[19] W. Liniger and R. Willoughby, Efficient integration for stiff systems of ordinary differential equations, SIAM J. Num. Anal. 7 (1970), pp. 47-66.

[20] H. Van de Vyver, Frequency evaluation for exponentially fitted Runge-Kutta methods, J. Comput. Appl. Math. 184 (2005), pp. 442-463.

[21] G. Vanden Berghe, M. Van Daale and H. Vande Vyver, Exponential fitted Runge-Kutta methods of collocation type: fixed or variable knot points, J. of Comput. and Appl. Math. 159 (2003), pp. 217-239.

[22] J.D. Lambert, Numerical Methods for Ordinary Differential Systems. The initial Value Problem, Wiley (1991), Chichester.

[23] J.C. Butcher, The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods, Wiley (1987), Chichester.

[24] J. Vigo-Aguiar, J. Martín-Vaquero and H. Ramos, Exponential Fitting BDF-Runge-Kutta Algorithms, Computer Physics Communications (submitted).

[25] J.P. Coleman and L.Gr. Ixaru, P-stability and exponential-fitting methods for y"=f(x,y), IMA J. Numer. Anal. 16 (1996), pp. 179-199.

[26] R. Scherer, A necessary condition for B-stability, BIT 19 (1979), pp. 111-115.

[27] H.J. Stetter, Analysis of discretization methods for ordinary differential equations, Springer (1973), Berlin.

[28] J. Martín-Vaquero and J. Vigo-Aguiar, Exponential fitting BDF algorithms: explicit and implicit 0-stable methods, J. Comp. Appl. Math. 192 (2006), pp. 100-113.

[29] J. Vigo-Aguiar, J. Martín-Vaquero, J. and R. Criado, On the stability of exponential Fitting BDF Algorithms, Journal of Computational and Applied Mathematics 175 (2005), pp. 183-194.

[30] J. Vigo-Aguiar and J. Martín-Vaquero, Exponential Fitting BDF Algorithms and their properties, Applied Mathematics and Computation (accepted).

[31] W.S. Edwards, L.S. Tuckerman, R.A. Friesner and D.C. Sorensen, Krylov methods for the incompressible Navier-Stokes equations, J. Comput. Physics 110 (1994), pp. 82-102.

[32] R.A. Friesner, L.S. Tuckerman, B.C. Dornblaser and T.C. Russo, A method for exponential propagation of large systems of stiff nonlinear differential equations, J. Sci. Comput. 4 (1989), pp. 327-354.

[33] E. Gallopoulos and Y. Saad, Efficient solution of parabolic equations by Krylov approximation methods, SIAM J. Sci. Stat. Comput. 13 (1992), pp. 1236-1264.

[34] R. Kosloff, Propagation methods for quantum molecular dynamics, Annu. Rev. Phys. Chm 45 (1994), pp. 145-178.

[35] R.B. Sidje, Expokit: software package for computing matrix exponentials, ACM Trans. Math. Software 24 (1998), pp. 130-156.

[36] J.E. Frank and P.J. van der Houwen, Parallel Iteration of the Extended Backward Differentiation Formulas, Report MAS-R9913 (1999), CWI.

[37] P. Kaps, Rosenbrock-type methods, in: G. Dahlquist and R. Jeltsch, editors, Numerical methods for stiff initial value problems, Bericht nr. 9, Inst für Geometrie und Praktische Mathematik der RWTH Aachen (1981).

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# Functional Support Vector Machines and Generalized Linear Models for Glacier Geomorphology Analysis

**J. M. Matías[1], C. Ordóñez[2], J. Taboada[2] and T. Rivas[2]**

[1] *Department of Statistics, University of Vigo*

[2] *Department of Natural Resources, University of Vigo*

emails: `jmmatias@uvigo.es`, `cgalan@uvigo.es`, `jtaboada@uvigo.es`, `trivas@uvigo.es`

### Abstract

We propose a functional pattern recognition approach to the problem of identifying the topographic profiles of glacial and fluvial valleys, using a functional version of support vector machines for classification. We compare a proposed functional version of support vector machines with functional generalized linear models and their vectorial versions: generalized linear models and support vector machines that use the original observations as input. The results indicate the benefit of our proposed functional support vector machines and, in more general terms, the advantages of using a functional rather than a vectorial approach.

*Key words: Digital elevation models, functional data analysis, functional general lineal model, support vector machines, topographic profiles*
*MSC 2000: 62J12, 62J99, 68T05, 68T35, 86A40*

## 1 Introduction

Geomorphology is the science that studies landforms in the Earth's crust and the processes that form them. Valleys, which are principally shaped by the erosional activity of glaciers and rivers, are classified in two main classes in terms of their shape [1]–U-shaped valleys and V-shaped valleys. U-shaped valleys, which have typically been formed by glaciers, have steeply sloped sides and a central concave part. V-shaped valleys, which are formed by rivers, usually have gentler slopes and narrower, more angular bases. Another type of fluvial river valley also exists, which is characterized by a flat rather than angular base.

Geomorphology experts have traditionally distinguished between different types of valleys on the basis of visual analyses of the topographical profiles of the valleys, contour maps, and aerial photographs. Another way to study and compare valley shapes is to simulate the topographic profiles using mathematical models, e.g., catenary functions [2]; [12], power law regression models [11]; [3] or generalized power law [8]. However, since these models are overly simple and fail to reflect the precise shapes of the real profiles, this leads to error in reproducing the landforms and in differentiating between the different types of valleys. For this reason, and also given the recent advent of models capable of simulating valley erosion and evolution, more advanced techniques with a statistical focus are being used to describe and compare topographic profiles, as follows:

1. Pre-defined non-linear regression models with a parametric focus, from which the models with the best goodness-of-fit are chosen [5].

2. Functional data analysis [9], which applies two distinct strategies [4], namely, an unsupervised approach that clusters profiles and interprets the clusters in terms of the majority elements (e.g. the U shape), and a supervised approach that constructs predictive models of the profiles based on scalar covariables (elevation, etc.).

We propose a new machine-learning philosophy approach based on functional techniques, which involved the construction of an expert system, trained using a supervised approach, with artificially constructed theoretical profiles that enables the system to determine the nature of new real profiles. Since real profiles rarely fall into one morphological group or another, this system also indicates the degree of belongingness of a profile to a particular group (i.e., the a posteriori probability of membership in a particular group).

To implement this approach, we used functional pattern recognition models, specifically functional generalized linear models [6] (functional 0-1 regression and logistic regression, [7]), as also a functional version of the support vector machines [10] in the functional space generated by a set of basic functions.

Although in this initial research phase, we use just two morphological groups, the method can be easily extended to include more groups using multiclass functional pattern recognition techniques (to be the subject of a future article).

Our article is structured as follows:

1. Firstly we describe functional pattern recognition techniques, including the functional generalized linear model, particular cases of this model used in this work, and the proposed functional pattern recognition support vector machines.

2. Next we describe the details of the application problem, as also the results of the application of the above techniques to the data-smoothed functional versions. These results are compared with the results obtained from applying the vectorial versions of the same techniques to the original scalar data.

3. The final section describes our conclusions.

# 2 Application to the Identification of Valley Profiles

## 2.1 Training and Test Samples

In order to evaluate the capacity of the functional support vector machines to discriminate between the different types of valley profiles, simulated profiles and real profiles for glacial valleys and fluvial valleys were used. In an initial phase, simulated profiles were generated in order to train the expert system, whereas the real profiles were reserved to test the behavior of the system. This system can later be enriched by incorporating real profiles whose morphology is known in the training sample.

The type of profiles considered in this initial research phase were U-shaped and V-shaped, although the method can incorporate other types of profiles (to be studied in future research).

The training sample should include the full range of profiles in similar conditions as in reality, in other words, with observations subject to noise and in different positions. The profiles generated were contaminated with Gaussian noise with mean zero for both the U-shaped and V-shaped valleys, the slope of one of the two branches of the profiles were modified from the origin—with a view to reflecting the asymmetry observed in reality—by assigning a lower value to one than the other.

In order to simulate the U-shaped profiles, catenary functions were used, expressed as follows [12]:

$$y = b_0 + b_1 \cosh\left(x/b_1\right) + \varepsilon \tag{1}$$

where $y$ is the vertical distance and $x$ the horizontal distance from $(0,0)$ in the coordinate system, where $b_0$ and $b_1$ are coefficients, and where $\epsilon \sim N\left(0,\sigma^2\right)$ is an additive term of error used to simulate deviations from the real profiles with respect to the catenary function. The coefficient $b_1$ determines how quickly the catenary opens up.

Figure 1 shows some of these catenaries, all with the same extremes but with different depths.

In order to represent the V-shaped profiles, the general power model (GPL) [5] was used, expressed as follows:

$$y = \alpha \left|x - x_0\right|^{\beta} + y_0 + \varepsilon \tag{2}$$

The parameter $\beta$ provides a direct measure of the profile curvature, with values near 1 corresponding to V-shaped profiles and values near 2 corresponding to U-shaped profiles.

The term $(x_0, y_0)$ provides an estimate of the location of the minimum of the profile.

Figure 2 shows various V-shaped profiles for the particular case of a minimum located at the origin of the coordinates. The additive term for noise, as in the catenaries, is a normal distribution with a zero mean.

The real profiles used to evaluate the goodness-of-fit of the system were extracted from a digital elevation model (DEM) with a resolution 30 meters representing Sierra Nevada, a mountain range located in southern Spain with some of Spain's most important skiing slopes. The graph on the left of Figure 3 depicts the DEM, discretized in altitude intervals of 100 meters, for altitudes varying between 410 meters and 1999

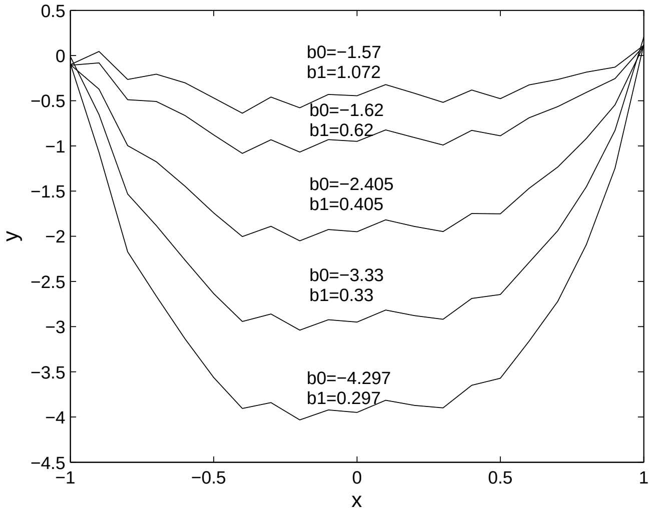


Figure 1: Sample of catenaries used to simulate the U-shaped profiles.

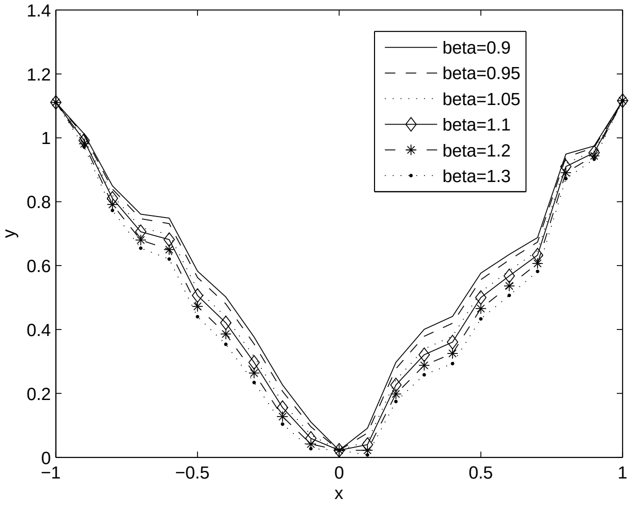


Figure 2: Profiles simulated to train for V-shaped valleys.

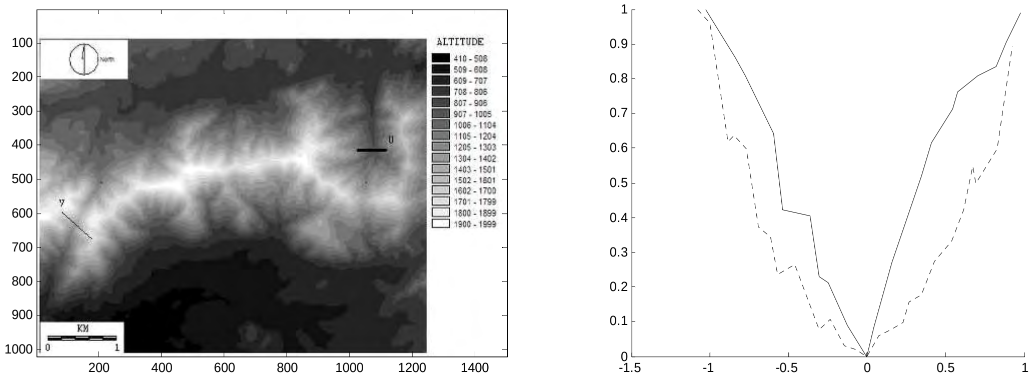


Figure 3: DEM of Sierra Nevada, showing two profiles, one U-shaped and the other V-shaped (left). The real U-shaped and V-shaped profiles extracted from (and marked in) the DEM (right).

meters. The graph on the right of Figure 3 depicts two profiles extracted from the model, one corresponding to a glacial valley and the other corresponding to a fluvial valley (note that the coordinates are normalized in order to facilitate comparison).

In total, 137 training profiles (54 V-shaped and 83 U-shaped) and 25 test profiles (13 V-shaped and 12 U-shaped) were obtained (Figure 4).

Any differences in location of the center of the curves may imply differences in the profiles that are based solely on the relative position of the center of the profile and not on its U or V shape. For example, two mildly asymmetric V-shaped profiles could be considered to be very different if their centers are not properly aligned.

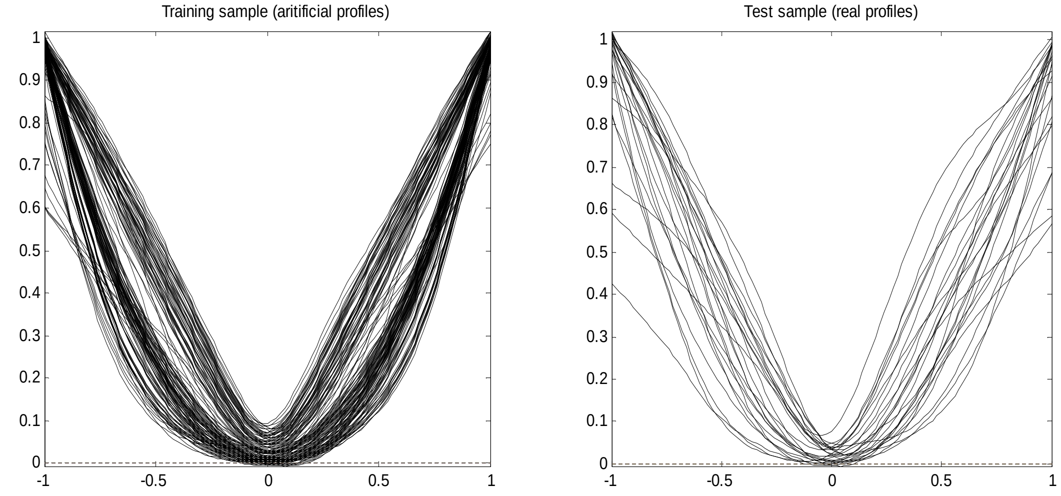
To simultaneously compare profiles on the basis of valley shape and not valley size, a similar scale is needed. For this reason, all the training and test profiles were normalized for the X and Y axes, the former in the interval $[-1\ 1]$ (with the minimum at zero), and the latter in $[0\ 1]$.

The training and test profiles were smoothed using a cubic B-splines basis with 30 breaks (32 basic functions). A number of different options for the number of breaks were evaluated producing similar results. Figure 4 shows the profiles that were registered as functional data in the manner described.

## 2.2 Results

Once the test and training profiles had been registered as functional data, the following techniques were used to construct the functional pattern recognition model: functional linear regression with a $0 - 1$ response variable, functional logistic regression, and functional support vector machines with linear and Gaussian kernels.

The parameters for the models (regularizer and Gaussian kernel parameter for the functional support vector machines, and regularizer for the functional logistic regres-


Training sample (aritificial profiles)

Test sample (real profiles)

Figure 4: Artificial U and V profiles (137) in the training sample (left), and real profiles (25) in the test sample (right).

| | Error Rate | |
|---|---|---|
| **Model** | **Train** | **Test** |
| Vectorial Linear | 0.01 | 0.20 |
| Vectorial Logistic | 0.00 | 0.24 |
| Vectorial SVM Linear kernel | 0.00 | 0.08 |
| Vectorial SVM Gaussian kernel | 0.00 | 0.08 |
| Functional Linear | 0.00 | 0.16 |
| Functional Logistic | 0.00 | 0.12 |
| Functional SVM Linear kernel | 0.00 | 0.04 |
| Functional SVM Gaussian kernel | 0.00 | 0.04 |

Table 1: Classification error rate for the training and test samples for the compared models.

sion) were selected using 10-fold cross-validation.

The results obtained are depicted in Table 1, which shows, for reference purposes, the results obtained using vectorial versions of the above models applied to the scalar observations of the profiles (from which the functional versions were constructed).

As can be observed, there were no training errors in any of the cases (except for the vectorial linear model), probably because of the theoretical nature of the training data and the relatively low level of noise. Nonetheless, in the test sample of real profiles, it can be observed how the functional model results improved significantly on the results produced by the vectorial versions of the same models. Likewise, of the functional models, the functional support vector machines produced the best results, irrespective of the kernel used (at least for this problem).

An analysis of the errors would indicate that these are associated with profiles that are genuinely difficult to classify as either U-shaped or V-shaped, which, in turn, would indicate the need to extend the typology to other classes not covered by these two main shapes (this will be the subject of future research).

## 3 Conclusions

In this work we evaluated the suitability of functional data analysis techniques for the morphological classification of the topographic profiles of U-shaped and V-shaped valleys.

From the results obtained it can be deduced that the functional techniques are an improvement on the vectorial techniques in terms of distinguishing between the two profile types, and that very satisfactory results can be obtained simply by training the system using simulated profiles. The functional models that produced the best results were the proposed functional support vector machines.

In this initial phase, the models were trained using only simulated profiles. However, on a gradual basis, these models will be further enriched with real profiles once their morphological characteristics have been established.

Future research will include the application of the methodology to a larger number of morphological groups using a multiclass pattern recognition approach, and enrichment of the model by means of the inclusion of derivatives of the profiles as functional covariables.

## Acknowledgements

J. M. Matías's research was supported by the Spanish Ministry of Education and Science, Grant No. MTM2005-00820.

## References

[1] J. Campbell, *Frost and fire*, Edinburgh: Edmonston and Douglas, vol I., 1865.

[2] W.Davis, *The Mission Range, Montana*, Geographical Review **2** (1916) 267-288.

[3] W.Graf, *The geomorphology of the glacial valley cross section.* Arctic and Alpine Research **2** (1970) 303-312.

[4] M. Greenwood, *M. Functional Data Analysis for Glaciated Valley Profile Analysis*, Ph.D., Department of Statistics, University of Wyoming. December, 2004.

[5] M. Greenwood and N. Humphrey, *Glaciated valley profiles: An application of nonlinear regression*, Computing Science and Statistics **34** (2002) 452-460.

[6] P. McCullagh and J. A. Nelder, *Generalized Linear Models*, Chapman & Hal, 1989.

[7] H. G. Muller, *Functional modelling and classification of longitudinal data*, Scandinavian Journal of Statistics, **32** (2005) 223–240.

[8] F. Pattyn and W. Van Huele, *Power law or power flaw?*, Earth Surface Processes and Landforms **23** (1998) 761-767.

[9] J. O. Ramsay and B. W. Silverman, *Functional data analysis*, Springer, 1997.

[10] B. Scholkopf and A. J. Smola, *Learning with Kernels*, MIT Press, 2002.

[11] H. Svensson, *Is the cross-section of a glacial valley a parabola?*, Journal of Glaciology, **3** (1959) 362-363.

[12] W.D. Thornbury, *Principles of geomorphology*, Wiley, New York, 1954.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# On the Numerical Evaluation of Option Prices in the Variance Gamma Model

**Anita Mayo**[1]

[1] *Mathematics Department, Baruch College, CUNY*

emails: `anita_mayo@baruch.cuny.edu`

**Abstract**

Because the pricing equations in Levy models contain integrals, it is difficult to develop rapid numerical methods for solving them. Although the integrals are not periodic, the standard evaluation methods use the FFT, and therefore require large computational regions to ensure accuracy. In earlier work we developed efficient methods for pricing options in two Levy models, the Merton model and the Kou double exponential model. The methods rely on the fact that in those models the density functions for the jump distributions satisfy ordinary or partial differential equations, so differential methods can be used to evaluate the integrals. In this paper we present effective numerical methods for pricing options in another Levy model that has been shown to work particularly well in actual market conditions, the Variance Gamma model. We also provide numerical results.

*Key words: Option pricing, Variance Gamma*

## 1 Introduction

The Black Scholes pricing model does not produce the heavy tails and asymmetry that one sees in practice. That is, large price movements occur more often than the Black Scholes model predicts, and log scaled upward and downward movements are not equally likely. These and other problems have led to the development of alternate pricing models. A common assumption is that in addition to the standard continuous Black Scholes process, the process the asset price follows has a component with jumps. Such processes are called jump processes.

The earliest jump model is due to Merton [12] who assumes the process the asset follows is the sum of the Black Scholes process and a jump process with lognormally distributed jumps with constant parameters. Other models which combine a local volatility function with a jump process have also been used for pricing options (Andersen and Andreasen, [3]). Kou [8] has suggested a model with double exponential distribution for the log jump size. Models where the process is a pure jump process and there is no continuous component have also been developed. For example, Madan, Carr, and Chang [9] more recently proposed what they call a Variance Gamma model. This model has been shown to be especially well suited for modeling in current market conditions, and is the model we are considering in this paper.

Some of these jump models are analytically tractable. For example, in the Merton model there is an analytic formula for European option prices in terms of Black Scholes prices, and there are formulas for certain option prices in the Kou and Variance Gamma models. But though analytic formulas exist for pricing certain options, numerical methods are required for pricing the most common ones, especially when the option payoffs are path dependent.

Most numerical methods for finding option prices involve solving partial integro-differential equations (PIDEs). In particular, in an early paper Amim [2] used simple but expensive tree methods to price European options. Later others (D'Halluin Forsyth and Vetzal,[6], [7]) implemented more sophisticated methods for evaluating vanilla European options, barrier options, and American options, and provided convergence results. These partial integro-differential equations have also also been solved by Matache, von Petersdorff and Schwab [10], Cont and Voltchkova [5] among others.

We note that combining a standard discretization method for the differential terms with a straightforward quadrature method for evaluating the integral term is costly since the integral must be approximated at each point of the mesh used for discretizing the the differential terms. However, it is possible to reduce the expense of evaluating the integral at all points of the computational grid by making an exponential change of variables. This converts the integral term into a correlation integral which can be evaluated at all mesh points simultaneously using the Fast Fourier Transform. This approach has often been used [2], [7].

One difficulty with Fourier methods is that they assume the function being evaluated is periodic. Since the integrals are not periodic, it is necessary to extend the computational region, which is expensive, especially in higher dimensions. Fourier methods are also slow to converge if the probability density is not smooth, as it is in the Kou double exponential model and in the Variance Gamma model.

In earlier work we [10], [4], presented a different and more efficient class of methods based on the fact that the integrals often satisfy differential equations which can be solved more rapidly than the integrals can be accurately evaluated by quadrature, even using Fourier methods.

In particular, we used the fact that at any moment of time the integral in the Merton PIDE is equal to a translation of a solution of a heat equation with initial values equal to the solution of the PIDE at that time. The evaluation time $\tau$ of the solution of the heat equation is half the variance of the jump process, and the translation amount is the expected value of the jump process. Therefore, we could evaluate the correlation integral by solving the heat equation numerically. Since the variance of the Poisson process is normally small we didn't need many time steps to solve the heat equation very accurately. The fact that Fourier methods are more expensive than usual differential methods for solving the heat equation meant that our methods were sometimes significantly faster. The methods we developed also extend to problems with similar density functions.

The method we developed for evaluating the convolution integral in the Kou double exponential model uses the fact that after a linear change of variables the integrand is separable. Specifically, after the change of variables the value of the integral at a point $x$ can be expressed as the product of an exponential function of $x$ and an integral where $x$ only appears as the lower or upper limit of integration. Therefore the integral can be evaluated at all $n$ points of the grid using only $Cn$ operations, where $C$ is a small number. The evaluation the integral can also be viewed as the solution of a first order ordinary differential equation. Again, our technique for evaluating the correlation integrals in the Kou model can be extended to evaluating the integrals in other similar models.

In this paper we extend the methods we have developed for pricing in the Kou model to pricing in the Variance Gamma model. The idea is that except very near the origin one can approximate the Variance Gamma density by a function that is piecewise the product of a linear function with the exponential with the same exponent as in the Variance Gamma density. That is, one can decompose the computational region into several subregions, and in each subregion one can approximate the Variance Gamma density by the product of the exponential term in the density and a different linear function. Once one has chosen the subregions, finding the best approximation only requires a few operations per region. Our numerical experiments demonstrate that one can achieve a very good approximation using around 4 or 5 subregions, so the cost of this part of the calculation is negligible.

The primary advantage of using such an approximation is that it greatly reduces the cost of evaluating the correlation integral. As in our method for pricing in the Kou model, one can evaluate the correlation integrals at each point of the computational mesh at a cost of just a few operations per mesh point.

An important issue in our methods is how to choose the approximating linear function. More precisely, it is necessary to find a good method for choosing the subregions. Finding the optimal subregions reduces to a a nonlinear least squares problem that in some cases could prove expensive to solve. On the other hand, as one might expect, having the nodes equally spaced does not produce extremely good results. We have chosen an intermediate method.

The Variance Gamma density cannot be well approximated by products of polynomials and exponentials very near the origin. Therefore, in a small region around each mesh point we must evaluate the integral directly by quadrature. This region only consists of a few mesh points, so the cost of this part of the calculation is small. It is important to note that although the Variance Gamma density is unbounded at the origin, the integrand of the correlation integral is always bounded.

An important advantage of our method is that it is not necessary to evaluate the integral terms using a uniform mesh, nor does one obtain increased speed by requiring the number of mesh points to be a power of 2. In particular, it should be possible to use a very nonuniform grid when solving the pricing equation for digital options, concentrating the mesh points near the payoff discontinuity. This cannot be done as efficiently using Fourier methods.

The organization of this paper is as follows. In section 2 we present the pricing PIDE and discretization method for the differential terms, in the next section we present our method for evaluating the integrals in the equation, and in the last section we present results of numerical experiments.

## 2 Pricing PIDE in the VG model

A Variance Gamma process is an infinite activity process of the form

$$X(t, \sigma, \theta, \nu) = b(\gamma(t, 1, \nu), \theta, \sigma)$$

where $\gamma(t, 1, \nu)$ is a unit mean gamma process and $b(t, \theta, \sigma) = \theta t + \sigma W(t)$ is a Brownian motion with drift $\theta$ and volatility $\sigma$. $W(t)$ is a standard Brownian motion, and $\nu$ is the variance of the gamma distributed time. If $r$ is the risk free interest rate and $q$ is the dividend rate, the risk neutral process for the price $S$ of a risky asset is

$$S(t) = S(0)e^{(r-q+\omega)t+X(t)}.$$

Here

$$\omega = \int_{-\infty}^{\infty} (1 - e^y)\, k(y)dy, \tag{2.1}$$

where $k(y)$ is the Levy measure

$$k(y) = \frac{e^{-\lambda_p y}}{\nu y} \text{ for } y > 0 \text{ and } k(y) = \frac{e^{-\lambda_n |y|}}{\nu |y|} \text{ for } y < 0$$

with

$$\lambda_p = \sqrt{\frac{\theta^2}{\sigma^4} + \frac{2}{\sigma^2 \nu}} - \frac{\theta}{\sigma^2}$$
$$\lambda_n = \sqrt{\frac{\theta^2}{\sigma^4} + \frac{2}{\sigma^2 \nu}} + \frac{\theta}{\sigma^2}.$$

The constant $\nu$ is a measure of the activity level, and $\lambda_m$ and $\lambda_p$ determine the exponential rates of decay of the density on the left and the right [9].

Under these assumptions European option prices $V(S, \tau)$ satisfy the PIDE

$$V_\tau + (r - q + \omega)SV_S + \int_{\infty}^{\infty} [V(Se^y, \tau) - V(S, \tau)]\, k(y)dy = rV. \tag{2.2}$$

Before solving the equation we first make the standard exponential change of variables $S = e^x$ and reverse time $(t = T - \tau)$:

$$V_t - (r - q + \omega)V_x = \int_{-\infty}^{\infty} [V(x + y, t) - V(x, t)]\, k(y)dy - rV. \tag{2.3}$$

We then discretize the differential and integral terms. Suppose $\{x_i\}$ are the discretization points, $h$ is the space step, and $V^n = (V_0^n, V_1^n, V_2^n, , , V_M^n)$, $n = 1, , , , , N$ are the the solution values at the $n$th time step. Following [1] we let $\omega(h) = \int_{y>h}(1-e^y)k(y)dy$, and using (2.1) rewrite (2.3):

$$V_t-(r-q+\omega(h))V_x = \int_{-\infty}^{\infty}\left[V(x+y,t)-V(x,t)\right]k(y)dy - \int_{y\le h}(e^y-1)V_x dy - rV. \quad (2.4)$$

We have chosen to discretize the differential terms using the Crank Nicolson scheme. Thus, at the $n$th time step we let

$$\frac{V_i^{n+1}-V_i^n}{\Delta t} = a\frac{V_{i+1}^{n+1}-V_{i-1}^{n+1}}{2h} + a\frac{V_{i+1}^{n}-V_{i-1}^{n}}{2h} - \frac{r}{2}\left(V_i^n + V_i^{n+1}\right) + I(V^n)$$

where $\Delta t$ is the time step, $a = q-r-\omega(h)$, and $I(V^n)$ is our approximation to the integrals in (2.4) at the $n$ time step.

Boundary values are needed at the edges of the computational domain. When pricing call options it is common to assume that $V_0^n = 0$ for all $n$, and $V_M^n = S_M - Ke^{-rt_n}$ where $K$ is the strike price, $S_M = e^{x_M}$ is the asset value at the upper edge of the computational region, and $t_n$ is the time at the $n$th time step. For a put we assume $V_0^n = Ke^{-rt_n}$ and $V_M^n = 0$.

At each time step the above equations and boundary conditions give rise to a tridiagonal linear system of equations, which can be solved at a cost of $5M$ operations per point at the first time step, and $3M$ operations at succeeding ones.

The Crank Nicolson scheme is in general second order accurate in $h$, but the above scheme is explicit in the integral and only first order accurate in $\Delta t$. In order to try to achieve second order accuracy in $\Delta t$ we have have used Richardson extrapolation. We note that several authors have solved the pricing PIDE using schemes that are implicit in the integral. In particular Wang et al. [13] used a semi Lagrangian discretization of the differential terms, and both Picard iteration and BiCGstab method to solve the implicit equations. We also note that when one is pricing European options one can can use an operator splitting method originally suggested by Andersen and Andreasen We used this method when we priced options in the Kou model [11], but although it is formally second order accurate we did not find the method to perform consistently better than extrapolation. Since our primary contribution is a rapid and accurate method for evaluating the integral we have not yet implemented an implicit method, nor have we priced any American or other exotic options. In future work we will address such issues.

At each time step one must also approximate the integral at all the points $\{x_i\}$. Most schemes for evaluating the integral rely on the fact that it is a correlation integral, and use a Fourier method. However, because the integral is not periodic, one must extend the computational region in order to avoid ”wrap around” effects. This increases the cost of the calculation.

In the next section we present a more efficient method for evaluating the integrals.

# 3 Evaluation of the Integral Terms

In this section we present our method for approximating the integrals in the Variance Gamma pricing equation:

$$\int_{-\infty}^{0} (V(x+y)-V(x))\, C\frac{e^{-\lambda_n|y|}}{|y|}dy+\int_{0}^{\infty} (V(x+y)-V(x))\, C\frac{e^{-\lambda_p y}}{y}dy-V_x(x)\int_{-h}^{h} (e^y-1)\, k(y)dy$$

$$= I^-(x)+I^+(x)+V_xI_h$$

We first show how to approximate $I^+(x)$. By the further change of variables $s=x+y$ $I^+(x)$ can be written

$$I^+(x)=\int_{x}^{\infty} (V(s)-V(x))\, \frac{e^{-\lambda(s-y)}}{(s-y)}ds$$

For any $x$ and $\epsilon$ this integral can be decomposed into a part $I_\epsilon(x)$ near $x$, and the remaining part $I_r(x)$:

$$I^+(x)=\int_{x}^{x+\epsilon} (V(s)-V(x))\, \frac{e^{-\lambda(s-x)}}{(s-x)}ds+\int_{x+\epsilon}^{\infty} (V(s)-V(x))\, \frac{e^{-\lambda(s-x)}}{(s-x)}ds$$

For $\epsilon, x_M>0$ the interval $[\epsilon, x_M]$ can be decomposed into $NP$ subregions $[\epsilon=a_0,a_1],[a_1,a_2],,[a_{NP-1},a_{NP}=x_M]$, and on each subregion $[a_i,a_{i+1}]$ the function $\frac{e^{-\lambda_p x}}{x}$ can be approximated by the product of the exponential $e^{\lambda_p x}$ and a linear function, $b_ix+c_i$. The approximations need not be continuous from one subinterval to the next.

With these approximations one can approximate the integral $I^r(x)=\sum_i I_i^r(x)$ where

$$I_i^r(x)=\int_{x+a_i}^{x+a_{i+1}} (V(s)-V(x))\, (b_i(s-x)+c_i)e^{-\lambda(s-x)}ds$$

$$=e^{\lambda x}\left[b_i\int_{x+a_i}^{x+a_{i+1}} V(s)se^{-\lambda s}ds+(c_i-b_ix)\int_{x+a_i}^{x+a_{i+1}} V(s)e^{-\lambda s}ds\right]$$

$$-e^{\lambda x}V(x)\left[b_i\int_{x+a_i}^{x+a_{i+1}} se^{-\lambda s}ds+(c_i-b_ix)\int_{x+a_i}^{x+a_{i+1}} e^{-\lambda s}ds\right]$$

The last two integrals in the above equation can be evaluated analytically.

The other two integrals can be approximated at all $M$ mesh points at a cost of $O(M)$ operations. For example, suppose we have approximated the first integral $\int_{x_j+a_i}^{x_j+a_{i+1}} V(s)e^{-\lambda s}ds$ at $x=x_j$. To evaluate the integral at $x=x_{j+1}$ we use the approximation

$$\int_{x_{j+1}+a_i}^{x_{j+1}+a_{i+1}} V(s)e^{-\lambda s}ds=\int_{x_j+a_i}^{x_j+a_{i+1}} V(s)e^{-\lambda s}ds$$

$$-\frac{(V(x_{j+1}+a_{i+1})+V(x_j+a_{i+1})}{2\lambda}\left(e^{-\lambda(x_{j+1}+a_{i+1})}-e^{-\lambda(x_j+a_{i+1})}\right)$$

$$+\frac{V(x_{j+1}+a_i)+V(x_j+a_i)}{2\lambda}\left(e^{-\lambda(x_{j+1}+a_i)}-e^{-\lambda(x_j+a_i)}\right)$$

Thus, to compute the integral at the next mesh point we only need add and subtract two terms. The exponentials in the above formulas can be stored and reused. The second integral can similarly be approximated at a point $x_{j+1}$ from its values at $x_j$.

To evaluate $I^+$ we must extend the solution $V(x)$ beyond the edge of the computational region. When pricing a call, for $x$ large we set the option value to $e^x - Ke^{-rt}$, and for $x \leq x_0$ we set it to 0. For puts we set option values to 0 for $x$ large, and $Ke^{-rt} - e^x$ for $x \leq x_0$. We assume that the computational region is large enough so these asymptotic approximations are sufficiently accurate.

We use essentially the same method to approximate $I^-(x)$ at all mesh points.

We also need to evaluate the integral $I_\epsilon(x)$. We always chose $\epsilon$ to be an integer number of mesh widths, $\epsilon = mh$. (In our calculations we have generally chosen $m = M/50$ mesh widths.) On the interval $[x_j + h, x_j + \epsilon]$, we approximate the integral using direct quadrature. Finally, we note that by Taylor series approximations (see [1]) one can show that the term $\int_{y \leq h}(V(x+s) - V(s) - V_x(e^s - 1))k(s)ds = O(h^2)$, and can therefore be neglected in the calculation.

It follows that the entire calculation is linear in the number of discretization points $M$. We also note that in our quadrature formulas truncation errors are due to approximating $V(x)$ by it's average value between mesh points, i.e. we integrate $e^{-\lambda x}$ and $xe^{-\lambda x}$ analytically.

An important issue in our method is the choice of the linear functions $b_i x + c_i$. When the nodes $\{a_i\}$ are prescribed beforehand we generally choose $\{b_i, c_i\} i = 1, , , NP-1$ so that

$$u(b_1, c_1, , , , , , b_{NP-1}, c_{NP-1}) = \sum_{i=1}^{NP-1} \int_{a_i}^{a_{i+1}} \left(b_i s + c_i - \frac{1}{s}\right)^2 e^{-\lambda s} ds = \text{ min}$$

If no other conditions are imposed this implies that for each $1 \leq i \leq NP - 1$

$$\frac{\partial u}{\partial b_i} = 2 \int_{a_i}^{a_{i+1}} s \left(b_i s + c_i - \frac{1}{s}\right) e^{-\lambda s} ds = 0 \tag{3.1}$$

and

$$\frac{\partial u}{\partial c_i} = 2 \int_{a_i}^{a_{i+1}} \left(b_i s + c_i - \frac{1}{s}\right) e^{-\lambda s} ds = 0. \tag{3.2}$$

The above $NP - 1$ two by two linear systems of equations determine $\{b_i\}$ and $\{c_i\}$.

We also performed calculations where we required the approximations be continuous from one interval to the next. Continuity implies

$$b_i x_{i+1} + c_i = b_{i+1} x_{i+1} + c_{i+1}, \tag{3.3}$$

which in turn implies the values of $\{b_i\}$ determine the values of $\{c_i\}$ for $i = 2, , , NP-1$. We also require

$$\frac{\partial u}{\partial b_i} = 2 \int_{a_i}^{a_{i+1}} s \left(b_i s + c_i - \frac{1}{s}\right) e^{-\lambda s} ds = 0 \text{ for } i = 1, 2, , , NP - 1. \tag{3.4}$$

and

$$\frac{\partial u}{\partial c_1} = 2\int_{a_i}^{a_{i+1}} (b_1 s + c_1 - 1)\, e^{-\lambda s} ds = 0. \tag{3.5}$$

This set of equations (3.3)- (3.5) gives rise to a block lower bidiagonal matrix equation easily solved by $LU$ decomposition. However, as one might expect, we always obtained more accurate solutions when we did not impose continuity.

We also performed calculations where the nodes $\{a_i\}$ were not specified a priori. That is, for fixed $NP-1$ we found $3NP-4$ numbers $b_1, c_i 1 \leq i \leq NP-1$, and $a_j, 2 \leq j \leq NP-1$ such that

$$v(b_i, c_i, a_i) = \sum_{i=1}^{M-1} \int_{a_i}^{a_{i+1}} \left(b_i s + c_i - \frac{1}{s}\right)^2 e^{-\lambda s} ds = \text{ min} \tag{3.6}$$

and

$$b_i a_{i+1} + c_i = b_{i+1} a_{i+1} + c_{i+1}. \tag{3.7}$$

Because the nodes $\{a_i\}$ are not given this problem is nonlinear.

By (3.7) the values of $\{c_i\}$ $i = 2, , , NP-1$ can be expressed in terms of the values of $\{b_i, a_i\}$ and $c_1$. Therefore, we can view $v$ as a function of the $2NP-2$ variables $\{b_i\} 1 \leq i \leq NP-1, \{a_i\}, 2 \leq i \leq NP-1$.

To determine values of the variables we use the fact that at the minimum of $v$ its derivatives with respect to all them are 0:

$$\frac{\partial v}{\partial b_i} = \frac{\partial v}{\partial a_i} = \frac{\partial v}{\partial c_1} = 0. \tag{3.8}$$

In our numerical experiments we solved this system of equations by a modified damped Newton's method. We note that for a given interpolation domain $[\epsilon, x_M]$ and value of $\lambda$ the above calculations need only be performed once, i.e. the approximations do not depend on the values of any other financial parameters. However, the calculations are much more expensive than ones where nodes are specified.

We also note that it may also be possible to approximate the Variance Gamma density with other functions. Another possibility is having the approximating function be piecewise the product of a quadratic with an exponential. If one makes such an approximation one can still evaluate the correlation integral at a cost of a few operations per meshpoint, although the constant will be larger. However one should be able to use a smaller number of subregions, and this would decrease the cost of the calculation.

## 4 Numerical Results

In this section we report on results of numerical experiments we performed. We note that the only way in which our method differs from others is in the way that we evaluate the integrals. We therefore first performed calculations to test the accuracy of our quadrature method.

We first tested our method on the function $V(x) = x$ for which the the integral $\int_a^b (V(x+s) - V(x)) \frac{e^{\lambda s}}{s} ds$ can be evaluated analytically. Since $V(x+s) - V(x) = s$, at any point $x$ the value of the integral is $\int_a^b e^{-\lambda s} ds = -\frac{1}{\lambda}\left(e^{-\lambda b} - e^{-\lambda a}\right)$. Furthermore, when using our quadrature formula on this function there is no truncation error, so errors are only due to interpolating the density function. In these experiments we let $a = .01535$ and $b = 6.139$.

Results of our calculations are given in Table 1. In the table $NP$ is the number of subregions, and $\lambda$ is the exponent in the density (We assumed that $\lambda_p = \lambda_n = \lambda$. The numbers in the last column are errors, which were the same at all quadrature points. In these and other experiments we always let $C = 1$, when $\lambda$ was 5 we used the nodes $a_i = \epsilon + i/ds^{2.33}$ and when $\lambda$ was 10 (or 15) we let $a_i = \epsilon + i/ds^{2.53}$ where $ds = x_M - x_0$. We used M = 200 quadrature points to perform the calculation.

Table 1: Evaluation of Integral with $V(x) = x$

| NP | $\epsilon$ | $\lambda$ | error |
|---|---|---|---|
| 2 | 5. | .246 | 9.89E-6 |
| 3 | 5. | .246 | 8.28E-6 |
| 4 | 5. | .246 | 1.70E-6 |
| 2 | 10. | .246 | 4.89E-6 |
| 3 | 10. | .246 | 2.34E-6 |
| 4 | 10. | .246 | 9.61E-7 |
| 5 | 10. | .246 | 5.94E-7 |
| 2 | 5. | .123 | 3.34E-5 |
| 3 | 5. | .123 | 1.78E-5 |
| 4 | 5. | .123 | 8.97E-6 |
| 5 | 5. | .123 | 6.41E-6 |
| 3 | 5. | .062 | 5.78E-4 |
| 4 | 5. | .062 | 4.26E-5 |
| 5 | 5. | .062 | 2.43E-5 |

From these results we see that using 4 subregions or panels for interpolating provides a reasonable level of accuracy, and increasing the number does not change the results significantly. Choosing $\epsilon$ smaller, of course, decreases the level of accuracy, but also decreases the cost of the calculation for a given value of $h$.

