PRÁCTICA 9: Integración

9.1 Resumen de Teoría

9.1.1 Introducción

El cálculo es la matemática del cambio. Como los ingenieros deben tratar en forma continua con sistemas y procesos que cambian, el cálculo es una herramienta esencial en nuestra profesión. En la esencia del cálculo están dos conceptos matemáticos relacionados: la diferenciación y la integración. En esta práctica vamos a centrarnos en los métodos numéricos para la integración.

Los métodos numéricos de integración pueden clasificarse en dos grupos:

- **Fórmulas de Newton Cotes:** Los que usan valores dados de la función $f(x)$ en abscisas equidistantes.
- **Fórmulas de Cuadratura Gaussiana:** Los que usan valores de $f(x)$ en abscisas desigualmente espaciadas, determinadas por ciertas propiedades de familias de polinomios ortogonales.

Los métodos de Newton Cotes son los tipos de integración numérica más comunes. Su estrategia es remplazar a la función complicada o de datos tabulados por un polinomio de interpolación que es fácil de integrar. Es decir:

$$I = \int_a^b f(x) \cong \int_a^b p_n(x)dx$$

En donde $p_n(x)$ es el polinomio de interpolación:

$$p_n(x) = a_0x + a_1x^2 + a_2x^n + \cdots + a_nx^n$$

Estas fórmulas de cuadratura de tipo interpolatorio, eligen los puntos de interpolación (llamados nodos de la fórmula) igualmente separados de una de las dos formas siguiente:

- Fórmulas cerradas: los límites de integración a y b son nodos de la fórmula.
- Fórmulas abiertas: ninguno de los límites de integración es nodo de la fórmula.

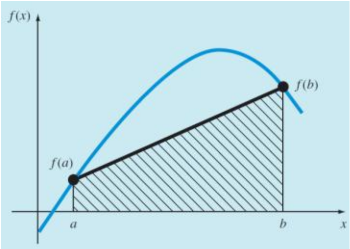
De las fórmulas de Newton-Cotes cerradas, las más sencillas son la de los Trapecios (2 nodos) y la de Simpson (3 nodos). A estos métodos se le llaman ***métodos de cuadratura*** porque “cuadratura” es la palabra clásica para denominar el cálculo de áreas.

9.1.2 La Regla del Trapecio

La regla del trapecio es la primera de las fórmulas cerradas de integración de Newton-Cotes. Corresponde al caso donde el polinomio de interpolación es de primer grado.

Geométricamente, la regla del trapecio es equivalente a aproximar el área del trapecio bajo la línea recta que une *f*(*a*) y *f*(*b*) en la figura. Recuerde que la fórmula para calcular el área de un trapezoide es la altura por el promedio de las bases. Por tanto:

$$\int_a^b f(x)dx \cong \frac{b-a}{2}[f(a)+f(b)]$$



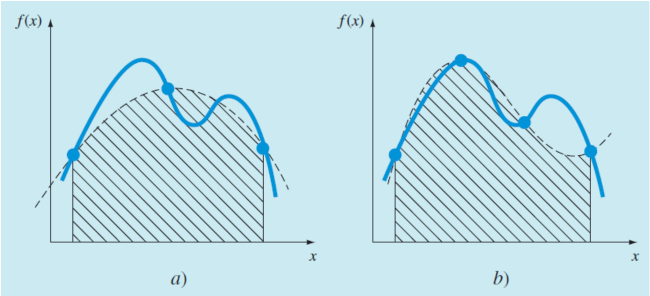
Es evidente que al usar un segmento de línea recta para aproximar la integral bajo una curva, se tiene un error que puede ser importante. Si llamamos E a la diferencia entre el valor exacto de la integral y el valor aproximado (es decir el *Error absoluto*), se demuestra que existe $c \in (a, b)$, tal que:

$$E = -\frac{(b-a)^3}{12} f''(c) \rightarrow |E| \leq \frac{(b-a)^3}{12} M , \qquad con\ M = \max |f''(x)|, \qquad a < x < b$$

con lo que la fórmula es de grado de precisión 1, es decir, que es exacta para los polinomios de grado menor o igual que 1. El error además depende del cubo de la longitud del intervalo. Si disminuimos la longitud del intervalo, el error disminuye rápidamente. Esto se puede lograr dividiendo el intervalo en subintervalos, aplicándolo en cada uno de ellos y sumando los resultados. El inconveniente es que aumentamos el número de operaciones.

9.1.3 La Regla de Simpson 1/3

Usar la fórmula de Simpson 1/3 para integrar una función en un intervalo, equivale a sustituir, dentro de la integral, la función a integrar por el polinomio de interpolación de grado dos que pasa por los puntos de la función de los extremos y el punto medio del intervalo. Es decir, sustituimos la función por una parábola e integramos, como vemos en la figura *a*).



Es pues una fórmula de Newton-Cotes cerrada con tres nodos:

$$x_0 = a, \quad x_1 = \frac{(a+b)}{2}, \quad x_2 = b$$

Se demuestra que la fórmula es:

$$\int_a^b f(x)dx \approx \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

Si llamamos de nuevo *E* al *Error absoluto*, esto es, a la diferencia entre el valor exacto de la integral y el valor aproximado, se demuestra que existe $c \in (a, b)$, tal que:

$$E = -\frac{(b-a)^5}{2880} f''''(c) \rightarrow |E| \leq \frac{(b-a)^5}{2880} M\,, \qquad con\ M = \max |f''''(x)|, \qquad a < x < b$$

Por tanto el grado de precisión es tres, y es exacta para los polinomios de grado menor o igual que 3. En otras palabras, ¡da resultados exactos para polinomios cúbicos aun cuando se obtenga de una parábola! Como vemos la regla de Simpson 1/3 es más exacta que la regla del trapecio. El error depende de la potencia quinta de la longitud del intervalo y de nuevo si disminuimos la longitud del intervalo, el error disminuye rápidamente, más que en el caso de los trapecios.

