NUMERICAL METHODS FOR DIFFERENTIAL EQUATIONS IN SYSTEM SIMULATION AND IN PARAMETER ESTIMATION

P.W. HEMKER
Mathematical Centre
Amsterdam
The Netherlands

The numerical simulation of biochemical systems, as well as the fitting of theoretical curves to experimental data, is seriously hampered by the fact that standard methods for the numerical solution of differential equations are not suitable for the solution of the equations that arise from enzyme kinetics. This has led to serious difficulties (cf. Garfinkel and Hess,1964). This is the reason why we set out to systematically investigate numerical methods suitable to approach the socalled "stiff equations".

1, <u>The approximate solution of ordinary differential equations</u>

1.1 Introduction

If we are given a first order differential equation

$$dy/dt = f(t,y) \qquad (1.1)$$

we may represent it graphically as a collection of slopes, for at each value of the independent variable t and the dependent variable y the equation defines a dy/dt. If we are now given a point A, through which our solution is required to pass (the *initial condition*), we may easily sketch this solution by drawing a curve smoothly through the slopes. We then have approximately integrated the differential equation. In fact, the numerical techniques for solving systems of differential equations are only elaborations of this simple graphical technique.

We are directly faced with the major difference between the analytical and the numerical solution of a differential equation. The differential system defines mathematically a unique solution, "the" solution of the system. However, when the differential system is given numerically, the equations and the initial conditions normally involve one or more rounded constants, which have a permissible range of variation; these correspond to a set of possible solutions. Moreover, the numerical processes of obtaining a solution involve errors, increasing further the variation in the possible set of solutions. A numerical procedure picks out a single member of this set.

In most computer libraries, standard routines are available that will perform the integration in a great number of cases. These routines usually are based on a fourth of fifth order Runge-Kutta type algorithm, or on predictor-corrector or rational extrapolation methods. However, in the study of biochemical systems a set of differential equations arises that are particularly difficult to solve by ordinary procedures. The solutions to these equations contain rapidly as well as slowly varying components. They arise when a system is very stable for some kind of perturbations, but much less stable for another kind. This, for instance, is the case in chemical systems, where some reaction rates are much faster than others. The differential equations exhibiting such a behaviour are called *stiff equations*, since they were first encountered during the numerical solution of a mechanical system containing a stiff spring.

As an example of a stiff equation we show the differential equation

$$dy/dt = -2.5\, y + (5t + 3)\,(t + 1)^{-2} \qquad (1.2)$$

Solutions can be found, following the slopes in the t-y-plane.

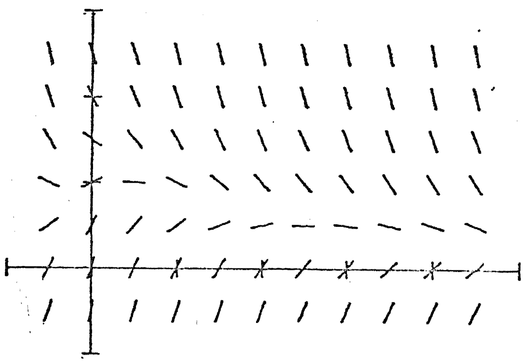

Fig. 1. Slopes of equation (1.2).

A differential equation is stiff when all solutions, corresponding to different initial conditions, rapidly converge to the same set of slowly varying integral curves (the *asymptotic solutions*). In figure 1 a number of slopes are shown for the differential equation (1.2). In figure 2, a number of solutions are shown for the same differential equation. It clearly can be seen that all solutions converge to the same asymptotic solution. It is a feature of stiff equations that the *initial phase* of a solution is characterized by a time constant which has another order of magnitude than the *asymptotic phase*.

