Chapter 20

A Q-Matrix Cookbook

How to Write Only One Program to Calculate the Single-Channel and Macroscopic Predictions for Any Kinetic Mechanism

DAVID COLQUHOUN and ALAN G. HAWKES

1. Introduction

It is clear from the examples in Chapter 18 (this volume) that the algebra involved in kinetic arguments can be quite lengthy, even for simple mechanisms with only three states. For more complex mechanisms it becomes rapidly worse. Furthermore, this complicated algebra would have to be carried out separately for every kinetic mechanism that was of interest. On the other hand, the use of matrix notation allows perfectly general solutions to be written down. Not only are the results general, but they are also compact and simple-looking. They do not result in pages of complicated-looking algebra. For example, once a solution has been obtained for a quantity such as the distribution of the burst length, this result can be applied to *any* kinetic mechanism that is postulated. There is no need for further algebra (or for further programming) when a new mechanism is considered. With a general computer program, all that is needed is to supply the program with a definition of the states and the values of the transition rates for the mechanism you wish to study.

Chapter 18 (this volume) includes a brief introduction to matrix-based theory. More comprehensive treatments are given, for example, by Colquhoun and Hawkes (1982). The paper by Colquhoun and Hawkes (1981) contains some explicit algebraic examples for common mechanisms, but this paper should not be consulted for the underlying theory, which is dealt with more elegantly in the 1982 paper.

The purpose of this chapter is to provide a guide to programming the sort of general matrix results needed, so that you can obtain numerical predictions from them. It is intended as a practical 'cookbook' guide rather than an explanation of the underlying theory. We consider a number of examples based on one particular mechanism. Although these do not exhaust all the possible things you can do, they exhibit the usefulness of the approach, and the computational elements needed are readily adapted to other situations.

Note to the reader: At the author's request this chapter will use British spelling and the abbreviations ms and μs instead of msec and μsec.

DAVID COLQUHOUN • Department of Pharmacology, University College London, London WC1E 6BT, England. ALAN G. HAWKES • European Business Management School, University of Wales Swansea, Swansea SA2 8PP, Wales.

Single-Channel Recording, Second Edition, edited by Bert Sakmann and Erwin Neher. Plenum Press, New York, 1995.

B. Sakmann, E. Neher (eds.), *Single-Channel Recording*,
DOI 10.1007/978-1-4419-1229-9_20, © Springer Science+Business Media, LLC 2009

Within the body of this chapter we assume that the reader is familiar with the basic notation and operations of matrix algebra such as addition, subtraction, and multiplication of matrices, and the determinant of a square matrix; the fourth basic operation, division, is known as *matrix inversion,* the inverse (if it exists) of a matrix **A** being denoted by $\mathbf{A}^{-1}$. These basic operations are described briefly in the Appendix. *If you are not familiar with the basic operations, please read the Appendix before proceeding further.*

Modern computer libraries have excellent routines for matrix manipulation, so it is not too hard to write an entirely general program to produce numerical results for *any* specified mechanism. For example there are the widely available NAG routines in FORTRAN and PASCAL, similar routines in the packages LINPACK and EISPACK, and very good matrix algebra facilities within the languages/packages such as APL, GENSTAT, SPLUS, and MATLAB and the computer algebra packages such as MAPLE, MATHEMATICA, MACSYMA, REDUCE, and DERIVE. General computational guidance and some code can be found in the invaluable book, *Numerical Recipes,* by Press *et al.* (1993).

Because of all this variety, we do not, in general, give detailed computer code in any language, except a few simple examples in languages that are particularly appropriate for this kind of work but that may be less well known. We concentrate instead on discussing the principles involved in one or two of the trickier parts and give numerical examples against which others can check the results obtained from their own programs.

Numerical Solutions and Explicit Solutions

One virtue of writing out the algebra the hard way is that one can see every term in the equations, and therefore one may be able (if the results are not too complex) to get a feeling for how the equations work, e.g., which are the important bits of the equation, and which bits can be neglected. On the other hand, if it gets complex, one may get easily confused and make mistakes.

At the other extreme, it is sometimes possible to solve the equations *numerically* without solving them *algebraically* at all. For example, the macroscopic behaviour of a kinetic system is described by a set of linear first-order differential equations, and standard algorithms exist in all computer libraries (e.g., the Runge–Kutta method) for producing the numerical solution of such sets of equations, given values for the rate constants, etc. The solution comes out as a set of numbers, e.g., the fraction of open channels at each of a set of specified times. Not much work is needed to get the results, but, on the other hand, the results have no generality. Thus, if the value of a rate constant or concentration is changed, the whole calculation must be done again.

In some cases, this sort of numerical solution is all that is possible. For example, if the ligand concentration is varying with time (as during a synaptic current), then the coefficients of the differential equations are not constant, and explicit solutions are usually not possible. If, however, the coefficients *are* constant (e.g., concentrations and membrane potential are constant throughout), then solutions to most problems can be found in the form of a sum of exponential terms. Once this has been done, values can be calculated at any time point with little further effort, and, moreover, the time constants involved in these expressions may be interpreted physically. The problem is that the algebraic expressions for these time constants, and for the coefficients of the exponential terms, are very complicated except in the simplest cases. In mechanisms with three distinct states, for example, the time constants must be found as the solution of a quadratic equation (illustrated in Chapter 18). With four states the time constants are the solutions of a cubic equation. Although a cubic equation can, like a quadratic, be solved explicitly, the results are even more untidy.

With more states than four, the higher-degree polynomials that must be solved for the time constants are not generally solvable in an explicit form. This fact blurs the distinction between numerical and explicit results. The use of matrix notation allows explicit solutions, in the form of sums of exponential terms, to be written compactly and elegantly. But, in order to calculate numbers from these 'explicit' results, it will usually be most convenient (though not necessary) to evaluate the time constants of the exponentials. This process involves solving a polynomial equation for the time constants, as just described, a process that is known, in matrix jargon, as finding the *eigenvalues* of a matrix, as explained below. This polynomial, if higher order than a cubic, will have to be solved by some sort of numerical method (routines for doing this are available in all computer matrix libraries, so you do not have to do this yourself). This does mean, though, that even explicit solutions will involve numerical steps when one wishes to calculate values from them.

2. Basic Notation and a Particular Mechanism

In this section we introduce the basic notation we will use and exemplify it by a particular mechanism that we will use throughout the chapter to give some numerical examples (see also the final section in Chapter 18, this volume). Further notation will be introduced as needed for calculating particular quantities. In general, an ion channel can be considered at any time to be in one of k physical states, which we will label by the integers 1 to k, and the dynamics are governed by the transition rates q_{ij}, which here are interpreted as giving rise to probabilities as follows: if the channel is in state i at time t then, for a small time increment δt,

$$P[\text{channel moves to state } j \text{ during the time interval } (t, t+\delta t)] \simeq q_{ij}\delta t \qquad (1)$$

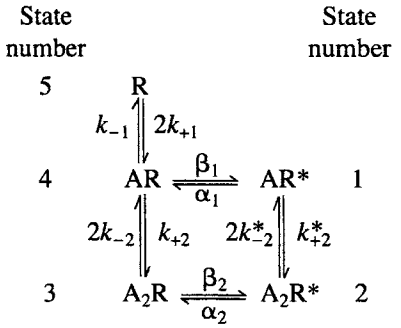
where $\simeq$ means 'approximately equal to'. More precisely, q_{ij} is the limit as δt tends to zero of the ratio of the above probability and the increment δt (see Chapter 18, this volume, Section 1.2). Clearly, this definition makes sense only when i and j are different (i.e., $i \neq j$). The matrix $\mathbf{Q}$ is a square array of values with k rows and k columns. The entry in the ith row and jth column of this array is q_{ij}, and the value (for $i \neq j$) is usually simply the rate constant for transitions between states (these transition rates all have dimensions of reciprocal time, so association rate constants must be multiplied by the ligand concentration, as discussed in Chapter 18 (this volume), to obtain the transition rate). Thus equation 1 is the microscopic probabilistic manifestation of the law of mass action. This defines all the entries in $\mathbf{Q}$ except for those along its diagonal (i.e., the values for $i = j$). These diagonal elements q_{ii} are then chosen so that the sum of the values in each row is zero; thus, q_{ii} is negative, and $-q_{ii}$ represents the total rate at which the channel leaves state i, or $-1/q_{ii}$ is the average duration of a sojourn in state i.

In order to relate the algebra to the experimentally observed channel behaviour, the next step is to classify the k states according to their observable characteristics. The simplest classification is into open states and shut states, and we will use the symbol $\mathcal{A}$ to denote the index set of the open states and $\mathcal{F}$ to denote the set of shut states (examples are given below). If we are interested in burst characteristics, it will be useful to subclassify the shut states as being 'brief' or 'long'. The brief shut states are those that have short lifetimes on average and can be associated with short shut times within a burst; the index set of these will be denoted by $\mathcal{B}$. The remaining 'long' shut states, associated with long gaps between

bursts, will be denoted by $\mathscr{C}$. Then $\mathscr{F} = \mathscr{B} \cup \mathscr{C}$, the union of the two sets; i.e., all shut states are classified as either brief or long. It is convenient to number the states so that $\mathscr{A}$ contains the smallest index numbers, $\mathscr{C}$ the largest and $\mathscr{B}$ the intermediate values. Let the numbers of states in each group be denoted $k_{\mathscr{A}}$, $k_{\mathscr{B}}$, and $k_{\mathscr{C}}$; then $k_{\mathscr{A}} + k_{\mathscr{B}} + k_{\mathscr{C}} = k$.

2.1. A Five-State Mechanism

All the illustrations in this chapter are done with respect to the five-state mechanism shown in equation 110 of Chapter 18 (this volume) and reproduced in equation 2. There are $k_{\mathscr{A}} = 2$ open states, which are numbered 1 and 2 in equation 2, so the set of open states is $\mathscr{A} = \{1,2\}$.



(2)

We suppose that the two states for which agonist molecules are bound, but the channel is shut, are both short-lived, so we number them as states 3 and 4 and classify them as $\mathscr{B}$ states; i.e., we take $\mathscr{B} = \{3,4\}$, with $k_{\mathscr{B}} = 2$. It should be noticed at this point that, since we wish to identify brief observed shut times with sojourns in this set of states, it is actually not good enough to say that the lifetimes of states 3 and 4 are both (on average) brief; we actually require that a sojourn within subset $\mathscr{B}$ should be (on average) brief. The latter does not necessarily follow from the former because, if the rate constants were such that there were many $3 \leftrightarrow 4$ transitions before $\mathscr{B}$ was left, it is possible that a long time could be spent within $\mathscr{B}$ even though states 3 and 4 were both short-lived. Finally the set $\mathscr{C} = \{5\}$ consists of the single shut state, $k_{\mathscr{C}} = 1$, in which no agonist molecules are bound to the channel receptors. This is supposed to have a long lifetime; this will, of course, be true only when the agonist concentration is low, so the calculations refer only to this condition.

The transition rate from state 3 to state 4 is labeled, on diagram 2, as $2k_{-2}$ rather than k_{-2}. This is because two agonist molecules are bound to state 3, and one *or* the other must dissociate to make a transition to state 4. If the dissociation rate constant for a *single* agonist–receptor complex is k_{-2}, and if the two bound molecules both behave in the same way, the fact that *either* may dissociate makes the total transition rate $2k_{-2}$. The rate for a single site, k_{-2}, is known as a *microscopic rate constant,* whereas the net transition rate, $2k_{-2}$, is a *macroscopic rate constant.* Similar remarks apply to the $2 \rightarrow 1$ transition and to the $5 \rightarrow 4$ transition. We shall use microscopic rate constants here because they are the most fundamental physical quantities (but remember the implicit assumption of equivalent binding sites).

2.2. The Q Matrix

The general Q matrix for this mechanism is shown in equation 127 of Chapter 18 (this volume). As in Colquhoun and Hawkes (1982), we take for our calculations the particular values $\beta_1 = 15\ s^{-1}$, $\beta_2 = 15{,}000\ s^{-1}$, $\alpha_1 = 3000\ s^{-1}$, $\alpha_2 = 500\ s^{-1}$, $k_{-1} = k_{-2} = 2000\ s^{-1}$, $k_{+1} = 5 \times 10^7\ M^{-1}\ s^{-1}$, $k_{+2} = k^*_{+2} = 5 \times 10^8\ M^{-1}s^{-1}$. Then the principle of microscopic reversibility implies that we must also have $k^*_{-2} = (1/3)s^{-1}$ (see Chapter 18, this volume, Section 7). Finally, we take the agonist concentration to be $x_A = 100$ nM; thus, for example, $q_{54} = 2k_{+1}x_A = 2 \times (5 \times 10^7\ M^{-1}s^{-1}) \times (100 \times 10^{-9}\ M) = 10\ s^{-1}$, as shown in row 5, column 4 of **Q**, below. When these values are substituted into the general forms given in equation 127 of Chapter 18 (this volume), we get the Q matrix:

$$\mathbf{Q} = \left[\begin{array}{cc|ccc} -3050 & 50 & 0 & 3000 & 0 \\ 0.666667 & -500.666667 & 500 & 0 & 0 \\ \hline 0 & 15000 & -19000 & 4000 & 0 \\ 15 & 0 & 50 & -2065 & 2000 \\ 0 & 0 & 0 & 10 & -10 \end{array}\right] \tag{3}$$

Alternatively, it may be more convenient to work on a millisecond time scale, in which case we multiply the above transition rates by 10^{-3} giving, in ms^{-1},

$$\mathbf{Q} = \left[\begin{array}{cc|ccc} -3.050 & 0.05 & 0 & 3 & 0 \\ 0.000666667 & -0.500666667 & 0.5 & 0 & 0 \\ \hline 0 & 15 & -19 & 4 & 0 \\ 0.015 & 0 & 0.05 & -2.065 & 2 \\ \hline 0 & 0 & 0 & 0.01 & -0.01 \end{array}\right] \tag{4}$$

This has also an advantage in having entries closer in value to 1 than the previous version of **Q**, as some matrix routines can become numerically unstable if the entries are too large or too small. This is the example we will work with. The element q_{21} is actually $(2/3) \times 10^{-3}$; again, because of numerical sensitivity of some matrix operations, one should make this reasonably accurate, and we have expressed it as a decimal to six significant figures.

The matrix in equation 3 has been partitioned into blocks, indicated by the horizontal and vertical hairlines; these divisions correspond to the division of states into the open states (set $\mathcal{A}$) and the shut states (set $\mathcal{F}$). The matrix in equation 4 has also been partitioned into blocks, the divisions in this case correspond to the division of states into the sets $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. This is an example of the general partition of **Q** into the forms

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{\mathcal{A}\mathcal{A}} & \mathbf{Q}_{\mathcal{A}\mathcal{F}} \\ \mathbf{Q}_{\mathcal{F}\mathcal{A}} & \mathbf{Q}_{\mathcal{F}\mathcal{F}} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_{\mathcal{A}\mathcal{A}} & \mathbf{Q}_{\mathcal{A}\mathcal{B}} & \mathbf{Q}_{\mathcal{A}\mathcal{C}} \\ \mathbf{Q}_{\mathcal{B}\mathcal{A}} & \mathbf{Q}_{\mathcal{B}\mathcal{B}} & \mathbf{Q}_{\mathcal{B}\mathcal{C}} \\ \mathbf{Q}_{\mathcal{C}\mathcal{A}} & \mathbf{Q}_{\mathcal{C}\mathcal{B}} & \mathbf{Q}_{\mathcal{C}\mathcal{C}} \end{bmatrix}, \tag{5}$$

where, for example, $\mathbf{Q}_{\mathcal{A}\mathcal{B}}$ indicates a submatrix of the matrix **Q** obtained by selecting the

rows belonging to the index set $\mathcal{A}$ (in this case 1,2) and the columns belonging to the index set $\mathcal{B}$ (in this case 3,4, so $\mathbf{Q}_{\mathcal{AB}}$ has two rows and two columns). Thus, from equation 4 we have in the present case

$$\mathbf{Q}_{\mathcal{AB}} = \begin{bmatrix} 0 & 3 \\ 0.5 & 0 \end{bmatrix} \tag{6}$$

The use of submatrices of this sort is very common in all calculations that concern single-channel properties. In contrast, calculations concerning the average properties of a large number of channels (i.e., relaxation or noise analysis) use the whole **Q** matrix as it stands, as illustrated in Section 4 below. The fact that single-channel observations allow use of a smaller matrix is one way of viewing the reason single channels can provide simpler and more direct inferences than macroscopic observations.

