Health Physics Vol. 56, No. 6 (June), pp. 857–868, 1989
Printed in the U.S.A.

0017-9078/89 $3.00 + .00

● *Paper*

A MICROCOMPUTER ALGORITHM FOR SOLVING FIRST-ORDER COMPARTMENTAL MODELS INVOLVING RECYCLING

A. Birchall
National Radiological Protection Board, Chilton, Didcot, Oxon OX11 0RQ, England

and

A. C. James
Battelle Pacific Northwest Laboratories, Richland, WA 99352

(*Received* 20 *June* 1988; *accepted* 5 *December* 1988)

***Abstract*—A general algorithm for solving first-order compartmental models including recycling systems has been developed and its implementation on a microcomputer is described. Matrix algebra is used to obtain for any compartmental model an analytical solution, which is expressed as the exponential of a matrix of rate constants. A special technique is used in the algorithm to enable this exponential to be evaluated with a rapidly converging series. Truncation errors incurred in this process are estimated automatically. Thus, in an extreme case, where these errors may be significant, the appropriate action can be taken. Given a particular model, the user enters the model parameters into a rate matrix according to a simple rule. The algorithm then uses this matrix to solve the model, and thus no specialized mathematical knowledge is needed. The algorithm is given in a short BASIC program (60 lines) listed in an appendix. No additional software is required. By running this program on a standard microcomputer, the user can solve models of any complexity: those up to 15 compartments in seconds and those up to 30 compartments within a minute. The algorithm is thus ideally suited to solve kinetic models describing the transport of radionuclides in the environment or the translocation of elements in biological systems such as the metabolic models recommended by the International Commission on Radiological Protection (ICRP). Given the initial amount of material in each compartment at time $t = 0$, together with its radioactive decay constant, the algorithm gives both the amount in each compartment at any future time t and the number of disintegrations that will have occurred in each compartment up to time t. The computer program, shown in an appendix, could easily be used to calculate disintegrations over any time interval of interest, or to predict the quantities or fractions of an intake expected to be present in any *in vivo* or excretion compartments of interest. Thus, the algorithm can be useful in both the design and conduct of bioassay and internal dose assessment procedures.**

INTRODUCTION

AN EARLIER paper (Birchall 1986) presented the advantages of a simple method for solving first-order compartmental models to enable health physicists to deal with the ubiquitous kinetic models employed in radiological protection and other fields. The information about the model was contained in a rate matrix, comprised of the initial amounts and translocation rate constants for each compartment. An algorithm was presented which operated directly on the rate matrix to solve the system. The disadvantage of this algorithm is that it cannot be directly applied to the recycling compartmental models often required to represent environmental and metabolic systems. This paper, therefore, presents a new algorithm, which can be used to solve models containing any number of recycling compartments, where the initial amounts and fundamental translocation rate constants are known.

The new method preserves the simplicity and ease of use of the previous approach. The concept of a rate matrix, which contains all the information about a specific model, is retained, and, as before, the algorithm operates directly on the rate matrix to solve the system. Given the initial amount of material in each compartment at time $t = 0$, together with its radioactive decay constant, the algorithm gives both the amount in each compartment at any future time t and the number of transformations that will have occurred in each compartment up to time t.

The model is solved analytically using matrix algebra, and the algorithm is therefore fast enough to be ideally suited for implementation on any standard microcomputer. An advantage of the method is that the speed of solution is independent of the complexity of the model. Furthermore, the singularity problem encountered in the use of analytical solutions of linear chains when the total

removal rate constants of any two compartments are equal (Skrable et al. 1974) is avoided by this matrix method of solution. Because, in extreme cases, arithmetic errors can be significant when implementing the algorithm, their magnitude is estimated.

There follows a description of the algorithm and the errors associated with its application, together with a computer program written in 60 lines of BASIC* to demonstrate its use.

METHOD

The first step in solving a particular compartmental model is to number each of the compartments 1 to N. In the previous algorithm (Birchall 1986), the compartments had to be numbered in such a way that material always moved towards higher-numbered compartments. This restricts its use to non-recycling models, or those in which separate uptake retention equations have been previously derived for those compartments involved in the recycling of the material of interest. For the algorithm described in this paper, the compartments can be numbered in any order regardless of recycling. In addition, the fundamental translocation rate constants and initial amounts can be used directly to obtain numerical results, thereby overcoming the requirement for obtaining separate eigenvalues and uptake retention equations for those compartments that recycle the material of interest.

The second step is to represent the model by a rate matrix [**R**]; each element R_{ij} of the matrix contains a numerical value r_{ij} for the translocation rate constant from compartment i to compartment j, and each diagonal element R_{ii} contains a value for the initial amount $x_i(0)$ in compartment i.

After the user has assigned values to the rate matrix [**R**] and the radioactive decay constant of the material λ, the algorithm operates directly on this matrix to give for any time t the amount in each compartment at time t and the total number of disintegrations that have occurred in each compartment up to time t.

THE ALGORITHM

Analytical solution to the model

The algorithm first transforms the rate matrix [**R**] into a new matrix [**A**], by first replacing each diagonal element of [**R**] by the negative of the sum of each of the row elements and then transposing it. This enables one to use the matrix [**A**] directly, to solve the general system of simultaneous differential equations which describe the model. For example, if r_{ij} and a_{ij} are the values of the elements of the rate matrix [**R**] and the matrix [**A**], respectively, then

$$a_{ij} = r_{ji}, \quad \text{for} \quad i \neq j, \quad \text{and}$$

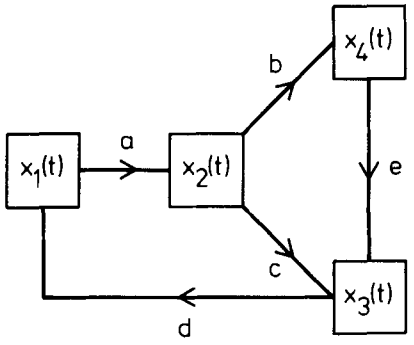
$$a_{ii} = -\sum_{\substack{j=1 \\ j \neq i}}^{N} \equiv -K_i.$$

The value of each diagonal element a_{ii} is the negative of the total rate constant K_i which equals the instantaneous fraction of the content of the ith compartment removed per unit time by all pathways other than by radioactive decay. To deal with radioactive decay at an instantaneous fractional rate λ, the diagonal elements a_{ii} of [**A**] are thus further reduced by the decay constant λ. Figure 1 shows the matrix [**A**] for a specific compartmental model. Once the matrix [**A**] is formed, it can be used directly to solve the model. If the initial amounts $x_i(0)$ in each compartment i are contained in a column vector $\mathbf{x}(0)$, it is shown, in Appendix A, that the amount in compartment i at any subsequent time t is given by

$$x_i(t) = e^{[A]t} \cdot x_i(0), \tag{1}$$

where $e^{[A]}$ is the exponential of the matrix [**A**].

(a)



(b)

$x_1(0)$	a		
	$x_2(0)$	c	b
d		$x_3(0)$	
		e	$x_4(0)$

(c)

-a		d	
a	-(b+c)		
	c	-d	e
	b		-e

Fig. 1. (a) A simple recycling compartmental model; (b) its corresponding rate matrix [**R**]; and the (c) matrix [**A**].

* Microsoft Corporation, 10700 Northup Way, Bellevue, WA 98004.

If $x_i(0)$ is in atoms, eqn (1) can be integrated with respect to time and multiplied by λ to give the total number of disintegrations $u_i(t)$:

$$u_i(t) = \lambda[\mathbf{A}]^{-1}[e^{[\mathbf{A}]t} - [\mathbf{I}]]x_i(0), \quad (2)$$

where $[\mathbf{A}]^{-1}$ is the inverse matrix of [A], and [I] is the identity matrix.

