Decay chain differential equations: Solution through matrix algebra
Decay chain differential equations: Solution through matrix algebra

M. Amaku\textsuperscript{a,*}, P.R. Pascholati\textsuperscript{b}, V.R. Vanin\textsuperscript{b}

\textsuperscript{a} Faculdade de Medicina Veterinária e Zootecnia, Universidade de São Paulo, 05508-270 São Paulo, SP, Brazil
\textsuperscript{b} Instituto de Física, Universidade de São Paulo, Caixa Postal 66318, 05314-970 São Paulo, SP, Brazil

A R T I C L E   I N F O

Article history:
Received 21 May 2009
Received in revised form 8 July 2009
Accepted 24 August 2009
Available online 26 August 2009

PACS:
23
02.10.Ud

Keywords:
Radioactivity
Decay chain equations
Radioactive decay and buildup
Matricial method
Recurrence relation
Triangular matrix inverse

A B S T R A C T

A matricial method to solve the decay chain differential equations system is presented. The quantity of each nuclide in the chain at a time \( t \) may be evaluated by analytical expressions obtained in a simple way using recurrence relations. This method may be applied to problems of radioactive buildup and decay and can be easily implemented computationally.

© 2009 Elsevier B.V. All rights reserved.

Let us consider a decay chain with \( n \) nuclides such that the \( i \)th nuclide decays to the \((i+1)\)th nuclide of the chain. If \( N_i(t) \) is the quantity of \( i \)th nuclide at time \( t \) and \( \lambda_i \) is its decay constant, we may write

\[
\begin{align*}
\frac{dN_1(t)}{dt} &= -\lambda_1 N_1(t), \\
\frac{dN_2(t)}{dt} &= \lambda_1 N_1(t) - \lambda_2 N_2(t), \\
\vdots \\
\frac{dN_n(t)}{dt} &= \lambda_{n-1} N_{n-1}(t) - \lambda_n N_n(t).
\end{align*}
\]

(1)

This system of differential equations may be solved in an involved way by successive substitution, as did Rutherford\textsuperscript{[1]}, who studied the problem in which there are two products besides the primary nuclide. Extending these results to any number of products, Bateman\textsuperscript{[2]} in 1910 derived the so-called “Bateman’s equations”, but using a Laplace transform method, which may be found in a compact form in Mann, Rytz and Spernol\textsuperscript{[3]}. This method was reconsidered by Pressyanov\textsuperscript{[4]} in 2002.

To introduce an alternative algebraic solution, we define \( \mathbf{N}(t) \) as

\[
\mathbf{N}(t) = \begin{bmatrix}
N_1(t) \\
N_2(t) \\
\vdots \\
N_n(t)
\end{bmatrix}
\]

(2)

and the triangular bidiagonal matrix

\[
\Lambda = \begin{bmatrix}
-\lambda_1 & 0 & 0 & \cdots & 0 \\
\lambda_1 & -\lambda_2 & 0 & \cdots & 0 \\
0 & \lambda_2 & -\lambda_3 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & -\lambda_n
\end{bmatrix}
\]

(3)

With these definitions, the system of equations (1) may be written as

\[
\frac{d\mathbf{N}}{dt} = \Lambda \mathbf{N}.
\]

(4)

If we consider nuclei immersed in a neutron flux \( \phi \) (neutrons cm\(^{-2}\) sec\(^{-1}\)) and that the \( i \)th nuclide has a capture cross-section \( \sigma_i \), then, besides the spontaneous radioactive decay, we also have a neutron reaction at a rate \( \phi \sigma_i N_i \). The corresponding differential equations system is similar to (1), with \( k_i = \lambda_i + \phi \sigma_i \), i.e.,
\[
d\frac{dN_1(t)}{dt} = -k_1 N_1(t),
\]
\[
d\frac{dN_2(t)}{dt} = \lambda_1 N_1(t) - k_2 N_2(t),
\]
\[
\vdots
\]
\[
d\frac{dN_n(t)}{dt} = \lambda_{n-1} N_{n-1}(t) - k_n N_n(t),
\]
which may also be written as (4). This approach applies better to nuclear reactors, where the flux is more stable; in particle accelerators, the beam current often changes in the long run, but the ensuing inhomogeneous equation can be solved straightforwardly, as will be shown at the end of this letter.

To the best of our knowledge, the algebraic solution of Eq. (4), described by Moral and Pacheco [5], was introduced in 1969 by Onega [6]. The algebraic approach was used by Semkow [7] to derive a solution with branching. Yuan and Kernan [8] extended the algebraic solution for solving generic exit-only radioactive decay problems, using a more complex mathematical framework to derive an analytic solution based on a recurrence relation. In this note, we show a didactic and straightforward approach to derive a recurrence relation, which enabled us to use general properties of linear equation systems, isolating the solution of the system from the initial conditions.

