ORNL/TM-7379

MASTER

OAK
RIDGE
NATIONAL
LABORATORY

UNION
CARBIDE

Calculation of Probabilities of Transfer, Recurrence Intervals, and Positional Indices for Linear Compartment Models

J. H. Carney
D. L. DeAngelis
R. H. Gardner
J. B. Mankin
W. M. Post

ENVIRONMENTAL SCIENCES DIVISION
Publication No. 1544

OPERATED BY
UNION CARBIDE CORPORATION
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

DISCLAIMER

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A05; Microfiche A01

ORNL/TM-7379

Contract No. W-7405-eng-26

CALCULATION OF PROBABILITIES OF TRANSFER, RECURRENCE INTERVALS, AND POSITIONAL INDICES FOR LINEAR COMPARTMENT MODELS[1]

J. H. Carney,[2] D. L. DeAngelis, R. H. Gardner,
J. B. Mankin,[2] and W. M. Post

ENVIRONMENTAL SCIENCES DIVISION
Publication No. 1544

[1]Research sponsored by the National Science Foundation's Ecosystem Studies Program under Interagency Agreement DEB 77-25781.

[2]Computer Sciences Division, Union Carbide Corporation-Nuclear Division.

Date Published: February 1981

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
DEPARTMENT OF ENERGY

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

ABSTRACT

J. H. CARNEY, D. L. DEANGELIS, R. H. GARDNER, J. B. MANKIN, and W. M. POST. 1981. Calculation of probabilities of transfer, recurrence intervals, and positional indices for linear compartment models. ORNL/TM-7379. Oak Ridge National Laboratory, Oak Ridge, Tennessee. 76 pp.

Six indices are presented for linear compartment systems that quantify the probable pathways of matter or energy transfer, the likelihood of recurrence if the model contains feedback loops, and the number of steps (transfers) through the system. General examples are used to illustrate how these indices can simplify the comparison of complex systems or organisms in unrelated systems.

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

TABLE OF CONTENTS

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

LIST OF TABLES

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

LIST OF FIGURES

THIS PAGE

WAS INTENTIONALLY

LEFT BLANK

X

INTRODUCTION

When considering problems involving the flow of nutrients, toxic chemicals, or radionuclides through an ecosystem, one is often interested in the likelihood of a molecule of nutrient or toxic chemical starting at one part of the ecosystem and reaching another part, on the average time it will take to get there, and the number of ecosystem components it will pass on its journey. These problems can be solved (at least partially) by means of compartmental models. In such models, the pathways from one compartment to another are made explicit and the rates of transfer are specified. The task of solving the above problems becomes one of proper model conceptualization (e.g., adequate representation of the real system) and computation of numerical indices that reflect the passage of material (or energy) through the system. Such a capability provides the modeler with a means of comparing systems whose rates of transfers and interconnections differ.

Although there have been quantitative discussions of recycling (Harte and Morowitz 1975, Wise et al. 1968), connectedness of ecosystems (MacArthur 1955), and measurement of trophic distances (Gallopin 1972, Tansky 1976, Kercher and Shugart 1976), we are unaware of a useful computational form. In this report, accordingly, we develop some indices, provide examples of their use, and present an interactive computer program for calculating these values for donor-controlled linear compartment models.

Throughout this report, the definitions for all indices are based on a unit of either matter or energy. For computational consistency and convenience, all models are augmented by a compartment representing the forcing functions. This makes possible the calculation of values for models where more than one compartment is forced. Hence, in those indices where the number of steps from compartment 1 to compartment N is calculated, e.g., Index 3, ID, the first compartment refers to the source of material or energy.

Two basic quantities can be used to define the response of the system to external influences. The first is the probability, p_{ij}, that a given unit (of matter or energy), residing initially in compartment j, will next be located in compartment i. It is clear that, since the unit must eventually either stay in the j^{th} compartment, move to one of the other compartments of the system, or else leave the system, then

$$\sum_{i=1}^{n} p_{ij} + p_{n+1,j} = 1 \quad , \tag{1}$$

where n is the number of compartments in the system and $p_{n+1,j}$ is the probability of loss from the system. The second basic quantity is t_{ij}, the average time it takes the unit to arrive in compartment i, measured from the time it entered compartment j.

In a fully described system, the values of p_{ij} and t_{ij} for all transfers between the connected compartments are specified, as well as the inputs to the system. This is illustrated in Fig. 1.

A common way of representing the flow of mass or energy through a system is by means of a set of differential equations. In such a set

ORNL-DWG 79-17813

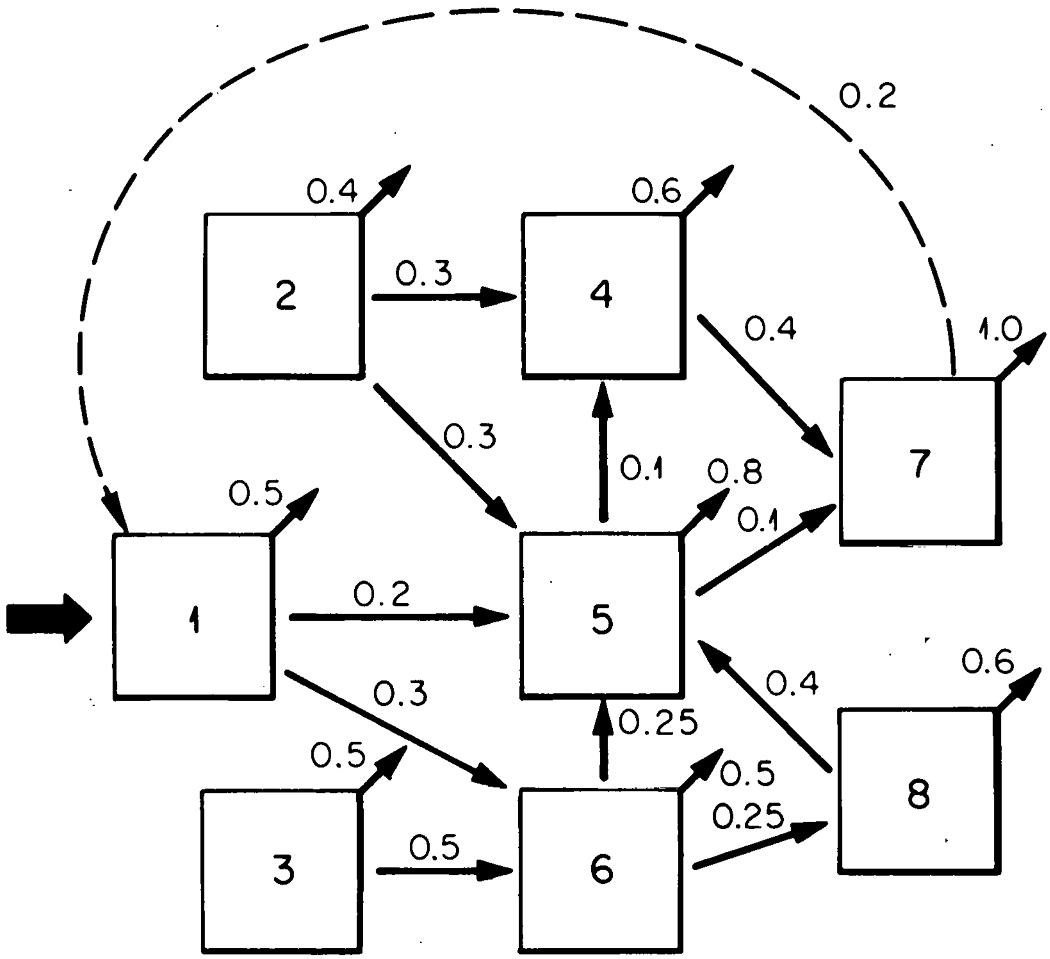

Fig. 1. A food web model illustrating indices 1, 2, and 3. The forcing function is indicated by the bold arrow. The upward slanting arrows indicate losses from the system, and the decimal quantities are the transition probabilities.

of equations, for example, the rate of change of mass or energy in the i^{th} compartment would be given by

$$\frac{dx_i}{dt} = u_i + \sum_{\substack{k=1 \\ k \neq i}}^{n} a_{ik}x_k - \sum_{\substack{k=1 \\ k \neq i}}^{n} a_{ki}x_i - r_i x_i \quad , \tag{2}$$

where x_i is the mass (or energy) in compartment i, a_{ik} is the transfer rate from compartment k to compartment i, u_i is the forcing function into compartment i, and r_i is the rate of loss from compartment i to the external world. The first summation in Eq. (2) represents inputs from other compartments to compartment i, while the second summation represents outputs from compartment i to all other compartments. For each of the n compartments in the system, there is an equation of the form of Eq. (2).

The differential equation representation can be reinterpreted in terms of the p_{ij}'s and t_{ij}'s mentioned above. The probability, p_{ij}, that a given unit initially in compartment j will next reside in compartment i is simply

$$p_{ij} = \frac{a_{ij}}{r_j + \sum_{\substack{i=1 \\ i \neq j}}^{n} a_{ij}} = \frac{a_{ij}}{-a_{jj}} \quad , \tag{3}$$

while the average residence time of the unit in compartment j before moving to a new compartment is approximately

$$\tau_j \simeq \frac{1}{-a_{jj}} \quad , \tag{4}$$

The compartmental model defined by Eq. (2) has the property that if the fluxes from compartment j to compartments i and k are the same (i.e., $a_{ij} = a_{kj}$), then the average time it takes for a unit that is originally in compartment j and then moves to compartment k is the same as the average time it takes for a unit originally in compartment j to move to compartment k. In this sense, the compartmental model defined by Eq. (2) is slightly less general than the system described in terms of the p_{ij}'s and t_{ij}'s. In the latter system, the time it takes for a unit to get from one compartment to another, t_{ij}, can be prescribed independently of the fluxes between these compartments. One can imagine there being pipelines between the compartments, in which units may spend varying amounts of time in transit.

Below we describe our proposed indices (Table 1) for the compartment system. In the computer calculations of the indices, matrix methods are used but, for simplicity, these will be described in the Appendix A.

Table 1. Positional indices

Index Number	Acronym	Purpose[a]
1	PPTC	Calculates the probability of a unit reaching a compartment.
2	ENPTC	Calculates the expected number of passes of a unit through a compartment.
3	TD	Calculates trophic distance.
4	MTT	Calculates the mean time of transfer of a unit from compartment 1 to N.
5	MRT	Calculates the average time it takes for a unit to recur in a given compartment if there are direct or indirect feedback loops to that compartment.
6	ETP	Calculates the trophic position as given by Kercher and Shugart (1975).

[a]A unit refers to any arbitrary quantity of matter or energy (e.g., a molecule or a calorie).

DESCRIPTION OF INDICES

Index 1: Probability of a Unit Passing Through a Compartment (PPTC)

This index calculates, for each compartment, the probability that a unit of mass or energy, starting at compartment 1 in the system, will reach some compartment N at least once.

Consider the hypothetical compartment model in Fig. 1. There are eight compartments, which might perhaps be eight species in a food web. The decimal numbers on the arrows between compartments are the probabilities of transfer, p_{ij}. For example, the probability that a unit of mass or energy, given that it has arrived in compartment 6, moves next to compartment 5, is $p_{56} = 0.25$. The values of the transfer times between compartments, t_{ij}, are irrelevant to the present index.

The possible paths between compartments 1 and 7, along with the path lengths and probabilities of occurrence, are given in Table 2.

Table 2. Possible paths from Compartment 1 to 7 for the food web illustrated in Fig. 1

Path	Length of path	Probability of following path
1→5→7	2.0	(0.20)(0.10) = 0.02
1→5→4→7	3.0	(0.20)(0.10)(0.40) = 0.008
1→6→5→7	3.0	(0.30)(0.25)(0.10) = 0.0075
1→6→5→4→7	4.0	(0.30)(0.25)(0.10)(0.40) = 0.003
1→6→8→5→7	4.0	(0.30)(0.25)(0.40)(0.10) = 0.003
1→6→8→5→4→7	5.0	(0.30)(0.25)(0.40)(0.10)(0.4) = 0.0012

The probability that the unit starting at compartment 1 ever arrives at compartment 7 is,

$$\begin{aligned} PPTC(1\rightarrow 7) &= p_{75}p_{51} + p_{74}p_{45}p_{51} + p_{75}p_{56}p_{61} + p_{74}p_{45}p_{56}p_{61} \\ &\quad + p_{75}p_{58}p_{86}p_{61} + p_{74}p_{45}p_{58}p_{86}p_{61} \\ &= 0.0427 \end{aligned} \qquad (5)$$

The general algorithm for calculating PPTC(1 N) is

$$PPTC(1\rightarrow N) = p_{N1} + \sum_{i=1}^{n} p_{Ni}p_{i1} + \sum_{i=1}^{n}\sum_{j=1}^{n} p_{Ni}p_{ij}p_{j1} + \cdots, \qquad (6)$$

where the index N of the final compartment is permitted to occur only once in each term. This restriction reflects the index's representing only the probability of first passage through compartment N. It can be shown that PPTC(1→N) is always less than or equal to 1.0.

Index 2: Expected Number of Passes of a Unit Through a Compartment (ENPTC)

Suppose now that a feedback loop is introduced into the system pictured in Fig. 1 (see the dotted line connecting compartment 7 to compartment 1). The possibility exists for a given unit to make repeated transits through compartment 7. The expected number of passes a unit will make through compartment 7 before leaving the system can be written as the infinite series,

$$\begin{aligned} \text{ENPTC}(1\rightarrow7) &= \text{PPTC}(1\rightarrow7) + \text{PPTC}(1\rightarrow7)\ \text{PPTC}(7\rightarrow7) \\ &\quad + \text{PPTC}(1\rightarrow7)\ \{\text{PPTC}(7\ 7)\}^2 + \ldots \\ &= 0.0427 + 0.2(0.0427)^2 + (0.2)^2(0.0427)^3 + \ldots \\ &\simeq 0.0455 \end{aligned} \tag{7}$$

The first term [PPTC(1→7)] is the probability that the unit reaches compartment 7 at least once, and the second term is the probability that the unit reaches compartment 7 at least twice. The remaining terms follow the same pattern.