In computer programs for the evaluation of single-channel models, it is, therefore, very commonly required to find a submatrix of **Q**, i.e., in the example above, to move the values in rows 1,2 and columns 3,4 as in equation 4 into rows 1,2 and columns 1,2, as in equation 6. It is easy to write a subroutine in any language that will move the elements in any specified rows and columns into the top left-hand corner in this way. Some languages make this very simple. For example, the computer language APL has very good array-handling properties, and one example is that if one assigns, say, A←1 2 and B←3 4, then the expression Q[A;B] is equivalent to $\mathbf{Q}_{\mathcal{AB}}$.

3. Equilibrium State Occupancies

The fraction of molecules in each state at equilibrium can be obtained from explicit algebraic expressions, which are not difficult to obtain even for complex mechanisms. However, it is very convenient to have them evaluated by the same computer program that does the subsequent, more complex, kinetic calculations. Evaluation of equilibrium occupancies from the **Q** matrix directly raises some problems that may not be obvious, so this problem will be discussed next.

We shall denote the occupancy of state i at time t as $p_i(t)$. Let $\mathbf{p}(t)$ be a row vector with k elements (a $1 \times k$ matrix; see Appendix) that contains these occupancies for each of the k, states.

$$\mathbf{p}(t) = [p_1(t) \quad p_2(t) \quad \cdots \quad p_k(t)] \tag{7}$$

where $p_i(t)$ is the probability that the channel is in state i at time t. The corresponding vector of derivatives is

$$\frac{\mathrm{d}\mathbf{p}(t)}{\mathrm{d}t} = \left[\frac{\mathrm{d}p_1(t)}{\mathrm{d}t} \quad \frac{\mathrm{d}p_2(t)}{\mathrm{d}t} \quad \cdots \quad \frac{\mathrm{d}p_k(t)}{\mathrm{d}t}\right] \tag{8}$$

With this notation, the kinetic equations that describe the system can be written as

$$\frac{\mathrm{d}\mathbf{p}(t)}{\mathrm{d}t} = \mathbf{p}(t)\mathbf{Q} \tag{9}$$

This result follows directly from the law of mass action, which describes the rate of change of the concentration of each reactant (or, equivalently, of the fraction of the system in each state). The result can easily be verified for particular examples by multiplying out the right-hand side. Note, though, that matrix notation has allowed us to write an equation, equation 9, that is correct for *any* mechanism, however complex and with any number of states.

The term *steady state* means that these occupancies (probabilities) do not change with time, and so the derivatives are zero. In the absence of an energy input, the existence of steady state means that the system is at equilibrium (see Chapter 18, this volume, Section 7). After sufficient time ($t \rightarrow \infty$), equilibrium will be reached, and the equilibrium occupancy of state i is denoted $p_i(\infty)$. These equilibrium values are in the equilibrium vector of probabilities $\mathbf{p}(\infty)$, which, from equation 9, satisfies the equation

$$\mathbf{0} = \mathbf{p}(\infty)\mathbf{Q} \tag{10}$$

subject to $\Sigma p_i(\infty) = 1$, because the total probability (or total occupancy) must be 1. Here, the symbol $\mathbf{0}$ represents a matrix with elements that are all zeros. [Mathematical note: for *ergodic* mechanisms, and this will include all reversible mechanisms with at most one closed set of states (*i.e.*, a set that the channel cannot get out of), $\mathbf{p}(t)$ will tend to a unique limit vector $\mathbf{p}(\infty)$ as $t \rightarrow \infty$, independently of initial conditions, and this will be the same as the equilibrium vector $\mathbf{p}$ above. In these circumstances the above equation will have a unique solution.]

Several systematic methods for obtaining the steady-state distribution for any mechanism are available (all, of course, are just standard methods for solving simultaneous equations).

3.1. The Determinant Method

The best-known method is, perhaps, that based on the use of determinants (e.g., Huang, 1979; Colquhoun and Hawkes, 1987). The Appendix gives a brief definition of a determinant, and all matrix program libraries contain routines for calculation of determinants, so it will never be necessary to program this oneself. The procedure is as follows. To find the value for $p_i(\infty)$, cross out the ith row and the ith column of $\mathbf{Q}$. Then calculate the determinant of the matrix that remains (which now has $k - 1$ rows and $k - 1$ columns). Call this determinant d_i. Then $p_i(\infty)$ can be calculated as d_i divided by the sum of all k values of d. In general

$$p_i(\infty) = d_i / \sum_{j=1}^{k} d_j \tag{11}$$

3.2. The Matrix Method

In order to obtain numerical results from a computer program, it is generally more convenient to use matrix methods than to use determinants. This is, however, not quite as straightforward as it seems, essentially because the number of unknowns is one less than the number of equations, so one equation is superfluous. This is reflected in the fact that $\mathbf{Q}$ has a determinant of zero (because the rows each add up to 0) and so cannot be inverted (see Appendix). There are two ways around this problem. One is to define a modified

(*reduced*) **Q** matrix that has only $k - 1$ rows and columns rather than k (where k is the number of states in which the system can exist). This gets rid of the superfluity and thus allows a straightforward solution. The other, and perhaps more straightforward, solution is to use a trick (described below) to solve the equations directly. Both methods will be described.

3.2.1. The Reduced Q-Matrix Method

The procedure is to subtract the elements in the bottom row of **Q** from each of the other rows and to omit the last column. Thus, if we denote the reduced **Q** matrix as **R,** with elements r_{ij}, then

$$r_{ij} = q_{ij} - q_{kj} \quad \text{for} \quad 1 \leq i, \quad j \leq k - 1 \tag{12}$$

Thus, for the **Q** matrix in equation 4, the reduced version would be

$$\mathbf{R} = \begin{bmatrix} -3.050 & 0.05 & 0 & 2.99 \\ 0.000666667 & -0.500666667 & 0.5 & -0.01 \\ 0 & 15 & -19 & 3.99 \\ 0.015 & 0 & 0.05 & -2.075 \end{bmatrix} \tag{13}$$

Define also a reduced version of the row vector that contains the equilibrium probabilities, with the last value, $p_k(\infty)$, omitted. The last value can be found at the end from the fact that the probabilities must add to 1. Call this vector

$$\mathbf{p}(\infty)' = [p_1(\infty) \quad p_2(\infty) \quad \cdots \quad p_{k-1}(\infty)]$$

Finally, define a row vector, $\mathbf{r} = [q_{k1} \quad q_{k2} \ldots q_{k,k-1}]$, that contains the first $k - 1$ elements of the bottom row of **Q.** In this example, from equation 4, we have

$$\mathbf{r} = [0 \quad 0 \quad 0 \quad 0.01] \tag{14}$$

The equations $\mathbf{p}(\infty)\mathbf{Q} = \mathbf{0}$ imply $\mathbf{p}(\infty)'\mathbf{R} + \mathbf{r} = \mathbf{0}$. Because **R,** unlike **Q,** can be inverted (it is not singular), the solution can be written as

$$\mathbf{p}(\infty)' = -\mathbf{r}\mathbf{R}^{-1} \tag{15}$$

All computer matrix libraries contain procedures for matrix inversion so it is not necessary to program this operation oneself (indeed is is not desirable, because enormous specialist effort has gone into writing algorithms that will give numerically accurate results).

3.2.2. Solution of $\mathbf{p}(\infty)\mathbf{Q} = \mathbf{0}$ Directly

This can be done by adding a unit column (all values are 1) on to the right-hand end of **Q** to produce a matrix with k rows and $k + 1$ columns. Call this matrix **S.** Also define

a row vector, **u** say, containing k values all equal to 1. For example, from the **Q** matrix in equation 4, let **u** = [1 1 1 1 1] and

$$\mathbf{S} = \begin{bmatrix} -3.050 & 0.05 & 0 & 3 & 0 & 1 \\ 0.000666667 & -0.500666667 & 0.5 & 0 & 0 & 1 \\ 0 & 15 & -19 & 4 & 0 & 1 \\ 0.015 & 0 & 0.05 & -2.065 & 2 & 1 \\ 0 & 0 & 0 & 0.01 & -0.01 & 1 \end{bmatrix} \tag{16}$$

The solution for the equilibrium occupancies can then be found, for any mechanism, as

$$\mathbf{p}(\infty) = \mathbf{u}(\mathbf{S}\mathbf{S}^{\mathrm{T}})^{-1} \tag{17}$$

In this result, $\mathbf{S}^{\mathrm{T}}$ represents the transpose of **S** (see Appendix), so $\mathbf{S}\mathbf{S}^{\mathrm{T}}$ is a matrix with k rows and k columns, but, unlike **Q**, it is not singular so it can be inverted to get the required solution. Hawkes and Sykes (1990) discuss this solution and show that in APL it is computed remarkably simply (see Appendix 1).

With any of these three methods, the solution for our example matrix (equation 4) is, to four significant figures,

$$\mathbf{p}(\infty) = (0.00002483 \quad 0.001862 \quad 0.00006207 \quad 0.004965 \quad 0.9931). \tag{18}$$

Under these conditions the channel is shut most of the time; it is open only 0.189% of the time, and for 99.3% of the time it is in the unoccupied state (state 5).

4. Relaxation to Equilibrium

4.1. General Solutions for the Rate of Approach to Equilibrium

In any problem that involves the average behaviour of a large number of channels, the problem is to find how occupancies change with time. This description encompasses all macroscopic voltage-jump and concentration-jump experiments, for example. The problem, then, is to solve equation 9 for $\mathbf{p}(t)$; this vector contains the occupancy of each state at time t. We expect, under the conditions mentioned at the start, that the time course of these occupancies will be described by the sum of $k - 1$ exponential terms (Colquhoun and Hawkes, 1977).

4.1.1. Initial Occupancies at Equilibrium

The aim, in a jump experiment, is to keep the membrane potential and ligand concentrations constant at all times (except for the actual moment of the jump). The approach to equilibrium after a jump should therefore be dictated by the **Q** matrix calculated for the conditions (the potential and concentrations) that exist *after* the jump. In order to calculate this time course, we need to know the occupancy of each state at the moment ($t = 0$, say) when the jump was applied, i.e., the initial occupancies, $\mathbf{p}(0)$. If the system was at equilibrium

before the jump was imposed, then these initial occupancies will simply be the equilibrium occupancies, calculated as described in Section 3 from the **Q** matrix that describes the conditions *before* the jump.

4.1.2. Initial Occupancies Not at Equilibrium

In a more complex case we might wish to calculate the time course of the response to a brief pulse of membrane potential or concentration. This will involve two separate calculations, one for the onset of the response from the moment ($t = 0$) that the pulse starts and another for the 'offset' of the response after the pulse ends. In this case it is reasonable to suppose that the system has equilibrated before the pulse is applied, so the initial occupancies for calculating the onset of the response can be found as above. However, if the pulse is brief (duration t_p say), there will not be time for the system to come to equilibrium before the end of the pulse. The initial occupancies for calculating the 'offset' time course will simply be the occupancies, $\mathbf{p}(t_p)$, that were found from the onset calculation to obtain at the moment $t = t_p$ when the pulse ends, and the calculation can then be completed using the **Q** matrix appropriate to the conditions after the end of the pulse.

Formally, the solution of differential equation 9 is just

$$\mathbf{p}(t) = \mathbf{p}(0)e^{\mathbf{Q}t} \tag{19}$$

where $\mathbf{p}(0)$ contains the occupancy probabilities at $t = 0$. This result, despite being completely general for any mechanism, looks no more complicated than its scalar (nonmatrix) equivalent, which would describe only the simplest two-state shut ↔ open reaction. This astonishingly simple result is, in a sense, all that there is to be said about calculating the time course on the basis of some specified reaction scheme. Needless to say, though, there is a bit more to be said. In particular, it may not be at all obvious what $e^{\mathbf{Q}t}$ means. In this expression, e represents the usual (scalar) constant (2.71828 . . . , the base of natural logarithms), but the exponent $\mathbf{Q}t$ is a matrix! The exponential of a matrix is an unfamiliar object even to many mathematicians. It is to some extent reassuring to find that it is defined by the usual power series

$$e^{\mathbf{Q}t} = \mathbf{I} + \mathbf{Q}t + \frac{(\mathbf{Q}t)^2}{2!} + \frac{(\mathbf{Q}t)^3}{3!} + \frac{(\mathbf{Q}t)^4}{4!} + \cdots \tag{20}$$

where **I** is the identity matrix (see Appendix), with matrices and their powers replacing the usual scalar terms with which we are familiar. It is immediately clear from this that, since **Q** is a $k \times k$ matrix, $e^{\mathbf{Q}t}$ is itself also a $k \times k$ matrix. The (infinite) series in equation 20 involves nothing more complex than multiplying and adding matrices, so it can easily be evaluated, stopping after a finite number of terms of course. However, use of this expansion, though sometimes satisfactory, is not generally the fastest or the most accurate way of evaluating the exponential numerically. Furthermore, its use does not generate the $k - 1$ exponential components that an experienced experimenter expects to see.

4.2. Evaluation of p(*t*) as a Sum of Exponential Components

The trick needed to accomplish this is a beautifully elegant technique called the *spectral expansion* of a matrix. This is critical to most of the results in this chapter, and the problem

is therefore looked at in more detail in Section 9. For the moment, we merely state that $e^{\mathbf{Q}t}$ can be written in the form

$$e^{\mathbf{Q}t} = \sum_{i=1}^{k} \mathbf{A}_i \exp(-\lambda_i t), \tag{21}$$

where λ_i are *eigenvalues* of the matrix $-\mathbf{Q}$, k in number (equal to the number of states), and the $\mathbf{A}_i$ are a set of square matrices derived from $\mathbf{Q}$ and known as the spectral matrices of $\mathbf{Q}$. How these quantities are obtained from $\mathbf{Q}$ will be discussed in Section 9, below. Given this relationship, our solution can be written in the form

$$\mathbf{p}(t) = \mathbf{p}(0)e^{\mathbf{Q}t} = \mathbf{p}(0) \sum_{i=1}^{k} \mathbf{A}_i \exp(-\lambda_i t) \tag{22}$$

We now have the exponentials in the familiar scalar form. However there are k terms, but we are expecting only $k - 1$ exponentials. The explanation of this is that $\mathbf{Q}$ is singular, so that one of the eigenvalues, say λ_1, is zero; thus, $\exp(-\lambda_1 t) = \exp(0) = 1$, regardless of t. The other eigenvalues can all be shown to be real and positive, so that $\exp(-\lambda_i t)$ tends to zero as $t \to \infty$ for all eigenvalues except λ_1. Thus, the limit, letting $t \to \infty$, of the above equation shows that

$$\mathbf{p}(\infty) = \mathbf{p}(0)\mathbf{A}_1 \tag{23}$$

and so

$$\mathbf{p}(t) = \mathbf{p}(\infty) + \mathbf{p}(0) \sum_{i=2}^{k} \mathbf{A}_i \exp(-\lambda_i t)$$

or, in terms of time constants,

$$\mathbf{p}(t) = \mathbf{p}(\infty) + \mathbf{p}(0) \sum_{i=2}^{k} \mathbf{A}_i \exp(-t/\tau_i) \tag{24}$$

where $\tau_i = 1/\lambda_i$ is the time constant of the ith component. The approach to the equilibrium is thus a mixture of $k - 1$ exponential components. Notice that the *same* set of $k - 1$ time constants describe the time course of change for all the states; all that differs from one state to another is the amplitude of the components, i.e., the size (and sign) of the coefficients that multiply each exponential term.