These stiff equations, describing phenomena with widely spread time constants, cause difficulties upon computation, because of the requirement of *numerical stability*, i.e. we want numerical perturbations (rounding errors etc.) not to accumulate during the numerical process. Standard methods only are numerically stable when a differential equation is integrated with time steps that are of the same order of magnitude as the shortest time constant of the system under consideration. Thus it will be very time consuming to obtain an asymptotic solution to a stiff differential equation. Since, on the other hand, stiff systems are very stable - in the sense that they are insensitive to some kind of perturbations - , this may be an indication that there are algorithms which remove the difficulties. Clearly, there is no sharp division between stiff and non-stiff equations, and so it remains difficult to combine the ease of standard methods with the power of methods suitable to overcome stiffness. A quantitative description of stiffness therefore is necessary.

1.2 Quantitative description of stiffness

Now we will give a method to describe quantitatively the stiffness of a system of differential equations. Consider the system of differential equation written in vector notation

$$\frac{d}{dt}\vec{y}(t) = \vec{f}(t,\vec{y}). \qquad (1.3)$$

If the vector function $\vec{f}$ is differentiable with respect to $\vec{y}$, we can expand $\vec{f}$ in a Taylor series with respect to $\vec{y}$ at the point $\vec{y}_o$:

$$\frac{d}{dt}\vec{y}(t) = \vec{h}(t) + J(t,\vec{y}_o)(\vec{y}-\vec{y}_o) + \dots , \qquad (1.4)$$

where $\vec{h}(t) = \vec{f}(t,\vec{y}_o)$ is a vector, and $J(t,\vec{y}_o)$ represents the *Jacobian matrix* of the system at the point $(t,\vec{y}_o)$:

$$J(t,\vec{y}_o) = (\partial f_i/\partial y_j)_{t,\vec{y}=\vec{y}_o}. \qquad (1.5)$$

In the case where $\vec{h}(t)$ only slowly varies with t, we obtain a good quantitative description of the local behaviour of the solutions by locating the eigenvalues of the matrix J in the complex plane. In order to explain this, we consider the solution in a neighbourhood of a point $(t_o,\vec{y}_o)$ and we linearize eq. (1.4):

$$(d\vec{y}/dt)_{t=t_o} = \vec{h}(t_o) + J(t_o,\vec{y}_o)(\vec{y}-\vec{y}_o). \qquad (1.6)$$

By assuming that the eigenvalues of $J(t_o,\vec{y}_o)$ all are different, the local analytical solution can be written

$$\vec{y}(t)-\vec{y}(t_o) = \vec{b} + \Sigma c_i\vec{w}_i\, e^{\lambda_i(t-t_o)} , \tag{1.7}$$

where $\{\lambda_i\}$ and $\{\vec{w}_i\}$ are the eigenvalues and eigenvectors of $J(t_o,\vec{y}_o)$ and where $\vec{b}$ and $\{c_i\}$ are determined by the linear equations

$$J(t_o,\vec{y}_o)\vec{b} + \vec{h}(t) = 0$$

$$\vec{b} + \Sigma c_i\vec{w}_i = 0 .$$

Equation (1.7) shows that the time-dependent behaviour of the solutions is mainly determined by the eigenvalues of J (that are the inverse values of the time constants of the system). Only the behaviour induced by $\vec{h}(t)$ and non-linearity have been left out of consideration. A stable system of differential equations will have its eigenvalues in the left half of the complex plane ($\mathrm{Re}\ \lambda_i < 0$). A stiff system is characterised by a wide spread of the values $|\lambda_i|$ ($\mathrm{Re}\ \lambda_i < 0$).

1.3 Numerical stability

A numerical process is called *numerically unstable* if errors induced by the process (e.g. rounding errors) will grow systematically, affecting the results of the calculation in an inadmissible way. A process is called *numerically stable* if an error, once induced, will decrease.

