Computing the atomic rearrangement pathways for pure electron nuclides capture by a five-shell model

Gonzalo Galiano

Departamento de Matemáticas, Universidad de Oviedo, 33007 Oviedo, Spain

Abstract

We present a program to compute the different pathways of atomic rearrangement which follow the capture of one electron by the nucleus, as well as their probabilities. We consider a $KL_1L_2L_3M$ five-shell model although the structure of the program is essentially the same for other five-shell models like the KLMNO model.

PROGRAM SUMMARY

Title of the program: Paths5
Catalogue number:
Program obtainable from:
Licensing provisions: none
Computer: any IBM compatible with 80386 32–bit processor
Operating system: MS-DOS 3.30 and higher
Programming language used: Fortran 77
Memory required to execute with typical data: 45 Kwords
No. of bits in a word: 32
No. of lines in distributed program, including test data, etc.: 794
Keywords: Electron capture, atomic rearrangement, five shell model.
Nature of the physical problem: The computation of the liquid scintillation counting efficiency for electron capture nuclides requires the determination of the atomic pathways which follows the capture of one electron by the nucleus, see [3]. A KLM three-shell model involves only 22 pathways which can be easily computed by hand. However, the complexity of determining the rearrangement pathways highly increases when more shells are considered in the model. We propose an algorithm to compute these pathways for a five-shell model which can be extended to more complex situations.
Method of solution: We consider a tree of five levels where three anihilation–creation of holes operators are defined. We run over all the different possibil-

ities of action of these operators, obtaining in this way all the rearrangement pathways.

Restriction of the complexity of the problem: We average M subshells. We do not consider the atomic rearrangement for M and outer shells.

Typical running time: 0.7 sec of cpu time on a Pentium 122Mh, 16Mb, IBM–compatible computer.

LONG WRITE-UP

1 Introduction

It has been shown in [3] and [1] that the counting efficiency for electron capture and internal conversion by a KLM three-shell model is in excellent agreement with experiment for low atomic number nuclides, e.g., ^{55}Fe, ^{65}Zn or ^{51}Cr. Due to the lack of accuracy observed for nuclides with higher Z-values, the model was extended in [2] to the consideration of a $KLMN$ four-shell model. The main difficulty in this extension was the determination of the different pathways that the model generates. Thus, meanwhile the KLM model generates 22 pathways, in the $KLMN$ model there arise 262 different pathways.

We present in this paper a program to compute the pathways generated by a $KL_1L_2L_3M$ model, i.e., a model which takes into consideration the three sub-shells conforming the L-shell. The algorithm is also valid for other five-shell models like the $KLMNO$ model. However, let us mention that $KL_1L_2L_3M$ model generates 378 pathways (counting multiplicities due to symmetries) meanwhile $KLMNO$ model generates 607560 pathways.

2 Probabilities and effective energies for the pathways

In a five-shell model, the electron capture process creates a hole in one of the inner shells or sub-shells. This hole leads to the emission of either an X-ray, which may be detected or not detected, either an Auger (A) electron or either a Coster-Kronig (CK) electron, all of them generating new vacancies in higher levels. The new holes are filled again by a cascade of electrons which shifts holes to higher levels. We can achieve all atomic rearrangement pathways by only combining the three annihilation-creation operators X-ray, A and CK. The X-ray operator shifts a hole in Y-shell or sub-shell to a higher shell. The Auger operator $A(Y)$ annihilates a hole in Y-shell but creates two new holes in outer shells or sub-shells. Finally, the Coster-Kronig operator $CK(Y)$ fills a hole in Y-sub-shell creating two new holes in outer shells or sub-shells. Notice that A and CK operators are essentially the same but the domain in which they act (shells or sub-shells, respectively). This distinction is, however, responsible for the large amount of pathways of the $KLMNO$ model (where only X-ray and A operators must be considered) when compared to the $KL_1L_2L_3M$ model.