4.3. Expressing the Coefficients as Scalars

In the result in equation 24, the exponential terms $\exp(-t/\tau_i)$ are ordinary scalars, but the coefficients of these exponential terms are still in matrix form. Both sides of equation 24 are $1 \times k$ matrices (vectors), the ith entry being $p_i(t)$. The last term, with the summation sign, involves the products of $\mathbf{p}(0)$, which is $1 \times k$, with $\mathbf{A}_i$, which is $k \times k$, and this product

is $1 \times k$. If the matrices on the right-hand side are multiplied out, the result can be put into an entirely scalar form. Thus, the time course of the occupancy of the jth state, $p_j(t)$, can be written as

$$p_j(t) = p_j(\infty) + w_{2j}e^{-t/\tau_2} + w_{3j}e^{-t/\tau_3} + \cdots + w_{kj}e^{-t/\tau_k} \tag{25}$$

where w_{ij} ($i = 2, 3, \ldots, k$) are the $k - 1$ coefficients that define the amplitudes of the components for the jth state (w_{ij} is the coefficient for the component with time constant τ_i). The result in equation 25 can be written more compactly as

$$p_j(t) = p_j(\infty) + \sum_{i=2}^{i=k} w_{ij}e^{-t/\tau_i} \tag{26}$$

These coefficients are given by

$$w_{ij} = \sum_{r=1}^{r=k} p_r(0)a_{rj}^{(i)} \tag{27}$$

where $a_{rf}^{(i)}$ denotes the value in the rth row and jth column of $\mathbf{A}_i$. Thus, once the λ_i and the $\mathbf{A}_i$ have been found (see below), the occupancy of any state at any time can easily be calculated.

4.4. The Current through a Channel

Calculations of the sort outlined above will most commonly be aimed at calculating an observable quantity such as the time course of the current through ion channels. There may, in general, be more than one open state through which current can pass: in the notation introduced earlier there are $k_{\mathcal{A}}$ open states. These states may not all have the same conductance. Thus, if γ_i is the conductance of the ith open state, the expected current at time t, for one channel, will be

$$(V - V_{\text{rev}})[\gamma_1 p_1(t) + \gamma_2 p_2(t) + \cdots + \gamma_{k_{\mathcal{A}}} p_{k_{\mathcal{A}}}(t)]$$

i.e., just add up the probabilities of all the open states, each multiplied by the appropriate conductance, and then multiply that sum by the effective voltage across the membrane (the difference between the membrane potential, V, and the reversal potential, V_{rev}, at which no current flows). In macroscopic studies there are typically many channels, N say, so the observed current should be N times the above expected current. Using equation 24 we get, in scalar form, the current as the sum of $k - 1$ exponential terms,

$$I(t) = I(\infty) + \sum_{i=2}^{k} b_i \exp(-t/\tau_i), \tag{28}$$

where the (scalar) coefficients of each exponential are

$$b_i = N(V - V_{\text{rev}})\mathbf{p}(0)\mathbf{A}_i\mathbf{v} \tag{29}$$

In this result the column vector **v** has k elements of which the first $k_{\mathcal{A}}$ are the conductances γ_i, and the rest are zeroes (the conductance of the shut states). Multiplying by this is equivalent to adding the probabilities as described above. Multiplying out the matrices in equation 29 shows that the coefficients can be expressed entirely in terms of scalar quantities, thus:

$$b_i = N(V - V_{\text{rev}}) \sum_{r=1}^{r=k} \sum_{j=1}^{j=k_{\mathcal{A}}} p_r(0)\gamma_j a_{rj}^{(i)} \tag{30}$$

Without the factor N, equations 28–30 also give the time course of the current you would expect to see if you averaged several repetitions of the step experiment with a single channel (see Chapter 18, this volume).

4.5. Numerical Results

Using the methods discussed in Section 9, we find the eigenvalues of $-\mathbf{Q}$ are

$$\begin{cases} \lambda_1 = 0 \text{ ms}^{-1} \\ \lambda_2 = 0.1018 \text{ ms}^{-1} \\ \lambda_3 = 2.022 \text{ ms}^{-1} \\ \lambda_4 = 3.094 \text{ ms}^{-1} \\ \lambda_5 = 19.41 \text{ ms}^{-1} \end{cases} \tag{31}$$

and these correspond to time constants $\tau_i = 1/\lambda_i$ (apart from the case $\lambda_1 = 0$) of

$$\begin{cases} \tau_2 = 9.821 \text{ ms} \\ \tau_3 = 0.4945 \text{ ms} \\ \tau_4 = 0.3233 \text{ ms} \\ \tau_5 = 51.52 \ \mu\text{s} \end{cases} \tag{32}$$

The k spectral matrices are as follows:

$$\mathbf{A}_1 = \begin{bmatrix} 0.00002483 & 0.001862 & 0.00006207 & 0.004965 & 0.9931 \\ 0.00002483 & 0.001862 & 0.00006207 & 0.004965 & 0.9931 \\ 0.00002483 & 0.001862 & 0.00006207 & 0.004965 & 0.9931 \\ 0.00002483 & 0.001862 & 0.00006207 & 0.004965 & 0.9931 \\ 0.00002483 & 0.001862 & 0.00006207 & 0.004965 & 0.9931 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} 1.670\text{E-}5 & 0.03497 & 0.0009297 & 0.001728 & -0.03764 \\ 4.662\text{E-}4 & 0.9763 & 0.02596 & 0.04825 & -1.051 \\ 3.719\text{E-}4 & 0.7787 & 0.0207 & 0.03848 & -0.8383 \\ 8.641\text{E-}6 & 0.01809 & 0.0004811 & 0.0008941 & -0.01948 \\ -9.411\text{E-}7 & -0.001971 & -0.00005239 & -0.00009738 & 0.002121 \end{bmatrix}$$

$$\mathbf{A}_3 = \begin{bmatrix} 0.04051 & -0.06356 & 0.006321 & 2.779 & -2.762 \\ -0.0008475 & 0.00133 & -0.000132 & -0.05813 & 0.05778 \\ 0.002525 & -0.003961 & 0.0003934 & 0.1732 & -0.1721 \\ 0.01389 & -0.0218 & 0.002165 & 0.9531 & -0.9473 \\ -0.00006905 & 0.0001083 & -0.00001076 & -0.004737 & 0.004708 \end{bmatrix}$$

$$\mathbf{A}_4 = \begin{bmatrix} 0.9594 & 0.0272 & -7.9E\text{-}3 & -2.785 & 1.807 \\ 0.0003627 & 0.00001028 & -2.987E\text{-}6 & -0.001053 & 0.000683 \\ -0.00316 & -0.0000896 & 2.602E\text{-}5 & 0.0009174 & -0.00595 \\ -0.01393 & -0.0003949 & 1.147E\text{-}4 & 0.04043 & -0.02622 \\ 0.00004517 & 0.000001281 & -3.719E\text{-}7 & -0.0001311 & 0.00008504 \end{bmatrix}$$

$$\mathbf{A}_5 = \begin{bmatrix} 1.455E\text{-}7 & -0.0004734 & 0.0005968 & -1.377E\text{-}4 & 1.419E\text{-}5 \\ -6.312E\text{-}6 & 0.02053 & -0.02588 & 5.971E\text{-}3 & -6.157E\text{-}4 \\ 2.387E\text{-}4 & -0.7765 & 0.9788 & -2.258E\text{-}1 & 2.328E\text{-}2 \\ -6.883E\text{-}7 & 0.002239 & -0.002823 & 6.512E\text{-}4 & -6.714E\text{-}5 \\ 3.548E\text{-}10 & -0.000001154 & 0.000001455 & -3.357E\text{-}7 & 3.461E\text{-}8 \end{bmatrix} \quad (33)$$

Note that every row of the matrix $\mathbf{A}_1$ is the same as the equilibrium vector shown in equation 18, so its columns consist of identical numbers. This is why we get this limit as $t \to \infty$ regardless of the initial probability vector $\mathbf{p}(0)$.

There are just two open states, with conductances γ_1 and γ_2 say, so we have

$$\mathbf{v} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (34)$$

4.6. Example of a Concentration Jump

Suppose the membrane potential is $V = -100$ mV, with a reversal potential $V_{rev} = 0$ mV, and that the conductances of the two open states are $\gamma_1 = 40$ pS and $\gamma_2 = 50$ pS. There was zero drug concentration before time $t = 0$, so, at equilibrium, all channels are in the unoccupied shut state (state 5). The initial vector is therefore

$$\mathbf{p}(0) = (0 \quad 0 \quad 0 \quad 0 \quad 1)$$

At $t = 0$, the concentration is suddenly increased to 100 nM (the concentration used to calculate the $\mathbf{Q}$ matrix in equation 4 and its eigenvalues and spectral matrices in equations

31–33. The current will then rise from $I(0) = 0$ at $t = 0$ toward its equilibrium value for an agonist concentration of 100 nM. For $N = 1$ channel, this is

$$I(\infty) = (V - V_{rev})[\gamma_1 p_1(\infty) + \gamma_2 p_2(\infty)]$$

$$= -9.4095 \times 10^{-3} \text{ pA}$$

The current will follow a time course described by the sum of four exponential terms, as in equation 28, with time constants as specified in equation 32. The coefficients, b_i, for each of these terms can be calculated by equation 29 or 30, and the values (in picoamperes) are $b_2 = 9.8563 \times 10^{-3}$ pA, $b_3 = -0.2655 \times 10^{-3}$ pA, $b_4 = -0.1871 \times 10^{-3}$ pA, and $b_5 = 0.005770 \times 10^{-3}$ pA. By far the largest component is the second, that with $\tau_2 = 9.821$ ms (its amplitude, b_2, is 37 times greater than that of the next largest component), so the relaxation is quite close to being a single exponential with this time constant. Notice that the sum of the four b_i values comes to $+9.409 \times 10^{-3}$ pA, thus ensuring that the current at $t = 0$ is indeed zero.

5. Distribution of Open Times and Shut Times

The theory enabling prediction of the distributions of open times and shut times in a single channel record is described in some detail in Colquhoun and Hawkes (1982). We will not go into the theory here but merely quote results (the form CH82 followed by a number will be used to refer to equations from that paper) and comment on computational aspects.

5.1. Distribution of Open Times

The distribution of all open times is given by (CH82-3.64) as

$$f(t) = \boldsymbol{\phi}_o \exp(\mathbf{Q}_{\mathcal{A}\mathcal{A}}t)(-\mathbf{Q}_{\mathcal{A}\mathcal{A}})\mathbf{u}_{\mathcal{A}} \tag{35}$$

The beauty of this result is that, despite being quite general, it looks (apart from an initial and final vector) very much like the simple exponential distribution, $f(t) = \lambda \exp(-\lambda t)$, with $-\mathbf{Q}_{\mathcal{A}\mathcal{A}}$ in place of λ. In equation 35, $\boldsymbol{\phi}_o$ is a row vector ($1 \times k_{\mathcal{A}}$) containing the probabilities of starting an open time in each of the $k_{\mathcal{A}}$ open states; $\mathbf{Q}_{\mathcal{A}\mathcal{A}}$ is a $k_{\mathcal{A}} \times k_{\mathcal{A}}$ matrix, the subsection of the $\mathbf{Q}$ matrix relating to the open states only (see Section 2), and $\mathbf{u}_{\mathcal{A}}$ is a column vector ($k_{\mathcal{A}} \times 1$) whose elements are all 1 (this has the effect of summing over the $\mathcal{A}$ states—see Appendix 1, equation A4). Thus, the result in equation 35 is scalar. To evaluate it we need only have routines to multiply matrices and a way of evaluating $\exp(\mathbf{Q}_{\mathcal{A}\mathcal{A}}t)$. The latter can be found in exactly the same way as used for finding $\exp(\mathbf{Q}t)$, as outlined in Section 4 and specified in detail in Section 9. The only differences are that (1) this time we have a smaller matrix; i.e., from equation 4,

$$\mathbf{Q}_{\mathcal{A}\mathcal{A}} = \begin{bmatrix} -3.050 & 0.05 \\ 0.000666667 & -0.500666667 \end{bmatrix} \tag{36}$$

and (2) unlike $\mathbf{Q}$, this time the matrix is unlikely to be singular, so none of the eigenvalues will be zero. In this example, the eigenvalues of $-\mathbf{Q}_{\mathcal{A}\mathcal{A}}$ are

$$\lambda_1 = 0.500654 \text{ ms}^{-1}, \qquad \lambda_2 = 3.05001 \text{ ms}^{-1} \tag{37}$$

so the time constants are

$$\tau_1 = 1.99739 \text{ ms}, \qquad \tau_2 = 0.327867 \text{ ms}$$

The distribution of open times will have two ($k_{\mathcal{A}}$) exponential components with these time constants. To get the distribution in the form of a sum of (scalar) exponentials, we again, as in Section 4, use the spectral expansion trick. By direct analogy with equation 21, we can write

$$\exp(\mathbf{Q}_{\mathcal{A}\mathcal{A}}t) = \sum_{i=1}^{k_{\mathcal{A}}} \mathbf{A}_i \exp(-\lambda_i t) \tag{38}$$

where $\mathbf{A}_i$ now represents the $k_{\mathcal{A}}$ spectral matrices (each $k_{\mathcal{A}} \times k_{\mathcal{A}}$) of $-\mathbf{Q}_{\mathcal{A}\mathcal{A}}$, and λ_i are the eigenvalues of $-\mathbf{Q}_{\mathcal{A}\mathcal{A}}$, already given in equation 37. In the present example, the spectral matrices are

$$\mathbf{A}_1 = \begin{bmatrix} 5.1288 \times 10^{-6} & 0.019613 \\ 2.6150 \times 10^{-4} & 0.999995 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} 0.999995 & -0.019613 \\ -2.6150 \times 10^{-4} & 5.1288 \times 10^{-6} \end{bmatrix}. \tag{39}$$

To complete the calculation, we need to find the relative areas of these two components. In general, a mixture of exponential densities can be represented as

$$f(t) = \Sigma a_i (1/\tau_i) \exp(-t/\tau_i) \tag{40}$$

the sum running over the number of components (here $i = 1$ to $k_{\mathcal{A}}$). Here a_i represents the area of the ith component (the total area being 1). From equations 35 and 38, the areas are given by

$$a_i = -\tau_i \boldsymbol{\phi}_0 \mathbf{A}_i \mathbf{Q}_{\mathcal{A}\mathcal{A}} \mathbf{u}_{\mathcal{A}} \tag{41}$$

We now have everything needed to evaluate these areas, apart from the initial vector, $\boldsymbol{\phi}_0$. This is given (CH82-3.63) as

$$\boldsymbol{\phi}_0 = \mathbf{p}_{\mathcal{F}}(\infty)\mathbf{Q}_{\mathcal{F}\mathcal{A}} / \mathbf{p}_{\mathcal{F}}(\infty)\mathbf{Q}_{\mathcal{F}\mathcal{A}}\mathbf{u}_{\mathcal{A}} \tag{42}$$

In this expression, $\mathbf{p}_{\mathcal{F}}(\infty)$ represents the part of the equilibrium occupancies found in equation 18 for the shut states only. The equilibrium vector was

$$\mathbf{p}(\infty) = (0.00002483 \quad 0.001862 \mid 0.00006207 \quad 0.004965 \quad 0.9931) \tag{43}$$

but a vertical line has been added that partitions $\mathbf{p}(\infty)$ into $\mathbf{p}_{\mathcal{A}}(\infty)$ (the first two elements, for the open states) and $\mathbf{p}_{\mathcal{F}}(\infty)$ (the last three elements for the shut states). Thus

$$\mathbf{p}_{\mathcal{F}}(\infty) = (0.00006207 \quad 0.004965 \quad 0.9931) \tag{44}$$

and when this is postmultiplied by $\mathbf{Q}_{\mathcal{F}\mathcal{A}}$, which, from equation 4, is the $k_{\mathcal{F}} \times k_{\mathcal{A}}$ matrix

$$\mathbf{Q}_{\mathcal{F}\mathcal{A}} = \begin{bmatrix} 0 & 15 \\ 0.015 & 0 \\ 0 & 0 \end{bmatrix} \tag{45}$$

the result is a $1 \times k_{\mathcal{A}}$ vector that forms the numerator of $\boldsymbol{\phi}_0$ in equation 42. The denominator, $\mathbf{p}_{\mathcal{F}}(\infty)\mathbf{Q}_{\mathcal{F}\mathcal{A}}\mathbf{u}_{\mathcal{A}}$, is a simple scalar, the sum of the elements in the numerator, which ensures that the elements of $\boldsymbol{\phi}_0$ add up to 1. When these are multiplied out, the result is

$$\boldsymbol{\phi}_0 = (0.07407 \quad 0.92593) \tag{46}$$

Thus, any individual opening has a 7.4% chance of starting in open state 1 and a 92.6% chance of starting in open state 2. The areas can now be found from equation 41 and come to $a_1 = 0.9276$ and $a_2 = 0.07238$. The final probability density function for open times, in the form given in equation 40, is therefore

$$f(t) = 0.9276(1/1.997)e^{-t/1.997} + 0.07238(1/0.3279)e^{-t/0.3279} \tag{47}$$

The slower component, $\tau_1 = 1.99739$ ms, predominates, having 92.8% of the area.