The general algorithm for ENPTC(1→N) is

$$\begin{aligned} \text{ENPTC}(1\rightarrow N) &= \text{PPTC}(1\rightarrow N) + \text{PPTC}(1\rightarrow N)\ \text{PPTC}(N\rightarrow N) \\ &\quad + \text{PPTC}(1\rightarrow N)\{\text{PPTC}(N\rightarrow N)\}^2 + \ldots \end{aligned} \tag{8}$$

As an interesting sidelight, let us demonstrate a connection between the index ENPTC and the equilibrium values of a differential equation compartment model. Consider a very simple two-compartment model (Fig. 2) described by the equations

$$\frac{dx_1}{dt} = 1.0 - x_1 + 0.5x_2 \tag{9}$$

$$\frac{dx_2}{dt} = 0.5x_1 - x_2 \quad . \tag{10}$$

The steady-state values of x_1 and x_2 are $x_1 = 1.333...$ and $x_2 = 0.666...$. It can be shown that ENPTC(1→1) $\simeq$ 1.333 and ENPTC(1→2) $\simeq$ 0.666. Hence, if there is some rate of input, u_1, of

ORNL-DWG 79-17814

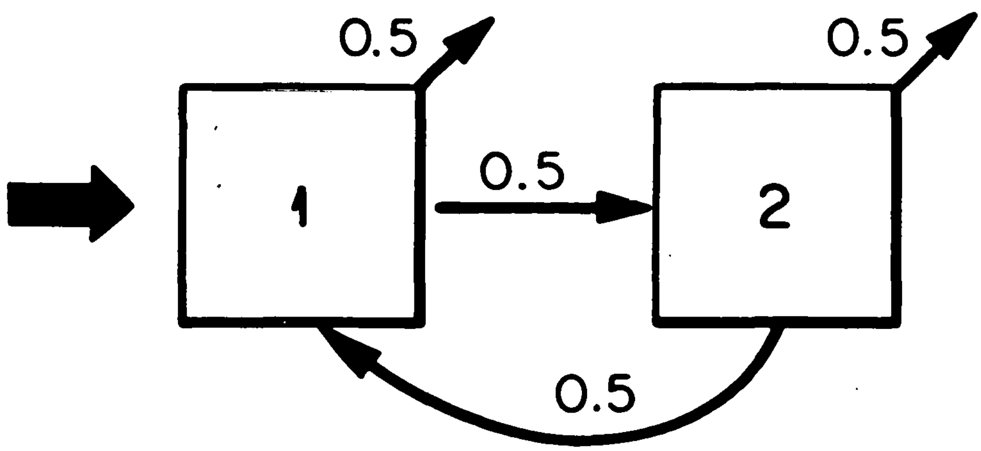

Fig. 2. A simple two-compartment model with the decimal quantities indicating the values of the transfer coefficients.

units into the system, then ENPTC(1→N) is the number of units in compartment N at steady state.

Note that the probability that a unit passes through compartment N exactly M times, RNT, is

$$\mathrm{RNT}(1\rightarrow N;M) = \mathrm{PPTC}(1\rightarrow N)\,\{\mathrm{PPTC}(N\rightarrow N)\}^{M-1}\,\{1 - \mathrm{PPTC}(N\rightarrow N)\}\ . \qquad (11)$$

In this expression, the factor 1 - PPTC(N→N) is the probability that the unit does not return to compartment N again after the Mth passage through.

Index 3: <u>Trophic Distance (TD)</u>

Consider again the hypothetical compartment model pictured in Fig. 1. By "trophic distance" we mean the average number of links, or transitions from compartment to compartment, that a unit makes in going from one compartment to another, possibly remote, compartment in a system. In particular, we are interested in the number of links between compartment 1 and some other compartment, N.

Basically, the trophic distance is found by summing overall possible probabilities of pathways between the compartments and multiplying the length of each particular path by the likelihood of its being followed. For example, for the system in Fig. 1, the average number of transitions needed to reach compartment 7 from compartment 1 is found by first summing the conditional probabilities (conditional meaning that the unit reaches compartment 7) of the unit taking each path between compartments 1 and 7 and multiplying by the lengths of the paths. The conditional probabilities are next divided by PPTC(1→7). The average number of links, or the trophic distance, TD(1→7), is then

$$\begin{aligned} TD(1\rightarrow 7) &= (2p_{75}p_{51} + 3p_{74}p_{45}p_{51} + 3p_{75}p_{56}p_{61} + 4p_{74}p_{45}p_{56}p_{61} \\ &\quad + 4p_{75}p_{58}p_{86}p_{61} + 5p_{74}p_{45}p_{58}p_{86}p_{61})/PPTC(1\rightarrow 7) \\ &= 0.1165/0.0427 \simeq 2.728 \quad . \end{aligned} \tag{12}$$

The general algorithm for calculating TD(1→N) is

$$TD(1\rightarrow N) = (p_{N1} + 2\sum_{i=1}^{n} p_{Ni}p_{i1} + 3\sum_{i=1}^{n}\sum_{j=1}^{n} p_{Ni}p_{ij}p_{j1} + \ldots)/PPTC(1\rightarrow N) \quad . \tag{13}$$

The effects of feedback on trophic distance are especially pronounced. Consider the food chain shown is Fig. 3. As the feedback (α) from compartment 5 to compartment 3 increases, the trophic distance between compartments 1 and 6 increases.

Index 4: <u>Mean Time of Transfer from Compartment 1 to N (MTT)</u>

This index calculates the average time required by a unit to travel from compartment 1 to compartment N. The transition probabilities, p_{ij}, associated with traveling each possible pathway from compartment 1 to compartment N, and the amount of time spent traveling along each of these segments, t_{ij}, determine this interval. For example, in the system pictured in Fig. 4, there is a 0.3 probability of the unit traveling from compartment 1 to compartment 3, and it takes 2 days to complete this path segment.

ORNL-DWG 79-17815

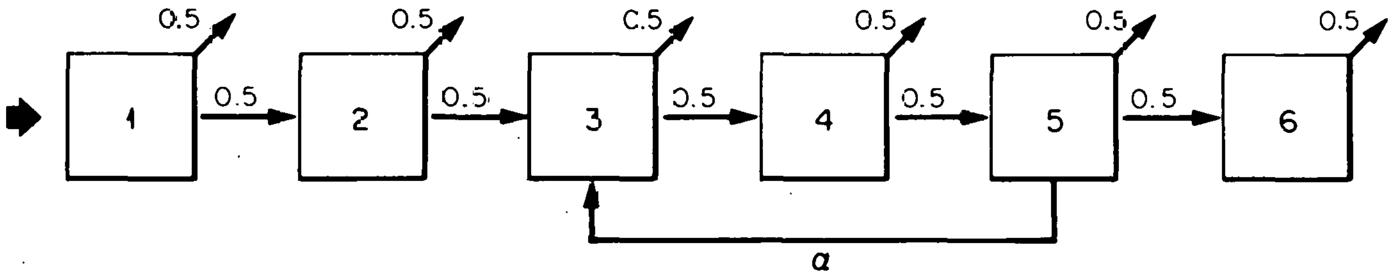

Fig. 3. Food chain model illustrating the behavior of trophic distance, TD (1 → N), as a function of the feedback coefficient, α. The forcing function, indicated by the bold arrow, is to compartment 1. The upward slanting arrows indicate losses from the system, and the decimal quantities above the arrows connecting compartments are the transition probabilities.

ORNL-DWG 79-17818R

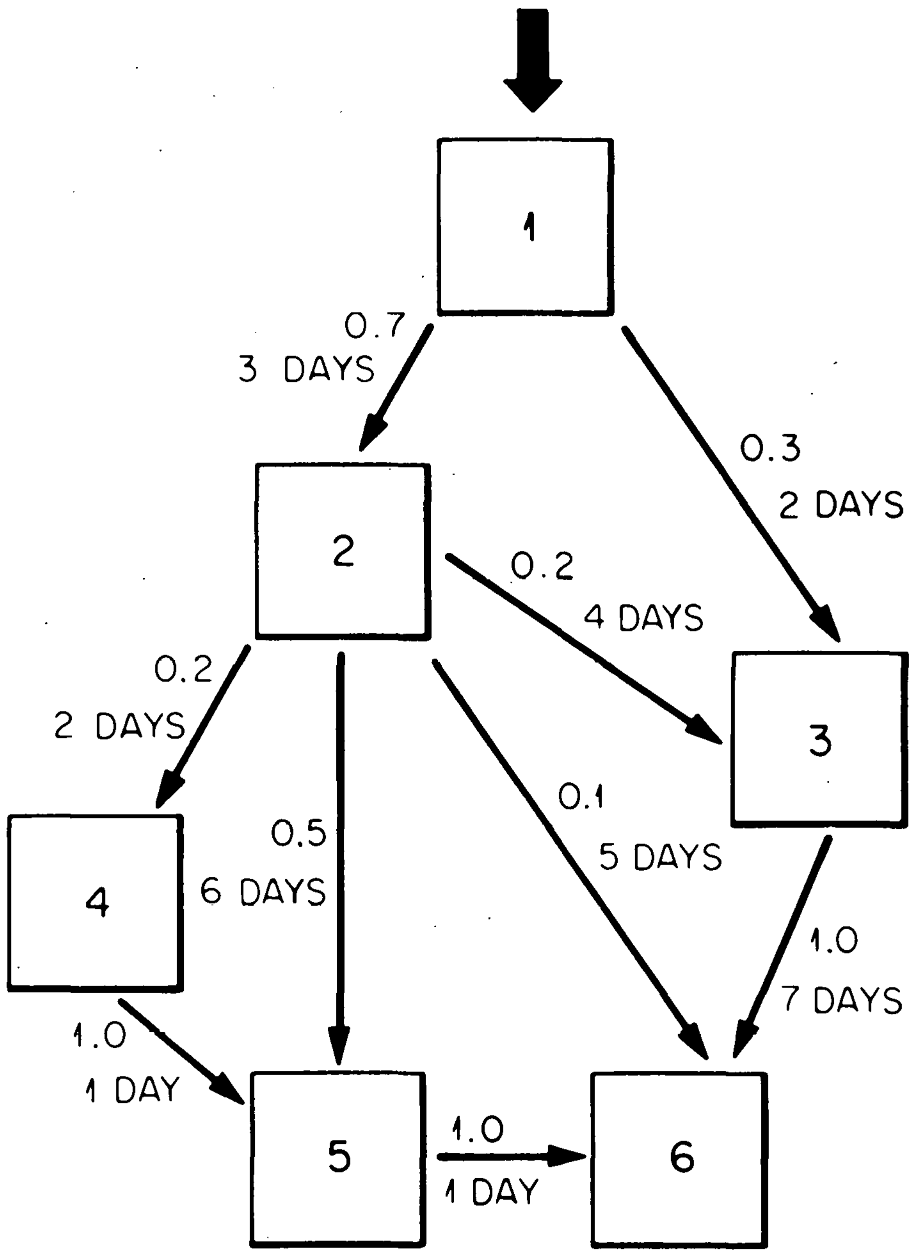

Fig. 4. A food web model illustrating the concept of the mean time of transfer, MTT ($1 \rightarrow N$), of a unit from compartment 1 to compartment N. The decimal quantities above the arrows connecting compartments are probabilities of a unit traveling on a given segment. The time in days needed to travel along each segment is also given.

The possible paths from compartment 1 to compartment 6, as well as the associated transfer times and probabilities, are shown in Table 3.

Table 3. Possible paths from compartment 1 to 6 for the food web illustrated in Fig. 4

Path	Length of path (days)	Probability of following path
1→2→6	8	(0.70)(0.10) = 0.07
1→3→6	9	(0.30)(1.0) = 0.30
1→2→3→6	14	(0.70)(0.20)(1.0) = 0.14
1→2→5→6	10	(0.70)(0.50)(1.0) = 0.35
1→2→4→5→6	10	(0.70)(0.20)(1.0)(1.0) = 0.14

The mean time of transfer, MTT(1→6), from compartment 1 to compartment 6 is,

$$\begin{aligned} MTT(1\to 6) &= (8p_{21}p_{62} + 9p_{31}p_{63} + 14p_{21}p_{32}p_{63} + 10p_{21}p_{52}p_{65} \\ &\quad + 10p_{21}p_{42}p_{54}p_{65})/1.0 \\ &= 8(0.70)(0.10) + 9(0.30)(1.0) + 14(0.70)(0.20)(1.0) \\ &\quad + 10(0.70)(0.50)(1.0) + 10(0.70)(0.20)(1.0)(1.0) \\ &= 10.12 \text{ days} \end{aligned} \tag{14}$$

The general algorithm for MTT(1 N) is

$$\text{MTT}(1\to N) = \Big\{t_{N1}p_{N1} + \sum_{i=1}^{n} p_{Ni}p_{i1}\,(t_{i1} + t_{Ni}) + \sum_{i=1}^{n}\sum_{j=1}^{n} p_{Ni}p_{ij}p_{j1}\,(t_{Ni} + t_{ij} + t_{j1}) + \ldots\Big\}/\text{PPTC}(1\to N)\ , \tag{15}$$

where N is the total number of compartments in the system.

It is possible to show that MTT(1→N) is often positively correlated with TR, the time of return to equilibrium following a perturbation, for a food chain model. Let us show this for a simple four-species chain described by the equations

$$\dot{x}_1 = x_1\,(u_1 - a_{12}x_2 - g_1x_1) \tag{16}$$

$$\dot{x}_2 = x_2\,(\gamma a_{12}x_1 - a_{23}x_3 - g_2x_2) \tag{17}$$

$$\dot{x}_3 = x_3\,(\gamma a_{23}x_2 - a_{34}x_4 - g_3x_3) \tag{18}$$

$$\dot{x}_4 = x_4\,(\gamma a_{34}x_3 - g_4x_4) \tag{19}$$

where u_1 is the input into the lowest (autotroph) compartment, the γ's are the efficiencies, and the g_i's are density-dependent loss rates. Although these are nonlinear equations, we can apply our indices to the set up equations linearized about the equilibrium point of the system. We wish to evaluate TR and MTT in the vicinity of the equilibrium point, at which $\dot{x}_1 = \dot{x}_2 = \dot{x}_3 = \dot{x}_4 = 0$; the equilibrium point is $(x_1^0, x_2^0, x_3^0, x_4^0)$, where

$$x_4^0 = \frac{u_1}{a_{12}A + g_1B}$$

$$x_3^0 = \frac{g_4}{\gamma a_{34}} x_4^0$$

$$x_2^0 = \frac{a_{34}x_4^0 + g_3x_3^0}{\gamma a_{23}}$$

$$x_1^0 = \frac{a_{23}x_3^0 + g_2x_2^0}{\gamma a_{12}}$$

where

$$A = \frac{1}{\gamma a_{23}} \left(a_{34} + \frac{g_3g_4}{a_{34}}\right)$$

$$B = \frac{g_1}{\gamma a_{12}} \left(\frac{a_{23}g_4}{a_{34}} + g_2A\right) .$$

This demonstrates that x_1^0, x_2^0. x_3^0, and x_4^0 vary linearly with u_1. Next we look at the stability matrix, $\underline{\underline{M}}$, associated with Eqs. (16), (17), (18), and (19)

$$\underline{\underline{M}} = \begin{bmatrix} -g_1x_1^0 & a_{12}x_1^0 & 0 & 0 \\ \gamma a_{12}x_2^0 & -g_2x_2^0 & -a_{23}x_2^0 & 0 \\ 0 & \gamma a_{23}x_3^0 & -g_3x_3^0 & -a_{34}x_3^0 \\ 0 & 0 & \gamma a_{34}x_4^0 & -g_4x_4^0 \end{bmatrix} . \quad (20)$$

The eigenvalue equation is

$$\underline{\underline{M}} \cdot \underline{y} = \lambda \underline{y} \quad . \tag{21}$$

Since all the elements in $\underline{\underline{M}}$ are proportional to u_1, then we can write

$$\underline{\underline{M}} = u_1 \, \underline{\underline{M}}' \tag{22}$$

where $\underline{\underline{M}}'$ is independent of u_1, so that Eq. (21) becomes

$$u_1 \, \underline{\underline{M}}' \, \underline{y} = \lambda \underline{y} \quad . \tag{23}$$

Hence λ is proportional to u_1.