We obtain all possible pathways in five steps corresponding to five levels in a graphic tree, see Figure 1. In the first step we start with a set of holes ^1B

generated by the capture of one electron in K, L_1, L_2 or L_3 shells:

$$^1B = \{K, L_1, L_2, L_3\}.$$

In the next steps, we apply X-ray, A and CK operators to all possible holes. The second step conforms the set ^2B of 30 events, which we list below

$$^2B = \{A(K) = L_1L_1, L_1L_2, L_1L_3, L_1M, L_2L_2, L_2L_3, L_2M, L_3L_3, L_3M, MM;$$

$$C(L_1) = L_2M, L_3M, MM; \quad C(L_2) = L_3M, MM, \quad C(L_3) = MM;$$

$$X(K) = L_{1d}, L_{1u}, L_{2d}, L_{2u}, L_{3d}, L_{3u}, M_d, M_u;$$

$$X(L_1) = X(L_2) = X(L_3) = M_d, M_u\}$$

The subscripts d and u denote detection and non-detection for X-rays, respectively. The following steps shift out holes to M shell and conform a final set ^5B of 378 pathways. The probability for each event in the final set ^5B is a Markov chain

$$P(B_i) = P(B_{i_1})\, P(B_{i_2}/B_{i_1})\, P(B_{i_3}/B_{i_2})\, P(B_{i_4}/B_{i_3})\, P(B_{i_5}/B_{i_4}), \quad (1)$$

where B_i refers to all final events while B_{i_j} represents all events into the five sets ^jB, $j = 1, ..., 5$.

Let us point out that in $KLMNO$ model we also start with 4 possible events. However, in the second step we already have 40 events and in successive steps the number of events increases very quickly until we reach the 607560 possible events of the last step.

Due to the presence of symmetries, many of the computed events have the same probability. We can reduce the total number of events to 264 pathways by assigning a weight to each expression. This output plus the necessary data to fully determine the formulas (1) (like the detection or non-detection of X-ray probabilities, the fluorescence yield or the Auger emission probability) allows to compute the counting efficiency for pure electron capture nuclides, the main application of this program.

3 Program structure

The program PATHS5 contains a main program and eight subroutines. The output of the main program is a collection of 31 character chains which identify the possible rearrangement pathways. These chains are generated by running

over all the possibilities of composition of the three operators X-ray, A, and CK which, in programming terminology, accounts for running 16 nested loops (one for each possible branch in the tree plus the first capture). We shall illustrate the procedure of obtaining a pathway by an example, see Figure 1: in the first loop an electron in the K shell has been captured by the nuclide (first level), i.e., the event K in ^1B occurred. Next, in the second level (two loops), an Auger electron fills the hole in the K shell and creates one hole in each of the L_1 and L_3 sub-shells, i.e., the event

$$A(K) = L_1 L_3$$

in ^2B occurred. Since the growing of a branch only terminates when a hole is produced in the M shell, the process continue in both branches. So, in the third level (four loops), a CK electron fills the hole in L_1 shell and creates holes in L_2 and M shells, meanwhile the hole in L_3 is filled by a CK electron which creates two holes in the M shell, so we obtain the event

$$CK(L_1) = L_2 M, \quad CK(L_3) = MM$$

from ^3B. This event implies that three branches stop growing and only one continue. Therefore, in the fourth level (eight loops although in this example only two actually run) the operators only act on the hole in L_2 shell. In our example, this hole is filled by a CK electron which creates holes in L_3 and M shells, so we obtain the event $CK(L_2) = L_3 M$ in ^4B. Finally, in the fifth level all the branches must stop. In this example, the hole in L_3 shell is filled by a CK electron which creates two holes in the M shell, i.e., $CK(L_3) = MM$ in ^5B. We then write this pathway by the 31 characters chain

$$KL_1L_3L_2MMML_3M \_\_\_\_\_MM _____, \tag{2}$$

which identifies the pathway, i.e., the vacancies created in the process of rearrangement and the operators which filled them. Notice that (2) keeps all the information needed to compute the probability of the pathway, given by

$$\text{prob} = P(K) * P(A(K) = L_1L_3) * P(CK(L_1) = L_2M) * \tag{3}$$
$$* P(CK(L_3) = MM) * P(CK(L_2) = L_3M) * P(CK(L_3) = MM).$$