We will illustrate the idea of numerical stability by a very simple but representative example. With the Euler method we solve the single differential equation

$$dy/dt = \lambda y + g(x), \qquad (\lambda<0). \tag{1.8}$$

Choosing a fixed stepsize h, departing from a point y(t), we will calculate y(t+h) at every step. According to the *Euler method* we set

$$\begin{aligned} y(t + h) &= y(t) + hf(t,y(t)) \\ &= y(t) + h\lambda y(t) + hg(t) \\ &= (1+h\lambda)y(t) + hg(t). \end{aligned} \tag{1.9}$$

The value of y(t), already calculated, consists of the true value $\tilde{y}(t)$ and an error ε :

$$y(t) = \tilde{y}(t) + \varepsilon . \tag{1.10}$$

This error ε will cause an error $(y+h\lambda)\varepsilon$ in the calculated value of y(t+h)

$$\begin{aligned} y(t + h) &= (1 + h\lambda)(\tilde{y}(t) + \varepsilon) + hg(t) \\ &= (1 + h\lambda)\tilde{y}(t) + hg(t) + (1+h\lambda)\varepsilon \\ &= \tilde{y}(t + h) + (1 + h\lambda)\varepsilon . \end{aligned} \tag{1.11}$$

The requirement that an error, once induced, decreases is equivalent with

$$|(1+h\lambda)\varepsilon|<|\varepsilon| \qquad \text{or} \qquad 0<h<|\tfrac{2}{\lambda}| \ . \tag{1.12}$$

We see that the requirement of numerical stability gives us a bound for the admissible stepsize. In figure 2 we show some integration steps with $\lambda = -2.5$ and $h=1$.

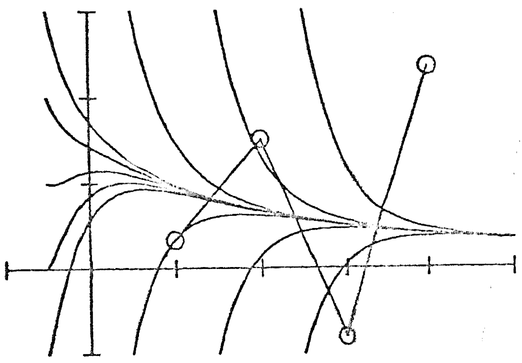

Fig. 2. Unstable integration with the Euler method.

We also show that there are simple methods that do not restrict the stepsize. However, these methods have the disadvantage that in each step of the integration process a (nonlinear) system of equations must be solved. As an example we solve the same differential equation (1.8) with the *backward Euler method*. Now we set

$$\begin{aligned} y(t+h) &= y(t) + h.f(t+h,y(t+h)) \\ &= y(t) + h\lambda y(t+h) + h.g(t+h) \\ &= (1-h\lambda)^{-1}\,(y(t) + h.g(t+h)). \end{aligned} \tag{1.13}$$

Here an error ε in $y(t)$ causes an error $\varepsilon/(1-h\lambda)$ in $y(t+h)$. For this method the condition of numerical stability reads

$$|1-h\lambda|>1. \tag{1.14}$$

Hence in this case numerical stability does not impose any restriction on the stepsize.

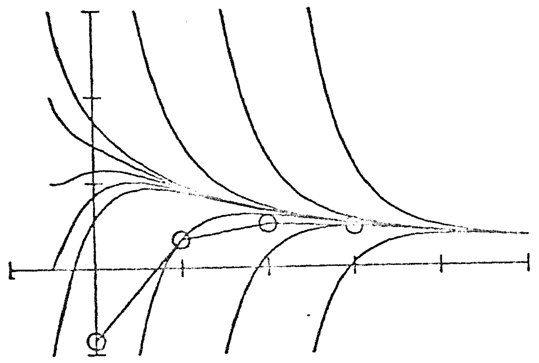

Fig. 3. Stable integration with the backward-Euler method.

The form of our numerical stability conditions (1.12) and (1.14) also give some justification for the suppression of the term $\vec{h}(t)$ in the quantitative description of stiffness.