We next tested our method of evaluating the integral with $V(x)$ equal to the payoff values for a call with strike $K = 100$, that is $V(x) = e^x - K$ if $e^x \geq K$, and $V(x) = 0$ otherwise. We compared the results with the exact values of the integral, which can be expressed in terms of the exponential integral function. (We used MATLAB to determine the values.) Results are given in Table 2. In this table $M$ is the number of discretization points and $S = e^x$ is the asset price at which we evaluated the integral. In this eample we always used $NP = 4$ subregions for interpolating the VG density, and integrated over the interval [.00674,5.43], We let $\epsilon = .0691$ This $\epsilon$ corresponds to m=2 for M=100, to m=4 for M=200, m=8 for M=400, and m=16 for M=800. The exact values of the integral for $\lambda = 5.$ are 21.650 at $S = 100.$ and 23.815 at $S = 110.$ For $\lambda = 15.0$ the value of the integral is 6.2572 at $S = 100.0$.

Table 2: Evaluation of Integral with $V(x) = x$

| $M$ | S | $\lambda$ | error | | |
|---|---|---|---|---|---|
| 100 | 100. | 5 | 2. | 21.730 | 0.080 |
| 200 | 100. | 5 | 4. | 21.671 | 0.021 |
| 400 | 100. | 5 | 8. | 21.657 | 0.006 |
| 800 | 110. | 5 | 16. | 21.651 | 0.001 |
| 100 | 110. | 5 | 2. | 23.742 | 0.0027 |
| 200 | 110. | 5 | 4. | 23.626 | 0.0011 |
| 400 | 110. | 5 | 8. | 23.819 | 0.0004 |
| 800 | 110. | 5 | 16. | 23.816 | 0.0001 |
| 100 | 100. | 15 | 2. | 6.316 | 0.059 |
| 200 | 100. | 15 | 4. | 6.279 | 0.022 |
| 400 | 100. | 15 | 8. | 6.264 | 0.007 |
| 800 | 110. | 15 | 16. | 6.259 | 0.002 |

These results show that see that the method is almost second order accurate, and we can achieve this level of accuracy using relatively few panels.

We also performed calculations where we used nodes determined by solving (3.8). It turned out that although these optimal nodes provided better results, the results were not significantly different. That is, they were usually only better, by a (linear) factor of 4, and the convergence rate was essentially the same.

In our final set of experiments we priced the European call and put options with $\lambda_n = \lambda_p = 5$, $r = .1, T = 1, K = 100$ at $S = 100$. Almendral and Oosterlee [1] priced the call option using a fine mesh, and determined the price to be 15.131. We obtained the same result. The price of the put is 5.6147. In our calculations we again chose $\epsilon = .0691$, but used $NP = 5$ subregions for interpolating, and used extrapolation with respect to $\Delta t$. The numbers in the second column are the number of time steps in the finer mesh, and the numbers in the third column are the computed values.

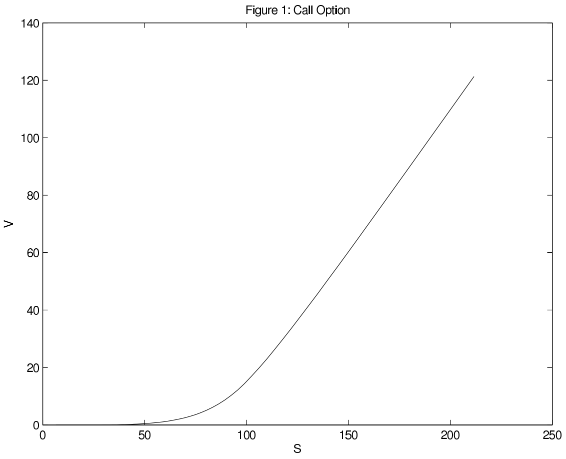
Table 3: Evaluation of Call

| M | nt | Comp. vale | error | rate |
|---|---|---|---|---|
| 100 | 30 | 15.205 | 0.074 | . |
| 200 | 60 | 15.153 | 0.022 | 3.4 |
| 400 | 120 | 15.137 | 0.006 | 3.8 |
| 800 | 120 | 15.132 | 0.001 | 4.0 |

Table 4: Evaluation of Put

| M | nt | Comp. value | error | rate |
|---|---|---|---|---|
| 100 | 30 | 5.6509 | 0.0362 | . |
| 200 | 60 | 5.6220 | 0.0073 | 4.9 |
| 400 | 120 | 5.6168 | 0.0021 | 3.5 |
| 800 | 240 | 5.6152 | 0.0005 | 4.1 |

Figure 1 is the graph of the call price.



# 5 References

[1] A. Almendral and C. Oosterlee,*On American options under the Variance Gamma Process*, working paper.

[2] K. Amin, *Jump diffusion option valuation in discrete time*, Journal of Finance,**48**: (1993) 1833-1863.

[3] Andersen L. and J. Andreasen, *Jump diffusion processes: Volatility smile fitting and numerical methods for option pricing*, Rev. of Derivatives Research, **4** (2000) 231-262.

[4] P. Carr and A. Mayo, *On the numerical evaluation of option prices in jump diffusion models,* The European Journal of Finance, to appear, Proceedings of Forecasting Financial Markets Conference, Paris, (2004). June, 2004.

[5] Cont R. and E. Voltchkova, *A finite difference scheme for option pricing in jump diffusion and exponential Levy models*, Report 513, CMAP, Ecole Polytechnique, 2003.

[6] d'Halluin, Y., P.A. Forsyth, and G. Labahn, *A Penalty Method for American Options with jump diffusion processes*, Numerische Mathematik, **97** (2004) 321-352.

[7] Y. d'Halluin, P.A. Forsyth, and K. R. Vetzal, *A semi-Lagrangian approach for American Asian options under jump diffusion*, SIAM J. Sci. Comput., 27 (2005) 315-335.

[8] S. Kou, *A jump diffusion model for option pricing*, Management Science, 48, (2002) 1086-1101.

[9] Madan, P., P. Carr, and E. Chang, *The variance gamma process and option pricing*, European Financial Review, **1** (1998) 75-105.

[10] Matache, A., T. von Petersdorff, and C. Schwab, *Fast deterministic pricing of options on Levy driven assets*, RiskLab Research Report, ETH, Zurich (2002).

[11] Mayo, A., *Methods for the pricing of PIDES in exponential and Merton models*, Journal of Computational and Applied Math., to appear.

[12] R. Merton, *Option Pricing when underlying stock returns are discontinuous*, Journal of Financial Economics, **3** (1976) 125-144.

[13] I Wang, J. Wan, and P. Forsyth, *Robust Numerical Valuation of European and American options under the CGMY process*, working paper.

# Black-Scholes equation: Green's function solution for terminal-boundary value problems

**Max Melnikov**
*Labry School of Business and Economics*
*Cumberland University, USA*

e-mail: mmelnikov@cumberland.edu

## Abstract

A Green's function-based approach is proposed for obtaining solution to a variety of terminal-boundary value problems stated for the Black-Scholes equation that simulates the valuation of European option pricing. It is shown that analytic representations of required Green's functions can be constructed allowing a closed form solution to the considered problems.

*Key words*: *Black-Scholes equation, Green's function method*

## 1. Introduction

A linear backward in time parabolic type partial differential equation, which is referred to as the Black-Scholes equation [1], is widely used in the field of financial engineering in nowadays for quantitative and qualitative analysis of option pricing problems. Green's function-based methods are traditionally used in partial differential equations. They could also be productive for the Black-Scholes equation, but a limited number of computer-ready representations of Green's functions for this equation represents a considerable obstacle.

This study aims at the development of a Green's function-based analytic approach to a class of terminal-boundary value problems stated for the nonhomogeneous Black-Scholes equation

$$\frac{\partial V(S,t)}{\partial t}+\frac{\sigma^2S^2}{2}\frac{\partial^2V(S,t)}{\partial S^2}+rS\frac{\partial V(S,t)}{\partial S}-rV(S,t)=\Phi(S,t) \tag{1}$$

As an example, we consider a problem stated for Eq (1) in the semi-infinite strip-shaped region $\Omega=(S_1<S<S_2)\times(T>t>-\infty)$ of the $S,t$-plane.

Let a terminal condition be given by

$$V(S,T)=\varphi(S) \tag{2}$$

while the Dirichlet type boundary conditions

$$V(S_1,t)=A(t) \qquad \text{and} \qquad V(S_2,t)=B(t) \tag{3}$$

be imposed on the edges $S=S_1$ and $S=S_2$ of the $\Omega$ region.

In the above problem setting, $V=V(S,t)$ is the price of the derivative product, $\varphi(S)$ is the pay-off function of a given derivative problem at the expiration time $T$, with $S$ and $t$ being the share price of the underlying asset and time, respectively. The parameters $\sigma$ and $r$ represent the volatility of the underlying asset and the risk-free interest rate, respectively.

It is evident that upon introducing of a new unknown function $v(S,t)$

$$V(S,t)=v(S,t)+\frac{S-S_1}{S_2-S_1}[B(t)-A(t)]+A(t) \tag{4}$$

the terminal-boundary value problem in Eqs (1)-(3) reduces to

$$\frac{\partial v(S,t)}{\partial t}+\frac{\sigma^2 S^2}{2}\frac{\partial^2 v(S,t)}{\partial S^2}+rS\frac{\partial v(S,t)}{\partial S}-rv(S,t)=F(S,t) \tag{5}$$

$$v(S,T)=f(S) \tag{6}$$

with the homogeneous boundary conditions imposed as

$$v(S_1,t)=0 \qquad \text{and} \qquad v(S_2,t)=0 \tag{7}$$

In view of the relation in Eq (4), the right-hand side functions $F(S,t)$ and $f(S)$ in Eqs (5) and (6) are expressed in terms of the right-hand sides $\Phi(S,t)$ and $\phi(S)$ of Eqs (1) and (2) as

$$F(S,t)=\Phi(S,t)+\frac{S-S_1}{S_2-S_1}[r\,(B(t)-A(t))+A'(t)-B'(t)]$$

$$-A'(t)+rA(t)+rS\frac{A(t)-B(t)}{S_2-S_1}$$

and

$$f(S)=\varphi(S)-\frac{S-S_1}{S_2-S_1}[B(T)-A(T)]-A(T)$$

As it follows from the qualitative theory of partial differential equations [2], the solution to the problem setting in Eqs (5)-(7) can be written as the following sum of two integral representations

$$v(S,t)=\int\limits_{S_1}^{S_2} G(S,t;\widetilde{S})f(\widetilde{S})d\widetilde{S}+\int\limits_{t}^{T}\int\limits_{S_1}^{S_2} G(S,t-\widetilde{t};\widetilde{S})F(\widetilde{S},\widetilde{t})d\widetilde{S}d\widetilde{t} \tag{8}$$

where $G(S,t;\widetilde{S})$ is the Green's function to the homogeneous terminal-boundary value problem corresponding to that in Eqs (5)-(7).

Thus, to obtain a computer-friendly analytic solution to the terminal-boundary value problem posed by Eqs (1)-(3), one ought to have a compact form of the Green's function $G(S,t;\widetilde{S})$. The derivation procedure for such a form is described in detail in the next section.

## 2. Construction of the Green's function

Our approach to the construction of Green's function flows out from the procedure which was developed earlier [3] for problems in applied mechanics. It is based on a combination of two classical methods of applied mathematics. These are: the method of integral Laplace transform that is widely implemented in the heat and mass transfer, for example, and the method of variation of parameters traditionally used for finding general solution for linear high-order ordinary differential equations. In going through the procedure, we consider the terminal-boundary value problem

$$\frac{\partial v(S,t)}{\partial t}+\frac{\sigma^2 S^2}{2}\frac{\partial^2 v(S,t)}{\partial S^2}+rS\frac{\partial v(S,t)}{\partial S}-rv(S,t)=0 \tag{9}$$

$$v(S,T)=f(S) \tag{10}$$

$$v(S_1,t)=0 \qquad \text{and} \qquad v(S_2,t)=0 \tag{11}$$

Recall [2] that the solution to the above problem can be written in terms of the Green's function $G(S,t;\widetilde{S})$ to the homogeneous problem corresponding to that in Eqs (5)-(7) as

$$v(S,t)=\int\limits_{S_1}^{S_2} G(S,t;\widetilde{S})f(\widetilde{S})d\widetilde{S} \tag{12}$$

This representation is used, in the present study, to actually derive the Green's function $G(S,t;\widetilde{S})$. The emphasis is put, in our study, on the parabolic type single-parameter equation forward in time

$$\frac{\partial u(x,\tau)}{\partial \tau}=\frac{\partial^2 u(x,\tau)}{\partial x^2}+(c-1)\frac{\partial u(x,\tau)}{\partial x}-cu(x,\tau) \tag{13}$$

which traditionally [4, 5] arises from Eq (9) by introducing new independent variables $x$ and $\tau$

$$x=\ln S \qquad \text{and} \qquad \tau=\frac{\sigma^2}{2}(T-t) \tag{14}$$

The parameter $c$ in (13) is defined in terms of $r$ and $\sigma$ as $c=2r/\sigma^2$. Hence, Eq (13) has, in contrast to the Black-Scholes equation, constant coefficients. This significantly simplifies the situation.

Upon introducing the variables $x$ and $\tau$ in compliance with the relations of Eq (14), the terminal-boundary value problem of Eqs (9)-(11) transforms into the following initial-boundary value problem

$$u(x,0)=f(e^x) \tag{15}$$

$$u(a,\tau)=0 \qquad \text{and} \qquad u(b,\tau)=0 \tag{16}$$

for the equation in (13) posed on the semi-infinite strip-shaped region $(a<x<b)\times(0<\tau<\infty)$ in the $x,\tau$-plane, where

$$a=\ln S_1 \qquad \text{and} \qquad b=\ln S_2$$

Applying the Laplace transform

$$U(x;s)=L\{u(x,\tau)\}=\int_0^{\infty} e^{-s\tau}u(x,\tau)d\tau$$

to the setting in Eqs (13), (15) and (16), we obtain the following boundary value problem

$$\frac{d^2U(x;s)}{dx^2}+(c-1)\frac{dU(x;s)}{dx}-(s+c)U(x;s)=-f(e^x) \tag{17}$$

$$U(a;s)=0, \qquad U(b;s)=0 \tag{18}$$

for the Laplace transform $U(x;s)$ of $u(x,\tau)$.

In compliance with the method of variation of parameters, the general solution to Eq (17) is found as

$$U(x,s)=\int\limits_a^x \frac{e^{\alpha(x-\xi)}}{2\omega}\left(e^{(\xi-x)\omega}-e^{(x-\xi)\omega}\right)f(e^{\xi})d\xi$$

$$+M(s)e^{(\alpha+\omega)x}+N(s)e^{(\alpha-\omega)x} \tag{19}$$

where the parameter $\omega$ is defined as $\omega=\sqrt{s+\beta}$, while the parameters $\alpha$ and $\beta$ are expressed as

$$\alpha=\frac{1-c}{2} \qquad \text{and} \qquad \beta=\left(\frac{1+c}{2}\right)^2$$

Satisfaction of the boundary conditions of Eq (18) yields the system of linear algebraic equations

$$\begin{pmatrix} e^{(\alpha+\omega)a} & e^{(\alpha-\omega)a} \\ e^{(\alpha+\omega)b} & e^{(\alpha-\omega)b} \end{pmatrix}\times\begin{pmatrix} M(s) \\ N(s) \end{pmatrix}=\begin{pmatrix} 0 \\ \Psi(s) \end{pmatrix} \tag{20}$$

in $M(s)$ and $N(s)$, where

$$\Psi(s)=-\int\limits_a^b \frac{1}{2\omega}\left[e^{(\alpha-\omega)(b-\xi)}-e^{(\alpha+\omega)(b-\xi)}\right]f(e^{\xi})d\xi$$

Solving the system in (20), we obtain

$$M(s)=\int\limits_a^b \frac{e^{(\alpha-\omega)a}e^{\alpha(b-\xi)}\left[e^{(\xi-b)\omega}-e^{(b-\xi)\omega}\right]}{2\omega\left[e^{(a-b)\omega}-e^{(b-a)\omega}\right]}f(e^{\xi})d\xi$$

and

$$N(s) = -\int_a^b \frac{e^{(\alpha+\omega)a} e^{\alpha(b-\xi)} \left[e^{(\xi-b)\omega} - e^{(b-\xi)\omega}\right]}{2\omega\left[e^{(a-b)\omega} - e^{(b-a)\omega}\right]} f(e^{\xi})d\xi$$

Upon substituting these in (19), the latter reads as

$$U(x, s) = \int_a^x \frac{e^{\alpha(x-\xi)}}{2\omega}\left(e^{(\xi-x)\omega} - e^{(x-\xi)\omega}\right) f(e^{\xi})d\xi$$

$$+\int_a^b \frac{e^{\alpha(x-\xi)}\left[e^{(x-a)\omega} - e^{(a-x)\omega}\right]\left[e^{(\xi-b)\omega} - e^{(b-\xi)\omega}\right]}{2\omega\left[e^{(a-b)\omega} - e^{(b-a)\omega}\right]} f(e^{\xi})d\xi$$

which can be expressed in a single-integral form

$$U(x; s) = \int_a^b \frac{e^{\alpha(x-\xi)}}{2\omega\left[e^{(a-b)\omega} - e^{(b-a)\omega}\right]}$$

$$\times\Big\{e^{[(x+\xi)-(a+b)]\omega} + e^{[(a+b)-(x+\xi)]\omega}$$

$$-e^{[(a-b)+|x-\xi|]\omega} - e^{[(b-a)-|x-\xi|]\omega}\Big\} f(e^{\xi})d\xi$$

Transforming the factor $e^{(a-b)\omega} - e^{(b-a)\omega}$ in the denominator as

$$e^{(a-b)\omega} - e^{(b-a)\omega} = -e^{(b-a)\omega}\left[1 - e^{2(a-b)\omega}\right]$$

we rewrite the above representation for $U(x; s)$

$$U(x; s) = -\int_a^b \frac{e^{\alpha(x-\xi)}}{2\omega e^{(b-a)\omega}\left[1 - e^{2(a-b)\omega}\right]}$$

$$\times\Big\{e^{[(x+\xi)-(a+b)]\omega} + e^{[(a+b)-(x+\xi)]\omega}$$

$$-e^{[(a-b)+|x-\xi|]\omega} - e^{[(b-a)-|x-\xi|]\omega}\Big\} f(e^{\xi})d\xi \qquad (21)$$

The immediate inverse Laplace transform of $U(x; s)$ is problematic if the latter is kept in its current form. Therefore, we adjust it first by representing the factor $1/[1 - e^{2(a-b)\omega}]$ in the integrand in (21) as the sum of the geometric series

$$\frac{1}{1 - e^{2(a-b)\omega}} = \sum_{n=0}^{\infty} e^{2n(a-b)\omega}$$

whose common ratio $e^{2(a-b)\omega}$ is clearly less than one. This transforms (21) into

$$U(x; s) = \int_a^b \frac{e^{\alpha(x-\xi)}}{2\omega}\sum_{n=0}^{\infty}\Big\{e^{-[-|x-\xi|-2(n+1)(a-b)]\omega}$$

$$+e^{-[|x-\xi|-2n(a-b)]\omega}-e^{-[2b-(x+\xi)-2n(a-b)]\omega}$$
$$-e^{-[(x+\xi)-2a-2n(a-b)]\omega}\Big\} f(e^{\xi})d\xi \tag{22}$$

and the inverse Laplace transform of the above can be accomplished in the term-by-term manner. This yields the solution $u(x,\tau)$ to the initial-boundary value problem in Eqs (13), (15) and (16) in the form

$$u(x,\tau)=L^{-1}\{U(x,s)\}$$
$$=\int_a^b \frac{e^{\alpha(x-\xi)}e^{-\beta\tau}}{2\sqrt{\pi\tau}}\sum_{n=0}^{\infty}\Bigg\{\exp\left(-\frac{[|x-\xi|+2(n+1)(a-b)]^2}{4\tau}\right)$$
$$+\exp\left(-\frac{[|x-\xi|-2n(a-b)]^2}{4\tau}\right)-\exp\left(-\frac{[2b-(x+\xi)-2n(a-b)]^2}{4\tau}\right)$$
$$-\exp\left(-\frac{[(x+\xi)-2a-2n(a-b)]^2}{4\tau}\right)\Bigg\}f(e^{\xi})d\xi$$

which converts to a more compact form by rearranging the summation in the above series. This yields

$$u(x,\tau)=\int_a^b \frac{e^{\alpha(x-\xi)}e^{-\beta\tau}}{2\sqrt{\pi\tau}}\sum_{m=-\infty}^{\infty}\Bigg\{\exp\left(-\frac{[|x-\xi|+2m(a-b)]^2}{4\tau}\right)$$
$$-\exp\left(-\frac{[2b-(x+\xi)-2m(a-b)]^2}{4\tau}\right)\Bigg\}f(e^{\xi})d\xi$$

The solution $v(S,t)$ to the setting in Eqs (9)-(11) can be attained by the backward substitution of the variables $x$, $\tau$ and $\xi$ with $S$, $t$ and $\overline{S}$, respectively, in compliance with the relations in Eq (14). With $\alpha$ and $\beta$ replaced with the original parameters $r$ and $\sigma$ of the Black-Scholes equation, we obtain $v(S,t)$ in the form

$$v(S,t)=\int_{S_1}^{S_2}\frac{\exp\Big(-\frac{r-\sigma^2/2}{\sigma^2}\ln(S/\widetilde{S})-\frac{(r+\sigma^2/2)^2}{2\sigma^2}(T-t)\Big)}{\sigma\widetilde{S}\sqrt{2\pi(T-t)}}$$
$$\times\sum_{m=-\infty}^{\infty}\Bigg\{\exp\left(-\frac{[\ln(S/\widetilde{S})+2m\ln(S_1/S_2)]^2}{2\sigma^2(T-t)}\right)$$
$$-\exp\left(-\frac{[\ln(S_2^2/S\widetilde{S})-2m\ln(S_1/S_2)]^2}{2\sigma^2(T-t)}\right)\Bigg\}f(\widetilde{S})d\widetilde{S}$$

which can be transformed, by combining the logarithmic components in the series factor. This yields

$$v(S,t)=\int_{S_1}^{S_2}\frac{\exp\Big(-\frac{r-\sigma^2/2}{\sigma^2}\ln(S/\widetilde{S})-\frac{(r+\sigma^2/2)^2}{2\sigma^2}(T-t)\Big)}{\sigma\widetilde{S}\sqrt{2\pi(T-t)}}$$

$$\times \sum_{m=-\infty}^{\infty} \left\{ \exp\left( -\frac{[\ln(SS_1^{2m}/\widetilde{S}S_2^{2m})]^2}{2\sigma^2(T-t)} \right) \right.$$

$$\left. - \exp\left( -\frac{[\ln(S_2^{2(m+1)}/S\widetilde{S}S_1^{2m})]^2}{2\sigma^2(T-t)} \right) \right\} f(\widetilde{S})d\widetilde{S} \tag{23}$$

Thus, in view of the integral representation in (12), the kernel

$$G(S,t;\widetilde{S}) = \frac{\exp\left( -\frac{r-\sigma^2/2}{\sigma^2} \ln(S/\widetilde{S}) - \frac{(r+\sigma^2/2)^2}{2\sigma^2}(T-t) \right)}{\sigma\widetilde{S}\sqrt{2\pi(T-t)}}$$

$$\times \sum_{m=-\infty}^{\infty} \left\{ \exp\left( -\frac{[\ln(SS_1^{2m}/\widetilde{S}S_2^{2m})]^2}{2\sigma^2(T-t)} \right) - \exp\left( -\frac{[\ln(S_2^{2(m+1)}/S\widetilde{S}S_1^{2m})]^2}{2\sigma^2(T-t)} \right) \right\} \tag{24}$$

of the integral form for $v(S,t)$ in Eq (23) represents the Green's function to the homogeneous setting corresponding to that in Eqs (9)-(11).

The series in the above representation converges at a high rate. This implies that, in computing values of $G(S,t;\widetilde{S})$, any accuracy level required for real applications can be attained by appropriately truncating its series to the $M$-th partial sum as

$$G(S,t;\widetilde{S}) \approx \frac{\exp\left( -\frac{r-\sigma^2/2}{\sigma^2} \ln(S/\widetilde{S}) - \frac{(r+\sigma^2/2)^2}{2\sigma^2}(T-t) \right)}{\sigma\widetilde{S}\sqrt{2\pi(T-t)}}$$

$$\times \sum_{m=-M}^{M} \left\{ \exp\left( -\frac{[\ln(SS_1^{2m}/\widetilde{S}S_2^{2m})]^2}{2\sigma^2(T-t)} \right) - \exp\left( -\frac{[\ln(S_2^{2(m+1)}/S\widetilde{S}S_1^{2m})]^2}{2\sigma^2(T-t)} \right) \right\} \tag{25}$$

A multi-parameter numerical experiment has been conducted to develop practical recommendations as to the choice of the truncation parameter $M$ in Eq (25). Approximate values of $G(S,t;\widetilde{S})$ were computed, with a wide range of the parameters $r$, $\sigma$, $S_1$ and $S_2$ observed. The experiment strongly suggests that $M \geq 5$ is a sufficient condition for obtaining values of $G(S,t;\widetilde{S})$ from Eq (25) accurate to the sixth decimal place. This allows the form from Eq (25) to be practically used in valuating this Green's function.

## 3. Illustrative examples

Clearly, with the compact computer-friendly series expansion of the Green's function $G(S,t;\widetilde{S})$ that we just derived, the integral representation in Eq (8) delivers a closed analytic form solution $v(S,t)$ to the problem in Eqs (5)-(7), while Eq (4) brings the solution $V(S,t)$ to the nonhomogeneous terminal-boundary value problem in Eqs (1)-(3) that we started with.

In Figure 1, we depict a profile of the Green's function $G(S,t;\widetilde{S})$ shown in Eq (25), where the parameters in the problem statement were chosen as: $r=0.06$, $\sigma=0.8$, $S_1=1.0$, $S_2=5.0$, $T=2.0$, the source point was fixed as $\widetilde{S}=2.0$, while the series was truncated at $M=5$.

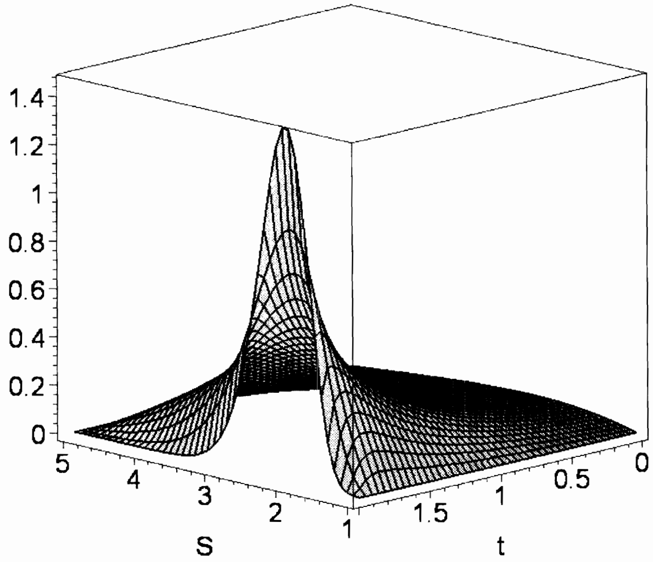


**Figure 1:** Profile of the Green's function $G\left(S,t;\tilde{S}\right)$ from Eq (25)

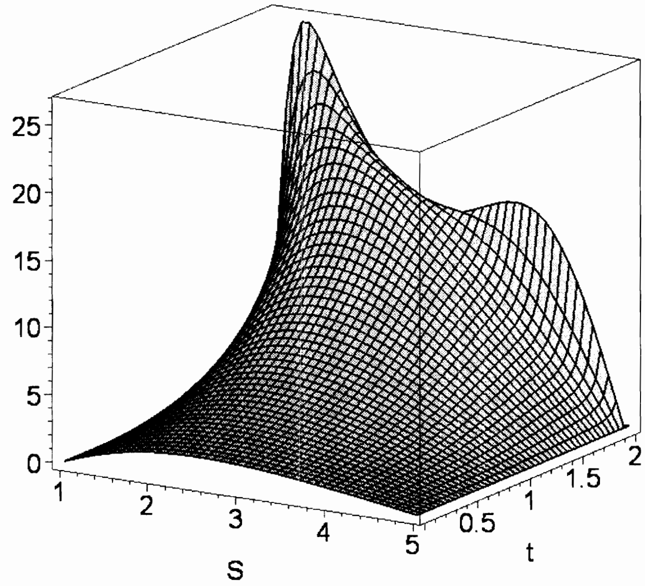


**Figure 2:** Solution of the problem in Eqs (9) – (11)

Figure 2 exhibits the solution $v(S,t)$ to the statement in Eqs (9)-(11), where $r=0.06$, $\sigma=0.8$, $S_1=1.0$, $S_2=5.0$, $T=2.0$, while the right-hand side function in the terminal condition of Eq (10) is chosen as

$$f(S)=5\left[\left(S-\frac{S_1+S_2}{2}\right)^2+2\right]$$

and the truncation parameter of the series in Eq (25) is fixed as $M=5$.

## 4. Other Green's functions

The proposed technique for the construction of computer-friendly representations of Green's functions is also productive in other problem settings for the Black-Scholes equation. To illustrate this assertion, we consider, as an example, a terminal-boundary value problem on the semi-infinite strip-shaped region $\Omega=(0<S<D)\times(T>t>-\infty)$ with boudary conditions imposed as

$$|v(0,t)|<\infty \qquad \text{and} \qquad \frac{\partial v(D,t)}{\partial S}=0 \tag{26}$$

Upon implementing the approach developed in this study, we obtain the following compact expression

$$G(S,t;\widetilde{S})=\frac{1}{\widetilde{S}}\left(\frac{S}{\widetilde{S}}\right)^{\alpha}\exp\left(-\beta\frac{\sigma^2}{2}(T-t)\right)$$
$$\times\left\{\frac{1}{\sigma\sqrt{2\pi(T-t)}}\left[\exp\left(-\frac{\left[\ln(S/\widetilde{S})\right]^2}{2\sigma^2(T-t)}\right)+\exp\left(-\frac{\left[\ln(S\widetilde{S}/D^2)\right]^2}{2\sigma^2(T-t)}\right)\right]\right.$$
$$\left.-\,\alpha\left(\frac{S\widetilde{S}}{D^2}\right)^{-\alpha}e^{\alpha^2\sigma^2(T-t)/2}\,\mathrm{erfc}\left(\frac{\alpha}{2}\sqrt{2\sigma^2(T-t)}-\frac{\ln(S\widetilde{S}/D^2)}{\sqrt{2\sigma^2(T-t)}}\right)\right\} \tag{27}$$

for the Green's function to the homogeneous problem corresponding to that in Eqs (9), (10) and (26). The parameters $\alpha$ and $\beta$ are expressed in terms of the parameters $r$ and $\sigma$ from the governing equation in (1) as

$$\alpha=\frac{\sigma^2/2-r}{\sigma^2} \qquad \text{and} \qquad \beta=\left(\frac{\sigma^2/2-r}{\sigma^2}\right)^2$$

and where erfc$(*)$ stays for a special function that is referred to in literature as the complementary Gauss error-function. Subroutines for effective valuation of this function represent in nowadays an inalienable part of every computer software. This makes the expression in (27) readily computable and attractive therefore for practicians working in the field of financial engineering.

For the homogeneous terminal-boundary value problem corresponding to that in Eqs (9) and (10) posed on the semi-infinite strip-shaped region $\Omega=(0<S<D)\times(T>t>-\infty)$ with boudary conditions imposed as

$$|v(0,t)|<\infty \qquad \text{and} \qquad v(D,t)=0$$

the Green's function is obtained in the form

$$G(S,t;\widetilde{S}) = \frac{1}{\widetilde{S}}\left(\frac{S}{\widetilde{S}}\right)^{\alpha}\exp\left(-\beta\frac{\sigma^2}{2}(T-t)\right)$$

$$\times\left\{\frac{1}{\sigma\sqrt{2\pi(T-t)}}\left[\exp\left(-\frac{\left[\ln(S/\widetilde{S})\right]^2}{2\sigma^2(T-t)}\right)-\exp\left(-\frac{\left[\ln(S\widetilde{S}/D^2)\right]^2}{2\sigma^2(T-t)}\right)\right]\right.$$

## Concluding remarks

Compact analytic representations of Green's functions can be obtained for a variety of terminal-boundary value problems for the Black-Scholes equation within the scope of the present study. They are easily accessible for both theoretical analysis and numerical work in the field and can readily be used in solving a variety of problems settings in financial engineering.

## References

[1] F. BLACK and M.S. SCHOLES, *The pricing of options and corporate liabilities*, J. Polit. Econ. 81 (1973) 637-654.

[2] V.I. SMIRNOV, *A Course of Higher Mathematics*, Pergamon Press, Oxford–New York, 1964.

[3] Y.A. MELNIKOV, *Influence Functions and Matrices*, Marcel Dekker, New York–Basel, 1998.

[4] P. WILMOTT, S. HOWISON and J. DEWYNNE, *The Mathematics of Financial Derivatives: A Student Introduction*, Cambridge University Press, 1995.

[5] D. SILVERMAN, *Solution of the Black-Scholes equation using the Green's function of the diffusion equation*, Manuscript, Department of Physics and Astronomy, University of California, Irvine (1999).

[6] S. NEFTCI, *An Introduction to the Mathematics of Financial Derivatives*, Academic Press, New York, 2000.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# $\mathcal{H}$-matrix Preconditioners for Invariant Probability Distribution in Dynamical Systems

**S. Oliveira**[1] **and F. Yang**[1]

[1] *Computer Science Department, The University of Iowa*

emails: `Oliveira@cs.uiowa.edu`, `fayang@cs.uiowa.edu`

### Abstract

An Hierarchical ($\mathcal{H}$)-matrix is a hierarchical sparse data structure. $\mathcal{H}$-matrices can be used to represent full or sparse matrices arising from integral equations or differential equations. The corresponding $\mathcal{H}$-matrix arithmetic based on the $\mathcal{H}$-matrix format defines approximate $\mathcal{H}$-matrix operations with a time complexity almost optimal. Previously we have used these approximations as preconditioners for iterative solvers for diverse applications. In this paper, we apply these $\mathcal{H}$-matrix preconditioners for solving the invariant probability distribution in dynamical systems. The experimental results show that the $\mathcal{H}$-matrix preconditioners are effective to solve the problem.

*Key words: Hierarchical matrices, dynamical systems, invariant probability distribution*

## 1 Introduction

Hierarchical-matrices ($\mathcal{H}$-matrices) and the corresponding hierarchical-matrix arithmetic was introduced and developed in [1, 2, 6, 8]. Since then, the $\mathcal{H}$-matrix approach has been applied to solve the linear systems arising from integral equations, partial differential equations[1, 3, 11, 12], etc.

The difference of $\mathcal{H}$-matrices from the ordinary matrices is that $\mathcal{H}$-matrices use a hierarchical tree structure to represent a hierarchical partitioning of a matrix and each block that is not partitioned further is represented either by a low rank matrix (Rk-matrix) or a full matrix.

The corresponding hierarchical-matrix arithmetic includes operations such as $\mathcal{H}$-matrix addition, $\mathcal{H}$-matrix multiplication, $\mathcal{H}$-matrix inversion, and $\mathcal{H}$-matrix LU factorization, which approximate the corresponding ordinary matrix operations. The hierarchical-matrix arithmetic takes advantage of those low rank blocks in the $\mathcal{H}$-matrix representation and uses approximation to achieve the optimal computation complexity of $O(n \log^a n)$. The fixed-rank $\mathcal{H}$-matrix arithmetic keeps the rank of Rk-matrix blocks below a fixed constant $k$, and the adaptive-rank $\mathcal{H}$-matrix arithmetic adjusts the rank of Rk-matrix blocks to maintain certain accuracy in approximation.

$\mathcal{H}$-matrices are suitable to represent certain full and sparse matrices. For a full matrix, its $\mathcal{H}$-matrix can be constructed based on the underlying geometric information of the problem. Approximation is needed to represent some full blocks as low rank Rk-matrices[6].

For a sparse matrix, algebraic approaches [7, 10] can be used to represent it in the $\mathcal{H}$-matrix format. Algebraic approaches are based on Heavy Edge Matching[9], Nested Dissection[5] or Bisection and they do not need the geometric information of the problems. They can work on the matrix graphs directly and the off diagonal blocks with only zero entries are represented as Rk-matrices with rank 0.

The $\mathcal{H}$-matrix approach provides an approximate but cheap way to perform matrix operations. It can be used to build preconditioners in iterative methods to solve large scale systems of linear equations.

The problem considered in this paper is to compute the invariant probability distribution $p$, in the dynamic system $x_{t+1} = f(x_t)$, where $f$ is the shift function. To get $p$ results in solving a dense system. Instead of solving the dense system directly, we use algebraic $\mathcal{H}$-matrix construction approach to partition and convert the dense matrix into an $\mathcal{H}$-matrix, which needs much less storage. Then $\mathcal{H}$-matrix arithmetic is applied to the $\mathcal{H}$-matrix to obtain the $\mathcal{H}$-matrix-LU factors, which are used as preconditioners in iterative methods. The numerical results show that the $\mathcal{H}$-LU preconditioners are cheap to calculate, yet they speed up the convergence of GMRES greatly.

The paper is organized as follows: Section 2 is an introduction to $\mathcal{H}$-matrices and $\mathcal{H}$-matrix arithmetic; In Section 3 we introduce the dynamic system and describe the approach we used to construct a $\mathcal{H}$-matrix; finally in Section 4 we present the numerical results.

## 2 $\mathcal{H}$-matrix and $\mathcal{H}$-matrix arithmetic

The structure and definition of $\mathcal{H}$-matrices is based on the index cluster tree $T_I$ and the block index cluster tree $T_{I\times I}$.

### 2.1 Index Cluster Tree and Block Cluster Tree

An index cluster tree $T_I$ defines a hierarchical partition tree over an index set $I = (0, \dots, n-1)$. $T_I$ has the following properties: its root is $I$; any node $i \in T_I$ either is a leaf or has children $S(i)$; the children of a node are pairwise disjoint. So the leaves of $T_I$ form a partition over $I$.

A block cluster tree $T_{I\times I}$ is a hierarchical partition tree over the product of index set $I \times I$. $T_{I\times I}$ is constructed based on $T_I$ and the admissibility condition: its root is $I \times I$; if $s \times t$ in $T_{I\times I}$ satisfies the admissibility condition, then it is an Rk-matrix leaf and the corresponding block is represented in the Rk-matrix format; else if $\#s < N_s$ or $\#t < N_s$, it is a full-matrix leaf and the corresponding block is represented in the full matrix format; otherwise $s \times t$ has children on the next level and its children (subblocks) are defined as $S(s \times t) = \{\, i \times j \mid i, j \in T_I \text{ and } i \in S(s),\ j \in S(t) \,\}$.

$N_s$ is a constant used to control the size of the leaf blocks in order to maintain the efficiency of the $\mathcal{H}$-matrix arithmetic. Usually we choose $N_s \in [10, 100]$.

An admissibility condition is used to determine whether a block should be approximated by an Rk-matrix. Different admissibility conditions can be used in the construction of $\mathcal{H}$-matrices. The papers [1, 2, 6] give further details on adapting the admissibility condition to the underlying problem or the cluster tree.

Given $T_{I\times I}$, an $\mathcal{H}$-matrix $H$ can be defined as follows: $H$ shares the same tree structure with $T_{I\times I}$; the data are stored in the leaves; for each Rk-matrix leaf $s \times t \in T_{I\times I}$, its corresponding block $H_{s\times t}$ is a Rk-matrix; and full matrix leaves satisfy that $\#s < N_s$ or $\#t < N_s$.

Fig. 1 shows an example of $T_I$, $T_{I\times I}$ and the corresponding $\mathcal{H}$-matrix.

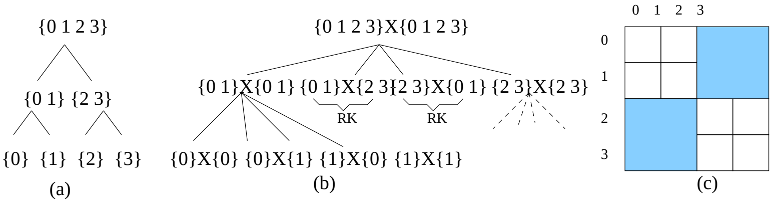

{0 1 2 3}
{0 1} {2 3}
{0} {1} {2} {3}
(a)
{0 1 2 3}X{0 1 2 3}
{0 1}X{0 1} {0 1}X{2 3} {2 3}X{0 1} {2 3}X{2 3}
RK
RK
{0}X{0} {0}X{1} {1}X{0} {1}X{1}
(b)
0 1 2 3
0
1
2
3
(c)

Figure 1: (a) is $T_I$, (b) is $T_{I\times I}$ and (c) is the corresponding $\mathcal{H}$-matrix. The dark blocks in (c) are Rk-matrix blocks and the white blocks are full matrix blocks.

## 2.2 $\mathcal{H}$-matrix Arithmetic

$\mathcal{H}$-matrix Arithmetic is based on the block tree structure and Rk-matrices.

Rk-matrices are the basic building blocks of $\mathcal{H}$-matrices. An $m \times n$ matrix $M$ is called an Rk-matrix if $rank(M) \leq k$ and it is represented in the form of a matrix product $M = AB^T$, where $A$ is $m \times k$ and $B$ is $n \times k$. The storage of a Rk-matrix is of $O(k(m+n))$. The multiplication of a Rk-matrix with another matrix yields a Rk-matrix. But the addition of two Rk-matrix gives a R2k-matrix. Truncated Singular Value Decomposition (SVD) [1, 6] can be used to add two Rk-matrices together and gives an approximate sum which is a Rk-matrix.

The following is an introduction of the operations that are defined in the $\mathcal{H}$-matrix arithmetic[1, 2]. Because of the hierarchical tree structure of $\mathcal{H}$-matrices, these operations are defined recursively, starting from the root of the cluster tree until reaching the leaves. The multiplication of a $\mathcal{H}$-matrix with a vector gives a vector and no approximation is needed. The addition of two $\mathcal{H}$-matrices with the same block cluster gives the approximate sum with the same block tree structure. Truncated SVD is used to add two Rk-matrix block to maintain the low rank.

The multiplication of two $\mathcal{H}$-matrices gives an approximate result with the same tree structure. Hierarchical conversion is used to approximate an $\mathcal{H}$-matrix by a Rk-matrix.

The inversion of an $\mathcal{H}$-matrix is based on Gauss-Jordan elimination, except that the ordinary matrix operations are replaced by the $\mathcal{H}$-matrix operations.

$\mathcal{H}$-LU factorization factors a $\mathcal{H}$-matrix and generates approximate LU factors in the $\mathcal{H}$-matrix format. First a $\mathcal{H}$-matrix triangular solve is defined, which solves an upper or a lower triangular system in the $\mathcal{H}$-matrix format. Then the $\mathcal{H}$-LU factorization of $\mathcal{H}$-matrix

$$\left[\begin{array}{cc} A_{11} & A_{12} \\ A_{21} & A_{22} \end{array}\right] = \left[\begin{array}{cc} L_{11} & 0 \\ L_{21} & L_{22} \end{array}\right] \left[\begin{array}{cc} U_{11} & U_{12} \\ 0 & U_{22} \end{array}\right] \tag{1}$$

is obtained by the following steps:

- first recursively call $\mathcal{H}$-LU factorization applied to $A_{11}$ and we get $L_{11}$ and $U_{11}$;
- second use a $\mathcal{H}$-matrix triangular solver to solve $A_{12} = L_{11}U_{12}$;
- third using a $\mathcal{H}$-matrix triangular solver to solve $A_{21} = L_{21}U_{11}$;
- finally recursively calling $\mathcal{H}$-LU factorization to factor $A_{22} -_{\mathcal{H}} L_{21} *_{\mathcal{H}} U_{12} = L_{22}U_{22}$.

if $A_{11}$ or $A_{22}$ is a full matrix block, then the LU-factorization for full matrices is called. Analogously $\mathcal{H}$-Cholesky factorization can be defined. The computational complexity of the $\mathcal{H}$-matrix arithmetic strongly depends on the structure of $T_{I\times I}$. Under fairly general assumptions on the block cluster tree $T_{I\times I}$ the complexity of $\mathcal{H}$-matrix operations is $O(n \log^{\alpha} n)$ [6, 8].