For a more general decay, with branching, the matrix \( A \) would be written as

\[
A = \begin{bmatrix}
\lambda_{11} & \lambda_{12} & \cdots & \lambda_{1n} \\
\lambda_{21} & \lambda_{22} & \cdots & \lambda_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\lambda_{n1} & \lambda_{n2} & \cdots & \lambda_{nn}
\end{bmatrix}.
\]

(6)

All the equations below use this general form because it is as easy to use as the more particular form given by (3).

Eq. (4) is the representation of a coupled equations system. To uncouple them, it is necessary to apply a similarity transformation on matrix \( A \). Proceeding this way, we obtain a diagonal matrix, whose elements are the eigenvalues of \( A \). These eigenvalues are given by the roots of a characteristic polynomial, defined as

\[
p(\beta) = \det(A - \beta I),
\]

(7)

where \( \beta \) stands for the eigenvalues. It is easy to show that \( p(\beta) \) has \( n \) roots given by \( \rho_m = A_{mm} \quad (m = 1, 2, \ldots, n) \).

Each eigenvalue has an eigenvector associated. To obtain these eigenvectors, we have to solve the following equation

\[
\begin{bmatrix}
\lambda_{11} - \lambda_{jj} \\
\lambda_{21} - \lambda_{jj} \\
\vdots \\
\lambda_{n1} - \lambda_{jj}
\end{bmatrix} \begin{bmatrix}
c_{1j} \\
c_{2j} \\
\vdots \\
c_{nj}
\end{bmatrix} = \begin{bmatrix} 0 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]

which gives \( c_{ij} = 0 \) for \( i < j \), and \( c_{jj} \neq 0 \) for \( i = j \). For \( i = j + 1 \), we have

\[
A_{j+1,j}c_{jj} + (A_{j+1,j+1} - A_{jj})c_{j+1,j} = 0,
\]

then

\[
c_{j+1,j} = \frac{A_{j+1,j}c_{jj}}{A_{jj} - A_{j+1,j+1}},
\]

and, for \( i = j + 2, \ldots, n \),

\[
A_{j+2,j}c_{jj} + A_{j+2,j+1}c_{j+1,j} + (A_{j+2,j+2} - A_{jj})c_{j+2,j} = 0
\]

which gives

\[
c_{j+2,j} = \frac{A_{j+2,j}c_{jj} + A_{j+2,j+1}c_{j+1,j}}{A_{jj} - A_{j+2,j+2}}.
\]

Generalizing, for \( i > j \), we may write the recurrence expression

\[
c_{ij} = \frac{\sum_{k=j}^{i-1} A_{ik} c_{kj}}{A_{jj} - A_{ii}}, \quad i = 2, \ldots, n, \quad j = 1, \ldots, i - 1.
\]

(10)

This recurrence relation is similar to Eq. (28) from Yuan and Kernan [8], who however included the initial conditions into the equation. In order to obtain a more straightforward way to solve the system, we isolate the analytical solution from the initial condition, departing from [8] and taking \( c_{ii} = 1 \), since it is arbitrary.

If we define a matrix \( C \) as

\[
C = [c_1 c_2 \cdots c_n],
\]

(11)

the diagonalization of matrix \( A \) is carried out through a similarity transformation, given by

\[
A_d = C^{-1}AC,
\]

(12)

where \( A_d \) stands for a diagonal matrix.

The elements of matrix \( C^{-1} = [b_{ij}] \) may be computed using the following recurrence expression (see Appendix A for details)

\[
b_{ij} = -\sum_{k=j}^{i-1} c_{ik} b_{kj}, \quad i > j,
\]

(13)

and \( b_{ii} = 1 \).

Expressed in terms of Eq. (12), Eq. (4) results in

\[
\frac{dN}{dt} = CA_d C^{-1} N.
\]

(14)

Based on the linearity of the equation system, Onega [6] derived a solution that accounts for the given initial conditions:

\[
N(t) = C [e^{\lambda t}] C^{-1} N(0),
\]

(15)

with

\[
[e^{\lambda t}] = \begin{bmatrix}
e^{-\lambda_1 t} \\
e^{-\lambda_2 t} \\
\vdots \\
e^{-\lambda_n t}
\end{bmatrix}.
\]

(16)

Onega [6], however, has not derived a recurrence relation for the matrix \( C \).