In the foregoing we only considered the behaviour of an error induced by steps already performed. It is clear that in each step some new errors are also introduced. First, we notice that, e.g. in the Euler method (1.9), the value set for $y(t+h)$ is not a very good approximation to the "real" value

$$y(t+h) = y(t) + h.y'(t) + h^2y''(t)/2 + \ldots \quad .$$

The neglected term (of order h^2) is called the *truncation error*. Since a numerical process must be finite, any method will introduce this kind of error. Secondly, we neglected *rounding errors*: in a computer arithmatic operations introduce errors since every real number is represented with finite precision.

Now we will show how these errors, introduced in each step, all act together. Let ε_n^* denote the total error in the calculated value $y(t_n)$, and let the contribution to ε_n^* caused by ε_{n-1}^* be given by $\alpha_n \varepsilon_{n-1}^*$ (α_n is the *amplification factor*, e.g. $1+h\lambda$ in our example with the Euler method). In every step a new error ε_n is introduced.

Thus we find

$$\varepsilon_n^* = \alpha_n \varepsilon_{n-1}^* + \varepsilon_n \ . \tag{1.15}$$

Our demands upon α_n were $|\alpha_n| \leqslant A < 1$. If we assume that there exists some positive E, so that $|\varepsilon_n| < E$ for all ε_n, we show that the total error of the computation is bounded by $\frac{E}{1-A}$. It follows from the following inequility.

For every n

$$|\varepsilon_n^*| \leqslant |\varepsilon_n| + |\alpha_n| |\varepsilon_{n-1}^*|$$
$$\leqslant |\varepsilon_n| + |\alpha_n| |\varepsilon_{n-1}| + |\alpha_n| |\alpha_{n-1}| |\varepsilon_{n-2}| + \dots$$
$$\leqslant E + AE + A^2E + \dots$$
$$= E\,(1 + A + A^2 \,.\, \dots) \leqslant \frac{E}{1-A}\,.$$

Analogous reasoning shows that $A = 1$ may give a linear growth and $A > 1$ an exponential growth of the total error.

1.4 Application to enzyme kinetics

Let us now take a simple problem from practice and let us give an example of a mathematical analysis. We choose this problem from enzyme kinetics because (1) it describes a system that frequently appears as a subsystem when one simulates real biochemical systems, and (2) it exhibits the typical features that hamper solution by standard methods: nonlinearity and stiff behaviour.

We treat a simple enzymatic reaction of the Michaelis-Menten type. This chemical reaction reads

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} C \overset{k_2}{\rightarrow} E + P\,.$$

An enzyme E combines with a reactant S at one stage and is (irreversibly) regenerated at a subsequent stage of the reaction. We will refer to this system as ESCEP. The rate constants are k_1, k_{-1}, and k_2. As a rule the concentration of E will be much less than the concentration of S. Besides, in many cases we have $k_{-1} >> k_2$. The mass-action law enables us to describe the concentrations S and C as a function of time

$$dS/dt = -k_1(E_o-C)S + k_{-1}C$$
$$dC/dt = k_1(E_o-C)S - (k_2+k_{-1})C\,. \qquad (1.16)$$

As initial conditions we have $S(0)=S_o$ and $C(0)=0$. In order to simplify the notation we write this equation in a dimensionless form by substituting

$$s(t) = S/S_o \qquad c(t) = C/E_o$$
$$\varepsilon = E_o/S_o \qquad \tau = tk_1E_o$$
$$p = (k_2+k_{-1})/(k_1S_o) \qquad q = k_{-1}/(k_1S_o)\,. \qquad (1.17)$$

We obtain

$$ds/d\tau = -(1-c)s + qc$$
$$\varepsilon dc/d\tau = (1-c)s - pc$$
$$s(0) = 1, \quad c(0) = 0. \qquad (1.18)$$

We know the following inequalities

$$\varepsilon, \tau, q > 0; \qquad p > q;$$
$$0 \leq c;\ s \leq 1.$$

Normally $\varepsilon \ll 1$ (ε is a small parameter)

and often $0 < p - q \ll q$.