What we can conclude from the above analysis is that, if all the real parts of the eigenvalues of M' are negative, then an increase in u_1 will make them more negative. Since the return time to equilibrium, TR, of a stable system is proportional to the smallest value of -1/Real (λ_i), where λ_i is the i^{th} eigenvalue, TR decreases with increasing u_1. That is, increasing u_1 will cause the system to return to equilibrium more quickly following a perturbation.

Note from the expressions for x_i^0 (i = 1,4) that as u_1 increases, the transfer constants $a_{12}x_2^0$, $a_{23}x_2^0$, and $a_{34}x_3^0$ also increase. Therefore, the transfer times t_{ij} decrease as u_1 increases, and MTT(1→N) also decreases. Therefore, MTT(1→N) is positively correlated with TR in this case, since both quantities decrease as u_1 increases and vice versa.

Index 5: Mean Recurrence Time (MRT)

This index applies only to models that contain feedbacks. The index calculates the amount of time required for a unit that was once in compartment N to return to compartment N.

If there is feedback in the system, it is possible that a molecule starting from compartment N will eventually return to the same compartment. For example, Fig. 5 illustrates a variation of Fig. 4 in which there is a probability of feedback from compartment 6 to compartment 2. Hence, a unit leaving compartments 2, 3, 4, 5, or 6 has a nonzero probability of returning to those compartments. We assume that at each time step the unit has a probability equal to 1.0 of leaving the compartment in which it currently resides.

A formula for MRT(N) can be found by adapting the formula for MTT(1→N). We obtain

$$\mathrm{MRT}(N) = \sum_{i=1}^{n} a_{Ni} a_{iN} (t_{Ni} + t_{iN}) + \sum_{i=1}^{n} \sum_{j=1}^{n} a_{Ni} a_{ij} a_{jN} (t_{Ni} + t_{ij} + t_{jN}) + \ldots / \mathrm{PPTC}(N \rightarrow N) \quad , \quad (24)$$

where N is the total number of compartments in the system.

Index 6: Effective Trophic Position (ETP)

The index of effective trophic position (ETP) has been defined in Kercher and Shugart (1975) for lower triangular models only (Table 4). The method presented here has been generalized to consider any linear compartment model without feedbacks. Only a summary of the concept of this index will be presented here (see Appendix A for the algorithm).

ORNL-DWG 79-17818

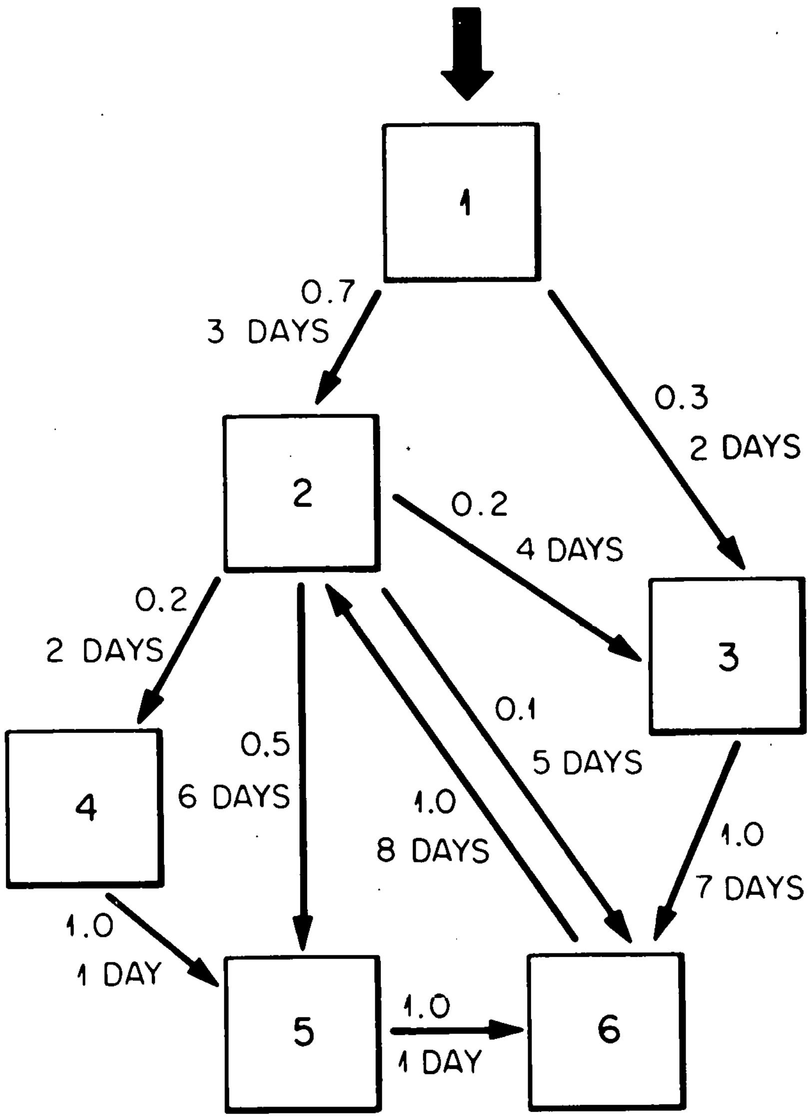

Fig. 5. A food web model illustrating the concept of the mean recurrence time, MRT(N), associated with compartment N. This model is similar to Figure 5, except for an additional transfer coefficient from compartment 6 to compartment 2.

Table 4. Values for Index 6 (Effective Trophic Position) for three lower triangular models[a]

Model	Compartment	$Flux_i$	G_i	Z
1	2	1	1	1
	3	0.1	1	2
	4	0.01	1	3
2	2	1	1	1
	3	0.0909	0.5	1.74
	4	0.1	1	2
3	2	1	1	1
	3	0.0909	0.5	1.74
	4	0.0091	0.5	2.74
	5	0.0909	0.5	1.74
	6	0.0091	0.5	2.74

[a]In all the models compartment 1 acts as a dummy compartment that is forced (no respiration losses and transfer from 1→2 that is 1.0). The transfer rates between all other compartments are 0.1 and respiration rates for all compartments are 0.9. Model 1 is a straight chain from 2→3→4. Model 2 is the same as Model 1 except there is an additional transfer from 2→4. Model 3 consists of 2 chains originating from compartment 2. One follows the path 2→3→4. The other follows the path 2→5→6.

ETP of a food web member is defined as a function of the net energy input (e.g., cal t^{-1}) to the population. Consider, for example, a simple food chain at steady state where each member of the food chain has the same transfer efficiency, a (net production/ingestion). If the input entering the herbivore from the autotroph is G then aG is delivered to the first carnivore, a^2G to the second carnivore, etc. The flux entering the k^{th} member is

$$Flux_k = a^{k-1}G = Ge^{(k-1)\ ln\ (a)}$$

Solving for trophic position k

$$k = 1 + \frac{\ln(\text{Flux}_k/G)}{\ln(a)} \tag{25}$$

The calculations for ETP are accomplished in a similar, though more complicated, fashion. Assume a flux matrix has been calculated where

$$\text{Flux} = \left[F_{ij}\right]_{N \times N}$$

F_{ij} is the energy transfer from j to i in unit time at steady state. Compartment 1 is an energy source. The proportion of energy entering the system going to the maintainance of consumer i (G_i) is

$$G_i = \sum_{j=2}^{N} \frac{F_{ij}G_i}{\sum_{k=1}^{N} F_{kj}} + F_{i1}$$

and

$$G_2 = F_{21}$$

Since the total flux entering consumer i is $\sum_{k=1}^{N} F_{ik}$, then $\sum_{k=1}^{N} F_{ik}/G_i$ is the flux entering the system devoted to the maintainance of i. As in Eq. (25) we can calculate ETP of consumer i as

$$Z_i = 1 + \frac{1}{\ln \bar{a}} \ln \left[\frac{\sum_{k=1}^{N} F_{ik}}{G_i}\right]$$

where $\bar{a}$ is the average transfer efficiency of the entire food web.

UTILITY OF INDICES

Uncertainties associated with the modeling process make it difficult to compare the behavior of ecosystems. If several ecosystem models predict a set of reasonably similar values, then the choice of a specific model will rely on a variety of factors which include the objectives of the model, the physical description of the mathematics, and confidence in the robustness of the model (Mankin et al. 1977). Models with extensive feedbacks (e.g., nutrient cycling) are sensitive to the explicit value of the transfers describing the cycle; and, because these models are condensations of reality, subtle differences in model conceptualization can produce critical differences between predictions (O'Neill and Gardner 1979, Carney et al. 1979). We believe that the indices presented here can be an important tool for comparing models.

Models which behave in a reasonably similar manner can be analytically compared and compartment specific values for several indices verified in the field. For instance, four alternate models of a calcium model were formed that predicted identical values of calcium at steady state. However, the nature of the compartments and connectances within a model differed. Values of PPTC and ENPTC were conserved in all models, but the mean transit time and the total time of passage differed significantly between models. If this value is used as a measure of resilience (response to perturbations), then there exists significant differences between models in spite of their overall similarity. The critical differences identified by this analysis allows a method of field validation of the models (Carney et al. 1979). Once the model has been selected, then comparisons between ecosystems are possible.

These indices can also provide some theoretical insight into the general behavior of model systems. An initial effort along this line (DeAngelis et al. 1978) has shown that transit time is a measure of response of the system to perturbation. In food chain models this index (MIT) also indicates that longer food chains can respond as fast as shorter food chains to perturbations if the energy flux is high enough. Other indices hold similar promise of insight into system dynamics.

REFERENCES

Carney, J. H., R. H. Gardner, J. B. Mankin, and D. L. DeAngelis. 1979. Cycling indices for ecosystem models, Summer Computer Simulation Conference 1979. Proceedings, pp. 389-393. Toronto, Ontario, Canada.

DeAngelis, D. L., R. H. Gardner, J. B. Mankin, W. M. Post, and J. H. Carney. 1978. Energy flow and the number of trophic levels in ecological communities. Nature 273:406-407.

Gallopin, G. C. 1972. Structural properties of food webs. pp. 241-282. IN B. C. Patten (ed.), Systems Analysis and Simulation in Ecology, Vol. II. Academic Press, New York.

Harte, J., and H. Morowitz. 1975. Nutrient transit time diversity: A novel measure of ecological organization and stability. Lawrence Berkeley Laboratory publication No. 4441, 8 p.

Kercher, J. R., and H. H. Shugart, Jr. 1975. Trophic structure, effective trophic position, and connectivity in food webs. Am. Nat. 109:191-206.

MacArthur, R. H. 1955. Fluctuations of animal populations, and a measure of community stability. Ecology 36:533-536.

Mankin, J. B., R. V. O'Neill, H. H. Shugart, and B. W. Rust. 1975. The importance of validation in ecosystem analysis. IN G. S. Innis (ed.), New Directions in the Analysis of Ecological Systems, Part 1. Simulation Councils Proceedings Series 15(1):63-71. Simulation Councils, Inc., LaJolla, California.

O'Neill, R. V., and R. H. Gardner. 1979. Sources of uncertainty in ecological models, pp. 447-463. IN B. P. Zeigler, M. S. Elzas, G. J. Kliv, and T. I. Oren (eds.), Simulation Models and Methodology, North Holland Press.

Tansky, M. 1976. Structure, stability and efficiency of ecosystems. Prog. Theor. Biol. 4:205-262.

Westley, G. W., and J. A. Watts. 1970. The Computing Center Numerical Analysis Library. CTC-39. Oak Ridge National Laboratory, Oak Ridge, Tennessee.

Wise, M. E., S. B. Osborn, J. Anderson, and R. W. S. Tomlinson. 1968. A stochastic model for turnover of radiocalcium based on observed power laws. Math. Biosci. 2:199-224.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

APPENDIX A
COMPUTATIONAL ALGORITHMS

I. GENERAL ALGORITHM

This section presents a general algorithm that is used in all indices except Index 6. We consider systems that may be represented by a system of equations. Our concern here is in transition probabilities through the system when a unit is input at the source. Therefore, consider the system of equations

$$Ax + u = 0 \quad , \tag{A-1}$$

where x is the n-dimensional state vector, A is the nxn coefficient matrix, and u is the n-dimensional vector of forcing functions. If D is a diagonal matrix whose elements are given by

$$d_{ii} = 1/\sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ji} \quad ,$$

we can then write

$$Py + u = 0 \quad ,$$

where

$$x = Dy$$

and

$$P = AD \quad .$$

If the unit initially enters from the source, we can consider the initial vector y_0 as given by

$$I\ y_0 + u = 0 \quad ,$$

or

$$y_0 = -\ u \quad .$$

If the forcing vector u is restricted such that

$$\sum_{i=1}^{n} \left| u_i \right| = 1 \quad ,$$

the vector y can be considered as the probability state vector and the matrix P as the probability transition matrix. Therefore, we represent the progress of the unit through the system by the Markov process

$$y_{k+1} = P y_k \quad . \qquad \text{(A-3)}$$

This gives the sequence of probability state vectors

$$\begin{aligned}
y_1 &= P y_0 \\
y_2 &= P y_1 = P^2 y_0 \\
y_3 &= P y_2 = P^3 y_0 \\
&\vdots \qquad \vdots \qquad \vdots \\
y_m &= P y_{m-1} = P^m y_0 \\
&\vdots \qquad \vdots \qquad \vdots
\end{aligned}$$

At step m, the state will be determined by m transformations of the probability matrix, P, on the initial state, y_0. If we restrict ourselves to unit inputs, say to compartment 1, at step m

$$(y_m)_j = \sum_{i=1}^{n} P_{j_1}{}^{m} (y_0)_1 \tag{A-4}$$

$$= P_{j_1}^{m} \quad .$$

Therefore, the probability state vector will be identical to the first column of P^m. This eliminates the necessity of calculating the probability state vector. Indices 1 through 5 make use of this algorithm with suitable modifications.

II. INDEX 1: PPTC

Index 1 is the probability that a unit will pass through a compartment when it enters the system at the source. In computing this index, we need only the transfer probabilities, so the diagonal elements of the P matrix are set to zero and the contents of each compartment is transferred out at each step. For the moment, assume the system contains no loops. Then the probability that the unit will pass through compartment j ($j \neq 1$) is

$$I_j = \sum_{i=1}^{m} (y_i)_j = \sum_{i=1}^{m} (P^i)_{j1} \quad , \tag{A-5}$$

where m is that step in which the first column of P^m is a null vector.

We are interested only in the probability that a unit will pass through compartment j. Once it has entered compartment j, it is of no further interest. This poses no problem unless the system contains loops. The probability of a unit passing through compartment j may be affected by loops that do not include compartment j, but it is not affected by loops that include compartment j. Therefore, if we set all transfers out of compartment j equal to zero at each time step (i.e., the j^{th} column is a null vector), Eq. (A-5) becomes

$$I_j^{(1)} = \sum_{i=1}^{m} (P^i)_{j1} \quad ,$$

$$(P^i)_{kj} = 0, \quad k = 1,2\ldots,n \quad ,$$

where $I_j^{(1)}$ is the value of Index 1 for compartment j.

III. INDEX 2: ENPTC

Computationally, Index 2 is calculated identically as Index 1 except for the inclusion of feedback loops. Thus the only adjustment needed to the Algorithm for Index 1 is the deletion of the requirements that no transfers be made out of compartment j.