When channel openings can be divided into bursts, there are many other open-time distributions of potential interest, e.g., the distribution of the first opening of a burst of openings or of all openings in bursts with one opening, etc. All of them have the basic form for the probability density:

$$f(t) = \boldsymbol{\phi} \exp(\mathbf{Q}_{\mathcal{A}\mathcal{A}}t)\mathbf{c} \tag{48}$$

where $\boldsymbol{\phi}$ is a suitable row vector containing the probabilities of starting an open time in each of the $k_{\mathcal{A}}$ open states, and $\mathbf{c}$ is some appropriate column vector (expressions are given by Colquhoun and Hawkes, 1982, for various cases). They all involve $\exp(\mathbf{Q}_{\mathcal{A}\mathcal{A}}t)$, so they all have $k_{\mathcal{A}}$ components with the same time constants as before. Only $\boldsymbol{\phi}$ and $\mathbf{c}$, and hence the relative areas, differ from one sort of distribution to another.

5.2. Distribution of Shut Times

The distribution of all shut times can be found in exactly the same way as just described for open times. In fact, all that has to be done is to interchange $\mathcal{A}$ and $\mathcal{F}$ in the equations

already given. The distribution is thus

$$f(t) = \boldsymbol{\phi}_s \exp(\mathbf{Q}_{\mathcal{FF}}t)(-\mathbf{Q}_{\mathcal{FF}})\mathbf{u}_{\mathcal{F}} \tag{49}$$

where the initial vector, which gives the probabilities that a shut period starts in each of the $k_{\mathcal{F}}$ shut states, is

$$\boldsymbol{\phi}_s = \mathbf{p}_{\mathcal{A}}(\infty)\mathbf{Q}_{\mathcal{AF}}/\mathbf{p}_{\mathcal{A}}(\infty)\mathbf{Q}_{\mathcal{AF}}\mathbf{u}_{\mathcal{F}} \tag{50}$$

To complete the calculations, we need the 'shut' portion of the $\mathbf{Q}$ matrix; i.e., from equation 4,

$$\mathbf{Q}_{\mathcal{FF}} = \begin{bmatrix} -19 & 4 & 0 \\ 0.05 & -2.065 & 2 \\ 0 & 0.01 & -0.01 \end{bmatrix} \tag{51}$$

and

$$\mathbf{Q}_{\mathcal{AF}} = \begin{bmatrix} 0 & 3 & 0 \\ 0.5 & 0 & 0 \end{bmatrix} \tag{52}$$

with, from equation 43,

$$\mathbf{p}_{\mathcal{A}}(\infty) = (0.00002483 \quad 0.001862) \tag{53}$$

The calculations proceed in exactly the same way as for open times. We find the $k_{\mathcal{F}}$ (=3 in this case) eigenvalues and spectral matrices for $\mathbf{Q}_{\mathcal{FF}}$ and use them to express the shut-time distribution in the form of $k_{\mathcal{F}}$ exponential components, as in equation 40. The relative areas of the components are (cf. equation 41)

$$a_i = -\tau_i \boldsymbol{\phi}_s \mathbf{A}_i \mathbf{Q}_{\mathcal{FF}} \mathbf{u}_{\mathcal{F}} \tag{54}$$

where the time constants, τ_i, are now those for the shut-time distribution. The results are as follows. The eigenvalues of $-\mathbf{Q}_{\mathcal{FF}}$ are

$$\lambda_1 = 0.263895 \times 10^{-3}\ \text{ms}^{-1}, \qquad \lambda_2 = 2.06293\ \text{ms}^{-1}, \qquad \text{and} \quad \lambda_3 = 19.0118\ \text{ms}^{-1};$$

the corresponding time constants, $\tau_i = 1/\lambda_i$, are

$$\tau_1 = 3789.4\ \text{ms}; \qquad \tau_2 = 0.484747\ \text{ms}; \qquad \tau_3 = 52.5989\ \mu\text{s}. \tag{55}$$

The relative areas of these components, from equation 54, are

$$a_1 = 0.261946; \qquad a_2 = 0.00836704; \qquad a_3 = 0.729687 \tag{56}$$

Thus, most shut times are either very short, 52.6 μs (73%), or very long, 3789 ms (26%).

The intermediate quantities needed to obtain the above areas are $\boldsymbol{\phi}_s$, given by equation 50 and the spectral matrices $\mathbf{A}_i$. The results are

$$\boldsymbol{\phi}_s = [0.92593 \quad 0.07407 \quad 0] \tag{57}$$

$$\mathbf{A}_1 = \begin{bmatrix} 2.613\text{E-}6 & 9.931\text{E-}4 & 0.2040 \\ 1.241\text{E-}5 & 4.717\text{E-}3 & 0.9690 \\ 1.275\text{E-}5 & 4.845\text{E-}3 & 0.9953 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} 6.934\text{E-}4 & 0.2349 & -0.2288 \\ 2.936\text{E-}3 & 0.9946 & -0.9689 \\ -1.430\text{E-}5 & -4.845\text{E-}3 & 4.720\text{E-}3 \end{bmatrix}$$

$$\mathbf{A}_3 = \begin{bmatrix} 0.9993 & -0.2359 & 2.483\text{E-}2 \\ -2.949\text{E-}3 & 6.960\text{E-}4 & -7.326\text{E-}5 \\ 1.552\text{E-}6 & -3.663\text{E-}7 & 3.855\text{E-}8 \end{bmatrix} \tag{58}$$

6. Distribution of the Number of Openings per Burst

It is a very common observation that channel openings occur in *bursts* of several openings in quick succession rather than singly. This will be the case when, as in our example, the shut-time distribution contains some components that are very brief (short shuttings within a burst) and some that are very long (shut times between bursts). This is the case in our numerical example, as found in equations 55 and 56.

A burst of openings must obviously contain at least one opening. In general, it may contain r openings separated by $r - 1$ brief shuttings, where r is random. For the present purposes we define a 'shut time within a burst' as a shut time spent entirely within the short-lived shut states, set $\mathcal{B}$. There are, in this case, $k_{\mathcal{B}} = 2$ such states, states 3 and 4 (see equations 4–6). An entry into the long-lived shut state (state 5) will produce a 'shut time between bursts'. There is only one way out of state 5, with rate $q_{54} = 0.01$ ms^{-1}, so the mean lifetime of a single sojourn in state 5 is 1/0.01 = 100 ms. Notice that this is much shorter than the 3789-ms component of shut times; this is because the channel will oscillate several times between the shut states 5, 4, and 3 between one burst and the next and is likely to visit state 5 several times before the next opening occurs (i.e., before the next burst starts). A channel in state 4 is much more likely to return to state 5 (rate 2 ms^{-1}) than either to proceed to state 3 (rate = 0.05 ms^{-1}) or to open to state 1 (rate = 0.015 ms^{-1}).

In order to analyze the burst structure we need two new matrices, denoted $\mathbf{G}_{\mathcal{AB}}$ and $\mathbf{G}_{\mathcal{BA}}$, which are defined as

$$\mathbf{G}_{\mathcal{AB}} = -\mathbf{Q}_{\mathcal{AA}}^{-1}\mathbf{Q}_{\mathcal{AB}} \qquad \mathbf{G}_{\mathcal{BA}} = -\mathbf{Q}_{\mathcal{BB}}^{-1}\mathbf{Q}_{\mathcal{BA}} \tag{59}$$

The interpretation of $\mathbf{G}_{\mathcal{AB}}$ is as follows. It is a $k_{\mathcal{A}} \times k_{\mathcal{B}}$ matrix, the i,j element (that in row i, column j) of which gives the probability that an open channel, initially in state i (one of

the $\mathcal{A}$ states), will eventually (after any number of transitions among the open states) arrive in state j, one of the short-lived shut states (set $\mathcal{B}$). $\mathbf{G}_{\mathcal{B}\mathcal{A}}$ has an exactly analogous interpretation. The product $\mathbf{G}_{\mathcal{A}\mathcal{B}}\mathbf{G}_{\mathcal{B}\mathcal{A}}$, which is a $k_{\mathcal{A}} \times k_{\mathcal{A}}$ matrix, thus describes routes from open states to brief shut states and back to open. This is just what happens during a burst of openings, so it is intuitively reasonable that the distribution of the number of openings per burst depends on this product. The distribution is, of course, discontinuous: the number of openings per burst (r) can take only the values 1, 2, 3, ... The probability, $P(r)$, of observing r openings per burst is (CH82-3.5):

$$P(r) = \boldsymbol{\phi}_{\mathrm{b}}(\mathbf{G}_{\mathcal{A}\mathcal{B}}\mathbf{G}_{\mathcal{B}\mathcal{A}})^{r-1}(\mathbf{I} - \mathbf{G}_{\mathcal{A}\mathcal{B}}\mathbf{G}_{\mathcal{B}\mathcal{A}})\mathbf{u}_{\mathcal{A}} \tag{60}$$

This is a *geometric distribution* (see also Chapters 18 and 19, this volume). As for the open time distribution, it starts with a $1 \times k_{\mathcal{A}}$ vector, $\boldsymbol{\phi}_{\mathrm{b}}$, which contains the $k_{\mathcal{A}}$ probabilities that the first opening in a burst starts in each of the $k_{\mathcal{A}}$ open states. These are not, in general, the same as the probabilities (in $\boldsymbol{\phi}_{\mathrm{o}}$, see equations 42 and 46) for *any* opening. This 'start-of-burst' vector is (CH82-3.2)

$$\boldsymbol{\phi}_{\mathrm{b}} = \frac{\mathbf{p}_{\mathcal{C}}(\infty)(\mathbf{Q}_{\mathcal{C}\mathcal{B}}\mathbf{G}_{\mathcal{B}\mathcal{A}} + \mathbf{Q}_{\mathcal{C}\mathcal{A}})}{\mathbf{p}_{\mathcal{C}}(\infty)(\mathbf{Q}_{\mathcal{C}\mathcal{B}}\mathbf{G}_{\mathcal{B}\mathcal{A}} + \mathbf{Q}_{\mathcal{C}\mathcal{A}})\mathbf{u}_{\mathcal{A}}} \tag{61}$$

Taking $\boldsymbol{\phi}_{\mathrm{b}}$ first, we have in our example, from equation 43,

$$\mathbf{p}_{\mathcal{C}}(\infty) = 0.9931 \tag{62}$$

(there is only one long-lived shut state, $k_{\mathcal{C}} = 1$, so this is a simple scalar, the equilibrium occupancy of state 5), and from equation 4 we have the various submatrices of $\mathbf{Q}$ needed to evaluate equations 59 and 61 as

$$\mathbf{Q}_{\mathcal{A}\mathcal{B}} = \begin{bmatrix} 0 & 3 \\ 0.5 & 0 \end{bmatrix}, \quad \mathbf{Q}_{\mathcal{B}\mathcal{B}} = \begin{bmatrix} -19 & 4 \\ 0.05 & -2.065 \end{bmatrix}, \quad \mathbf{Q}_{\mathcal{B}\mathcal{A}} = \begin{bmatrix} 0 & 15 \\ 0.015 & 0 \end{bmatrix},$$

$$\mathbf{Q}_{\mathcal{C}\mathcal{B}} = [0 \quad 0.01], \quad \mathbf{Q}_{\mathcal{C}\mathcal{A}} = [0 \quad 0]$$

and $\mathbf{Q}_{\mathcal{A}\mathcal{A}}$, which has already been given in equation 36. Multiplying the matrices gives

$$\boldsymbol{\phi}_{\mathrm{b}} = (0.275362 \quad 0.724638) \tag{63}$$

Comparing this with $\boldsymbol{\phi}_{\mathrm{o}}$, evaluated in equation 46, shows that the first opening of a burst has a greater probability (0.275) of starting in open state 1 than is the case for *all* openings (0.0741). This is clearly a result of the fact that, before the burst, the channel must have been in state 5, from which it must move to state 4, which communicates directly with state 1. Conversely, once the burst has started, much of the time is spent oscillating between shut state 3 and open state 2, so many of these openings start in open state 2.

Next, calculate $\mathbf{G}_{\mathcal{A}\mathcal{B}}$ and $\mathbf{G}_{\mathcal{B}\mathcal{A}}$ from equation 59. Inversion of $\mathbf{Q}_{\mathcal{A}\mathcal{A}}$ gives

$$\mathbf{Q}_{\mathcal{A}\mathcal{A}}^{-1} = \begin{bmatrix} -0.327876 & -0.0327439 \\ -4.36586 \times 10^{-4} & -1.99738 \end{bmatrix} \tag{64}$$

so we find

$$\mathbf{G}_{\mathcal{AB}} = \begin{bmatrix} 0.016372 & 0.983628 \\ 0.998690 & 0.0013098 \end{bmatrix} \qquad \mathbf{G}_{\mathcal{BA}} = \begin{bmatrix} 0.0015371 & 0.793519 \\ 0.0073011 & 0.019214 \end{bmatrix}. \tag{65}$$

Multiplication of these gives

$$\mathbf{G}_{\mathcal{AB}}\mathbf{G}_{\mathcal{BA}} = \begin{bmatrix} 0.0072068 & 0.031890 \\ 0.0015446 & 0.792504 \end{bmatrix} \tag{66}$$

We can now evaluate the distribution in equation 60 for any value of r. The value of $(\mathbf{G}_{\mathcal{AB}}\mathbf{G}_{\mathcal{BA}})$ raised to the power $r - 1$ can be calculated by repeatedly multiplying $\mathbf{G}_{\mathcal{AB}}\mathbf{G}_{\mathcal{BA}}$ by itself the necessary number of times.

However, once again, the spectral resolution trick proves useful. First, it provides a much quicker way of raising a matrix to a power than the obvious method of repeated multiplication. Second, it allows the distribution to be put in the scalar form of a mixture of simple geometric distributions (directly analogous with the exponentials in equations 40 and 47). The matrix $(\mathbf{G}_{\mathcal{AB}}\mathbf{G}_{\mathcal{BA}})^n$ can be expressed (see Appendix 1) in the form

$$(\mathbf{G}_{\mathcal{AB}}\mathbf{G}_{\mathcal{BA}})^n = \sum_{i=1}^{i=k_{\mathcal{A}}} \mathbf{A}_i \rho_i^n \tag{67}$$

where ρ_i are the eigenvalues of $\mathbf{G}_{\mathcal{AB}}\mathbf{G}_{\mathcal{BA}}$ (they are not denoted λ here because they are dimensionless rather than being rates), and $\mathbf{A}_i$ are the spectral matrices of $\mathbf{G}_{\mathcal{AB}}\mathbf{G}_{\mathcal{BA}}$, found as before (see Section 9). This enables the distribution in equation 60 to be put into the entirely scalar form of a mixture of simple geometric distributions:

$$P(r) = \sum_{i=1}^{i=k_{\mathcal{A}}} a_i(1 - \rho_i)\rho_i^{r-1} \qquad (r = 1, 2, \ldots) \tag{68}$$

where a_i represents the area of each component. It can be seen that the ith component distribution has the form of a geometrically decaying series, the value of ρ_i (which is less than 1) being the factor by which P is reduced each time r is increased by 1. This is a discrete analogue of exponential decay, and the 'mean number of openings per burst' for each component (analogous with the time constant for exponentials) is

$$\mu_i = 1/(1 - \rho_i). \tag{69}$$

These results show that the areas of the components can be calculated as

$$a_i = \frac{\boldsymbol{\phi}_{\mathrm{b}}\mathbf{A}_i(\mathbf{I} - \mathbf{G}_{\mathcal{AB}}\mathbf{G}_{\mathcal{BA}})\mathbf{u}_{\mathcal{A}}}{(1 - \rho_i)} \tag{70}$$

In our example, we find the eigenvaues of $\mathbf{G}_{\mathcal{AB}}\mathbf{G}_{\mathcal{BA}}$ to be

$$\rho_1 = 0.0071441 \qquad \rho_2 = 0.792567 \tag{71}$$

and these, from equation 69, correspond to components with

$$\mu_1 = 1.0072 \qquad \mu_2 = 4.8208 \tag{72}$$

The spectral matrices of $\mathbf{G}_{\mathcal{AB}}\mathbf{G}_{\mathcal{BA}}$ are

$$\mathbf{A}_1 = \begin{bmatrix} 0.999920 & -0.040603 \\ -1.9666 \times 10^{-3} & 7.9857 \times 10^{-5} \end{bmatrix} \qquad \mathbf{A}_2 = \begin{bmatrix} 7.9857 \times 10^{-5} & 0.040603 \\ 1.9666 \times 10^{-3} & 0.999920 \end{bmatrix} \tag{73}$$

Thus, from equation 70, the areas of the two components are

$$a_1 = 0.262793 \qquad a_2 = 0.737207 \tag{74}$$

In words, 73.7% of the area is accounted for by a component with a 'mean' of 4.82 openings per burst, but 26.3% of the area corresponds to a 'mean' of 1.007 openings per burst. There are more bursts that have only one opening than would be predicted from the former component alone.