Evaluation of $e^{[\mathbf{A}]t}$

Various numerical methods exist for taking the exponential of a square matrix (Bronson 1969). Almost all of these methods require one to calculate the eigenvalues and eigenvectors. For example, if [Q] is the matrix of eigenvectors and $\exp(\phi)$ is the diagonal matrix whose elements are the exponentials of successive eigenvalues ϕ, then

$$e^{[\mathbf{A}]} = [\mathbf{Q}]\exp(\phi)[\mathbf{Q}]^{-1}. \quad (3)$$

The process of deriving the eigenvalues and eigenvectors is not straightforward. The standard Newtonian method for solution of an $N \times N$ matrix involves forming and solving the characteristic equation, which is an N-degree polynomial. The roots of this polynomial are the N eigenvalues. Unfortunately, this method is both difficult to implement and computationally slow. It is therefore not suitable as a general method. Iterative methods, such as the power methods, are more suitable but are designed only to find single eigenvalues. To find the complete eigensolution to a matrix, it is necessary to use transformation methods, which are more complicated. These reduce the original matrix to a tridiagonal one with the same eigenvalues by performing a series of similarity transformations. The eigenvalues of the tridiagonal matrix can then be evaluated more easily with standard techniques. The problem becomes even more complicated in the case under consideration when the matrix is nonsymmetric. The standard reductions do not then lead to the tridiagonal form. These problems are discussed elsewhere (Acton 1970).

To avoid such problems, a simpler approach is adopted here. The exponential of [A] is evaluated by a series expansion:

$$e^{[\mathbf{A}]} = [\mathbf{I}] + [\mathbf{A}] + \frac{1}{2!}[\mathbf{A}]^2 + \frac{1}{3!}[\mathbf{A}]^3 + \cdots. \quad (4)$$

This method is not recommended in standard texts on numerical analysis because large powers of [A] have to be calculated if a sufficient number of terms are to be included. This is not only time-consuming but it also leads to large rounding errors. In general, the larger the elements of [A], the more terms are needed to evaluate $e^{[\mathbf{A}]}$ to the same accuracy, and thus $e^{[\mathbf{A}]t}$ is more difficult to evaluate for larger t. To illustrate this, we considered a compartmental model in which each of four compartments was connected to every other by transfer rate constants equal to unity, and the initial amount in each compartment was set to 25. The amount in each compartment at various times t was calculated from eqn (1) with series expansion to evaluate $e^{[\mathbf{A}]t}$. The series was terminated when the last term contributed less than 10^{-5} of the sum of previous terms. The time taken to solve the model is proportional to the number of terms since each additional term, beyond the first two terms, requires a matrix multiplication, the most time-consuming operation. The number of terms required to evaluate $e^{[\mathbf{A}]t}$ for various t are given in the second column of Table 1. It is clear that the number of terms increases rapidly as t increases and that the method soon becomes impractical.

However, this problem can be avoided by noting that

$$e^{[\mathbf{A}]} = [e^{[\mathbf{A}]/x}]^x \quad \text{for} \quad x \neq 0. \quad (5)$$

Letting $x = 2^n$, where n is an integer, one finds

$$e^{[\mathbf{A}]} = [e^{[\mathbf{A}]/2^n}]^{2^n}. \quad (6)$$

An improved method of evaluating $e^{[\mathbf{A}]t}$ thus presents itself. One can:

(a) Reduce all the elements of [A] by a factor 2^n. It can be shown that fast convergence is achieved if n is chosen so that no element exceeds 0.2.

(b) Evaluate the exponential of the reduced matrix $[\mathbf{A}]/2^n$ by series expansion.

(c) Multiply this exponential by itself n times recursively. In this way, the number of multiplications needed is reduced from 2^n to n.

The three right-hand columns of Table 1 give the factors by which [A] is reduced (2^n), the number of terms

Table 1. A comparison of two methods for calculating $e^{[\mathbf{A}]t}$.

	Method				
	Series expansion		Modified series expansion		
Time, t	No. of terms	No. of matrix multiplications	Reduction factor (2^n)	No. of terms (k)	No. of matrix multiplications (n + k -2)
0.01	4	2	1	4	2
0.02	4	2	1	4	2
0.05	6	4	1	6	4
0.1	6	4	1	6	4
0.2	8	6	8	4	5
0.5	13	11	16	4	6
1	19	17	16	6	8
2	31	29	32	6	9
5	62	60	128	5	10
10	97	95	256	5	11
20	156	154	512	5	12
50	*	*	1024	6	14
100	*	*	2048	6	15
200	*	*	4096	6	16
500	*	*	8192	6	17
1000	*	*	16384	6	18
2000	*	*	32768	6	19
5000	*	*	131072	5	20

* method fails

(k) required for convergence, and the total number of matrix multiplications ($n + k - 2$) needed to evaluate $e^{[A]t}$ at various times for the test model above. The number of matrix multiplications for the modified procedure is to be compared with values in the second column obtained by the conventional method of series expansion.

It can be seen that the modified series expansion method requires fewer matrix multiplications and is therefore faster for all values of t. Actually, when t increases exponentially as in Table 1, the time taken to solve the model by the modified method increases linearly. This follows because the reduction factor 2^n is proportional to t, and the number of matrix multiplications at large t is approximately equal to n. Thus, $e^{[A]t}$ can be evaluated rapidly for much larger values of t. An additional advantage in reducing the number of matrix multiplications is that rounding errors are also reduced, as discussed later in the subsection, *Accuracy.*

Algorithm structure

The operation of the algorithm is described in steps (a) to (g) in Table 2. An example of a computer program in Microsoft BASIC illustrating the solution of a general recycling compartmental model is given in Appendix B. The line numbers of this example program corresponding to each step of the algorithm are given in the final column of Table 2.

APPLICATION OF THE ALGORITHM

Types of models that can be solved

The algorithm can be applied to all first-order compartmental models, both recycling and non-recycling, including all of the metabolic models recommended by the ICRP. The rate matrix is simply defined by the model, and the algorithm operates automatically on it to solve the system. Radioactive decay is treated automatically by specifying the decay rate constant λ.

The description of the systemic metabolism of elements in ICRP Publication 30 (ICRP 1979a, 1979b, 1981) varies from one element to another depending on the information available. Sometimes information given relates to the fundamental translocation rate constants and other

Table 2. Detailed description of the algorithm.

STEP	DESCRIPTION	LINE NUMBERS
(a) INITIALISE	:Number compartments in model 1 to N	
	:Enter no. of compartments in model	40
	:Enter decay constant λ	50
	:Enter truncation value, ϵ	60
	:Enter rate matrix elements R_{ij}	100
(b) CALCULATE [A]	:Transpose a_{ij} (= r_{ji})	260
	:Calculate diagonals a_{ii}	270-280
	:Multiply [A] by t	300
(c) REDUCE [A]	:Calculate reduction factor IZ	320-340
	:Reduce [A]	360
(d) CALCULATE $e^{[A]}$	:Series expansion for $e^{[A]}$	390-460
	:Test for convergence	480
(e) SQUARE $e^{[A]}$	:Square $e^{[A]}$ recursively IZ times	550-580
(f) CALCULATE $[A]^{-1}$	:Test to see if required	590
	:Invert matrix	1000-1060
(g) COMPUTE RESULTS	:Calculate $x_i(t)$ using eq. (1)	640
	:Calculate $u_i(t)$ using eq. (2)	660-700
	:Print results	710-740
	:Estimate and print errors	750-810

times to the eigenvalue or effective rate constants. Our algorithm can be used in both cases, as shown in the following two examples.

In the case of the description of the systemic metabolism of I, a recycling model that gives specific fundamental rate constants is described for the three systemic compartments identified in the model: blood, thyroid, and other tissue. The algorithm in this paper can be used directly to obtain the contents of these compartments as well as the contents of excretion compartments for either direct uptake into the blood, for inhalation intakes, or ingestion intake of I, or any combination of these. An example of how one can use the algorithm to calculate the dose equivalent from β disintegrations in the thyroid gland following ingestion of 1 Bq of ^{131}I using the standard ICRP metabolic model for I in an adult is given in Appendix C.