Mostly $\mathcal{H}$-matrix operations only give approximate results, so we can not use them to solve the problem directly. But they are suitable for constructing preconditioners in iterative methods. The candidate of $\mathcal{H}$-matrix preconditioners are $\mathcal{H}$-matrix inverses, $\mathcal{H}$-LU factors and $\mathcal{H}$-Cholesky factors. Among them, $\mathcal{H}$-inverses are most expensive to compute, so usually we use $\mathcal{H}$-LU factors, or $\mathcal{H}$-Cholesky factors if the matrices are symmetric.

# 3 Model problem

## 3.1 Model problem

The problem is to find the invariant probability distribution $p$ in the following dynamic system:

$$x_{t+1} = f(x_t), x \in [0, 1], f(x) \in [0, 1]. \tag{2}$$

To discretize (2), $[0, 1]$ is divided into $n$ subintervals of equal length $l = 1/n$. In our case $f$ is defined as $f = \alpha x(1 - x)$, where $\alpha$ is a constant. So for each $x \in [x_i, x_{i+1}]$, $f$ maps $x$ to some interval: $f(x) \in [x_j, x_{j+1}]$.

A matrix $A = [a_{ij}]$ can be constructed, where $a_{i,j}$ is the probability that function $f$ maps $x \in [x_j, x_{j+1}]$ to the interval $[x_i, x_{i+1}]$. Matrix $A$ has the following properties: $A$ is sparse and nonsymmetric and $\sum_j a_{ij} = 1$. Fig. 2 shows an example of the distribution of nonzero entries in matrix $A$.

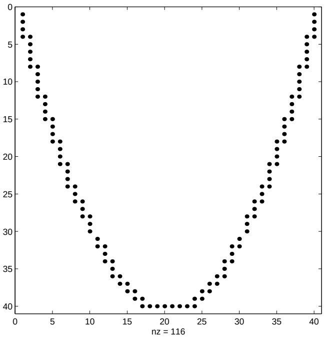


Figure 2: The distribution of nonzero entries in A. The black dots represent nonzero entries.

Each entry $p_i$ in the invariant probability distribution vector $p$ is the probability that $x \in [x_i, x_i + 1]$, and $Ap = p$.

So to get $p$, we need to solve the following system:

$$(A - I)p = 0, \text{ where } e^T p = 1, \tag{3}$$

where $e$ is a vector of 1's.

The system (3) is singular. To avoid solving the singular system, we solve the following nonsingular linear system:

$$(A + ee^T - I)p = e. \tag{4}$$

The system (4) is nonsingular, but it is full. But most entries of $(A + ee^T - I)$ are 1's, which means some of its blocks can be represented exactly in Rk-matrices of rank 1. So we use iterative methods with $\mathcal{H}$-matrix preconditioners to solve the above system.

### 3.2 H-matrix construction

To build $\mathcal{H}$-matrix preconditioners for solving (4), first we need to represent matrix $(A + ee^T - I)$ in the $\mathcal{H}$-matrix format. $(A + ee^T - I)$ is nonsymmetric, so the algebraic $\mathcal{H}$-matrix construction approaches which are based on the matrix graphs can not be applied directly.

We choose the algebraic $\mathcal{H}$-matrix construction based on bisection.

So the process to construct an $\mathcal{H}$-matrix $H$ for (4) based on bisection works in the following way: the root of $H$ is $I \times I$; for each node in $H$ if it corresponds to a block of rank 1, then it is a leaf and the block is represented in Rk-matrix format; otherwise if the number of rows or columns of the block is $\leq N_s$, the node is a leaf and the block is represented in full matrix format; otherwise the block is split into four subblocks of roughly equal size and the node has four children.

The above process can represent $A + ee^T - I$ exactly as an $\mathcal{H}$-matrix. Fig. 3 shows an example of a $\mathcal{H}$-matrix representation of matrix $A + ee^T - I$ with 8 rows and 8 columns. In this example, $Ns = 1$ and the blank blocks represented Rk-matrices of rank 1 and the blocks with dots inside represent full matrix blocks.

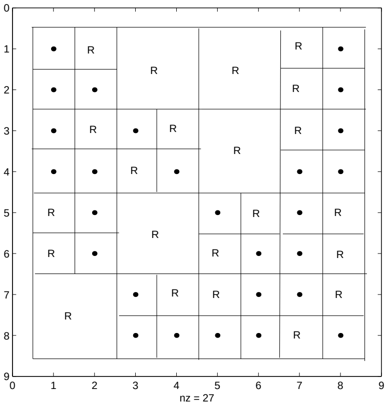


Figure 3: An example of the $\mathcal{H}$-matrix representation of matrix $A + ee^T - I$. Letter R indicate Rk-matrix blocks.

# 4 Experimental Results

In this section, we present the numerical results of applying the $\mathcal{H}$-matrix approach to solve the system (4).

To solve the problem, we first represent the matrix of (4) in the $\mathcal{H}$-matrix format. Then we compute $\mathcal{H}$-LU factors, which are used as the preconditioners for GMRES.

In our experiment, we use the fixed-rank $\mathcal{H}$-matrix arithmetic, since it gives better overall performance than the adaptive $\mathcal{H}$-matrix arithmetic. In the fixed-rank $\mathcal{H}$-matrix arithmetic, we set $k = 4$. That means the rank of the RK-matrix blocks is remained $\leq 4$. We also set the constant $Ns = 40$ to control the size of the leaf block in the $\mathcal{H}$-matrix format.

The size of the problems tested is $1024, 8192, 65536$ and $261344$ respectively. The experiments were carried out on a dual processor computer with 64-bit Athlon 6 4200++ CPUs and 3GB of memory.

To see the computation complexity at each stage in solving the problem, we split the total time that is needed to solve the problem into two parts: the time to compute $\mathcal{H}$-LU preconditioners (set-up time) and the time of GMRES iteration (GMRES iteration time). Fig. 4 shows the set-up time, the time of the GMRES iteration, and the total time.

Based on Fig. 4, the set-up time contributes to a major portion of the total time, compared the time of GMRES iteration. Yet the time of computing $\mathcal{H}$-LU preconditioners increases almost linearly as we increase the size of the problem, even though (4) is a dense system. That means the $\mathcal{H}$-matrix arithmetic is efficient to compute the preconditioners to solve these systems.

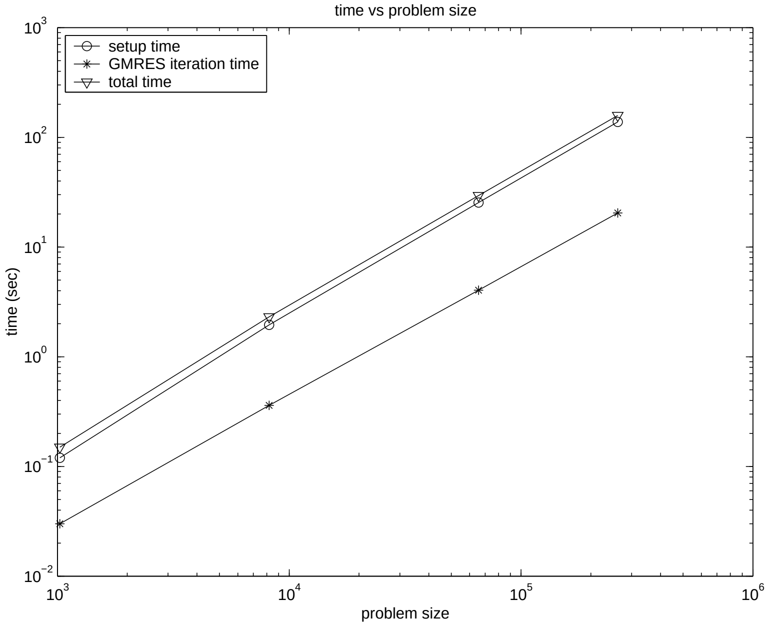


Figure 4: The plot of the setup time, the GMRES iteration time and the total time.

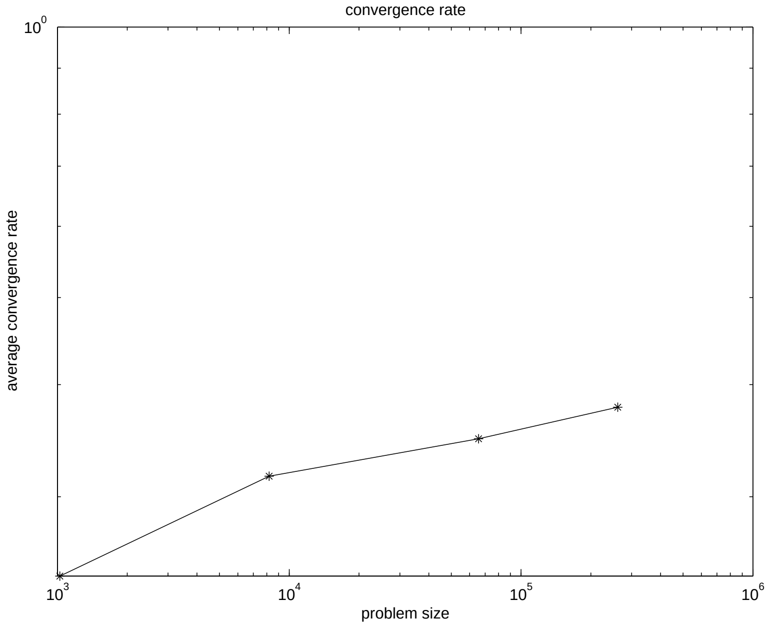


Figure 5: The convergence rate vs. problem size.

Fig. 5 shows the convergence rates. As the size of the problem increases, we can see the convergence rates decrease gracefully.

Based on above results, we can see that $\mathcal{H}$-LU preconditioners are cheap to compute yet they speed up GMRES iterative method greatly.

## References

[1] S. Börm, L. Grasedyck, and W. Hackbusch. Introduction to hierarchical matrices with applications. *Engineering Analysis with Boundary Elements*, 27:405–422, 2003.

[2] S. Börm, L. Grasedyck, and W. Hackbush. Hierarchical matrices. 2005.

[3] S. Le Borne. Hierarchical matrix preconditioners for the Oseen equations. *Comput. Vis. Sci.*, 2006. to appear.

[4] S. Le Borne and S. Oliveira. Joint domain-decomposition $\mathcal{H}$-lu preconditioners for saddle-point problems. *Electronic Transactions on Numerical Analysis*, 2007. to appear.

[5] A. George. Nested dissection of a regular finite element mesh. *SIAM J. Numer. Anal.*, 10:345–363, 1973.

[6] L. Grasedyck and W. Hackbusch. Construction and arithmetics of $\mathcal{H}$-matrices. *Computing*, 70(4):295–334, 2003.

[7] L. Grasedyck, R. Kriemann, and S. Le Borne. Parallel black box domain decomposition based H-LU preconditioning. *Math. Comp*, 2005. submitted.

[8] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. part I: Introduction to $\mathcal{H}$-matrices. *Computing*, 62:89–108, 1999.

[9] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392, 1999.

[10] S. Oliveira and F. Yang. An algebraic approach for $\mathcal{H}$-matrix preconditioners. *Computing*, 2007. to appear.

[11] S. Oliveira and F. Yang. H-matrix preconditioners for saddle-point systems from meshfree discretization. proceeding of ICCES'07, pages 567–574, 2007.

[12] S. Oliveira and F. Yang. Hierarchical preconditioners for parabolic optimal control problems. In *Proceedings of ICCS07*, LNCS. Springer, 2007. To appear.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE2007*
*Chicago, 20-23 June 2007*

# An Alternative to FFT for the Spectral Analysis of Short or Discontinuous Time Series

**Stella Pytharouli, Panos Psimoulis, Elina Kokkinou and Stathis Stiros**
*Department of Civil Engineering, University of Patras*

emails: spitha@upatras.gr, ppsimo@upatras.gr, elilikok@upatras.gr, stiros@upatras.gr

## Abstract

This is a template for the proceedings of CMMSE 2007, with instruction for authors and some sample text. There are some requirements for the spectral analysis using FFT (a minimum length of timeseries, continuous rate of sampling), and this prevents from its use with a number of data in various fields of engineering and natural sciences. To overcome these problems, we present the "Normperiod" Code. This code is based on the Lomb Normalized Periodogram and it permits the analysis of discontinuous or non-equidistant time series without prior interpolations. It is also very simple, it can be easily modified, it can be used in the analysis of both very short and very long time series and it also permits the determination of the statistical significance of obtained results. The efficiency of this code is demonstrated on the basis of two examples: (1) the spectral analysis of a steel bridge vibration record due to a passing train and (2) the spectral analysis of Kremasta Dam reservoir level fluctuations over a period of 37 years.

*Key words: spectral analysis, unevenly spaced data, Lomb periodogram, fortran code*

## 1. Introduction

The most common spectral analysis technique is Fourier Transforms and more specifically the algorithm of Fast Fourier Transforms (FFT; [1]). Since FFT was developed mainly for digital signal processing purposes the algorithm was adjusted to the characteristics of digital signals, i.e. hundreds or even thousands equally-spaced values, a small signal-to-noise ratio etc.

However, in certain cases the requirements of this technique (a certain minimum number of data, constant sampling rate) are not satisfied for various reasons (interruptions in sampling or irregularities in the records to be analyzed are common). Additionally, in numerous cases data are collected non-automatically, at irregular intervals; for instance geodetic, astronomic, geological, volcanological, climatological data etc [2], [3]. This makes necessary an alternative technique for spectral analysis. Such a technique that could easily be applied to non-equidistant data was proposed by [4] about thirty years ago.

In this study we are based on the Lomb algorithm and present the "Normperiod" code, a code that can analyze short and long irregularly spaced time series. This code also permits estimation of the statistical significance of the obtained results. The effectiveness of the code is demonstrated on the basis of two case studies.

## 2. The Lomb Normalized Periodogram

[4] and later [5] developed an algorithm for the spectral analysis of both evenly and unevenly data, as well as of short time series. This algorithm, known as the 'Lomb normalized periodogram (LNP)', is based on the least-squares fitting of a periodic polynomial to the available data and leads to a spectrogram. What is known as "amplitude" in the FFT spectrum is in our case is named "LNP", it corresponds to a normalized amplitude and for a specific period T is defined by

$$P(T)=\frac{1}{2\sigma^2}\left\{\frac{\left[\sum_{j=1}^{N}(x_j-\bar{x})\cos\frac{2\pi(t_j-\tau)}{T}\right]^2}{\sum_{j=1}^{N}\cos^2\frac{2\pi(t_j-\tau)}{T}}+\frac{\left[\sum_{j=1}^{N}(x_j-\bar{x})\sin\frac{2\pi(t_j-\tau)}{T}\right]^2}{\sum_{j=1}^{N}\sin^2\frac{2\pi(t_j-\tau)}{T}}\right\} \tag{1}$$

where the parameter $\tau$ is defined by the equation

$$\tan\left(\frac{4\pi\tau}{T}\right)=\frac{\sum_{j=1}^{N}\sin\left(\frac{4\pi t_j}{T}\right)}{\sum_{j=1}^{N}\cos\left(\frac{4\pi t_j}{T}\right)} \tag{2}$$

and N number of data points

$t_i$ time at which the displacement i was measured

$\bar{x}$ mean of the data values $\bar{x}\equiv\frac{1}{N}\sum_{i=1}^{N}x_i$ (3)

$\sigma^2$ variance of the data values $\sigma^2\equiv\frac{1}{N-1}\sum_{i=1}^{N}(x_i-\bar{x})^2$ (4)

This process is repeated for all values of T, and hence a spectrogram is produced. In addition, the significance level of LNP (i.e. of the normalized amplitude) is defined by equation

$$z_0=-\ln[1-(1-p)^{\frac{1}{N}}] \tag{5}$$

where $z_0$ power level above which the value P(T) of the LNP is statistically significant with (1-p)*100% confidence level.
p significance level
N number of data points.

Two basic advantages of this code are first, that results are rather insensitive to limitations of the Nyquist frequency, if the spectrogram is based on non-equally spaced data. And second, because it is based on least-squares and eq. (1) the LNP values can be produced even with a short times series, since this equation permits a very high redundancy.

## 3. The "Normperiod" code

"Normperiod" code was developed on the basis of the procedure proposed by [4]. It computes (a) the value P(T) of the periodogram using eq. (1) and (b) the value of P(T) for a (1-p)*100% confidence level using eq. (5).

The code is written in Fortran programming language, runs in DOS environment and efforts were made in order to be easily modifiable by potential users.

The INPUT consists of two files written in simple format and the values of four parameters ; the total length N of the time series to be analyzed, the total time interval $t_r$ covered by the time series, the desired confidence level p (i.e. 95%) and the value for the parameter named "fratio", an integer taking values 1, 2, 3.. which defines how many times higher than the Nyquist frequency fc the user desires to compute the periodogram (for more information see [6]).

The total number of frequencies $f_i$ at which the periodogram is determined and is given by the formula [1]

$$N_p = \frac{fratio * 4N}{2} \tag{6}$$

where N is the length of the time series

The interval between two consecutive frequencies for the calculation of the periodogram is equal to

$$\text{fstep} = \frac{fratio * \frac{N}{t_r} - \frac{1}{4t_r}}{N_p} \tag{7}$$

The output file contains the values of the frequency spectrum of the time series. It consists of two columns representing the frequency f and the corresponding dimensionless value P(T) of the LNP (power of spectrum). The format of the output file permits a spectrum plot when exported into a graphic environment. Further details can be found in [6].

The application and effectiveness of the Lomb periodogram is highlighted in the following case studies.

## 4. Case study 1: Spectral Analysis of the RTS monitoring record of a railway bridge

In order to investigate the response of the midspan of a 30m long steel railway bridge in Central Greece under dynamic loads (passing trains) we carried out a number of experiments. Using GPS (Global Positioning System) and RTS (robotic total station) technology, we recorded the movements of a control point located on the middle span of the bridge before, during and after a passing train.

Data used in this study consist of RTS recordings of the vertical movements-response of the bridge to a passing train (Fig.1a).

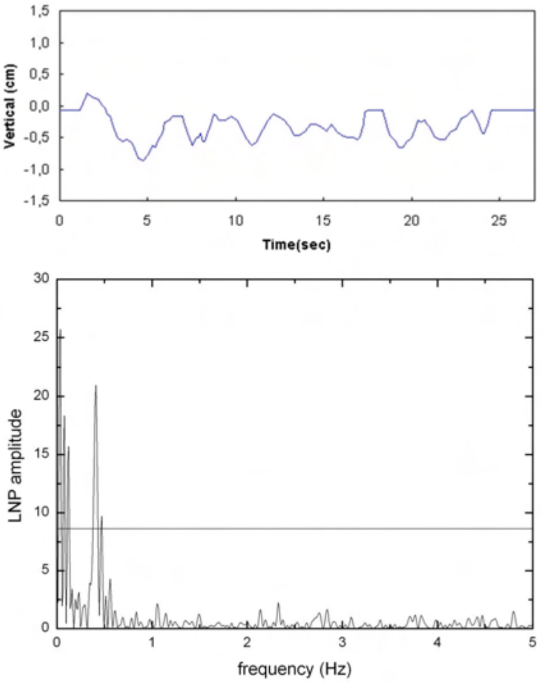


Fig. 1 (a) Vertical displacements recorded by robotic theodolite (RTS) at the deck of a short-span railroad bridge in central Greece. (b) The corresponding frequency spectrum using the "Normperiod" code. Straight line represents the 95% confidence level. A dominant frequency of 0.45Hz is revealed. Frequencies at the left edge of spectrum are not statistically significant (edge effect; [1])

What is evident is that this time series consists of three parts: a first and third part indicating very small amplitude, apparent displacements before and after the passing of the train (i.e. when no displacement of the bridge is expected). These parts of the time series reflect and define the measurement noise and the accuracy level in our data. And a middle section indicating a significant oscillation with duration of several tens of seconds. Analysis of this last section can permit to define the dynamic characteristics of this structure.

Spectral analysis was not possible using FFT, for the available time series was too short (approximately 120 values). Zero padding could be used, but it would lead to biased results, unacceptable for this particular case. Furthermore, data were not sampled at a constant rate, and hence any transformation of the available time series to a new one, based on interpolation techniques would lead to additional noise. For this reason we used the method of the Lomb Normalized Periodogram and the "Normperiod" code. The result of this spectral analysis is shown in Fig. 1b. A dominant frequency equal to 0.45Hz was revealed. This frequency probably corresponds to the interaction between train and bridge deck [8].

## 5. Case study 2: Spectral Analysis of the fluctuations of the Kremasta Dam reservoir level

Dams deform in response to changes in their reservoir load, and the spectral analysis of the fluctuations of the reservoir levels is important to understand dam dynamics [9], [7]. In this study we present the results of the analysis of the Kremasta dam (Greece), one of the major earthen dams in Europe. Available data consisted of the daily values of the reservoir level along a period of 37years. The total length of the reservoir level data set was 13944 values, but this long record contained gaps of about 1.5% of its total length. Spectral analysis was based on LNP and the periodogram was calculated using the "Normperiod" code. Only the peaks that were detected above the 95% confidence level were assumed as statistically significant. Spectral analysis of the reservoir level fluctuations revealed > 10 statistically significant frequencies. The period corresponding to one year representing the annual change of water was prevailing (fig.2b).

Since the percentage of gaps was small, as noticed above, we computed the spectrum of these data using FFT and interpolations. For comparison, both FFT and LNP spectra are shown in Fig.2. Results for both methods were almost the same for main frequencies, certifying the accuracy of "Normperiod" results, but the FFT results were more noisy.

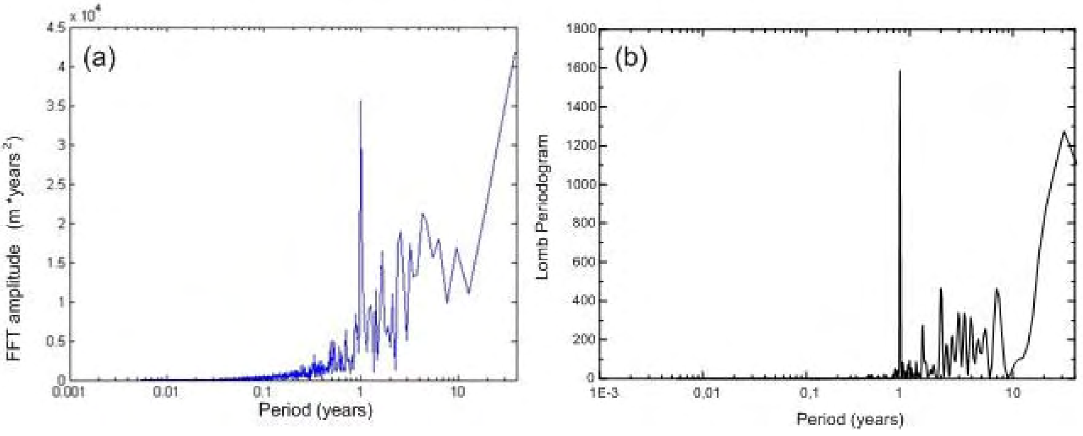


Fig. 2 Spectral analysis of the Kremasta Dam reservoir level values: (a) FFT spectrum. Periods with values between 8 months and 9.5 years were detected, (b) Lomb periodogram. More than 10 significant frequencies were detected with values between the range 1yr – 8.6yrs. In both spectra the period of 1 year is prevailing.

## 6. Conclusions

The case studies presented above indicate that the “Normperiod” code, based on the LNP, has an important merit. It can effectively and easily analyze both short and very long, evenly and non-evenly spaced, as well as discontinuous time series providing information about their spectral characteristics as well as the statistical significance of the results. This new code is available free of charge and can be used in various fields of engineering [10] and natural sciences.

## References

[1] W. H. Press, S. A. Teukolsky, W. T. Vellerling, B. P. Flannery, *Numerical Recipies in C. The Art of Scientific Computing*, Cambridge University Press, 1988.

[2] S. D. Pagiatakis, *Stochastic significance of peaks in the least-squares spectrum*, *J. Geodesy* **73** (1999) 67–78.

[3] P. Hertz and E. Feigelsson, Sample time series in astronomy, in *Proceedings of Applications of time series analysis in Astronomy and Meteorology*, University of Padova, Italy, 6 – 10 September, 1993.

[4] N. R. Lomb, *Least-Squares Frequency Analysis of Unequally Spaced Data*, Astrophys.Space Sci. **39** (1976) 447–462.

[5] J. D. Scargle, *Studies in Astronomical Time Series Analysis. II. Statistical Aspects of Spectral Analysis of Unevenly Spaced Data*, Astrophys. J. **263** (1982) 835–853.

[6] S. I. Pytharouli and S. C. Stiros, *Spectral Analysis of unevenly spaced or discontinuous data using the “Normperiod” code*, Comp. Struc. doi:10.1016/j.compstruc.2007.02.022 (2007).

[7] S. PYTHAROULI, V. KONTOGIANNI, P. PSIMOULIS AND S. STIROS, Kremasta Dam (Greece): Long-term behaviour and performance of the highest earthfill dam in Europe, in *Proceedings of the 5th International Conference on Dam Engineering*, Lisbon, Portugal, 14 – 16 February, (2007) 465-471.

[8] E.M. KOKKINOU, P.A. PSIMOULIS, S.I. PYTHAROULI AND S.C. STIROS, Monitoring of the Gorgopotamos steel railway bridge using a robotic total station (RTS), in Proceedings of the 8th HSTAM International Congress on Mechanics, Patras 12 – 14 July (2007).

[9] S. I. PYTHAROULI AND S. C. STIROS, *Ladon dam (Greece) deformation and reservoir level fluctuations: evidence for a causative relationship from the spectral analysis of a geodetic monitoring record*, Eng. Struc. **27** (2005) 361–370.

[10]P. OMENZETTER AND J. M. W. BROWNJOHN, *Application of Time Series Analysis for Bridge Monitoring*, Smart Materials and Structures. **15(1)** (2006) 129-138.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE2007*
*Chicago, 20-23 June 2007*

# Computing Solutions of Systems of Nonlinear Polynomial Inequalities and Equations

**R. Sharma[1], and O. P. Sha[1]**

[1] *Design Laboratory, Department of Ocean Engineering and Naval Architecture, Indian Institute of Technology, Kharagpur (WB) – 721302, India*

emails: rajivatri@yahoo.com, ops@naval.iitkgp.ernet.in

## Abstract

A fundamental problem in computer aided geometric design (CAGD) and computational science and engineering (CSE) is the efficient computation of all roots of a system of nonlinear polynomial equations in $n$ variables that are contained within an $n$-dimensional box. This paper presents algorithms to bound solution sets of systems of polynomial inequalities and equations using the expansion of a multivariate polynomial into Bernstein polynomials, and this expansion is used to find tight bounds for the range of the polynomial over a given box. The presented techniques to solve such problems rely on expansion and subdivision. Then, the Bernstein expansion, and tight bounds for the range of the polynomial are used to present algorithms for solutions of a system of polynomial inequalities and equations, and to solve constrained convex optimization problems involving polynomials. In order to isolate all of the roots within the given domain, we use the tighter bounds, subdivision, and implement an existence test. The numerical example dealing with the enclosure of the solution set of systems of polynomial inequalities and equations, and the solution of constrained convex optimization problems are presented.

*Key words: Bernstein polynomials; CAD; CAGD; CAM; CSE, constrained convex optimization; geometric modeling; polynomial inequalities; polynomial equations range enclosure; robust stability; solid modelling*
*MSC2000: 65D17, 65H10, and 68U07*

## 1. Introduction

Systems of polynomial inequalities and equations appear in difference branches of science and engineering, e.g. for example, in geometric intersection computations for computer aided geometric design (CAGD), computer aided design (CAD), computer aided manufacturing (CAM); chemical equilibrium problems; chemical combustion and kinematics; and control and robust stability in electrical science, etc., for application area details see Morgan [1]. A fundamental problem in the systems of polynomial inequalities and equations is the efficient computation of all solutions to a system of nonlinear polynomial inequalities and equations within some finite domain. Specifically, this problem arises in many different applications for example, in CAGD/CAD/CAM: it is often necessary to identify all characteristic points of an intersection curve between two surfaces, in order to trace out all of the branches of the curve; if the surfaces are piecewise rational polynomial, we must solve a system of nonlinear polynomial equations in order to identify these points; in feature recognition: computing the medial axis transform requires the determination of branch points, which can frequently be formulated as the solution set of a polynomial system; and in robotic motions: computing distance functions and extremum, in arranging distributive systems in complex mechanical platforms, etc. The methods for the solution of such a system can be classified as methodologies based on elimination theory, continuation, and subdivision. The methods based upon elimination theory for constructing Gröbner bases rely on symbolic computations which are inherently computationally expensive (i.e. for example solution of Problem 1 discussed later in Section 3 by a symbolic computation based method will take about 7200 seconds) for polynomials. Hence, these are unsuitable for larger problems. Additionally, the methods based upon elimination theory and continuation exhaustively computes more information than is really needed. For example they determine all complex solutions of the system though in applications often only the solutions in a given area of interest (i.e. within a box) are desired. The subdivision based methods apply a domain splitting approach that starts with the box of interest, and these algorithms sequentially splits the box of interest it into subboxes eliminating infeasible boxes by using bounds for the range of the polynomials under consideration over each of them and ending up with a union of boxes that contains all solutions to the system which lie within the given box. The methods utilizing this approach shall include interval computation techniques as well as the methods that apply the expansion of a multivariate polynomial into Bernstein polynomials. Theoretically, any interval computation method for solving a system of nonlinear equation (e.g. Hansen [2]; Jaulin et al. [3]; Kearfott [4]; and Neumaier [5]) can be applied to a polynomial system. The techniques specially designed for polynomial systems are more efficient in computing time. Therefore, in the present work we investigate a subdivision based method. Among the existing algorithms, the subdivision methods are used in-practice because of their performance and efficiency. The Interval Projected Polyhedral (IPP) algorithm (Sherbrooke and Patrikalakis [6]; Patrikalakis and Maekawa [7]) is most widely used in-practice. Sherbrooke and Patrikalakis [6], Patrikalakis and Maekawa [7], use Bernstein expansion sequences of bounding boxes for solutions of the nonlinear polynomial systems of equations. They have presented two methods: the first method projects control polyhedra onto a set of coordinate planes and the second exploits linear programming. But, neither have they explored the relationship between the Bernstein coefficients on neighboring subboxes, nor any existence test for a box to contain a solution that might have improved the computational efficiency of the algorithm. Furthermore, they formulate the optimization problem by simply a linear programming problem (LPP). The LPP can be solved efficiently, but in the absence of a functional analysis the solution can not be guaranteed to a global maximum/minimum and only local maxima/minima can be computed.

In this paper, we describe techniques for computing all real solutions to a system of $n$ nonlinear polynomial equations in $n$ unknowns over an $n$-dimensional rectangular domain. We use Bernstein expansion, and incorporate the relationship between the Bernstein coefficients on neighboring subboxes in all the algorithms. Additionally, from 'real analysis' we include an existence test (Neumaier [5]) for a box to contain a solution. As for the optimization problem, we use convex analysis (Rockafellar [8]; Magaril-Ilyaev and Tikhomirov [9]) and treat the optimization problem as a problem of convex optimization (Bertsekas et al. [10]; and Boyd and Vandenberghe [11]) that guarantees the computation of a global maximum/minimum.

The remaining of the paper is organized as follows: Section 2 briefly presents the mathematical preliminaries that are essential to present our work. The theoretical background to enclose the solution set of a system of polynomial inequalities is presented in Section 3. And, also we present the Bernstein expanded and interval subdivision algorithm (BEIS). 4. Section 4 presents the enclosing of the solution set of a system of polynomial equations, and Bernstein expanded projected polyhedron algorithm (BEPP) is presented. The constrained convex optimization is explained in Section 5, and the Bernstein expanded convex optimization (BECO) algorithm is presented. Section 6 concludes the paper by identifying some future applications, and scope of research. Comprehensive theoretical details and a thorough treatment of the results of this paper can be found in Sharma and Sha [12].

## 2. Preliminaries

Following [7], Farin [13], and Garloff [14]; let a polynomial $p$ with real coefficients be given,

$$p(x) = \sum_{i=0}^{l} a_i x^i \tag{1}$$

where $a_i$ is a real number, and $x^i$ is integer powered. It is of interest to know the range of $p$ over an interval $[\mathrm{a},\mathrm{b}]$, i.e.,

$$p([\mathrm{a},\mathrm{b}]) = \{p(x); x \in [\mathrm{a},\mathrm{b}]\}. \tag{2}$$

We can assume without loss of generality that $[\mathrm{a},\mathrm{b}]$ is the 'unit interval $U = [0,1]$'. Now, we present $p$ as a linear combination of the Bernstein polynomials of the same degree,

$$B_i = \binom{l}{i} x^i (1-x)^{l-i}, i = 0,\ldots,l \tag{3}$$

and i.e.,

$$p(x) = \sum_{i=0}^{l} b_i B_i(x). \tag{4}$$

In Equation (4) the coefficients $b_i$ are the 'Bernstein coefficients', and are treated as weighted sums of the coefficients of $p$,

$$b_i = \sum_{j=0}^{i} \left( \frac{\binom{i}{j}}{\binom{l}{j}} \right) a_j, i = 0, \ldots, l. \tag{5}$$

Using a difference scheme derived from de Casteljau algorithm these coefficients are computed efficiently. The bounds for the range of $p$ over $U$ are computed directly from Bernstein coefficients utilizing range enclosing property;

$$p(U) \subseteq \left[ min_{i=0}^{l} b_i, max_{i=0}^{l} b_i \right]. \tag{6}$$

Similarly, we formulate this property by introducing the control points $\begin{pmatrix} \frac{i}{l} \\ b_i \end{pmatrix}, i = 0, \ldots, l$, as the convex hull property,

$$\left\{ \begin{pmatrix} x \\ p(x) \end{pmatrix} : x \in U \right\} \subseteq convx \left\{ \begin{pmatrix} \frac{i}{l} \\ b_i \end{pmatrix} : i = 0, \ldots, l \right\} \tag{7}$$

where $convx S$ denotes the convex hull of $S$ (i.e. convex set that is smallest, and contains $S$). For practical implementation these bounds are improved by subdividing (i.e. the number of divisions is decided with iterative method, and with user specified tolerance) the interval, and the same procedure is applied over the divided subintervals. The important steps in subdivision algorithm are given in Fig. 1. Then $p(U)$ is contained in the union of the individual convex hulls of the control points on the subdivided intervals. The Bernstein coefficients of $p$ on subdivided intervals are calculated from those on $U$ by implementing a recursive arithmetic algorithm (i.e. derived from de Casteljau algorithm). Since, the algorithm is recursive, Bernstein coefficients are computed efficiently. The sequential lower and upper bounds computed in the recursive algorithm converge with quadratic convergence (i.e. to $min\, p(U)$ and $max\, p(U)$). We obtain the Bernstein coefficients $b_i^{'}$ of the derivative of $p$ by simply forming forward difference of its Bernstein coefficients,

$$b_i^{'} = l(b_{i+1} - b_i), i = 0, \ldots, l - 1. \tag{8}$$

For the multivariate case, the Bernstein polynomials are defined as the product of the univariate Bernstein polynomials,

$$B_i(x_1), \ldots, B_n(x_n), \tag{9}$$

where $n$ is the number of variables in the multivariate polynomial. Now, we assign all indices of the univariate case multiindices (i.e. $i = (i_1, \ldots, i_n)^T$, as vectors and where all the $n$ elements are positive integers) in multivariate case. The division of multiindices $i$ and $l$ is computed element wise,

$$\left(\frac{i}{l}\right) := \left(\left(\frac{i_1}{l_1}\right),\dots,\left(\frac{i_n}{l_n}\right)\right)^T . \tag{10}$$

For any $x \in \mathrm{R}^{\mathrm{n}}$, its multipower decomposition is expressed as,

$$x^i := \prod_{\tau=1}^{n} x_\tau^{i_\tau} \tag{11}$$

and for the $n$-fold summation, we use,

$$\sum_{i=0}^{l} := \sum_{i_1=0}^{l_1} ,\dots, \sum_{i_n=0}^{l_n} \tag{12}$$

and define the generalized binomial coefficients by,

$$\binom{l}{i} := \prod_{\tau=1}^{n} \binom{l_\tau}{i_\tau} . \tag{13}$$

i. Start.
Number of subdivisions = $l+1$,
Subdivided intervals = $\left(\left[\mathrm{a},\mathrm{a}+\left(\frac{\mathrm{b}-\mathrm{a}}{l+1}\right)\right],\dots,\left[\left(\mathrm{b}-\left(\frac{\mathrm{b}-\mathrm{a}}{l+1}\right)\right),\mathrm{b}\right]\right)$,
If the root is computed, go to (ii).
If the root is not computed,
Number of subdivisions = $2.(l+1)$,
Subdivided intervals = $\left(\left[\mathrm{a},\mathrm{a}+\left(\frac{\mathrm{b}-\mathrm{a}}{2.(l+1)}\right)\right],\dots,\left[\left(\mathrm{b}-\left(\frac{\mathrm{b}-\mathrm{a}}{2.(l+1)}\right)\right),\mathrm{b}\right]\right)$,
If the root is computed, go to (ii).
If the root is not computed,
Number of subdivisions = $3.(l+1)$,
Subdivided intervals = $\left(\left[\mathrm{a},\mathrm{a}+\left(\frac{\mathrm{b}-\mathrm{a}}{3.(l+1)}\right)\right],\dots,\left[\left(\mathrm{b}-\left(\frac{\mathrm{b}-\mathrm{a}}{3.(l+1)}\right)\right),\mathrm{b}\right]\right)$,
If the root is computed, go to (ii).
If the root is not computed,
Compute the left-most root of the derivative polynomial, and go to (ii).
ii. Stop.

Fig. 1.
The subdivision algorithm for interval.

Now, the set $U$ is a unit box of dimension $n$,

$$u = [0,1]^n . \tag{14}$$

The Bernstein coefficients form an $n$ dimensional array (i.e. patch), and the $n$ variate polynomial $p$ of degree $l = (l_1,\dots,l_n)^T$ is represented as in the form of Equation (1) using transformation of Equation (9). The Bernstein polynomials are given by Equation (2). Furthermore, as in Equation (8), we obtain the Bernstein coefficients of the partial derivatives of $p$ by forming forward differences of the Bernstein coefficients of $p$ in the direction that corresponds to the co-ordinate direction.

## 3. Enclosing the solution set of a system of polynomial inequalities: Bernstein expanded and interval subdivision algorithm

Following Prasolovand Leites [15], and Borwein and Erdelyi [16], let us consider a system of polynomial inequalities, i.e.,

$$p_i(x)\rangle 0, i = 1,\dots,k \text{ and } x \in X \tag{15}$$

where the $n$ variate polynomials $p_i$ and the $n$ dimensional box $X$ are given, and it is desired to find the solution set $\sum$ (i.e. set of vectors $x$ that satisfies Equation (15)) of the sytem. Additionally, to present practical implementation, let a box,

$$Q = \left[\underline{q}_1, \overline{q}^1\right] \times \dots \times \left[\underline{q}_n, \overline{q}^n\right] \tag{16}$$

and a univariate polynomial,

$$p(x,\boldsymbol{q}) = \sum_{i-0}^{m} a_i(\boldsymbol{q}) x^i \tag{17}$$

be given with coefficients dependent on parameters $q_1,\dots,q_n$ and $\boldsymbol{q} = (q_1,\dots,q_n)^T$ , i.e.,

$$\alpha_k(\boldsymbol{q}) = \sum_{i=0}^{l(k)} \alpha_i^{(k)} q^i \, . \tag{18}$$

The parametric vector $\boldsymbol{q}$ is varying inside $Q$. Now the problem is to determine $D$-stablity (i.e. a polynomial is $D$-stable if all its zeros lie inside the prescribed subset $D$ of the complex plane) region of the polynomial (i.e. Equation (17) in the given parametric box $Q$, i.e.,

$$\{\mathbf{q} \in Q : p(x,\mathbf{q}) \neq \varphi, \forall x \notin D\} \tag{19}$$

where $\varphi$ is a null set. Now, we present the Bernstein expanded and interval subdivision (BEIS) algorithm. To start the computation we need a approximation of the solution set $\sum$. For numerical purpose we define inner and outer approximations. The inner approximation ($\sum_I$) is defined as the union of subboxes of $Q$ on which all the polynomials $p_i$ are positively valued. Similarly, the outer approximation ($\sum_O$) is defined as the union of subboxes of $Q$ on which all the polynomials $p_i^*$ are negatively valued. Then, the regions are clipped sharply to generate the boundary ($\delta\sum$). The boundary ($\delta\sum$) is defined by the union of subboxes of $Q$ on which the polynomials $p_i$ are positively valued, but on which at least there is one polynomial that is negative valued. The positivity of a polynomial is checked by the sign of its Bernstein coefficients using the range enclosing property (i.e. the set $\sum_I$ consists of the subboxes on which the Bernstein coefficients of all the polynomials $p_i$ are positive) as given in Equation (6).

The above-mentioned algorithm, and all other algorithms of the present work have been implemented in C++ using its object-oriented features on a Silicon Graphics$^{TM*}$ Origin$^{TM*}$ 200 workstation.

**Problem 1:** From the application of stability criteria (Abdallah et al. [17]), we consider the following system of polynomial inequalities for the three positive parameters $x_1, x_2, x_3$, and the conditions are,

$$x_1.(x_2)^2 \rangle 0 \tag{20}$$

$$-x_1.x_2 + x_1 + (x_3)^2 - x_3 - 1 \rangle 0 \tag{21}$$

$$x_1.x_2 - x_1.x_3 - 2x_1 + (x_3)^3 + 4(x_3)^2 + 4x_3 \rangle 0 \tag{22}$$

$$\begin{aligned} &x_1.(x_2)^3 - x_1.(x_2)^2.x_3 - 4x_1.(x_2)^2 + 2x_1.x_2.x_3 + 4x_1.x_2 + 2x_2.(x_3)^3 + 5x_2.(x_3)^2 + 2x_2.x_3 \\ &(x_3)^3 - 4(x_3)^2 - 4x_3 \rangle 0 \end{aligned} \tag{23}$$

$$x_1.x_2 - 2x_1 - x_2.(x_3)^2 - 4x_2.x_3 - 4x_2 + 2(x_3)^2 + 3x_3 - 2 \rangle 0 \tag{24}$$

with $x_1 \in [90,130]$, $x_2 \in [-1,3]$, and $x_3 \in [0,30]$. In less than a second, the Bernstein expansion provides an inner approximation of the solution set. To visualize the solution set we select $x_1$ = 110. The set of feasible and unfeasible regions (i.e. region bounded by the closed curve is feasible, and the region outside of closed curve is unfeasible) for inner approximations of parameters $x_2$, and $x_3$ for this parametric value (i.e. $x_1$ =110) is shown in Fig. 2.

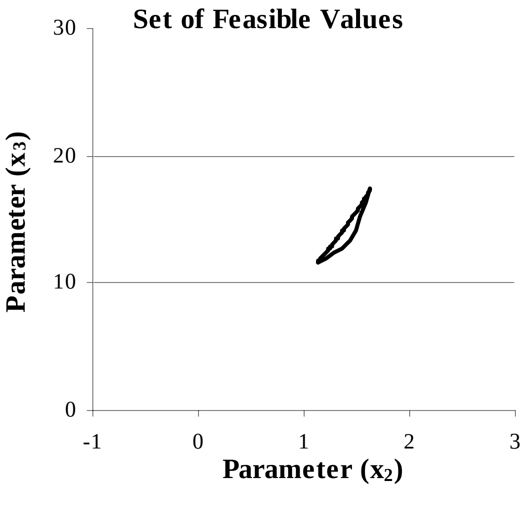


Fig. 2. The set of feasible values for the parameters $x_2$, and $x_3$ for $x_1$= 110.