Nota. Existe una variante llamada regla de Simpson 3/8 que consiste en tomar dos nodos intermedios, igualmente espaciados entre *f*(*a*) y *f*(*b*). Los cuatro puntos se pueden unir mediante un polinomio de tercer grado, como se ve en la figura *b*).

Aunque podemos plantear fórmulas de integración basadas en polinomios de interpolación con más puntos, éstas no son demasiado interesantes porque, como los polinomios de interpolación de grado alto oscilan mucho, son propensas a errores en intervalos grandes. Una buena alternativa para disminuir el error es (como acabamos de ver) aumentar el número de nodos utilizando las *fórmulas compuestas*. Estas se obtienen dividiendo el intervalo [*a*, *b*] en *n* subintervalos y aplicando a cada uno de estos subintervalos una fórmula de cuadratura sencilla.

9.1.4 La Regla del Trapecio Compuesta

La Regla del Trapecio Compuesta se basa en dividir el intervalo de integración [*a*, *b*] en *n* subintervalos de igual longitud. Para ello, se denota:

$$x_i = x_0 + i\left(\frac{b-a}{n}\right) = x_0 + i * h, \qquad i = 0,1,2, \dots, n$$

De modo que con $h = \left(\frac{b-a}{n}\right)$:

$$x_0 = a;\ \ x_1 = a + h;\ \ x_2 = a + 2h;\ \ x_n = a + nh = b$$

y se hace:

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x)dx$$

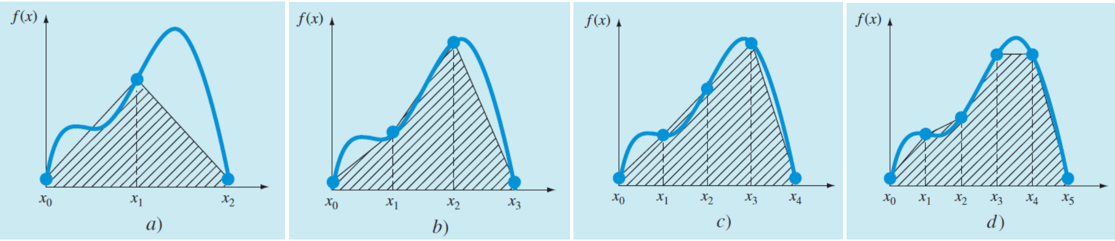
resolviendo cada una de las integrales anteriores, aplicando la regla del trapecio. Se demuestra que sustituyendo la regla del trapecio en cada integral se obtiene:

$$\int_a^b f(x)dx \approx \frac{b-a}{2n}\left[f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)\right] = \frac{h}{2}\left(f(x_0) + 2\sum_{i=1}^{n-1} f(x_i) + f(x_n)\right)$$

Se demuestra que existe $c \in (a, b)$, tal que el *Error absoluto E* (la diferencia entre el valor exacto de la integral y el valor aproximado) es:

$$E = -\frac{(b-a)^3}{12n^2} f''(c) \rightarrow |E| \leq \frac{(b-a)^3}{12n^2} M, \qquad con\ M = \max|f''(x)|, \qquad a < x < b$$

La siguiente figura ilustra la regla del trapecio compuesta en varios casos: *a*) Dos segmentos, *b*) tres segmentos, *c*) cuatro segmentos y *d*) cinco segmentos.



En el caso de la integración numérica el error de redondeo no aumenta al disminuir h y aumentar el número de operaciones. Por ello se dice que este procedimiento es estable cuando h tiende a 0, al contrario que la diferenciación numérica.

9.1.5 La Regla de Simpson Compuesta

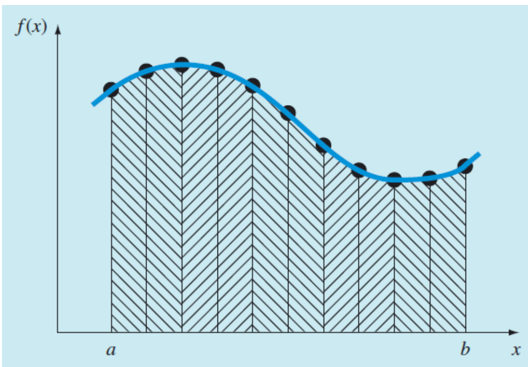
Así como en la regla del trapecio, la regla de Simpson se mejora al dividir el intervalo de integración en varios segmentos de un mismo tamaño. Se demuestra que sustituyendo la regla de Simpson en cada integral se obtiene:

$$\int_a^b f(x)dx \approx \frac{b-a}{3n}\left[f(a) + 4\sum_{i=1,3,5}^{n-1} f(x_i) + 2\sum_{i=2,4,6}^{n-2} f(x_i) + f(b) \right]$$

Si llamamos E a la diferencia entre el valor exacto de la integral y el valor aproximado, se demuestra que existe $c \in (a, b)$, tal que:

$$E = -\frac{(b-a)^5}{180\, n^4} f''''(c) \rightarrow |E| \leq \frac{(b-a)^5}{180\, n^4} M, \qquad con\ M = \max|f''''(x)|, \qquad a < x < b$$

La siguiente figura ilustra la regla de Simpson 1/3 compuesta. Obsérvese que el método se puede emplear sólo si el número de segmentos es par.