For stable Co, the ICRP Publication 30 systemic retention of the whole body is derived from experimental data following a single acute uptake into the blood of two males injected with a chloride solution of radioactive ^{60}Co. When fitted to a sum of exponentials, the data reveal a stable Co systemic uptake retention function $R_s(t)$ for the whole body expressed by a sum of exponential terms with constant coefficients. The rate constants and coefficients of these exponential terms are effective parameter values that already incorporate the dynamic process of recycling of Co between systemic compartments. Individual exponential terms of $R_s(t)$ cannot in reality be associated with any particular systemic compartment; however, they can be treated in our algorithm as mutually exclusive, one-way, catenary compartments for the purpose of predicting the content of the systemic whole body or total systemic excretion. The total systemic excretion can be treated just like any other compartment of the body except that the only removal of a radionuclide is by radioactive decay. In applying our algorithm to this case, each exponential term of $R_s(t)$ is treated as a separate compartment which is cleared directly to total systemic excretion at an instantaneous fractional rate given by the rate constant in that exponential. The coefficient of a particular exponential term gives the effective fraction of an uptake into the systemic circulation that deposits in that compartment.

Chronic intakes to a single compartment i at a fixed rate P_i can also be modelled by using an additional compartment. This new compartment x in the rate matrix [**R**] is set initially to contain a very large amount L ($R_{xx} = L$), and it is linked to the compartment i by a relatively small rate constant r_{xi} ($R_{xi} = r_{xi}$), such that the product of L and r_{xi} gives the desired input rate P_i. The rate of intake is then constant, provided the clearance half-life of the feeding compartment, excluding radioactive decay, is large compared to the time intervals considered. It is assumed in the program in Appendix B that all of the compartments (including the feeding compartment) are subject to radioactive decay, and in some situations this would cause a significant reduction in the input rate. This problem can be solved by modifying line 270 of the program, so that the diagonal element of the [**A**] matrix, A_{xx}, is initially set to 0 and not to $-\lambda$. An acute intake B into compartment i at $t = 0$ is simply modelled by setting $R_{ii} = B$.

The situation of a partitioned compartment where specific fractions of material are required to follow different pathways can be represented by defining new rate constants. Where a compartment i is emptied by mechanisms other than radioactive decay with a total rate constant K_i via several pathways, and predetermined fractions F_{ij} follow each pathway ij, the transfer rate constants R_{ij} are given simply by the products $K_i \times F_{ij}$.

In some special cases, when the algorithm uses λ automatically to calculate disintegrations, the results may not be sufficiently accurate. An alternative method for dealing with these cases is to treat the radioactive decay constant λ simply as a transfer rate. For example, in the model shown in Fig. 1, disintegrations in compartment 1 can also be calculated by adding two compartments, 5 and 6, to the model. Compartment 1 is then linked to 5 by λ, and compartments 2, 3 and 4 are each linked to 6 by λ. Hence, the number of disintegrations in compartment 1 up to time t is simply the amount in compartment 5 at time t, and so on. To implement this using the program in Appendix B, one must set the value of λ to zero. This method can have the advantage of increasing accuracy in some cases, as discussed later for small λ. Its disadvantage is that the additional compartments increase computation time.

Some radionuclides decay to products that are themselves radioactive. By representing radioactive decay as a transfer rate, one can readily extend the algorithm to evaluate disintegrations in chains of daughter radionuclides. This is achieved by replicating the model for each radioactive daughter and by linking the replicate compartments with the respective decay constants that link members of the radioactive series. A method for dealing automatically with any number of radioactive daughters, which exploits a symmetry in the resulting rate matrix, has been discussed previously (Birchall 1986). Furthermore, since the rate constants for the daughters are specified independently of those for the parent, this method of solution applies equally well in situations where daughters behave differently in a metabolic sense.

Of course, every compartment named in the metabolic model for a parent radionuclide also must be named for each of its progeny; otherwise, no characteristic translocation rate constants would be available to describe the metabolism of such progeny. Provided that it has the same compartments named as its parent, a given daughter nuclide can have additional compartments named in its metabolic model. Each progeny, however, must have at least all of these compartments named in the metabolic model for its immediate parent. ICRP Publication 30 (ICRP 1979) states that unless evidence is available to the contrary, daughter nuclides and all subsequent progeny produced in the body stay with and behave metabolically like the parent radionuclide. It is hoped that our algorithm and its ease of use might encourage others to do research to obtain the basic information needed for treating the metabolism of daughter radionuclides differently from

their parents, thus leading to more realistic metabolic models.

Execution time

The time taken to solve a model with N compartments completely is independent of the complexity of the model because no use is made of the sparsity of the rate matrix; the time does, however, increase roughly with N^3. In practice, the time taken to solve a model depends on the computer used and whether the program is compiled. As a rough guide, the program in Appendix B, run with any compiled BASIC on an IBM†-compatible microcomputer, completely solves models with up to 15 compartments in seconds. A model containing 30 compartments takes about a minute. For example, an N-compartment model with $N(N-1)$ random rate constants and a specified radioactive decay rate has been solved with the program in Appendix B on an Olivetti‡ M240 PC with an 8087 arithmetic co-processor, using Turbo BASIC.§ The time taken to solve the model completely for $n = 5$, 10, 15, 20, 25, 30, 40 and 50 was typically 0.7, 3, 10, 23, 45, 76, 190 and 360 s, respectively.

Accuracy

The accuracy of the algorithm depends on the accuracy of evaluating $e^{[A]}$. The elements of [A] are first scaled down by a factor 2^n, denoted here by Z, enabling its exponential to be evaluated by series expansion with only few terms. The exponential of this reduced matrix is then raised to the power Z by n recursive multiplications. Since the series expansion of the reduced matrix is truncated when a specified accuracy, ϵ, is attained, this error is propagated and magnified during the subsequent squaring procedure. In general, the relative error that results from raising a quantity to the power Z can be expressed by series expansion:

$$(1+\epsilon)^Z = 1 + Z\epsilon + \frac{Z}{2}(Z-1)\epsilon^2 + \cdots.$$

For small $Z\epsilon$, second- and higher-order terms can be ignored. The resulting relative error is therefore of the order of $Z\epsilon$. The algorithm is implemented in double-precision arithmetic, and thus ϵ can be reduced to 10^{-16}. If, for example, an overall relative error of 10^{-5} is required in the model solution, Z can be as large as 10^{11}. The practical consequences of this limited range of Z are discussed later.

Various numerical techniques for evaluating the exponential of a matrix have been reviewed by Moler and Van Loan (1978), where a full analysis of errors associated with "scaling and squaring techniques" is given. These authors conclude that the resulting algorithm is very effective.

† IBM, International Business Machines, Rochester, MN 55901.
‡ Ing. C. Olivetti & C., S.p.A. 77, Via Jervis—10015 Ivrea (Italy).
§ Borland, 4585 Scotts Valley Drive, Scotts Valley, CA 95066.

To implement this scaling and squaring method, the algorithm must be developed to select a suitable reduction factor Z. If Z is too small, the series expansion requires many terms to converge, and evaluation may be prohibitively slow. If Z is too large, the number of matrix multiplications required increases, and the error $Z\epsilon$ increases. A compromise is therefore necessary. Moler and Van Loan (1978) discussed various methods for optimizing the choice of Z depending on the required degree of accuracy and the elements of [A]. In our case, the elements of [A] are derived from rate constants in compartmental models. The choice of Z, i.e., the smallest power of 2 that reduces the largest element in [A] to less than 0.2, is made on pragmatic grounds. First, the value 0.2 was found to typify elements in [A] required to minimize the total number of matrix multiplications when solving various hypothetical and practical compartmental models. Second, this criterion is simple and readily implemented.