7. Distribution of Burst Length

The total duration of a burst of openings is the sum of the durations of the r open times and $r - 1$ shut times that constitute the burst. The probability density of these burst lengths is given (CH82-3.17, 3.4) as

$$f(t) = \boldsymbol{\phi}_{\mathrm{b}}[\exp(\mathbf{Q}_{\mathscr{E}\mathscr{E}}t)]_{\mathcal{AA}}(-\mathbf{Q}_{\mathcal{AA}})(\mathbf{I} - \mathbf{G}_{\mathcal{AB}}\mathbf{G}_{\mathcal{BA}})\mathbf{u}_{\mathcal{A}} \tag{75}$$

(see also Section 13.4 of Chapter 18, this volume).

Once again, we need only to do some matrix multiplication, apart from finding the exponential of yet another matrix. This time we need to find $\exp(\mathbf{Q}_{\mathscr{E}\mathscr{E}}t)$, where $\mathscr{E}$ is the set of 'burst states' consisting of the open states, $\mathcal{A}$, and the short shut states, $\mathcal{B}$. Thus, $\exp(\mathbf{Q}_{\mathscr{E}\mathscr{E}}t)$ is a $k_{\mathscr{E}} \times k_{\mathscr{E}}$ matrix, where $k_{\mathscr{E}} = k_{\mathcal{A}} + k_{\mathcal{B}}$ (= 4 in our example), and the distribution will have $k_{\mathscr{E}}$ time constants $\tau_i = 1/\lambda_i$ where λ_i are now the eigenvalues of $-\mathbf{Q}_{\mathscr{E}\mathscr{E}}$. As before, we shall also need the $k_{\mathscr{E}}$ spectral matrices, $\mathbf{A}_i$, of $\mathbf{Q}_{\mathscr{E}\mathscr{E}}$, and these will each be of dimension $k_{\mathscr{E}} \times k_{\mathscr{E}}$. However, equation 75 contains a bit of notation that has not occurred before. Although $\exp(\mathbf{Q}_{\mathscr{E}\mathscr{E}}t)$ is a $k_{\mathscr{E}} \times k_{\mathscr{E}}$ matrix (4 × 4 in our case), the notation $[\exp(\mathbf{Q}_{\mathscr{E}\mathscr{E}}t)]_{\mathcal{AA}}$ means that we need only take the upper left $k_{\mathcal{A}} \times k_{\mathcal{A}}$ portion of it (the upper left 2 × 2 block in our case). This corresponds elegantly to the fact that a burst must both start and end in an open state and allows a general expression for a complicated quantity like the burst length to be written, in equation 75, in a form that is little more complicated than the distribution of open times.

As usual, the spectral expansion trick allows the distribution to be written as a scalar mixture of $k_{\mathscr{E}}$ exponential components, as in equation 40. In this case, the areas of each

component are given by

$$a_i = -\tau_i \phi_b [\mathbf{A}_i]_{\mathcal{AA}} \mathbf{Q}_{\mathcal{AA}} (\mathbf{I} - \mathbf{G}_{\mathcal{AB}} \mathbf{G}_{\mathcal{BA}}) u_{\mathcal{A}} \tag{76}$$

The quantities ϕ_b, $\mathbf{G}_{\mathcal{AB}}$, $\mathbf{G}_{\mathcal{BA}}$, etc. are all found as before. In our example, we have

$$\mathbf{Q}_{\mathcal{CC}} = \begin{bmatrix} -3.050 & 0.05 & 0 & 3 \\ 0.000666667 & -0.500666667 & 0.5 & 0 \\ 0 & 15 & -19 & 4 \\ 0.015 & 0 & 0.05 & -2.065 \end{bmatrix} \tag{77}$$

The eigenvalues of this are $\lambda_1 = 0.10160\ \text{ms}^{-1}$, $\lambda_2 = 2.01260\ \text{ms}^{-1}$, $\lambda_3 = 3.0933\ \text{ms}^{-1}$, and $\lambda_4 = 19.408\ \text{ms}^{-1}$, and the corresponding time constants are

$$\tau_1 = 9.84244\ \text{ms} \quad \tau_2 = 0.49687\ \text{ms} \quad \tau_3 = 0.323283\ \text{ms} \quad \tau_4 = 51.5246\ \mu\text{s} \tag{78}$$

This time, we shall not list all the spectral matrices, but the first one is

$$\mathbf{A}_1 = \left[\begin{array}{cc|cc} 1.8765 \times 10^{-5} & 0.037102 & 9.8702 \times 10^{-4} & 2.03952 \times 10^{-3} \\ 4.94699 \times 10^{-4} & 0.978107 & 0.026020 & 0.053767 \\ \hline 3.9481 \times 10^{-4} & 0.780608 & 0.020766 & 0.042910 \\ 1.0198 \times 10^{-5} & 0.020162 & 5.3638 \times 10^{-4} & 1.1083 \times 10^{-3} \end{array}\right] \tag{79}$$

In this matrix, the upper left hand $k_{\mathcal{A}} \times k_{\mathcal{A}}$ part has been marked by thin lines, so that part of $\mathbf{A}_1$ to be used to calculate the areas in equation 76 is

$$(\mathbf{A}_1)_{\mathcal{AA}} = \begin{bmatrix} 1.8765 \times 10^{-5} & 0.037102 \\ 4.94699 \times 10^{-4} & 0.978107 \end{bmatrix} \tag{80}$$

The areas, from equation 76, come out to be

$$a_1 = 0.73561 \quad a_2 = 0.01424 \quad a_3 = 0.25007 \quad a_4 = 0.0000772 \tag{81}$$

Only the first and third components have sufficiently large areas to be detectable experimentally.

7.1. Comparison with Relaxation

The time constants for this burst length distribution (rather than those for open times) are very similar to those found in Section 4 (equation 32) for the macroscopic relaxation (at the same agonist concentration), the time course of which thus depends largely on the burst length. The predominant component in both has a time constant of 9.82 ms. All the other components were small (and so not likely to be detectable in an experiment) in the case of

the relaxation, and the same is true for components 2 and 4 in the burst length distribution, which have very small areas. However, the burst-length distribution gives information that could not have been found from the relaxation, because the component with $\tau_3 = 0.323283$ ms has 25% of the area and therefore should be easily measurable in experiments.

7.2. Total Open Time per Burst

The distribution of the total open time per burst, expressions for which are given by Colquhoun and Hawkes (1982), is simpler than that for the burst length. Like the simple open time distribution, it has only $k_{\mathcal{A}}$ components (two in our case), and these are quite similar to the two components with non-negligible area in the burst-length distribution just discussed. The similarity of the distributions is, of course, a result of the fact that most shut times within a burst are very short in this example, so their omission makes little difference.

8. Channel Openings after a Jump

The whole discussion so far has concerned channels at equilibrium. It is also possible to study channel openings following application of some perturbation such as voltage or concentration jump. Following the jump there will be a period, before a new equilibrium is established, when openings can be observed in nonequilibrium conditions. The necessary theory for calculating distributions from the **Q** matrix under such conditions has been given by Colquhoun and Hawkes (1987). In general, the *time constants* are still calculated from the subsections of **Q**, so they are the same as found at equilibrium. What differs are the relative areas associated with each time constant.

A typical calculation of this sort would be to predict the probability distribution of the time to the first opening (the *first latency*) after the jump. In practice this can be a little tricky because, depending on the state of the system before and after the jump, some of the relevant submatrices become singular if the agonist concentration is zero. Furthermore, there may then be a nonzero chance of the channel failing ever to open at all. We will not, therefore give general results here.

We give only a simple case for our example mechanism when there is no agonist present initially, so that the channel must start in state 5. If there is an instantaneous jump in agonist concentration to level x_A, giving the general **Q** matrix of equation 4, then the time to first opening is simply a shut time starting from state 5. It therefore has a probability distribution that is found exactly as described in Section 5 for shut times, except that in this case the initial vector is taken as $\boldsymbol{\phi}_s = (0 \quad 0 \quad 1)$, rather than using the equilibrium initial vector given in equation 50 and numerically in equation 57. The time constants are therefore exactly the same as in equation 55, the $\mathbf{A}_i$ are as given in equation 58, but with this new $\boldsymbol{\phi}_s$ substituted into equation 54, we now have areas

$$a_1 = 1.000138 \qquad a_2 = -0.0001392 \qquad a_3 = 1.224 \times 10^{-6} \tag{82}$$

Note that although these areas sum to 1 as usual, the second one is negative. It is easily verified that, with these values substituted into equation 40, the corresponding probability density function is zero at $t = 0$, rises to a maximum, and then decays back to zero. This occurs because the system starts in state 5 and must make a transition into state 4 before

there is any chance at all of an opening taking place. The density is depicted in Fig. 1a; on this time scale it is not easy to see exactly what is happening near $t = 0$, and the distribution is very nearly just a single exponential with mean 3789 ms but with just a little something funny at the origin. Figure 1b shows more clearly the behaviour at the origin; the density rises smoothly over the first 2 ms and is then almost flat up to 5 ms before decaying as in Fig. 1a.

9. Calculating the Exponential of a Matrix

In the preceding sections we have seen that all that is needed to calculate a wide range of theoretical distributions for any mechanism is some straightforward matrix algebra and the ability to find the exponential of a matrix. In most cases one would want to do this via the spectral expansion, giving rise to mixed exponential (or geometric) functions, but we also discuss briefly a method that does not require this.

9.1. Functions of a Matrix

The central, and beautiful, result that underlies the tricks described above can be put in the following form. For any analytic function, f, and any $n \times n$ matrix $\mathbf{M}$ whose eigenvalues are distinct, the corresponding function of the matrix $\mathbf{M}$, denoted $f(\mathbf{M})$, can be written in the form

$$f(\mathbf{M}) = \sum_{i=1}^{i=n} \mathbf{A}_i f(\lambda_i) \tag{83}$$

where the λ_i are the eigenvalues of $\mathbf{M}$ and the $\mathbf{A}_i$ are its spectral matrices, the calculation of which is described below. The beauty of this result is that when we have some function of a matrix such as $\exp(\mathbf{Q})$, the meaning of which is not immediately apparent, it is changed into a function of the λ values, $\exp(\lambda)$, and, since the λ values are ordinary scalar numbers, the problem disappears.

For example, equation 83 shows immediately, for $f(\mathbf{M}) = \mathbf{M}^0$, that

$$\sum_{i=1}^{i=n} \mathbf{A}_i = \mathbf{I} \tag{84}$$

and for $f(\mathbf{M}) = \mathbf{M}$, we also see immediately that

$$\sum_{i=1}^{i=n} \mathbf{A}_i \lambda_i = \mathbf{M} \tag{85}$$

These two results are useful in checking calculations of the spectral matrices, $\mathbf{A}_i$.

It may also be mentioned that the spectral matrices have the interesting property that multiplying together any pair of them results in a matrix that is all zeroes; i.e., $\mathbf{A}_i \mathbf{A}_j = \mathbf{0}$ $(i \neq j)$. In fact, this property is responsible for many of their useful characteristics.

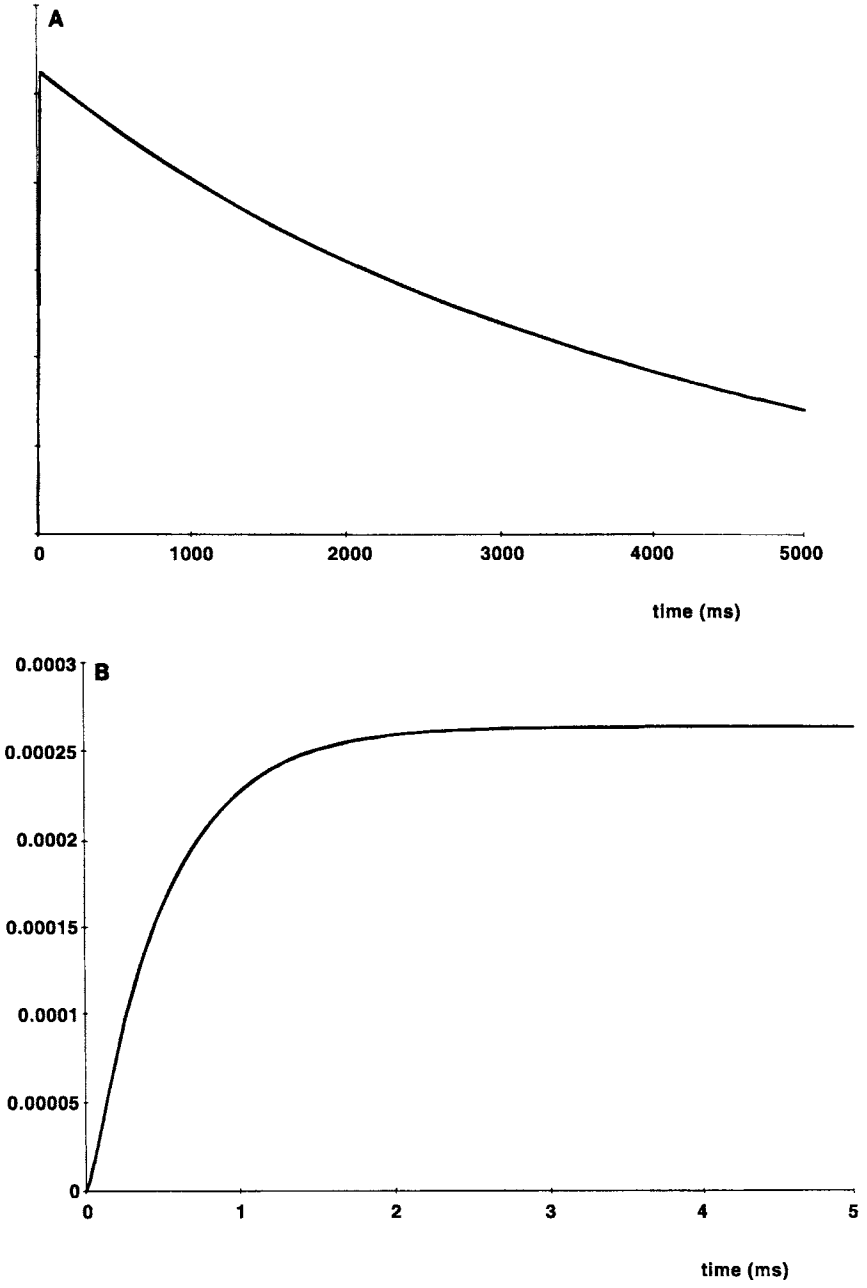


Figure 1. Probability density of first latency (a) over 5 sec and (b) near the origin.

The two particular examples of the application that we have used in the foregoing sections are, first, to calculate powers of a matrix in Section 6 (equation 67), i.e., taking $f(\mathbf{M}) = \mathbf{M}^r$

$$\mathbf{M}^r = \sum_{i=1}^{i=n} \mathbf{A}_i \lambda_i^r \tag{86}$$

and, second, the ubiquitous exponential of a matrix, $f(\mathbf{M}) = \exp(\mathbf{M})$,

$$\exp(\mathbf{M}) = \sum_{i=1}^{i=n} \mathbf{A}_i \exp(\lambda_i). \tag{87}$$

In most of our examples the matrix $\mathbf{M}$ represents $\mathbf{Q}t$ or some subsection of $\mathbf{Q}$ multiplied by t.