## 4. Enclosing the solution set of a system of polynomial equations: Bernstein expanded projected polyhedron algorithm

Following [15], let us consider a system of equations, i.e.,

$$p_i(\mathbf{x}) = 0, i = 1,...,k\ ;\ \mathbf{x} \in Q\ ; \text{and } \mathbf{x} = (x, y, ...,), \tag{25}$$

where $x, y, ...,$ depend upon the number of variables, and again the $p_i$ are polynomials in $n$ variables and $Q$ is an $n$-dimensional box. These types of polynomial system of equations are important in many applications areas of engineering sciences (e.g. CAGD - geometric intersection computations; chemical and mechanical kinematics - chemical equilibrium problems, combustion, and kinematics, etc.). Now, we present the Bernstein expanded projected polyhedron (BEPP) algorithm. The important steps in solution algorithm for system of polynomial equations are given in Fig. 3.

**Problem 2:** From the area of CAGD ([6]), we consider the problem of computation of significant points of a planar algebraic curve. This problem in computer aided design deals with the discovery and subsequent tracing of all branches of an implicit algebraic curve $p_i(x, y) = 0$. The computation of '*turning points (i.e. where* $p = \frac{\partial p}{\partial x} = 0$ *or* $p = \frac{\partial p}{\partial y} = 0$*)*' or '*critical points (i.e. where* $\frac{\partial p}{\partial x} = \frac{\partial p}{\partial y} = 0$*)*' is important in solving this problem, Sakkalis and Farouki [18]. Let us find the critical points of,

$$p(x, y) = -64v^4 + 128v^3 - 96u^2.v^2 + 140u.v - 139v^2 + 96u^2.v - 140u.v + 75v - 96u^4 + 276u^3 - 313u^2 + 165u - 36 = 0\ . \tag{26}$$

After the partial differentiation the polynomial reduces to two simultaneous equations which are solved. Now, this problem has nine solutions. All these solutions are computed within a tolerance of $10^{-10}$. The computational results are shown in Table 1. The details and a figure of the curve can be found in [12].

Table 1. Computational results for Problem 2.

| Computational features | |
|---|---|
| Tolerance = $10^{-10}$ | Steps in interval subdivision (ref. Fig. 1) = 3 |
| Number of solutions = 9 | Number of existence tests performed = 30 |
| Number of boxes = 3367 | Computational time (in seconds) = 90 |

## 5. Constrained convex optimization

Following [10], and [11], a constrained optimization problem is defined as,

$$\min_{x \in M} f(x) \tag{27}$$

where the set $M$ of the feasible solutions is given by inequality and equality constraints,

$$g_i(x) \le 0, i = 1,...,n_1, \tag{28}$$

$$h_j(x) = 0, j = 1,...,n_2, \tag{29}$$

$$x \in X\ . \tag{30}$$

Here $D$ is a subset of $\mathrm{R}^n$, $X$ is a box in $D$, and $f$, $g_i$, and $h_j$ are real valued functions within the domain $D$.

i. Start.

a. Implement affine parametric transformation from interval $[\mathrm{a,b}]$ to the 'unit interval $U=[0,1]$'.

b. Transform the basis of functions from '*monomial*' to '*Bernstein*'.

c. Utilize 'the linear precision property' of the 'Bernstein polynomial' as given in Equation (3).

d. Subdivide the interval using algorithm (ref. Fig. 1).

e. Univariate case: Create the graph, and convex hull of the function $p_i(\mathbf{x})=0,i=1,\ldots,k$. Then, the graph will be a Bézier curve,

$$\mathbf{p}(t)=\begin{pmatrix} t \\ p(t) \end{pmatrix}=\sum_{i=0}^{l}\begin{pmatrix} \frac{i}{l} \\ b_i^B \end{pmatrix}B_{i,l}(t)$$ where $0\le t\le 1$, and $\left(\frac{i}{l},b_i^B\right)^T$ are the control points.

The problem of finding roots of the polynomial is treated as a problem of finding the intersection of the convex hull of Bézier curve with the parameter axis.

Test the sign of the polynomials $p_i$ on the subboxes that have been obtained by subdivision (ref. Fig. 1) by using Bernstein expansion. Retain the regions of subboxes with positive signs, discard the regions of subboxes with negative signs, and sharply define the boundary. Discard the subboxes which cannot contain a solution by applying the existence test (i.e. if a univariate continuous function $f$ has a sign change at the endpoints of an interval then this interval contains a zero of $f$; Neumaier, 1990).

Scale the subboxes so that it will become $[0,1]$ using affine parameter transformation and go back to Step 1.

Compute the intersection of the Bézier curve with the parameter axis.

Go to step (ii).

f. Two variables, and multivariate case: Create the graph, and convex hull of the function $p_i^o(\mathbf{x})=0,i=1,\ldots,k$, and $o=1,2,3\ldots,k_V$; where $k_V$ is the number of variables. Then, the individual graph for each variable will be a Bézier surface,

$$\mathbf{p^o}(s,t)=\begin{pmatrix} s \\ t \\ p^o(s,t) \end{pmatrix}=\sum_{i=0}^{l}\begin{pmatrix} \frac{i}{l} \\ \frac{j}{l} \\ b_{i,j}^{B^o} \end{pmatrix}B_{i,l}^o(s)B_{j,l}^o(t)$$ where $0\le s\le 1$, $0\le t\le 1$, and

$\left(\frac{i}{l},\frac{j}{l},b_{i,j}^{B^o}\right)^T$ are the control points.

The problem of finding roots of the polynomial consists of three steps: find the intersection of surfaces with co-ordinate plane, project the control points of $\mathbf{p^o}(s,t)$ onto co-ordinate planes, and for each co-ordinate plane construct the $2D$ convex hulls. Intersection of the convex hull with the horizontal axis is the root.

Test the sign of the polynomials $p_i$ on the subboxes that have been obtained by subdivision (ref. Fig. 1) by using Bernstein expansion. Retain the regions of subboxes with positive signs, discard the regions of subboxes with negative signs, and sharply define the boundary. Discard the subboxes which cannot contain a solution by applying the existence test (i.e. if a univariate continuous function $f$ has a sign change at the endpoints of an interval then this interval contains a zero of $f$; Neumaier, 1990).

Scale the subboxes so that it will become $[0,1]$ using affine parameter transformation and go back to Step 1.

Compute the intersection of the convex hull with the horizontal axis.

ii. Stop.

Fig. 3. The solution algorithm for system of polynomial equations.

The examples of these types of problems can be found in engineering science, e.g., chemical science: pooling and blending; multi-component separation; phase stability analysis; and parametric modeling and simulation; aircraft science: weight allocation in aircraft engines. We are interested in the optimization problems in which the objective and constraint functions are all multivariate polynomials. In engineering sciences it is important to compute the global minimum/maximum rather than only local minima/maxima. In general, the optimization problems are tackled with a purely numerical procedure, and numerical optimization procedure is viewed as a black box. Though, functional analysis of $f(x)$can provide a better insight into the behaviour of $f(x)$, but it is not done in-practice frequently. It can be argued that the optimization problems resulting from practical applications are often very complex with large number variables and constraints, and hence functional analysis may be mathematically complicated. This is really not true. A simple procedure can be explored for functional analysis of $f(x)$ that will avoid mathematical complexities. And, functional analysis of $f(x)$ will reveal whether the $f(x)$ is a convex function, or a concave function or a highly non-linear function. If $f(x)$ is a convex function then the optimization problem of Equations (27) – (30), can be reformulated as a convex optimization problem. And, there are advantages in that, and these are: Basic computational advantages: problem can then be solved, very reliably and efficiently using interior-point methods or other special methods for convex optimization, and these solution methods are reliable enough to be embedded in a computer-aided design or analysis tool, or even a real-time reactive or automatic control system; and theoretical advantages: the associated dual problem, for example, often has an interesting interpretation in terms of the original problem, and sometimes leads to an efficient or distributed method for solving it, [11].

Following, [10], and [11]; an optimization is called a convex optimization problem if the objective and constraint functions are convex, and this means that they satisfy the inequality,

$$g_i(\alpha x+\beta y)\le \alpha g_i(x)+\beta g_i(y) \tag{31}$$

for all $x,y\in \mathrm{R}^n$ and all $\alpha$, $\beta\in \mathrm{R}$ with $\alpha+\beta=1,\alpha\ge 0,\beta\ge 0$. To transform the problem as defined in Equations (27) – (30) to the problem of convex optimization, we use lower and upper bounds of convex functions, [8] and [9]. Now, let us define,

$$f_{LB}(x)\le f(x)\le f_{UB}(x) \tag{32}$$

$$\min_{x\in N} f(x) \tag{33}$$

where $N\subseteq M$, $N\equiv\left(E-E_{S_{f_{LB}(x)}}-E_{S_{f_{UB}(x)}}\right)$, $E=$ domain space of $f(x)$, $E_{S_{f_{LB}(x)}}=$ domain space of $f_{LB}(x)$, and $E_{S_{f_{UB}(x)}}$ = domain space of $f_{UB}(x)$. Now, we present the Bernstein expanded convex optimization (BECO) algorithm. The important steps in solution algorithm for system of polynomial equations are given in Fig. 4.

i. Start.
   - a. Define the optimization problem, $\min_{x \in M} f(x)$, $g_i(x) \leq 0, i = 1, \ldots, n_1$, $h_j(x) = 0, j = 1, \ldots, n_2$, and $x \in X$ .
   - b. Implement affine parametric transformation from given domain intervals $[\mathrm{a}, \mathrm{b}]$ etc., to the 'unit interval $U = [0,1]$'.
   - c. Transform the basis of functions from '*monomial*' to '*Bernstein*'.
   - d. Utilize 'the linear precision property' of the 'Bernstein polynomial' as given in Equation (3).
   - e. Subdivide the domain intervals using algorithm (ref. Fig. 1).
   - f. Univariate case:
     Define $f_{LB}(x) \leq f(x) \leq f_{UB}(x)$, and $\min_{x \in N} f(x)$, where $N \subseteq M$ .
     Define clipped constrained solution domain, $N \equiv \left(E - E_{S_{f_{LB}(x)}} - E_{S_{f_{UB}(x)}}\right)$, $E =$ domain space of $f(x)$, $E_{S_{f_{LB}(x)}}$ = domain space of $f_{LB}(x)$, and $E_{S_{f_{UB}(x)}}$ = domain space of $f_{UB}(x)$.
     Solve the convex optimization problem with convex solution methods (i.e. gradient descent method, infeasible start Newton method, primal-dual interior-point methods). In our implementation we have used primal-dual interior-point methods.
     Go to step (ii).
   - g. Two variable or multivariable case:
     Compute affine lower and upper bound functions as the solution of a linear programming (LP) problem.
     Integrated these bounded functions, and partition the problem into subproblems by subdivision.
     Define clipped constrained solution domain for each subdivided problem, $N \equiv \left(E - E_{S_{f_{LB}(x)}} - E_{S_{f_{UB}(x)}}\right)$, $E =$ domain space of $f(x)$, $E_{S_{f_{LB}(x)}}$ = domain space of $f_{LB}(x)$, and $E_{S_{f_{UB}(x)}}$ = domain space of $f_{UB}(x)$.
     Solve the convex optimization problem with convex solution methods (i.e. gradient descent method, infeasible start Newton method, primal-dual interior-point methods). In our implementation we have used primal-dual interior-point methods.
     Go to step (ii).

ii. Stop.

Fig. 4. The solution algorithm for cconstrained convex optimization problem.

**Problem 3:** From the area of aircraft science (Golinski, [19]), we consider the minimization problem,

$$\min f(x),$$

$$f(x) = 0.7854x_1.(x_2)^2.\left(3.3333(x_3)^2 + 14.9334x_3 - 43.0934\right) - 1.5080x_1.\left((x_6)^2 + (x_7)^2\right)$$

$$7.4770\left((x_6)^3 + (x_7)^3\right) + 0.7854\left(x_4(x_6)^2 + x_5(x_7)^2\right) \tag{34}$$

subjected to eleven constraints that are rewritten in polynomial forms. Additionally, the constrained bounds on the seven variables are given. The computed minimum value of $f(x)$ is at 2994.7852346761. The computational results are shown in Table 2. The details can be found in [12].

**Table 2. Computational results for Problem 3.**

| Computational features | |
|---|---|
| Tolerance = $10^{-10}$ | Steps in interval subdivision (ref. Fig. 1) = 3 |
| Number of solutions = 1 | Number of subproblems as LP created = 1, 20, 000 |
| Computational time(in seconds) = 450 | |

## 6. Conclusions

In this work, we have presented algorithms that are formulated to determine real roots. However, by substitution of $u+iv$ for $x$ in equations and then splitting each equation into real and imaginary parts shall allow the computation of complex roots. Also, we have only considered the so called balanced system of equations, and the algorithms may be extended to systems of $m$ equations in $n$ unknowns ($m \neq n$). These over and under determined systems appear in the computation of singularities on a planar implicit algebraic curve and in engineering design. These problems shall require a constrained formulation or multiple axis projection of polyhedra in the BEPP method. The use of Bernstein expansion requires efficient memory allocation because space required grows exponentially with the number of variables. Now, we have to explore a method that will improve memory allocation for larger number of parameters. We have implemented our algorithms in '*floating point arithmetic (FPA)*' that guarantee only moderate tolerance (i.e. around $10^{-10}$ - $10^{-15}$), and the rounding errors appearing in the calculation cannot be controlled. A implementation in '*interval arithmetic (IA)*' using '*rounded interval arithmetic (RIA)*', ([2], and [3]), shall allow us to control rounding errors to compute '*result within a guaranteed accuracy*' with high tolerance (i.e. around $10^{-15}$ - $10^{-25}$). The detailed analysis of efficiency and complexity of the algorithms has not been addressed in this work. Also, the relative merits of our methods and other existing techniques can be studied for a comparative study. Our future work shall go in this direction, and currently this is under investigation.

## Acknowledgements

This research was supported by the Defense Research and Development Organization (DRDO), MoD, GOI, India; via the grant-in-aid scheme number: ERIP/ER/0400237/M/01. The authors thank Prof. (Retd.) R. P. Gokarn for his assistance and comments.

## Trademarks and copyrights

[*] Trademark and copyright with Silicon Graphics Corporation, USA.

## References

[1] A. P. Morgan, *Solving Polynomial Systems using Continuation for Engineering and Scientific problems*, Prentice Hall, Englewood Cliffs, NJ, USA, (1987).

[2] E. R. Hansen, *Global Optimization Using Interval Analysis*, Pure and Applied Mathematics, Marcel Dekker, USA, (1992).

[3] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*, Springer; 1st edition, Germany, (2004).

[4] R. B. Kearfott, *Rigorous Global Search: Continuous Problems*, Kluwer Academic Publishers, Dordrecht Boston, London, UK, (1996).

[5] A. Neumaier, *Interval Methods for Systems of Equations*, Cambridge University Press, Cambridge, UK, (1990).

[6] E. C. Sherbrooke, and N. M. Patrikalakis, "Computation of the solutions of nonlinear polynomial systems", *Computer Aided Geometric Design*, 10, pp. 379-405, (1993).

[7] N. M. Patrikalakis, and T. Maekawa, *Shape Interrogation for Computer Aided Design and Manufacturing*, Springer-Verlag, Heidelberg, 1st edition, Germany, (2002).

[8] R. T. Rockafellar, *Convex Analysis*, Princeton Landmarks in Mathematics and Physics, Princeton University Press, reprint edition, USA, (1996).

[9] G. G. Magaril-Ilyaev, and V. M. Tikhomirov, *Convex Analysis: Theory and Applications*, Translations of Mathematical Monographs, American Mathematical Society, USA, (2003).

[10] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar, *Convex Analysis and Optimization*, Athena Scientific Publisher, MIT, Cambridge, USA, (2003).

[11] S. Boyd, and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, UK, (2004).

[12] R. Sharma, and O. P. Sha (2006) *Solution of Non-linear Systems of Polynomial Inequalities and Equations*, Design Laboratory Memorandum 06 – 7, Design Laboratory, Department of Ocean Engineering and Naval Architecture, Indian Institute of Technology, Kharagpur (WB) – 721302, India, July 2006, 06(7): 1 – 36, (2006).

[13] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, 5th edition, Morgan Kaufmann, USA, (2001).

[14] J. Garloff, The Bernstein expansion and its applications, *TE – 1*, Fachbereich Informatik, Hochschule Für Technik, Wirtschaft Und Gestaltung Konstanz, Konstanz, Germany, pp. 1-8, (2006).

[15] V. V. Prasolov, and D. Leites, *Polynomials*, Springer, Germany, 1st edition, November 18, (2004).

[16] P. Borwein, and T. Erdelyi, *Polynomials and Polynomial Inequalities*, Springer, Germany, 1st edition, September 27, (1995).

[17] C. Abdallah, P. Dorato, R. Liska, S. Steinberg, and W. Yang, Applications of quantifier elimination theory in control theory, in *Proceedings of 4th IEEE Mediteranean Symposium on Control and Automation*, Maleme, Crete, Greece, pp. 340-345, (1996).

[18] T. Sakkalis, and R. T. Farouki, Singular points of algebraic curves, *Journal of Symbolic Computations*, 9, pp. 405-421, (1990).

[19] J. Golinski, Optimal synthesis problems solved by means of nonlinear programming and random methods, *Journal of Mechanisms*, 5, pp. 287-309, (1970).

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# Balanced Proper Orthogonal Decomposition for Model Reduction of Infinite Dimensional Linear Systems

**John R. Singler**[1] **and Belinda A. Batten**[1]

[1] *Department of Mechanical Engineering, Oregon State University*

emails: `John.Singler@oregonstate.edu`, `bbatten@engr.orst.edu`

### Abstract

In this paper, we extend a method for reduced order model derivation for finite dimensional systems developed by Rowley to infinite dimensional systems. The method is related to standard balanced truncation, but includes aspects of the proper orthogonal decomposition in its computational approach. The method is also applicable to nonlinear systems. The method is applied to a convection diffusion equation.

*Key words: balanced truncation, proper orthogonal decomposition, infinite dimensional systems*

## 1 Introduction and Overview

In this work, we formally extend Rowley's balanced POD algorithm [8] to the infinite dimensional case. The resulting algorithm is a POD-type procedure to design an approximate balanced transformation of an infinite dimensional linear system

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \quad x(0) = x_0, \\ y(t) &= Cx(t), \end{aligned} \tag{1}$$

over a Hilbert space $X$ with inner product $(\cdot,\cdot)$. We assume the linear operator $A : D(A) \subset X \to X$ generates an exponentially stable $C_0$-semigroup $e^{At}$, and the operators $B : U \to X$ and $C : X \to Y$ are bounded and finite rank. We also assume the input and output spaces are finite dimensional; specifically $U = \mathbb{R}^m$ and $Y = \mathbb{R}^p$.

Model reduction via balanced truncation is performed by first determining a balanced realization in which the controllable and observable states of (1) coincide. Then, the balanced model is truncated based on the eigenvalues of the product of the observability and controllability Gramians by eliminating the states corresponding to modes that are difficult to control and observe. Specifically, define the controllability and observability operators $\mathcal{B} : L^2(0,\infty;U) \to X$ and $\mathcal{C} : X \to L^2(0,\infty;Y)$ by

$$\mathcal{B}u = \int_0^\infty e^{At} Bu(t)\, dt, \quad [\mathcal{C}x](t) = Ce^{At}x.$$

The adjoint operators $\mathcal{B}^* : X \to L^2(0,\infty;U)$ and $\mathcal{C}^* : L^2(0,\infty;Y) \to X$ are given by

$$[\mathcal{B}^*x](t) = B^*e^{A^*t}, \quad \mathcal{C}^*y = \int_0^\infty e^{A^*t}C^*y(t)\, dt.$$

The controllability and observability Gramians, $L_B \in \mathcal{L}(X)$ and $L_C \in \mathcal{L}(X)$, are defined by

$$L_B x = \mathcal{B}\mathcal{B}^*x = \int_0^\infty e^{At}BB^*e^{A^*t}x\, dt, \quad L_C x = \mathcal{C}^*\mathcal{C}x = \int_0^\infty e^{A^*t}C^*Ce^{At}x\, dt.$$

The eigenvalues of $L_C L_B \in \mathcal{L}(X)$ are equal to the squares of the singular values of the Hankel operator $\mathcal{H} : L^2(0,\infty;U) \to L^2(0,\infty;Y)$ defined by

$$[\mathcal{H}u](t) = [\mathcal{C}\mathcal{B}u](t) = \int_0^\infty Ce^{A(t+s)}Bu(s)\, ds.$$

An important fact is that that the Hankel singular values are independent of the chosen coordinate system, or system realization.

The coordinate change that balances the system—the balancing transformation—produces observability and controllability Gramians that are equal and diagonal. In the infinite dimensional setting, the Gramians are equal to a diagonal operator on $\ell^2$, the space of square summable sequences; see [5, 7] and the review in [4]. The Hankel singular values are then ordered from greatest to least, and the states corresponding to the "small" singular values are truncated to produce a low order model. This method is rather standard and known in the literature (see, e.g., [9]).

The balanced POD algorithm determines a truncated approximate balancing transformation $T_r : \mathbb{R}^r \to X$ and its left inverse $S_r : X \to \mathbb{R}^r$ (i.e., $S_r T_r = I_r$). To obtain a low order model, approximate the solution $x(t)$ of the linear system (1) by Galerkin projection as

$$x(t) \approx x_r(t) = T_r S_r x(t) = T_r a(t), \quad \text{where} \quad a(t) = S_r x(t). \tag{2}$$

Substituting this approximate solution into the linear system yields the reduced order model

$$\begin{aligned} \dot{a}(t) &= A_r a(t) + B_r u(t), \quad a(0) = a_0, \\ y(t) &= C_r a(t), \end{aligned} \tag{3}$$

where $A_r = S_r A T_r$, $B_r = S_r B$, $C_r = CT_r$, and $a_0 = S_r x_0$.

We may apply this Galerkin projection to obtain low order models of more general, in fact nonlinear, systems. For example, suppose the model takes the form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + F(x(t)) + Bu(t) + Dw(t), \quad x(0) = x_0, \\ y(t) &= Cx(t) + Ew(t), \end{aligned} \tag{4}$$

where $F$ is a nonlinear operator and $w$ is a disturbance. Design the approximate balancing transformation about the linearized system and use the approximation for the solution (2) to obtain the model

$$\begin{aligned} \dot{a}(t) &= A_r a(t) + F_r(a(t)) + B_r u(t) + D_r w(t), \quad a(0) = a_0, \\ y(t) &= C_r a(t) + E_r w(t), \end{aligned} \tag{5}$$

where $A_r$, $B_r$, $C_r$, and $a_0$ are as above, $D_r = S_r D$, $E_r = E$, and $F_r(a) = S_r F(T_r a)$.

# 2 Formal Derivation of the Algorithm

We now give a formal derivation of the balanced POD algorithm for the infinite dimensional setting described above. We do not attempt to rigorously justify the derivation; in some cases we simply proceed by analogy with the finite dimensional case. Convergence analysis of the algorithm is left for future work.

The complete algorithm is presented in Section 3 below. One possible numerical implementation of the algorithm is given in Section 3.1.

## 2.1 Special Forms of the Gramians

One of the main components of the balanced POD algorithm is to compute approximate factors of the Gramians using simulation data. This is possible because of the special form of the Gramians.

Given the specific assumptions regarding the input and output operators, $B$ and $C$, in Section 1, we can write them in the form

$$Bu = \sum_{j=1}^{m} b_j u_j, \quad Cx = [\,(c_1, x),\, \dots\,, (c_p, x)\,]^T,$$

where $u = [\,u_1,\, \dots\,, u_m\,]^T \in U$, and each $b_j$ and $c_j$ are in $X$.

This allows us to rewrite the Gramians. First, define the functions $w_j(t) = e^{At} b_j$, for $j = 1, \dots, m$. Then $w_j$ is the solution of the evolution equation

$$\dot{w}_j(t) = A w_j(t), \quad w_j(0) = b_j.$$

The controllability operator $\mathcal{B} : L^2(0, \infty; U) \to X$ defined above takes the form

$$\mathcal{B}u = \int_0^\infty e^{At} B u(t)\, dt = \int_0^\infty \sum_{j=1}^{m} w_j(t) u_j(t)\, dt,$$

and its adjoint operator $\mathcal{B}^* : X \to L^2(0, \infty; U)$ is easily computed to be

$$[\mathcal{B}^* x](t) = [\,(w_1(t), x),\, \dots\,, (w_m(t), x)\,]^T.$$

Therefore, the controllability Gramian $L_B = \mathcal{B}\mathcal{B}^* \in \mathcal{L}(X)$ is given by

$$L_B x = \int_0^\infty \sum_{j=1}^{m} w_j(t)(w_j(t), x)\, dt.$$

To treat the observability Gramian, we need the adjoint operator $C^* \in \mathcal{L}(Y, X)$ given by

$$C^* y = \sum_{j=1}^{p} c_j y_j,$$

where $y = [\, y_1, \, \dots \, , y_p \,]^T \in Y$. We follow a similar procedure as used for $B$ and define $z_j(t) = e^{A^*t} c_j$, for $j = 1, \dots, p$. Then $z_j$ is the solution of the adjoint equation

$$\dot{z}_j(t) = A^* z_j(t), \quad z_j(0) = c_j.$$

The adjoint of the observability operator $\mathcal{C}^* : L^2(0, \infty; Y) \to X$ takes the form

$$\mathcal{C}^* y = \int_0^\infty e^{A^*t} C^* y(t) \, dt = \int_0^\infty \sum_{j=1}^{p} z_j(t) y_j(t) \, dt$$

and the operator $\mathcal{C} : X \to L^2(0, \infty; Y)$ is given by $[\mathcal{C}x](t) = [\, (z_1(t), x), \, \dots \, , (z_p(t), x) \,]^T$. Therefore, the observability Gramian $L_C = \mathcal{C}^* \mathcal{C} \in \mathcal{L}(X)$ is

$$L_C x = \int_0^\infty \sum_{j=1}^{p} z_j(t)(z_j(t), x) \, dt.$$

## 2.2 The Empirical Gramians

The Gramians can be approximated using time snapshots of the states $w_i(t)$ and $z_i(t)$. Specifically, we approximate the time integrals with the quadratures

$$\begin{aligned} L_B x &= \int_0^\infty \sum_{i=1}^{m} w_i(t)(w_i(t), x) \, dt &\approx \quad L_B^{n_1} x &= \sum_{i=1}^{m} \sum_{j=1}^{n_1} \alpha_j^2 w_i(t_j)(w_i(t_j), x), \\ L_C x &= \int_0^\infty \sum_{i=1}^{p} z_i(t)(z_i(t), x) \, dt &\approx \quad L_C^{n_2} x &= \sum_{i=1}^{p} \sum_{k=1}^{n_2} \beta_k^2 z_i(t_k)(z_i(t_k), x). \end{aligned}$$

Here, $\{\alpha_j^2\}$ and $\{\beta_k^2\}$ are quadrature weights corresponding to the sets of quadrature points $\{t_j\}$ and $\{t_k\}$; different quadrature points and weights can be used for each $w_i$ and $z_i$ if desired. Since $w_i$ are $z_i$ are solutions to linear evolution equations, they are continuous in time and therefore have a well defined value at the quadrature points. The approximate Gramians $L_B^{n_1} \in \mathcal{L}(X)$ and $L_C^{n_2} \in \mathcal{L}(X)$ are called *empirical Gramians.*

Following Rowley in the finite dimensional case, we factor the empirical Gramians. Define "vectors" of weighted snapshots

$$\tilde{w} = [\, \alpha_1 w_1(t_1), \, \dots \, , \alpha_{n_1} w_1(t_{n_1}), \, \dots \, , \alpha_1 w_m(t_1), \, \dots \, , \alpha_{n_1} w_m(t_{n_1}) \,]^T \in X^{N_1}, \quad (6)$$

$$\tilde{z} = [\, \beta_1 z_1(t_1), \, \dots \, , \beta_{n_2} z_1(t_{n_2}), \, \dots \, , \beta_1 z_p(t_1), \, \dots \, , \beta_{n_2} z_p(t_{n_2}) \,]^T \in X^{N_2}, \quad (7)$$

where $N_1 = mn_1$, $N_2 = pn_2$, and $X^q = X \times \cdots \times X$ ($q$ times). These vectors allow the empirical Gramians to be written as $L_B^{n_1} = PP^*$ and $L_C^{n_2} = Q^* Q$, where the operators

$P : \mathbb{R}^{N_1} \to X$ and $Q : X \to \mathbb{R}^{N_2}$ are defined by

$$Pa = \sum_{i=1}^{N_1} a_i \tilde{w}_i, \quad Qx = [\,(\tilde{z}_1, x), \, \ldots \, , (\tilde{z}_{N_2}, x)\,]^T,$$

and their adjoint operators $P^* : X \to \mathbb{R}^{N_1}$ and $Q^* : \mathbb{R}^{N_2} \to X$ are given by

$$P^* x = [\,(\tilde{w}_1, x), \, \ldots \, , (\tilde{w}_{N_1}, x)\,]^T, \quad Q^* a = \sum_{i=1}^{N_2} a_i \tilde{z}_i.$$

Note that $P$ and $Q$ and their adjoints depend on the quadrature points and weights; however, we suppress this dependence for notational simplicity.

### 2.3 The Approximate Balanced Transformation

Recall that the eigenvalues of the product of the Gramians can be used to compute a balancing transformation for the linear system. The balanced system is then truncated to form a reduced order model. We approximate the product of the Gramians $L = L_C L_B$ using the empirical Gramians, i.e., $L \approx L^n = L_C^{n_2} L_B^{n_1}$. Using the above factors, we have $L^n = Q^* Q P P^*$. Following Curtain and Zwart ([6, Lemma 8.2.9, pages 401–402]), it is easy to show that $L^n$ is compact and that the nonzero eigenvalues of $L^n$ are equal to the squares of the nonzero singular values of $QP$.

The operator $QP$ is a bounded linear mapping from $\mathbb{R}^{N_1}$ to $\mathbb{R}^{N_2}$; therefore, it can be represented as an $N_2 \times N_1$ matrix $\Gamma$ with entries $\Gamma_{ij} = (\tilde{z}_i, \tilde{w}_j)$. Let the singular value decomposition of $\Gamma$ be given by

$$\Gamma = U \Sigma V^* = [U_1 \ U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^* \\ V_2^* \end{bmatrix} = U_1 \Sigma_1 V_1^*, \tag{8}$$

where $\Sigma_1 \in \mathbb{R}^{s \times s}$ is diagonal and invertible, $s = \operatorname{rank}(\Gamma)$, $U_1^* U_1 = I_s = V_1^* V_1$, and $I_s$ is the identity matrix in $\mathbb{R}^{s \times s}$.

In the finite dimensional case, Rowley showed that an approximate balancing transformation is given by the operators $T_1 : \mathbb{R}^s \to X$ and $S_1 : X \to \mathbb{R}^s$ defined by

$$T_1 = P V_1 \Sigma_1^{-1/2}, \quad S_1 = \Sigma_1^{-1/2} U_1^* Q.$$

In this paper, we assume the same is true for the infinite dimensional setting and leave theoretical analysis of the algorithm for future work due to size restrictions.

The operators $T_1 : \mathbb{R}^s \to X$ and $S_1 : X \to \mathbb{R}^s$ have the representations

$$T_1 a = \sum_{j=1}^{s} a_j \varphi_j, \quad S_1 x = [\,(\psi_1, x), \, \ldots \, , (\psi_s, x)\,]^T,$$

where the (primary) balanced POD modes $\{\varphi_i\}$ and the adjoint balanced POD modes $\{\psi_i\}$ are given by

$$[\,\varphi_1, \, \ldots \, , \varphi_s\,]^T = \Sigma_1^{-1/2} V_1^* \tilde{w}, \qquad [\,\psi_1, \, \ldots \, , \psi_s\,]^T = \Sigma_1^{-1/2} U_1^* \tilde{z}.$$

As in the finite dimensional case, the primary and adjoint balanced POD modes are biorthogonal, i.e., $(\psi_i, \varphi_j) = \delta_{ij}$. To see this, note $S_1 T_1 a = [\,(\psi_i, \varphi_j)\,] a$ for any $a \in \mathbb{R}^s$. Also, by definition,

$$S_1 T_1 a = \Sigma_1^{-1/2} U_1^* Q P V_1 \Sigma_1^{-1/2} a = I_s a.$$

Thus, $[\,(\psi_i, \varphi_j)\,] = I_s$, or $(\psi_i, \varphi_j) = \delta_{ij}$.

The approximate balancing transformations are truncated by picking $r < s$ and setting

$$T_r a = \sum_{j=1}^{r} a_j \varphi_j, \quad S_r x = [\,(\psi_1, x), \, \ldots \, , (\psi_r, x)\,]^T.$$

Thus, only the first $r$ primary and adjoint balanced POD modes need to be computed. Also, we have $S_r T_r = I_r$, and the modes can be computed by

$$[\,\varphi_1, \, \ldots \, , \varphi_r\,]^T = \Sigma_r^{-1/2} V_r^* \tilde{w}, \qquad [\,\psi_1, \, \ldots \, , \psi_r\,]^T = \Sigma_r^{-1/2} U_r^* \tilde{z}, \tag{9}$$

where $\Sigma_r$, $U_r$, and $V_r$ are appropriate truncations of $\Sigma_1$, $U_1$, and $V_1$.

# 3 The Balanced POD Algorithm

The construction of the operators $T_r$ and $S_r$ as shown above completes the balanced POD algorithm. As outlined in Section 1, we use these transformation to obtain the reduced order model (3). The complete procedure can be summarized as follows:

1. Approximate the solutions $w_j$ of the differential equations

$$\dot{w}_j(t) = A w_j(t), \quad w_j(0) = b_j, \tag{10}$$

   for $j = 1, \ldots, m$, where $Bu = \sum_{j=1}^m b_j u_j$.

2. Approximate the solutions $z_j$ of the adjoint differential equations

$$\dot{z}_j(t) = A^* z_j(t), \quad z_j(0) = c_j, \tag{11}$$

   for $j = 1, \ldots, p$, where $Cx = [\,(c_1, x), \, \ldots \, , (c_p, x)\,]^T$.

3. Form the matrix $\Gamma$, where $\Gamma_{ij} = (\tilde{z}_i, \tilde{w}_j)$, and the weighted snapshot vectors $\tilde{w}$ and $\tilde{z}$ defined in (6) and (7), respectively.

4. Compute the singular value decomposition of $\Gamma$ as in (8), choose $r < \operatorname{rank}(\Gamma)$, and form the first $r$ primary and adjoint balanced POD modes defined in (9):

$$[\,\varphi_1, \, \ldots \, , \varphi_r\,]^T = \Sigma_r^{-1/2} V_r^* \tilde{w}, \qquad [\,\psi_1, \, \ldots \, , \psi_r\,]^T = \Sigma_r^{-1/2} U_r^* \tilde{z},$$

   where $\Sigma_r$, $U_r$, and $V_r$ are appropriate truncations of $\Sigma_1$, $U_1$, and $V_1$.

5. Use the modes to form the matrices in the reduced order model (3):

$$\begin{array}{rcccl} A_r & = & S_r A T_r & = & [\,(A\varphi_j, \psi_i)\,] \in \mathbb{R}^{r \times r}, \\ B_r & = & S_r B & = & [\,(b_j, \psi_i)\,] \in \mathbb{R}^{r \times m}, \\ C_r & = & C T_r & = & [\,(\varphi_j, c_i)\,] \in \mathbb{R}^{p \times r}, \\ a_0 & = & S_r x_0 & = & [\,(x_0, \psi_1), \, \ldots \, , (x_0, \psi_r)\,]^T \in \mathbb{R}^r. \end{array} \tag{12}$$

### 3.1 Finite Dimensional Galerkin Approximations

The algorithm presented above is flexible since we may use any procedure to approximate the solutions $w_i$ and $z_i$ of the linear differential equations (10) and (11). We describe the balanced POD algorithm with Galerkin approximations.

Let $W_1 = \operatorname{span}\{\xi_j\}_{j=1}^{k} \subset D(A)$ and $W_2 = \operatorname{span}\{\eta_j\}_{j=1}^{\ell} \subset D(A^*)$ be finite dimensional subsets of $X$. We compute the solutions of the primary and adjoint differential equations by the finite dimensional Galerkin approximations

$$w_\alpha(t) \approx \sum_{j=1}^{k} r_{j\alpha}(t)\xi_j, \quad z_\beta(t) \approx \sum_{j=1}^{\ell} s_{j\beta}(t)\eta_j,$$

for $\alpha = 1, \ldots, m$ and $\beta = 1, \ldots, p$. Here, $k$ is the same for each $\alpha$ and $\ell$ is the same for each $\beta$; this is not necessary in general, but it does simplify the resulting algorithm. Using these Galerkin approximations, the balanced POD algorithm becomes:

1. Form the $k \times k$ matrices $\tilde{M}_k = [\,(\xi_j, \xi_i)\,]$ and $\tilde{A}_k = [\,(A\xi_j, \xi_i)\,]$. Approximate the Galerkin coefficient vectors $r_\alpha = [\, r_{1\alpha}, \ldots, r_{m\alpha}\,]^T$ by solving the equations
$$\tilde{M}_k \dot{r}_\alpha(t) = \tilde{A}_k r_\alpha(t), \quad \tilde{M}_k r_\alpha(0) = [\,(b_\alpha, \xi_i)\,], \qquad \alpha = 1, \ldots, m. \qquad (13)$$

2. Form the $\ell \times \ell$ matrices $\hat{M}_\ell = [\,(\eta_j, \eta_i)\,]$ and $\hat{A}_\ell = [\,(A^*\eta_j, \eta_i)\,]$. Approximate the Galerkin coefficient vectors $s_\beta = [\, s_{1\beta}, \ldots, s_{p\beta}\,]^T$ by solving the equations
$$\hat{M}_\ell \dot{s}_\beta(t) = \hat{A}_\ell s_\beta(t), \quad \hat{M}_\ell s_\beta(0) = [\,(c_\beta, \eta_i)\,], \qquad \beta = 1, \ldots, p.$$

3. Define the weighted snapshot coefficient matrices $R \in \mathbb{R}^{N_1 \times k}$ and $S \in \mathbb{R}^{N_2 \times \ell}$ by
$$\begin{aligned} R &= [\,\alpha_1 r_1(t_1), \ldots, \alpha_{n_1} r_1(t_{n_1}), \ldots, \alpha_1 r_m(t_1), \ldots, \alpha_{n_1} r_m(t_{n_1})\,]^T, \\ S &= [\,\beta_1 s_1(t_1), \ldots, \beta_{n_2} s_1(t_{n_2}), \ldots, \beta_1 s_p(t_1), \ldots, \beta_{n_2} s_p(t_{n_2})\,]^T. \end{aligned}$$
Then the weighted snapshot vectors $\tilde{w}$ and $\tilde{z}$ defined in (6) and (7), respectively, are approximated by
$$\tilde{w} \approx R[\,\xi_1, \ldots, \xi_k\,]^T, \quad \tilde{z} \approx S[\,\eta_1, \ldots, \eta_\ell\,]^T.$$
Also, the matrix $\Gamma$ is approximated by $\hat{\Gamma} = SNR^T$, where the $\ell \times k$ matrix $N$ is given by $N = [(\eta_i, \xi_j)]$.

4. Compute the singular value decomposition of $\hat{\Gamma}$ as in (8) and choose $r < \operatorname{rank}(\hat{\Gamma})$. Then the first $r$ primary and adjoint balanced POD modes are approximated by
$$\begin{aligned} [\,\varphi_1, \ldots, \varphi_r\,]^T &\approx \Sigma_r^{-1/2} V_r^* R[\,\xi_1, \ldots, \xi_k\,]^T, \\ [\,\psi_1, \ldots, \psi_r\,]^T &\approx \Sigma_r^{-1/2} U_r^* S[\,\eta_1, \ldots, \eta_\ell\,]^T, \end{aligned}$$
where $\Sigma_r$, $U_r$, and $V_r$ are appropriate truncations of $\Sigma_1$, $U_1$, and $V_1$. Let $\Phi = \Sigma_r^{-1/2} V_r^* R \in \mathbb{R}^{r \times k}$ and $\Psi = \Sigma_r^{-1/2} U_r^* S \in \mathbb{R}^{r \times \ell}$. Then for each $i$,
$$\varphi_i \approx \sum_{j=1}^{k} \Phi_{ij}\xi_j, \quad \psi_i \approx \sum_{j=1}^{\ell} \Psi_{ij}\eta_j.$$

5. Substitute the approximate modes into the reduced order model matrices (12):

$$\begin{aligned}
A_r &= [\,(A\varphi_j, \psi_i)\,] \approx \Psi[\,(A\xi_j, \eta_i)\,]\Phi^T, \\
B_r &= [\,(b_j, \psi_i)\,] \approx \Psi[\,(b_j, \eta_i)\,], \\
C_r &= [\,(\varphi_j, c_i)\,] \approx [\,(\xi_j, c_i)\,]\Phi^T, \\
a_0 &= [\,(x_0, \psi_1), \ldots, (x_0, \psi_r)\,]^T \approx \Psi[\,(x_0, \eta_1), \ldots, (x_0, \eta_\ell)\,]^T.
\end{aligned}$$

## 3.2 Comparison to the Finite Dimensional Algorithm

The Galerkin method presented above gives one way to compare the infinite dimensional balanced POD algorithm presented here which we term "balance POD then discretize" with the finite dimensional POD algorithm applied to a discretization of an infinite dimensional system which we call "discretize then balance POD".

In the "discretize then balance POD" approach, one applies the Galerkin method (or some other discretization scheme) to the linear system (1) to obtain the ordinary differential equation system (13) in step 1 above along with the finite dimensional output equation $y_k = \tilde{C}_k r_\alpha$, where $\tilde{C}_k = [\,(\xi_j, c_i)\,]$. Finite dimensional balanced POD is then performed on this system to obtain a reduced order model.

If certain conditions are satisfied, the "balance POD then discretize" approach presented here produces the same reduced order model as the "discretize then balance POD" approach outlined above. It can be checked that the following conditions are sufficient:

- The Galerkin subspaces $W_1$ and $W_2$ must be equal (therefore, $k = \ell$).
- The Galerkin scheme must satisfy $\tilde{A}_k^* = \hat{A}_k$.
- The same quadrature points and weights are used.
- The inner product for the finite dimensional balanced POD must be weighted by the matrix $\tilde{M}_k$, i.e., $(a, b) = a^T \tilde{M}_k b$.

In this case, the matrix $\Phi^T$ is produced by the finite dimensional balanced POD algorithm, and the same reduced order model results from both approaches.

We note that certain problems and numerical schemes may not satisfy the first two conditions above. For example, if the domain of $A$ does not equal the domain of $A^*$, the first condition may be difficult or impossible to satisfy. Also, certain Galerkin schemes may not satisfy the duality property required in the second condition; for an example with a delay equation, see [3]. In these cases, the "discretize then balance" approach may not produce an actual approximate balancing transformation.

# 4 Numerical Results

All numerical results in this section are for the convection diffusion equation

$$\begin{aligned}
w_t(t,x) &= \mu w_{xx}(t,x) - \kappa w_x(t,x) + b(x)u(t), \\
y(t) &= \int_0^1 c(x)w(t,x)\,dx, \\
w(t,0) &= 0, \quad w(t,1) = 0, \quad w(0,x) = w_0(x).
\end{aligned}$$

with $\mu = 0.1$ and $\kappa = 1$. The functions $b(x)$ and $c(x)$ are piecewise constant with $b(x) = 1$ when $0.1 < x < 0.3$, $c(x) = 1$ when $0.6 < x < 0.7$, and both are zero otherwise. The linear operators are defined as

$$Aw = \mu w_{xx} - \kappa w_x,\ D(A) = H^2 \cap H_0^1, \quad A^* w = \mu w_{xx} + \kappa w_x,\ D(A^*) = D(A).$$

The solutions of the primary and dual linear systems were approximated with standard piecewise linear finite elements using equally spaced nodes. The solutions were integrated over $0 \leq t \leq 2$ using Matlab's `ode15s` solver with default error tolerances. The quadrature points were chosen as the time points returned from `ode15s` and the trapezoid rule was used for the quadrature weights. Time refinement was performed by decreasing the error tolerances of the ODE solver.