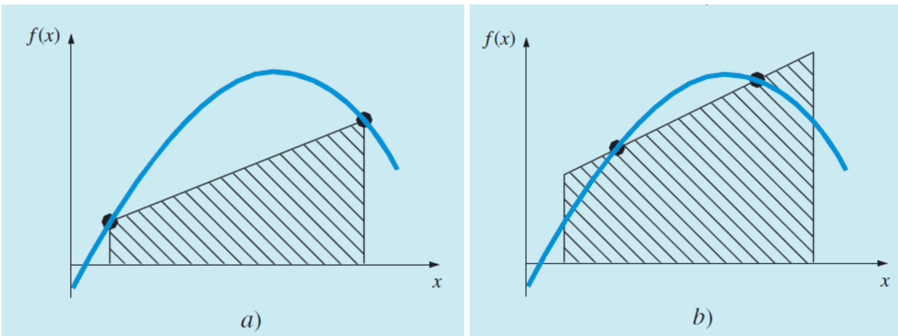
9.1.6 Fórmulas de Cuadratura Gaussiana

Para disminuir el error de las fórmulas de cuadratura, además de usar más puntos para generar el polinomio de interpolación que aproxima a la función (trapecios, dos puntos; Simpson, tres puntos...) y de dividir el intervalo de integración en subintervalos como se hace en las fórmulas compuestas, se usan otras estrategias como en las fórmulas de Cuadratura Gaussianas, donde los puntos no se escogen igualmente separados, sino que se escogen de forma que el grado de precisión sea máximo.

Las formulas usadas hasta la actualidad para aproximar integrales se basaban en polinomios de interpolación, considerando valores de la función igualmente espaciados fenómeno que conducen a cierta imprecisión. Con la finalidad de mejorar esta condición la cuadratura Gaussiana se usan puntos de evaluación, o nodos que no son igualmente espaciados. Se eligen nodos $x_1, x_{2,} x_3, \dots, x_n$ en el intervalo $[a, b]$ y coeficientes $c_1, c_{2,} c_3, \dots, c_n$ que minimicen el error que se espera obtener en la aproximación:

$$\int_a^b f(x)dx \approx \sum_{i=1}^{n} c_i f(x_i)$$

La siguiente gráfica compara la regla del trapecio con una mejor estimación de la integral que se consigue tomando el área bajo la línea recta que pasa por dos puntos intermedios. Estos puntos se ubican en una forma adecuada, de tal manera que se equilibran los errores positivo y negativo.



Veamos el método de integración de Gauss para el caso más simple de dos puntos en el intervalo [-1,1], es decir:

$$\int_{-1}^{1} f(x)dx \approx c_1 f(x_1) + c_2 f(x_2)$$

Tenemos cuatro incógnitas (c_1, c_2, x_1, y x_2) que deben calcularse y, en consecuencia, se requieren cuatro condiciones para determinarlas con exactitud. Es posible obtener esas condiciones al suponer que la fórmula proporciona resultados exactos cuando $f(x)$ es, 1, x, x^2 y x^3. Entonces tenemos:

$$c_1.1 + c_2.1 = \int_{-1}^{1} dx = 2; \qquad c_1.x_1 + c_2.x_2 = \int_{-1}^{1} xdx = 0$$

$$c_1.x_1^2 + c_2.x_2^2 = \int_{-1}^{1} x^2 dx = \frac{2}{3}; \qquad c_1.x_1^3 + c_2.x_2^3 = \int_{-1}^{1} x^3 dx = 0$$

De donde se obtiene:

$$c_1 = 1, c_2 = 1, x_1 = \frac{-\sqrt{3}}{3}, x_2 = \frac{\sqrt{3}}{3}$$

Es decir:

$$\int_{-1}^{1} f(x)dx \approx f\left(\frac{-\sqrt{3}}{3}\right) + f\left(\frac{\sqrt{3}}{3}\right)$$

Aparte de la fórmula de dos puntos descrita, se pueden desarrollar versiones con más puntos en la forma general:

$$\int_{-1}^{1} f(x)dx \approx c_1 f(x_1) + c_2 f(x_2) + \cdots + c_n f(x_n)$$

donde n = número de puntos. Los valores de las c y las x para fórmulas de hasta seis puntos se resumen en la tabla siguiente:

Puntos	Factor de ponderación	Argumentos de la función	Error de truncamiento
2	c_0 = 1.0000000	x_0 = −0.577350269	$\cong f^{(4)}(\xi)$
	c_1 = 1.0000000	x_1 = 0.577350269	
3	c_0 = 0.5555556	x_0 = −0.774596669	$\cong f^{(6)}(\xi)$
	c_1 = 0.8888889	x_1 = 0.0	
	c_2 = 0.5555556	x_2 = 0.774596669	
4	c_0 = 0.3478548	x_0 = −0.861136312	$\cong f^{(8)}(\xi)$
	c_1 = 0.6521452	x_1 = −0.339981044	
	c_2 = 0.6521452	x_2 = 0.339981044	
	c_3 = 0.3478548	x_3 = 0.861136312	
5	c_0 = 0.2369269	x_0 = −0.906179846	$\cong f^{(10)}(\xi)$
	c_1 = 0.4786287	x_1 = −0.538469310	
	c_2 = 0.5688889	x_2 = 0.0	
	c_3 = 0.4786287	x_3 = 0.538469310	
	c_4 = 0.2369269	x_4 = 0.906179846	
6	c_0 = 0.1713245	x_0 = −0.932469514	$\cong f^{(12)}(\xi)$
	c_1 = 0.3607616	x_1 = −0.661209386	
	c_2 = 0.4679139	x_2 = −0.238619186	
	c_3 = 0.4679139	x_3 = 0.238619186	
	c_4 = 0.3607616	x_4 = 0.661209386	
	c_5 = 0.1713245	x_5 = 0.932469514	

Nota. Se puede demostrar que las raíces de los *polinomios de Legendre* son los nodos que se utilizan para resolver nuestro problema.