9.2. Calculation of the Spectral Matrices

Now we come, at last, to describing how actually to calculate the spectral matrices, the step that underlies almost everything described in the preceding sections. It is not hard because library routines are available to do all the difficult bits.

First we will just describe the bare bones of the algorithm that enables computation of the quantities $\mathbf{A}_i$ and λ_i that were first introduced in equation 21. Later we will discuss the mathematics of it a little, but that can be skipped by those who do not care to know it.

Any good computer library contains routines that do a trick called 'finding the eigenvalues of a general square matrix'. The term *eigenvalue* is discussed briefly in Appendix 1, but all we need to know here is that the required values of λ_i are nothing other than the eigenvalues of the matrix $-\mathbf{Q}$, and the time constants are the reciprocals of these values, $\tau_i = 1/\lambda_i$. We therefore simply put the values for the transition rates into $\mathbf{Q}$, change the signs of all the elements to get $-\mathbf{Q}$, and use a standard routine to calculate the eigenvalues and hence the time constants (alternatively, find the eigenvalues of $\mathbf{Q}$, and change their sign). A matrix of size $k \times k$, such as $\mathbf{Q}$, will generally have k eigenvalues. The matrix $\mathbf{Q}$ (and $-\mathbf{Q}$) is singular, which means that one of the eigenvalues will be zero (and we do not divide by zero to get a time constant!); it is usually convenient to arrange for this to be the first one, so we will assume $\lambda_1 = 0$.

Standard computer library routines will also find the *eigenvectors* of the matrix $-\mathbf{Q}$ (usually these will be calculated in the same routine as the eigenvalues). The routine will often produce the k eigenvectors, each a column vector $\mathbf{x}_i$ associated with the value λ_i, already in the form of the columns of a matrix $\mathbf{X}$ (if it does not supply them in this form, then use the $\mathbf{x}_i$ to create a matrix $\mathbf{X}$). Thus, $\mathbf{X}$ can be thought of as a set of columns

$$\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_k] \tag{88}$$

This is now inverted to produce a matrix which we shall call $\mathbf{Y} = \mathbf{X}^{-1}$; again, a standard routine will do that. We shall denote the ith row of $\mathbf{Y}$ as $\mathbf{y}_i$ (a $1 \times k$ matrix). We are now

in a position to calculate the *k spectral matrices*, denoted $\mathbf{A}_i$, that are needed to complete the trick. These are calculated as

$$\mathbf{A}_i = \mathbf{x}_i\mathbf{y}_i, \qquad i = 1 \text{ to } k \tag{89}$$

These are each square $k \times k$ matrices, as they arise from a column postmultiplied by a row (see example A5 in the Appendix). You may not easily find this final part as a ready-made program, but it is very simple to program, just a few things to multiply together. Then $e^{\mathbf{Q}t}$ is represented in the form of equation 21, which leads, as we have seen in several examples, to various functions of interest: all have the form of a mixture of exponentials with rate constants τ_i and weights that are computed by pre- and postmultiplying the matrices $\mathbf{A}_i$ by various appropriate vectors.

9.3. Other Exponentials

We have described the spectral representation of $e^{\mathbf{Q}t}$. Exactly the same procedure applies if you want to use $\mathbf{Q}_{\mathcal{AA}}$, $\mathbf{Q}_{\mathcal{BB}}$, etc. instead of $\mathbf{Q}$. The only differences are that, instead of k, you will get $k_{\mathcal{A}}$, $k_{\mathcal{B}}$, etc. eigenvalues and $\mathbf{A}_i$ matrices, and usually none of the eigenvalues will take the value zero.

9.4. Calculation of Spectral Matrices Using the NAG Library

As a particular example of the sort of code needed, the following is a subroutine (in FORTRAN) that calls the NAG library subroutine F02AGF, which returns the eigenvalues and eigenvectors of a general square matrix. If, for example, we want the spectral matrices and eigenvalues of $\mathbf{Q}_{\mathcal{FF}}$, we put the appropriate elements of $\mathbf{Q}$ into an array QFF and call QMAT thus:

```
call QMAT(QFF, Amat, kF, eigenval, ifail)
```

This subroutine might be made more complete by allowing adjustable array dimensions, by adding a routine to sort the eigenvalues (along with their associated eigenvectors) into ascending or descending order, and by adding checks on the accuracy of matrix inversion (e.g., by checking that $\mathbf{XY} = \mathbf{I}$ to acceptable accuracy), but the version shown works fine.

```
      subroutine QMAT(Q,Amat,k,eigenval,ifail)
c  Calculation of eigenvalues and spectral matrices using the NAG library
c  routine F02AGF
c  INPUT:
c    Q=double precision matrix (k×k)
c    k=size of Q
c  OUTPUT:
c    eigenval(i)=eigenvalues of Q (i=1,...,k)
c    A(m)=mth spectral matrix of Q (m=1,...,k). Each A(m) is k×k in
c      size and Amat(i,j,m) is value in ith row and jth column of A(m)
c    ifail=0 if there are no errors in the NAG routine, F02AGF
```

```
c
      IMPLICIT double precision (A-H,O-Z)
      double precision Q(10,10),Amat(10,10,10),eigenval(10)
      double precision X(10,10),Y(10,10)
c   next 2 lines for F02AGF
      double precision QD(10,10),eigimag(10),Ximag(10,10)
      integer*4 iwork(10)
c
c   Define QD = −Q; (this also preserves the input value of Q)
        do i = 1,k
            do j = 1,k
                QD(i,j) = −Q(i,j)
            enddo
        enddo
c
        km = 10      !maximum k, defined by declarations
        ifail = 0
        call F02AGF(QD,km,k,eigenval,eigimag,X,km,Ximag,km,iwork,ifail)
c   X now has columns that are the real parts of the k column eigenvectors
c   of Q. Now invert X using a matrix inversion subroutine; put result in Y.
        call MATINV(X,k,km,Y,km)
c   Calculate the spectral matrices, A(m), from X and Y.
        do m=1, k
            do i=1, k
                do j=1, k
                    Amat(i, j, m)=X(i, m)*Y(m, j)
                enddo
            enddo
        enddo
        RETURN
        end
```

Another generally useful bit of code is the following fragment that will calculate scalars, w_m, by premultiplying $\mathbf{A}_m$ by a row vector, defined as an array *row*(*i*), and postmultiplying it by a column vector defined as an array *col*(*i*). The distributions described in earlier sections mostly end up in this form. The subscript m is used here for $\mathbf{A}_m$ so we can keep to the usual notation of using *i,j* for rows and columns.

```
do m = 1,k
    w(m) = 0.0
    do i = 1,k
        do j = 1,k
            w(m) = w(m) + row(i) * Amat(i,j,m) * col(j)
        enddo
    enddo
enddo
```

9.5. Calculation of Matrix Exponentials Using MAPLE

In recent years a number of programs have been developed that are able to carry out symbolic manipulation, such as factorization, differentiation, integration, and many other mathematical operations as well as doing numerical calculations. We give here a simple example in one of these programs, called MAPLE.

Let us take the case of the shut-time distributions, for which we need the exponential of $\mathbf{Q}_{\mathcal{F}\mathcal{F}}$. First we must start the linear algebra package and then create the matrix

```
with(linalg):
Qff:=matrix(3, 3, [−19, 4, 0, 0.05, −2.065, 2, 0, 0.01 ,−0.01]);        (90)
```

Then $e^{\mathbf{Q}_{\mathcal{F}\mathcal{F}}t}$ is evaluated at $t = 2$, say, simply be typing

```
exponential(Qff*2);                                                      (91)
```

giving the result

$$\begin{bmatrix} 1.381 \times 10^{-5} & 0.004786 & 0.2002 \\ 5.982 \times 10^{-5} & 0.02078 & 0.9529 \\ 1.251 \times 10^{-5} & 0.004764 & 0.9948 \end{bmatrix}. \tag{92}$$

If instead we want the spectral expansion, the eigenvalues are returned by

```
eigenvals(−Qff);                                                         (93)
```

giving

```
[19.01, 2.063, 0.0002639],                                               (94)
```

and

```
eigenvects(−Qff);                                                        (95)
```

gives the above eigenvalues together with the eigenvectors, which we report below as columns of a matrix $\mathbf{X}$ in the same order as the corresponding eigenvalues in equation 94:

$$\mathbf{X} = \begin{bmatrix} 9.996 & 0.2358 & 0.02042 \\ -0.02949 & 0.9986 & 0.09702 \\ 1.55 \times 10^{-5} & -0.004864 & 0.09965 \end{bmatrix} \tag{96}$$

From these one can obtain the spectral matrices as described at the beginning of this section. Notice, however, that the eigenvalues in equation 94 have come out in the reverse order from those above equation 55, so the $\mathbf{A}_i$ calculated from $\mathbf{X}$ in equation 96 will also come

out in reverse order compared to those in equation 58; none of this matters as long as you keep track of what goes with what. As discussed in the Appendix, the eigenvectors are not unique, as they can each be multiplied by arbitrary nonzero constants (so do not worry too much if your program gives different results), but the resulting $\mathbf{A}_i$ should be the same.

9.6. Calculation of Spectral Matrices and Matrix Exponentials Using MATHEMATICA

Another powerful modern computer algebra package, having broadly similar features to MAPLE, is MATHEMATICA. We illustrate the use of this also with some simple examples. As in the previous section we will look at shut times.

$$\text{Qff} = \{\{-19, 4, 0\}, \{0.05, -2.065, 2\}, \{0, 0.01, -0.01\}\}; \tag{97}$$

is equivalent to the MAPLE statement 90, and MatrixForm[Qff] will print it so that it looks like the result in equation 51.

$$\text{MatrixExp[Qff 2];} \tag{98}$$

is the equivalent of statement 91 and gives the same result. If we want to use the spectral expansion explicitly, we can use a series of statements as follows.

The eigenvalues and eigenvectors can be found from

$$\{\text{vals,vects}\} = \text{Eigensystem}[-\text{Qff}]; \tag{99}$$

The eigenvectors are stored as rows, and we use columns, so

$$\text{X} = \text{Transpose[vects]}; \qquad \text{Y} = \text{Inverse[X]}; \tag{100}$$

Now we can define a function, let us call it expqt to calculate $e^{\mathbf{Q}t}$ for any t using equation 107, below. Thus,

$$\text{expqt[t\_]:= X.DiagonalMatrix[E\^{}(-vals t)].Y;} \tag{101}$$

We can now use this function for any t; for example,

$$\text{expqt[2]}$$

now gives the same result as equation 98.

Notice that in these examples a space indicates multiplication by a scalar, and a dot indicates matrix multiplication.

To extend the example, we can do the calculations for the probability density of first latency discussed in Section 8. We need to set some vectors

$$\text{phis} = \{0,0,1\}; \qquad \text{uf} = \{1,1,1\};$$

then define a new function

```
latencypdf[t_]:= -phis.expqrt[t].Qff.uf;                    (102)
```

which is a representation of equation 49. Then we can use this for any t; for example, latencypdf[0] returns the value 0, confirming the discussion in Section 8, where it was said that the density starts at 0 when $t = 0$. A plot of this density over the interval $0 \leq t \leq 5$ ms can be had simply by issuing the command

```
Plot[latencypdf[t],{t,0,5}]
```

You need a few extra options to label it and make it pretty.

The spectral matrices have, in effect, been built into equation 101. If you want to find them explicitly, they can be found as

```
A1 = Outer[Times,vects[[1]],Y[[1]]]                         (103)
```

This does the calculation of equation 89 for the first eigenvector and the first row of the matrix **Y**. The second and third spectral matrices can be found similarly, simply replacing the number 1 wherever it occurs in equation 103 by 2 or 3, as appropriate. The results should be the same as equation 58, although not necessarily in the same order (see discussion above concerning MAPLE).

9.7. Another Way to Calculate the Exponential of a Matrix

Most people will want to use the spectral representation method because they feel that they can interpret the time constants that it yields. Sometimes, however, you may not need that, and an alternative would do. A number of ways are discussed by Moler and van Loan (1978). We discuss here one method that we have found robust; an APL program that implements it appears in Hawkes (1984).

Let $\mathbf{M} = \mathbf{Q}t$, for some fixed t. Then we want to calculate $e^{\mathbf{M}}$. There are two essential parts to this method.

9.7.1. Core Method

If **M** is in some sense 'small', then it is easy and quite accurate to use a truncated series calculation similar to equation 20; i.e.,

$$e^{\mathbf{M}} = \mathbf{I} + \mathbf{M} + \frac{\mathbf{M}^2}{2!} + \frac{\mathbf{M}^3}{3!} + \cdots \frac{\mathbf{M}^N}{N!} \tag{104}$$

We have found stopping after $N = 13$ terms is sufficient. What is meant by small? Take the sum of the modulus of each element of **M** in the ith row; do this for each row and let Δ be the biggest of these sums. The core method works well if $\Delta < 1/2$.

9.7.2. Squaring Method

If $\Delta > 1/2$, then find an integer r such that $\Delta/R < 1/2$, where $R = 2^r$. The essence of the method is that

$$(e^{\mathbf{M}/R})^R = e^{\mathbf{M}} \tag{105}$$

so calculate $\mathbf{T} = e^{\mathbf{M}/R}$ using the core method, and then find $\mathbf{T}^R$ by matrix multiplication. This is quite cunning, but the choice of R to have the form 2^r is also very cunning because it means you only need to do r matrix multiplications instead of $R - 1$, by calculating successive squares. For example, if $r = 4$ so $R = 16$, then $\mathbf{T}^2 = \mathbf{T} \times \mathbf{T}$; $\mathbf{T}^4 = \mathbf{T}^2 \times \mathbf{T}^2$; $\mathbf{T}^8 = \mathbf{T}^4 \times \mathbf{T}^4$; $\mathbf{T}^{16} = \mathbf{T}^8 \times \mathbf{T}^8$; and you have calculated $\mathbf{T}^{16}$ with a series of just four multiplications instead of 15 if you did it the hard way: $\mathbf{T}^{16} = \mathbf{T} \times \mathbf{T} \times \mathbf{T} \times \mathbf{T} \times \mathbf{T} \ldots \times \mathbf{T}$.

9.8. Further Mathematical Notes

In this section we discuss briefly the mathematical justification of the spectral expansion. It is not essential reading for those who merely want to do the calculations but serves to satisfy the mathematically curious.

In the Appendix we describe how a square matrix, **M**, can, in some circumstances, be represented in the form $\mathbf{M} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}$, where $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues of **M** and the columns of the matrix **X** consist of the corresponding eigenvectors of **M**. It can be shown that this is always possible if **M** is the **Q** matrix of a reversible Markov process and that the eigenvalues are real and negative (apart from the one zero value $\lambda_1 = 0$), and the eigenvectors have real elements. The eigenvalues, λ_i, of the matrix $-\mathbf{Q}$ are simply minus the eigenvalues of **Q** and are therefore nonnegative; the eigenvectors of **Q** and $-\mathbf{Q}$ can be taken as the same. This is also true if **Q** is replaced by any of the submatrices $\mathbf{Q}_{\mathcal{A}\mathcal{A}}$, $\mathbf{Q}_{\mathcal{B}\mathcal{B}}$, $\mathbf{Q}_{\mathcal{F}\mathcal{F}}$, $\mathbf{Q}_{\mathcal{E}\mathcal{E}}$ with dimensions reduced from k to $k_{\mathcal{A}}$ or $k_{\mathcal{B}}$, etc., and in those cases none of the eigenvalues are zero.