In practice, the elements of [A] are formed from the dimensionless product of the rate constants R_{ij} and the time interval t. The scaling factor Z is thus of the order $R_{max}t$, and the relative error is of the order $R_{max}t\epsilon$. In the limiting case, with $\epsilon = 10^{-16}$ and a required relative error of, say, 10^{-5}, it follows that

$$R_{max}t < 10^{11}. \quad (7)$$

As t tends to infinity, this condition is not satisfied, and control of the relative error is lost. In a practical system, however, the range of interest of t is limited for the following reason. A compartmental system is either "closed" and thus reaches equilibrium, or "open" and thus empties. In many cases, the value of t required to achieve a steady state is determined by the smallest rate constant R_{min} when $R_{min}t$ is of the order 10, which requires a time t of about $10/R_{min}$. When this time is substituted into eqn (7), the limit for the ratio of the largest to the smallest rate constant is obtained:

$$\frac{R_{max}}{R_{min}} < 10^{10}. \quad (8)$$

In such cases, the general algorithm solves any model to a relative accuracy better than 10^{-5} over the time range of interest provided the range of rate constants within the model does not exceed 10 orders of magnitude. Even for problems where the compartmental amounts are very small, the algorithm evaluates the model with the required relative accuracy up to a maximum time t_{max} defined by eqn (7). For this type of problem, however, the time required to reach a steady state may be very large compared to $1/R_{min}$, and the time range of interest may well exceed t_{max}. This would introduce serious errors. Similarly, for models involving rate constants that differ by more than ten orders of magnitude, significant truncation errors may arise.

To ensure that the implementation of the algorithm is both convergent and stable, estimates of the errors caused by truncation and rounding must be made. Truncation errors are estimated by multiplying the error ϵ in

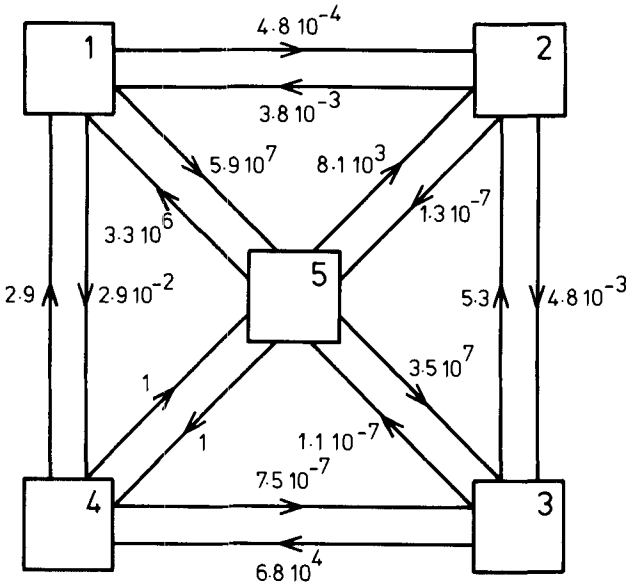


Fig. 2. A hypothetical example of a closed model with five compartments and an extreme range of transfer rate constants.

the series expansion of $e^{[A]t}$ by the matrix-reduction factor Z. Rounding errors are estimated by applying the conservation principle. Since the sum of the amounts, and the sum of disintegrations, in all compartments of a model can be calculated accurately and independently, they can be compared to those given by the algorithm. These two errors are referred to as "overall errors." The program listed in Appendix B automatically gives (a) the theoretical truncation error; (b) the estimated overall error in the amounts; (c) the estimated overall error in the number of disintegrations. In practice, truncation errors tend to dominate, but rounding errors become increasingly important for larger models. It is recommended that if any of these three estimated errors is unacceptably high, a smaller value of ϵ should be used in line 60 of the program. This will increase the accuracy at the expense of increasing the computation time.

To examine the utility of the estimated errors and the accuracy of the algorithm under extreme conditions, several hypothetical examples are considered below. In all of these cases, the algorithm was operating with maximum accuracy, i.e., with $\epsilon = 10^{-16}$.

Tests using hypothetical examples

The first hypothetical test case was that of a closed compartmental system with a severely wide range of transfer rate constants, as shown in Fig. 2. All rate constants were chosen at random in the logarithmic interval 10^{-8} to 10^{8} except those linking compartments 4 and 5, which were given the value unity. In this example, the initial amount in each compartment is 20. The residual amount at times ranging from 10^{-8} to 10^{8} were calculated using the algorithm. An estimate of the error in the amount in each compartment was obtained by comparing its value with that calculated on a mainframe computer using the FACSIMILE routine for solving general differential equations (Curtis 1977; Curtis and Sweetenham 1985). The results of this test are given in Table 3. The

Table 3. Relative errors in the amounts evaluated for the closed compartmental model in Fig. 2.

Time t	Rounded compartmental amount (relative error)					Estimated overall error	Theoretical error, $Z\epsilon$
	1	2	3	4	5		
10^{-8}	$1.16\ 10^{1}$ (*)	$2.00\ 10^{1}$ (*)	$2.73\ 10^{1}$ (*)	$2.00\ 10^{1}$ (*)	$2.11\ 10^{1}$ (*)	$< 1\ 10^{-16}$	$4\ 10^{-16}$
10^{-7}	$2.79\ 10^{-1}$ (*)	$2.00\ 10^{1}$ (*)	$5.73\ 10^{1}$ (*)	$2.03\ 10^{1}$ (*)	$2.15\ 10^{1}$ (*)	$7\ 10^{-16}$	$3\ 10^{-15}$
10^{-6}	$1.32\ 10^{-6}$ (*)	$2.00\ 10^{1}$ (*)	$5.61\ 10^{1}$ (*)	$2.38\ 10^{1}$ (*)	$2.64\ 10^{-6}$ (*)	$1\ 10^{-14}$	$5\ 10^{-14}$
10^{-5}	$2.74\ 10^{-6}$ (*)	$2.00\ 10^{1}$ (*)	$3.04\ 10^{1}$ (*)	$4.95\ 10^{1}$ (*)	$5.51\ 10^{-6}$ (*)	$3\ 10^{-13}$	$4\ 10^{-13}$
10^{-4}	$4.43\ 10^{-6}$ (*)	$2.00\ 10^{1}$ (*)	$7.15\ 10^{-2}$ (*)	$7.99\ 10^{1}$ (*)	$8.90\ 10^{-6}$ (*)	$7\ 10^{-13}$	$3\ 10^{-12}$
10^{-3}	$4.43\ 10^{-6}$ (*)	$2.00\ 10^{1}$ (*)	$4.59\ 10^{-3}$ (*)	$8.00\ 10^{1}$ (*)	$8.91\ 10^{-6}$ (*)	$6\ 10^{-12}$	$5\ 10^{-11}$
10^{-2}	$4.43\ 10^{-6}$ (*)	$2.00\ 10^{1}$ (*)	$4.59\ 10^{-3}$ (*)	$8.00\ 10^{1}$ (*)	$8.91\ 10^{-6}$ (*)	$6\ 10^{-11}$	$4\ 10^{-10}$
10^{-1}	$4.43\ 10^{-6}$ (*)	$2.00\ 10^{1}$ (*)	$4.59\ 10^{-3}$ (*)	$8.00\ 10^{1}$ (*)	$8.91\ 10^{-6}$ (*)	$4\ 10^{-10}$	$3\ 10^{-9}$
1	$4.44\ 10^{-6}$ (*)	$1.99\ 10^{1}$ (*)	$4.59\ 10^{-3}$ (*)	$8.01\ 10^{1}$ (*)	$8.92\ 10^{-6}$ (*)	$3\ 10^{-8}$	$5\ 10^{-8}$
10^{1}	$4.47\ 10^{-6}$ (*)	$1.93\ 10^{1}$ (*)	$4.63\ 10^{-3}$ (*)	$8.07\ 10^{1}$ (*)	$8.99\ 10^{-6}$ (*)	$5\ 10^{-7}$	$4\ 10^{-7}$
10^{2}	$4.70\ 10^{-6}$ (*)	$1.52\ 10^{1}$ (*)	$4.86\ 10^{-3}$ (*)	$8.48\ 10^{1}$ (*)	$9.45\ 10^{-6}$ (*)	$3\ 10^{-7}$	$3\ 10^{-6}$
10^{3}	$4.86\ 10^{-6}$ (*)	$1.23\ 10^{1}$ (*)	$5.03\ 10^{-3}$ (*)	$8.77\ 10^{1}$ (*)	$9.77\ 10^{-6}$ (0.00001)	$7\ 10^{-6}$	$5\ 10^{-5}$
10^{4}	$4.86\ 10^{-6}$ (0.0001)	$1.23\ 10^{1}$ (0.0002)	$5.03\ 10^{-3}$ (0.0001)	$8.77\ 10^{1}$ (0.0001)	$9.77\ 10^{-6}$ (0.0002)	$2\ 10^{-4}$	$4\ 10^{-4}$
10^{5}	$4.88\ 10^{-6}$ (0.005)	$1.24\ 10^{1}$ (0.005)	$5.05\ 10^{-3}$ (0.005)	$8.81\ 10^{1}$ (0.005)	$9.82\ 10^{-6}$ (0.005)	$5\ 10^{-3}$	$4\ 10^{-3}$
10^{6}	$4.84\ 10^{-6}$ (0.004)	$1.23\ 10^{1}$ (0.004)	$5.01\ 10^{-3}$ (0.004)	$8.73\ 10^{1}$ (0.004)	$9.73\ 10^{-6}$ (0.004)	$4\ 10^{-3}$	$6\ 10^{-2}$
10^{7}	$4.97\ 10^{-6}$ (0.02)	$1.26\ 10^{1}$ (0.02	$5.14\ 10^{-3}$ (0.02)	$8.97\ 10^{1}$ (0.02)	$9.99\ 10^{-6}$ (0.02)	$2\ 10^{-2}$	$5\ 10^{-1}$
10^{8}	$1.12\ 10^{-5}$ (2.3)	$2.83\ 10^{1}$ (2.3)	$1.16\ 10^{-2}$ (2.3)	$2.02\ 10^{2}$ (2.3)	$2.25\ 10^{-5}$ (2.3)	2.3	3.6