Hasta ahora hemos impuesto que los límites de integración sean –1 y 1. Pero es posible utilizar un simple cambio de variable para transformar otros límites de integración a esta forma. Si consideramos:

$$t = \frac{2x - a - b}{b - a}$$

Transformamos la variable x del intervalo $[a, b]$ en la variable t del intervalo [-1,1], de forma que:

$$\int_a^b f(x)dx = \int_{-1}^{1} f\left(\frac{(b-a)t + b + a}{2}\right)\frac{(b-a)}{2}dt$$

9.2 Comandos Maxima para Integración

Vamos a comenzar por presentar los comandos que incorpora Maxima para realizar integrales. De esa forma podremos comprobar la bondad de las aproximaciones que vamos a obtener en las siguientes secciones. Este tema ya se trató en la Práctica 1 y ahora lo vamos a repasar y ampliar.

En Maxima es posible calcular de forma simbólica integrales definidas, mediante el comando:

> **integrate(f(x),x,a,b)** que da la integral definida de la función f(x) en el intervalo [a, b]

La integración numérica de una función $f(x)$ en un intervalo $[a, b]$ se realiza con alguno de los siguientes comandos:

> **quad qags(f(x),x,a,b)** para a y b finitos
> **quad qagi(f(x),x,a,b)** para intervalos ilimitados
> **romberg(f(x),x,a,b)** para a y b finitos

Si se introduce en el menú **Análisis-Integrar** (y marcamos integración definida e integración numérica) podemos elegir entre el método **quadpack** o **romberg**. Por ejemplo:

```
(%i1)  kill(all)$
(%i1)  integrate(1/(x^2+1), x, 0, 1);
```

(%o1) $\frac{\pi}{4}$

```
(%i2)  %,numer;
(%o2)  0.7853981633974483
(%i3)  quad_qags(1/(x^2+1), x, 0, 1);
```

(%o3) $[0.7853981633974484, 8.719671245021583\ 10^{-15}, 21, 0]$

```
(%i4)  romberg(1/(x^2+1),x,0,1);
(%o4)  0.785398159599199
```

Se puede mejorar esta aproximación de romberg cambiando la variable **rombergtol**, tras haber cargado el paquete **romberg** mediante **load(romberg).**

```
(%i5)  load(romberg)$
(%i6)  rombergtol:1.0*10^-15$
(%i7)  romberg(1/(x^2+1),x,0,1);
(%o7)  0.7853981633974485
```

Para calcular integrales impropias con intervalo de integración no acotado, sólo se puede usar el comando **quad qagi.**

```
(%i8)  quad_qagi(1/(x^2+1), x, 0, inf);
```

(%o8) $[1.570796326794897, 2.577791520551928\ 10^{-10}, 45, 0]$

Si usamos la paleta basta seleccionar **quadpack** y el propio programa se encarga de seleccionar el comando apropiado.

9.3 Fórmula del Trapecio Simple y Compuesta

Comencemos por interpolar la función en [a, b] por una recta y programar la fórmula del *trapecio simple*:

$$\int_a^b f(x)dx \cong \frac{b-a}{2}[f(a)+f(b)]$$

Podemos construir un **block** como éste:

```
(%i1)    kill(all)$
(%i1)    /* Función trapesim(f_,a,b) */
         /* Regla del Trapecio Simple
             para aproximar una integral definida */
         /* ARGUMENTOS DE ENTRADA: */
         /* f ... .Función integrando */
         /* a,b .. Extremos del intervalo de integración */
         /* ARGUMENTO DE SALIDA: */
         /* intaprox ... Valor aproximado de la integral */;
         trapesim(f_,a,b):=block([numer],numer:true,
             intaprox:(b-a)/2*(f_(a)+f_(b)),
              print("Valor aproximado de la integral =",
                 intaprox))$
```

Por ejemplo, si queremos aproximar la integral de $f(x) = \cos(x^2)$ en el intervalo [0, 1], tenemos:

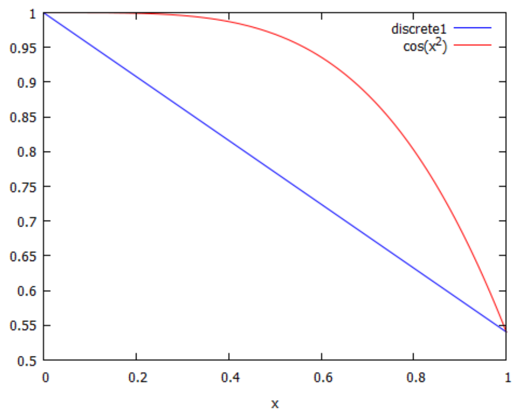
```
(%i2)    f(x):=cos(x^2)$
(%i3)    trapesim(f,0,1)$
Valor aproximado de la integral =  0.7701511529340699
```

Y si comparamos con **quadpack** vemos lo mala que es, lógicamente, la aproximación.

```
(%i4)    quad_qags(cos(x^2), x, 0, 1);
(%o4)    [0.904524237900272,1.004223635248405 10^-14,21,0]
```

La gráfica nos revela claramente que estamos haciendo:

```
(%i5)    datos:[[0,f(0)],[1,f(1)]],numer$
(%i6)    plot2d([[discrete,datos],f(x)],[x,0,1]
              ,[style,lines,lines])$
```