In the program we enshorten this expression by assigning a single character to each one of the factors in (3), writing for instance A instead of $P(K)$, G instead of $P(A(K) = L_1L_3)$, and so on. The actual writing for the pathway (2) is

$$AGOTRT. \tag{4}$$

This form of writing the pathways has two advantages: first, it uses less memory (we pass from writing a chain of at most 31 characters in (2) to a chain of at most 8 characters in (4)). Second, we may classify these chains by their length, making easier the identification of repeated pathways. The actual output of the main program is a reordering of the chain in (4) consisting on rewriting the chain in alphabetic order, i.e.,

$$AGORTT.$$

We generate in this way 7 temporal data files containing chains of (ordered) characters of length 2, 3,...,8. The next step consists on identifying the repeated chains (pathways with the same probability). Now we only have to compare chains with the same length by the ASCII ordering operator (similar to the alphabetic ordering). Once this is done we are ready to write the output of the program: a collection of FORTRAN formulas given in terms of capture probabilities, Auger and Coster-Kronig probabilities, fluorescence yields and X-ray detection or non-detection probabilities. At this stage, the effective energy expressions are just a direct character translation from these probabilities.
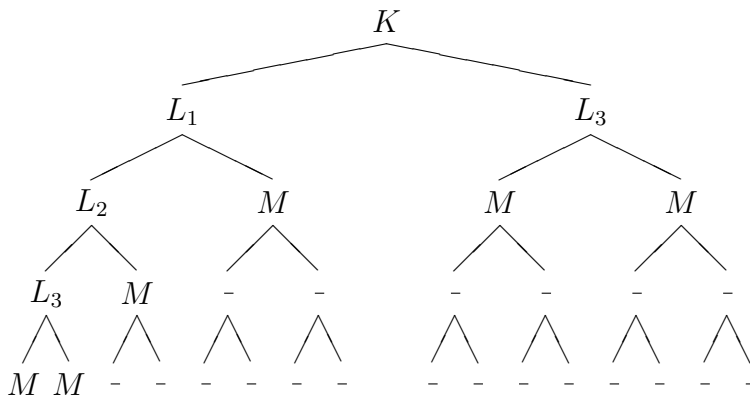

Figure 1

4   Input-output data files

Since the output of PATHS5 is a FORTRAN program, we shall make a distinction between the input data associated to both of them.

Part of the input data for PATHS5 is integrated into the program. It consists of a matrix containing the holes that the three operators X-ray, A and CK may annihilate and create. The output data from PATHS5 is the FORTRAN program COMP. To write this program, PATHS5 uses an input data text file called AUX1.DAT. Program COMP contains the probability formulas (the aim

of the program) and data to test them (which amounts to test PATHS5). The test consists of the addition of all the formulas for the given data. Obviously, this must sum 1.

References

[1] J.M. Los Arcos, A. Grau Malonda and A. Fernández, *Computer Physics Comm.* **44** (1987) 209.

[2] G. Galiano, A. Grau Malonda and A. Grau Carles, CAPMULT, the counting efficiency for electron capture by a KLMN four-shell model, *Computer Physics Comm.* **87** (1995) 432–437.

[3] A. Grau Malonda, *Appl. Rad. Isotop.* **33** (1982) 371.

The output file of program PATHS5 is a FORTRAN program which tests the probability formulas

```
IMPLICIT REAL(a-z)           PR(117)=2.*A*E*P*Q*F1
REAL PR(264)                 PR(118)=2.*A*E*P*T*A1
INTEGER I10                  PR(119)=2.*A*E*P*A1*E1
PK=0.2                       PR(120)=2.*A*E*P*A1*F1
PL=0.3                       PR(121)=2.*A*E*P*T*B1
PM=0.1                       PR(122)=2.*A*E*P*B1*E1
PN=0.4                       PR(123)=2.*A*E*P*B1*F1
  ⋮                            ⋮
A=PK                         PR(232)=1.*A*I*R*R*T*T
B=PL                         PR(233)=2.*A*I*R*R*T*E1
C=PM                         PR(234)=2.*A*I*R*R*T*F1
D=PN                         PR(235)=1.*A*I*R*R*E1*E1
E=AK*KLL                     PR(236)=2.*A*I*R*R*E1*F1
F=AK*KLM                     PR(237)=1.*A*I*R*R*F1*F1
  ⋮                            ⋮
PR( 1)=1.*A*N                PR(263)=2.*A*E*O*O*R*R*E1*F1
PR( 2)=1.*A*W                PR(264)=1.*A*E*O*O*R*R*F1*F1
PR( 3)=1.*A*Z                X=0.
PR( 4)=1.*B*Q                do I10=1,264
PR( 5)=1.*B*A1               X=X+PR(I10)
PR( 6)=1.*B*B1               enddo
PR( 7)=1.*C*S                write(*,*)X
  ⋮                          end
```