(*) Relative error $< 10^{-5}$, i.e. better than 5 significant figures given by FACSIMILE (Curtis 1977)

Table 4. Relative errors in the amounts evaluated for compartment 4 in the second test case.

Time	Amount in compartment 4		Estimated relative error
t	Exact*	Algorithm	
10^{-10}	5.0000×10^{-29}	5.0000×10^{-29}	$< 10^{-16}$
10^{-9}	4.9998×10^{-27}	4.9998×10^{-27}	$< 10^{-16}$
10^{-8}	4.9983×10^{-25}	4.9983×10^{-25}	$< 10^{-16}$
10^{-7}	4.9834×10^{-23}	4.9834×10^{-23}	$< 10^{-16}$
10^{-6}	4.8374×10^{-21}	4.8374×10^{-21}	$< 10^{-16}$
10^{-5}	3.6788×10^{-19}	3.6788×10^{-19}	1×10^{-15}
10^{-4}	9.0000×10^{-18}	9.0000×10^{-18}	3×10^{-15}
10^{-3}	9.9000×10^{-17}	9.9000×10^{-17}	6×10^{-14}
10^{-2}	9.9900×10^{-16}	9.9900×10^{-16}	2×10^{-13}
10^{-1}	9.9990×10^{-15}	9.9990×10^{-15}	3×10^{-13}
1	9.9999×10^{-14}	9.9999×10^{-14}	3×10^{-11}
10^{1}	1.0000×10^{-12}	1.0000×10^{-12}	1×10^{-9}
10^{2}	1.0000×10^{-11}	1.0000×10^{-11}	3×10^{-9}
10^{3}	1.0000×10^{-10}	1.0000×10^{-10}	3×10^{-8}
10^{4}	1.0000×10^{-9}	1.0000×10^{-9}	3×10^{-7}
10^{5}	1.0000×10^{-8}	1.0000×10^{-8}	5×10^{-6}
10^{6}	1.0000×10^{-7}	9.9999×10^{-8}	3×10^{-5}
10^{7}	1.0000×10^{-6}	9.9996×10^{-7}	8×10^{-5}
10^{8}	1.0000×10^{-5}	1.0005×10^{-5}	1×10^{-3}
10^{9}	1.0000×10^{-4}	1.0175×10^{-4}	4×10^{-2}
10^{10}	9.9999×10^{-4}	1.6989×10^{-3}	1×10^{7}
10^{11}	9.9995×10^{-3}	3.8729×10^{1}	4×10^{4}
10^{12}	9.9950×10^{-2}	5.7767×10^{10}	2×10^{13}
10^{13}	9.9502×10^{-1}	5.9040×10^{108}	2×10^{111}
10^{14}	9.5163×10^{0}	-	-
10^{15}	6.3212×10^{1}	-	-
10^{16}	9.9995×10^{1}	-	-
10^{17}	1.0000×10^{2}	-	-

* Amount calculated to 5 significant figures by FACSIMILE (Curtis 1977)

table also shows the overall error estimated by the algorithm at each time interval and the theoretical error $Z\epsilon$, where Z is the factor by which the matrix elements in [A] are reduced and ϵ is the truncation error (i.e., 10^{-16}) associated with the series expansion of $e^{[A]}$.

It is seen from Table 3 that the amounts in each compartment are correct to within the overall estimated error at all times over the range 10^{-8} to 10^{8}. The estimated overall error is seen to reflect the relative error in each compartment even when the amounts in the compartments differ by many orders of magnitude. For example, the amounts in compartments 1 and 4 at time $t = 10^{6}$ differ by seven orders of magnitude, yet they have the same relative error. Furthermore, the estimated overall error is predicted by the theoretical error $Z\epsilon$.

In practice, all closed systems effectively reach a steady state. It is therefore only necessary to evaluate them up to this point. In this example, it can be shown that the amount in any compartment does not change by more than 0.001% after $t = 10^{3}$. This is well within the range of t evaluated accurately, i.e., with relative error $< 10^{-5}$.

A second test case is a 4-compartment system which is deliberately conditioned to result in extremely slow movement of material towards compartment 4. Four transfer rate constants, all equal to 10^{-5}, link compartments 1 to 2, 2 to 1, 1 to 3, and 3 to 4, with all the material (100) in compartment 1 at $t = 0$. At $t = \infty$, all material would clearly be in compartment 4. The final state is not reached until $t \approx 10^{16}$. The amount in compartment 4

Table 5. Relative errors in the amounts and disintegrations evaluated for the compartmental model in Fig. 2.