Es obvia la necesidad de mejorar la aproximación y para ello vamos ahora a programar la regla del *trapecio compuesta*, cuya fórmula para *n* subintervalos de igual longitud era:

$$x_0 = a;\ \ x_1 = a + h;\ \ \ x_2 = a + 2h;\ \ \ x_n = a + nh = b$$

$$h = \left(\frac{b-a}{n}\right)$$

$$\int_a^b f(x)dx \cong \frac{b-a}{2n}\left[f(a)+2\sum_{i=1}^{n-1} f(x_i)+f(b)\right]$$

Vamos a programar un nuevo **block**. En este caso, para simplificar, vamos a utilizar la orden **sum** de Maxima para construir el sumatorio de la fórmula. Así evitamos el uso interno de bucles **for**. Esta técnica la usaremos también en el resto de los bloques de la práctica.

```
(%i1)    kill(all)$
(%i1)    /* Función trapecom(f_,a,b,inter) */
         /* Regla del Trapecio Compuesta
             para aproximar una integral definida */
         /* ARGUMENTOS DE ENTRADA: */
         /* f ... .Función integrando */
         /* a,b .. Extremos del intervalo de integración */
         /* inter .Número de intervalos de división */
         /* ARGUMENTO DE SALIDA: */
         /* intaprox ... Valor aproximado de la integral */;
         trapecom(f_,a,b,n):=block([numer],numer:true,
             h:(b-a)/n,
             intaprox:(b-a)/(2*n)*(f_(a)+
                 2*sum(f_(a+h*i),i,1,n-1)+f_(b)),
             print("Valor aproximado de la integral =",
                 intaprox))$
```

Lo aplicamos para aproximar de nuevo la integral anterior, usando por ejemplo, 5 subintervalos.

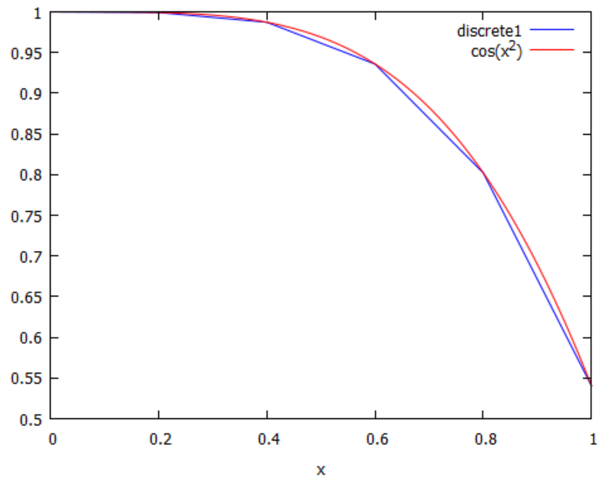
```
(%i2)    f(x):=cos(x^2)$
(%i3)    trapecom(f,0,1,5)$
Valor aproximado de la integral =  0.8989142249065806
```

La mejora es ya evidente (¡y sólo con 5 subintervalos!) y es obvio que cuantos más subintervalos consideremos, más nos aproximamos al valor dado por **quadpack.**

```
(%i4)    quad_qags(cos(x^2), x, 0, 1);
(%o4)    [0.904524237900272,1.004223635248405 10^-14,21,0]
(%i5)    trapecom(f,0,1,50)$
Valor aproximado de la integral =  0.9044681397798161
(%i6)    trapecom(f,0,1,500)$
Valor aproximado de la integral =  0.9045236769196108
```

La siguiente gráfica muestra el caso más simple, con 5 subintervalos.

```
(%i7)    puntos:makelist([i,f(i)],i,0,1,1/(5)),numer$
(%i8)    plot2d([[discrete,puntos],f(x)],[x,0,1])$
```



También podemos obtener una salida muy "elegante" con el siguiente comando:

```
(%i9)    print('integrate(f(x),x,0,1),"~",intaprox)$
```

$$\int_0^1 \cos\left(x^2\right) dx \quad \sim \quad 0.9045236769196108$$

En este **block** el criterio de salida utilizado ha sido el más básico: simplemente el número de subintervalos elegido por nosotros. Sin embargo existen otras formas de programarlo. Podríamos, por ejemplo como en cualquier método iterativo, finalizar el cálculo cuando el *error relativo porcentual* ε_a sea menor que un valor previamente fijado de *tolerancia* ε_s:

$$\varepsilon_a = \left|\frac{x^{(k)} - x^{(k-1)}}{x^{(k)}}\right| 100\% < \varepsilon_s$$

donde $x^{(k)}$ es el valor aproximado para k subintervalos y $x^{(k-1)}$ el valor aproximado para $(k-1)$ subintervalos.

Pero en los métodos de integración numérica, tenemos una ventaja. Y es que la teoría nos proporciona siempre una cota para el *Error absoluto* cometido. En el caso que ahora nos ocupa, sabemos que la expresión del *Error absoluto*, *E*, de la fórmula del trapecio compuesta está acotada por:

$$|E| \leq \frac{(b-a)^3}{12n^2} M\,, \qquad con\ M = \max|f''(x)|, \qquad a < x < b$$

Si la función a integrar es de clase C^2 y si se pueda acotar el valor absoluto de $f''(x)$ en el intervalo dado, puede mejorarse el programa, obteniendo previamente el número n de subdivisiones a realizar para conseguir que el módulo del error absoluto cometido sea menor que un valor dado.