We compare the results of the balanced POD algorithm with standard balancing computations. We focus on the Hankel singular values and the balancing modes since these are used to construct the reduced order model. For this example problem, the two approaches give identical results when refined until convergence. In the balanced POD computations, spatial refinement was more important for convergence than time refinement. This is not surprising since the solutions of the primary and dual linear systems are not highly variable in time.

In Figure 1, we show the first 20 approximate Hankel singular values for standard balancing and for balanced POD. The methods produce identical results. For each computation, we used 256 equally spaced finite element nodes. The singular values are converged — further refinement in space (and in time for balanced POD) produces little change. The remaining singular values are below machine precision. The first 5 singular values contain over 99.99% of the "energy" or information in the dataset.

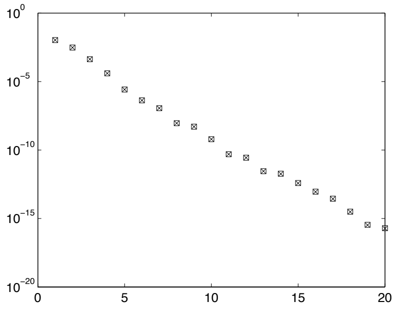


Figure 1: Approximate Hankel singular values for standard balancing (squares) and balanced POD (x).

In Figures 2 and 3, we show primary and adjoint balanced POD modes. All modes are converged and standard balancing produces identical results, as it should for this example. In general, the higher numbered modes are slower to converge under refinement with both standard balancing and balanced POD. For these computations, 128 equally spaced finite element nodes were used.

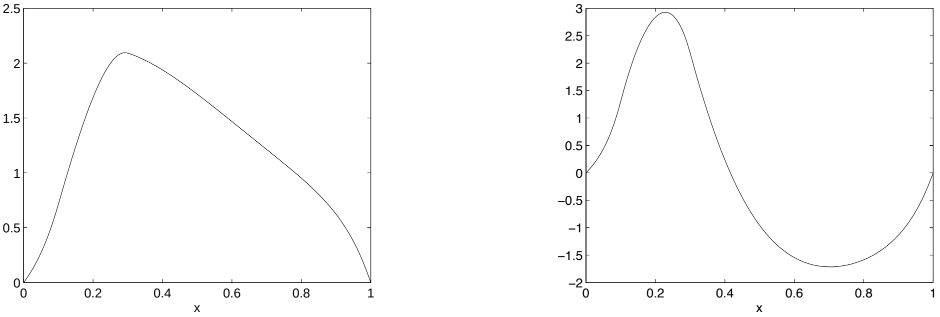


Figure 2: Balanced POD mode 1 (left) and mode 2 (right).

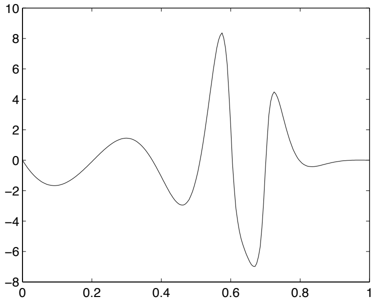


Figure 3: Fifth adjoint balanced POD mode.

# 5 Conclusions and Future Work

In this paper, we extended Rowley's balanced POD algorithm to infinite dimensional systems. In addition, we compared finite and infinite dimensional algorithms and gave conditions when balanced POD "commutes" with discretization. Preliminary numerical results for the convection diffusion equation indicate convergence of the algorithm by comparing the balanced POD with standard balancing computations.

This method shows promise for reduced order model design. In particular, it is computationally tractable for infinite dimensional systems, even if approximating finite dimensional systems have very high dimensions. Additionally, it is applicable even if matrices from approximating systems are not available. One only needs to be able to approximate solutions of standard and dual linear systems. Moreover, there is potential to use error estimators for the solutions of the linear equations to show where to refine to improve accuracy.

We point out, however, that balanced POD may not be feasible for: 1) systems with solutions that decay slowly to zero or are highly oscillatory in time because they may need a large number of time quadrature points, or 2) systems that have a large number of inputs.

In a future paper, we will complete the convergence analysis of this method. In addition, we will compare this approach with balanced truncation methods using large scale matrix Lyapunov solvers (see [1, 2] and the references therein). Even in the case that matrix solvers perform better, balanced POD may still be preferable due to the advantages listed above. Future work includes extending this approach to systems with unbounded input and output operators.

## References

[1] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*, SIAM, Philadelphia, PA, 2005.

[2] P. Benner, V. Mehrmann, and D. C. Sorensen, eds., *Dimension Reduction of Large-Scale Systems*, Springer-Verlag, Berlin, 2005.

[3] J. Borggaard, J. A. Burns, E. Vugrin, and L. Zietsman, *On strong convergence of feedback operators for non-normal distributed parameter systems*, in Proceedings of the IEEE Conference on Decision and Control, vol. 2, 2004, pp. 1526 – 1531.

[4] R. F. Curtain, *Model reduction for control design for distributed parameter systems*, in Research Directions in Distributed Parameter Systems, SIAM, Philadelphia, PA, 2003, pp. 95–121.

[5] R. F. Curtain and K. Glover, *Balanced realisations for infinite-dimensional systems*, in Operator Theory and Systems (Amsterdam, 1985), Birkhäuser, Basel, 1986, pp. 87–104.

[6] R. F. Curtain and H. J. Zwart, *An Introduction to Infinite-Dimensional Linear System Theory*, Springer-Verlag, New York, 1995.

[7] K. Glover, R. F. Curtain, and J. R. Partington, *Realization and approximation of linear infinite-dimensional systems with error bounds*, SIAM Journal on Control and Optimization, 26 (1988), pp. 863–898.

[8] C. W. Rowley, *Model reduction for fluids, using balanced proper orthogonal decomposition*, Internat. J. Bifur. Chaos Appl. Sci. Engrg., 15 (2005), pp. 997–1013.

[9] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*, Prentice-Hall, 1996.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# A Model Chemotaxis System and Its Numerical Solution

**Michael W. Smiley**[1]

[1] *Department of Mathematics, Iowa State University*

emails: `mwsmiley@iastate.edu`

**Abstract**

A system of partial differential equations modeling the attraction of a population of cells to a biochemical concentration in its environment is considered. The system incorporates convective and diffusive effects, either of which may dominate. A numerical method is presented that allows for both possible features, and is monotone and conservative. An efficient implementation of the method in a 2-dimensional setting is described. This includes the use of fast Fourier transform methods to solve the linear systems that arise in the algorithm. *Key words: chemotaxis,convection,diffusion,fast Fourier transform*

## 1 A model chemotaxis system

We consider a system of two equations

$$\begin{aligned} u_t + \nabla\cdot(u\xi(v)\nabla v) &= \nabla\cdot(\kappa\nabla u) + g(u,v,x,y,t),\\ v_t &= \nabla\cdot(\sigma\nabla v) + f(u,v,x,y,t), \end{aligned} \tag{1}$$

which is to hold for $(x,y)\in\Omega,\ t>0$, for a bounded domain $\Omega$, together with Neumann boundary conditions $\partial u/\partial\nu = \partial v/\partial\nu = 0$ on the boundary $\partial\Omega$ of $\Omega$ ($\partial/\partial\nu$ denotes the outer normal derivative along the boundary). In the first equation, $\xi(v)$ is function of $v$ that describes the chemotactic sensitivity of the cells $u$ to a biochemical $v$. Systems of this form were originally proposed in a seminal paper by Keller and Segel [3].

A numerical scheme for solving (1) has been proposed and analyzed in [5]. Assuming $\Omega = (0,1)\times(0,1)$ is a (non-dimensionalized) rectangle in the plane, we set $x_i = i\Delta x,\ i = 0,1,\ldots,M_\alpha$ and $y_j = j\Delta y,\ j = 0,1,\ldots,M_\beta$, where $M_\alpha$, $M_\beta$ are the number of subdivisions in the $x$ and $y$ directions and $\Delta x = 1/M_\alpha$, $\Delta y = 1/M_\beta$. The approach uses control volumes $R_{i,j} = [x_{i-1},x_i]\times[y_{j-1},y_j]$ with cell centers $(\overline{x}_i,\overline{y}_j)$. With $u^{n+1}_{i,j}\approx u(\overline{x}_i,\overline{y}_j,t_{n+1})$ and $v^{n+1}_{i,j}\approx v(\overline{x}_i,\overline{y}_j,t_{n+1})$ the scheme has the general form

$$\begin{aligned} u^{n+1}_{i,j} &= \mathcal{C}_{i,j}(\{v^{n+1}\},\{u^n\}) + \Delta^\kappa_{i,j}\{u^{n+1}\} + \Delta t\, g^{n+1}_{i,j}(u^{n+1}_{i,j},v^{n+1}_{i,j}),\\ v^{n+1}_{i,j} &= v^n_{i,j} + \Delta^\sigma_{i,j}\{v^{n+1}\} + \Delta t\, f^{n+1}_{i,j}(u^{n+1}_{i,j},v^{n+1}_{i,j}), \end{aligned} \tag{2}$$

where $\Delta^{\kappa}_{i,j}\{u^{n+1}\} \approx \nabla\cdot(\kappa\nabla u)$, $\Delta^{\sigma}_{i,j}\{v^{n+1}\} \approx \nabla\cdot(\sigma\nabla v)$ that are standard 5 or 9 point stencils (cf. [2]), $g^{n+1}_{i,j}(u^{n+1}_{i,j}, v^{n+1}_{i,j}) = g(u^{n+1}_{i,j}, v^{n+1}_{i,j}, \overline{x}_i, \overline{y}_j, t_{n+1})$ and similarly $f^{n+1}_{i,j}(u^{n+1}_{i,j}, v^{n+1}_{i,j}) = f(u^{n+1}_{i,j}, v^{n+1}_{i,j}, \overline{x}_i, \overline{y}_j, t_{n+1})$. The term $\mathcal{C}_{i,j}(\{v^{n+1}\},\{u^n\})$ is an approximation of the convective term $\nabla\cdot(u\xi(v)\nabla v)$ based on the use of characteristics. It can be written as a sum $\mathcal{C}_{i,j}(\{v^{n+1}\},\{u^n\}) = \sum_{k,m=-1}^{1} M^{i,j}_{k,m} u^n_{i+k,j+m}$ and is describe more completely below.

Due to the implicit way diffusion is handled and the properties of the multipliers $M^{i,j}_{k,m}$ the scheme preserves positivity of solutions under very mild assumptions on $f$ and $g$. Thus, if $u^n_{i,j} \geq 0$ and $v^n_{i,j} \geq 0$ then $u^{n+1}_{i,j} \geq 0$ and $v^{n+1}_{i,j} \geq 0$. It is also conservative in the sense that in the absence of sources or sinks it conserves mass. These and other properties are treated more completely in [5], where $v$ is considered known *a priori*. The properties just mention extend easily to the present case.

An expanded version of the term $\mathcal{C}_{i,j}(\{v^{n+1}\},\{u^n\}) = \sum_{k,m=-1}^{1} M^{i,j}_{k,m} u^n_{i+k,j+m}$, in conservative form, is given as follows. Assuming $v^{n+1} = \{v^{n+1}_{i,j}\}$ is known, we define values, which are related to the characteristics associated with the left hand side of the first equation in (1), by (see [5] for details)

$$\theta^x_{i,j} = -\frac{\Delta t}{\Delta x}\Big(\frac{\xi(v^{n+1}_{i+1,j}) + \xi(v^{n+1}_{i,j})}{2}\Big)\Big(\frac{v^{n+1}_{i+1,j} - v^{n+1}_{i,j}}{\Delta x}\Big)$$
$$\theta^y_{i,j} = -\frac{\Delta t}{\Delta y}\Big(\frac{\xi(v^{n+1}_{i,j+1}) + \xi(v^{n+1}_{i,j})}{2}\Big)\Big(\frac{v^{n+1}_{i,j+1} - v^{n+1}_{i,j}}{\Delta y}\Big).$$

These formulas are valid at interior points. Special cases in the formulas below that occur along the boundaries of the domain must be accounted for

Assuming these values of $\theta^x_{i,j}$, $\theta^y_{i,j}$ satisfy $|\theta^x_{i,j}| \leq 1$, $|\theta^y_{i,j}| \leq 1$, and

$$1 \geq \max\{0, \theta^x_{i-1,j}\} - \min\{0, \theta^x_{i,j}\}, \quad 1 \geq \max\{0, \theta^y_{i,j-1}\} - \min\{0, \theta^y_{i,j}\},$$

for all $i, j$, we compute in order

$$a_{i,j} = \min\{0, \theta^x_{i,j}\}, \quad A_{i,j} = \max\{0, \theta^x_{i,j}\},$$
$$b_{i,j} = \min\{0, \theta^y_{i,j}\}, \quad B_{i,j} = \max\{0, \theta^y_{i,j}\},$$

$$S^{ab}_{i,j} = a_{i,j}b_{i,j}, \quad S^{aB}_{i,j} = a_{i,j}B_{i,j-1}, \quad S^{Ab}_{i,j} = A_{i-1,j}b_{i,j}, \quad S^{AB}_{i,j} = A_{i-1,j}B_{i,j-1},$$

and

$$\mathcal{E}^{x,n}_{i,j} = a_{i,j}u^n_{i,j} + A_{i,j}u^n_{i+1,j}, \quad \mathcal{E}^{y,n}_{i,j} = b_{i,j}u^n_{i,j} + B_{i,j}u^n_{i,j+1},$$
$$\mathcal{V}^n_{i,j} = S^{ab}_{i,j}u^n_{i,j} + S^{aB}_{i,j+1}u^n_{i,j+1} + S^{Ab}_{i+1,j}u^n_{i+1,j} + S^{AB}_{i+1,j+1}u^n_{i+1,j+1}.$$

Then

$$\mathcal{C}_{i,j}(\{v^{n+1}\},\{u^n\}) = \sum_{k,m=-1}^{1} M^{i,j}_{k,m} u^n_{i+k,j+m}$$
$$= u^n_{i,j} + \mathcal{E}^{x,n}_{i,j} - \mathcal{E}^{x,n}_{i-1,j} + \mathcal{E}^{y,n}_{i,j} - \mathcal{E}^{y,n}_{i,j-1} + \mathcal{V}^n_{i,j} - \mathcal{V}^n_{i,j-1} - \mathcal{V}^n_{i-1,j} + \mathcal{V}^n_{i-1,j-1}$$

This implicitly defines the matrix $\mathcal{C}(\{v^{n+1}\},\{u^n\})$. In addition to its theoretical importance, the conservative form of this term provides efficiencies in its computation.

## 2 Discretizations of Poisson's Equation in 2-D

Since the method (2) is partly implicit, efficient methods are needed to solve the resulting linear systems. These are closely related to finite difference approximations of Poisson's equation, $-\nabla\cdot(\kappa\nabla u) = g$, which are well-known and can be derived in several ways. If homogeneous Neumann boundary conditions are used (as we intend to do) then 0 is an eigenvalue of algebraic multiplicity 1, and the Fredholm alternative applies. This is also true in the discrete version of the problem.

Let $\delta_x u_{i,j} = u_{i+1,j} - u_{i,j}$, $\delta_y u_{i,j} = u_{i,j+1} - u_{i,j}$, and

$$\delta_{xy} u_{i,j} = u_{i+1,j+1} + u_{i,j} - u_{i+1,j} - u_{i,j+1}.$$

In terms of these differences the standard 5-pt stencil is

$$-\Big(\frac{\kappa}{(\Delta x)^2}(\delta_x u_{i,j} - \delta_x u_{i-1,j}) + \frac{\kappa}{(\Delta y)^2}(\delta_y u_{i,j} - \delta_y u_{i,j-1})\Big) = g_{i,j}. \tag{3}$$

and the 9-pt stencil is

$$\begin{aligned} -\Big[\frac{\kappa}{(\Delta x)^2}(\delta_x u_{i,j} - \delta_x u_{i-1,j}) + \frac{\kappa}{(\Delta y)^2}(\delta_y u_{i,j} - \delta_y u_{i,j-1}) + \\ \tfrac{1}{12}\big(\frac{\kappa}{(\Delta x)^2} + \frac{\kappa}{(\Delta y)^2}\big)(\delta_{xy} u_{i,j} - \delta_{xy} u_{i,j-1} - \delta_{xy} u_{i-1,j} + \delta_{xy} u_{i-1,j-1})\Big] = g_{i,j} \end{aligned} \tag{4}$$

The discrete problems associated with these stencils are linear systems which can be presented in different ways (cf. Chapter 6 of [1] and Chapter 4 of [6]). A standard presentation of either system involves a $(N \times N)$ block tridiagonal matrix, with $N = M_\alpha M_\beta$, and $(N \times 1)$ column vectors $u$ and $g$ whose entries are the values of $u(x, y)$ and $g(x, y)$ at the cell centers $(\overline{x}_i, \overline{y}_j)$. We assume these values are stored in the vectors $u, g$ so that the value at $(\overline{x}_i, \overline{y}_j)$ is the $i+(j-1)M_\alpha$ entry of the column vector. We consider two alternatives to the standard presentation, each of which has certain advantages. In one case we use Kronecker products (cf. [6]) and the column vectors $u$ and $g$; in the other case we use matrix versions of these vectors. Using the notational device described in §1.1 of [6], we let $u_{\alpha\times\beta} = [u_{i,j}]$, be the $(M_\alpha \times M_\beta)$ matrix version of $u$, where $u_{i,j} \approx u(\overline{x}_i, \overline{y}_j)$. Similarly $g_{\alpha\times\beta} = [g_{i,j}]$ is the matrix version of the column vector $g$.

The matrices associated with the stencils in either case will be given in terms of matrices $A_\alpha$ and $A_\beta$ that are $(M_\alpha \times M_\alpha)$ and $(M_\beta \times M_\beta)$ versions of the tridiagonal matrix

$$A = \begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}. \tag{5}$$

If $A$ is $(M \times M)$ and $u = [u_j]$, $g = [g_j]$ are $(M \times 1)$ column vectors then $Au = g$ is a cell-centered finite volume approximation of the 1-dimensional Poisson problem

$-u_{xx} = g(x)$, subject to homogeneous Neumann boundary conditions. It's not difficult to see that scheme (3) can be written as

$$\frac{\kappa}{(\Delta x)^2} A_\alpha u_{\alpha\times\beta} + \frac{\kappa}{(\Delta y)^2} u_{\alpha\times\beta} A_\beta = g_{\alpha\times\beta}.$$

Using basic properties of Kronecker products (cf. [6]), and the fact that $A_\beta$ is symmetric, this matrix-matrix equation can be written as the matrix-vector system

$$[\frac{\kappa}{(\Delta x)^2}(I_\beta \otimes A_\alpha) + \frac{\kappa}{(\Delta y)^2}(A_\beta \otimes I_\alpha)]u = g.$$

It somewhat harder to show that scheme (4) can be written in the form

$$\frac{\kappa}{(\Delta x)^2} A_\alpha u_{\alpha\times\beta} + \frac{\kappa}{(\Delta y)^2} u_{\alpha\times\beta} A_\beta - \tfrac{1}{12}(\frac{\kappa}{(\Delta x)^2} + \frac{\kappa}{(\Delta y)^2}) A_\alpha u_{\alpha\times\beta} A_\beta = g_{\alpha\times\beta},$$

Since $w_{\alpha\times\beta} = A_\alpha u_{\alpha\times\beta} A_\beta$ is the matrix-matrix equivalent to of the matrix-vector product $w = (A_\beta \otimes A_\alpha)u$ it follows that (4) can also be written as

$$[\frac{\kappa}{(\Delta x)^2}(I_\beta \otimes A_\alpha) + \frac{\kappa}{(\Delta y)^2}(A_\beta \otimes I_\alpha) - \tfrac{1}{12}(\frac{\kappa}{(\Delta x)^2} + \frac{\kappa}{(\Delta y)^2})(A_\beta \otimes A_\alpha)]u = g.$$

## 3 The connection with the discrete cosine transform

There are several discrete trigonometric transform pairs that can be computed in the framework of fast Fourier transforms. Here we establish a connection between the matrices arising in our descretizations and one of these transform pairs.

Let $A$ be the $(M \times M)$ tridiagonal matrix given in (5). It can be shown that the eigenvalues $\lambda_k$ and eigenvectors $v^k \in \mathbb{R}^M$ for $A$ are

$$\lambda_k = 2(1 - \cos(\tfrac{k\pi}{M})) = 4\sin^2(\tfrac{k\pi}{2M}), \quad v^k = [\cos(\tfrac{(j-\frac{1}{2})k\pi}{M})], \quad k = 0, \ldots, M-1.$$

These vectors are the discrete analogs of the continous eigenfunctions $v(x) = \cos(k\pi x)$. In fact, $v^k_j = \cos(k\pi\overline{x}_j)$. A routine calculation shows that $(v^k)^T v^m = 0$, $k \neq m$, and $(v^k)^T v^k = \frac{1}{2}M$ if $k > 1$ or $M$ if $k = 0$. Thus, if $u \in \mathbb{R}^M$ then

$$u = \tfrac{1}{2}a_0 v^0 + \sum_{k=1}^{M-1} a_k v^k \text{ if and only if } a_k = \tfrac{2}{M}u^T v^k = \tfrac{2}{M}\sum_{j=1}^{M} u_j \cos(\tfrac{(j-\frac{1}{2})k\pi}{M})$$

In this case the components of $u = [u_j]$ are given by

$$u_j = e_j^T u = \tfrac{1}{2}a_0 + \sum_{k=1}^{M-1} a_k v_j^k = \tfrac{1}{2}a_0 + \sum_{k=1}^{M-1} a_k \cos(\tfrac{(j-\frac{1}{2})k\pi}{M})$$

Notice that since $v^0$ is the column vectors of ones, $\frac{1}{2}a_0$ is the mean value of $u$. This is the discrete analog of the continuous case.

The identities above show that $u = [u_j]$ and $a = [a_k]$ are a discrete cosine transform pair. They are a slightly modified version of the *Discrete Cosine Transform II* pair

$$y_k = \sum_{j=1}^{M} x_j \cos(\tfrac{(j-\frac{1}{2})k\pi}{M}) \quad x_j = \tfrac{2}{M}\Big(\tfrac{1}{2}y_0 + \sum_{k=1}^{M-1} y_k \cos(\tfrac{(j-\frac{1}{2})k\pi}{M})\Big)$$

that is consider in [6].

Let $V = [v^0, v^1, \ldots, v^{M-1}]$ be the $(M \times M)$ matrix whose columns are the eigenvectors of $A$, and $\Lambda = \mathrm{diag}(\lambda_0, \ldots, \lambda_{M-1})$ be the corresponding diagonal matrix of eigenvalues, so that $A = V\Lambda V^{-1}$ is the eigen-decomposition of $A$. Also let $\tilde{D}$ be the diagonal matrix defined in terms of it inverse by $\tilde{D}^{-1} = \mathrm{diag}(2, 1, \ldots, 1)$, and set $D^{-1} = \frac{M}{2}\tilde{D}^{-1}$. From the remarks above it follows that

$$V^T V = D^{-1} \quad \text{and} \quad V^T = D^{-1}V^{-1}.$$

Furthermore, the relationships between the *Discrete Cosine Transform II* pair $x$ and $y$ can be written as the matrix vector products

$$y = V^T x \quad \text{and} \quad x = \tfrac{2}{M}V\tilde{D}y = VDy.$$

It follows that the fast trigonometric transforms given as Algorithms 2.4.6 and 2.4.7 in [6] are $2.5M\log_2 M$ algorithms for computing $y = V^T x$ and $x = VDy$ respectively, assuming $M$ is a power of 2.

# 4 Fast Poisson Solvers

As shown in section 2, a discretization of the 2-dimensional Poisson equation, $-\nabla \cdot (\kappa\nabla)u = g$, subject to Neumann boundary conditions, that uses a 5 point stencil, results in a matrix problem of the form

$$\kappa[m_\alpha(I_\beta \otimes A_\alpha) + m_\beta(A_\beta \otimes I_\alpha)]u = g \tag{6}$$

where $A_\alpha$ is an $(M_\alpha \times M_\alpha)$ version of the matrix $A$ defined in (5), $A_\beta$ is an $(M_\beta \times M_\beta)$ version, $m_\alpha = 1/(\Delta x)^2$, $m_\beta = 1/(\Delta y)^2$, and $u, g$ are column vectors of length $N = M_\alpha M_\beta$. In this matrix-vector formulation we have used the fact that $A_\beta$ is symmetric.

Let $V_\alpha$, $\Lambda_\alpha$, $D_\alpha$ (resp. $V_\beta$, $\Lambda_\beta$, $D_\beta$) be the matrices corresponding to $A_\alpha$ (resp. $A_\beta$) as defined in section 3 so that

$$A_\alpha V_\alpha = V_\alpha\Lambda_\alpha, \quad V_\alpha^T V_\alpha = D_\alpha^{-1}, \qquad A_\beta V_\beta = V_\beta\Lambda_\beta, \quad V_\beta^T V_\beta = D_\beta^{-1}.$$

To clarify the indexing we will assume

$$\begin{gathered} V_\alpha = [v_{j,k}^\alpha], \quad v_{j,k}^\alpha = \cos(\tfrac{(j-\frac{1}{2})(k-1)\pi}{M_\alpha}), \quad 1 \le j, k \le M_\alpha, \\ \Lambda_\alpha = \mathrm{diag}(\lambda_1^\alpha, \ldots, \lambda_{M_\alpha}^\alpha), \quad \lambda_k^\alpha = 4\sin^2(\tfrac{(k-1)\pi}{2M_\alpha}), \quad 1 \le k \le M_\alpha. \end{gathered}$$

with the same convention also used for $V_\beta$, $\Lambda_\beta$.

Since $A_\alpha$ and $A_\beta$ are symmetric, it also follows that

$$V_\alpha^T A_\alpha = \Lambda_\alpha V_\alpha^T, \qquad V_\beta^T A_\beta = \Lambda_\beta V_\beta^T,$$

or equivalently

$$V_\alpha^T A_\alpha \big(V_\alpha^T\big)^{-1} = \Lambda_\alpha, \quad V_\beta^T A_\beta \big(V_\beta^T\big)^{-1} = \Lambda_\beta.$$

We use these versions of the eigenvalue decompositions of $A_\alpha$ and $A_\beta$ to obtain forward and inverse discrete cosine transforms, as will be seen below. From these identities and the properties of Kronecker products it follows that

$$\begin{aligned}[V_\beta^T \otimes V_\alpha^T](I_\beta \otimes A_\alpha)[\big(V_\beta^T\big)^{-1} \otimes \big(V_\alpha^T\big)^{-1}] = (I_\beta \otimes \Lambda_\alpha),\\ [V_\beta^T \otimes V_\alpha^T](A_\beta \otimes I_\alpha)[\big(V_\beta^T\big)^{-1} \otimes \big(V_\alpha^T\big)^{-1}] = (\Lambda_\beta \otimes I_\alpha).\end{aligned}$$

Thus the change of variables

$$f = [V_\beta^T \otimes V_\alpha^T]g, \quad w = [V_\beta^T \otimes V_\alpha^T]u$$

results in the diagonal system

$$\kappa[m_\alpha(I_\beta \otimes \Lambda_\alpha) + m_\beta(\Lambda_\beta \otimes I_\alpha)]w = f \tag{7}$$

Using $f_{\alpha\times\beta}$ and $w_{\alpha\times\beta}$, to denote the matrix analogs of the column vectors $f, w$ this system can alternatively be written as

$$\kappa(m_\alpha \Lambda_\alpha w_{\alpha\times\beta} + m_\beta w_{\alpha\times\beta}\Lambda_\beta) = f_{\alpha\times\beta}$$

which is readily solved entry-wise:

$$\big(w_{\alpha\times\beta}\big)_{i,j} = \frac{\big(f_{\alpha\times\beta}\big)_{i,j}}{\kappa(m_\alpha\lambda_i^\alpha + m_\beta\lambda_j^\beta)}.$$

Of course division by zero must be avoided and a strategy for selecting a unique solutions should be employed. Clearly the denominators $m_\alpha\lambda_i^\alpha + m_\beta\lambda_j^\beta$ are the eigenvalues of the matrix $[m_\alpha(I_\beta \otimes A_\alpha) + m_\beta(A_\beta \otimes I_\alpha)]$, and the zero eigenvalue appears when $i = j = 1$. The Fredholm alternative in this case requires $\big(f_{\alpha\times\beta}\big)_{1,1} = 0$ for a solution to exist. Assuming this is the case, there is a 1-parameter family of solutions, $\big(w_{\alpha\times\beta}\big)_{1,1} \in \mathbb{R}$. The choice $\big(w_{\alpha\times\beta}\big)_{1,1} = 0$ picks out the solution with mean value zero.

To see how the matrices $w_{\alpha\times\beta}$, $f_{\alpha\times\beta}$ are related to the matrices $u_{\alpha\times\beta}$, $g_{\alpha\times\beta}$ we use the following property of Kronecker products. Since (assuming $A$ is $(n \times n)$, $B$ is $(m \times m)$) $A \otimes B = (A \otimes I_m)(I_n \otimes B)$, and

$$y = (I_n \otimes B)x \Leftrightarrow y_{m\times n} = Bx_{m\times n}, \quad z = (A \otimes I_m)y \Leftrightarrow z_{m\times n} = y_{m\times n}A^T$$

it follows that

$$z = [A \otimes B]x \Leftrightarrow z_{m\times n} = Bx_{m\times n}A^T.$$

Hence

$$f_{\alpha\times\beta} = V_\alpha^T g_{\alpha\times\beta} V_\beta, \qquad w_{\alpha\times\beta} = V_\alpha^T u_{\alpha\times\beta} V_\beta.$$

Since $V_\alpha^T V_\alpha = D_\alpha^{-1}$ and $V_\beta^T V_\beta = D_\beta^{-1}$, it follows that if $w_{\alpha\times\beta} = V_\alpha^T u_{\alpha\times\beta} V_\beta$ then

$$u_{\alpha\times\beta} = \left(V_\alpha^T\right)^{-1} w_{\alpha\times\beta} V_\beta^{-1} = V_\alpha D_\alpha w_{\alpha\times\beta} D_\beta V_\beta^T.$$

This observation leads to the following algorithm (a fast Poisson solver) for solving (6):

$$\begin{gathered} f_{\alpha\times\beta} = V_\alpha^T g_{\alpha\times\beta} V_\beta, \\ \left(w_{\alpha\times\beta}\right)_{i,j} = \frac{\left(f_{\alpha\times\beta}\right)_{i,j}}{\kappa(m_\alpha\lambda_i^\alpha + m_\beta\lambda_j^\beta)}, \\ u_{\alpha\times\beta} = V_\alpha D_\alpha w_{\alpha\times\beta} D_\beta V_\beta^T. \end{gathered} \tag{8}$$

The first and third steps above are 2-dimensional discrete cosine and discrete inverse cosine transforms, which can be computed by using the fast versions of the corresponding 1-dimensional transforms.

If the finite difference approximation of Poisson's equations uses a 9 point stencil then the matrix-vector equation has the form

$$\kappa[m_\alpha(I_\beta\otimes A_\alpha) + m_\beta(A_\beta\otimes I_\alpha) - \tfrac{1}{12}(m_\alpha + m_\beta)A_\beta\otimes A_\alpha)]u = g \tag{9}$$

Again using a property of Kronecker products we find

$$[V_\beta^T\otimes V_\alpha^T](A_\beta\otimes A_\alpha)[\left(V_\beta^T\right)^{-1}\otimes\left(V_\alpha^T\right)^{-1}] = (\Lambda_\beta\otimes\Lambda_\alpha)$$

Thus the same change of variables that was used above results in the diagonal system

$$\kappa[m_\alpha(I_\beta\otimes\Lambda_\alpha) + m_\beta(\Lambda_\beta\otimes I_\alpha) - \tfrac{1}{12}(m_\alpha + m_\beta)(\Lambda_\beta\otimes\Lambda_\alpha)]w = f,$$

or

$$\kappa\big(m_\alpha\Lambda_\alpha w_{\alpha\times\beta} + m_\beta w_{\alpha\times\beta}\Lambda_\beta - \tfrac{1}{12}(m_\alpha + m_\beta)\Lambda_\alpha w_{\alpha\times\beta}\Lambda_\beta\big) = f_{\alpha\times\beta}.$$

Solving for the components of $w_{\alpha\times\beta}$ we obtain

$$\left(w_{\alpha\times\beta}\right)_{i,j} = \frac{\left(f_{\alpha\times\beta}\right)_{i,j}}{\kappa\big(m_\alpha\lambda_i^\alpha + m_\beta\lambda_j^\beta - \tfrac{1}{12}(m_\alpha + m_\beta)\lambda_i^\alpha\lambda_j^\beta\big)}.$$

Implicitly this shows that the eigenvalues of the discrete system (9) are

$$\kappa\big(m_\alpha\lambda_i^\alpha + m_\beta\lambda_j^\beta - \tfrac{1}{12}(m_\alpha + m_\beta)\lambda_i^\alpha\lambda_j^\beta\big).$$

As with the 5 point stencil, 0 is an eigenvalue of algebraic multiplicity 1. The same remarks that were made above apply also to this case. We note that all other eigenvalues are positive. This can be shown by considering the function

$$h(x,y) = ax + by - \tfrac{1}{12}(a+b)xy, \qquad (a,b>0),$$

and finding extreme values of the square $[0,4]\times[0,4]$. Clearly $\lambda_i^\alpha, \lambda_j^\beta \in [0,4]$.

# 5 Solution of the Method Equations

Our goal in this section is to describe an efficient algorithm for solving the approximate chemotaxis system (2). Throughout we use $u, v, h, \dots$ to denote column vectors and $u_{\alpha\times\beta}, v_{\alpha\times\beta}, h_{\alpha\times\beta}, \dots$ to denote the corresponding matrix representations of these column vectors as in previous sections

## 5.1 A special case

If $f(u, v, x, y, t)$ is actually independent of $u$ then the system (2) can be solved sequentially, first for $v^{n+1}$ and then for $u^{n+1}$:

$$\begin{gathered} v_{i,j}^{n+1} - \Delta_{i,j}^{\sigma}\{v^{n+1}\} - \Delta t\, f_{i,j}^{n+1}(v_{i,j}^{n+1}) = v_{i,j}^{n} \\ u_{i,j}^{n+1} - \Delta_{i,j}^{\kappa}\{u^{n+1}\} - \Delta t\, g_{i,j}^{n+1}(u_{i,j}^{n+1}, v_{i,j}^{n+1}) = \mathcal{C}_{i,j}(\{v_{i,j}^{n+1}\}, \{u_{i,j}^{n}\}) \end{gathered}$$

In these equations the approximate Laplacians may be given in terms of 5-pt or 9-pt stencils. In the linear case, $f = f(x, y, t)$, the system for $v^{n+1}$ becomes a simple modification of the discrete Poisson equation

$$v_{i,j}^{n+1} - \Delta_{i,j}^{\sigma}\{v^{n+1}\} = h_{i,j}$$

where $h_{i,j} = v_{i,j}^{n} + \Delta t\, f_{i,j}^{n+1}$ is an array of known values. Thus a matrix-vector equation of the form (in the case of a 5-pt stencil)

$$v + \sigma\Delta t[m_\alpha(I_\beta \otimes A_\alpha) + m_\beta(A_\beta \otimes I_\alpha)]v = h \tag{10}$$

must be solved for $v = v^{n+1}$. This system can be solved by essentially the same algorithm as given in (8). The only modification is that the eigenvalues used in the second step are now $1 + \sigma\Delta t(m_\alpha\lambda_i^\alpha + m_\beta\lambda_j^\beta)$.

In the nonlinear case we will need to solve a system of the form

$$v + \sigma\Delta t[m_\alpha(I_\beta \otimes A_\alpha) + m_\beta(A_\beta \otimes I_\alpha)]v - \Delta t\, f(v) = h.$$

where now the matrix representation of $h$ is $h_{\alpha\times\beta} = [v_{i,j}^{n}]$ and $f(v)$ is the column vector whose associated matrix is $f(v)_{\alpha\times\beta} = [f(v_{i,j}^{n+1}, \overline{x}_i, \overline{y}_j, t_{n+1})]$.

Let

$$\begin{gathered} \mathcal{M}^\sigma = I + \sigma\Delta t[m_\alpha(I_\beta \otimes A_\alpha) + m_\beta(A_\beta \otimes I_\alpha)], \text{ or} \\ \mathcal{M}^\sigma = I + \sigma\Delta t[m_\alpha(I_\beta \otimes A_\alpha) + m_\beta(A_\beta \otimes I_\alpha) - \tfrac{1}{12}(m_\alpha + m_\beta)A_\beta \otimes A_\alpha)], \end{gathered}$$

depending on whether a 5 or 9 point stencil is used. Clearly the solution of the equation $\mathcal{M}^\sigma v = h$ can be obtained via the fast Poisson solver process as described above for (10). In the present case we need to solve $\mathcal{M}^\sigma v - \Delta t\, f(v) = h$ and Newton's method can be used for this purpose. It is readily seen that this problem fits into the standard framework for which local convergence can be proven (cf. [4]).

Define

$$\mathcal{F}(v) = \mathcal{M}^\sigma v - \Delta t\, f(v) - h,$$

so that the system we need to solve is $\mathcal{F}(v) = 0$. It easy to see that

$$\mathcal{F}'(v) = \mathcal{M}^\sigma - \Delta t\, \mathcal{D}(v)$$

where $\mathcal{D} = \mathrm{diag}(d_1, \dots, d_N)$, $N = M_\alpha M_\beta$, is a diagonal matrix whose vector of diagonal entries $d$ has the equivalent matrix representation

$$d_{\alpha\times\beta} = [\frac{\partial f}{\partial v}(v_{i,j}, \overline{x}_i, \overline{y}_j, t_{n+1})].$$

Thus applying Newton's method to this problem we obtain

$$\mathcal{M}^\sigma \delta v - \Delta t\, \mathcal{D}(v^{(k)})\delta v = \mathcal{F}'(v^k)\delta v = \mathcal{F}(v^{(k)}) = \mathcal{M}^\sigma v^k - \Delta t\, f(v^k) - h,$$
$$v^{(k+1)} = v^{(k)} - \delta v.$$

Since the change of variables used to diagonalize $\mathcal{M}^\sigma$ in general does not preserve the diagonal structure of $\mathcal{D}(v^{(k)})$ an efficient solution of the linear system for the Newton increment $\delta v$ cannot proceed along these lines. However an iterative approach can be used that makes use of the efficiency of a fast Poisson solver. (Alternatively, an inexact Newton method could be used.) Consider the iterative scheme

$$\mathcal{M}^\sigma (\delta v)^{(p+1)} - \Delta t\, \mathcal{D}(v^{(k)})(\delta v)^{(p)} = \mathcal{M}^\sigma v^k - \Delta t\, f(v^k) - h. \tag{11}$$

Clearly the iteration matrix for this scheme is $\Delta t\, (\mathcal{M}^\sigma)^{-1}\mathcal{D}(v^{(k)})$. Since the eigenvalues of $\mathcal{M}^\sigma$ are all bounded below by 1, with equality for one of the eigenvalues, the spectral radius of $(\mathcal{M}^\sigma)^{-1}$ is $\rho((\mathcal{M}^\sigma)^{-1}) = 1$. And, since $\mathcal{D}(v^{(k)})$ is diagonal,

$$\rho(\mathcal{D}(v^{(k)})) = \max\{|d_1|, \dots, |d_N|\} = \max_{i,j} \left|\frac{\partial f}{\partial v}(v_{i,j}^{(k)}, \overline{x}_i, \overline{y}_j, t_{n+1})\right| \le \|f_v\|_\infty.$$

Hence the iterative scheme (11) will convergence if $\Delta t < 1/\|f_v\|_\infty$. An efficient scheme that can be used in computing the iterates $(\delta v)^{(p)}$ is described in section **??** below.

### 5.2 The general case

In the general scheme (2) cannot be solved sequentially since the values $\{v_{i,j}^{n+1}\}$ will depend on $\{u_{i,j}^{n+1}\}$. We suggest a iterative scheme that can be used to solve this coupled system. Let $\mathcal{M}^\sigma$ be defined as before and set

$$\mathcal{M}^\kappa = I + \kappa\Delta t[m_\alpha(I_\beta \otimes A_\alpha) + m_\beta(A_\beta \otimes I_\alpha)], \text{ or}$$
$$\mathcal{M}^\kappa = I + \kappa\Delta t[m_\alpha(I_\beta \otimes A_\alpha) + m_\beta(A_\beta \otimes I_\alpha) - \tfrac{1}{12}(m_\alpha + m_\beta)A_\beta \otimes A_\alpha)],$$

depending on whether a 5 or 9 point stencil is used. Set

$$\mathcal{F}(u, v) = \mathcal{M}^\sigma v - \Delta t\, f(u, v) - h,$$

where now $f(u, v)_{\alpha\times\beta} = [f(u_{i,j}, v_{i,j}, \overline{x}_i, \overline{y}_j, t_{n+1})]$, and again $h_{\alpha\times\beta} = [v_{i,j}^n]$. Next we set

$$\mathcal{G}(u, v) = \mathcal{M}^\kappa u - \Delta t\, g(u, v) - \mathcal{C}(v),$$

where $g(u,v)_{\alpha\times\beta} = [g(u_{i,j}, v_{i,j}, \overline{x}_i, \overline{y}_j, t_{n+1})]$, and $\mathcal{C}(v)_{\alpha\times\beta} = [\mathcal{C}_{i,j}(\{v\},\{u^n\})]$. The approximations $u^{n+1}, v^{n+1}$ at the next time step are solutions of the couple nonlinear system $\mathcal{F}(u,v) = 0$, $\mathcal{G}(u,v) = 0$. An iterative scheme that can be used to solve this system is: $u^{(0)} = u^n$, and for $k = 0, 1, \ldots$

$$\mathcal{F}(u^{(k)}, v^{(k+1)}) = 0, \quad \mathcal{G}(u^{(k+1)}, v^{(k+1)}) = 0. \tag{12}$$

Solving the first equation can be done in essentially the same way as for the special system considered in the previous subsection, and once $v^{(k+1)}$ has been determined the same is true of the second. Convergence of this iteration scheme is discussed below.

### 5.3 Convergence of the Iteration Scheme

In this subsection we show that the iteration scheme (12), which is a nonlinear version of Gauss-Seidel iteration, converges under rather standard conditions on the time step $\Delta t$ and the Lipschitz's constants for the nonlinear terms. We assume there are constants $L_f$ and $L_g$ such that the real-valued functions $f$, $g$ satisfy

$$\begin{aligned} |f(u_1,v_1,x,y,t) - f(u_2,v_2,x,y,t)| &\le L_f(|u_1-u_2| + |v_1-v_2|), \\ |g(u_1,v_1,x,y,t) - g(u_2,v_2,x,y,t)| &\le L_g(|u_1-u_2| + |v_1-v_2|), \end{aligned}$$

for all $u_1, u_2, v_1, v_2 \in [0,\infty)$, $(x,y) \in \Omega$ and $t \ge 0$.

Consider two successive iterates $v^{(k)}$ and $v^{(k+1)}$ which satisfy $\mathcal{F}(u^{(k-1)}, v^{(k)}) = 0$ and $\mathcal{F}(u^{(k)}, v^{(k+1)}) = 0$, respectively, or equivalently

$$\begin{aligned} \mathcal{M}^\sigma v^{(k)} &= \Delta t\, f(u^{(k-1)}, v^{(k)}) + h \\ \mathcal{M}^\sigma v^{(k+1)} &= \Delta t\, f(u^{(k)}, v^{(k+1)}) + h. \end{aligned}$$

Subtracting and using the Lipschitz continuity of $f$ gives

$$\|v^{(k+1)} - v^{(k)}\| \le \Delta t\, L_f \|\, (\mathcal{M}^\sigma)^{-1} \,\| \Big( \|u^{(k)} - u^{(k-1)}\| + \|v^{(k+1)} - v^{(k)}\| \Big).$$

Thus, with $K_f^\sigma = L_f \|\, (\mathcal{M}^\sigma)^{-1} \,\|$ we find

$$\|v^{(k+1)} - v^{(k)}\| \le \frac{\Delta t\, K_f^\sigma}{1 - \Delta t\, K_f^\sigma} \|u^{(k)} - u^{(k-1)}\|, \tag{13}$$

provided of course that $\Delta t\, K_f^\sigma < 1$.