	Decay constant, λ													
	10^{9}		10^{6}		10^{3}		1		10^{-3}		10^{-6}		10^{-9}	
Time	amt/dis		amt/dis		amt/dis		amt/dis		amt/dis		amt/dis		amt/dis	
10^{-8}	10^{-15}	10^{-16}	$<10^{-16}$	10^{-15}	10^{-15}	10^{-10}	$<10^{-16}$	10^{-8}	10^{-16}	10^{-6}	10^{-16}	10^{-2}	$<10^{-16}$	>1
10^{-7}	10^{-13}	10^{-16}	10^{-15}	10^{-14}	10^{-15}	10^{-11}	10^{-15}	10^{-8}	10^{-15}	10^{-5}	10^{-16}	10^{-3}	10^{-15}	>1
10^{-6}	*	10^{-16}	10^{-14}	10^{-14}	10^{-14}	10^{-11}	10^{-14}	10^{-8}	10^{-14}	10^{-5}	10^{-15}	10^{-3}	10^{-14}	>1
10^{-5}	*	10^{-16}	10^{-13}	10^{-16}	10^{-13}	10^{-11}	10^{-13}	10^{-8}	10^{-13}	10^{-5}	10^{-13}	10^{-2}	10^{-13}	>1
10^{-4}	*	10^{-16}	10^{-12}	10^{-16}	10^{-12}	10^{-11}	10^{-14}	10^{-10}	10^{-12}	10^{-5}	10^{-13}	10^{-3}	10^{-12}	>1
10^{-3}	*	10^{-16}	*	10^{-16}	10^{-11}	10^{-11}	10^{-11}	10^{-8}	10^{-11}	10^{-5}	10^{-12}	10^{-3}	10^{-12}	>1
10^{-2}	*	10^{-16}	*	10^{-16}	10^{-10}	10^{-15}	10^{-11}	10^{-9}	10^{-10}	10^{-5}	10^{-10}	10^{-2}	10^{-10}	>1
10^{-1}	*	10^{-16}	*	10^{-16}	10^{-11}	10^{-16}	10^{-9}	10^{-8}	10^{-9}	10^{-5}	10^{-10}	10^{-3}	10^{-9}	>1
1	*	10^{-16}	*	10^{-16}	*	10^{-16}	10^{-7}	10^{-8}	10^{-8}	10^{-5}	10^{-8}	10^{-2}	10^{-8}	>1
10^{1}	*	10^{-16}	*	10^{-16}	*	10^{-16}	10^{-7}	10^{-11}	10^{-7}	10^{-5}	10^{-7}	10^{-2}	10^{-7}	>1
10^{2}	*	10^{-16}	*	10^{-16}	*	10^{-16}	10^{-6}	10^{-16}	10^{-6}	10^{-5}	10^{-7}	10^{-3}	10^{-7}	>1
10^{3}	*	10^{-16}	*	10^{-16}	*	10^{-16}	*	10^{-16}	10^{-5}	10^{-6}	10^{-5}	10^{-2}	10^{-5}	>1
10^{4}	*	10^{-16}	*	10^{-16}	*	10^{-16}	*	10^{-16}	10^{-4}	10^{-8}	10^{-4}	10^{-2}	10^{-4}	>1
10^{5}	*	10^{-16}	*	10^{-16}	*	10^{-16}	*	10^{-16}	10^{-3}	10^{-13}	10^{-3}	10^{-2}	10^{-3}	>1
10^{6}	*	10^{-16}	*	10^{-16}	*	10^{-16}	*	10^{-16}	*	10^{-13}	10^{-3}	10^{-3}	10^{-3}	>1
10^{7}	*	10^{-16}	*	10^{-16}	*	10^{-16}	*	10^{-16}	*	10^{-13}	10^{-2}	10^{-6}	10^{-2}	>1
10^{8}	*	10^{-16}	*	10^{-16}	*	10^{-16}	*	10^{-16}	*	10^{-13}	>1	10-9	>1	>1

* Errors in the amounts are impossible to evaluate since practically all the material has decayed.

was calculated with the algorithm for times ranging from 10^{-10} to 10^{17}. Table 4 shows these amounts, their estimated relative errors, and the values calculated by FACSIMILE (Curtis 1977; Curtis and Sweetenham 1985).

It is seen that the calculated amounts in compartment 4 are again correct to within the estimated overall error even if these amounts are extremely small (i.e., 10^{-29}). Although compartment 4 is not evaluated accurately for $t > 10^9$, the estimated error reflects this condition.

Finally, to test the application of the algorithm to cases involving radioactive decay costants, the severe compartmental model in Fig. 2 was used for material with radioactive decay constants ranging from 10^{-9} to 10^9. It can be shown that the amount and disintegrations in each compartment over the time interval 10^{-8} to 10^8 are calculated to within the estimated overall error. Table 5 shows the relative error in both the amount and disintegrations as a function of both the time t and the radioactive decay constant λ.

It is seen that the errors in the amounts follow the previous pattern—they increase with t. In practice, however, the time range of interest is limited because all compartments are effectively empty after a finite number of radioactive half-lives, and no further disintegrations can occur. It is also seen that the errors in the amounts are insensitive to the radioactive decay rate λ. Conversely, the accuracy of the calculated disintegrations increases with t and does depend on λ; as λ decreases, the accuracy also decreases. Thus, the algorithm is least accurate when t and λ are both very small. In such circumstances, the alternative method of treating λ as a transfer rate constant can be used, as described earlier.

The amount and disintegrations in each compartment of the model in Fig. 2 have been recalculated over the same ranges of t and λ using the alternative method in which λ is treated as a translocation rate constant describing transfer to additional compartments where "decayed atoms" are accumulated. The relative errors in the amount and disintegrations are shown in Table 6. Comparison with the values in Table 5 shows that the errors on the amounts are the same. The errors on disintegrations are, however, different; the accuracy increases for smaller t and for smaller λ. This behavior of the alternative method complements that of the standard method, and it can be exploited in situations where the automatic treatment of λ gives insufficient accuracy.

SUMMARY

The algorithm presented here enables any first-order compartmental model, both recycling and non-recycling, to be solved. In practice, the accuracy and speed of execution depend both on the model and the method of implementing the algorithm. It can be implemented on any standard microcomputer using the program presented here.

Table 6. Relative errors in the amounts and disintegrations evaluated for the compartmental model in Fig. 2 when λ is treated as a translocation rate constant.

	Decay constant, λ													
	10^9		10^6		10^3		1		10^{-3}		10^{-6}		10^{-9}	
Time	amt	dis	amt	dis	amt	dis	amt	dis	amt	dis	amt	dis	amt	dis
10^{-8}	10^{-15}	10^{-15}	$<10^{-16}$	$<10^{-16}$	10^{-16}	10^{-16}	$<10^{-16}$	10^{-16}	10^{-16}	$<10^{-16}$	10^{-16}	10^{-16}	$<10^{-16}$	$<10^{-16}$
10^{-7}	10^{-13}	10^{-16}	10^{-15}	10^{-16}	10^{-15}	10^{-15}	10^{-15}	10^{-16}	10^{-15}	10^{-16}	10^{-16}	10^{-16}	10^{-15}	10^{-16}
10^{-6}	*	$<10^{-16}$	10^{-14}	10^{-15}	10^{-14}	10^{-14}	10^{-14}	10^{-14}	10^{-14}	10^{-14}	10^{-15}	10^{-15}	10^{-14}	10^{-14}
10^{-5}	*	$<10^{-16}$	10^{-13}	10^{-14}	10^{-13}	10^{-13}	10^{-13}	10^{-14}	10^{-12}	10^{-12}	10^{-13}	10^{-13}	10^{-13}	10^{-13}
10^{-4}	*	10^{-16}	10^{-12}	10^{-14}	10^{-12}	10^{-13}	10^{-14}	10^{-13}	10^{-12}	10^{-13}	10^{-13}	10^{-13}	10^{-12}	10^{-13}
10^{-3}	*	$<10^{-16}$	*	10^{-14}	10^{-11}	10^{-11}	10^{-11}	10^{-11}	10^{-11}	10^{-11}	10^{-12}	10^{-12}	10^{-12}	10^{-12}
10^{-2}	*	10^{-15}	*	10^{-14}	10^{-10}	10^{-11}	10^{-11}	10^{-11}	10^{-10}	10^{-10}	10^{-10}	10^{-10}	10^{-10}	10^{-11}
10^{-1}	*	10^{-16}	*	10^{-14}	10^{-11}	10^{-13}	10^{-9}	10^{-9}	10^{-9}	10^{-9}	10^{-10}	10^{-10}	10^{-9}	10^{-10}
1	*	10^{-15}	*	10^{-14}	*	10^{-11}	10^{-7}	10^{-8}	10^{-8}	10^{-8}	10^{-8}	10^{-8}	10^{-8}	10^{-8}
10^1	*	10^{-16}	*	10^{-14}	*	10^{-11}	10^{-7}	10^{-8}	10^{-7}	10^{-7}	10^{-7}	10^{-7}	10^{-7}	10^{-7}
10^2	*	10^{-16}	*	10^{-14}	•	10^{-11}	10^{-6}	10^{-8}	10^{-6}	10^{-7}	10^{-7}	10^{-7}	10^{-7}	10^{-7}
10^3	*	10^{-16}	*	10^{-15}	*	10^{-11}	*	10^{-8}	10^{-5}	10^{-6}	10^{-5}	10^{-5}	10^{-5}	10^{-6}
10^4	*	10^{-16}	*	10^{-14}	*	10^{-11}	*	10^{-8}	10^{-4}	10^{-5}	10^{-4}	10^{-4}	10^{-4}	10^{-4}
10^5	*	10^{-15}	*	10^{-14}	*	10^{-11}	*	10^{-8}	10^{-3}	10^{-5}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
10^6	*	10^{-15}	*	10^{-14}	*	10^{-12}	*	10^{-8}	•	10^{-6}	10^{-3}	10^{-3}	10^{-3}	10^{-3}
10^7	*	10^{-15}	*	10^{-14}	*	10^{-12}	*	10^{-8}	•	10^{-6}	10^{-2}	10^{-3}	10^{-2}	10^{-2}
10^8	*	10^{-16}	*	10^{-14}	*	10^{-11}	*	10^{-8}	*	10^{-5}	>1	10^{-2}	>1	0.5

* Errors in the amounts are impossible to evaluate since practically all the material has decayed.