Si calculamos n de forma que:

$$\frac{(b-a)^3}{12n^2}M \le \varepsilon$$

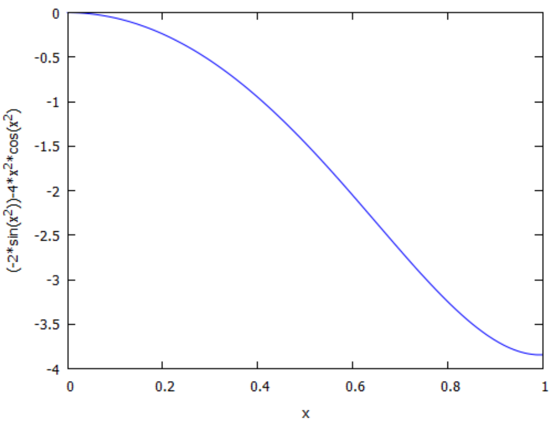
tendremos garantizado que el error sea menor que ε, por lo que n deberá cumplir:

$$\frac{(b-a)^3}{12\varepsilon}M \le n^2 \ \rightarrow n \ge \sqrt{\frac{(b-a)^3}{12\varepsilon}M}$$

Por tanto, empezaremos por calcular M, es decir deberemos hallar el máximo del valor absoluto de $f''(x)$ en el intervalo $[a, b]$.

Por ejemplo, en el caso anterior, si fijamos una cota del Error absoluto de $\varepsilon = 10^{-6}$, tenemos que:

```
(%i10)  define(der2(x),diff(f(x),x,2))$
(%i11)  plot2d(der2(x),[x,0,1])$
```



Obtenemos de la gráfica el máximo en valor absoluto y tenemos que:

```
(%i12)  floor(sqrt((1-0)^3/(12*10^-6)*3.844));
(%o12)  565
```

Es decir, que si dividimos el intervalo de integración en 565 subintervalos, garantizamos la cota del error absoluto pedida, aunque es casi seguro que podría alcanzarse con menos subintervalos.

```
(%i13)  trapecom(f,0,1,565)$
Valor aproximado de la integral = 0.9045237985701283
(%i14)  quad_qags(cos(x^2), x, 0, 1);
(%o14)  [0.904524237900272,1.004223635248405 10^-14,21,0]
(%i15)  abs(0.904524237900272-0.9045237985701283);
(%o15)  4.393301437088937 10^-7
```

9.4 Regla de Simpson Simple y Compuesta

Comencemos ahora por interpolar la función en [*a*, *b*] por el polinomio de grado dos que pasa por los puntos de la función de los extremos y el punto medio del intervalo. Es decir vamos a programar la fórmula del *Simpson 1/3 simple*:

$$\int_a^b f(x) \cong \frac{b-a}{6}\left[f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right]$$

Un posible **block** sería:

```
(%i1)   kill(all)$
(%i1)   /* Función simpsim(f_,a,b) */
        /* Regla de Simpson Simple
           para aproximar una integral definida */
        /* ARGUMENTOS DE ENTRADA: */
        /* f ... .Función integrando */
        /* a,b .. Extremos del intervalo de integración */
        /* ARGUMENTO DE SALIDA: */
        /* intaprox ... Valor aproximado de la integral */;
        simpsim(f_,a,b):=block([numer],numer:true,
            intaprox:(b-a)/6*(f_(a)+4*f_((a+b)/2)+f_(b)),
             print("Valor aproximado de la integral =",
                intaprox))$
```

Para probarla consideremos el siguiente ejemplo. Queremos aproximar en el intervalo [2, 3] la integral de:

$$f(x) = \frac{1}{5-x}$$

```
(%i2)   f(x):=1/(5-x)$
(%i3)   sol1:simpsim(f,2,3)$
Valor aproximado de la integral = 0.40555555555556
```

Vamos a calcular el error absoluto cometido. Para ello comparamos con el comando **integrate**, en aritmética de punto flotante (**numer**).

```
(%i4)   sol2:integrate(f(x), x, 2, 3),numer;
(sol2)  0.4054651081081645
(%i5)   abs(sol1-sol2);
(%o5)   9.044744739106214 10^-5
```

Obtenemos como vemos un gran resultado. El motivo lo podemos ver en la siguiente figura. El polinomio de interpolación de grado 2 y la función son casi iguales en el intervalo.

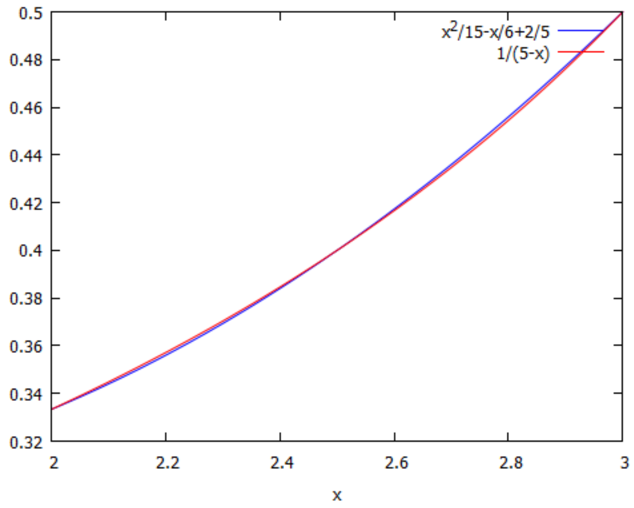
```
(%i6)   load(interpol)$
(%i7)   datos:[[2,f(2)],[5/2,f(5/2)],[3,f(3)]];
```

(datos) $[[2,\frac{1}{3}],[\frac{5}{2},\frac{2}{5}],[3,\frac{1}{2}]]$

```
(%i9)   lagrange(datos)$
        define(s(x),expand(%));
```