Apart from that, the numerical values of p, q and ε may differ much in individual cases (cf. Briggs and Haldane, 1925)

In order to show that the system (1.18) is a stiff one, we calculate the Jacobian matrix of the system, together with its trace and its determinant:

$$J = \begin{bmatrix} c-1 & q+s \\ (1-c)/\varepsilon & -(p+s)/\varepsilon \end{bmatrix},$$

$$\mathrm{tr}(J) = -(1-c + (p+s)/\varepsilon),$$

$$\det(J) = (p-q)(1-c)/\varepsilon .$$

The eigenvalues of J being λ_M and λ_m,

obviously $\lambda_M < \lambda_m < 0$ and

$$2(1+\lambda_M/\lambda_m) \geqslant \frac{(\lambda_m + \lambda_M)^2}{\lambda_m \cdot \lambda_M} = \frac{\mathrm{tr}(J)^2}{\det(J)} >$$

$$> \left(\frac{p+s}{\varepsilon}\right)^2 \Big/ \left(\frac{p-q}{\varepsilon}\right) = \frac{(p+s)^2}{\varepsilon(p-q)} . \tag{1.19}$$

Hence, it appears that both eigenvalues are negative and that their ratio λ_M/λ_m is very large. These are the characteristic properties of a stiff differential equation.

This analysis clarifies why the simulation of systems that contain the system ESCEP as a subsystem, often demands an excessive use of computer time when standard routines are used: in order to solve the equations such routines are forced to take time-steps of order $h \approx 1/\lambda_M$, whereas the significant time constant of the system is $1/\lambda_m$.

1.5 An analytical approximation method

In enzyme kinetics some approximate solutions are well known for the system ESCEP, viz. the Briggs-Haldane formula (Briggs and Haldane, 1925), and the Gutfreund formula (Gutfreund, 1965, see also Hemker and Hemker, 1969). A method for the solution of systems of differential equations, in which a highest derivative is multiplied by a small parameter (can be seen in (1.18)), is furnished by the *theory of singular perturbation problems* (see J.D. Cole, 1968). We show that, in the case where ε is a small parameter, this theory will give a combination of the Briggs-Haldane and the Gutfreund formulae as a first approximation to the solution of (1.16). Higher order approximations can be obtained (Heineken et.al.,1967).

We consider system (1.18) and we try to find a solution that is asymptotically correct for $\varepsilon \to 0$. To that end we first take $\varepsilon = 0$ to obtain

$$\begin{aligned} ds/d\tau &= -(1-c)s + qc \\ 0 &= (1-c)s - pc \\ s(0) &= 1,\ c(0) = 0. \end{aligned} \qquad (1.20)$$

Solving the system we get

$$c = \frac{s}{s+p} \qquad (1.21)$$

(i.e. the dimensionless form of the Briggs-Haldane formula) and

$$ds/d\tau = -(p-q).\ \frac{s}{s+p} \qquad (1.22)$$

This single differential equation admits an implicit solution to $s(\tau)$:

$$s(\tau) + p\ \ln(s(\tau)) + (p-q)\tau = 1. \qquad (1.23)$$

This is the first order approximation to $s(\tau)$, which is asymptotically truefor $\varepsilon \to 0$. Given some values for p, q, and τ, it is very easy to compute the numerical value $s(\tau)$ from this formula.

However, with equations (1.23), and (1.21), the second initial condition $c(0)=0$ cannot be satisfied. To match this condition we introduce at $\tau=0$ a *local coordinate* $\zeta=\tau/\varepsilon$. Substituting $\zeta\varepsilon$ into equation (1.18) we get a description of the initial phase of the system

$$\begin{aligned} ds/d\zeta &= -\varepsilon(1-c)s + \varepsilon qc \\ dc/d\zeta &= (1-c)s - pc \\ s(0) &= 1,\ c(0) = 0. \end{aligned} \qquad (1.24)$$

Taking again $\varepsilon = 0$, we obtain

$$\begin{aligned} ds/d\zeta &= 0 \\ dc/d\zeta &= (1-c)s - pc \end{aligned} \qquad (1.25)$$

this admits the solution

$$\begin{aligned} s(\zeta) &= 1 \\ c(\zeta) &= \frac{1}{1+p}\left[1-e^{-(1+p)\zeta}\right] \end{aligned} \qquad (1.26)$$

(i.e. the dimensionless form of the Gutfreund formula).