When implemented on a microcomputer, a relative accuracy $< 10^{-5}$ can always be achieved provided: (a) the range of rate constants in a model does not exceed 10 orders of magnitude and (b) the product of the time and the largest rate constant does not exceed 10^{11}. Truncation and rounding errors are automatically estimated by the algorithm to confirm that the method is both convergent and stable whenever it is used. By compiling and running the program on a standard microcomputer one can solve models with 15 compartments in seconds, and those with 30 compartments in a minute.

The algorithm, therefore, is especially suited for implementation on a microcomputer for solving models of interest in the field of radiological protection. Furthermore, it is very simple to apply: the initial amount in each compartment i is represented by the diagonal elements R_{ii} and each rate constant in a particular model describing transfer from compartment i to j is represented as an element R_{ij} of a rate matrix [**R**]. The algorithm then operates directly on this matrix. Additional software or specialized mathematical knowledge is not needed no matter how complex the compartmental model. If desired, the whole algorithm could be written as a compact subroutine which solves the compartmental model contained in the rate matrix. Users would then be free to design their own input and output to suit their particular need.

REFERENCES

Acton, F. S. Numerical methods that work. New York: Harper & Row; 1970.

Birchall, A. A microcomputer algorithm for solving compartmental models involving radionuclide transformations. Health Phys. 50(3):389–397; 1986.

Bronson, R. Matrix methods. London: Academic Press; 1969.

Curtis, A. R. FACSIMILE: Introduction to a computer program for initial value problems. United Kingdom Atomic Energy Authority, AERE-R 8979; 1977.

Curtis, A. R.; Sweetenham, W. P. FACSIMILE: Release H user's manual. United Kingdom Atomic Energy Authority, AERE-R 11771; 1985.

Greenhalgh, J. R.; Fell, T. P.; Adams, N. Doses from intakes of radionuclides by adults and young people. National Radiological Protection Board, NRPB R-162; 1985.

International Commission on Radiological Protection. Limits for intakes of radionuclides by workers. Oxford: Pergamon Press; ICRP Publication 30, Part 1; Ann. ICRP 2(3/4):1–116; 1979a.

International Commission on Radiological Protection. Limits for intakes of radionuclides by workers. Oxford: Pergamon Press; ICRP Publication 30, Supplement to Part 1; Ann. ICRP 3(1–4); 1979b.

International Commission on Radiological Protection. Limits for intakes of radionuclides by workers. Oxford: Pergamon Press; ICRP Publication 30, Part 3; Ann. ICRP 6(2/3); 1981.

International Commission on Radiological Protection. Radionuclide transformations, energy and intensity of emissions. Oxford: Pergamon Press; ICRP Publication 38, Ann. ICRP 11–13; 1983.

Moler, C.; Van Loan, C. Nineteen dubious ways to compute the exponential of a matrix. SIAM Rev. 20(4):801–836; 1978.

Skrable, K. W.; French, C.; Chabot, G.; Major, A. A general equation for the kinetics of linear first order phenomena and suggested applications. Health Phys. 27:155–157; 1974.

APPENDIX A

Derivation of matrix solution for recycling models

The rate of change in the amount of material in any compartment of a model is generally determined by two competing processes: the increase from pathways entering and the decrease from pathways leaving the compartment. If the element R_{ij} of the rate matrix [**R**] has the numerical value r_{ij}, which represents the instantaneous fraction of the content of compartment i translocated per unit time to compartment j, the set of linear first-order differential equations describing the system can be written

$$\frac{dx_i}{dt} = \sum_{\substack{j=1 \\ j \neq i}}^{N} r_{ji}x_j - x_i \sum_{\substack{j=1 \\ j \neq i}}^{N} r_{ij},$$

where x_i and x_j are understood to be functions of time t representing the contents of compartments i and j, respectively.

A matrix denoted by [**A**] is then defined such that its elements a_{ij} are given by

$$a_{ij} = r_{ji}, \quad \text{for} \quad i = 1 \text{ to } N, \quad j = 1 \text{ to } N, \quad \text{and} \quad i \neq j$$

$$\text{and} \quad a_{ii} = -\sum_{\substack{j=1 \\ j \neq i}}^{N} r_{ij}, \quad \text{for} \quad i = 1 \text{ to } N.$$

Substituting a_{ij} and a_{ii} in this equation, one finds

$$\frac{dx_i}{dt} = \sum_{\substack{j=1 \\ j \neq i}}^{N} a_{ij}x_j + a_{ii}x_i,$$

thus,

$$\frac{dx_i}{dt} = \sum_{j=1}^{N} a_{ij}x_j.$$

If the N values of x_j are regarded as a column vector $\mathbf{x}$, the right-hand side of this equation is equivalent to a matrix multiplication of [**A**] by $\mathbf{x}$, i.e.,

$$\frac{d\mathbf{x}}{dt} = [\mathbf{A}]\mathbf{x}.$$

This can be solved by the rules of matrix algebra given in any standard text (Bronson 1969) to yield

$$\mathbf{x} = e^{[\mathbf{A}]t}\mathbf{x}(0),$$

where $\mathbf{x}(0)$ is the column vector of initial amounts in each compartment.