(%o9) $s(x):=\frac{x^2}{15}-\frac{x}{6}+\frac{2}{5}$

```
(%i10)  plot2d([s(x),f(x)],[x,2,3])$
```



Para prevenir situaciones no tan favorables vamos ahora a programar la regla de *Simpson compuesta*, cuya fórmula para *n* subintervalos de igual longitud era:

$$\int_a^b f(x) \cong \frac{b-a}{3n}\left[f(a)+4\sum_{i=1,3,5}^{n-1} f(x_i)+2\sum_{i=2,4,6}^{n-2} f(x_i)+f(b)\right]$$

A la hora de programar el **block** vamos a utilizar de nuevo la orden **sum** de Maxima para construir el sumatorio de la fórmula. Recordemos además que el método sólo se puede emplear si el número de subintervalos es par. Además debemos tener cuidado y distinguir la aportación de los nodos pares e impares.

```
(%i1)   kill(all)$
(%i1)   /* Función simpcom(f_,a,b,inter) */
        /* Regla de Simpson Compuesta
           para aproximar una integral definida */
        /* ARGUMENTOS DE ENTRADA: */
        /* f ... .Función integrando */
        /* a,b .. Extremos del intervalo de integración */
        /* inter .Número de intervalos de división */
        /* ARGUMENTO DE SALIDA: */
        /* intaprox ... Valor aproximado de la integral */;
        simpcom(f_,a,b,n):=block([numer],numer:true,
           h:(b-a)/n,
           intaprox:(b-a)/(3*n)*(f_(a)
               +4*sum(f_(a+(2*i-1)*h),i,1,n/2)
               +2*sum(f_(a+(2*i)*h),i,1,n/2-1)+f_(b)),
           print("Valor aproximado de la integral =",
               intaprox))$
```

Recordemos que:

$$x_0 = a;\ \ x_1 = a + h;\ \ \ x_2 = a + 2h;\ \ \ x_n = a + nh = b;\ \ con\ \ \ h = \left(\frac{b-a}{n}\right)$$

De modo que las leyes que siguen los nodos impares y pares son:

$$x_{2i-1} = a + (2i-1)h;\ \ i = 1, \dots, \frac{n}{2}$$

$$x_{2i} = a + (2i)h;\ \ i = 1, \dots, \frac{n}{2} - 1$$

Si aplicamos ahora para 10 subintervalos tenemos:

```
(%i2)    f(x):=1/(5-x)$
(%i3)    sol1:simpcom(f,2,3,10)$
Valor aproximado de la integral = 0.4054652741709214
(%i4)    sol2:integrate(f(x), x, 2, 3),numer;
(sol2)   0.4054651081081645
(%i5)    abs(sol1-sol2);
(%o5)    1.660627568789153 10^-7
```

Mejorando, lógicamente, la aproximación frente al área exacta calculada mediante la regla de Barrow.

Podemos hacer ahora lo mismo que para la regla del trapecio compuesta. Vamos a calcular primero en cuántos subintervalos iguales hay que dividir el intervalo de integración para conseguir una aproximación con error absoluto menor que un épsilon dado. En este caso:

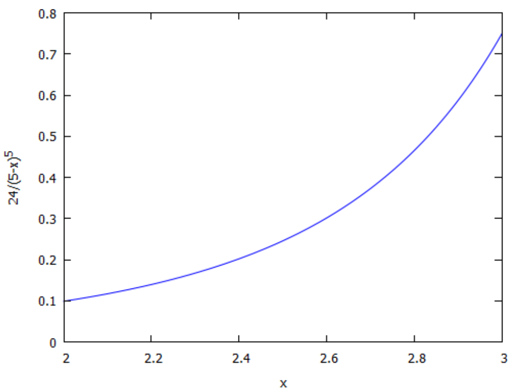
$$|E| \leq \frac{(b-a)^5}{180\, n^4} M\,, \qquad con\ M = \max |f''''(x)|, \qquad a < x < b$$

De donde:

$$\frac{(b-a)^5}{180\varepsilon} M \leq n^4\ \rightarrow n \geq \sqrt[4]{\frac{(b-a)^5}{180\varepsilon} M}$$

En el caso anterior veamos cuántos subintervalos son necesarios para asegurar un error menor que 10^{-12}.

```
(%i6)    define(der4(x),diff(f(x),x,4))$
(%i7)    plot2d(der4(x),[x,2,3])$
```



```
(%i8)    floor(((3-2)^5/(180*10^-12)*0.75)^(1/4));
(%o8)    254
(%i9)    sol1:simpcom(f,2,3,254)$
Valor aproximado de la integral = 0.4054651081085662
(%i10)   sol2:integrate(f(x), x, 2, 3),numer;
(sol2)   0.4054651081081645
(%i11)   abs(sol1-sol2);
(%o11)   4.016786903093816 10^-13
```