Now we have to satisfy the condition that the end of the initial phase matches the beginning of the asymptotic phase.

So we have the *matching conditions* (cf. Cole, 1968)

$$\lim_{\zeta\to\infty} s(\zeta) = 1 = \lim_{\tau\to\infty} s(\tau)$$

and (1.27)

$$\lim_{\zeta\to\infty} c(\zeta) = \frac{1}{1+p} = \lim_{\tau\to\infty} c(\tau)$$

The first order approximation - with respect to ε - to the solution of (1.18) is now easily obtained:

$$s(\tau) = s(\tau) \text{ defined by } (1.23)$$
$$c(\tau) = \frac{s(\tau)}{s(\tau)+p} - \frac{1}{1+p} e^{-(1+p)\tau/\varepsilon} . \qquad (1.28)$$

2. A survey of modern numerical techniques

2.1 General remarks

This brief space does not allow us to give a list of all methods (algorithms, features, comparions, etc.) that are available for solving initial value problems. The reader can find an extensive exposition of this kind in Lapidus and Seinfeld (1971). We only want to give a bird's eye view on the main types of integration techniques that are used by the numerical analyst and we will stress those methods that may be of use in the simulation of real (bio)chemical systems. References are given to the literature where methods are explained in more detail and where computer programs are available. Books containing general information on the subject are a.o. Henrici (1962) and Gear (1971).

Any method that solves an initial value problem step by step will approximate the mathematical solution if steps are taken small enough and, at least theoretically, this approximation will become better when smaller time steps are taken. However, numerical stability may command extremely small steps. In order to examine the stability behaviour of a method during the integration of the system of differential equations

$$dy/dt = f(t,y), \qquad (2.1)$$

it is useful to consider the Jacobian matrix

$$J = (\partial f_i/\partial y_j)$$

and its eigenvalues $\{\lambda_j\}$, that also served to quantitate stiffness (see section 1.2). In general this Jacobian matrix depends on t and y, and therefore the eigenvalues are a set of real and conjugate complex numbers, each one depending on t and y. To each method is associated a *stability region*, i.e. a set of complex numbers $h\lambda$ (h stepsize, λ eigenvalue of J) for which the method is numerically stable. Thus, a system of differential equations (2.1) only can be solved with a stepsize h such that all values $\{h\lambda_j\}$ lie inside the stability region of the method.

In section 1 we already have become acquainted with the stability regions of the Euler and the backward-Euler methods. The stability region of the Euler method, given by $|1+h\lambda|<1$ (cf. eq. 1.12) is a disc in the complex $h\lambda$-plane with radius 1 and centre -1. The stability region of the backward-Euler method given by $|1-h\lambda|>1$ (eq. 1.14) is the outside of a disc with radius 1 and centre +1.

Methods which are stable for all $h\lambda$ with Re $h\lambda < 0$ are said to be *A-stable* (Dahlquist, 1963). Methods which are stable for all $h\lambda$ with Re $h\lambda < d < 0$ and for all real values $h\lambda < 0$ are called *stiffly stable* (Gear, 1968).