Since an infinite number of passes are now possible through compartment j, a stopping criteria is needed. We suggest a reasonable limit to be PPTC $(j \rightarrow j) < 0.0001$. It should be noted that for all indices that include loops this criteria will be applied.

IV. INDEX 3: TD

This index is the expected number of steps that a given unit will take in the transition from source to compartment j. We take the viewpoint that the unit is already in compartment j, and we want to know the number of steps taken in its transition from the source. This perspective allows us to simplify the calculations by ignoring losses. Also, if compartment j is within a loop, it may recirculate. Since we are interested in the expected number of steps, the feedback loop is not broken in this case. The expected number of steps from the source to compartment j is

$$I_j^{(3)} = \sum_{i=1}^{m} i(P^i)_{j1} / \sum_{i=1}^{m} (P^i)_{j1} \qquad \text{(A-7)}$$

where $I_j^{(3)}$ is the value of Index 3 for compartment j.

V. INDEX 4: MTT

To accomplish the proper form of Index 4 with the p's as products and the t's as sums, independent transformations on the P (probability) and T (time) matrices were needed.

The P matrices are computed as in the Index 1 algorithm with all transfers out of compartment j set to 0. That is, no feedback loops to compartment j exist.

The T matrices were computed for each step m so that

$$T^m = T^o(I) * T^{m-1} + T^o * T^{m-1}(I)$$

where

T(I) indicates all elements that are not null elements are replaced by 1. The first term on the right-hand side extracts the transit times from previous time steps. The second term extracts the transit times for the present time step.

Combining the P and T matrices, the computational form for Index 4 is

$$I_j^{(4)} = \frac{\sum_{i=1}^{m} (P^i)_{j1} * (T^i)_{j1}}{\sum_{i=1}^{m} (P^i)_{j1}}$$

In some cases, several pathways of equal lengths are generated. The resulting probabilities are stored separately so that multiplication by the correct transit times occurs. Due to this storage allocation, the upper bound on the number of equal pathways stored is 100.

VI. INDEX 5: MRT

Index 5 is formulated by adapting the algorithm of Index 4 so that probabilities and times are summed for ($j \rightarrow j$) elements rather than ($1 \rightarrow j$). Hence,

$$I_j^{(5)} = \frac{\sum_{i=1}^{m} (P^i)_{jj} * (T^i)_{jj}}{\sum_{i=1}^{m} (P^i)_{jj}}$$

VII. INDEX 6: ETP

The steps involved in the algorithm for Index 6 are listed below.

(1) Input the matrix A of transfer coefficients where compartment 1 is the only one that is forced. Compartment 1 is a dummy compartment with all $a_{i1} = 1$, and $a_{11} = -\sum_i a_{i1}$.

Forcing vector is then represented by

$$F = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

(2) Find equilibrium values for all compartments x_i

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ x_n \end{bmatrix} - \begin{bmatrix} -a_{11} & a_{12} & \cdot & \cdot & a_{1n} \\ a_{21} & -a_{22} & \cdot & \cdot & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & & & & \cdot \\ a_{n1} & \cdot & \cdot & \cdot & -a_{nn} \end{bmatrix}^{-1} \begin{bmatrix} -1 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix} = -A^{-1}F$$

(3) Make X into a diagonal matrix

$$X = \begin{bmatrix} x_1 & & & 0 \\ & x_2 & & \\ & & \ddots & \\ 0 & & & x_n \end{bmatrix}$$

and multiply by A matrix to form flux matrix FLUX

$$FLUX = AX = \begin{bmatrix} -a_{11}x_1 & a_{12}x_2 & \cdots & a_{1n}x_n \\ a_{21}x_1 & \cdot & & \cdot \\ \cdot & & & \cdot \\ a_{n1}x_1 & \cdots & \cdots & -a_{nn}x_n \end{bmatrix}$$

Let $F_{ij} = a_{ij}x_j$

$$FLUX = \begin{bmatrix} F_{11} & F_{12} & \cdots & F_{1n} \\ \cdot & & & \cdot \\ \cdot & & & \cdot \\ F_{n1} & \cdots & \cdots & F_{nn} \end{bmatrix}$$

(4) The computation of trophic position is calculated from the flux matrix. The computation is presented in four steps.

(a) Calculate G_i i=2,N

G_i is the proportion of input into the system that goes to the maintenance of compartment i.

$$G_i = F_{i1} + \sum_j \frac{F_{ij}F_{j1}}{\sum\limits_{\substack{n \\ n\neq j}} F_{nj}} + \sum_j \sum_k \frac{F_{ij}F_{jk}F_{k1}}{\sum\limits_{\substack{n \\ n\neq j}} F_{nj} \sum\limits_{\substack{n \\ n\neq k}} F_{nk}}$$

$$\sum_j \sum_k \sum_m \frac{F_{ij}F_{jk}F_{km}F_{m1}}{\sum\limits_{\substack{n \\ n\neq j}} F_{nj} \sum\limits_{\substack{n \\ n\neq k}} F_{nk} \sum\limits_{\substack{n \\ n\neq m}} F_{nm}} + \cdot \cdot \cdot \cdot$$

(b) Calculate $ZFLUX_i$. This is the total direct flux into compartment i.

$$ZFLUX_i = \sum_{\substack{j \\ i\neq j}} F_{ij}$$

(c) Calculate $\bar{a}$

$$a = \frac{1}{n} \sum_i \sum_{\substack{j\neq 1 \\ a_{ij}\neq 0 \\ j\neq 1}} a_{ij}$$

where n = number of non-zero a_{ij}.

(d) Calculate trophic position of compartment i, z_i

$$z_i = \frac{1}{\log \bar{a}} \log \left[\frac{ZFLUX_i}{G_i}\right] + 1$$

Note: This index will not allow feedback loops. G_i will not converge with loops since $F_{ij} > 1$.

THIS PAGE
WAS INTENTIONALLY
LEFT BLANK

APPENDIX B

PROGRAM DOCUMENTATION

POSIT is an interactive FORTRAN Computer Program used to calculate six indices that describe aspects of the flow of matter or energy in linear compartment models. POSIT provides the user with simple input and convenient output for the analysis of linear compartmental models.

POSIT is modularly structured with each index calculated in separate subroutines. Information and work space is passed through common blocks. The main program is a driver for input of model parameters and output of specified position calculations. MAIN also provides options for editing data, calculating steady state values, and creating new data files. Table B-1 lists POSIT variables and their purposes.

Execution of POSIT

Execution of this program on the ORNL PDP-KL10 may be effected by typing:

.EX POSIT.F4 [ppn]

where (ppn) is the project programer number (DEC manual reference). For approximately one year after the appearance of this document, the source code for POSIT will be located DECTAPE D2315 and may be copied from this source. Subsequently, the program will be available directly from the authors.

Table B-1. List of variables

Name	Type[a]	Dimension	Purpose
		Input variables	
N	i		Number of compartments
A	rd	14,15	Matrix of transfer rates
TIMEO	rd	14,15	Matrix of transfer times
I	i		Ith and Jth location of A and TIMEO elements
J	i		
B	rd	14	Vector of forcing functions
		Execution variables	
PROBA	rd	14,14	Normalized A matrix
WORK1	rd	14,14	Work arrays used for matrix multiplication
WORK2	rd	14,14	
TSUM	rd	14,14	Sume of transfer times from compartment i to j
FLUX	rd	14,14	A*B
		Output variables	
POSIT	rd	6,14	Matrix of position calculations

[a]i = integer, rd = real double precision

POSIT Input

Input requirements for POSIT consist of model data and specific instructions for interactive execution. Input variables (Table B-1) include the number of compartments in the model (N), the transfer matrix (A), the rate of transfer matrix (TIMEØ), and the forcing function vector (B). Input of transfer matrices (A and TIMEØ) is simplified by reading in I and J values that indicate the location of nonzero elements. The N + 1 row of the A matrix contains the parameters for losses from the system (at least one element of this row must be nonzero).

These input values can either be read from a user data file or from input directly during execution (see Table B-2 for a sample data file). All values are accepted in free format with a row of zeros indicating the completion of data input. Specific input ordering is: I,J, A(I,J), TIMEØ (I,J) values per record. For data file input, N forcing function values follow. The default values for the TIMEØ matrix and the B vector values are set such that TIMEØ(I,J) = 1.0 and B(1) = 1, B(i) = 0, i = 2,N.

Program Options

After initializing terminal type and desired position calculations, program control is transferred to the user. At this point, the user is asked for some command by the question "NEXT OPTION?" A summary of commands and their meanings is given (Table B-3) as a result of the user specifying the command L.

Table B-2. Sample data file[a]

	I	J	a_{ij}	T_{ij}
Transfers:	5	1	0.2	1.0
	6	1	0.3	1.0
	5	2	0.3	1.0
	4	2	0.3	1.0
	6	3	0.5	1.0
	7	4	0.4	1.0
	4	5	0.1	1.0
	7	5	0.1	1.0
	8	6	0.25	1.0
	5	6	0.25	1.0
	1	7	0.2	1.0
	5	8	0.4	1.0
Respirations (losses):	9	1	0.5	1.0
	9	2	0.4	1.0
	9	3	0.5	1.0
	9	4	0.6	1.0
	9	5	0.8	1.0
	9	6	0.5	1.0
	9	7	0.8	1.0
	9	8	0.6	1.0

[a]This data file corresponds to the model in Fig. 1.

Table B-3. Sample computer session*

```
.EX POSIT [6137,347]
LINK:  LOADING
[LNKXCT POSIT EXECUTION]

TERMINAL TYPE?  (0 LISTS)
0

1 --- TERMINAL WITH HARD COPY
2 --- SCOPE TERMINAL

TERMINAL TYPE?  (0 LISTS)
1

SUMMARY OF POSITION DEFINITIONS?
Yes

1 --- PROB. MOLECULE PASSES THRU A COMPARTMENT (PPTC)
2 --- EXPECTED NUMBER OF PASSES FROM COMP. 1 TO N (ENPTC)
3 --- TROPHIC DISTANCE BETWEEN COMP. 1 AND COMP. N (TD)
4 --- MEAN TIME OF TRANSFER FROM COMP. 1 TO N (MTT)
5 --- AVERAGE RECURRENCE TIME (FEEDBACK ONLY) (MRT)
6 --- EFFECTIVE TROPHIC POSITION (ETP)

NUMBER OF POSITION CALCULATIONS?  (6 MAXIMUM)
5

POSITIONS TO BE CALCULATED?  (SEPARATED BY COMMA)
1,2,3,4,5

NEXT OPTION?  (L LISTS)
L

CP --- CHANGE WHICH POSITIONS ARE CALCULATED
ZM --- ZERO MATRICES AND INPUT NEW PARAMETERS
AM --- AUGMENT MATRIX AND/OR CORRECT EXISTING MATRIX
CF --- CHANGE FORCING FUNCTION
TM --- TYPE OUT MATRIX AND FORCING FUNCTION
```

Table B-3. (continued)

```
CO --- CHANGE OUTPUT TYPE
EX --- EXECUTE
L  --- LIST OPTIONS
SD --- SAVE DATA IN USER DEFINED DATA FILE
SS --- PRINT STEADY STATE VALUES
PM --- CHANGE MINIMUM PROBABILITY
SP --- STOP PROGRAM


NEXT OPTION? (L LISTS)
ZM

NUMBER OF COMPARTMENTS?
8

INPUT TYPE?  (0 LISTS)
0

1 --- INPUT FROM DATA FILE
2 --- INPUT FROM TERMINAL

INPUT TYPE?  (0 LISTS)
2

TIME MATRIX INPUT?  (0 LISTS)
0

1 --- TIME MATRIX SET TO ONE
2 --- INPUT TIME MATRIX

TIME MATRIX INPUT?  (0 LISTS)
1

A MATRIX OF MEAN TRANSFERS
TYPE IN I, J, A VALUE SEPARATED BY COMMAS
0, 0, 0 WILL STOP INPUT
5  1  .2
6  1  .3
9  1  .5
5  2  .3
9  2  .4
```

Table B-3. (continued)

```
6  3  .5
9  3  .5
7  4  .4
9  4  .6
4  5  .1
7  5  .1
9  5  .8
8  6  .25
5  6  .25
9  6  .5
1  7  .2
9  7  .8
5  8  .4
9  8  .6
0  0  0
```

NEXT OPTION? (L LISTS)
TM

INPUT VALUES:

I	J	A VALUE	T VALUE
1	7	0.200000	1.000000
4	2	0.300000	1.000000
4	5	0.100000	1.000000
5	1	0.200000	1.000000
5	2	0.300000	1.000000
5	6	0.250000	1.000000
5	8	0.400000	1.000000
6	1	0.300000	1.000000
6	3	0.500000	1.000000
7	2	0.400000	1.000000
7	4	0.400000	1.000000
7	5	0.100000	1.000000
8	6	0.250000	1.000000
9	1	0.500000	1.000000
9	3	0.500000	1.000000
9	4	0.600000	1.000000
9	5	0.800000	1.000000
9	6	0.500000	1.000000
9	7	0.800000	1.000000
9	8	0.600000	1.000000

Table B-3. (continued)

```
FORCING FUNCTION:
      1.0000    0.0000    0.0000    0.0000    0.0000    0.0000

NEXT OPTION?  (L LISTS)
SS

COMPARTMENT          STEADY STATE
     1                1.00861360
     2                0.00000000
     3                0.00000000
     4                0.03076271
     5                0.30762713
     6                0.30258407
     7                0.04306780
     8                0.07564602

NEXT OPTION?  (L LISTS)
EX

COMP. NO.   PPTC     ENPTC    TD       MTT      MRT
    1       1.0000   1.0736   1.0000   1.0000   1.0380
    2       0.0000   0.0000   0.0000   0.0000   0.0000
    3       0.0000   0.0000   0.0000   0.0000   0.0000
    4       0.0307   0.0327   3.6514   2.4872   3.4426
    5       0.3050   0.3274   2.6322   1.4426   2.6842
    6       0.3017   0.3221   2.0185   1.0183   3.9026
    7       0.0427   0.0458   3.9137   2.7283   3.2651
    8       0.0756   0.0805   3.2189   2.3998   4.2857

NEXT OPTION?  (L LISTS)
SD

OUTPUT DATA FILE?
MDL1

NEXT OPTION?  (L LISTS)
SP
STOP
```

Table B-3. (continued)

```
END OF EXECUTION
CPU TIME:  0.66 ELAPSED TIME:  3:53.20
EXIT
```

*This sample corresponds to the model in Fig. 1 and the parameters in Table A-2.

Input questions and responses are initiated by the commands ZM and AM. The ZM option causes the transfer matrices to be set to zero and then expects new data. The AM command is used to correct, delete, or add to existing matrix values. Current transfer matrix and forcing function values are typed for the user in response to TM.

Forcing function values can be input or changed from the default value by typing CF. The SD option causes transfer matrices and forcing function values to be written in a user defined data file.

After input is completed, a useful check for correctness is the calculation of steady state values (SS option) for the model. EX is the command which causes the position indices to be calculated and output. If necessary, POSIT provides a more detailed output (CO option) which results in the A, TIMEØ, and PROBA (probability matrix) matrices to be outputed for each position calculation. It should be noted that for scope terminals, a carriage return after each matrix output is necessary to continue execution.

The option CP allows the user to change the positions that are calculated. SP concludes the terminal session by stopping execution.