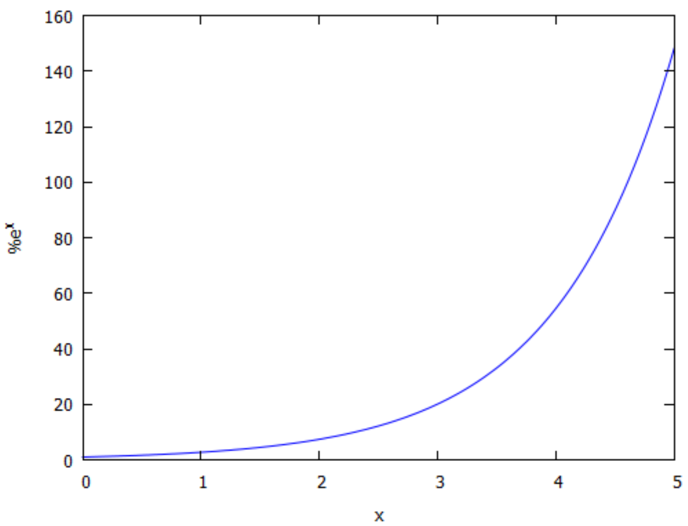
Para finalizar este apartado comparemos en un nuevo ejemplo la precisión de los 4 métodos que hemos implementado. Veamos cómo se comportan para calcular:

$$\int_0^5 e^x dx$$

```
(%i1)    kill(all)$
(%i1)    f(x):=exp(x)$
(%i9)    trapesim(f,0,5)$
         trapecom(f,0,5,100)$
         simpsim(f,0,5)$
         simpcom(f,0,5,100)$
Valor aproximado de la integral = 373.5328977564415
  Valor aproximado de la integral = 147.4438688978377
  Valor aproximado de la integral = 165.1192791211588
  Valor aproximado de la integral = 147.4131642195661
```

Y comparamos con el comando **integrate**.

```
(%i11)   ratprint:false$
         integrate(f(x),x,0,5),numer;
(%o11)   147.4131591025766
(%i12)   plot2d([f(x)],[x,0,5])$
```



9.5 Fórmulas de Cuadratura Gaussiana

Como vimos otra forma de disminuir el error de las fórmulas de cuadratura, es usar la de las fórmulas de Cuadratura Gaussianas, donde los puntos no se escogen igualmente separados, sino que se escogen de forma que el grado de precisión sea máximo. La fórmula general es:

$$\int_{-1}^{1} f(x)dx \approx c_1 f(x_1) + c_2 f(x_2) + \cdots + c_n f(x_n)$$

donde n = número de puntos. Los valores de las c y las x para fórmulas de hasta seis puntos se resumen en la tabla que vimos en teoría.

Para probarla consideremos de nuevo el ejemplo que vimos en la regla de Simpson. Vamos a aproximar en el intervalo [-1, 1] la integral de:

$$f(x) = \frac{1}{1+x^2}$$

Consideraremos tres puntos, como en la regla de Simpson Simple, para ver cuál de las dos parábolas aproxima mejor.

```
(%i1)    kill(all)$
(%i1)    f(x):=1/(1+x^2)$
(%i2)    solcuag:0.5555556*f(-0.774596669)
         +0.8888889*f(0)+0.5555556*f(0.774596669);
(solcuag) 1.583333400162372
(%i5)    solsim:simpsim(f,-1,1)$
Valor aproximado de la integral = 1.666666666666667
(%i6)    solex:integrate(f(x), x, -1, 1),numer;
(solex)  1.570796326794897
```

Como vemos a continuación obtenemos mucha mejor aproximación con la fórmula de cuadratura gaussiana que con la regla de Simpson Simple, a pesar de usar en ambos casos el mismo número de puntos.

```
(%i7)    abs(solcuag-solex);
(%o7)    0.012537073367747523
(%i8)    abs(solsim-solex);
(%o8)    0.095870339871769996
```

La clave está en la ubicación de los puntos.

9.6 Ejercicios Propuestos

Ejercicio 1. Calcular utilizando la regla del trapecio simple:

$$\int_0^2 (3x+4)dx; \ \int_0^\pi sen\frac{x}{4}\ dx$$

Calcular los errores absolutos cometidos comparando con el comando **integrate**, en aritmética de punto flotante (**numer**). Interpretar gráficamente los resultados.

Ejercicio 2. Calcular utilizando la regla de Simpson simple:

$$\int_0^2 (x^3+x^2+2x-1)dx; \ \int_0^\pi sen\frac{x}{4}\ dx$$

Calcular los errores absolutos cometidos comparando con el comando **integrate**, en aritmética de punto flotante (**numer**). Interpretar gráficamente los resultados.

Ejercicio 3. Dada la integral:

$$\int_0^3 x^2cos2x\ dx$$

a) Calcularla por la regla del trapecio compuesta, dividiendo el intervalo de integración en 20 subintervalos, y hallar el error absoluto cometido.
b) Obtener el número de subintervalos que deberán utilizarse para garantizar que el error absoluto sea menor que una milésima y comprobar el resultado obtenido.

Ejercicio 4. Dada la integral:

$$\int_0^\pi x^2sen5x\ dx$$

a) Calcularla por la regla de Simpson compuesta, dividiendo el intervalo de integración en 10 subintervalos, y hallar el error absoluto cometido.
b) Obtener el número de subintervalos que deberán utilizarse para garantizar que el error absoluto sea menor que una millonésima y comprobar el resultado obtenido.

Ejercicio 5. Calcular de forma aproximada la integral:

$$\int_{-1}^1 xe^x\ dx$$

utilizando cuadratura gaussiana con 4 nodos, y hallar el error absoluto cometido.

Ejercicio 6. Repetir el problema anterior, pero usando sólo 3 nodos y comparar la solución con la obtenida mediante la regla de Simpson simple.

Ejercicio 7. Utiliza los métodos de integración del Trapecio compuesto y de Simpson compuesto para n = 10 y 20, para calcular:

$$\int_5^{10} sen(x^2)\ dx$$

Compara los resultados. La función es difícil de integrar numéricamente porque oscila mucho en este intervalo, dibújala para verlo.

Calcula el número de subintervalos que deberán utilizarse para garantizar que el error absoluto sea menor que una millonésima en ambos casos.