The recurrence expressions for the elements of matrix \( C \) (Eq. (10)) and its inverse \( C^{-1} \) (Eq. (13)) may be computed using the same loop, reducing the number of lines needed to implement a computational code, and also the time of execution in comparison with a numerical procedure of matrix inversion. Besides the loop, the algorithm may be implemented in only one line of instructions for Eq. (15), using a software such as Matlab, for instance.
An example of the improvement represented by the analytical solution shown here can be found in the $^{144}\text{Sm}[^{16}\text{O},xn]Z$ cross-section determination by off-line X-ray measurements following sample bombardment by Pascholati et al. [9]. As mentioned before when discussing Eq. (5), the differential equation that applies to the nuclide numbers during activation in a particle flux that varies in time $t$, $F(t)$, is best derived from Eq. (4) with the addition of an inhomogeneous term,

$$\frac{dN}{dt} = AN + F(t)R,$$

where $R_i$ is the formation ratio of nuclide $i$, proportional to the number of parent nuclides in the target, assumed constant, and the reaction cross-section. The same $C$ matrix used to solve the homogeneous equation (4) can be used to solve Eq. (17) by the integrand factor method, with the result

$$\text{N}(t) = C \int_{0}^{t} \left[ e^{\Lambda(t-t')} \right] F(t') dt' C^{-1} R.$$

In practice, $F(t)$ is replaced by average values measured in time intervals short compared to the nuclides half-lives. In the experiment described in Ref. [9], the particle flux was integrated in 100 consecutive time bins, spanning all the irradiation time. If it were not for the isolation of the initial conditions which allowed factorization of the $C$ matrix, the effect of the flux dependence on time would have required the recalculation of the solution for each time bin, as must be done by the method used in [8]. Moreover, the same $C$ matrix used to solve the equation for the formation phase, Eq. (17), applies to the subsequent decay phase, governed by Eq. (4). Hence, the solution described here made it possible to build a simpler program than would be required by the other methods. All the formulas for the complete activation and decay process, accounting for particle current variation, can be seen in Ref. [9].

In conclusion, the solution of decay chain differential equation systems shown here is based on the recurrence relation (10), similar to an expression obtained by Yuan and Kernan [8] using a more complex mathematical framework. To solve Eq. (4), with a general $\Lambda$ matrix like (6), it was not necessary to break up the system of branched decay chains into independent linear chains, as in the method presented by Mirzadeh and Walsh [10]. Compared either with the method of successive substitutions or the Laplace transform method, the matricial approach provides a more straightforward solution (easily implementable computationally) for the system of differential equations of a decay chain.

**Acknowledgements**

PRP and MA acknowledge financial support from CNPq.

**Appendix A. Recurrence relation for the inverse of a triangular matrix**

Let $C^{-1} = [b_{ij}]$ be the inverse of matrix $C = [c_{ij}]$. $C$ is a unit lower triangular matrix. Therefore, $C^{-1}$ is also a unit lower triangular matrix [11,12], and we have that

$$c_{ij} = 0, \quad b_{ij} = 0, \quad i < j.$$

The element in the $i$th row and the $j$th column of $CC^{-1}$ (the identity matrix) is

$$\sum_{k=1}^{n} c_{ik} b_{kj} = 1, \quad i = 1, \ldots, n. \quad (20)$$

For a given $i$, we may write the previous equation as

$$\sum_{k=1}^{i-1} c_{ik} b_{kj} + c_{ii} b_{ij} + \sum_{k=i+1}^{n} c_{ik} b_{kj} = 1, \quad (21)$$

and, using Eq. (19), that is, $b_{ki} = 0$ for $k < i$, and $c_{ik} = 0$ for $i < k$, we find that

$$c_{ii} b_{ij} = 1. \quad (22)$$

Hence, as $c_{ii} = 1$, we have that

$$b_{ij} = 1, \quad i = 1, \ldots, n. \quad (23)$$

The product of the $i$th row of matrix $C$ by the $j$th column of its inverse, for $i > j$, is given by

$$\sum_{k=1}^{n} c_{ik} b_{kj} = 0, \quad i = 2, \ldots, n, \quad j = 1, \ldots, i - 1, \quad (24)$$

which is equivalent to

$$\sum_{k=1}^{i-1} c_{ik} b_{kj} + c_{ii} b_{ij} + \sum_{k=i+1}^{n} c_{ik} b_{kj} = 0. \quad (25)$$

As $c_{ik} = 0$, for $i < k$, we obtain

$$b_{ij} = - \sum_{k=1}^{i-1} c_{ik} b_{kj}. \quad (26)$$

Taking into account that $b_{kj} = 0$ for $k < j$, we find

$$b_{ij} = - \sum_{k=1}^{j-1} c_{ik} b_{kj}. \quad (27)$$

Therefore, the elements of matrix $C^{-1}$ may be computed using Eqs. (19), (23) and (27).

**References**