Program Limitations

1. Underflow occurs when respiration values are zero.
2. Overflow occurs on highly looped models.
3. Data files are expected to have the extension DAT.
4. Positions 3, 4, and 5 tend to execute very slowly if highly looped.

APPENDIX C
Listing of POSIT

Listing of the Fortran program which calculates the transfer probabilities, recurrence intervals, and positional indices for linear compartment models.

```
              IMPLICIT REAL*4(A-H,O-Z)
              LOGICAL INPUT
              DIMENSION AMUX(14,15),Z(15),POSIT(7,14),IPOS(7),SS(14)
              COMMON/BLKT/TIMEO(15,15),PROBMN
              COMMON/BLKIND/INDEX(20),JNDEX(20),N
              COMMON/BLKMAT/A(15,15),B(15)
              DIMENSION OPTION(12),IZERO(4),PTITLE(6),XX(2)
              DATA OPTION/'CP','ZM','AM','CP','TM','CO',
            . 'EX','L ','SD','SS','PM','SP'/
              DATA IZERO/0,0,0,0/
              DATA PTITLE/'PPTC','ENPTC',' TD ',' MTT ',' MRT ','ETP'/
              DATA INPUT/.FALSE./
C
C   TERMINAL TYPE --- ITERM
C   1 --- NO ACCEPT COMMANDS
C   2 --- ACCEPT COMMANDS
C
1             TYPE 5
5             FORMAT('OTERMINAL TYPE? (0 LISTS)')
              ACCEPT 25,ANS
              IF(ANS.EQ.'1'.OR.ANS.EQ.'2') GO TO 8
              TYPE 7
7             FORMAT('01 --- TERMINAL WITH HARD COPY'/
            . ' 2 --- SCOPE TERMINAL ')
              GO TO 1
8             ITERM=2
              IF(ANS.EQ.'1') ITERM=1
C
C   POSITION DEFINITIONS
C
100           TYPE 10
10            FORMAT('OSUMMARY OF POSITION DEFINITIONS?')
              ACCEPT 25,ANS
25            FORMAT(A1)
              IF(ANS.NE.'Y') GO TO 30
              TYPE 20
20            FORMAT('0 1 --- PROB. MOLECULE PASSES THRU A COMPARTMENT (PPTC)'/
            . '   2 --- EXPECTED NUMBER OF PASSES FROM COMP.1 TO N (ENPTC)'/
            . '   3 --- TROPHIC DISTANCE BETWEEN COMP. 1 AND COMP. N (TD)'/
            . '   4 --- MEAN TIME OF TRANSFER FROM COMP. 1 TO N (MTT)'/
            . '   5 --- AVERAGE RECURRENCE TIME (FEEDBACK ONLY) (MRT)'/
            . '   6 --- EFFECTIVE TROPHIC POSITION (ETP)')
30            TYPE 40
40            FORMAT('ONUMBER OF POSITION CALCULATIONS? (6 MAXIMUM)')
              ACCEPT *,NPOS
              IF(NPOS.GT.7.OR.NPOS.LT.1) GO TO 30
              TYPE 50
50            FORMAT('OPOSITIONS TO BE CALCULATED? (SEPERATED BY COMMA)')
              ACCEPT *,(IPOS(K),K=1,NPOS)
              GO TO 1000
C
C   ZERO A MATRIX
C
200           CONTINUE
C
C   SET DEFAULT VALUES
C
              PROBMN=.0001
              IOUT=2
              B(1)=1.D0
              DO 99 I=2,14
99            B(I)=0.D0
              INPUT=.TRUE.
              DO 109 J=1,15
              DO 109 I=1,14
              TIMEO(I,J)=0.
109           AMUX(I,J)=0.
```

APPENDIX C (continued)

```
              TYPE 110
      110     FORMAT('ONUMBER OF COMPARTMENTS?')
              ACCEPT *,N
      115     TYPE 120
      120     FORMAT('OINPUT TYPE ? (O LISTS)')
              ACCEPT 25,INP
              IF(INP.EQ.'1'.OR.INP.EQ.'2') GO TO 131
              TYPE 125
      125     FORMAT('01 --- INPUT FROM DATA FILE'/
          .   ' 2 --- INPUT FROM TERMINAL')
              GO TO 115
      C
      C  TIME MATRIX INPUT OPTION
      C
      131     TYPE 132
      132     FORMAT('OTIME MATRIX INPUT? (O LISTS)')
              ACCEPT 25,TINP
              IF(TINP.EQ.'1'.OR.TINP.EQ.'2') GO TO 134
              TYPE 133
      133     FORMAT('01 --- TIME MATRIX SET TO ONE'/
          .   ' 2 --- INPUT TIME MATRIX')
              GO TO 131
      134     KTIME=1
              IF(TINP.EQ.'2') KTIME=2
              IF(INP.EQ.'2') GO TO 300
      C
      C  DATA FILE INPUT
      C
      126     TYPE 127
      127     FORMAT('OINPUT DATA FILE?')
              ACCEPT 128,FNAM
      128     FORMAT(A10)
              OPEN(UNIT=25,ACCESS='SEQIN',FILE=FNAM)
              IUNIT=25
              GO TO 140
      C
      C  INPUT A MATRIX
      C
      300     IUNIT=5
              IF(KTIME.EQ.1) TYPE 129
      129     FORMAT('OA MATRIX OF MEAN TRANSFERS'/
          .   ' TYPE IN I, J, A VALUE SEPERATED BY COMMAS'/
          .   ' 0,0,0 WILL STOP INPUT')
              IF(KTIME.EQ.2) TYPE 130
      130     FORMAT('OA MATRIX OF MEAN TRANSFERS AND TIME MATRIX'/
          .   ' TYPE IN I, J, A VALUE, T VALUE SEPERATED BY COMMAS'/
          .   ' 0,0,0,0 WILL STOP INPUT')
      140     READ(IUNIT,*) J,I,(XX(K),K=1,KTIME)
              IF(I.LE.0.OR.J.LE.0) GO TO 150
              TIMEO(J+1,I+1)=1.0
              IF(KTIME.EQ.2) TIMEO(J+1,I+1)=XX(2)
              AMUX(I,J)=XX(1)
              GO TO 140
      150     IF(IUNIT.EQ.25) READ(IUNIT,*,END=151) (B(I),I=1,N)
      151     NN=N+1
              GO TO 1000
      C
      C  INPUT FORCING FUNCTION
      C
      400     TYPE 135
      135     FORMAT('OFORCING FUNCTION (NCOMP VALUES)')
              ACCEPT *, (B(I),I=1,N)
              GO TO 1000
      C
      C  TYPE OUT AUGMENTED MATRIX
      C
      500     TYPE 160
      160     FORMAT('OINPUT VALUES:'//'  I',2X,' J',4X,
          .'  A VALUE    ',4X,'T VALUE    ')
              IFORM=0
              DO 163 J=1,NN
              DO 163 I=1,N
              IF(ABS(AMUX(I,J)).LT.1000.AND.ABS(TIMEO(J+1,I+1)).LT.1000)
          .   GO TO 163
              IFORM=1
              GO TO 164
      163     CONTINUE
```

APPENDIX C (continued)

```
164     DO 165 J=1,NN
        DO 165 I=1,N
        IF(AMUX(I,J).EQ.0.) GO TO 165
        IF(IFORM.EQ.0) TYPE 155,J,I,AMUX(I,J),TIMEO(J+1,I+1)
155     FORMAT(' ',2(I2,2X),2(F12.6,2X))
        IF(IFORM.EQ.1) TYPE 156,J,I,AMUX(I,J),TIMEO(J+1,I+1)
156     FORMAT(' ',2(I2,2X),2(E12.6,2X))
165     CONTINUE
C
C  TYPE OUT FORCING FUNCTION
C
        TYPE 166,(B(I),I=1,N)
166     FORMAT(//'0FORCING FUNCTION:'/(' ',8F10.4))
        GO TO 1000
C
C  OUTPUT TYPE?
C
600     TYPE 180
180     FORMAT('0OUTPUT? (0 LISTS TYPES)')
        ACCEPT 25,ANS
        IF(ANS.EQ.'1'.OR.ANS.EQ.'2') GO TO 191
        TYPE 190
190     FORMAT('01 --- FULL OUTPUT'/
     .  ' 2 --- SUMMARY OF POSITION VALUES')
        GO TO 600
191     IOUT=1
        IF(ANS.EQ.'2') IOUT=2
        GO TO 1000
C
C  EXECUTION
C
C
C  SET A VALUES TO AMUX VALUES
C
700     DO 167 I=1,N
        DO 167 J=1,NN
167     A(I,J)=AMUX(I,J)
C
C  CALCULATE A DIAGONAL VALUES
C
        DO 170 I=1,N
        DO 170 J=1,NN
        IF(I.EQ.J) GO TO 170
        A(I,I)=A(I,I)-A(I,J)
170     CONTINUE
C
C  CHECK IF RESPIRATION VALUES ARE ZERO
C
        DO 171 I=1,N
        IF(A(I,I).NE.0.) GO TO 171
        TYPE 172,I
172     FORMAT('0NO RESPIRATION LOSS FROM COMPARTMENT ',I2/
     .  ' VALUE SET TO ONE')
        A(I,I)=1.
171     CONTINUE
        DO 250 K=1,NPOS
        IF(IOUT.EQ.2) GO TO 236
        TYPE 210,IPOS(K)
210     FORMAT('0EXECUTION OF POSITION ROUTINE ',I1)
        TYPE 220
220     FORMAT('0A MATRIX ')
        DO 230 I=1,N
230     TYPE 235,(A(I,J),J=1,N)
235     FORMAT(' ',10F10.4)
        IF(ITERM.EQ.2) ACCEPT 25,ANS
236     IF(IPOS(K).EQ.1) CALL POSIT1(Z,IOUT,ITERM)
        IF(IPOS(K).EQ.2) CALL POSIT2(Z,IOUT,ITERM)
        IF(IPOS(K).EQ.3) CALL POSIT3(Z,IOUT,ITERM)
        IF(IPOS(K).EQ.4) CALL POSIT4(Z,IOUT,ITERM)
        IF(IPOS(K).EQ.5) CALL POSIT5(Z,IOUT,ITERM)
        IF(IPOS(K).EQ.6) CALL POSIT6(Z,IOUT,ITERM)
        DO 240 I=1,N
240     POSIT(K,I)=Z(I)
250     CONTINUE
C
C  PRINT RESULTS
```

APPENDIX C (continued)

```
C
        TYPE 260,(PTITLE(IPOS(K)),K=1,NPOS)
260     FORMAT(///'0','COMP. NO. ',7(2X,A5,2X))
        IFORM=0
        DO 265 I=1,N
        DO 265 K=1,NPOS
        IF(ABS(POSIT(K,I)).LT.1000.) GO TO 265
        IFORM=1
        GO TO 269
265     CONTINUE
269     DO 270 I=1,N
        IF(IFORM.EQ.0) TYPE 280,I,(POSIT(K,I),K=1,NPOS)
280     FORMAT('0',4X,I2,3X,7F9.4)
        IF(IFORM.EQ.1) TYPE 285,I,(POSIT(K,I),K=1,NPOS)
285     FORMAT('0',4X,I2,3X,7E9.3)
270     CONTINUE
C
C  NEXT OPTION?
C
1000    TYPE 1010
1010    FORMAT('ONEXT OPTION ? (L LISTS)')
        ACCEPT 88,OPT
88      FORMAT(A5)
        DO 1020 I=1,12
1020    IF(CPT.EQ.OPTION(I)) GO TO 1012
        TYPE 1011,OPT
1011    FORMAT('0',A2,' IS AN ILLEGAL OPTION')
        GO TO 1000
1012    IF(INPUT) GO TO 1014
        IF(I.EQ.2.OR.I.EQ.1.OR.I.EQ.6.OR.I.EQ,8.OR.I.EQ.11.OR.I.EQ.12)
     .   GO TO 1014
        TYPE 1013
1013    FORMAT('0MATRIX HAS NOT BEEN INPUT --- TYPE ZM TO INPUT')
        GO TO 1000
1014    GO TO (100,200,300,400,500,600,700,800,850,875,899,900),I
000     TYPE 001
801     FORMAT('0CP --- CHANGE WHICH POSITIONS ARE CALCULATED'/
     .  ' ZM --- ZERO MATRICES AND INPUT NEW PARAMETERS'/
     .  ' AM --- AUGMENT MATRIX AND/OR CORRECT EXISTING MATRIX'/
     .  ' CF --- CHANGE FORCING FUNCTION'/
     .  ' TM --- TYPE OUT MATRIX AND FORCING FUNCTION'/
     .  ' CC --- CHANGE OUTPUT TYPE'/
     .  ' EX --- EXECUTE'/
     .  ' L  --- LIST OPTIONS'/
     .  ' SD --- SAVE DATA IN USER DEFINED DATA FILE'/
     .  ' SS --- PRINT STEADY STATE VALUES'/
     .  ' PM --- CHANGD MINIMUM PROD. FROM J TO J FROM .0001'/
     .  ' SE --- STOP PROGRAM')
        GO TO 1000
C
C  OPTION TC SAVE A MATRIX IN DATA FILE
C
850     TYPE 851
851     FORMAT('0OUTPUT DATA FILE?')
        ACCEPT 128,FNAM
        OPEN(UNIT=26,ACCESS='SEQOUT',FILE=FNAM)
        DO 860 J=1,NN
        DO 860 I=1,N
        IF(AMUX(I,J).EQ.0.) GO TO 860
        WRITE(26,855) J,I,AMUX(I,J),TIMEO(J+1,I+1)
855     FORMAT(2(I2,2X),2(E20.15,2X))
860     CONTINUE
        WRITE(26,856) IZERO
856     FORMAT(4I2)
        WRITE(26,857) (B(I),I=1,N)
857     FORMAT(6(E12.6,1X))
        CLOSE(UNIT=26,ACCESS='SEQOUT',FILE=FNAM)
        GO TO 1000
C
C  CALCULATE AND PRINT STEADY STATE VALUES
C
875     DO 876 I=1,N
        SS(I)=-B(I)
        DO 876 J=1,NN
876     A(J,I)=AMUX(I,J)
        DO 877 I=1,N
        A(I,I)=0.
        DO 877 J=1,NN
        IF(I.EQ.J) GO TO 877
```

APPENDIX C (continued)

```
              A(I,I)=A(I,I)-A(J,I)
  877         CONTINUE
              CALL DMATEQ(A,SS,N,1,15)
              IFORM=0
              DO 878 I=1,N
              IF(ABS(SS(I)).LT.1000.) GO TO 878
              IFORM=1
              GO TO 879
  878         CONTINUE
  879         IF(IFORM.EQ.0) TYPE 880,(I,SS(I),I=1,N)
  880         FORMAT('0COMPARTMENT',6X,'STEADY STATE'/
             - (5X,I2,11X,F12.8))
              IF(IFORM.EQ.1) TYPE 881,(I,SS(I),I=1,N)
  881         FORMAT(5X,I2,11X,E12.6)
              GO TO 1000
C
C  CHANGE PROBABILITY MINIMUM FOR .0001
C
  899         TYPE 888
  888         FORMAT('0PROBABILITY MINIMUM?')
              ACCEPT *,PROBMN
              GO TO 1000
  900         STOP
              END
```