Equation (A15) of the Appendix then implies that

$$\mathbf{Q}^r = \mathbf{X}(-\mathbf{\Lambda})^r\mathbf{X}^{-1} \tag{106}$$

Now from equation 20 we find that

$$\begin{aligned} e^{\mathbf{Q}t} &= \mathbf{I} + \mathbf{Q}t + \frac{(\mathbf{Q}t)^2}{2!} + \frac{(\mathbf{Q}t)^3}{3!} + \frac{(\mathbf{Q}t)^4}{4!} + \cdots \\ &= \mathbf{I} + \mathbf{X}(-\mathbf{\Lambda}t)\mathbf{X}^{-1} + \frac{\mathbf{X}(-\mathbf{\Lambda}t)^2\mathbf{X}^{-1}}{2!} + \frac{\mathbf{X}(-\mathbf{\Lambda}t)^3\mathbf{X}^{-1}}{3!} + \frac{\mathbf{X}(-\mathbf{\Lambda}t)^4\mathbf{X}^{-1}}{4!} + \cdots \\ &= \mathbf{X}\left[\mathbf{I} + (-\mathbf{\Lambda}t) + \frac{(-\mathbf{\Lambda}t)^2}{2!} + \frac{(-\mathbf{\Lambda}t)^3}{3!} + \frac{(-\mathbf{\Lambda}t)^4}{4!} + \cdots\right]\mathbf{X}^{-1} \end{aligned}$$

The expression in brackets is obviously $e^{-\Lambda t}$, but if we look at it in detail we see that it is a sum of diagonal matrices, so it must again be diagonal. Then

$$e^{\mathbf{Q}t} = \mathbf{X}e^{-\Lambda t}\mathbf{X}^{-1} \tag{107}$$

Furthermore, the ith element on the diagonal of $e^{-\Lambda t}$ is just

$$1 + (-\lambda_i t) + \frac{(-\lambda_i t)^2}{2!} + \frac{(-\lambda_i t)^3}{3!} + \frac{(-\lambda_i t)^4}{4!} + \cdots = \exp(-\lambda_i t)$$

The key result (equation 21) therefore follows immediately from equation A20 of the Appendix. We have thus justified the general statement (equation 83) for the particular case of the exponential function (remembering that, for convenience, we choose to work with the eigenvalues of $-\mathbf{Q}$ instead of $\mathbf{Q}$).

10. Time Interval Omission

All of the preceding distributions are derived under ideal assumptions. It has long been recognised, however, that the limitations on observational resolution caused by noise and filtering by the recording equipment can distort the observation of open times and shut times through failure to observe very short intervals. Some discussion of this phenomenon is given in Chapter 18 (this volume). We are unable to go into details here but note that, although it is naturally more complicated, and the nice mixture-of-exponentials feature now only appears as a very good asymptotic approximation, the kinds of matrix operations involved are mostly no more difficult than those discussed here. The main complication arises from the need to find some sort of generalised eigenvalues, which is equivalent to finding at what values of some parameter a certain determinant of a matrix vanishes: that is not all that difficult. Some details are given by Hawkes *et al.* (1992).

11. Concluding Remarks

The results of the classical theory are quite easy in terms of matrix algebra provided one can calculate the exponential of certain matrices. Computer software is readily available in many languages or packages that will carry out all of the necessary operations, including (sometimes with a little bit of effort) those exponentials. Thus, it should not be too difficult for anyone to put together a program in his or her own favourite system. We hope this chapter helps to clarify what is needed.

The necessary tools can be summmarised as follows. It is important to note that all calculations should be done using double-precision arithmetic. You will need:

1. Subroutines/procedures to add, subtract, multiply, and invert matrices.
2. A routine to extract a submatrix (consisting of specified rows and columns) from a larger matrix (unless the other routines are capable of operating directly on subsections of a matrix).

3. A routine to find the eigenvalues and eigenvectors of a general square matrix, preferably sorted into ascending or descending order of the eigenvalues, and to calculate the spectral matrices from them.

The numerical examples provide a benchmark against which you can test the results from your own programs. Good luck!

Appendix 1. A Brief Introduction to Matrix Notation

This account is a brief synopsis. For further details see, for example, Stephenson (1965) or Mirsky (1982).

A1.1. Elements of a Matrix

A matrix is a table and is usually denoted by a bold type symbol. It is rather like a spreadsheet, the entries in the table being defined by the row and column in which they occur. The entry in the ith row and the jth column of the matrix **A**, an *element* of **A**, is usually denoted in lower case italic as a_{ij}. A matrix with n rows and m columns is said to be an $n \times m$ matrix, or to have *shape* $n \times m$. If $n = m$, we have a square matrix. Thus, for example, 2×2 and 2×3 matrices can be written as

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{bmatrix}. \tag{A1}$$

Clearly, a 1×1 matrix has just one element and can, for most purposes, be treated as an ordinary number (called a *scalar*, which has no shape), though strictly speaking it is not.

The elements for which $i = j$ are called *diagonal elements* (e.g., a_{11} and a_{22}, in this example), and the rest ($i \neq j$) are called *off-diagonal elements.*

A1.2. Vectors

Vectors are nothing new. In the context of matrix algebra they are simply matrices that happen to have only one row (a *row vector* or $1 \times n$ matrix) or only one column (a *column vector* or $n \times 1$ matrix). They are manipulated just like any other matrix.

A1.3. Equality of Matrices

Two matrices are said to be *equal* if all the corresponding elements of each are equal. Clearly, in that case, the two matrices must both be the same shape. For example $\mathbf{A} = \mathbf{B}$ means, in the 2×2 case,

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

which is a shorthand way of writing the four separate relationships: $a_{11} = b_{11}$, $a_{12} = b_{12}$, $a_{21} = b_{21}$, and $a_{22} = b_{22}$.

A1.4. Addition and Subtraction of Matrices

This is very easy. You just add (or subtract) the corresponding elements in each of them (clearly, the two matrices must be the same shape). Thus, for example,

$$\mathbf{C} = \mathbf{A} + \mathbf{B} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} (a_{11} + b_{11}) & (a_{12} + b_{12}) \\ (a_{21} + b_{21}) & (a_{22} + b_{22}) \end{bmatrix}$$

shape: 2×2 2×2 2×2

or, more briefly, $c_{ij} = a_{ij} + b_{ij}$ for all i and j. Subtraction, $\mathbf{A} - \mathbf{B}$, is defined in the obvious equivalent manner.

A1.5. Multiplication of Matrices

The definition of the product of two matrices may, at first sight, seem a bit perverse, but it turns out to be exactly what is needed for the convenient representation of, for example, simultaneous equations or for representation of all the possible routes from one state to another [see, for example, section 2 of Colquhoun and Hawkes (1982)]. Multiplication goes 'row into column.' For example:

$$\mathbf{C} = \mathbf{AB} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} (a_{11}b_{11} + a_{12}b_{12}) & (a_{11}b_{12} + a_{12}b_{22}) \\ (a_{21}b_{11} + a_{22}b_{21}) & (a_{21}b_{12} + a_{22}b_{22}) \end{bmatrix} \quad \text{(A2)}$$

shape: 2×2 2×2 2×2

Thus, the element in the ith row and the jth column of the product, **C**, is obtained by taking the ith row of **A** and the jth column of **B**, multiplying the corresponding elements, and adding all these products. Clearly, the number of columns in **A** must be the same as the number of rows in **B**. If **A** is an $n \times m$ matrix, and **B** is $m \times k$, then the product, $\mathbf{C} = \mathbf{AB}$ will be an $n \times k$ matrix. More formally, the element c_{ij} is obtained from the sum of products

$$c_{ij} = \sum_{r=1}^{m} a_{ir} b_{rj}$$

Some computer languages allow matrices to be added or multiplied symbolically, e.g., by simply writing **A** = **B** * **C**; in others a subroutine or procedure must be called to do this. In the APL language, matrix multiplication is a special case of a powerful idea called the *inner product* operator. The APL expression $C{\leftarrow}A + . \times B$, although not standard mathematical notation, reflects the fact that the result is obtained by a combination of adding and multiplying (guess what the result of $A \times . + B$ is).

It is important to note that, contrary to the case with ordinary (scalar) numbers, it is *not* generally true that **AB** and **BA** are the same (multiplication of matrices is *not* necessarily *commutative*). Indeed, both products will exist only if **A** has shape $n \times m$ and **B** has shape $m \times n$: then the product **AB** has shape $n \times n$, whereas **BA** has shape $m \times m$. The two products will only have the same shape if **A** and **B** are square matrices of the same shape, and even then the products are not necessarily the same. We must therefore distinguish between *premultiplication* and *postmultiplication.* In the product **AB,** it is said that **A** premultiplies **B** (or that **B** postmultiplies **A**).

If a matrix is multiplied by an ordinary (scalar) number, x say, this means simply that every element of the matrix is multiplied by x. Thus $x\mathbf{A}$, which is equal to $\mathbf{A}x$, is a matrix with elements xa_{ij}.

It can be shown that matrix multiplication is *associative;* i.e., parentheses are not needed because, for example, **A(BC)** = **(AB)C,** which can therefore be written unambiguously as the triple product **ABC.**

A1.6. Some More Examples of Matrix Multiplication

A few more examples may help to clarify the rules given above. If we postmultiply a row vector (a $1 \times n$ matrix) by a column vector ($n \times 1$ matrix), we get an ordinary (scalar) number (a 1×1 matrix). For example,

$$\mathbf{ab} = [a_1 \quad a_2 \quad a_3]\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = (a_1b_1 + a_2b_2 + a_3b_3) \tag{A3}$$

shape: 1×3 $\quad$ 3×1 $\quad$ 1×1

When looking at a matrix expression, it is always useful to note the size of each array in the expression, as written below the results above. This makes it instantly obvious, for example, that the result in equation A3 is scalar. Likewise, the expression **aXb** where **a** is a $1 \times n$ row vector, **X** is an $n \times m$ matrix, and **b** is an $m \times 1$ column vector, is clearly also scalar. Note, however, that we sometimes do need to distinguish between a scalar and a 1×1 matrix; for example, if **C** is a 3×3 matrix and **a** and **b** are as above, then **(ab)C** makes sense if **ab** is regarded as a scalar, but it is not equal to **a(bC)** because the multiplication **bC** is not possible (you cannot multiply a 3×1 and a 3×3 matrix), and you have broken the associative law mentioned above. If you consider **ab** as a 1×1 matrix, however, you cannot multiply that by **C** either, so that is all right. Users of logical software, such as APL or MATHEMATICA may need to be careful of this distinction.

Note that it follows from equation A3 that $\mathbf{aa}^{\mathrm{T}}$, where $\mathbf{a}^{\mathrm{T}}$ is the transpose of **a** (see below), is a sum of squares, thus:

$$\mathbf{aa}^{\mathrm{T}} = [a_1 \quad a_2 \quad a_3]\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \sum a_i^2$$

shape: 1×3 $\quad$ 3×1 $\quad$ 1×1

Similarly, if **u** is a unit column vector (all elements $u_i = 1$), then premultiplying it by a row vector simply sums the elements of the latter (this is a common feature in the distributions described in this chapter), thus:

$$\mathbf{au} = [a_1 \quad a_2 \quad a_3]\begin{bmatrix}1\\1\\1\end{bmatrix} = \Sigma\, a_i \tag{A4}$$

If the multiplication is done the other way around, **ba** instead of **ab,** we are multiplying an $n \times 1$ matrix by a $1 \times n$ matrix, so the result has shape $n \times n$, thus:

$$\mathbf{ba} = \begin{bmatrix}b_1\\b_2\\b_3\end{bmatrix}[a_1 \quad a_2 \quad a_3] = \begin{bmatrix}b_1a_1 & b_1a_2 & b_1a_3\\b_2a_1 & b_2a_2 & b_2a_3\\b_3a_1 & b_3a_2 & b_3a_3\end{bmatrix} \tag{A5}$$

shape: 3×1 1×3 3×3

A1.7. The Identity Matrix

The matrix equivalent of the number 1 is the identity matrix, denoted **I.** This is a square matrix for which all the diagonal elements are 1, and all others are zero. Actually there are lots of different identity matrices, one for each possible shape, 1×1, 2×2, 3×3, etc.; often we do not bother to indicate the shape because it is usually obvious from the context. It has the property that multiplication of any matrix by **I** of the appropriate shape does not change the matrix. For the 2×2 and 3×3 cases, we have, respectively,

$$\mathbf{I} = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} \quad \text{and} \quad \mathbf{I} = \begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix} \tag{A6}$$

Thus, for example, using the matrices defined in equation A1,

$$\mathbf{AI} = \mathbf{IA} = \mathbf{A}$$

$$\mathbf{CI} = \mathbf{IC} = \mathbf{C}$$

Note that the **I** matrix that postmultiplies **C** must have shape 3×3, whereas all the others must be 2×2.

A1.8. Determinants

A determinant is a *number* (not a table) that can be calculated from the elements of a square matrix. The determinant is usually written just like the matrix, except that it is enclosed

by vertical lines rather than brackets (it therefore looks rather similar to the matrix, and it is important to remember that the whole symbol represents a single number). For example, the determinant of the 2 × 2 matrix in (A1), denoted det(**A**), is

$$\det(\mathbf{A}) \equiv |\mathbf{A}| \equiv \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21} \tag{A7}$$

The right-hand side of this shows how the number is calculated from the elements of **A.** A matrix that has a determinant of zero, det(**A**) = 0, is said to be a *singular* matrix; such a matrix has a linear relationship between its rows or between its columns and cannot be inverted (see below).

For larger matrices the definition gets a bit more complicated. One does not have to worry too much about the details, as most scientific software has functions to calculate a determinant. The next paragraph may therefore be safely skipped.

For an $n \times n$ matrix **A,** the determinant is defined as

$$\det(\mathbf{A}) \equiv |\mathbf{A}| = \sum (-1)^{\sigma(\pi)} a_{1\pi(1)} a_{2\pi(2)} \cdots a_{n\pi(n)}$$

where the summation is over all permutations π of the integers 1 to n, and $\sigma(\pi)$ equals 1 if π is an 'even' permutation and equals -1 if π is on 'odd' permutation. For example, if **A** is a 3 × 3 matrix, then

$$|\mathbf{A}| = a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33}$$

For more detail see any standard text, such as one of those referenced above.

A1.9. 'Division' of Matrices

If **C** = **AB,** then what is **B**? In ordinary scalar arithmetic we would simply divide both sides by **A** to get the answer, provided **A** was not zero. When **A** and **B** are matrices, the method is analogous. We require some matrix analogue of what, for ordinary numbers, would be called the reciprocal of a square matrix **A.** Suppose that some matrix exists, which we shall denote $\mathbf{A}^{-1}$, that behaves like the reciprocal of **A** in the sense that, by analogy with ordinary numbers,

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I} = \mathbf{A}^{-1}\mathbf{A}$$

where **I** is the identity matrix. The matrix $\mathbf{A}^{-1}$ is called the *inverse* of **A.** Thus, the solution to the problem posed initially can be found by premultiplying both sides of **C** = **AB** by $\mathbf{A}^{-1}$ to give the result as **B** = $\mathbf{A}^{-1}$**C.** In the case of a 2 × 2 matrix the inverse can be written explicitly as

$$\mathbf{A}^{-1} = \begin{bmatrix} a_{22}/\det(\mathbf{A}) & -a_{12}/\det(\mathbf{A}) \\ -a_{21}/\det(\mathbf{A}) & a_{11}/\det(\mathbf{A}) \end{bmatrix}$$

where det(**A**) is the number defined above. It can easily be checked that this result is correct

by multiplying it by **A**: the result is the identity matrix, **I**. It is apparent from this that the inverse cannot be calculated if det(**A**) = 0 because this would involve division by zero. Thus, singular matrices (see above) cannot be inverted.

The inverse of a product can be found as

$$(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$$

provided **A** and **B** are both invertible.

We will not define here how to write down explicitly the elements of the inverse of a matrix for larger matrices. The explicit form rapidly becomes very cumbersome, and accurate numerical calculation of the inverse of large matrices requires special techniques that are available in any computer library.