2.2 Linear multistep methods

A linear k-step method for the solution of initial value problem (2.1) is defined by the vector equation

$$y_{n+1} = h\beta f(t_{n+1}, y_{n+1}) + \Phi_{n+1}, \qquad (2.2)$$

where Φ_{n+1} is a linear combination of values y_{n-i} and $f(t_{n-i}, y_{n-i})$ $(i = 0,1,\ldots, k-1)$ that already have been computed. If $\beta = 0$, the method is *explicit* and *implicit* if $\beta \neq 0$. When a constant stepsize h is used, the formula is normally written

$$\sum_{i=0}^{k} \alpha_i y_{n-i+1} + h\beta_i f(t_{n-i+1}, y_{n-i+1}) = 0. \qquad (2.3)$$

A method is defined by a choice of the parameters α_i and β_i $(i = 0,1,\ldots,k)$ and methods are available which are stable and accurate for $h \to 0$. A comprehensive theory on these methods exists (see e.g. Henrici, 1962) the main results being:

1. the order of accuracy of a stable k-step method cannot exceed k+1
2. all explicit linear multistep methods have a finite stability region
3. the maximum order of an A-stable linear multistep method is 2.

A large number of linear multistep methods have been proposed. However, it seems that three types are of practical interest, each type being available for different values of k. The three types are (cf. equation (2.3)):

1. The *explicit Adams* or *Adams-Bashforth* methods, characterized by $\beta_o = 0, \alpha_i = 0$ $(i = 2,\ldots,k)$. These methods have small stability regions that decrease with increasing k. However, the formulae directly give a value y_{n+1}, and therefore are often used in conjunction with the application of an implicit formula (2.3). The resulting explicit methods are called *predictor-corrector methods*.
2. The *implicit Adams* or *Adams-Moulton* methods characterized by $\alpha_i = 0$ $(i = 2,3,\ldots,k)$. The order of accuracy being k+1, these methods have the highest possible order of accuracy in the class of linear multistep methods. The stability region of these methods is bounded

for k>1, but when stability does not limit the stepsize, these methods may be very efficient.

3. The *stiffly stable methods* (cf. Gear,1968). These implicit methods are characterized by $\beta_i = 0$ ($i = 1,2,\dots,k$) and by the order of accuracy being k. These methods only exist for $k\leqslant 6$. Because of their special stability properties, they are very efficient in the case of stiff equations.

We notice that one has to pay for the nice properties of the implicit methods by the fact that we have to solve a (non-linear) set of algebraic equations at each stage of the integration process. On the other hand, when accuracy or stability questions do not arise, explicit methods may be efficient because of the simplicity of the procedure.

Frequently the use of Runge-Kutta methods is advised in order to find the k-1 values y_i and $f(t_i,y_i)$ ($i = 1,2,\dots,k-1$) that are needed to start the multistep methods. However, a good routine will have the flexability not only to adjust the stepsize, but also to start with a linear 1-step method and to adjust the order of the methods during the integration process. Routines have been published by Gear (1971) in FORTRAN and by Hemker (1971) in ALGOL 60.

2.3 Runge-Kutta methods

Another family of integration formulae are the Runge-Kutta methods. They are of the type

$$k_i = h.f(t_n+\mu_i h \quad , y_n + \sum_{j=1}^{m} \lambda_{ij} k_j) \qquad i=1,\dots,m \qquad (2.4)$$

$$y_{n+1} = y_n + \sum_{j=1}^{m} \theta_j k_j$$

Each method is defined by a choice of the parameters μ_i, θ_i, and λ_{ij} and, again, a great number of methods are available, which are stable and accurate for $h\to 0$. If $\lambda_{ij} = 0$ for $i\leqslant j$, the methods are *explicit*: y_{n+1} can be obtained by the successive computation k_i ($i= 1, \dots,m$). Otherwise, if $\lambda_{ij} \neq 0$ for any $i \leq j$, the method is *implicit* and the computation of k_i ($i = 1,\dots,m$) requires the solution of a (large) system of non-linear equations.

As was the case with linear multistep methods, the explicit formulae only have a finite stability region. However, the large number of parameters (μ_i, θ_i, and λ_{ij}) leaves the possibility to find explicit formulae which have in common their high accuracy but differ by their stability properties. For instance, methods are available which have optimal stability regions for real or imaginary eigenvalues of the Jacobian matrix (v.d.Houwen, 1970a). Also methods are available which are stable with respect both to the small eigenvalues and to some eigenvalues (real or conjugate complex), anywhere in the left half of the complex plane (v.d.Houwen,1970b). The latter type, where one selects the parameters based on some judgement about the solution, are called *exponentially fitted Runge-*

Kutta methods. They represent the asymptotic solution as well as the components of the solution with the known (large negative) eigenvalues. A survey of Runge-Kutta formulae with increased stability regions can be found in v.d.Houwen (1972b).