APPENDIX C (continued)

```
      SUBROUTINE POSIT1(P1,IOUT,ITERM)
C*************************************************************
C                                                            *
C   POSIT1 CALCULATES THE PROBABILITY A MOLECULE PASSES      *
C   THROUGH A COMPARTMENT --- PPTC                           *
C                                                            *
C*************************************************************

      IMPLICIT REAL*4(A-H,O-Z)
      DIMENSION P1(15)
      COMMON/BLK1/F(15),PROBA(15,15),P(15,15),PP(15,15)
      COMMON/BLKMAT/ A(15,15),B(15)
      COMMON/BLKIND/ INDEX(20),JNDEX(20),N
C
C INITIALIZE PROBABILITY MATRIX AND FORCING FUNCTION
C
      NN=N+1
C
C  NORMALIZE FORCING FUNCTION B
C
      SUM=0.
      DO 50 I=1,N
50    SUM=SUM+B(I)
      DO 51 I=1,N
51    F(I)=-B(I)/SUM
      F(NN)=0.
      IF(IOUT.EQ.2) GO TO 53
      TYPE 52,(F(I),I=1,N)
52    FORMAT('0FORCING FUNCTION  VALUES'/1H ,8F10.3)
      IF(ITERM.EQ.2) ACCEPT 101,ANS
53    DO 300 KK=1,NN
      DO 5 I=1,NN
      P1(I)=0.D0
      PROBA(I,I)=-1.D0
5     CONTINUE
      P1(KK)=-1.D0
C
C  NORMALIZE A TO FORM PROB. MATRIX
C
      DO 10 I=1,N
      DO 10 J=1,N
      IF(I.EQ.J) GO TO 10
      PROBA(J,I)=A(I,J)/(-A(I,I))
10    CONTINUE
      IF(KK.NE.1.OR.IOUT.EQ.2) GO TO 201
      TYPE 199
199   FORMAT('0PROBA MATRIX')
      DO 200 I=1,N
      TYPE 100,(PROBA(I,J),J=1,N)
100   FORMAT(' ',8F10.4)
200   CONTINUE
      IF(ITERM.EQ.2) ACCEPT 101,ANS
101   FORMAT(A5)
201   CONTINUE
      IF(KK.EQ.NN) GO TO 301
C
C  SOLVE FOF POSITION BY SOLVING EQN. PROBA*P1=-F
C
      CALL DMATEQ(PROBA,P1,N,1,15)
C
C  TYPE OUT ORIGINAL STEADY STATE
C
C     IF(KK.EQ.1) TYPE 245,P1
245   FORMAT('0ORIGINAL STEADY STATE VALUES:'/1H ,14F10.4)
      DO 250 I=1,N
250   P(KK,I)=P1(I)
300   CONTINUE
301   CALL DMATEQ(PROBA,P,N,1,15)
C     TYPE 52,(F(I),I=1,N)
C
C  CALCULATE POSITION
C
      DO 350 I=1,N
350   P1(I)=P(I)/P(I,I)
      RETURN
      END
```

APPENDIX C (continued)

```
      FUNCTION DMATEQ(A,B,III,JJJ,ID)
C*********************************************************************
C                                                                    *
C   DMATEQ IS A SUBROUTINE TO SOLVE THE  SYSTEM A*X=B.               *
C      A WRITE UP MAY BE FOUND IN CTC-39 (WESTLEY AND WATTS 1970).   *
C                                                                    *
C*********************************************************************
       REAL*4  A,B,R,S,D
      DIMENSION A(ID,1),B(ID,1)
      KK=III
      NV=IABS(JJJ)
      D=1.
      IF(JJJ.LT.0)  D=0.
      KKM=KK-1
      DO 90 I=1,KKM
      S=0.
      DO 10 J=I,KK
      R=ABS(A(J,I))
      IF(R.LT.S)GO TO 10
      S=R
      L=J
   10 CONTINUE
      IF(L.EQ.I)  GO TO 50
      DO 20 J=I,KK
      S=A(I,J)
      A(I,J)=A(L,J)
      A(L,J)=S
   20 CONTINUE
      IF(NV.LE.0)  GO TO 40
      DO 30 J=1,NV
      S=B(I,J)
      B(I,J)=B(L,J)
   30 B(L,J)=S
   40 D=-D
   50 IF(A(I,I).EQ.0.)  GO TO 90
      IPO=I+1
      DO 80 J=IPO,KK
      IF(A(J,I).EQ.0.0)  GO TO 80
      S=A(J,I)/A(I,I)
      A(J,I)=0.
      DO 60 K=IPO,KK
   60 A(J,K)=A(J,K)-A(I,K)*S
      IF(NV.LE.0)  GO TO 80
      DO 70 K=1,NV
   70 B(J,K)=B(J,K)-B(I,K)*S
   80 CONTINUE
   90 CONTINUE
      DO 100 I=1,KK
  100 D=D*A(I,I)
      IF(NV.LE.0)  GO TO 130
      KMO=KK-1
      DO  120 K=1,NV
      B(KK,K)=B(KK,K)/A(KK,KK)
      DO 120 I=1,KMO
      N=KK-I
      DO 110 J=N,KMO
      B(N,K)=B(N,K)-A(N,J+1)*B(J+1,K)
  110 CONTINUE
      B(N,K)=B(N,K)/A(N,N)
  120 CONTINUE
   99 CONTINUE
  130 DMATEQ=D
      RETURN
       END
```

APPENDIX C (continued)

```
      SUBROUTINE POSIT2(P,IOUT,ITERM)
C***********************************************************
C                                                          *
C   POSIT2 CALCULATES THE NUMBER OF PASSES FROM            *
C   COMPARTMENT 1 TO COMPARTMENT N                         *
C                                                          *
C***********************************************************
      IMPLICIT REAL*4(A-H,O-Z)
      REAL*4 P(15)
      COMMON/BLK1/PP(15),PROBA(15,15),WORK1(15,15),WORK2(15,15)
      COMMON/BLKMAT/ A(15,15),B(15)
      COMMON/BLKIND/ INDEX(20),JNDEX(20),N
      COMMON/BLKT/TIMEO(14,15),PROBMN
C
C  NORMALIZE FORCING FUNCTION
C
      PSUM=0.
      DO 1 I=1,N
1     PSUM=PSUM+B(I)
      DO 2 I=1,N
      PROBA(I+1,1)=B(I)/PSUM
2     WORK2(I+1,1)=PROBA(I+1,1)
      NN=N+1
      DO 5 I=1,NN
      P(I)=0.D0
      PROBA(I,I)=0.D0
      WORK2(I,I)=0.D0
5     CONTINUE
C
C  PROBA IS THE NORMALIZED A MATRIX
C  WITH THE DIAGONAL ELEMENTS ZERO
C
      DO 10 I=1,N
      DO 10 J=1,N
      IF(I.EQ.J) GO TO 10
      PROBA(J+1,I+1)=A(I,J)/(-A(I,I))
      WORK2(J+1,I+1)=PROBA(J+1,I+1)
10    CONTINUE
      NN=N+1
      IF(IOUT.EQ.2) GO TO 14
      TYPE 199
199   FORMAT('0PROBA MATRIX')
      DO 200 I=1,NN
200   TYPE 100,(PROBA(I,J),J=1,NN)
100   FORMAT(' ',8F10.4)
      IF(ITERM.EQ.2) ACCEPT 101,ANS
101   FORMAT(A5)
C
C  FIND LENGTHS OF PATHWAYS
C
14    PMIN=1.
      GO TO 301
15    DO 22 I=1,NN
      DO 20 K=1,NN
      WORK2(I,K)=0.D0
      DO 20 J=1,NN
20    WORK2(I,K)=WORK2(I,K)+WORK1(I,J)*PROBA(J,K)
C
C  A NONZERO DIAGONAL ELEMENT INDICATES A LOOP
C
      IF(WORK2(I,I).EQ.0.D0) GO TO 22
      PMIN=AMIN1(PMIN,WORK2(I,I))
22    CONTINUE
C
C  STORE PROB. FROM COMPARTMENT 1 TO N
C  WITH LENGTH STEPNO
C
301   ICHECK=0
      DO 25 I=1,NN
      IF(WORK2(I,1).EQ.0.) GO TO 25
      ICHECK=1
      P(I)=P(I)+WORK2(I,1)
25    CONTINUE
      IF(ICHECK.EQ.0) GO TO 35
      IF(PMIN.LT.PROBMN) GO TO 35
C
```

APPENDIX C (continued)

```
C  RESET WORK1 ARRAY
C
         DO 30 I=1,NN
         DO 30 J=1,NN
30       WORK1(I,J)=WORK2(I,J)
         GO TO 15
C
C  NUMBER OF STEPS EXHAUSTED
C
35       DO 40 I=1,N
         P(I)=P(I+1)
40       CONTINUE
         RETURN
         END
```

APPENDIX C (continued)

```
      SUBROUTINE POSIT3(P,IOUT,ITERM)
C****************************************************************
C                                                              *
C   POSIT3 CALCULATES THE TROPHIC DISTANCE BETWEEN             *
C   COMPARTEENT 1 AND COMPARTMENT N --- TD                     *
C                                                              *
C****************************************************************
      IMPLICIT REAL*4(A-H,O-Z)
      REAL*4 P(15)
      COMMON/BLK1/PP(15),PROBA(15,15),WORK1(15,15),WORK2(15,15)
      COMMON/BLKMAT/ A(15,15),B(15)
      COMMON/BLKIND/ INDEX(20),JNDEX(20),N
      COMMON/BLKT/TIMEO(15,15),PROBMN
C
C  NORMALIZE FORCING FUCTION
C
      FSUM=0.
      DO 1 I=1,N
1     FSUM=FSUM+B(I)
      DO 2 I=1,N
2     PROBA(I+1,1)=B(I)/FSUM
C
C  CALCULATE PROBA --- NORMALIZED A MATRIX
C
      NN=N+1
      DO 10 I=1,NN
      P(I)=0.
      PP(I)=0.
10    PROBA(I,I)=0.D0
      DO 111 I=1,N
      DO 111 J=1,N
      IF(I.EQ.J) GO TO 111
      PROBA(J+1,I+1)=A(I,J)/(-A(I,I))
111   CONTINUE
      IF(IOUT.EQ.2) GO TO 114
      TYPE 199
199   FORMAT('0PROBA MATRIX')
      DO 200 I=1,NN
200   TYPE 100,(PROBA(I,J),J=1,NN)
100   FORMAT(' ',8F10.4)
      IF(ITERM.EQ.2) ACCEPT 101,ANS
101   FORMAT(A5)
114   CONTINUE
C
C  FIND LENGTHS OF PATHWAYS
C
      DO 1000 NCOMP=2,NN
      DO 999 I=1,NN
      DO 999 J=1,NN
999   WORK1(I,J)=PROBA(I,J)
      PMIN=1.
      STEPNO=1.D0
      GO TO 301
21    STEPNO=STEPNO+1.D0
      DO 22 I=1,NN
      DO 20 K=1,NN
      WORK2(I,K)=0.D0
      DO 20 J=1,NN
20    WORK2(I,K)=WORK2(I,K)+WORK1(I,J)*PROBA(J,K)
C
C  A NONZERO DIAGONAL ELEMENT INDICATES A LOOP
C
      IF(I.EQ.NCOMP) WORK2(I,I)=0.
      IF(WORK2(I,I).EQ.0.D0) GO TO 22
      PMIN=AMIN1(PMIN,WORK2(I,I))
22    CONTINUE
C
C  STORE PROB. FROM COMPARTMENT 1 TO N
C  WITH LENGTH STEPNO
C
301   ICHECK=0
      DO 25 I=1,NN
      IF(WORK2(I,1).EQ.0.) GO TO 25
      ICHECK=1
      IF(I.NE.NCOMP) GO TO 25
      P(I)=P(I)+WORK2(I,1)*STEPNO
      PP(I)=PP(I)+WORK2(I,1)
25    CONTINUE
```

APPENDIX C (continued)

```
      IF(ICHECK.EQ.0) GO TO 1000
      IF(EMIN.LT.PROBMN) GO TO 1000
C
C  RESET WORK1 ARRAY
C
28    DO 30 I=1,NN
      DO 30 J=1,NN
30    WORK1(I,J)=WORK2(I,J)
      GO TO 21
1000  CONTINUE
C
C  NUMBER OF STEPS EXHAUSTED
C  CALCULATE POSITION
C
      DO 40 I=1,N
      IF(PP(I+1).EQ.0.) GO TO 39
      P(I)=P(I+1)/PP(I+1)
      GO TO 40
39    P(I)=0.
40    CONTINUE
      RETURN
      END
```

APPENDIX C (continued)

```
      SUBROUTINE POSIT4(P,IOUT,ITERM)
C**************************************************************
C                                                             *
C     POSIT4 CALCULATES THE MEAN TRANSFER TIME FROM           *
C     COMPARTMENT I TO J --- MTT                              *
C                                                             *
C**************************************************************
      IMPLICIT REAL*4(A-H,O-Z)
      COMMON/BLKT/TIME(15,15),PROBMN
      COMMON/BLKMAT/A(15,15),B(15)
      COMMON/BLKIND/INDEX(20),JNDEX(20),NC
      COMMON/BLKMTT/ PROB(15,15),TIMEO(15,15),WORK1(15,15,100),
     . WORK2(15,15,100),TIME1(15,15,100),TIME2(15,15,100),
     . NA2(15,15),NA1(15,15),TI0(15,15),TI1(15,15),
     . PP(15),PROBA(15,15)
      DIMENSION P(15)
C
C  NORMALIZE FORCING FUNCTION
C
      NN=NC+1
      PSUM=0.
      DO 1 I=1,NC
1     PSUM=PSUM+B(I)
C
C  CALCULATE PROBA --- NORMALIZED A MATRIX
C
      DO 10 I=1,NN
      PP(I)=0.
      P(I)=0.
      DO 10 J=1,NN
10    PROBA(I,J)=0.
      DO 111 I=1,NC
      DO 111 J=1,NC
      IF(I.EQ.J.OR.A(I,I).EQ.0.) GO TO 111
      PROBA(J+1,I+1)=A(I,J)/(-A(I,I))
111   CONTINUE
C
C  ADD COMPARTMENT 1 AS A DUMMY COMPARTMENT
C
      DO 11 I=1,NC
      PROBA(I+1,1)=B(I)/PSUM
      TIME(I+1,1)=PROBA(I+1,1)
11    CONTINUE
C
C  INITIALIZE VALUES
C
      DO 13 I=1,NN
      PP(I)=PROBA(I,1)
13    P(I)=PROBA(I,1)*TIME(I,1)
      SMAX=0.
      DO 1000 NCOMP=2,NN
C
C  TAKE OUT ALL LINKS AWAY FRCM COMPARTMENT
C
      DO 17 I=1,NN
      DO 17 J=1,NN
      PROB(I,J)=PRODA(I,J)
      TIMEO(I,J)=TIME(I,J)
      IF(J.NE.1) TIMEO(I,J)=TIMEO(I,J)/(-A(J-1,J-1))
      IF(J.NE.NCOMP) GO TO 17
      PROB(I,J)=0
      TIMEO(I,J)=0.
17    CONTINUE
      IF(IOUT.EQ.2) GO TO 1888
      TYPE 187,((PROB(I,J),J=1,NN),I=1,NN)
      TYPE 188,((TIMEO(I,J),J=1,NN),I=1,NN)
188   FORMAT(' TIMEO'/(' ',7F6.1))
187   FORMAT(' PROB'/(' ',7F6.1))
      IF(ITERM.EQ.2) ACCEPT 12,ANS
12    FORMAT(A1)
C
C  CALCULATE TIO
C
1888  DO 18 I=1,NN
      DO 18 J=1,NN
      NA2(I,J)=0
```