A1.10. Differentiation of a Matrix

This involves no new ideas. You just differentiate each element of the matrix. Thus, the expression $d\mathbf{A}/dt$ simply means, in the 2 × 2 case,

$$\frac{d\mathbf{A}}{dt} = \begin{bmatrix} da_{11}/dt & da_{12}/dt \\ da_{21}/dt & da_{22}/dt \end{bmatrix} \tag{A8}$$

A1.11. Transpose of a Matrix

Swapping the rows and columns of a matrix is referred to as transposition. If $\mathbf{A} = [a_{ij}]$, then its transpose, denoted $\mathbf{A}^{\mathrm{T}}$, is $\mathbf{A}^{\mathrm{T}} = [a_{ji}]$. In the case of the matrices defined in equation A1, we have, therefore,

$$\mathbf{A}^{\mathrm{T}} = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \qquad \mathbf{C}^{\mathrm{T}} = \begin{bmatrix} c_{11} & c_{21} \\ c_{12} & c_{22} \\ c_{13} & c_{23} \end{bmatrix}$$

The transpose of a product is

$$(\mathbf{AB})^{\mathrm{T}} = \mathbf{B}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}$$

A1.12. Eigenvalues and Eigenvectors of a Matrix

Any square matrix can be represented in a simple form that has nice properties that depend on its eigenvalues and eigenvectors. If **M** is an $n \times n$ matrix, then a nonzero ($n \times 1$) column vector **x** is said to be an *eigenvector* of **M** if there is a scalar, λ, such that

$$\mathbf{Mx} = \lambda\mathbf{x} \tag{A9}$$

Then λ is called the *eigenvalue* corresponding to **x**. Such an eigenvector is not unique because you can multiply **x** by any nonzero scalar and the equation above is still satisfied, with the same value of λ.

Because it makes no difference if any matrix or vector is multiplied by an identity matrix, the above equation can be written as $\mathbf{Mx} = \lambda\mathbf{Ix}$ or

$$(\mathbf{M} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0} \tag{A10}$$

Now, any square matrix, when multiplied by a nonzero vector, can only yield a zero result if it is *singular*, i.e., if its determinant is zero. Thus,

$$|\mathbf{M} - \lambda\mathbf{I}| = 0 \tag{A11}$$

The eigenvalues are the solutions to this equation. When the equation is evaluated algebraically, it turns out to be a polynomial in λ of degree n and so, by a well-known theorem in algebra, it has n solutions, $\lambda_1, \lambda_2, \ldots \lambda_n$. For example, in the case $n = 2$, using equation A7,

$$0 = |\mathbf{M} - \lambda\mathbf{I}| = \begin{vmatrix} m_{11} - \lambda & m_{12} \\ m_{21} & m_{22} - \lambda \end{vmatrix} = (m_{11} - \lambda)(m_{22} - \lambda) - m_{12}m_{21}$$

or

$$\lambda^2 - \lambda(m_{11} + m_{22}) + m_{11}m_{22} - m_{12}m_{21} = 0$$

This is a quadratic equation in λ having two roots λ_1 and λ_2. Note that, from school algebra, we have that the sum of the roots, $\lambda_1 + \lambda_2$, is equal to minus the coefficient of λ in the equation, namely $m_{11} + m_{22}$ and the product of the roots, $\lambda_1\lambda_2$, is equal to the constant term $m_{11}m_{22} - m_{12}m_{21}$, which we recognize as being the determinant det(**M**). These are examples of two quite general results:

1. The sum of the eigenvalues of a square matrix equals the sum of the its diagonal elements, which is known as the *trace* of the matrix.
2. The product of the eigenvalues equals the determinant of the matrix.

Although there are always n roots of an $n \times n$ matrix, they are not necessarily distinct. For example, if **M** is the 2×2 identity matrix, the above equations become $(1 - \lambda)^2 = 0$, so both roots are 1; i.e., $\lambda = 1$ is a *repeated* root. Finding roots of polynomials is not always easy, but there are standard computer programs widely available to find these eigenvalues and the eigenvectors $\mathbf{x}_i$ that go with them. For given λ_i, the corresponding eigenvector satisfies equation A9 or, equivalently, A10. Note that equation A10 is very much like the transpose of equation 10 for finding an equilibrium vector (if you identify $\mathbf{M} - \lambda_i\mathbf{I}$ with **Q**), so it is not too difficult. Remember that the scaling of an eigenvector is arbitrary: any constant times the eigenvector is also an eigenvector, but the arbitrary scaling factors cancel out during subsequent calculations so they are not important for the applications discussed here.

The set of equations

$$\mathbf{Mx}_i = \lambda_i\mathbf{x}_i, \qquad i = 1 \text{ to } n$$

can be written as a single matrix equation, $\mathbf{MX} = \mathbf{X\Lambda}$, or, simply interchanging the two sides of the equation,

$$\mathbf{X\Lambda} = \mathbf{MX} \tag{A12}$$

where $\mathbf{X}$ is a matrix whose columns are the eigenvectors $\mathbf{x}_i$. Thus, $\mathbf{X}$ can be written as

$$\mathbf{X} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_n] \tag{A13}$$

There is one eigenvector for each eigenvalue, so it does not matter which order we put the eigenvalues in, as long as the eigenvectors are kept in the corresponding order. $\mathbf{\Lambda}$ is an $n \times n$ *diagonal* matrix with the eigenvalues λ_i down the diagonal and zeroes everywhere else.

In general, matters can get complicated from here on if some of the eigenvalues are repeated roots; in that case we need something called the Jordan canonical form, which can be found in advanced textbooks on algebra or, in the Markov process case, Cox and Miller (1965, Chapter 3). Fortunately, in the case of reversible Markov processes, these complications do not generally arise. So we will assume, sufficiently for our purpose, that the eigenvalues are all distinct, and then it can be shown that the matrix $\mathbf{X}$ is invertible, and so we can postmultiply equation A12 to obtain

$$\mathbf{X\Lambda X}^{-1} = \mathbf{MXX}^{-1} = \mathbf{MI} = \mathbf{M} \tag{A14}$$

The value of all this comes when we want to raise the matrix $\mathbf{M}$ to some power. For example,

$$\mathbf{M}^2 = \mathbf{X\Lambda X}^{-1}\mathbf{X\Lambda X}^{-1} = \mathbf{X\Lambda I\Lambda X}^{-1} = \mathbf{X\Lambda}^2\mathbf{X}^{-1}$$

It is easy to see that we can keep on doing this, so that

$$\mathbf{M}^r = \mathbf{X\Lambda}^r\mathbf{X}^{-1} \tag{A15}$$

The important thing about this is that, although, in general, raising the matrix $\mathbf{M}$ to the power r is quite difficult, it is very easy for the diagonal matrix $\mathbf{\Lambda}$: the matrix $\mathbf{\Lambda}^r$ is simply another diagonal matrix whose diagonal elements are simply powers of the eigenvalues λ_i^r so you only have to calculate powers of scalars.

It should be noted that, in general, the eigenvalues and the elements of the eigenvectors of $\mathbf{M}$ may be complex numbers. Fortunately, this is not the case in ion-channel models.

There is another way in which we can represent equation A15. To generalize a little, let $\mathbf{D}$ be any $n \times n$ diagonal matrix with diagonal elements d_i, and let $\mathbf{Y}$ denote the inverse $\mathbf{X}^{-1}$, which we now consider as a set of *rows*

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_i \\ \vdots \\ \mathbf{y}_n \end{bmatrix} \tag{A16}$$

Then it is easy to see that the matrix product $\mathbf{DY}$ can be represented as a similar set of rows

$$\mathbf{DY} = \begin{bmatrix} \mathbf{y}_1 d_1 \\ \mathbf{y}_2 d_2 \\ \vdots \\ \mathbf{y}_i d_i \\ \vdots \\ \mathbf{y}_n d_n \end{bmatrix} \tag{A17}$$

Note that, because the d_i are scalars, it does not matter if we write them before the $\mathbf{y}_i$ or after. Then

$$\mathbf{XDY} = [\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_n] \begin{bmatrix} \mathbf{y}_1 d_1 \\ \mathbf{y}_2 d_2 \\ \vdots \\ \mathbf{y}_i d_i \\ \vdots \\ \mathbf{y}_n d_n \end{bmatrix}$$

Now this looks just like a row times a column, similar to example A3, despite the fact that the elements here are vectors rather than scalars. But the nice thing about *partitioned matrices* is that, provided everything is the right shape, they behave formally just like ordinary matrices. Thus, in this case,

$$\mathbf{XDY} = \sum_{i=1}^{n} \mathbf{x}_i \mathbf{y}_i d_i \tag{A18}$$

But $\mathbf{x}_i \mathbf{y}_i$ is a column times a row, similar to example A4, and so the result is an $n \times n$ matrix

$$\mathbf{A}_i = \mathbf{x}_i \mathbf{y}_i \qquad (i = 1 \text{ to } n) \tag{A19}$$

These matrices $\mathbf{A}_i$ are called the *spectral matrices* of the matrix **M.** Then equation A18 can be written as

$$\mathbf{XDX}^{-1} = \mathbf{XDY} = \sum_{i=1}^{n} \mathbf{A}_i d_i \tag{A20}$$

As a particular example, equation A15 can now be written as

$$\mathbf{M}^r = \mathbf{X} \mathbf{\Lambda}^r \mathbf{X}^{-1} = \sum_{i=1}^{n} \mathbf{A}_i \lambda_i^r \tag{A21}$$

Thus, once we have calculated the eigenvalues and eigenvectors, and hence the spectral matrices, of the matrix **M,** we have only to calculate the powers of the (scalar) eigenvalues, multiply them by the constant matrices $\mathbf{A}_i$, and add them instead of doing a lot of matrix multiplications to get $\mathbf{M}^r$.

Appendix 2. Some APL Code

APL is a powerful computer language that uses an "executable notation," which differs from normal mathematical notation but is far more logically consistent. It has the advantage that, when you get used to the notation, the instructions you give the computer are essentially the same as you would write on paper. We illustrate some of the calculations for Sections 5 and 6 to show what is possible. The version we have used is DYALOG APL.

The hard part with any system is finding the eigenvalues and eigenvectors, and any code is too complex to show here, so let us assume there is a function EIGEN that results in a vector of eigenvalues and an **X** matrix of eigenvectors when supplied with a square matrix as argument. Suppose the transition rates shown in equation 4 are already stored in the matrix **Q.**

Identify the index sets of subclasses and create a vector **U** containing five 1's

$$\text{A}{\leftarrow}1\ 2\ \diamond\ \text{B}{\leftarrow}3\ 4\ \diamond\ \text{C}{\leftarrow},4\ \diamond\ \text{F}{\leftarrow}\text{BUC}\ \diamond\ \text{E}{\leftarrow}\text{AUB}\ \diamond\ \text{U}{\leftarrow}5\rho 1 \qquad \text{(A22)}$$

Note that the comma in the above line is important: it makes sure that C, consisting of a single number, is a vector, not a scalar; the $\diamond$ character simply allows several statements on one line. Next

$$y{\leftarrow}0\ 0\ 0\ 0\ 0\ 1\ \diamond\ \text{PINF}{\leftarrow}\text{y}\boxminus\!\!\circledcirc\text{Q},\ 1 \qquad \text{(A23)}$$

finds the equilibrium distribution, $\mathbf{p}(\infty)$, using the method of equation 17 (note that **u** in that equation is **S***y* while **S** is **Q,**1), with the numerical result of equation 18. The remaining numerical results arising from the APL statements shown below are reported in Section 5.

$$\text{PHIS}{\leftarrow}\text{PINF[A]}+.\times\text{Q[A;F]}\div\text{PINF[A]}+.\times\text{Q[A;F]}+.\times\text{U[F]} \qquad \text{(A24)}$$

finds the initial vector $\boldsymbol{\phi}_s$ (see equation 50 and the numerical result of equation 57).

Now let

$$\text{L X}{\leftarrow}\text{EIGEN Q[F;F]}\ \diamond\ \text{TAU}{\leftarrow}\div\text{L}\ \diamond\ \text{SM}{\leftarrow}(\downarrow\circledcirc\text{X})\circ.\times{\cdot\cdot}\downarrow\boxminus\text{X} \qquad \text{(A25)}$$

After that L contains the eigenvalues of $\mathbf{Q}_{\mathcal{F}\mathcal{F}}$, **X** the **X** matrix of eigenvectors, while **A** contains the set of all three matrices $\mathbf{A}_1$ $\mathbf{A}_2$ $\mathbf{A}_3$ by doing each of the vector products in equation 89; the symbol ¨ is the "each operator" in APL and $\boxminus$ is the matrix inversion function. The result is shown in equation 56. The coefficients a_i are then formed into the vector **a,** with three elements in this case, by a version of equation 52.

$$a{\leftarrow}\ -\text{TAU}\times(\subset\text{PHIS})+.\times{\cdot\cdot}\text{A}+.\times{\cdot\cdot}\subset\text{Q[F;F]}+.\times\text{U[F]} \qquad \text{(A26)}$$

with the numerical results shown in equation 56.

To get the distribution of the number of openings per burst, evaluate equation 59 as

$$\text{GAB}{\leftarrow}\ -(\boxminus\text{Q[A;A]})+.\times\text{Q[A;B]}\ \diamond\ \text{GBA}{\leftarrow}\ -(\boxminus\text{Q[B;B]})+.\times\text{G[B;A]} \qquad \text{(A27)}$$

with results shown in equation 65.

The initial vector given by equation 61 is evaluated as

$$\text{PHIB}\leftarrow num\div(num\leftarrow\text{PINF[C]}+.\times\text{Q[C;A]}+\text{Q[C;B]}+.\times\text{GBA})+.\times\text{U[A]} \quad \text{(A28)}$$

where the numerator is stored in *num* to save calculating the same thing twice.

The result is shown in equation 63.

This time the spectral expansion needed is given by

$$\text{RHO X}\leftarrow\text{EIGEN GAB}+.\times\text{GBA} \diamond \text{MU}\leftarrow\div 1-\text{RHO} \diamond \text{SM}\leftarrow(\downarrow\text{X})^{\circ}.\times\ddot{}\downarrow\boxplus\text{X} \quad \text{(A29)}$$

with numerical results for RHO, MU, and SM shown in equations 71–73, respectively. The calculation for MU follows from equation 69.

Now form an identity matrix of shape $k_{\mathcal{A}} \times k_{\mathcal{A}}$

$$\text{I}\leftarrow \text{DIAG } (\rho\text{A})\rho 1$$

The formula for the areas is then given by equation 70, and the numerical results by equation 74:

$$a\leftarrow \text{MU}\times(\subset\text{PHIB})+.\times\ddot{}\,\text{SM}+.\times\ddot{}\subset(\text{I}-\text{GAB}+.\times\text{GBA})+.\times\text{U[A]} \quad \text{(A30)}$$

Comparison of these equations with the corresponding ones in Sections 5 and 6 shows how easy it is (with a few little tricks you have to get used to) to translate the usual mathematical formulas into an equivalent executable notation. Results from the other sections can be obtained in much the same manner.

References

Colquhoun, D., and Hawkes, A. G., 1977, Relaxation and fluctuations of membrane currents that flow through drug-operated channels, *Proc. R. Soc. Lond.* [*B*]**199:**231–262.

Colquhoun, D., and Hawkes, A. G., 1981, On the stochastic properties of ion channels, *Proc. R. Soc. Lond.* [**B**]**211:**205–235.

Colquhoun, D., and Hawkes, A. G., 1982, On the stochastic properties of bursts of single ion channel openings and of clusters of bursts, *Phil. Trans. R. Soc. Lond.* [*B*]**300:**1–59.

Colquhoun, D., and Hawkes, A. G., 1987, A note on correlations in single ion channel records, *Proc. R. Soc. Lond.* [*B*]**230:**15–52.

Cox, D. R., and Miller, H. D., 1965, *The Theory of Stochastic Processes,* Chapman and Hall, London.

Hawkes, A. G., 1984, Complex numbers in APL, *Vector J. Br. APL Assoc.* **1:**107–120.

Hawkes, A. G., and Sykes, A. M., 1990, Equilibrium distributions of finite-state Markov processes, *IEEE Trans. Reliability* **29**(5):592–595.

Hawkes, A. G., Jalali, A., and Colquhoun, D., 1992, Asymptotic distributions of apparent open times and shut times in a single channel record allowing for the omission of brief events, *Phil. Trans. R. Soc. Lond.* [*B*]**337:**383–404.

Huang, C. Y., 1979, Derivation of initial velocity and isotopic exchange rate equations, *Methods Enzymol.* **63:**54–85.

Mirsky, L., 1982, *An Introduction to Linear Algebra,* Dover, New York.

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

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., 1993, *Numerical Recipes: The Art of Scientific Computing,* 2nd ed., Cambridge University Press, Cambridge.

Stephenson, G., 1965, *An Introduction to Matrices, Sets and Groups,* Longman, London.