For the next step we also need to assume

$$\|\mathcal{C}(v^{(k+1)}) - \mathcal{C}(v^{(k)})\| \le \Delta t\, L_C \|v^{(k+1)} - v^{(k)}\|,$$

for a constant $L_C$. This is a reasonable assumption given the make-up of the entries $\mathcal{C}_{i,j}(v^{(k)})$, if we assume a fixed grid. If $u^{(k)}$ and $u^{(k+1)}$ satisfy $\mathcal{G}(u^{(k)}, v^{(k)}) = 0$ and $\mathcal{G}(u^{(k+1)}, v^{(k+1)}) = 0$, respectively, then

$$\mathcal{M}^\kappa (u^{(k+1)} - u^{(k)}) = \Delta t \Big( g(u^{(k+1)}, v^{(k+1)}) - g(u^{(k)}, v^{(k)}) \Big) + \mathcal{C}(v^{(k+1)}) - \mathcal{C}(v^{(k)}).$$

Let $K_g^\kappa = L_g \| \left(\mathcal{M}^\kappa\right)^{-1} \|$ and $K_C = L_C \| \left(\mathcal{M}^\kappa\right)^{-1} \|$. Then proceeding as before we find

$$\|u^{(k+1)} - u^{(k)}\| \le \Delta t\, K_g^\kappa \Big( \|u^{(k+1)} - u^{(k)}\| + \|v^{(k+1)} - v^{(k)}\| \Big) + \Delta t\, K_C \|v^{(k+1)} - v^{(k)}\|,$$

and hence

$$\|u^{(k+1)} - u^{(k)}\| \le \frac{\Delta t\, (K_g^\kappa + K_C)}{1 - \Delta t\, K_g^\kappa} \|v^{(k+1)} - v^{(k)}\|, \tag{14}$$

provided that $\Delta t\, K_g^\kappa < 1$. Combining (13) and (14) gives

$$\|u^{(k+1)} - u^{(k)}\| \le \frac{\Delta t\, (K_g^\kappa + K_C)}{1 - \Delta t\, K_g^\kappa} \frac{\Delta t\, K_f^\sigma}{1 - \Delta t\, K_f^\sigma} \|u^{(k)} - u^{(k-1)}\|,$$

Shifting the index in (14) and combining with (13) shows the same estimate is valid for the sequence $\{v^{(k)}\}_{k=1}^\infty$. Therefore, a standard argument shows that if

$$\frac{\Delta t\, (K_g^\kappa + K_C)}{1 - \Delta t\, K_g^\kappa} \frac{\Delta t\, K_f^\sigma}{1 - \Delta t\, K_f^\sigma} < 1$$

then there are vectors $u^{n+1}, v^{n+1}$ such that $(v^{(k)}, u^{(k)}) \to (u^{n+1}, v^{n+1})$, as $k \to \infty$, and $\mathcal{F}(u^{n+1}, v^{n+1}) = \mathcal{G}(u^{n+1}, v^{n+1}) = 0$.

## Acknowledgements

This work has been partially supported by the National Institutes of Health under contract grant number R01 GM072005-01.

## References

[1] J.W.Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.

[2] W. Hundsdorfer, J.G. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, Springer-Verlag, New York, NY, 2003.

[3] E.F. Keller and L.A. Segel, Initiation of slime mold aggregation viewed as an instablity, *J. Theor. Biol.*, **26** (1970), 399-415.

[4] C.T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, PA, 1995.

[5] M.W. Smiley, A monotone conservative Eulerian-Lagrangian scheme for reaction-diffusion-convection equations modeling chemotaxis, *Numerical Methods for Partial Differential Equations*, **23** (2007), 553-586.

[6] C. Van Loan, *Computational Frameworks for the Fast Fourier Transform*, SIAM, Philadelphia, PA, 1992.

# Chaotic dynamics in a three species mutual interference aquatic population model with Holling type II functional response

Ranjit Kumar Upadhyay
Department of Applied Mathematics
Indian School of Mines, Dhanbad, Jharkhand-826 004, India.
E-mail: ranjit_ism@yahoo.com

## Abstract

In this paper, we propose a new mathematical model for aquatic populations, based on a modified version of the Leslie-Gower scheme. This model incorporates mutual interference in all the three populations, an extra mortality term in middle population (zooplankton) and also taking into account the toxin liberation process of TPP population. We investigate the dynamical behavior of the model system and observe the role of mutual interference and TPP by considering the Holling type II functional response of toxin liberation process. The computed bifurcation diagrams and two-dimensional parameter scans suggest that chaotic dynamics is robust to changes in changes against rates in toxin production by phytoplanktons as it exists in large range of $q$ values. Many forms of complex dynamics are observed, including period-doubling bifurcation with period-halving bifurcation cascade, saddle –node bifurcation. Our study suggests that toxic substances released by TPP population may act as bio-control by changing the state of chaos to order and extinction of predator species and mutual interference also induces chaos and acting as both stabilizing as well as destabilizing factors.

*Keywords:* Chaotic dynamics, mutual interference, toxin producing phytoplankton, aquatic system, functional form.

## 1. Introduction

Chaotic dynamics play an important role in continuous time models for ecological systems. There is some evidences that the real time evolution of species involving two or three species in a food chain could be characterized by chaotic attractors as observed in many natural food-chains [3, 12, 17, 19, 20]. Upadhyay and Rai [20] produced new examples of a chaotic population system in a simple tri-trophic food chain with Holling type II functional responses. Aziz- Alaoui [3] revisited the Upadhyay and Rai's model and found that the chaotic dynamics is observed via sequences of period-doubling bifurcation of limit cycles which however suddenly break down and reverse giving rise to a sequence of period-halving bifurcation leading to limit cycles. Upadhyay and Chattopadhyay [18] modified the model of Upadhyay and Rai [20], by introducing an extra mortality term in middle predator and interpret the system for aquatic environment consisting of TPP-Zooplankton-Molluscs food chain model. They observe that increasing the strength of toxic substance released by TPP population reduce the propensity of chaotic dynamics and changing the state of chaos to limit cycles and finally settled down to stable focus.

A number of studies have investigated the effect of mutual interference on the population dynamics. DeAngelis et al. [6] studied the dynamical properties of a continuous-time autonomous model system incorporating their interference model. This model was taken up by Hwang [13] to establish that the periodic orbits, if it exists, are unique. The models considered for interference have different mathematical expressions and different conceptual foundations [2]. From theoretical studies, and from some empirical evidence, a consensus has emerged to consider that interference has a stabilizing influence on population dynamics [5], although Hassell and May [11] pointed out that there was an upper limit on the interference constant beyond which the dynamics become unstable. Motivated from the above studies, we show that chaotic behaviour as described by Upadhyay and Rai [19, 20] could be controlled by an auto-control mechanism.

In this paper, we propose a new model of aquatic ecological system by introducing mutual interference in all the three populations, an extra mortality term in zooplankton population and also taking into account the toxin liberation pr ocess of TPP population. This model generalizes the several other known models in the literature like Upadhyay and Rai model [19, 20] and Hastings and Powell model [12]. Our study shows that chaotic dynamics is observed via sequences of period-doubling bifurcation of limit cycles. It also shows that chaotic dynamics is robust to changes in rates of toxin release. One of main objectives of our study is to examine the role of mutual interference parameters ($m_i, i = 1, 2, 3$) and the parameter $q$, the rate of toxin release by TPP population on the chaotic dynamics of the model system in response to different types of toxin release functions $f_1(x_1)$, which represents the toxin liberation process of TPP population.

This paper is organized as follows: In the next Section, we present the details of the model system. The methodology used is presented in Section 3, which helps us to select the biologically realistic parameter values to perform simulation experiments. Numerical results are summarized in Section 4 and some important conclusions are discussed in Section 5.

## 2. The Model System

Consider a situation where a prey population $x_1$ is predated by individuals of population $x_2$. The population $x_2$, in turn serves as a favourite food for individuals of population $x_3$. This inte raction is represented by the following system of a simple prey - specialist predator - generalist predator interaction:

$$\frac{dx_1}{dt} = g_1(x_1, x_2, x_3) \equiv a_1 x_1 - b_1 x_1^2 - w_0 \left( \frac{x_1}{x_1 + D_0} \right)^{m_1} x_2^{m_2}, \qquad (1a)$$

$$\frac{dx_2}{dt} = g_2(x_1, x_2, x_3) \equiv -a_2 x_2 + w_1 \left( \frac{x_1}{x_1 + D_1} \right)^{m_1} x_2^{m_2} - w_2 \left( \frac{x_2}{x_2 + D_2} \right)^{m_2} x_3^{m_2} - q\, f_1(x_1)\, x_2, \qquad (1b)$$

$$\frac{dx_3}{dt} = g_3(x_1, x_2, x_3) \equiv c x_3^{m_3} - w_3\, f_2(x_2)\, x_3^{m_3} \qquad (1c)$$

where $m_i > 0$ for $i = 1, 2, 3$, $a_1, a_2, b_1, q, w_0, w_1, w_2, w_3, c$ and $D_0, D_1, D_2, D_3, D_4 > 0$, $f_i \in C^{'}(R_+)$ for $i = 1, 2, 3$.

The parameters $m_i$ for $i = 1, 2, 3$ are mutual interference parameters that model the intraspecific competition among predators when hunting for prey [4, 7, 8, 9, 10].

In this model, TPP population (prey) of size $x_1$ serves as the only food for the specialist predator (zooplankton) population of size $x_2$. This zooplankton population, in turn, serves as a favorite food for the generalist predator (mollusks) population of size $x_3$. In this model, $a_1, a_2, b_1, w_0, w_1, w_2, w_3, c, D_0, D_1, D_2, D_3, D_4$ and $q$ are positive constants. The equations for rate of change of population size for prey and specialist predator have been written following the Volterra scheme that is, predator population dies out exponentially in the absence of its lone prey. The interaction between this predator $x_2$ and the generalist predator $x_3$ is modeled by the modified version of the Leslie-Gower scheme where the loss in a predator population is proportional to the reciprocal of per capita availability of its most favorite food. $a_1$ is the intrinsic growth rate of the prey population $x_1$, $a_2$ is the intrinsic death rate of the predator population $x_2$ in the absence of the only food $x_1$, $c$ measures the rate of self-reproduction of generalist predator $x_3$, $w_0$, $w_1$, $w_2$, $w_3$ are the maximum values which per capita growth rate can attain. $b_1$ measures the strength of intra-specific competition among the individuals of the prey species $x_1$. $D_0$ and $D_1$ quantify the extent to which environment provides protection to the prey $x_1$ and may be thought of as a refuge or a measure of the effectiveness of the prey in evading a predator's attack. $D_2$ is the value of $x_2$ at which per capita removal rate of $x_2$ becomes $w_2/2$. $D_3$ represents the residual loss in $x_3$ population due to severe scarcity of its favourite food $x_2$. For $m_1 = 1$, the coefficient $w_0/(x_1 + D_0)$, of the third term on the right hand side of eq. (1a) is obtained by considering the probable effect of the density of the prey's population on predators attack rate. If this coefficient is multiplied by $x_1$ (the prey population at any instant of time), it gives the attack rate on the prey per predator. Denote $p(x_1) = w_0 x_1/(x_1 + D_0)$, when, $x_1 \to \infty, p(x_1) \to w_0$, which is the maximum that it can reach. Some aquatic organisms condition their medium by releasing substances that stimulate growth of species, which have similar genetic make-up. Sparse populations rarely provide sufficient opportunities for social interaction necessary for reproduction. Here $f_1(x_1)$ represents the toxin liberation process of TPP population for which the mortality of zooplankton increases and as a result, the grazing pressure of zooplankton on TPP population decreases. We assume that

$f_1(0) = 0 \; and \; \frac{\partial f_1(x_1)}{\partial x_1} > 0.$ The parameter $q$ is the rate of toxin release by TPP population. Equations (1a-1c) describe the proposed model system.

It is easy to see that the functions $g_i, i = 1, 2, 3$ in (1a-1c) are continuous on $R_+^{\,3}$, in which $R_+ = [0, \infty)$. Clearly, when $m_i \geq 1$ the functions $\frac{\partial g_i}{\partial x_k}$ are continuous on $R_+^{\,3}$. Following Erbe et al.[7], we obtain conditions under which the solutions of (1a-1c) form a dynamical system. The question of the solutions of (1a-1c) forming a dynamical system needs further investigation especially when the parameters $m_i$ are sub-linear ($0 < m_i < 1$).

We now state the following assumptions, which will be verified in respect of our model system to conclude that the solutions of (1a-1c) form a dynamical system when the interference parameters $m_i$ for $i$= 1, 2,3 are sublinear.

(H1) there exists functions h$_j$ continuous on $R_+^{\,3}$ where

$$h_j(x_1, x_2, x_3) = x_j^{\,-m_i} g_j(x_1, x_2, x_3) \quad \text{with } 0 < m_j < 1, \text{ j= 1, 2, 3;}$$

(H2) $x_k^{m_K} \frac{\partial}{\partial x_k} h_j(x_1, x_2, x_3)$ are continuous on $R_+^{\,3}$, for j ≠ k = 1, 2, 3; and

(H3) All solutions of the system $\frac{du_i}{dt} = h_i(u_1, u_2, u_3)$ for $i$ =1, 2, 3 are continuous on $R_+^{\,3}$.

As in Erbe et al.[ 7], we consider the following change of variables for (1a-1c)

$$u_1 = x_1^{\,1-m_1}, \quad u_2 = x_2^{\,1-m_2}, \quad u_3 = x_3^{\,1-m_3}. \tag{2}$$

This would transform the system (1a-1c) into

$$u_1' = (1-m_1)\left[a_1u_1 - b_1u_1^{\frac{2-m_1}{1-m_1}} - w_0\frac{u_2^{\frac{m_2}{1-m_2}}}{\left(u_1^{\frac{1}{1-m_1}} + D_0\right)^{m_1}}\right] \equiv h_1(u_1, u_2, u_3) \tag{3a}$$

$$u_2' = (1-m_2)\left[-a_2u_2 + w_1\frac{u_1^{\frac{m_1}{1-m_1}}}{\left(u_1^{\frac{1}{1-m_1}} + D_1\right)^{m_1}} - w_2\frac{u_3^{\frac{m_2}{1-m_3}}}{\left(u_2^{\frac{1}{1-m_2}} + D_2\right)^{m_2}} - q\, f_1\left(u_1^{\frac{1}{1-m_1}}\right)u_2\right] \tag{3b}$$

$$\equiv h_2(u_1, u_2, u_3)$$

$$u_3' = (1 - m_3)\left[c - w_3 \; f_2\left(u_2^{\frac{1}{1-m_2}}\right)\right] \equiv h_3(u_1, u_2, u_3) \tag{3c}$$

Clearly the transformation (2) transforms the sublinear system (3c) into (3b) in which no sublinearities are present. Biologically, this amounts to requiring that the mutual interferences are not too strong.

The above discussions may be summarized as follows.

**Theorem 2.1** Consider the system of equations (1a-1c) in which $x_i(0) \geq 0$, $0 < m_i < 1$, for $i$=1, 2, 3. Assume that the assumptions (H1), (H2) and (H3) hold. Then, the solutions of the system equations (1a-1c) form a dynamical system in the sense of Nemytskii and Stepanov [15] provided the mutual interference parameters satisfy the following inequalities:

$$m_1 \geq \frac{1}{2}\,,\; m_2 \geq \frac{1}{2}\,,\; \text{ and } \; m_2 + m_3 \geq 1.$$

## 3. Methods of investigation

The model system presented above are multi-parameter system. Model parameters are selected in accordance with a method given in upadhyay et al. [19, 20]. A few hundred parameter combinations (choosing two at a time) are possible. This is simply not feasible for any one to scan the system in all the parameter spaces. Application of non-linear dynamics is unison with the knowledge of biology of the system enables one to choose parameter combinations for simulation experiments. The most crucial part of the present methodology is the following conjecture:

*Two coupled Kolmogorov systems in oscillatory mode would yield either cyclic (stable limit cycles and quasi-periodic) or chaotic solutions depending on the strength of coupling between the two.*

In the present case, the set of parameter values for which the system admits a limit cycle solution is found to be

$$a_1 = 2.0, b_1 = 0.05, w_0 = 1.0, a_2 = 1.0, w_1 = 2.0, D_1 = 10, w_2 = 0.55, D_2 = 10, q = 0.003,$$
$$c = 0.03, w_3 = 1.0, D_3 = 10, D_4 = 10, m_1 = 0.95, m_2 = 0.95, m_3 = 2.0.$$

There is one more important aspect of these simulation experiments i.e., choosing the step size for the variation of a system parameter from a parameter combination within the chosen range. It depends on the nature of the parameter concerned: whether it is a slow varying or fast varying one.

The most useful way to study such a dynamical system is to monitor the amplitude (maxima) of the subsequent oscillations as the control parameter of the system is varied. A small change in parameter values may lead to a bifurcation: an abrupt, qualitative change in the dynamics.

## 4. Numerical Results

In order to better understand the dynamics of the model system, we turn to numerical simulations. Computer simulations were performed on MATLAB for the system equation (1a-1c). The search for chaos was carried out using the Physics Academy Software (AIP, New York) ODE workbench package. Our primary interest is to explore the occurrence of chaotic dynamics in the model system. We try to observe the role of toxin producing phytoplankton on the chaotic dynamics in such ecosystems. We also examine the role of mutual interference parameter $m_i$ and the parameter $q$, the rate of toxin release by TPP population on the chaotic dynamics of the model system.

Model system is integrated numerically using six-order Runge-Kutta method along with predictor corrector method. It is observed that the model system (1a-1c) has a chaotic solution at the following set of parameter values (see Fig. 1)

$$\begin{aligned}&a_1 = 1.93, b_1 = 0.06, \quad w_0 = 1.0, \quad D_0 = 10.0, \quad a_2 = 1.0, \quad w_1 = 2.0,\\ &D_1 = 10.0, \; w_2 = 0.405, D_2 = 10.0, \; c = 0.03, w_3 = 1.0, D_3 = 20.0,\\ &m_1 = 1.0, \; m_2 = 1.0, \; m_3 = 2.0, \; q = 0.0.\end{aligned} \tag{9}$$

To confirm the existence of chaos, the dynamics of the model system is studied by constructing bifurcation diagrams. For Holling type II functional response form for toxin liberation process, we have plotted the successive maxima of top predator $x_3$ as a function of the parameter $q$ (rate of toxin substances release by TPP population) keeping other parameters fixed as given in eqn. (9) for model system (1a-1c). The figures 2(a-b) are representing the bifurcation diagrams of model system (1a-1c) with $f_1(x_1)$ as Holling type II functional response. These figures show clearly the transition from chaos to order through sequences of period-halving bifurcation. Therefore, for the model system, it is observed that, increase of value of toxic substances released by TPP population has a stabilizing effect. The blow –up bifurcation diagram (see Fig. 2b) show that the model system possesses rich variety of dynamical behaviour for bifurcation parameter $q$ in the ranges [0, 0.07] for Holling type II functional response. A period –doubling cascade is observed. After the accumulation point, the behaviour settles down onto a chaotic attractor. When $q$, the bifurcation parameter is decreased, new periodic orbits are created. Two different bifurcations are involved. First, period-doubling bifurcations that are easily identified in the main period doubling cascade but also within each periodic window. Second, saddle –node bifurcations, creating one stable limit cycle and one unstable periodic orbit, both having the same period, may be identified at the beginning of each periodic window. Here, the most observable periodic windows are the one associated with the main period-doubling cascade and the other associated with the saddle-node bifurcation inducing the stable period-1 limit cycle. Two co-existing period-doubling cascades are then observed.

Dynamical behavior of model system (1a-1c) depending on the results of bifurcation diagrams given in figs 2 is presented in table 1. We have observed stable focus, different order limit cycles and strange chaotic attractor in the different ranges of $q$, the rate of toxic substance released by TPP. From the table 1, it is observed that for the model system, the increase of value of toxic substances released by TPP has a stabilizing effect. Model system also shows the extinction of the predator species for higher values of $q$. These observations indicate that to maintain the order of an ecosystem functioning, Holling type II functional form for toxin liberation process is more appropriate.

Now, we have investigated the role of mutual interference parameters on the dynamics of trophic system in detail. The values of mutual interference parameters were chosen on the basis of the values reported in Katz [14]. We have observed stable focus, limit cycles and chaotic dynamics phenomena in the model system by changing the mutual interference parameters $m_i$, $i=1, 2, 3$ and the rate of toxin release by TPP population $q$, in the fixed range for different cases. We have also reported the function error or argument domain error, the region in the parameter space where no dynamics is observed. In this domain, the values of mutual interference parameters are not conducive for simulation experiment i.e., in real situation, no species can attain these values of mutual interference. Our approach is first to fix $m_1$ and $m_2$ then vary $m_3$ in the interval [1, 3] and $q$ in the interval [0,1) and then observe the exchange of states (stability - limit cycle - period doubling – chaos) in the model system for three different cases of $m_i$ (>,=,<).

The results for model system (1a -1c) are summarized below:

Case I: When $m_i > 0$

(A) For $f_1(x_1) = \dfrac{x_1}{(x_1 + D_4)}$ (Holling type II), $f_2(x_2) = \dfrac{1}{(x_2 + D_3)}$. (see Table 3)

(i) For $m_1 = m_2 = 1.05$ and $1.5 \le m_3 \le 3.0$, $0 \le q \le 1$.

Chaos exists at some discrete points. For example, chaos exists for $(m_3, q) = (1.75, 0.4), (2.0, 0.45), (2.0, 0.5), (2.0, 0.55), (2.25, 0.5), (2.25, 0.55)$. Rest of the points it shows the limit cycle attractor.

(ii) For $m_1 = m_2 = 2.0$ and $1 \le m_3 \le 3$, $0 \le q \le 1$.

From the simulation, it is found that in most of the cases, $x_2$ becomes extinct and $(x_1, x_3)$ rests on stable focus for higher values of $q$. For lower values of $q$, all the three populations rests on stable focus and limit cycle attractor in the phase plane. It is also observed that for $m_1 = m_2 = 1.25, 1.5, 1.75$ and for whole range of the parameter space $(m_3, q)$ (i.e., $1 \le m_3 \le 3$, $0 \le q \le 1$), the model system (1a-1c) predicts no dynamics. The simulation results show function error or argument domain error. These values of mutual interference parameters are not conducive for the simulation experiment i.e., in real life situation, no species can attain these values of mutual interference.

Case II: When $m_i = 1$ (i.e., $m_1 = m_2 = m_3 = 1$).

(A) For $f_1(x_1) = \dfrac{x_1}{(x_1 + D_4)}$ (Holling type II), $f_2(x_2) = \dfrac{1}{(x_2 + D_3)}$, $0 \le q \le 1$.

Chaos exists in the interval $0 \le q \le 0.25$. For $q \in [0.3, 0.4]$, and $q \in [0.45, 0.7]$, we obtain the limit cycle and stable focus behaviour respectively. For rest values of $q \in [0.8, 1.0]$, $(x_2, x_3)$ becomes extinct and $x_1$ rests on a stable focus but at $q = 0.75$, only $x_3$ becomes extinct and other species rests on stable focus. Fig. 3 shows the chaotic behaviour of the model system (1a-1c) in the domain $0.75 \le m_3 \le 2.25,\ 0 \le q \le 0.35$.

Case III: When $m_i < 0$
In this case, chaos does not exist at all. The domain in which we perform the two dimensional scans are
$m_1 = m_2 = 0.25,\ 0.5,\ 0.75,\ 0.95;\ m_3 = 0.25,\ 0.5,\ 0.75$ and $0 \le q \le 1$.

We obtain only function error in this domain except for $m_1 = m_2 = 0.95$. We also did the simulation for the above values of $m_1, m_2$ and $(m_3, q)$ in the domain $(0.25 \le m_3 \le 3,\ 0 \le q \le 1)$. Results are presented in the tabular form in table 4. From the tables, it is observed that the mutual interference also stabilize the system. From table 4, it is found that for $m_1 = m_2 = 0.25,\ 0.5,\ 0.75$ and $m_3$ in the range [1, 3] and $q$ in the range [0, 1], the dynamics is settled on stable focus. For $m_1 = m_2 = 0.95$ and in whole range of $m_3$ and $q$, stable focus and limit cycles are observed.

Case IV: (New Adventure)

On the experimental basis, by looking the conditions of the Theorem 2.1, we have taken different combination of the mutual interference parameters and observed its influence on the dynamics of the model system (1a-1c). The results are reported in table 2. We observed only function error. These values of mutual interference parameters are not conducive for the simulation experiment i.e., in real life situations; no species can attain these values of mutual interference parameters.

## 5. Conclusions

In this paper, we have attempted to answer the question does mutual interference and toxic substances released by TPP always stabilize the prey-predator dynamics in aquatic environment. Our simulation experiments suggest that the answer is definite 'yes'. From the tables, it is observed that for different values of mutual interference parameters in different ranges, dynamics of the model system is also influenced by the functional form of toxin liberation process. For $m_i < 1\ (i = 1, 2, 3)$, from table 4, no dynamics was observed (i.e., represented by function error in the tables) in the range $0.25 \le m_i \le 0.75$, but if we take any one of the interference parameters value near 1, we observe the system dynamics on stable focus. In this case (Holling type II functional response), the top predator becomes extinct as $m_3$ reaches 1. For $m_i > 1$, most of the time, dynamics rests on stable limit cycle or stable focus. From tables (3), it is found that for $m_1 = m_2 = 1.05$ (i.e. near 1) and $m_3$ in the range [1, 3], system dynamics settled on limit cycle attractor. In this case, model system also supports chaotic dynamics only at few discrete points. But for $m_1 = m_2 = 2$, and $m_3$ in the range [1, 3], mostly system dynamics settled on stable focus and middle predator becomes extinct for all the three form of functional responses. These results show that the interaction between predators is a stabilizing factor, one could thus argue that evolution has selected species of predators that have values close to 1, but smaller than 1. This could be an explanation of why Arditi and Akcakaya [1] found in many sets of his experimental data, $m_i$ was close to 1.

From the tables and 2D scan diagram, it was also observed that the model system supports chaotic dynamics for $m_i \geq 1$. It is caused by deterministic changes in system parameters not by exogenous stochastic influences. In this case, the ecological systems are not able to lock themselves onto a fully developed chaotic state. Since it is constantly influenced by exogenous stochastic fluctuations [16], it is forced to leave the chaotic state as initial condition changes. We also observe from bifurcation diagrams is that chaotic dynamics is robust to changes in changes against rates in toxin production by phytoplankton as it exist for large range of $q$ value. Period doubling bifurcations seem to be responsible for this kind of dynamical behaviour. The study suggests that toxic substances release by TPP plays an important role in the termination of planktonic blooms which is of great importance to human health, ecosystems, environment and fishery. It may also act as biological control by changing chaos to order and has stabilizing contribution to aquatic systems.

## References

[1] R. Ariditi, H.R. Akcakaya, Underestimation of mutual interference of predators, Oecologia 83 (1990) 358-361.

[2] R. Arditi, J-M. Callois, Y. Tyutyunov, C. Jost, Does mutual interference always stabilize predator-prey dynamics? A comparison of models, Comptes Rendus Biologies 327 (2004) 1037-1057.

[3] M.A. Aziz-Alaoui, Study of a Leslie-Gower-type tritrophic population model, Chaos Solitons and Fractals 14 (2002) 1275-1293.

[4] J.R. Beddington, Mutual interference between parasites on predators and its effects on searching efficiency, J. Animal Ecology 44 (1975), 331-340.

[5] M. Begon, M. Mortimer, D.J. Thompson, Population Ecology: A unified study of Animals and Plants, third ed. Blackwell Science, Oxford, U.K. 1996.

[6] D.L. DeAngelis, R.A. Goldstein, R.V. O'Neill, A model for trophic interaction, Ecology 56 (1975) 881-892.

[7] L.H. Erbe, H.I. Freedman, Modeling persistence and mutual interference among subpopulations of ecological communities, Bull. Math. Biol. 47 (1985) 295-304.

[8] L.H. Erbe, H. I. Freedman, V. Sree Hari Rao, Three species food chain models with mutual interference and time delays, Math.Biosciences 80 (1986) 57-80.

[9] H.I Freedman, V.Sree Hari Rao, The trade off between mutual interference and time lags in predator-prey Systems, Bull. Math. Biol. 45 (1983) 991-1004.

[10] M. P. Hassell, Mutual interference between searching insect parasites, J. Animal Ecology 40 (1971) 473-486.

[11] M.P. Hassell, R. M. May, Stability in insect host-parasite models, J. Animal Ecology 42 (1973) 693-726.

[12] A. Hastings, T. Powell, Chaos in three species food-chain, Ecology 72 (1991) 896-903.

[13] T.W. Hwang, Uniqueness of limit cycles of the predator-prey system with Beddington-DeAngelis functional response, J. Math. Anal. Appl. 290 (2004) 113-122.

[14] C.H. Katz, A nonequilibrium marine predator-prey interaction, Ecology 66 (1985) 1426-1438.

[15] V.V. Nemytskii and V.V. Stepanov, Qualitative Theory of Differential Equations, Princeton University Press, Princeton, 1960.

[16] V. Rai, R.K. Upadhyay, Chaotic population dynamics and biology of the top-predator, Chaos Solitons and Fractals 21 (2004) 1195-1204.

[17] G.D. Ruxton, Chaos in a three-species food chain with a lower bound on the bottom population, Ecology 71(1) (1996) 317-319.

[18] R.K. Upadhyay, J. Chattopadhyay, Chaos to order: Role of toxin producing phytoplankton in aquatic systems. Nonlinear Analysis: Modelling and Control 10(4)(2005) 383-396.

[19] R.K. Upadhyay, S.R.K. Iyengar, V. Rai, Chaos: an ecological reality?, Int. J. Bifurcation and Chaos 8(6) (1998) 1325-1333.

[20] R.K. Upadhyay, V. Rai, Why chaos is rarely observed in natural populations?, Chaos Solitons and Fractals 8(12) (1997) 1933-1939.

Table 1. Dynamical behavior (DB) of model system depending on the results of bifurcation diagrams given in fig. 2. Pi - limit cycle of period $i$ for ($i$ = 2, 3, 4, 5, 6), SF – stable focus, LC – limit cycle, LP – long period, SCA – strange chaotic attractor, EX- Extinction.

| Results of model 1 for Holling type II: $f(x) = x/(x + D_4)$ $D_4 = 10$ | |
|---|---|
| $q$ = | DB |
| 0.001-0.0111 | SCA |
| 0.0112 | P6 |
| 0.0113-0.0115 | P5 |
| 0.0116-0.0123 | P4 |
| 0.0124-0.059 | SCA |
| 0.06 | LP |
| 0.061 | P6 |
| 0.062-0.068 | P4 |
| 0.07-0.16 | P2 |
| 0.17 | LP |
| 0.18-0.39 | LC |
| 0.4-0.6 | SF |
| 0.7 | EX |

Table 2. Dynamical behaviour of the model system (1a-1c) depending on the results of the Theorem 2.1. The values of the common parameters are same as given in table 2. The mutual interference parameters $m_i, (i = 1, 2, 3)$ and $q$ varies in the given ranges.

| Values of $m_i$<br>$(a)$ $0 < m_i < 0.5$ and this violates $m_2 + m_3 \geq 0$ | Values of $q$ in $0 \leq q \leq 1$ | Dynamical Behaviour $f_1(x_1) = \dfrac{x_1}{(x_1 + D_4)}$ |
|---|---|---|
| (i) $m_1 = 0.25$ $m_2 = 0.45, m_3 = 0.35$ | 0-1.0 | Function Error |
| (ii) $m_1 = 0.25$ $m_2 = 0.25, m_3 = 0.25$ | 0-1.0 | Function Error |
| (iii) $m_1 = 0.45$ $m_2 = 0.25, m_3 = 0.45$ | 0-1.0 | Function Error |
| (iv) $m_1 = 0.35$ $m_2 = 0.45, m_3 = 0.45$ | 0-1.0 | Function Error |
| (v) $m_1 = 0.35$ $m_2 = 0.45, m_3 = 0.5$ | 0-1.0 | Function Error |
| (b) $0 < m_1 < 0.5$; $m_2 = m_3 = 0.5$. | | |
| $m_1 = 0.1, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45.$ | 0-1.0 | Function Error |

Table 3. Simulation experiments of model system (1a-1c) with $f_1(x_1)$ as Holling type II functional response. The values of the common parameters used in the model system are same as given in table 2 with $D_4 = 10.0$.

The mutual interference parameters $m_i > 1(i = 1, 2, 3)$, and $q$ varies in the range [0, 1].

| Values of<br>$m_1$ *and* $m_2$ | $m_3$ in<br>$1 \leq m_3 \leq 3$ | Values of $q$ in<br>$0 \leq q \leq 1$ | Dynamical behaviour |
|---|---|---|---|
| $m_1 = m_2 = 1.05$ | 1.5 | 0-0.0001; 0.002-1.0 | Function Error |
| | | 0.0002-0.001 | Limit cycle |
| | 1.75 | 0; 0.0004-0.001 | Function Error |
| | | 0.0001-0.0003; 0.002-0.0095 | Limit cycle |
| | | 0.01-0.085; 0.095-0.35; 0.65-1.0 | Function Error |
| | | 0.09; 0.5-0.6 | Limit cycle |
| | | 0.4 | Chaos |
| | 2.0 | 0-0.0002; 0.0005-0.0008 | Limit cycle |
| | | 0.0003-0.0004; 0.0009-0.007 | Function Error |
| | | 0.0075-0.02 | Limit cycle |
| | | 0.025-0.25 | Function Error |
| | | 0.3-0.4; 0.6-0.65 | Limit cycle |
| | | 0.45-0.55 | Chaos |
| | | 0.7-1.0 | Function Error |
| | 2.25 | 0-0.002 | Limit cycle |
| | | 0.003; 0.02-0.088; 0.095-0.25 | Function Error |
| | | 0.0035-0.015;0.09; 0.3-0.45 | Limit cycle |
| | | 0.5-0.55 | Chaos |
| | | 0.6-0.7 | Limit cycle |
| | | 0.75-1.0 | Function Error |
| | 2.5 | 0-0.02; 0.3-0.75 | Limit cycle |
| | | 0.025-0.29; 0.77-1.0 | Function Error |
| | 2.75 | 0.085, 0.095-0.28; 0.8-1.0 | Function Error |
| | | 0.09, 0.3-0.75 | Limit cycle |
| | 3.0 | 0.35-0.42; 0.7– 0.75 | Limit cycle |
| | | 0.44 - 0.66; 0.8-1.0 | Function Error |
| $m_1 = m_2 = 1.25$ | 1.0-3.0 | 0-1.0 | Function Error |
| $m_1 = m_2 = 1.5$ | 1.0-3.0 | 0-1.0 | Function Error |
| $m_1 = m_2 = 1.75$ | 1.0-3.0 | 0-1.0 | Function Error |
| $m_1 = m_2 = 2.0$ | 1.0 | 0 -1.0 | $x_1$ SF ; ($x_2$, $x_3$) extinct |
| | 1.25-2.0 | 0 -1.0 | ($x_1$, $x_3$) SF; $x_2$ extinct |
| | 2.25 | 0-0.1 | ($x_1$, $x_2$, $x_3$) SF |
| | | 0.15-1.0 | ($x_1$, $x_3$) SF; $x_2$ extinct |
| | 2.5 | 0 - 0.4 | ($x_1$, $x_2$, $x_3$) SF |
| | | 0.45 – 1.0 | ($x_1$, $x_3$) SF; $x_2$ extinct |
| | 2.75 | 0 –0.5 | ($x_1$, $x_2$, $x_3$) SF |
| | | 0.55 – 0.6 | ($x_1$, $x_2$, $x_3$) Limit cycle |
| | | 0.65-1.0 | ($x_1$, $x_3$) SF; $x_2$ extinct |
| | 3.0 | 0 – 0.5 | ($x_1$, $x_2$, $x_3$) SF |
| | | 0.55-0.85 | ($x_1$, $x_2$, $x_3$) Limit cycle |
| | | 0.86 – 1.0 | ($x_1$, $x_3$) SF; $x_2$ extinct |

Table 4. Simulation experiments of model system (1a-1c) with $f_1(x_1)$ as Holling type II functional response. The values of the common parameters used in the model system are same as given in table 2 with $D_4 = 10.0$. The mutual interference parameters $m_3$ and $q$ varies in the ranges [0.25,3] and [0, 1] respectively.

| Values of $m_1$ and $m_2$ | $m_3$ in $0.25 \le m_3 \le 3$ | Values of $q$ in $0 \le q \le 1$ | Dynamical behaviour |
|---|---|---|---|
| $m_1 = m_2 = 0.25$ | 0.25- 0.75 | $0-1.0$ | Function Error |
| | 1.00- 3.0 | 0-1.0 | Stable Focus |
| $m_1 = m_2 = 0.5$ | 0.25-0.75 | 0-1.0 | Function Error |
| | 1.0-3.0 | 0-1.0 | Stable Focus |
| $m_1 = m_2 = 0.75$ | 0.25-0.75 | 0-1.0 | Function Error |
| | 1.0-3.0 | 0-1.0 | Stable Focus |
| $m_1 = m_2 = 0.95$ | 0.25 | 0-0.004 | Limit cycle |
| | | 0.005-0.009 | Function Error |
| | | 0.01-0.1 | Limit cycle |
| | | 0.2-0.4 | Stable Focus |
| | | 0.45-1.0 | Function Error |
| | 0.5-0.75 | 0 -0.15 | Limit cycle |
| | | 0.2-0.4 | Stable Focus |
| | | 0.45-1.0 | Function Error |
| | 1.0 | 0-0.15 | Limit cycle |
| | | 0.2-0.7 | Stable Focus |
| | | 0.75-1.0 | $(x_1, x_2)$SF, $x_3$ extinct |
| | 1.25-2.25 | 0-0.15 | Limit cycle |
| | | 0.2-1.0 | Stable Focus |
| | 2.5 | 0-0.0002 | Limit cycle |
| | | 0.0003-0.0004 | Function Error |
| | | 0.0005-0.0006 | Limit cycle |
| | | 0.0007-0.0009 | Function Error |
| | | 0.001-0.006 | Limit cycle |
| | | 0.007-0.02 | Function Error |
| | | 0.03-0.15 | Limit cycle |
| | | 0.2-1.0 | Stable Focus |
| | 2.75 | 0- 0.15 | Integration Error |
| | | 0.2-1.0 | Stable Focus |
| | 3.0 | 0-0.24 | Integration Error |
| | | 0.25-1.0 | Stable Focus |

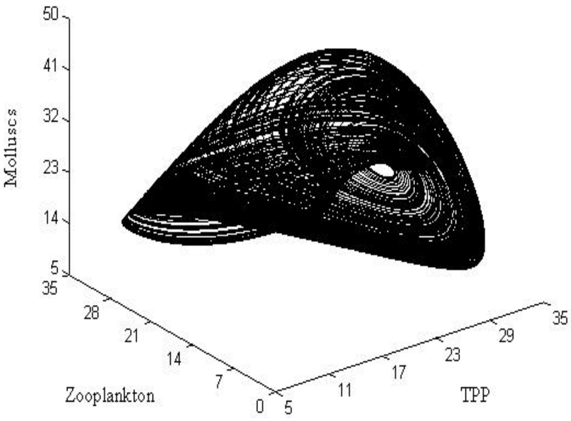


Fig. 1. Phase plane diagram for model system (1a-1c) depicting chaotic attractor for $q = 0$, other parameter are same as given in eq. (9).

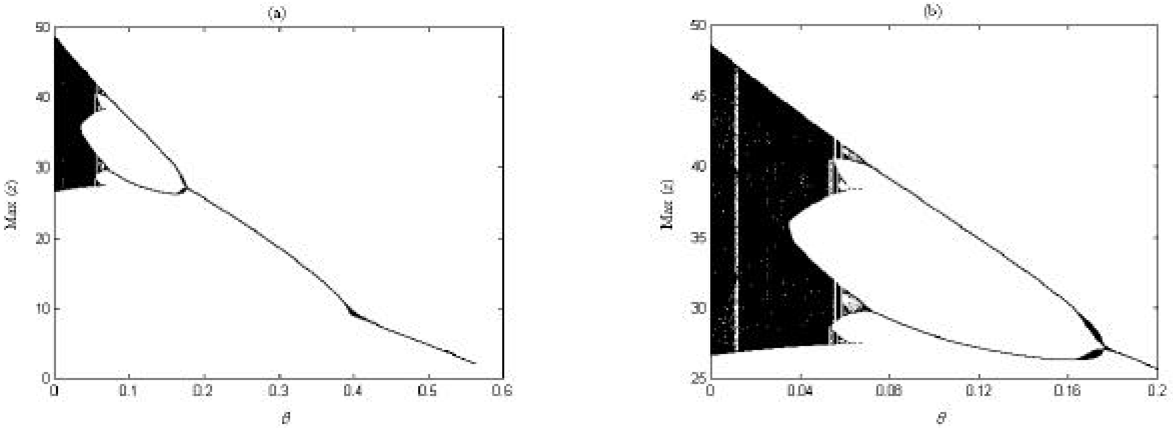


Fig. 2. (a) Bifurcation diagram as a function of $q$ for model sy stem with $f_1(x_1)$ of Holling type II. (b) Blown up bifurcation diagram of (a) in the range $0 \le q \le 0.2$. Here, $z$ stands for $x_3$ in model system (1a-1c).

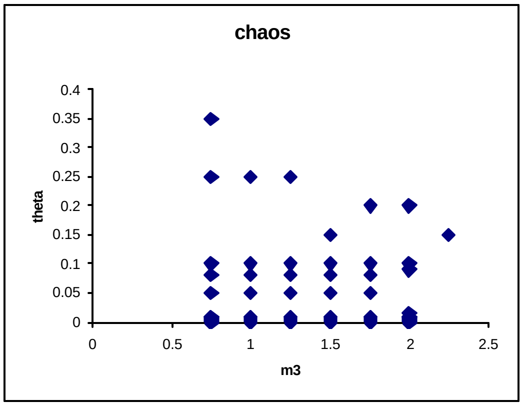


Fig. 3. Model system. 2D scan diagram between $(m_3, q)$ parameter space for Holling type II functional responses with the parameter values $a_1 = 2,\ b_1 = 0.05,\ w_2 = 0.55,\ D_4 = 10.0,\ q = 0.003.$ other parameters are same as given in eq. (9).

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# An improved nonlinear algorithm appropriate for solving special initial-value problems of the form $y' = f(y)$

**Jesús Vigo-Aguiar**[1] **and Higinio Ramos**[1]

[1] *Departamento de Matemática Aplicada, Grupo de Computación Científica.*
*University of Salamanca*

emails: `jvigo@usal.es`, `higra@usal.es`

### Abstract

Recently the authors have presented a nonlinear explicit scheme suitable for numerically solving first-order initial-value problems (IVP) of the form $y' = f(y)$. The algorithm is based on the local approximation of the function $f(y)$ by a second-order Taylor expansion where the resulting approximated differential equation is then solved without local truncation error. In this paper we proposed an improved approximation based on a Chebyshev interpolation polynomial of second degree taken at the Chebyshev-Gauss-Lobato points.

*Key words: initial value-problems, nonlinear methods, Chebyshev approximation*
*MSC 2000: 65L05*

## 1 Introduction

Many authors have worked on numerical methods suitable for solving initial value problems of special types in ordinary differential equations. Such problems are stiff problems, singularly perturbed problems, or problems whose solution $y(x)$ or the first derivative of the solution $y'(x)$ contain singularities on the interval of integration.