APPENDIX C (continued)

```
            TI0(I,J)=0.
            IF(TIME0(I,J).EQ.0.) GO TO 18
            TI0(I,J)=1.0
            NA2(I,J)=1
   18       CONTINUE
C
C  START PRCCESS
C
            PMIN=1.0
            STEPNO=1.
            DO 19 I=1,NN
            DO 19 J=1,NN
            TIME2(I,J,1)=TIME0(I,J)
   19       WORK2(I,J,1)=PROB(I,J)
            GO TO 301
  100       STEPNO=STEPNO+1.
            SMAX=AMAX1(SMAX,STEPNO)
            IF(IOUT.EQ.1)   TYPE 101,STEPNO,NCOMP
  101       FORMAT(' STEPNO=',F5.0,' NCOMP=',I5)
C
C  MULTIPLY A
C
            DO 300 I=1,NN
            DO 299 K=1,NN
            NA2(I,K)=0
            WORK2(I,K,1)=0.
            TIME2(I,K,1)=0.
            DO 20 J=1,NN
            N=NA1(J,K)
            IF(N.EQ.0) GO TO 20
            TEMPA=PROB(I,J)*WORK1(J,K,N)
            IF(TEMPA.EQ.0.) GO TO 20
            IF(N.EQ.1) GO TO 16
            NM1=N-1
            DO 15 L=1,NM1
            IF(NA.GT.300) GO TO 998
            NA=NA2(I,K)+L
   15       WORK2(I,K,NA)=WORK1(J,K,I)
   16       NA=NA2(I,K)+N
            IF(NA.GT.300) GO TO 998
            WORK2(I,K,NA)=TEMPA
            NA2(I,K)=NA
   20       CONTINUE
C
C  MULTIPLY T
C
            NT=0
            DO 35 J=1,NN
            T1=TI0(I,J)*TIME1(J,K,1)
            T2=TIME0(I,J)*TI1(J,K)
            IF(T1.EQ.0.AND.T2.EQ.0) GO TO 35
            N1=NA1(J,K)
            IF(N1.EQ.1) GO TO 34
            DO 33 L=2,N1
   33       TIME2(I,K,NT+L)=TIME1(J,K,L)+T2
   34       TIME2(I,K,NT+1)=T1+T2
            NT=NT+N1
   35       CONTINUE
            IF(NT.NE.NA2(I,K)) TYPE 36,I,K,STEPNO,NT,NA2(I,K)
   36       FORMAT(' I=',I2,' K=',I2,' STEPNO=',F10.0,
     .      ' NT=',I10,' NA(I,K)=',I10)
            IF(K.NE.1.OR.I.NE.NCOMP) GO TO 299
C
C  SAVE P=SUM OF P*T
C  SAVE PP =SUM P
C
            DO 40 N=1,NT
            P(I)=P(I)+WORK2(I,K,N)*(TIME2(I,K,N)-1.0)
   40       PP(I)=PP(I)+WORK2(I,K,N)
C           TYPE 41,STEPNO,I,NCOMP,P(I),PP(I)
   41       FORMAT(' STEPNO=',F5.0, ' I=',I2,' NCOMP=',I5,
     .      ' P(I)=',F10.4,' PP(I)=',F10.4)
  299       CONTINUE
C
C  FIND THE PROB. MINIMUM
C
```

APPENDIX C (continued)

```
      IF(WORK2(I,I,1) .EQ.0.) GO TO 300
      N=NA2(I,I)
      DO 50 L=1,N
50    PMIN=AMIN1(PMIN,WORK2(I,I,L))
300   CONTINUE
C
C  CHECK TO SEE IF THROUGH
C
301   ICHECK=0
      DO 55 I=1,NN
      IF(WORK2(I,1,1).EQ.0.) GO TO 55
      ICHECK=1
55    CONTINUE
      IF(ICHECK.EQ.0) GO TO 1000
      IF(PMIN.LT.PROBMN) GO TO 999
C
C  RESET WORK1 AND TIME1 ARRAYS
C
      DO 60 I=1,NN
      DO 60 J=1,NN
      N=NA2(I,J)
      IF(N.EQ.0) N=1
      DO 60 L=1,N
      WORK1(I,J,L)=WORK2(I,J,L)
      TIME1(I,J,L)=TIME2(I,J,L)
60    CONTINUE
C
C  RESET NA1 AND CALCULATE TI1
C
      DO 65 I=1,NN
      DO 65 J=1,NN
      NA1(I,J)=NA2(I,J)
      TI1(I,J)=0
      IF(TIME1(I,J,1).EQ.0.) GO TO 65
      TI1(I,J)=1.0
65    CONTINUE
      GO TO 100
998   TYPE 177,NCOMP
177   FORMAT(' NO. OF EQUAL PATHWAYS EXCEEDED FOR COMPARTMENT ',I2)
      GO TO 1000
999   TYPE 178,NCOMP
178   FORMAT(' PROB. MINIMUM REACHED FOR COMPARTMENT ',I2)
1000  CONTINUE
C
C  NUMBER OF STEPS EXHAUSTED
C  CALCULATE POSITION
C
      DO 70 I=1,NC
      IF(PP(I+1).EQ.0.) GO TO 69
      P(I)=P(I+1)/PP(I+1)
      GO TO 70
69    P(I)=0.
70    CONTINUE
      TYPE 200,SMAX
200   FORMAT(' MAXIMUM NO OF STEPS =',F5.0)
      RETURN
      END
```

APPENDIX C (continued)

```
      SUBROUTINE POSIT5(P,IOUT,ITERM)
C*************************************************************
C                                                            *
C       POSIT5 CALCULATES THE MEAN RECURRENCE TIME -- MRT    *
C                                                            *
C*************************************************************
        IMPLICIT REAL*4(A-H,O-Z)
        COMMON/BLKT/TIME(15,15),PROBMN
        COMMON/BLKMAT/A(15,15),B(15)
        COMMON/BLKIND/INDEX(20),JNDEX(20),NC
        COMMON/BLKMTT/ PROB(15,15),TIMEO(15,15),WORK1(15,15,100),
     .  WORK2(15,15,100),TIME1(15,15,100),TIME2(15,15,100),
     .  NA2(15,15),NA1(15,15),TI0(15,15),TI1(15,15),
     .  PP(15),PROBA(15,15)
        DIMENSION P(15)
C
C  NORMALIZE FORCING FUNCTION
C
        ICHECK=0
        NN=NC+1
        FSUM=0.
        DO 1 I=1,NC
1       FSUM=FSUM+B(I)
C
C  CALCULATE PROBA --- NORMALIZED A MATRIX
C
        DO 10 I=1,NN
        PP(I)=0.
        P(I)=0.
        DO 10 J=1,NN
10      PROBA(I,J)=0.
        DO 111 I=1,NC
        DO 111 J=1,NC
        IF(I.EQ.J.OR.A(I,I).EQ.0.) GO TO 111
        PROBA(J+1,I+1)=A(I,J)/(-A(I,I))
111     CONTINUE
C
C  ADD COMPARTMENT 1 AS A DUMMY COMPARTMENT
C
        DO 11 I=1,NC
        PROBA(I+1,1)=B(I)/FSUM
        TIME(I+1,1)=PROBA(I+1,1)
11      CONTINUE
C
C  INITIALIZE VALUES
C
        DO 13 I=1,NN
        PP(I)=PROBA(I,1)
13      P(I)=PROBA(I,1)*TIME(I,1)
        SMAX=0.
        DO 1000 NCOMP=2,NN
C
C  CHECK FOR FEEDBACK LOOPS -- IF NONE RETURN
C
C
        ICHECK=0
        DO 17 I=1,NN
        DO 17 J=1,NN
        PROB(I,J)=PROBA(I,J)
        TIMEO(I,J)=TIME(I,J)
        IF(J.NE.1) TIMEO(I,J)=TIMEO(I,J)/(-A(J-1,J-1))
17      CONTINUE
        IF(IOUT.EQ.2) GO TO 1888
        TYPE 187,((PROB(I,J),J=1,NN),I=1,NN)
        TYPE 188,((TIMEO(I,J),J=1,NN),I=1,NN)
188     FORMAT(' TIMEO'/(' ',7F6.1))
187     FORMAT(' PROB'/(' ',7F6.1))
        IF(ITERM.EQ.2) ACCEPT 12,ANS
12      FORMAT(A1)
C
C  CALCULATE TIO
C
1888    DO 18 I=1,NN
        DO 18 J=1,NN
        NA2(I,J)=0
```

APPENDIX C (continued)

```
      TI0(I,J)=0.
      IF(TIME0(I,J).EQ.0.) GO TO 18
      TI0(I,J)=1.0
      NA2(I,J)=1
18    CONTINUE
C
C  START PRCCESS
C
      PMIN=1.0
      STEPNO=1.
      DO 19 I=1,NN
      DO 19 J=1,NN
      TIME2(I,J,1)=TIME0(I,J)
19    WORK2(I,J,1)=PROB(I,J)
      GO TO 301
100   STEPNO=STEPNO+1.
      SMAX=AMAX1(SMAX,STEPNO)
      IF(IOUT.EQ.1)   TYPE 101,STEPNO,NCOMP
101   FORMAT(' STEPNO=',F5.0,' NCOMP=',I5)
C
C  MULTIPLY A
C
      DO 300 I=1,NN
      DO 299 K=1,NN
      NA2(I,K)=0
      WORK2(I,K,1)=0.
      TIME2(I,K,1)=0.
      DO 20 J=1,NN
      N=NA1(J,K)
      IF(N.EQ.0) GO TO 20
      TEMPA=PROB(I,J)*WORK1(J,K,N)
      IF(TEMPA.EQ.0.) GO TO 20
      IF(N.EQ.1) GO TO 16
      NM1=N-1
      DO 15 L=1,NM1
      IF(NA.GT.300) GO TO 998
      NA=NA2(I,K)+L
15    WORK2(I,K,NA)=WORK1(J,K,L)
16    NA=NA2(I,K)+N
      IF(NA.GT.300) GO TO 998
      WORK2(I,K,NA)=TEMPA
      NA2(I,K)=NA
20    CONTINUE
C
C  MULTIPLY T
C
      NT=0
      DO 35 J=1,NN
      T1=TI0(I,J)*TIME1(J,K,1)
      T2=TIME0(I,J)*TT1(J,K)
      IF(T1.EQ.0.AND.T2.EQ.0) GO TO 35
      N1=NA1(J,K)
      IF(N1.EQ.1) GO TO 34
      DO 33 L=2,N1
33    TIME2(I,K,NT+L)=TIME1(J,K,L)+T2
34    TIME2(I,K,NT+1)=T1+T2
      NT=NT+N1
35    CONTINUE
      IF(NT.NE.NA2(I,K)) TYPE 36,I,K,STEPNO,NT,NA2(I,K)
36    FORMAT(' I=',I2,' K=',I2,' STEPNO=',F10.0,
     .' NT=',I10,' NA(I,K)=',I10)
      IF(K.NE.1.OR.I.NE.NCOMP) GO TO 299
C
C  SAVE P=SUM OF P*T
C  SAVE PP =SUM P
C
      DO 40 N=1,NT
      P(I)=P(I)+WORK2(I,I,N)*(TIME2(I,I,N)-1.0)
40    PP(I)=PP(I)+WORK2(I,I,N)
299   CONTINUE
C
C  FIND THE PROB. MINIMUM
C
      IF(WORK2(I,I,1) .EQ.0.) GO TO 300
      ICHECK=1
      N=NA2(I,I)
      DO 50 L=1,N
50    PMIN=AMIN1(PMIN,WORK2(I,I,L))
300   CONTINUE
C
```

APPENDIX C (continued)

```
C  CHECK TO SEE IF THROUGH
C
301     ICHECK=0
        DO 55 I=1,NN
        IF(WORK2(I,1,1).EQ.0.) GO TO 55
        ICHECK=1
55      CONTINUE
        IF(ICHECK.EQ.0) GO TO 1000
        IF(PMIN.LT.PROBMN) GO TO 999
        IF(ICHECK.EQ.0.AND.STEPNO.EQ.25.) GO TO 350
C
C  RESET WORK1 AND TIME1 ARRAYS
C
        DO 60 I=1,NN
        DO 60 J=1,NN
        N=NA2(I,J)
        IF(N.EQ.0) N=1
        DO 60 L=1,N
        WORK1(I,J,L)=WORK2(I,J,L)
        TIME1(I,J,L)=TIME2(I,J,L)
60      CONTINUE
C
C  RESET NA1 AND CALCULATE TI1
C
        DO 65 I=1,NN
        DO 65 J=1,NN
        NA1(I,J)=NA2(I,J)
        TI1(I,J)=0
        IF(TIME1(I,J,1).EQ.0.) GO TO 65
        TI1(I,J)=1.0
65      CONTINUE
        GO TO 100
998     TYPE 177,NCOMP
177     FORMAT(' NO. OF EQUAL PATHWAYS EXCEEDED FOR COMPARTMENT ',I2)
        GO TO 1000
999     TYPE 178,NCOMP
178     FORMAT(' PROB. MINIMUM REACHED FOR COMPARTMENT ',I2)
1000    CONTINUE
C
C  NUMBER OF STEPS EXHAUSTED
C  CALCULATE POSITION
C
        DO 70 I=1,NC
        IF(PP(I+1).EQ.0.) GO TO 69
        P(I)=P(I+1)/PP(I+1)
        GO TO 70
69      P(I)=0.
70      CONTINUE
        TYPE 200,SMAX
200     FORMAT(' MAXIMUM NO OF STEPS =',F5.0)
        RETURN
350     TYPE 351
351     FORMAT(' **** NO FEEDBACK LOOPS OCCUR ****',
     .  ' *** POSITION 5 NOT CALCULATED *** ')
        RETURN
        END
```