APPENDIX B: A BASIC PROGRAM ILLUSTRATING THE ALGORITHM

```
5 REM ### EXAMPLE PROGRAM WRITTEN IN BASIC ###
10 DEFDBL A-Z:DEFINT I-K
20 DIM R(50,50),A(50,50),SUM(50,50),TERM(50,50),B(50,50),XT(50),XO(50)
30 DIM U(50),Q(50,100),QI(50,100)
40 N=2                                                  ' No.comps
50 LAM=LOG(2)/1000                                      ' Decay
60 TERR=1E-10                                           ' Trunk.Err
100 R(1,2)=.1:R(2,1)=.1:R(1,1)=100                      ' Rate Matrix
240 CLS:INPUT " 1. Enter Time : ",T:D=LAM*T
250 PRINT " 2. Calculating the A matrix
260 FOR I=1 TO N:FOR J=1 TO N:SUM(I,J)=0:A(I,J)=R(J,I):NEXT J,I
270 FOR I=1 TO N:A(I,I)=-LAM:FOR K=1 TO N:IF K=I THEN 290
280 A(I,I)=A(I,I)-R(I,K)
290 NEXT K,I:FOR I=1 TO N:XO(I)=R(I,I):SUM(I,I)=1:TERM(I,I)=1:NEXT I
300 FOR I=1 TO N:FOR J=1 TO N:Q(I,J)=A(I,J):A(I,J)=A(I,J)*T:NEXT J,I
320 MAX=0:FOR I=1 TO N:IF A(I,I)<MAX THEN MAX=A(I,I)
330 NEXT I:FOR IZ=0 TO 10000:IF -MAX/2^IZ<.2 THEN 350
340 NEXT IZ
350 PRINT " 3. Reducing [A] by a factor "2^IZ
360 FOR I=1 TO N:FOR J=1 TO N:A(I,J)=A(I,J)/2^IZ:NEXT J,I
380 PRINT " 4. Calculating the exponential of [A]";
390 FOR IR=1 TO 10000:PRINT IR;:FOR J=1 TO N:FOR I=1 TO N:B(I,J)=0:FOR K=1 TO N
420 B(I,J)=B(I,J)+TERM(I,K)*A(K,J):NEXT K,I,J
440 FOR I=1 TO N:FOR J=1 TO N:TERM(I,J)=B(I,J)/IR:SUM(I,J)=SUM(I,J)+TERM(I,J)
460 NEXT J,I:FOR I=1 TO N:FOR J=1 TO N:IF SUM(I,J)=0 THEN 490
480 IF TERM(I,J)/SUM(I,J) > TERR THEN 500
490 NEXT J,I:GOTO 520
500 NEXT IR:PRINT
520 PRINT " 5. Satisfactory Convergence Achieved"
540 PRINT " 6. Squaring [A] "IZ "times"
550 FOR ID=1 TO IZ:FOR I=1 TO N:FOR J=1 TO N:A(I,J)=0:FOR K=1 TO N
570 A(I,J)=A(I,J)+SUM(I,K)*SUM(K,J):NEXT K,J,I
580 FOR I=1 TO N:FOR J=1 TO N:SUM(I,J)=A(I,J):NEXT J,I,ID
590 IF LAM=0 THEN PRINT " 7. No need to invert [A]":GOTO 630
600 PRINT " 7. Taking the Inverse of [A]":GOSUB 10000
630 PRINT " 8. Computing Results":PRINT
640 FOR I=1 TO N:XT(I)=0:FOR K=1 TO N:XT(I)=XT(I)+SUM(I,K)*XO(K):NEXT K,I
650 IF LAM=0 THEN 710
660 FOR I=1 TO N:SUM(I,I)=SUM(I,I)-1:NEXT I
670 FOR I=1 TO N:FOR J=1 TO N:B(I,J)=0:FOR ID=1 TO N
680 B(I,J)=B(I,J)+QI(I,ID)*SUM(ID,J):NEXT ID,J,I
690 FOR I=1 TO N:U(I)=0:FOR J=1 TO N:U(I)=U(I)+B(I,J)*XO(J):NEXT J
700 U(I)=U(I)*LAM:NEXT I
710 PRINT "COMP.","INITIAL","AMOUNT","DISINTS"
720 SUMXO=0:FOR I=1 TO N:PRINT I,XO(I),XT(I),U(I):SUMXO=SUMXO+XO(I):NEXT I
730 SUM=0:USUM=0:FOR I=1 TO N:SUM=SUM+XT(I):USUM=USUM+U(I):NEXT I
740 L$="---------------":PRINT TAB(31) L$,L$:PRINT TAB(31) SUM,USUM
750 IF D>1000 THEN RESUM=0 ELSE SUMT=SUMXO*EXP(-D):RESUM=(SUM-SUMT)/SUMT
760 IF D>1000 THEN DIST=1 ELSE DIST=1-EXP(-D)
770 IF D<.0001 THEN DIST=D-D^2/2+D^3/6-D^4/24+D^5/120
780 IF DIST<>0 THEN REDIS=(USUM-SUMXO*DIST)/SUMXO/DIST
790 PRINT:PRINT "Estimated relative error on amount          : "RESUM
800 PRINT "Estimated relative error on disintegrations : " REDIS
810 PRINT "Theoretical truncation error                : " TERR*2^IZ
9999 STOP
10000 REM ### SUB TO INVERT AN NXN MATRIX [Q] to [QI]  ###
10010 FOR I1=1 TO N:FOR J1=N+1 TO 2*N:Q(I1,J1)=0:NEXT J1:Q(I1,I1+N)=1#:NEXT I1
10020 FOR IP1=1 TO N:FOR K1=1 TO N:QI(IP1,K1)=Q(IP1,K1+IP1)/Q(IP1,IP1):NEXT K1
10030 FOR J1=1 TO N:IF IP1=J1 THEN 10050
10040 FOR K1=1 TO N:QI(J1,K1)=Q(J1,K1+IP1)-Q(J1,IP1)*QI(IP1,K1):NEXT K1
10050 NEXT J1:FOR J1=1 TO N:FOR K1=1 TO N:Q(J1,K1+IP1)=QI(J1,K1):NEXT K1,J1,IP1
10060 RETURN
```

APPENDIX C

An example of how to use the algorithm to calculate the committed dose equivalent from β disintegrations in the thyroid gland following ingestion of 1 Bq of ^{131}I by use of the standard ICRP metabolic model for I in an adult.

A metabolic model for I is given in ICRP Publication 30 (ICRP 1979a). It is assumed that activity in the stomach moves to the small intestine with a mean residence time of 1 h where it is instantly translocated to blood. Of I entering the blood compartment, a fraction, 0.3, is translocated to the thyroid while the remainder goes directly to excretion. From the thyroid, the I is gradually distributed uniformly to all organs in the body, where a fraction 0.9 is recycled back to blood and the rest excreted. The half-lives in the blood and body compartments are 0.25 d and 12 d, respectively. The half-life in the thyroid is given as 120 d (ICRP 1979a) but has been subsequently amended to 80 d (ICRP 1981).

First, a compartmental model must be constructed and numbered, as in Fig. 3. Second, the program given in Appendix B must be modified by entering the following information:

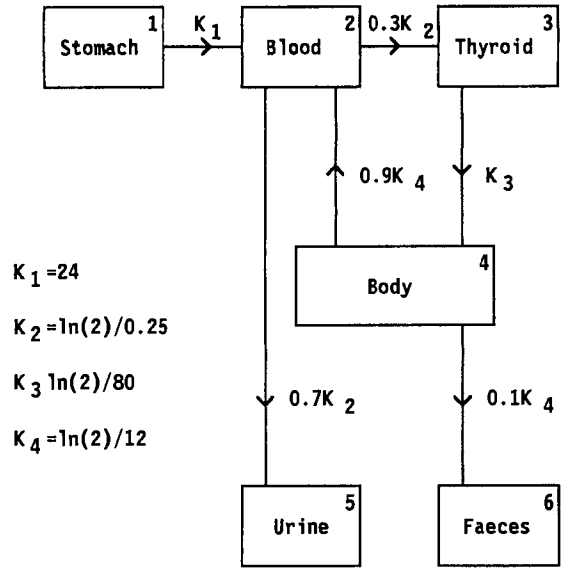


Fig. 3. The ICRP metabolic model for I in adults.

1. The number of compartments in the model, i.e.,
   40 $N = 6$.
2. The radioactive decay constant λ for ^{131}I d^{-1}, i.e.,
   50 LAM = LOG(2)/8.04.
3. The elements of the rate matrix **[R]**.
   3.1. The diagonal elements R_{ii} which contain the initial number of atoms in each compartment, i.e.,
   100 R(1,1) = 86400/LAM.
   3.2. The other elements R_{ij} which contain the fundamental transfer rate constants d^{-1}, i.e.,
   110 R(1,2) = 24
   120 R(2,3) = 0.3*LOG(2)/0.25
   130 R(2,5) = 0.7*LOG(2)/0.25
   140 R(3,4) = LOG(2)/80
   150 R(4,2) = 0.9*LOG(2)/12
   160 R(4,6) = 0.1*LOG(2)/12.

The program can now be run and prompts for a time, t, which should be entered in days. It then gives the number of atoms in each compartment at time t and the number of radioactive disintegrations that have occurred in each compartment up to time t. In this example, the number of β decays after a suitably long time (5000 d) is given to be $2.675\ 10^5$. This compares with the value quoted by ICRP (1979b) of $2.9\ 10^5$ disintegrations in the thyroid per unit ingested activity. The slight discrepancy is because the ICRP used a value of 0.33 for the fraction transferred from blood to thyroid, instead of the value 0.3 that they quote in the description of their metabolic model (Greenhalgh et al. 1985). If a value of 0.33 is used in our algorithm, the calculated number of disintegrations agrees with that quoted by the ICRP.

Using an average β-particle energy of 0.19 MeV (ICRP 1983) and a thyroid mass of 20 g (ICRP 1979a), one can calculate the β dose equivalent to the thyroid as 4.1×10^{-7} Sv.