We will consider the autonomous scalar initial-value problem given by

$$y' = f(y), \qquad y(a) = y_0\,, \tag{1}$$

where $y, f(y) \in \mathbb{R}$, and $x \in [a,b] \subset \mathbb{R}$, and it is assumed that the solution is unique. The conventional explicit one-step method for (1) is given by $y_{n+1} = \alpha\, y_n + h\, \Phi_f(y_n, h)$, where $\Phi_f(y_n, h)$ is the incremental function, and the subscript $f$ on the right hand side indicates that the dependence of $\Phi$ on $y_n$ is through the function $f$. The selected step size $h$ is such that for the mesh points we have $x_j = a + j\,h$, $j = 0, \ldots, k$. Typical examples of the above scheme are the linear one-step methods or the Runge-Kutta methods (which are essentially substitution methods)[1].

Nonlinear multistep methods are usually designed for dealing with unconventional problems for which the classical schemes generally perform poorly [2], [4], [5], [6], [7], [8], [9]. On the contrary, some of the nonlinear schemes specifically designed for singular problems do not perform well on non-singular problems [9]. In this paper we propose a non-linear scheme suitable for IVPs, based on the local approximation of the function $f(y)$ by a Chebyshev approximation. Some numerical experiments confirm the well performance of the method.

## 2 Approximation by a truncated Chebyshev series expansion

Recently the authors have presented in [7] an approach for handling some type of special problems. This scheme was based on the approximation of the right hand side of the differential equation by a second-order Taylor expansion and the use of an exact procedure for solving the approximated problem. In this paper we propose a similar approach, but this time we consider instead of the Taylor polynomial an interpolation polinomial of second degree based on the Chebyshev-Gauss-Lobatto nodes [10].

We show this approximation that will be used in the next section for the development of the numerical scheme.

Let be a function $f(y)$ defined on an interval named $[y_n, y_{n+1}]$. The transformation given by

$$y = y_n + \frac{h_y}{2}\,(1+\alpha)$$

where $h_y = y_{n+1} - y_n$ introduces a new variable $\alpha \in [-1,1]$ and the function

$$\bar{f}(\alpha) = f(y_n + \frac{h_y}{2}\,(1+\alpha))$$

may be approximated by means of a truncated series of Chebyshev polynomials in the form [11]

$$\bar{f}(\alpha) = \sum_{k=0}^{2}{}'' \, a_k\, T_k(\alpha) \tag{2}$$

where $T_k(\alpha)$ is the Chebyshev polynomial of fist kind of degree $k$, the double primes indicate that both the first and last terms in the summation are to be halved, and

$$a_k = \sum_{j=0}^{2}{}'' \, \bar{f}(\alpha_j)\, T_k(\alpha_j)\,,$$

where

$$\alpha_j = \cos(\theta_j)\,, \qquad j = 0, 1, 2 \tag{3}$$

and

$$\theta_j = \frac{(n-j)\pi}{n}$$

are the so-called Chebyshev-Gauss-Lobatto points.

In Fig. 1 we can see the absolute errors in the approximation of the function $f(y) = \cos^2(y)$ on the interval $[y_n, y_{n+1}] = [\pi/4, \pi/4 + 0.1]$ using the second-order Taylor polynomial about $y_n$ (red line) and by the interpolating polinomial of second degree given by the right hand side in (2) (blue line).

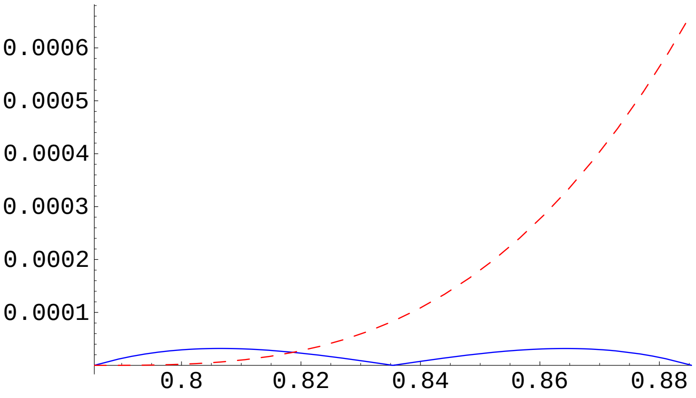


Figure 1: Absolute errors in the approximation by second-degree polynomials: Taylor (dashed line) and Chebyshev (solid line).

## 3 A non-linear explicit one-step scheme

Suppose we have solved numerically the problem in (1) up to a point $x_n$, and assuming the localization hypothesis $y_n = y(x_n)$, we want to obtain an approximate value for the solution at the point $x_{n+1} = x_n + h$, that is, $y_{n+1} \simeq y(x_{n+1})$. For this purpose we consider the approximation of the function $f(y)$ on the interval $[y_n, y_{n+1}]$ using the procedure in the above section. In this way we obtain

$$f(y) = \sum_{k=0}^{2}{}'' a_k\, T_k(\alpha)$$

where

$$a_k = \sum_{j=0}^{2}{}'' f\left(y_n + \frac{1}{2}\, h_y\, (1 + \alpha_j)\right)\, T_k(\alpha_j)\,, \tag{4}$$

with the $\alpha_j$ as in (3).

The value $y_{n+1}$ is not known, and we provide it by means of the Euler method, that is,

$$y_{n+1} = y_n + h\, f(y_n)$$

and so $h_y$ takes the form $h_y = h\, f(y_n)$.

With the above settings the approximated differential equation reads

$$y' = {\sum_{k=0}^{2}}'' a_k\, T_k \left( \frac{2(y-y_n)}{h\, f(y_n)} - 1 \right) \tag{5}$$

where now the $a_k$ are given by

$$a_k = {\sum_{j=0}^{2}}'' f\left( y_n + \frac{1}{2}\, h\, f(y_n)(1+\alpha_j) \right) T_k(\alpha_j)\,,$$

that is, we have approximated on $[x_n, x_{n+1}]$ the differential equation in (1) by the differential equation in (5) which is of Riccati type of the form

$$y' = a + b\, y + c\, y^2 \tag{6}$$

where $a, b, c$ are certain constants which depend on $h$ and $y_n$ through $f$. Explicitly these constants reads

$$\begin{aligned}
a &= \frac{(h^2 f_n^2 + 3h y_n f_n + 2y_n^2) f_n - 4(y_n^2 + h f_n y_n) f_{n+1/2} + (h f_n y_n + 2y_n^2) f_{n+1}}{f_n^2\, h^2} \\
b &= \frac{-(3f_n h + 4y_n) f_n + 4(f_n h + 2y_n) f_{n+1/2} - (f_n h + 4y_n) f_{n+1}}{f_n^2\, h^2} \\
c &= \frac{2(f_n - 2f_{n+1/2} + f_{n+1})}{f_n^2\, h^2}
\end{aligned}$$

where $f_n, f_{n+1/2}, f_{n+1}$ are abbreviations for

$$f_n = f(y_n)\,, \quad f_{n+1/2} = f(y_n + \frac{h}{2}\, f_n)\,, \quad f_{n+1} = f(y_n + h f_n)\,.$$

We can solve on the interval $[x_n, x_{n+1}]$ the problem in (6) with the initial condition $y(x_n) = y_n$ exactly, that is to say, without local truncation error, by considering the difference scheme

$$y_{n+1} = \begin{cases} y_n - \dfrac{2\tan(hS)}{(b+2c\, y_n)\tan(hS) - 2S}\, f(y_n)\,, & \Delta > 0\,, \\[2ex] y_n - \dfrac{2\tanh(hS)}{(b+2c\, y_n)\tanh(hS) - 2S}\, f(y_n)\,, & \Delta < 0\,, \\[2ex] y_n - \dfrac{2h}{b\, h + 2\, c\, h\, y_n - 2}\, f(y_n)\,, & \Delta = 0 \end{cases} \tag{7}$$

where

$$\Delta = 4\,a\,c - b^2\,, \qquad S = \frac{1}{2}\,\sqrt{|\Delta|}\,.$$

Inserting the above values of $a, b, c$ in (7) we obtain the numerical scheme. After applying this scheme we get an approximation for the true solution of (1) at $x_{n+1}$ given by $y_{n+1} \simeq y_{n+1}$ . Repeating the procedure along the nodes on the integration interval a discrete solution for the problem in (1) is formed.

## 4 Local truncation error

In order to obtain the expression for the local truncation error we proceed as in [1]. Let $u(x)$ be an arbitrary function defined in $[a, b]$ sufficiently differentiable, and we consider the functional $\mathcal{L}$ associated with the method in (7) defined by

$$\mathcal{L}(u(x), h) = u(x+h) - (u(x) - \Psi_u)\ , \tag{8}$$

where

$$\Psi_u = \begin{cases} \dfrac{2\tan(hS_u)}{(b_u + 2c_u\, y_n)\tan(hS_u) - 2S}\, u'(x)\,, \quad \Delta_u > 0\,, \\[2ex] \dfrac{2\tanh(hS_u)}{(b_u + 2c_u\, y_n)\tanh(hS_u) - 2S_u}\, u'(x)\,, \quad \Delta_u < 0\,, \\[2ex] \dfrac{2h}{b_u\, h + 2\, c_u\, h\, y_n - 2}\, u'(x)\,, \quad \Delta_u = 0 \end{cases}$$

with

$$\Delta_u = 4\, a_u\, c_u - b_u^2\,, \qquad Su = \frac{1}{2}\,\sqrt{|\Delta_u|}\,,$$

The constants $a_u\,, b_u\,, c_u$ are similar to those in (6) but with $y_n$ and $f(y(x))$ substituted by $u(x)$ and $u'(x)$ respectively.

After expanding in Taylor series about $x$ the right hand side in (8) we obtain that

$$\mathcal{L}(u(x), h) = \begin{cases} \left(\dfrac{1}{6}\, u^{(3)}(x) - \dfrac{u''(x)^2}{4\, u'(x)}\right) h^3 + \mathcal{O}(h^4)\,, \quad \Delta_u = 0\,, \\[2ex] -\dfrac{u''(x)^2}{6\, u'(x)}\, h^3 + \mathcal{O}(h^4)\,, \quad \Delta_u \neq 0\,, \end{cases}$$

which indicates that the method has second order.

# 5 Stability analysis

As the numerical method in (7) is exact when the right hand side of the differential equation is a polynomial up to second degree on $y$, the method is trivially A-stable.

In fact, if we apply the method to the scalar Dahlquist test problem given by

$$y'(x) = \lambda\, y(x)\,, \qquad Re(\lambda) < 0\,,$$

we obtain the difference equation

$$y_{n+1} = \exp(\lambda\, h)\, y_n\,,$$

where the stability function is given by $R(\lambda\, h) = \exp(\lambda\, h)$, from which it is deduced not only A-stability but in fact a stronger property of L-stability (see [12]).

# 6 Numerical results

## 6.1 A non linear equation

The first example to be considered is the IVP given by

$$y'(x) = \cos^2(y(x))\,, \qquad y(0) = \frac{\pi}{4}\,,$$

whose exact solution is $y(x) = \arctan(x+1)$. We have integrated this problem on the interval $[0, \pi]$ taking different values for the constant stepsize $h$. In Table 1 the results for this problem with the new method are shown, compared with those in [7]. The errors have been obtained as the absolute errors at the final point on the integration interval

$$E_{end} = |y(x_{end}) - y_{NI}|\,,$$

where $NI$ refers to the number of steps used in the integration.

| $NI$ | $E_{end}$ (Method in [7] ) | $E_{end}$ (New Method) |
|---:|:---:|:---:|
| 25 | $2.4234 \times 10^{-6}$ | $1.3669 \times 10^{-8}$ |
| 50 | $2.9198 \times 10^{-7}$ | $9.5624 \times 10^{-10}$ |
| 100 | $3.5822 \times 10^{-8}$ | $6.2627 \times 10^{-11}$ |
| 200 | $4.4358 \times 10^{-9}$ | $3.9959 \times 10^{-12}$ |
| 400 | $5.5187 \times 10^{-10}$ | $2.5468 \times 10^{-13}$ |
| 800 | $6.8824 \times 10^{-11}$ | $2.2870 \times 10^{-14}$ |

Table 1: Errors for $y'(x) = \cos^2(y(x))\,,\ y(0) = \pi/4$

A plot of the absolute errors for $NI = 200$ is shown in Fig. 2.

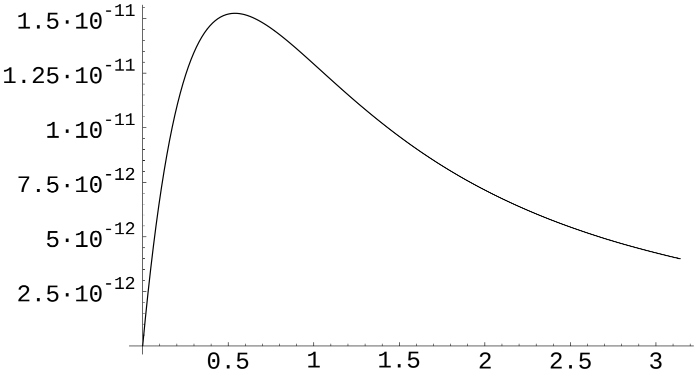


Figure 2: Absolute errors for Problem 6.1, $NI = 200$.

### 6.2 A singular problem

As a second example we consider the IVP given in by

$$y' = 1 + y^2 \,, \qquad y(0) = 1 \,, \quad x \in [0, b] \,, \tag{9}$$

The theoretical solution is $y(x) = \tan(x + \pi/4)$, which becomes unbounded in the neighborhood of the singularity at $x = \pi/4 \simeq 0.785398163397448$, and so the conventional numerical integrators result inefficient. This problem has been used as numerical test for different integrators intended for numerically solving singular initial value problems [2], [3], [4], [5], [6], [9]. As the right hand side of the differential equation is a polynomial in $y$ of second degree, our method integrates the problem exactly, that is, without local truncation error.

In Table 2 we have compared for different step sizes the results of our newly developed scheme with those obtained by Odekunle *el al.* [9] with an inverse Runge-Kutta scheme of order four, and with the results obtained in [7]. There are small differences between the results with the method in this paper an the results in [7], but these methods are clearly better for this problem and we observe the ability of both methods to overpass the singularity. The absolute errors were obtained at the point $x = 0.8$. Note that the smaller the step size is the bigger is the error, due to round off error considerations.

### 6.3 A singularly perturbed problem

The last example corresponds to the singularly perturbed IVP taken from [13]

$$\epsilon \, y'(x) = (x - 1) \, y(x) \,, \qquad y(0) = 1 \,,$$

| $h$ | *Odekunle* [9] | *Method* in [7] | *NewMethod* |
|---|---|---|---|
| 0.050000 | $3.1 \times 10^{-4}$ | $1.2 \times 10^{-12}$ | $6.0 \times 10^{-13}$ |
| 0.025000 | $1.8 \times 10^{-5}$ | $1.3 \times 10^{-12}$ | $3.2 \times 10^{-12}$ |
| 0.012500 | $2.7 \times 10^{-6}$ | $3.5 \times 10^{-12}$ | $4.9 \times 10^{-12}$ |
| 0.010000 | $3.8 \times 10^{-6}$ | $1.9 \times 10^{-12}$ | $1.9 \times 10^{-12}$ |
| 0.006250 | $2.3 \times 10^{-5}$ | $1.0 \times 10^{-11}$ | $1.2 \times 10^{-11}$ |
| 0.003125 | $7.2 \times 10^{-5}$ | $1.5 \times 10^{-11}$ | $1.9 \times 10^{-11}$ |

Table 2: Absolute errors at $x = 0.8$ for $y' = 1 + y^2\,,\ y(0) = 1$

which has exact solution given by

$$y(x) = \exp\left(\frac{x(x-2)}{2\,\epsilon}\right)\,.$$

We have considered $\epsilon = 0.001$ so that the solution drops quickly from its initial value of 1 to very small values and near the final point goes from very small values to 1, exhibiting $\mathcal{O}(\epsilon)$-thick layers near the initial and the final point on the integration interval. Note that in this case the function is $f(x, y)$ which implies a change in the algorithm, instead of (4) we have to consider the coefficients

$$a_k = \sum_{j=0}^{2}{}'' f\left(x_n + \frac{1}{2}\,h\,(1+\alpha_j), y_n + \frac{1}{2}\,h_y\,(1+\alpha_j)\right)\,T_k(\alpha_j)$$

and $f(x_n, y_n)$ instead of $f(y_n)$.

In Fig. 3 we have plotted the numerical solution for a really small range, joining the discrete points. Table 3 shows the errors obtained with the method in [7] and with the new method in this paper. The errors presented in this table are defined as

$$E_{max} = \max_{x_j \in [0,2]} \{|y(x_j) - y_j|\}\,,$$

where $NI$ refers as usual to the number of time steps.

We note that near $x = 0$ the new method performs very well, although the problem exhibits an initial layer there. Fig. 4, where the absolute errors along the integration interval have been plotted, shows this feature.

## References

[1] J. D. Lambert, *Numerical Methods for Ordinary Differential Systems*, John Wiley, England, 1991.

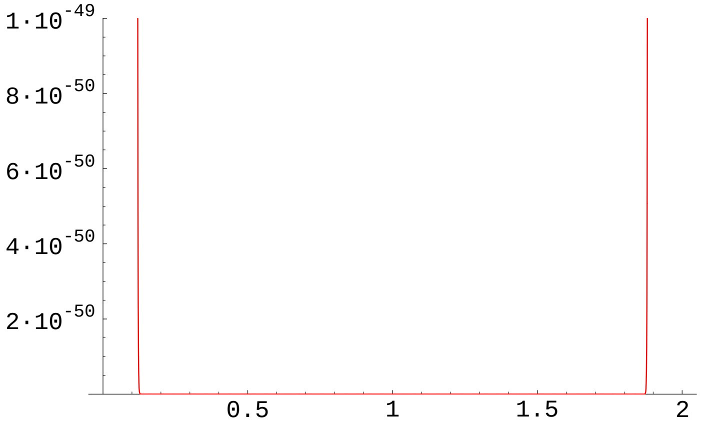


Figure 3: Numerical solution for Problem 6.3, $NI = 2500$.

| NI | $E_{max}$ (Method in [7]) | $E_{max}$ (New Method) |
|---|---|---|
| 2500 | 0.55067 | $1.43683 \times 10^{-2}$ |
| 5000 | 0.32968 | $1.73822 \times 10^{-3}$ |
| 7500 | 0.23407 | $5.05213 \times 10^{-4}$ |
| 75000 | 0.18126 | $2.09364 \times 10^{-4}$ |
| 12500 | 0.12598 | $8.91362 \times 10^{-5}$ |

Table 3: Maximum absolute errors for $\epsilon\, y' = (x-1)\, y\,,\;\; y(0) = 1\,,\, \epsilon = 0.001$

[2] S. O. Fatunla, *Numerical Methods for IVPs in ODEs*, Academic Press Inc., London, 1988.

[3] S. O. Fatunla, *Nonlinear multistep methods for initial value problems*, Comp. & Maths. with Appls. Vol. **8, No. 3** (1982) 231–239.

[4] S. O. Fatunla, *Numerical treatment of singular initial value problems*, Comp. & Maths. with Appls. Vol. **12B, No. 56** (1986) 1109–1115.

[5] F. D. Van Niekerk, *Rational one-step methods for initial value problems*, Comp. Math. Applic. Vol. **16, No. 12** (1987) 1035–1039.

[6] S. Abelman and D. Eyre, *A numerical stydy of multistep methods based on continued fractions*, Computers Math. Applic. Vol. **20, No. 8** (1990) 51–60.

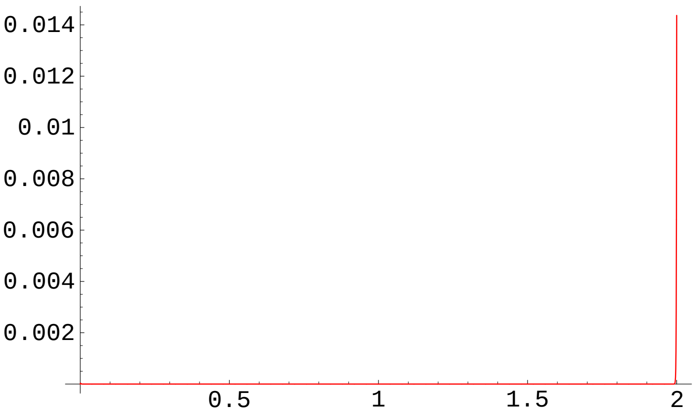


Figure 4: Absolute errors for Problem 6.3, $NI = 2500$.

[7] H. Ramos and J. Vigo-Aguiar, *A non-standard algorithm appropriate for solving singular and singularly perturbed autonomous initial-value problems*, Int. Journal of Computer Mathematics, To appear (1990)

[8] H. Ramos , *A non-standard explicit integration scheme for initial-value problems*, Applied Mathematics and Computation. Vol. **189, No. 8** (2007) 710–718.

[9] M. R. Odekunle, N. D. Oye, s. O. Adee and R. A. Ademiluyi, *A class of inverse Runge-Kutta schemes for the numerical integration of singular problems*, Appl. Math. Comput. **158** (2004) 149–158.

[10] W. S. Don and A. Solomonoff, *Accuracy enhancement for higher derivatives using Chebyshev collocation and a mapping technique*, SIAM J. Sci. Comput. **18**, No.4, (1997) 1040-1055

[11] L. Fox and I. B. Parker, *Chebyshev Polynomials in Numerical Analysis.* Oxford U. Press, London, 1968.

[12] E. Hairer and G. Wanner, *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems.*, Springer, Berlin, 1996.

[13] R. E. O'Malley, Jr., *Give Your ODEs a Singular Perturbation!*, Journal of Mathematical Analysis and Applications. **251** (2000) 433–450.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE2007*
*Chicago, 20-23 June 2007*

# SVD Stabilized Block Diagonal Preconditioner for Large Scale Dense Complex Linear Systems in Electromagnetics

**Yin Wang[1], Jeonghwa Lee[2] and Jun Zhang[1]**

[1] *Laboratory for High Performance Scientific Computing and Computer Simulation, Department of Computer Science, University of Kentucky*
[2] *Department of Computer Science, Shippensburg University*

emails: ywangf@csr.uky.edu, jlee@ship.edu, jzhang@cs.uky.edu

## Abstract

Preconditioned Krylov subspace methods are usually considered to be the methods of choice for solving large systems of linear equations arising from practical scientific and engineering modeling and simulations, such as in the electromagnetic applications. Due to its easy implementation and potential for parallelism, the block diagonal preconditioner is often used to accelerate the convergence rate of Krylov iterative methods. However, in some special electromagnetics computations, the block diagonal preconditioner actually slows down the Krylov subspace convergence. We propose to use singular value decomposition to stabilize the inverse of the blocks, which is generated from the multilevel fast multipole algorithm. Experimental results show that the new preconditioned iteration scheme converges faster compared with the standard block diagonal preconditioner and reduces the overall CPU time.

*Key words: Krylov subspace methods, Singular Value Decomposition (SVD), Block diagonal preconditioner, Multilevel fast multipole algorithm (MLFMA)*
*MSC2000: 65F10, 65R20, 65F30, 78A45*

## 1. Introduction

In computational electromagnetics, we use hybrid volume and surface integral equations to model three-dimensional (3D) arbitrarily shaped dielectric and conducting objects. The computed solution has applications in radar cross-section (RCS) prediction for coated targets, printed circuit, and microstrip antenna analysis. The hybrid integral equation can be discretized into a linear system of the form

$$Ax = b \tag{1}$$

by using the method of moments (MoM) [8, 14, 16, 22], where the coefficient matrix $A$ is a large dense complex valued matrix for large targets in electromagnetic scattering.

The complex linear equation (1) can be solved by using direct solution methods or iterative solution methods. In this paper, our attention is on using efficient iterative methods. In particular, we use the biconjugate gradient (BiCG) method which is one of the Krylov methods as our iterative solver [1, 9, 18].

Let $N$ be the number of unknowns of the equation (1). The complexity of BiCG type iterative methods is $O(N^{iter}N^2)$, where $N^{iter}$ indicates the number of iterations and $N^2$ is the computational cost of a matrix vector product operation, which accounts for the major cost of Krylov methods. We use the fast multipole method (FMM) to reduce the computational complexity of the matrix vector product operation from $O(N^2)$ to $O(N^{1.5})$ [7, 17]. Furthermore, with the multilevel fast multipole algorithm (MLFMA) this complexity can be reduced to $O(N\log N)$ [6, 13, 20].

In order to further reduce the computational cost for solving the linear system (1), we can apply a carefully constructed preconditioning technique to equation (1) to accelerate the convergence rate of Krylov methods in the form of

$$M^{-1}Ax = M^{-1}b \tag{2}$$

where $M^{-1}$ is a nonsingular matrix of order $N$. Our primary goal is to choose a robust and efficient preconditioner $M^{-1}$ to make the new linear system (2) much easier to solve and to reduce the overall CPU time.

Recently, the development and practice of efficient preconditioning techniques in iteratively solving dense linear systems has been the subject of growing interest [3, 4, 5, 11, 12]. The difficulty in constructing an efficient preconditioner for this class of dense linear system is that the global coefficient matrix $A$ is not explicitly available in the MLFMA implementation. So, in the MLFMA, the block diagonal preconditioner has been more popular [13, 20, 23] because we can easily construct the block diagonal preconditioner from the small block diagonal submatrices, which are available explicitly. The ILUT preconditioner and sparse approximate inverse (SAI) preconditioner are reported to be more efficient, but more complex, preconditioning techniques in this application [3, 10, 11, 12].

Some cases arising from electromagnetic scattering problems show a bad convergence behavior with the block diagonal preconditioning [10]. We try to figure out what causes this bad convergence behavior. It is our suspicion that the LU factorizations of the individual blocks are not stable due to the ill-conditioning of some blocks. We intend to stabilize the block diagonal preconditioner by using singular value factorization (SVD) on each individual block instead of using the LU factorization. In our numerical experiments, we choose the BiCG method as an iterative solver combined with the different preconditioning strategies. These preconditioning strategies under comparison are (a) no preconditioner, (b) standard block diagonal preconditioner, and (c) SVD stabilized block (SVDB) preconditioner. Our numerical results show that the SVDB preconditioning technique coupled with the BiCG method accelerates the convergence rate of some difficult cases compared with the standard block diagonal preconditioner.

## 2. Discretization of Hybrid Integral Equation and MLFMA

The hybrid integral equation approach combines the volume integral equation (VIE) and the surface integral equation (SIE) to model the scattering and radiation by mixed dielectric and conducting structures [14, 19]. The VIE is applied to the material region (V) and the SIE is enforced over the conducting surface (S). The integral equations can be formally written as follows:

$$\begin{aligned}&\{L_S(r,r')\cdot J_S(r') + L_V(r,r')\cdot J_V(r')\}_{\tan} = -E_{\tan}^{inc}(r), \quad r\in S,\\ &-E + L_S(r,r')\cdot J_S(r') + L_V(r,r')\cdot J_V(r') = -E^{inc}(r), \quad r\in V,\end{aligned}$$

where $E^{inc}$ stands for the excitation field produced by an instant radar, the subscript ''tan'' stands for taking the tangent component from the vector it applies to, and $L_\Omega$, $(\Omega = S,V)$, is an integral operator that maps the source $J_\Omega$ to electric field $E(r)$ and it is defined as:

$$L_S(r,r')\cdot J_\Omega(r') = i\omega\mu_b \int_{\Omega'} (I + k_b^{-2}\nabla\nabla)G(r,r')\cdot J_\Omega(r')d\Omega'.$$

Here $G(r,r') = e^{ik_b|r-r'|}/(4\pi\,|\,r-r'\,|)$ is the 3D scalar Green's function for the background media and $i=\sqrt{-1}$. We note that $J$ is related to $J_V$ in the above integral equations by $J_V = i\omega(\varepsilon_b - \varepsilon)E$. This results in a very general model as all the volume and surface regions can be modeled properly. The advantage of this approach is that in the coated object scattering problems, the coating material can be inhomogeneous, and in the printed circuit and microstrip antenna simulation problems the substrate can be of finite size. The simplicity of the Green's function in both the VIE and the SIE has an important impact on the implementation of the fast solvers. However, the additional cost here is the increase in the problem size since the volume that is occupied by the dielectric material is meshed. This results in larger memory requirement and longer solution time in solving the corresponding matrix equation. But this deficiency can be overcome by applying fast integral equation solvers such as the MLFMA [6].

Using the method of moments (MoM), the hybrid integral equations are discretized into a matrix equation of the form as the form

$$\begin{bmatrix} Z^{SS} & Z^{SV} \\ Z^{VS} & Z^{VV}\end{bmatrix}\cdot\begin{bmatrix} a^S \\ a^V\end{bmatrix} = \begin{bmatrix} U^S \\ U^V\end{bmatrix}, \tag{3}$$

where $a^S$ and $a^V$ stand for the vectors of the expansion coefficients for the surface current and the volume function, respectively [13, 14], and the matrix elements can be generally written as

$$Z_{jl} = i\omega\mu_b \int_\Omega d\Omega f_j^\Omega(r) \cdot \int_{\Omega'} d\Omega' (I + k_b^{-2}\nabla\nabla)\ G(r,r') \cdot \chi(r') f_l^{\Omega'}.$$

The material function $\chi(r') = 1$ if $\Omega'$ is a surface patch, and $\chi = (\varepsilon / \varepsilon_b - 1)$ if $\Omega'$ is a volume cell. The coefficient matrix arising from discretized hybrid integral equations is nonsymmetric. Once the matrix equation (3) is solved by numerical matrix equation solvers, the expansion coefficients $a^S$ and $a^V$ can be used to calculate the scattered field and the radar cross section (RCS). In antenna analysis problems the coefficients can be used to retrieve the antenna's input impedance and to calculate the antenna's radiation pattern. In the following, we use $A$ to denote the coefficient matrix in the Equation (3), $x = [a^S, a^V]^T$, and $b = [U^S, U^V]^T$ for simplicity.

The basic idea of the FMM is to convert the iteration of element-to-element to the interaction of group-to-group. Using the addition theorem for the free-space scalar Green's function, the matrix-vector product $Ax$ can be written as [6, 7]

$$Ax = (A_D + A_N)x + V_f \Lambda V_s x, \tag{4}$$

where $V_f$ , $\Lambda$, and $V_s$ are sparse matrices. In fact, the dense matrix $A$ can be structurally divided into three parts, $A_D$, $A_N$, and $A_F = V_f \Lambda V_s$. $A_D$ is the block diagonal part of $A$, $A_N$ is the block near-diagonal part of $A$, and $A_F$ is the far part of $A$. Here the terms "near" and "far" refer to the distance between groups of elements.

In FMM, those elements in $A_F$ are not explicitly computed and stored. Hence they are not numerically available. It can be shown that with optimum grouping, the number of nonzero elements in the sparse matrices in equation (3) are all on the order of $N^{1.5}$, and hence the operation count to perform $Ax$ is $O(N^{1.5})$ [7]. But if the above process is implemented in multilevel, the total cost can be further reduced to be proportional to $N \log N$ for one matrix-vector multiplication.

## 3. SVD Stabilized Block Preconditioner

There are two things we need to consider in order to choose an efficient preconditioner for solving the large dense linear system. One is to make the preconditioned matrix $M^{-1}A$ as close to the identity matrix $I$ as possible, and the other is to choose a preconditioner which is not too expensive to construct and has a potential to be parallelizable.

By using the block diagonal preconditioner, we construct a preconditioner $M^{-1}$ from the block diagonal submatrix $A_D$, and apply the block diagonal preconditioner

$M^{-1}$ to the linear system (1). That is, $M^{-1}Ax = M^{-1}b$. Here, $A_D$ is a block diagonal submatrix like in equation (4) consisting of several block submatrices such as $A_1, A_2, \cdots, A_m$. Each individual block can be decomposed independently by an *LU* factorization in the form of $A_i = L_i U_i (1 \le i \le m)$ and the linear system for each block $A_i$ will be solved as $L_i U_i x_i = b_i$. The block diagonal preconditioner $M^{-1}$ is of *m* independently inverted submatrix $A_1^{-1}, A_2^{-1}, \cdots, A_m^{-1}$, where $A_i^{-1} = (L_i U_i)^{-1}$, so $M^{-1} = A_D^{-1}$. It provides us a convenient structure to parallelize the preconditioner, because we can distribute the jobs into different processor for each block [13, 15, 20].

Our previous studies shows that block diagonal preconditioner sometimes fail to improve the ill-conditioned linear systems in this application [10]. To alleviate the problem, we test the shift of the diagonal entries, and in some cases the convergence becomes more slowly than the block diagonal preconditioner without a diagonal shift. We suspect that some of these blocks may be ill-conditioned or close to singular. Their LU factorizations may not be stable, in the sense that large size entries are created in the inverse LU factors. SVD is known to be a very powerful technique for dealing with matrices that are either singular or else numerically very close to singular. In many cases when LU factorization fails to give satisfactory results, SVD may tell us what the problem is or how to solve it [2]. The computational and memory cost of SVD of large matrices may be expensive. To avoid this, we apply SVD to each small block submatrix $A_i (1 \le i \le m)$. By using SVD, $A_i$ can be decomposed into three matrices as following [2, 15, 21, 24]

$$A_i = U_i \cdot \Sigma_i \cdot V_i^H, \tag{5}$$

where $U_i$ and $V_i$ are orthogonal matrices, and all the singular values are stored in the diagonal of $\Sigma_i$ as $\Sigma_i = diag[\sigma_1, \sigma_2, ..., \sigma_k]$, $\sigma_1 \ge \sigma_2 \ge \ldots \ge \sigma_k$, $k$ is the size of the block $A_i$.

The inverse of $A_i$ is computed as $A_i^{-1} = V_i \cdot \Sigma_i^{-1} \cdot U_i^H$. To stabilize the inverse, we replace some small singular values of $\Sigma_i$ by a larger value. This can be done with a threshold strategy. Given a threshold value $\varepsilon$, if there exists an integer $j$ such that $\sigma_j > \varepsilon$, but $\sigma_{j+1} \le \varepsilon$ ($1 \le j \le k$), then for every $l$ ( $j < l \le k$ ), we set $\sigma_l = \varepsilon$.

Then we have the updated $\bar{\Sigma}_i = diag[\sigma_1, \sigma_2, ..., \sigma_j, \varepsilon, ..., \varepsilon]$, and the stabilized inverse of $A_i$ is computed as:

$$A_i^{-1} = V_i \, \bar{\Sigma}_i^{-1} \, U_i^H \tag{6}$$

The computational procedure is given in Algorithm 3.1, in which, $\varepsilon$ is a user provided parameter.

**ALGORITHM 3.1** Computing the SVD stabilized block preconditioner from the block diagonal submatrices ( $k_i$ is the size of block $A_i$ )

```
0. For $i = 1,...,m$ (where $m$ is the number of blocks) Do
1. Obtain the $ith$ block $A_i$ from $A_D$ and run the SVD subroutine to get $U_i$, $\Sigma_i$, and $V_i^H$.
2. For each block, set the correct threshold $\varepsilon$ and find the correct rank $j$, then for every $l (j < l \le k_i)$, we set $\sigma_l = \varepsilon$.
3. Compute the $U_i^H$ and $V_i = (V_i^H)^H$.
4. Get $\bar{\Sigma}_i^{-1}$ by computing $1/\sigma_j$ ($1 \le j \le k$).
5. Compute an SVD stabilized block as $A_i^{-1} = V_i \bar{\Sigma}_i^{-1} U_i^H$
6. EndDo
```

The following explains how to choose the correct threshold value to update the singular values of each block in Algorithm 3.1. In our experiment, we use a static strategy to update the singular values, which means to choose a global threshold for every block. For a comparison, we can also compute some characteristics for each and every block during SVD thresholding, such as the mean value of all the singular values for each block or choosing the threshold $\varepsilon_i = (\sigma_1 \ / \sigma_k) \cdot error\_bound$. ($\varepsilon_i$ is the threshold for the $ith$ block, and $error\_bound$ is $1.0e-3$). This can make the threshold more reasonable when the singular values are quite different in different blocks. Both methods have the same purpose, which is to make the SVD stabilized block inverse better conditioned than the block diagonal one. Because we know that

$$cond(A_i) = \sigma_1 / \sigma_k \quad (k \text{ is the column size of block } A_i) \qquad (7)$$

where $\sigma_1$ is the largest singular value of the block $A_i$ and $\sigma_k$ is the smallest one. By using our updating strategies, the condition number of each block will be reduced to $\sigma_1 / \varepsilon$, which means the inverse of that block will be more stable. We expect this stabilization will make the new preconditioner more effective than the block diagonal preconditioner.

Of course, there is always a trade-off between stability and accuracy. Choosing a large threshold value produces a more stable, but less accurate inverse, while choosing a smaller threshold having the opposite effect.

## 4. Numerical Tests and Analysis

In this section, we present a few numerical experiments to demonstrate the efficiency of the SVD stabilized block preconditioner for accelerating the BiCG iterations. All cases are tested using one processor of an HP Superdome cluster at the University of Kentucky. The processor has 2 GB local memory and runs at 1.5 GHz. The code is written in Fortran 77 and is run in single precision. We examine the convergence behavior based on the number of preconditioned iterations and some theoretical facts (such as the condition number).

To demonstrate the performance of our preconditioned BiCG solver, we calculate the RCS of different conducting geometries with and without coating. The geometries considered include plates, spheres, and pipes (see Table 1). The mesh size for all the test structures are about one tenth of a wavelength.

Here are the explanations of the notations shown in the tables with numerical data and in some figures.

- *precond*: the preconditioner used with the BiCG method:
  - °**NONE:** no preconditioner;
  - °**BLOCK:** the block diagonal preconditioner ;
  - °**SVDB:** the SVD stabilized block preconditioner.
- level: the number of levels used in the MLFMA.
- $\varepsilon$: the threshold value used to update the singular values in Algorithm 3.1 (static strategy).
- $it_{num}$: the number of (preconditioned) BiCG iterations.
- $it_{cpu}$: the CPU time in seconds for the iteration phase.
- Condition#: the condition number of each block.

For the class of the problems we tested, the block diagonal preconditioner can improve the BiCG convergence only in the P3C case. In the remaining three other cases, the block diagonal preconditioner actually hampers the BiCG convergence. Except for the sphere case, SVDB preconditioner can reduce the BiCG iteration steps and the total iteration CPU time compared with the BiCG solver without a preconditioner.

Table 1: Information about the matrices used in the experiments (all length units are in λ0, the wavelength in free-space)

| cases | level | unknowns | matrices | nonzeros | target size and description |
|---|---|---|---|---|---|
| P1C | 4 | 1,416 | $A$ | 2,005,056 | Dielectric plate over conducting plate<br>2.98824×2×0.1<br>Frequency=0.2GHz |
| | | | $A_D$ | 66,384 | |
| R5C | 6 | 14,950 | $A$ | 223,502,500 | Conduction pipe with 4 dielectric coating rings inside,<br>36×3.86236×3.87<br>Frequency=5GHz |
| | | | $A_D$ | 607,838 | |
| P3C | 7 | 100,800 | $A$ | 10,160,640,000 | Antenna array<br>Array size: 22.25×22.25<br>Frequency=0.3GHz |
| | | | $A_D$ | 3,571,808 | |
| S2C | 4 | 10,800 | $A$ | 116,640,000 | Large conduction sphere<br>5×5×5<br>Frequency=0.3GHz |
| | | | $A_D$ | 555,200 | |

Table 2 and Fig. 1 show that the SVDB preconditioner is very efficient in case the geometry is a plate. We also find that the larger static threshold we choose, the less number of iterations we need. But not all the computed solutions are correct. From

Fig.1 (b), we can find that the solution of using mean value as the threshold is not correct. So, there should be a range to choose the threshold. If we choose the threshold beyond this range, we get false convergence or even no convergence at all.

Table 2: Numerical data of the P1C case

| *precond* | $\varepsilon$ | $it_{num}$ | $\mathrm{it}_{cpu}$ |
|---|---|---|---|
| NONE | - | 1013 | 45.75 |
| BLOCK | - | 1319 | 59.94 |
| SVDB | Mean value | 45 | 2.04 |
| | 10.0 | 557 | 25.82 |
| | 15.0 | 467 | 21.52 |
| | 20.0 | 400 | 18.05 |

Table 3: Numerical data of the R5C case

| *precond* | $\varepsilon$ | $it_{num}$ | $\mathrm{it}_{cpu}$ |
|---|---|---|---|
| NONE | - | 445 | 478.83 |
| BLOCK | - | 2000 | 2012.60 |
| SVDB | 6.0e-3 | 174 | 184.00 |
| | 7.0e-3 | 75 | 79.82 |
| | 8.0e-3 | 46 | 49.12 |

Fig. 1 shows the solutions of the P1C case using none preconditioner, block diagonal preconditioner, and SVDB preconditioner. Fig. 2 shows the solution of the R5C case and Fig.3 shows that of the S2C case. From these three figures we see that the RCS of the SVDB preconditioners for the S2C and R5C cases are much closer to the exact solution compared with the block diagonal preconditioner.

Table 4: Numerical data of the P3C case

| *precond* | $\varepsilon$ | $it_{num}$ | $\mathrm{it}_{cpu}$ |
|---|---|---|---|
| NONE | - | 2000 | 5892.03 |
| BLOCK | - | 1101 | 3257.47 |
| SVDB | 0.15 | 416 | 1072.81 |
| | 0.20 | 359 | 930.66 |
| | 0.25 | 289 | 733.48 |

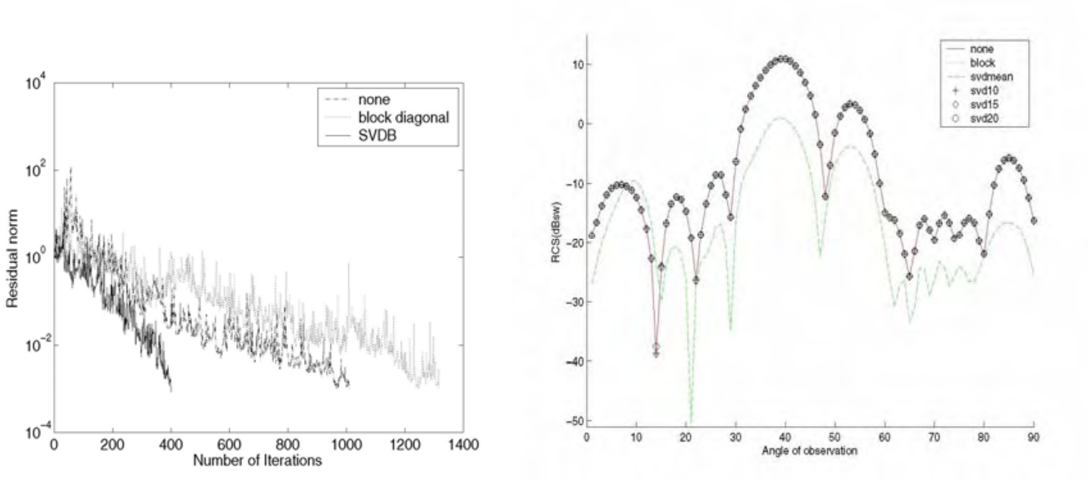
Table5: Numerical data of the S2C case

| *precond* | $\varepsilon$ | $it_{num}$ | $\mathrm{it}_{cpu}$ |
|---|---|---|---|
| NONE | - | 320 | 264.98 |
| BLOCK | - | 2000 | 1965.15 |
| SVDB | 1.50 | 1365 | 1331.31 |
| | 2.00 | 798 | 786.38 |
| | 2.50 | 340 | 331.00 |

Table 6 shows us some properties of the individual block in the P1C case, such as the largest and smallest singular values, the condition number for each block, and the condition number after applying SVD with threshold. The threshold we use in this table to update the singular value is 15. Generally, we hope our preconditioned linear system has smaller condition number than the original one.