APPENDIX C (continued)

```
      SUBROUTINE POSIT6(Z,IOUT,ITERM)
C***********************************************************
C                                                          *
C   POSIT6 CALCULATES EFFECTIVE TROPHIC POSITION --- ETP   *
C                                                          *
C***********************************************************
      IMPLICIT REAL*4(A-H,O-Z)
      COMMON/BLKMAT/A(15,15),B(15)
      COMMON/BLOCKA/X(14),TAU,RU(20)
      COMMON/BLKIND/INDEX(20),JNDEX(20),N
      COMMON/BLK1/ZPR(15),WORK1(15,15),WORK2(15,15),WORK3(15,15)
      REAL*4 TEMPA(15,15),FLUX(15,15),ZFLUX(15),Z(15),PSUM(15)
C
C  TEMPA --- LOWER TRIANGULAR FORM OF A MATRIX
C  X     --- INPUT TO DMATEQ AS B ; OUTPUT AS STEADY STATE VALUES
C
      DO 10 I=1,N
      Z(I)=0.
      RPD(I)=0.
      X(I)=-B(I)
      DO 10 J=1,N
10    TEMPA(J,I)=A(I,J)
C
C  SOLVE FOR STEADY STATE VALUES --- X
C
      CALL DMATEQ(TEMPA,X,N,1,15)
      IF(IOUT.EQ.2) GO TO 14
      TYPE 11,(X(I),I=1,N)
11    FORMAT('0STEADY STATE VALUES:'/' ', 14F10.4)
      IF(ITERM.EQ.2) ACCEPT 12,ANS
12    FORMAT(A1)
C
C  RELOAD TEMPA MATRIX SINCE DMATEQ DESTROYS IT
C
14    DO 15 I=1,N
      DO 15 J=1,N
      WORK1(J,I)=A(I,J)
      WORK3(J,I)=A(I,J)
15    TEMPA(J,I)=A(I,J)
      DO 16 I=1,N
      WORK1(I,I)=0.
      WORK3(I,I)=0.
16    CONTINUE
C
C  CHECK FOR FEEDBACK LOOPS
C
300   ICHECK=0
      DO 500 I=1,N
      DO 400 K=1,N
      WORK2(I,K)=0.D0
      DO 400 J=1,N
400   WORK2(I,K)=WORK2(I,K)+WORK1(I,J)*WORK3(J,K)
      IF(WORK2(I,I).NE.0.) GO TO 65
      IF(WORK2(I,1).NE.0.) ICHECK=1
500   CONTINUE
      DO 600 I=1,N
      DO 600 J=1,N
600   WORK1(I,J)=WORK2(I,J)
      IF(ICHECK.EQ.1) GO TO 300
C
C  FLUX MATRIX --- A*X
C  ZFLUX --- ROW SUM OF FLUX MATRIX
C
18    DO 20 I=1,N
      FLUX(I,I)=0.
      ZFLUX(I)=0.
      DO 20 J=1,N
      IF(I.EQ.J) GO TO 20
      FLUX(I,J)=TEMPA(I,J)*X(J)
      ZFLUX(I)=ZFLUX(I)+FLUX(I,J)
20    CONTINUE
      IF(IOUT.EQ.2) GO TO 239
      TYPE 21
21    FORMAT('0FLUX MATRIX')
      DO 22 I=1,N
22    TYPE 23,(FLUX(I,J),J=1,N)
23    FORMAT(' ',10F10.4)
C
```

APPENDIX C (continued)

```
C  FSUM --- COLUMN SUM OF FLUX MATRIIX
C
239     DO 240 J=1,N
        FSUM(J)=0.
        DO 240 I=1,N
240     FSUM(J)=FSUM(J)+FLUX(I,J)
        IF(IOUT.EQ.2) GO TO 224
        TYPE 24,(ZFLUX(I),I=1,N)
24      FORMAT('0ZFLUX VALUES'/' ',10F10.4)
        TYPE 241,(FSUM(I),I=1,N)
241     FORMAT('0FSUM VALUES '/' ',10F10.4)
        IF(ITERM.EQ.2) ACCEPT 12,ANS
C
C  NNZ --- NUMBER OF NON-ZERO ELEMENTS OF A MATRIX
C  ASUM --- SUM OF NON-ZERO ELEMENTS OF A MATRIX
C  AMEAN --- 1./NNZ*ASUM
C
224     NNZ=0
        ASUM=0.
        DO 25 I=1,N
        DO 25 J=2,N
        IF(I.EQ.J.OR.TEMPA(I,J).EQ.0.) GO TO 25
        ASUM=ASUM+TEMPA(I,J)
        NNZ=NNZ+1
25      CONTINUE
        AMEAN=1./NNZ*ASUM
        IF(IOUT.EQ.2) GO TO 29
        TYPE 26,AMEAN
26      FORMAT('0AMEAN VALUE=',F10.4)
        IF(ITERM.EQ.2) ACCEPT 12,ANS
C
C  WORK2 --- STORES SUMS OF ZPR VALUES
C
29      DO 30 I=1,N
        DO 30 J=1,N
30      WORK2(I,J)=FLUX(I,J)
        GO TO 41
C
C  PATHWAYS GREATER THAN ONE
C
35      DO 40 I=1,N
        DO 40 K=1,N
        WORK2(I,K)=0.
        DO 40 J=1,N
        IF(FSUM(J).EQ.0) GO TO 40
        WORK2(I,K)=WORK2(I,K)+(WORK1(I,J)*FLUX(J,K))/FSUM(J)
40      CONTINUE
C
C  CALCULATE ZPR(I)
C
41      ICHECK=0
        DO 45 I=1,N
        IF(WORK2(I,1).EQ.0.) GO TO 45
        ICHECK=1
        ZPR(I)=ZPR(I)+WORK2(I,1)
45      CONTINUE
        IF(ICHECK.EQ.0) GO TO 55
C
C  RESET WORK1 ARRAY
C
        DO 50 I=1,N
        DO 50 J=1,N
50      WORK1(I,J)=WORK2(I,J)
        GO TO 35
        IF(IOUT.EQ.2) GO TO 55
        TYPE 51,(ZPR(I),I=1,N)
51      FORMAT('0ZPR VALUES'/' ' 10F10.4)
C
C  ALL STEPS EXHAUSTED CALCULATE POSITION
C
55      AMLOG=1./ALOG(AMEAN)
        IF(IOUT.EQ.2) GO TO 58
        TYPE 57,AMLOG
57      FORMAT('01./LOG(AMEAN)=',F10.4)
        IF(ITERM.EQ.2) ACCEPT 12,ANS
58      DO 60 I=1,N
        IF(ZPR(I).EQ.0..OR.ZFLUX(I).EQ.0.) GO TO 60
        Z(I)=AMLOG*ALOG(ZFLUX(I)/ZPR(I))+1.
60      CONTINUE
```

APPENDIX C (continued)

```
      RETURN
C
C  ERROR --- FEEDBACK LOCPS EXIST
C
65    TYPE 70
70    FORMAT('0POSITION 6 NOT CALCULATED -- FEEDBACKS EXIST')
      RETURN
      END
```

INTERNAL DISTRIBUTION

1-2. S. I. Auerbach
3. C. F. Baes, III
4. L. W. Barnthouse
5. A. A. Brooks
6-10. J. H. Carney
11. Y. H. Chan
12. S. W. Christensen
13-17. D. L. DeAngelis
18. M. I. Dyer
19. M. P. Farrell
20. D. E. Fields
21-25. R. H. Gardner
26. C. T. Garten
27. C. W. Gehrs
28. S. V. Kaye
29. G. G. Killough
30. B. L. Kirk
31. R. J. Luxmoore
32. F. C. Maienschein
33-37. J. B. Mankin
38. C. W. Miller
39. J. K. Munro
40. J. D. Newbold
41. E. M. Oblow
42. J. S. Olson
43. R. V. O'Neill
44-48. W. M. Post
49. J. W. Ranney
50. J. Schutt
51. C. C. Travis
52. D. S. Vaughan
53. J. B. Waide
54. J. A. Watts
55. D. E. Weller
56. G. W. Westly
57. R. M. Yoshiyama
58. G. T. Yeh
59-78. ESD Library
79-80. Central Research Library
81-82. Laboratory Records Dept.
83. Laboratory Records, ORNL-RC
84. ORNL Y-12 Technical Library
85. ORNL Patent Office

EXTERNAL DISTRIBUTION

86. T. F. Allen, Department of Botany, University of Wisconsin, Madison, WI 53706
87. M. B. Beck, International Institute for Applied Systems Analysis, Schloss Laxenburg, Laxenburg, Austria
88. William S. Beller, Environmental Protection Agency, 2701 Largo Place, Bowie, MD 20715
89. L. J. Bledsoe, Center for Quantitative Science, University of Washington, Seattle, WA 98105
90. D. B. Botkin, Environmental Studies Program, University of California, Santa Barbara, CA 93106
91. J. D. Buffington, Environmental Impact Study Division, Argonne National Laboratory, Argonne, IL 60439
92. W. G. Cale, Environmental Sciences Program, University of Texas at Dallas, Richardson, TX
93. J. Thomas Callahan, Deputy Director, Ecosystems Studies Program, National Science Foundation, Washington, DC 20550
94. H. Caswell, Ecology Section U-42, University of Connecticut, Storrs, CT 06268
95. Assistant Secretary for Environment, U.S. Department of Energy, Washington, DC 20545
96. D. C. Coleman, Natural Resources Ecology Laboratory, Colorado State University, Fort Collins, CO 80523

97. William J. Coppoc, Texaco, Inc., P.O. Box 509, Beacon, NY 12508
98. H. Craig, Scripps Institute of Oceanography, La Jolla, CA 92037
99. R. C. Dahlman, Office of Health and Environmental Research, Department of Energy, Washington, DC 20545
100. L. L. Eberhardt, Ecosystems Dept., Battelle Northwest, P.O. Box 999, Richland, WA 99352
101. A. R. Ek, Department of Forest Resources, College of Forestry, University of Minnesota, St. Paul, MN 55208
102. W. P. Elliott, Division of Ecological Research, Department of Energy, Washington, DC 20545
103. J. T. Finn, Department of Biology, University of Massachusetts, Amherst, MA
104. J. F. Franklin, Forestry Sciences Laboratory, 3200 Jefferson Way, Corvallis, OR 97331
105. Robert M. Garrels, Department of Marine, University of South Florida, 830 S. First Street, St. Petersburg, FL 33701
106. W. L. Gates, Oregon State University, Corvallis, OR 97330
107. J. J. Gift, Ecological Analysts Inc., 8600 LaSalle Road, Baliman, MD 21204
108. Tagir Gilmanov, Department of Hydrobiology and Ecology, Moscow State University, Moscow, USSR.
109. Frank B. Golley, Director, Division of Environmental Biology National Science Foundation, Washington, DC 20550
110. E. Halfon, Canada Centre for Inland Waters, Burlington, Ontario, L7R 4A6 Canada
111. B. Hannon, Center for Advanced Computation, University of Illinois, Urbana, IL 61801
112. W. F. Harris, Program Manager, Ecosystem Studies Program, National Science Foundation, Washington, DC 20550
113. G. W. Harrison, Department of Mathematics, University of Georgia, Athens, GA 30602
114. J. E. Hobbie, The Ecosystem Center, Marine Biological Laboratory, Woods Hole, MA 02543
115. Kenneth Hood, Office of Research and Development, Environmental Protection Agency, 401 M Street, SW., Room 621, RD 683, Washington, DC 20460
116. G. Innis, Ecology Center, Utah State University, Logan, UT 84321
117. W. C. Johnson, Department of Biology, Virginia Polytechnic Institute and S. U., Blacksburg, VA 24061
118. J. R. Kercher, Biomedical Division, Lawrence Livermore Laboratory, P.O. Box 808, Livermore, CA 94550
119. J. Kitchell, Laboratory of Limnology, University of Wisconsin, Madison, WI 53706
120. Ray Lassiter, U.S. Environmental Protection Agency, Athens, GA 30602
121. D. P. Lettenmaier, Fisheries Research Institute, WH-10, University of Washington, Seattle, WA 98195
122. R. Levins, Department Population Sciences, Harvard School of Public Health, Boston, MA 02100

123. R. A. Lewis, EV-34, GTN, U.S. Department of Energy, Washington, DC 20545
124. O. Loucks, The Institute of Ecology, Holcomb Research Institute, Butler University, Indianapolis, IN 46208
125. Charles P. Maine, SE Towers Purdential Center, Boston, MA 02199
126. J. H. Matis, Institute of Statistics, Texas A and M University, College Station, TX 77840
127. R. May, Department of Biology, Princeton University, Princeton, NJ 08540
128. Helen McCammon, Director, Division of Environmental Research, Department of Energy, Washington, DC 20545
129. J. F. McCormick, Graduate Program in Ecology, University of Tennessee, Knoxville, TN 37916
130. P. C. Miller, Biology Department, San Diego State College, San Diego, CA 92115
131. Barren Moore, Ecosystems Center, Woods Hole Oceanographic Institute, Woods Hole, MA 02543
132. R. J. Mulholland, E. S. 303, Oklahoma State University, Stillwater, OK 74074
133. J. V. Nabholz, Office of pesticides and Toxic Substances, U.S. Environmental Protection Agency, 401 M Street SW, Washington, DC 20460
134. A. Nir, Weizman Institute, Rehovot, Israel
135. E. P. Odum, Department of Zoology, University of Georgia, Athens, GA 30602
136. H. T. Odum, Department of Environmental Engineering, University of Florida, Gainsville, FL 32601
137. W. S. Osburn, Division of Ecological Research, Office of Health and Environmental Research, Department of Energy, Washington, DC 20545
138. B. C. Patten, Department of Zoology, University of Georgia, Athens, GA 30602
139. J. J. Reisa, Council on Environmental Quality, 722 Jackson Place, Washington, DC 20006
140. F. P. Richards, SE Towers Prudential Center, Boston, MA 02199
141. J. E. Richey, Fisheries Research Institute, WH-10, University of Washington, Seattle, WA 98195
142. Paul G. Risser, Department of Botany and Microbiology, University of Oklahoma, Norman, OK 73019
143. D. M. Sharpe, Geography Department, Southern Illinois University, Carbondale, IL 62901
144. G. R. Shepherd, Department of Energy, Rm 4G-052, MS 4G-085, Forestal Building, Washington, DC 20545
145. David Slade, Office of Health and Environmental Research, Department of Energy, Washington, DC 20545
146. P. Sollins, Forest Science Laboratory, 3200 Jefferson Way, Corvallis, OR 97331
147. R. K. Steinhorst, Texas A & M University, Institute of Statistics, College Station, Texas 77843
148. Yu. M. Svirezhev, Computer Center of the USSR, Academy of Sciences, Vavilova Str. 40, Moscow 117333, USSR.

149. Wayne T. Swank, Coweeta Hydrologic Lab, U. S. Forest Service, Franklin, NC 28734
150. G. M. Van Dyne, College of Forestry and Natural Resources, Colorado State University, Fort Collins, CO 80521
151. P. Van Voris, Batelle-Columbus, 505 King Avenue, Columbus, OH 43201
152. Dr. J. Walker, Woodland Ecology, Division of Land-Use Research, CSIRO-Black Mountain, Canberra, A.C.T. 2601, Australia.
153. Richard H. Waring, Department of Forest Science, Oregon State University, Corvallis, OR 97331
154. R. L. Watters, Office of Health and Environmental Research, Department of Energy, Washington, DC 20545
155. J. R. Webster, Department of Biology, Virginia Polytechnic Institute, Blacksburg, VA 24061
156. T. E. Wehrly, Institute of Statistics, Texas A amd M University, College Station, TX 77840
157. R. C. Wiegert, Department of Zoology, University of Georgia, Athens, GA 30602
158. R. W. Wood, Office of Health and Environmental Research, Bldg. GTN, Germantown, MD 20767
159. G. M. Woodwell, Woods Hole Oceanographic Institute, Woods Hole, MA 02543
160. B. P. Zeigler, Department of Mathematics, Weizman Institute, Rehovot, Israel
161. Office of Assistant Manager for Energy Research and Development, DOE-ORO
162-188. Technical Information Center, Oak Ridge, TN 37830