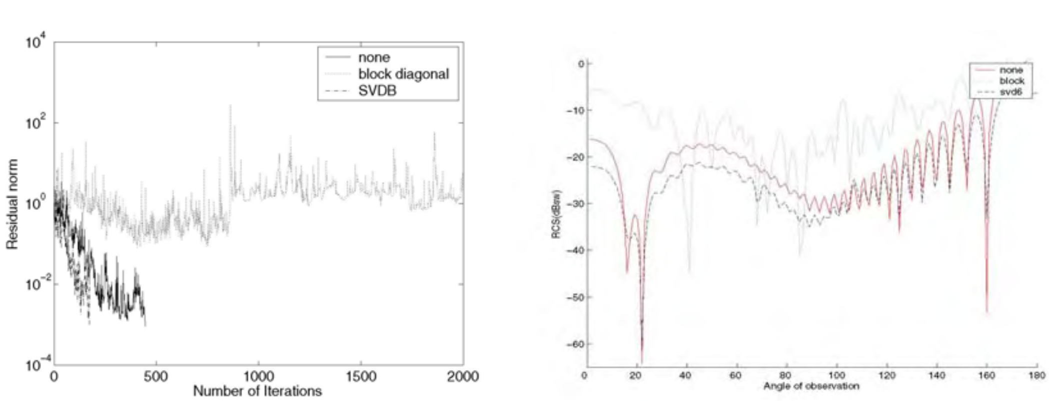
Table 6: The singular values and condition number of some blocks for the P1C case.

| Block | largest singular value | Smallest singular value | Condition # | Condition# after SVD (static) |
|---|---|---|---|---|
| $A_1$ | 952.8143 | 7.406 | 128.6543 | 63.52089 |
| $A_9$ | 1625.496 | 7.043 | 230.7880 | 108.3664 |
| $A_{17}$ | 1724.973 | 22.09911 | 78.05623 | 78.05623 |
| $A_{25}$ | 697.4736 | 8.50674 | 81.99069 | 46.49824 |
| $A_{32}$ | 952.8133 | 7.40594 | 128.6553 | 63.52089 |



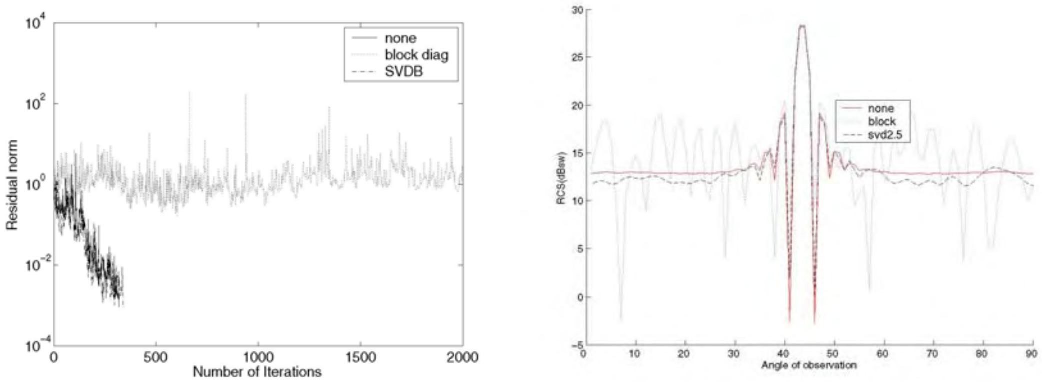
(a) (b)

Fig. 1. P1C case. (a) Convergence history comparison. (b) The solution accuracy



(a) (b)

Fig. 2. R5C case. (a) Convergence history comparison. (b) The solution accuracy



(a) (b)

Fig. 3. S2C case. (a) Convergence history comparison. (b) The solution accuracy

## 5. Conclusions

We proposed a preconditioning technique based on the SVD stabilization of the block diagonal submatrices in the MLFMA. This SVD stabilized block preconditioner is easily constructed and also can be parallelized.

The standard block diagonal preconditioner is not robust and inefficient to solve the dense linear system arising from the combined hybrid integral formulation of electromagnetic scattering problem. We conducted a few numerical tests to show that the SVD stabilized block preconditioner is effective in solving some electromagnetic scattering problems. We also performed comparison for the block diagonal preconditioner, SVD stabilized block preconditioner, and none preconditioner.

Our numerical results indicate that the SVD stabilized block preconditioner can work more efficient than the standard block diagonal preconditioner in every case we tested. It also reduces the CPU time for these cases. We introduced two strategies to choose the threshold values for generating the SVD stabilized preconditioner. The RCS results show that the static strategy works quite well for plate case.

## References

[1] O. AXELSSON, *Iterative Solution Methods*, Cambridge University Press, Cambridge, 1994.

[2] D. KALMAN, *A singularly valuable decomposition: The SVD of a matrix*, Coll. Math. J. **27(1)** (1996) 2-23.

[3] B. CARPENTIERI, I. S. DUFF, L. GIRAUD AND M. MAGOLU MONGA MADE, *Sparse symmetric preconditioners for dense linear systems in electromagnetism*, Numer. Linear Alg. Appl., **11** (2004) 753-771.

[4] T. F. CHAN AND K. CHEN, *On two variants of an algebraic wavelet preconditioner*, SIAM J. Sci. Comput., **24** (2002) 260-283.

[5] K. CHEN, *On a class of preconditioning methods for the dense linear systems from boundary elements*, SIAM J. Sci. Comput., **20(2)** (1998) 684-698.

[6] W. C. CHEW, J. M. JIN, E. MIDIELSSEN AND J. M. SONG, *Fast and efficient algorithms in computational electromagnetics*, Artech House, Boston, 2001.

[7] R. COIFMAN, V. ROKHLIN AND S. WANDZURA, *The fast multipole method for the wave equation: a pedestrian prescription*, IEEE Antennas Propagat. Mag., **35(3)** (1993) 7-12.

[8] B. M. KOLUNDZIJA, *Electromagnetic modeling of composite metallic and dielectric structures*, IEEE Trans. Micro. Theory Tech., **47(7)** (1999) 1021-1032.

[9] C. LANCZOS, *Solution of systems of linear equations by minimized iterations*, J. Res. Natl. Bur. Stand., **49** (1952) 33-53.

[10] J. LEE, J. ZHANG AND C. C. LU, *Incomplete LU preconditioning for large scale dense complex linear systems from electromagnetic wave scattering problems*, J. Comput. Phys., **185** (2003) 158-175.

[11] J. LEE, J. ZHANG AND C. C. LU, *Sparse inverse preconditioning of multilevel fast multipole algorithm for hybrid integral equations in electromagnetics*, IEEE Trans. Antennas Propagat., **52(9)** (2004) 2277-2287.

[12]J. LEE, J. ZHANG AND C. C. LU, *Performance of preconditioned Krylov iterative methods for solving hybrid integral equations in electromagnetics*, Appl. Comput. Electromagn. Society J., **18(4)** (2003) 54-61.

[13]C. C. LU AND W. C. CHEW, *A multilevel algorithm for solving a boundary integral equation of wave scattering*, IEEE Trans. Micro. Opt. Tech. Lett., **7(10)** (1994) 466-470.

[14]C. C. LU AND W. C. CHEW, *A coupled surface-volume integral equation approach for the calculation of electromagnetic scattering from composite metallic and material targets*, IEEE Trans. Antennas Propagat., **48(12)** (2000) 1866-1868.

[15]Y. SAAD AND J. ZHANG, *BILUTM: A domain-based multilevel block ILUT preconditioner for general sparse matrices*, SIAM J. Matrix Anal. Appl., **21(1)** (1999) 279-299.

[16]S. M. RAO, D. R. WILTON AND A. W. GLISSON, *Electromagnetic scattering by surface of arbitrary shape*, IEEE Trans. Antennas Propagat., **AP-30(3)** (1982) 409-418.

[17]V. ROKHLIN, *Rapid solution of integral equations of scattering theory in two dimensions*, J. Comput. Phys., **86(2)** (1990) 414-439.

[18]Y. SAAD, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, New York, NY, 1996.

[19]T. K. SHARK, S. M. RAO AND A. R. DJORDIEVIC, *Electromagnetic scattering and radiation from finite microstrip structures*, IEEE Trans. Micro. Opt. Tech., **38(11)** (1990) 1568-1575.

[20]J. M. SONG, C. C. LU AND W. C. CHEW, *Multilevel fast multipole algorithm for electromagnetic scattering by large complex objects*, IEEE Trans. Antennas Propagat. **AP-45(10)** (1997) 1488-1493.

[21]Y. WANG, J. LEE AND J. ZHANG, *A short survey on preconditioning techniques for large scale dense complex linear systems in electromagnetics,* Accepted by Int. J. Comput. Math.

[22]T. VAUPEL AND V. HANSEN, *Electrodynamic analysis of combined microstrip and coplanar/slotline structure with 3-D components based on a surface/volume integral equation approach,* IEEE. Trans. Micro. Theory Tech., **47(9)** (1999) 1150-1155.

[23]J. ZHANG, *Sparse approximate inverse and multilevel block ILU preconditioning techniques for general sparse matrices,* Appl. Numer. Math., **35(1)** (2000) 67-86.

[24]M. DEGIORGI, G. TIBERI, A. MONORCHIO, G. MANARA AND R. MITTRA, *An SVD-based method for analyzing electromagnetic scattering from plates and faceted bodies using physical optics bases,* Proceeding of IEEE Antennas and Propagation Society International Symposium, **1(A)** (2005) 147-150.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# New Parallel Symmetric SOR Preconditioners by Multi-Type Partitioning

**Dexuan Xie**[1]

[1] *Department of Mathematical Sciences, University of Wisconsin,*
*Milwaukee, WI 53201, USA*

emails: `dxie@uwm.edu`

### Abstract

This paper proposes a new parallel symmetric successive over-relation (PSSOR) preconditioner by a multi-type partition technique. It then proves that the PSSOR preconditioner is symmetric and positive-definite and equivalent to a SSOR preconditioner using a special ordering called the multi-type ordering. Thus, the PSSOR preconditioner can be effectively applied to the preconditioned conjugate gradient method (PCG), and can be analyzed under the framework of the classical SSOR theory. Numerical tests on an anisotropic model problem show that with the PSSOR preconditioner, PCG can have a much faster rate of convergence and a better parallel performance than with the red-black SSOR preconditioner. Moreover, the PSSOR preconditioner is shown numerically to have a condition number nearly equal to that of the sequential SSOR preconditioner when the anisotropic ratio is large enough.

*Key words: preconditioned conjugate gradient, SSOR, domain decomposition*
*MSC 2000: AMS 65Y05 (65F10)*

## 1 Introduction

The symmetric successive over-relaxation (SSOR) preconditioner is often used in engineering and scientific computing due to its simplicity in implementation [5, 7, 13]. It is known that the effects of the SSOR preconditioner on the convergence rate of the preconditioned conjugate gradient method (PCG) depends on the ordering of the mesh points on which a finite element or finite difference approximation to an elliptic boundary value problem is defined. The natural ordering usually results in the fastest convergence rate, but is difficult to implement in parallel. The red-black (or multicolor) ordering can overcome this difficulty, but seriously degrades the convergence rate of PCG compared to the natural ordering. To develop efficient parallel SSOR preconditioners, several other parallel orderings were proposed, which include the wavefront ordering [2], the local column-wise ordering [8], the many-color ordering [6], and domain decomposition orderings [3, 9, 10]. On current MIMD parallel computers, however, it is only domain decomposition orderings that were found to lead to more effective parallel SSOR preconditioners than the red-black SSOR preconditioner [9].

Recently, a new mesh domain partition and ordering, called the multi-type partition and ordering, was proposed and applied to define a new block parallel SOR (BPSOR) method [12]. The BPSOR method was shown to have the same asymptotic convergence rate as the corresponding sequential block SOR method if the coefficient matrix of the block linear system is "consistently ordered". In particular, three particular multi-type orderings are proposed based on strip and block mesh partitions, which lead to three effective BPSOR methods for solving the five-point like linear systems (in 2D) and the seven-point like linear systems (in 3D). In the point form, BPSOR is reduced to the PSOR method (a point parallel SOR method by mesh domain partitioning proposed in [11]). Obviously, the symmetric BPSOR (SBPSOR) method can be defined in the same way as the SSOR method is defined [13]. A new symmetric parallel SOR (PSSOR) preconditioner is then well defined by one iteration of SBPSOR with an initial guess of zero. This paper intends to give the PSSOR preconditioner a general mathematical formulation and to present a numerical study on its effect on the convergence rate of the PCG method. In particular, it proves that the PSSOR preconditioner is symmetric and positive-definite (SPD) if the coefficient matrix of the linear system is SPD. It also proves that the PSSOR preconditioner is equivalent to the SSOR preconditioner using the multi-type ordering. As a result, the analysis of the PSSOR preconditioner can be done within the framework of the classic SOR theory [13].

To study the numerical behaviors of the PSSOR preconditioner, two particular PSSOR preconditioners – the 2-type and 3-type PSSOR preconditioners – are constructed for a linear system arising from a five-point like finite difference approximation to a simple anisotropic model problem. Their condition numbers are calculated directly for several small model problems by MATLAB, demonstrating that the 2-type and 3-type PSSOR preconditioners can have a much smaller condition number than the red-black SSOR preconditioner. In the 2-type case, the PSSOR preconditioner is found to have a condition number nearly equal to that of the sequential SSOR preconditioner for a large anisotropic ratio. Hence, the PSSOR preconditioner is expected to have the same effect as the sequential SSOR preconditioner on the convergence of PCG.

Finally, the PCG using the PSSOR preconditioner for solving the anisotropic model problem was implemented on a MIMD parallel computer (the SGI Origin 2000 at the University of Wisconsin-Milwaukee) to investigate the parallel performance of the PSSOR preconditioner. Numerical results show that with the PSSOR preconditioner, PCG has a much faster convergence rate and much better parallel performance than with the red-black SSOR preconditioner. They also confirm that in the case of a large anisotropic ratio, the PCG using the 2-type PSSOR preconditioner has the same rate of convergence as the PCG using the sequential SSOR preconditioner. In the case of the grid size $h = 1/517$, a speedup of 12 was obtained for the 2-type PSSOR preconditioner on the 16 processors of the SGI Origin 2000.

The remainder of the paper is organized as follows: Section 2 introduces the multi-type partition and the related block linear system. Section 3 defines and analyzes the PSSOR preconditioner. Section 4 presents a condition number analysis to the 2-type and 3-type PSSOR preconditioners for solving the model problem. Finally, the parallel performance of the PCG using the PSSOR preconditioner is studied in Section 5.

## 2 The multi-type partition

Let the linear system $Au = f$ arise from a finite element or finite difference approximation to an elliptic boundary value problem with mesh domain $\Omega_h$. Here $A$ denotes a SPD and consistently ordered matrix with entries $a_{ij}$. If $a_{ij} \neq 0$ with $i \neq j$, mesh point $i$ is said to be connected to mesh point $j$. Two subdomains of $\Omega_h$ are said to be connected if at least two of their mesh points are connected. It is assumed that $p$ processors of a large scale MIMD parallel computer are requested to implement the PCG for solving the linear system in parallel.

To define the multi-type partition, the mesh domain $\Omega_h$ is partitioned into $p$ disjoint subdomains, $\Omega_{h,j}$ for $j = 1, 2, \ldots, p$, such that each subdomain is only connected to its neighboring subdomains. The mesh points of each subdomain are then grouped into $t$ different types according to the following three criterions: (i) connected types must be adjacent; (ii) no adjacent types are the same type; and (iii) every interior type has at least one adjacent type located in a neighboring subdomain. Here $t$ is a positive integer determined by the connection information among neighboring subdomains. It is clear that $t$ is less than or equal to the number of neighboring subdomains. Thus, the multi-type partition with $t$ types, which is also referred to as the $t$-type partition, is obtained.

Let $\Omega^i_{h,\mu}$ denote the $i$th type of subdomain $\mu$ for $\mu = 1, 2, \ldots, p$ and $i = 1, 2, \ldots, t$. When these $tp$ type subdomains are ordered in the natural ordering (i.e., from $i = 1$ to $t$ for each $\mu$ from 1 to $p$), the linear system $Au = f$ can be written in the block matrix form with

$$u = \begin{pmatrix} \mathcal{U}_1 \\ \mathcal{U}_2 \\ \vdots \\ \mathcal{U}_p \end{pmatrix}, \quad \mathcal{U}_\mu = \begin{pmatrix} \mathcal{U}^1_\mu \\ \mathcal{U}^2_\mu \\ \vdots \\ \mathcal{U}^t_\mu \end{pmatrix}, f = \begin{pmatrix} \mathcal{F}_1 \\ \mathcal{F}_2 \\ \vdots \\ \mathcal{F}_p \end{pmatrix}, \quad \mathcal{F}_\mu = \begin{pmatrix} \mathcal{F}^1_\mu \\ \mathcal{F}^2_\mu \\ \vdots \\ \mathcal{F}^t_\mu \end{pmatrix}, \tag{1}$$

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1p} \\ A_{21} & A_{22} & \cdots & A_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ A_{p1} & A_{p2} & \cdots & A_{pp} \end{bmatrix}, \quad A_{\mu\mu} = \begin{bmatrix} A^{11}_{\mu\mu} & A^{12}_{\mu\mu} & \cdots & A^{1t}_{\mu\mu} \\ A^{21}_{\mu\mu} & A^{22}_{\mu\mu} & \cdots & A^{2t}_{\mu\mu} \\ \vdots & \vdots & \ddots & \vdots \\ A^{t1}_{\mu\mu} & A^{t2}_{\mu\mu} & \cdots & A^{tt}_{\mu\mu} \end{bmatrix}, \tag{2}$$

$$\text{and } A_{\mu\nu} = \begin{cases} \begin{bmatrix} 0 & 0 & \cdots & 0 \\ A_{\mu\nu}^{21} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ A_{\mu\nu}^{t1} & \cdots & A_{\mu\nu}^{t,t-1} & 0 \end{bmatrix} & \text{for } \mu < \nu, \\ \begin{bmatrix} 0 & A_{\mu\nu}^{12} & \cdots & A_{\mu\nu}^{1t} \\ 0 & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & A_{\mu\nu}^{t-1,t} \\ 0 & 0 & \cdots & 0 \end{bmatrix} & \text{for } \mu > \nu. \end{cases} \tag{3}$$

Here $\mathcal{U}_\mu^i$ denotes a sub-vector of $u$ defined on the subdomain $\Omega_{h,\mu}^i$, $A_{\mu\mu}^{ii}$ is the sub-matrix of $A$ defined on $\Omega_{h,\mu}^i$, and $A_{\mu\nu}^{ij}$ with $\mu \neq \nu$ and $i \neq j$ is the sub-matrix that indicates the connection of $\Omega_{h,\mu}^i$ with $\Omega_{h,\nu}^j$. Clearly, if $\Omega_{h,\mu}^i$ is not adjacent to $\Omega_{h,\nu}^j$ with $\mu \neq \nu$ and $i \neq j$, then $A_{\mu\nu}^{ij} = 0$. In particular, it can be claimed that $A_{\mu\nu}^{ij} = 0$ for all $i \leq j$ if $\mu < \nu$ and for all $i \geq j$ if $\mu > \nu$. This gives the form of $A_{\mu\nu}$ with $\mu \neq \nu$ in (3).

## 3 The PSSOR preconditioner

Let $D_\mu$ and $L_\mu$ be the diagonal and strictly lower triangular matrices, respectively, satisfying $A_{\mu\mu} = D_\mu - L_\mu - L_\mu^T$ for $\mu = 1, 2, \ldots, p$. Here $T$ denotes the transpose of a matrix. The block matrix $A$ is split into the sum

$$A = D - B - B^T - N - N^T, \tag{4}$$

where $D$ and $B$ are two block diagonal matrices defined by

$$D = \mathrm{diag}(D_1, D_2, \ldots, D_p) \quad \text{and} \quad B = \mathrm{diag}(L_1, L_2, \ldots, L_p), \tag{5}$$

and $N$ is a strictly lower block triangular matrix defined by

$$N = -\begin{bmatrix} 0 & 0 & \cdots & 0 \\ A_{21} & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ A_{p1} & \cdots & A_{p(p-1)} & 0 \end{bmatrix}. \tag{6}$$

Based on the sum in (4), the PSSOR preconditioner is defined by

$$M = \frac{1}{\omega(2-\omega)}[D - \omega(B + N^T)]D^{-1}[D - \omega(B^T + N)], \tag{7}$$

where the relaxation parameter $\omega \in (0, 2)$.

If $\omega = 1$, the PSSOR preconditioner can be simplified as

$$M = [D - B - N^T]D^{-1}[D - B^T - N)], \tag{8}$$

which is called the parallel symmetric Gauss-Seidel preconditioner.

**Theorem 1** *If the matrix $A$ defined in (2) is symmetric and positive-definite, so is the PSSOR preconditioner $M$. Here $M$ is defined in (7).*

*Proof:* Set $\bar{R} = [D - \omega(B + N^T)]D^{-1}[D - \omega(B^T + N)]$. Since $M = \frac{1}{\omega(2-\omega)}\bar{R}$, one only needs to show that $\bar{R}$ is SPD .

Clearly, $\bar{R}$ is symmetric. From (5) it can be seen that $D$ consists of the main diagonal entries of $A$, which are positive since $A$ is positive-definite. Thus, $x^T Dx > 0$ and $x^T D^{-1}x > 0$ for nonzero vector $x$.

Set $\Lambda = D - \omega(B^T + N)$. It is clear that $x^T \Lambda x = x^T \Lambda^T x$. From (4) it can follow that $\Lambda + \Lambda^T = (2 - \omega)D + \omega A$. Hence, for any nonzero vector $x$, and $\omega \in (0, 2)$,

$$x^T \Lambda x = \frac{1}{2}x^T(\Lambda + \Lambda^T)x = \frac{1}{2}[(2 - \omega)x^T Dx + \omega x^T Ax] > 0.$$

Thus, the vector $y$ defined by $y = \Lambda x$ is not zero if $x \neq 0$. Hence, for any nonzero vector $x$,

$$\begin{aligned} x^T \bar{R}x &= x^T[D - \omega(B + N^T)]D^{-1}[D - \omega(B^T + N)]x \\ &= x^T \Lambda^T D^{-1}\Lambda x = y^T D^{-1}y > 0. \end{aligned}$$

This proves that $\bar{R}$ is positive-definite. The proof is completed.

Since $M$ is SPD, the inverse of $M$ exists and can be found as below:

$$M^{-1} = \omega(2 - \omega)[D - \omega(B^T + N)]^{-1}D[D - \omega(B + N^T)]^{-1}. \tag{9}$$

In application, the solution of $Mz = r$, $z = M^{-1}r$, is found as one iteration of the symmetric BPSOR method with an initial guess of zero. Here each symmetric BPSOR iteration contains two half-iterations: the first one is the BPSOR method and the other one is the BPSOR method using the reverse order, which is also called the backward BPSOR method. In fact, from [12] it is known that the BPSOR method has the following iterative expression

$$\begin{aligned} u^{(k+1)} &= [D - \omega(B + N^T)]^{-1}[(1 - \omega)D + \omega(B^T + N)]u^{(k)} \\ &+ \omega[D - \omega(B + N^T)]^{-1}r, \quad k = 0, 1, 2, \ldots, \end{aligned} \tag{10}$$

where $u^{(0)}$ is an initial guess. By reversing the ordering that is used in implementing each BPSOR iteration, it is easy to obtain the backward BPSOR iterative expression in the form

$$\begin{aligned} u^{(k+1)} &= [D - \omega(B^T + N)]^{-1}[(1 - \omega)D + \omega(B + N^T)]u^{(k)} \\ &+ \omega[D - \omega(B^T + N)]^{-1}r, \quad k = 0, 1, 2, \ldots. \end{aligned} \tag{11}$$

With $u^{(0)} = 0$, the first BPSOR iterate $u^{(1)} = \omega[D - \omega(B + N^T)]^{-1}r$. Then, one backward BPSOR iteration starting at $u^{(1)}$ immediately gives the solution of $Mz = r$.

The *pt* type subdomains $\{\Omega^i_{h,\mu}\}$ of the $t$-type partition can also be ordered from $\mu = 1$ to $p$ for each value of type $i$ from 1 to $t$ while the original ordering is retained within each type subdomain. Such an ordering is called the multi-type ordering. In the multi-type ordering, the sub-vectors $\{\mathcal{U}_j^i\}$ are reordered in the form

$$\hat{u} = \begin{pmatrix} \hat{\mathcal{U}}_1 \\ \hat{\mathcal{U}}_2 \\ \vdots \\ \hat{\mathcal{U}}_t \end{pmatrix} \text{ with } \hat{\mathcal{U}}_i = \begin{pmatrix} \mathcal{U}_1^i \\ \mathcal{U}_2^i \\ \vdots \\ \mathcal{U}_p^i \end{pmatrix}.$$

Clearly, there exists a permutation matrix, $P$, such that $\hat{u} = Pu$ with $u$ being given in (1). In terms of $P$, the reordered linear system by the multi-type ordering can be expressed as $\hat{A}\hat{u} = \hat{f}$ with $\hat{A} = PAP^T, \hat{u} = Pu$, and $\hat{f} = Pf$. Thus, the SSOR preconditioner using the $t$-type ordering, which is also called the $t$-type SSOR preconditioner, can be obtained as below:

$$\hat{M} = \frac{1}{\omega(2-\omega)}[\hat{D} - \omega\hat{L}]\hat{D}^{-1}[\hat{D} - \omega\hat{L}^T], \tag{12}$$

where $\hat{D}$ and $\hat{L}$ are diagonal and strictly lower triangular matrices, respectively, satisfying $\hat{A} = \hat{D} - \hat{L} - \hat{L}^T$.

**Theorem 2** *If $M$ and $\hat{M}$ are the PSSOR preconditioner and the $t$-type SSOR preconditioner defined in (7) and (12), respectively, then there exists a permutation matrix $P$ such that*

$$\hat{M} = PMP^T. \tag{13}$$

*Proof:* Clearly, there exists the permutation matrix $P$ such that $\hat{u} = Pu$. In terms of $P$, it is easy to see that $\hat{D} = PDP^T$. With $u^{(0)} = 0$, the first BPSOR iterate $u^{(1)}$ for solving $Au = f$ becomes

$$u^{(1)} = \omega[D - \omega(B + N^T)]^{-1}f.$$

Multiplying the above expression by $P$ from the left-hand side and using the identities $P^TP = I$, $Pf = \hat{f}$, and $PDP^T = \hat{D}$ give that $\hat{u}^{(1)} = \omega[\hat{D} - \omega P(B + M)P^T]^{-1}\hat{f}$. On the other hand, since $\hat{u}^{(0)} = Pu^{(0)} = 0$, the first SOR iterate for solving the reordered linear system $\hat{A}\hat{u} = \hat{f}$ becomes $\hat{u}^{(1)} = \omega[\hat{D} - \omega\hat{L}]^{-1}\hat{f}$. Combining these two expressions of $\hat{u}^{(1)}$ yields the identity

$$[\hat{D} - \omega P(B + N^T)P^T]^{-1}\hat{f} = [\hat{D} - \omega\hat{L}]^{-1}\hat{f} \quad \text{for all nonzero vector } \hat{f}.$$

From the above identity it follows that $[\hat{D} - \omega P(B+N^T)P^T]^{-1} = [\hat{D} - \omega\hat{L}]^{-1}$, which can be simplified as $\hat{L} = P(B + N^T)P^T$. Thus, combing it with $PDP^T = \hat{D}$ and $P^T = P^{-1}$ gives

$$[\hat{D} - \omega\hat{L}]\hat{D}^{-1}[\hat{D} - \omega\hat{L}^T] = P[D - \omega(B + N^T)]D^{-1}[D - \omega(B^T + N)]P^T.$$

Multiplying both sides of the above identity by the constant $1/[\omega(2-\omega)]$ yields $\hat{M} = PMP^T$. This completes the proof.

# 4 PSSOR preconditioners for an anisotropic model problem

A simple anisotropic model problem is given by

$$\left\{ \begin{array}{rcll} -(au_{xx}+bu_{yy}) & = & f(x,y) & \text{in } \Omega, \\ u & = & 0 & \text{on } \partial\Omega. \end{array} \right. \tag{14}$$

where $a$ and $b$ are positive constants with $a \geq b$, $\Omega = (0,1) \times (0,1)$, and $\partial\Omega$ denotes the boundary of $\Omega$. By the five-point finite difference formula, the model problem can be approximated as the following linear system

$$2(1+\frac{b}{a})u_{ij} - (u_{i+1,j}+u_{i-1,j}) - \frac{b}{a}(u_{i,j+1}+u_{i,j-1}) = \frac{h^2}{a} f_{ij}, \tag{15}$$

where $i,j = 1,2,\ldots,n-1, h = 1/n$ with $n > 1$, $f_{ij} = f(ih,jh)$, $u_{ij}$ denotes an approximation of the solution $u(ih,jh)$, and $u_{i0} = u_{in} = u_{0i} = u_{ni} = 0$ for $i = 0,1,2,\ldots,n$.

In the 2-type ordering ($t = 2$), the coefficient matrix $A$ of (2) becomes a tridiagonal block matrix in the form

$$A = \begin{bmatrix} A_{11} & A_{12} & & \\ A_{21} & A_{22} & \ddots & \\ & \ddots & \ddots & A_{p-1,p} \\ & & A_{p,p-1} & A_{pp} \end{bmatrix} \text{ with } A_{\mu\mu} = \begin{bmatrix} A^{11}_{\mu\mu} & A^{12}_{\mu\mu} \\ A^{21}_{\mu\mu} & A^{22}_{\mu\mu} \end{bmatrix}, \tag{16}$$

$$A_{\mu,\mu+1} = \begin{bmatrix} 0 & 0 \\ A^{21}_{\mu,\mu+1} & 0 \end{bmatrix} \quad \text{and} \quad A_{\mu+1,\mu} = A^T_{\mu,\mu+1}.$$

For the model problem (15), it can be found that

$$A^{11}_{\mu\mu} = \mathcal{B}, \quad A^{12}_{\mu\mu} = -cI, \quad A^{21}_{\mu,\mu+1} = -cI,$$

$$A^{22}_{\mu\mu} = \begin{bmatrix} \mathcal{B} & -cI & & \\ -cI & \mathcal{B} & \ddots & \\ & \ddots & \ddots & -cI \\ & & -cI & \mathcal{B} \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} \alpha & -1 & & \\ -1 & \alpha & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & \alpha \end{bmatrix},$$

where $\alpha = 2(1+b/a), c = b/a$, $I$ is an $(n-1) \times (n-1)$ identity matrix, $\mathcal{B}$ is an $(n-1) \times (n-1)$ tridiagonal matrix, $A^{22}_{\mu\mu}$ is an $\ell \times \ell$ block matrix if $\Omega^2_{h,\mu}$ contains $\ell$ mesh lines, and $\mu = 1,2,\ldots,p$.

In the 3-type partition, the block matrix $A$ of (2) becomes a five-diagonal block matrix with nonzero entries lying on the main diagonal, the second off-diagonal, and the $(\tilde{m}+1)$th off-diagonal. For example, for $p = 4$, the block matrix $A$ has the form

$$A = \begin{bmatrix} \mathcal{A} & \mathcal{G} & \mathcal{H} & \mathbf{0} \\ \mathcal{G}^T & \mathcal{A} & \mathbf{0} & \mathcal{H} \\ \mathcal{H}^T & \mathbf{0} & \mathcal{A} & \mathcal{G} \\ \mathbf{0} & \mathcal{H}^T & \mathcal{G}^T & \mathcal{A} \end{bmatrix} \text{ with } \mathcal{A} = \begin{bmatrix} A^{11} & A^{12} & \mathbf{0} \\ A^{12^T} & A^{22} & A^{23} \\ \mathbf{0} & A^{23^T} & A^{33} \end{bmatrix}, \quad A^{11} = \alpha, \tag{17}$$

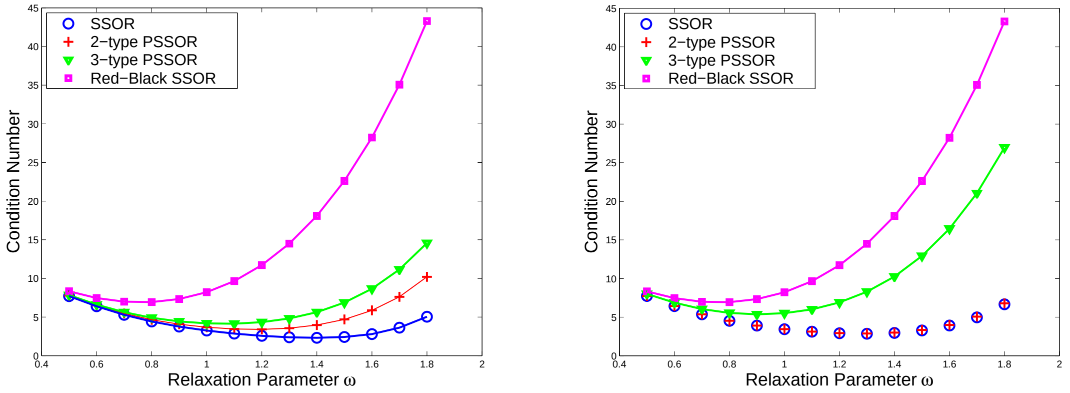


Figure 1: Comparison of the condition numbers of the PSSOR preconditioners with that of the sequential and red-black SSOR preconditioners for the model problem (15) with $h = 1/7$. The left plot for $a = b = 1$, and the right plot for $a = 10, b = 1$.

$$A^{22} = \begin{bmatrix} \alpha & -1 & 0 & 0 \\ -1 & \alpha & 0 & 0 \\ 0 & 0 & \alpha & -c \\ 0 & 0 & -c & \alpha \end{bmatrix}, \quad A^{33} = \begin{bmatrix} \alpha & -1 & -c & 0 \\ -1 & \alpha & 0 & -c \\ -c & 0 & \alpha & -1 \\ 0 & -c & -1 & \alpha \end{bmatrix},$$

$$A^{23} = \begin{bmatrix} -c & 0 & 0 & 0 \\ 0 & -c & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \qquad A^{12} = [-1 \quad 0 \quad -c \quad 0],$$

$\mathcal{G}$ is a $9 \times 9$ matrix with only three nonzero entries given by $g_{31} = g_{74} = g_{95} = -1$, and $\mathcal{H}$ is a $9 \times 9$ matrix with only three nonzero entries given by $h_{51} = h_{82} = h_{93} = -c$.

With the two matrix forms of $A$ given in (16) and (17), the two sequential SSOR preconditioners can be constructed, respectively, according to the formula $M = (D - \omega L)D^{-1}(D - \omega L^T)$, where $D$ is the diagonal matrix of $A$ and $L$ is the strictly lower triangular matrix such that $A = D - L - L^T$. Similarly, the 2-type and 3-type PSSOR preconditioners can also be constructed by formula (7). Furthermore, the red-black matrix form of $A$ can be obtained by reordering the matrix $A$ of (16) in the red-black ordering. Then, the red-black SSOR preconditioner can be constructed.

From the PCG theory [4] it is known that the effect of a preconditioner $M$ on the convergence rate of the PCG method can be studied directly by evaluating the condition number of $M^{-1}A$. The smaller the condition number, the faster the convergence rate [4]. For the model problem with $h = 1/7$, the condition numbers of the above four preconditioners were calculated on MATLAB, and reported in Figure 1.

Figure 1 shows that the 2-type and 3-type PSSOR preconditioners have much smaller condition numbers than the red-black SSOR preconditioner, while the SSOR preconditioner has the smallest condition number. Hence, the PSSOR preconditioner is more effective than the red-black SSOR preconditioner while the sequential SSOR

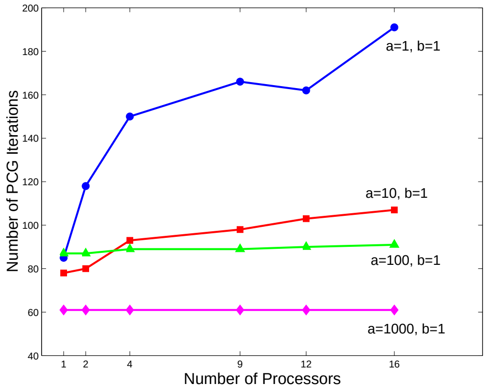


Figure 2: Convergence dependence of the PCG using the PSSOR preconditioner on the number of processors and the anisotropic ratio $a/b$.

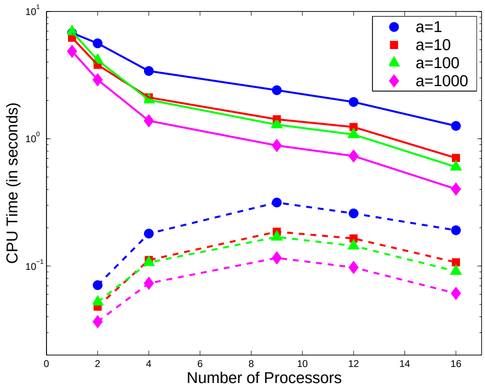


Figure 3: Parallel performance of the PCG using the PSSOR preconditioner. The dash lines indicate the total interprocessor data communication time.

preconditioner is the most effective among them. From the right plot (with $a = 10$ and $b = 1$) of Figure 1 it can be seen that the 2-type PSSOR preconditioner has the same condition number as the SSOR preconditioner. Figure 1 also shows that the condition number is a quadratic function of $\omega$. Hence, for each preconditioner, there exists an optimal value of $\omega$ at which the smallest condition number is reached.

## 5 Parallel performance of PSSOR preconditioner

To demonstrate the parallel performance of the PCG using the PSSOR preconditioner, numerical experiments were made on a MIMD parallel computer (the SGI Origin 2000 computer at the University of Wisconsin-Milwaukee, which has 16 R12000 400 MHz processors) for the model problem (15) with $f = 1.0$ and $h = 1/517$. Four anisotropic ratios $a/b$ set by $a = 1, 10, 100, 1000$ for $b = 1$ were considered. All the numerical tests used an initial guess of zero, and the same iteration stop rule in which the relative residue norm is less than $10^{-6}$. In addition, the optimal values of the relaxation parameter $\omega$ were used in these tests, which were determined by experiments.

The parallel program was written in Fortran 77 and MPI (the Message Passing Interface) [1]. The program was compiled using the optimization level $O2$. The CPU time was measured by the MPI function $MPI\_Wtime$, which returns the wall time in seconds. All calculations were done with double precision. The sequential PCG method using the SSOR preconditioner was implemented on one processor of the computer by using a sequential F77 program, where the F77 function $etime()$ was used to measure the CPU time.

Figure 2 shows that the PCG using the PSSOR preconditioner can have a faster convergence speed for a larger value of the ratio $a/b$. Also, it shows that the number of

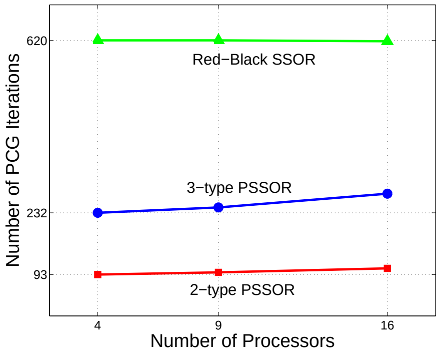


Figure 4: Convergence comparison of the PCG using the PSSOR preconditioner with the PCG using the red-black SSOR preconditioner.

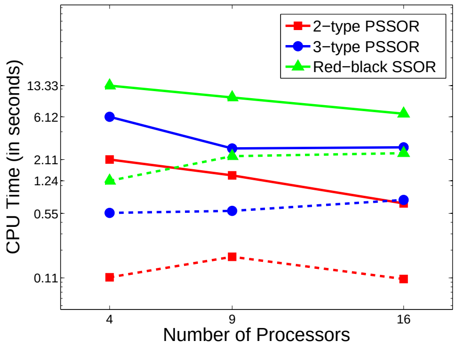


Figure 5: Parallel performance comparison of the PCG using the PSSOR preconditioner with the PCG using the red-black SSOR preconditioner.

processors has less effects on the convergence rate for a larger anisotropic ratio of $a/b$.

Figure 3 displays the parallel performances of the PCG using the PSSOR preconditioner. It shows that the total CPU time is a linear decreasing function of the number of processors. Compared to the total CPU time, the time consumed by interprocessor data communication is very small. A speedup of 12 was obtained on the 16 processors compared to the sequential PCG using the sequential SSOR preconditioner.

Figures 4 and 5 compare the parallel performance of the PCG using the 3-type PSSOR preconditioner with that of the PCG using the red-black SSOR preconditioner. Here $a = 10$ and $b = 1$. The block partitions of two by two, three by three and four by four blocks were used in the tests implemented on four, nine and sixteen processors, respectively. From these two figures it can be seen that the PSSOR preconditioner accelerated the convergence speed of PCG (in terms of the total number of PCG iterations determined by the iteration stop rule) for about 3 to 6 times compared to the red-black SSOR preconditioner, confirming that the PSSOR preconditioner is much more effective than the red-black SSOR preconditioner. Moreover, the PSSOR preconditioner reduced the total CPU time of the PCG using the red-black SSOR preconditioner by a factor of 3 to 6 on both calculation and interprocessor data communication. In addition, these two figures also show that the 2-type PSSOR preconditioner was more effective than the 3-type PSSOR preconditioner in improving the convergence rate and parallel performance of PCG.

## Acknowledgements

This project was supported by the National Science Foundation through grant DMS-0241236.

# References

[1] *MPI: A Message-Passing Interface Standard.* University of Tennessee, Knoxville, Tennessee, June 1995.

[2] C. Ashcraft and R. Grimes. On vectorizing incomplete factorization and SSOR preconditioners. *SIAM J. Sci. Statist. Comput.*, 9:122–151, 1988.

[3] C. Farhat. Multiprocessors in computational mechanics. In *Ph.D. thesis, Civil Engineering Department.* University of California, Berkeley, CA, 1986.

[4] G. H. Golub and C. F. van Loan. *Matrix Computations.* John Hopkins University Press, Baltimore, MD, Third edition, 1996.

[5] W. Hackbusch. *Iterative Solution of Large Sparse Systems of Equations.* Springer-Verlag, New York, 1994.

[6] D. Harrar and J. Ortega. Multicoloring with lots of colors. In *Proc. Third Internat. conference Supercomputing*, pages 1–6, Grete, Greece, 1989.

[7] Th. Lippert. Parallel SSOR preconditioner for lattice QCD. *Parallel Computing*, 25:1357–1370, 1999.

[8] R. Melhem. Towards efficient implementations of preconditioned conjugate methods on vector supercomputers. *Internat. J. Supercomput. Appl.*, 1:70–98, 1987.

[9] S. A. Stotland and J. M. Ortega. Orderings for parallel conjugate gradient preconditioners. *SIAM J. Sci. Comput.*, 18(3):854–868, 1997.

[10] H. A. van der Vorst. Large tridiagonal and block tridiagonal linear systems on vector and parallel computers. *Parallel Computing*, 5:45–54, 1987.

[11] D. Xie and L. Adams. New parallel SOR method by domain partitioning. *SIAM J. Sci. Comput*, 20:2261–2281, 1999.

[12] D. Xie. A new block parallel SOR method and its analysis. *SIAM J. Sci. Comput*, 27:1513-1533, 2006.

[13] D. M. Young. *Iterative Solution of Large Linear System.* Academic press, New York, 1971.

*Proceedings of the International Conference*
*on Computational and Mathematical Methods*
*in Science and Engineering, CMMSE 2007*
*Chicago, 20–23 June 2007.*

# A New Quadrature Using Integration Lattices

**Xiaoyan Zeng[1], Rong-Xian Yue[2] and Fred J. Hickernell[1]**

[1] *Department of Applied Mathematics, Illinois Institute of Technology*

[2] *Department of Applied Mathematics, Shanghai Normal University*

emails: zengxia@iit.edu, yue2@shtu.edu.cn, hickernell@iit.edu

## Abstract

In this paper we propose a new quadrature rule via approximating Fourier coefficients of variable transformed integrand with multidimensional integration lattice. The error analysis is derived for the proposed rule. A comparison is done between the new rule and the periodizing transformation methods. It shown that the new rule improves the periodizing transformation methods. In the last part of the paper, we also present a quadrature rule for weighted integrals whose integrand is in Banach space and we show it is semi-optimal.

*Key words: quadrature, integration lattice, transformed integrand*