A more complicated type of Runge-Kutta methods is found if the parameters λ_{ij}, ($j<i$; if $j>i$ then $\lambda_{ij}=0$) are rational expressions of the Jacobian matrix. These methods, called *semi-implicit Runge-Kutta methods*, do not require the solution of a nonlinear set but only of a linear set of algebraic equations at each stage of the integration process. They are A-stable and also may be exponentially fitted to two or three clusters of eigenvalues (v.d.Houwen, 1972a).

When the differential equations are not stiff, the use of simple explicit Runge-Kutta methods is very popular. Many good implementations with automatic stepsize control are available (e.g. Zonneveld,1964). If the equations are stiff and something is known about the position of the eigenvalues in the complex plane, the use of some explicit methods with special stability properties can be recommended. Routines, written in ALGOL 60, are available in Beentjes (1972) and in Dekker (1972). If one is not able to evaluate the Jacobian matrix and if nothing is known about the eigenvalues, implicit Runge-Kutta methods might be used (Ehle, 1968).

2.4 Trapezoidal rule with smoothing,backward Euler

In 1963, Dahlquist proved that no explicit linear multistep method can be A-stable and that the maximum order of an A-stable linear multistep method is 2. Moreover, for fixed stepsize h, the method with the minimum truncation error is the trapezoidal rule

$$y_{n+1} = y_n + h(f(t_n,y_n) + f(t_{n+1},y_{n+1}))/2. \qquad (2.5)$$

Since direct substitution causes convergence difficulties, when applied to stiff equations, the use of Newton-Raphson iteration is recommended to solve this nonlinear equation in y_{n+1}. Hence, the number of iterations necessary for convergence is a measure for the local nonlinearity of the differential equation and therefore can be used to control the stepsize. If we take a sufficiently small stepsize, we may linearize equation (1.1), obtaining

$$f(t,y) = h(t) + J(t,y_o)(y-y_o). \qquad (2.6)$$

Then the execution of a step can be written

$$y_{n+1} = C(hJ)y_n \quad \text{where} \quad C(hJ) = \frac{I-hJ/2}{I+hJ/2} \qquad (2.7)$$

Even though $\|C(hJ)\| \leqslant 1$ for Re $h\lambda < 0$ (i.e. the method is A-stable), we have $C(\lambda h) \to -1$ as $h\lambda \to -\infty$.
That is, the numerical process has a tendency to introduce into the solution some slowly damped oscillations which can be very troublesome during the calculation of the asymptotic phase of the solution. To overcome this difficulty Lindberg

(1971) suggested that one calculates the function values y_{p-1}, y_p, and y_{p+1}. Then one sets $\hat{y}_p = (y_{p-1} + 2y_p + y_{p+1})/4$ for some p and continues the integration from t_p using the smoothed value $\hat{y}_p$.

In conjunction with this smoothing process Lindberg also proposes global extrapolation to increase the accuracy. However, it remains a question whether this technique is appropriate for efficient calculation in system simulation, where only limited accuracy is required.

Another way to avoid the oscillations in an A-stable linear method is to use the backward Euler scheme, which is only first order accurate. Analogous to equation (2.7) one obtains

$$y_{n+1} = C(hJ)y_n, \qquad \text{where } C(hJ) = \frac{I}{I-hJ}, \tag{2.8}$$

which is also A-stable. However, one has that $C(h\lambda) \to 0$ as $\text{Re}\, h\lambda \to -\infty$, which is desirable in the asymptotic phase when the contributions to the solution corresponding to the eigenvalues with large negative real parts may be neglected.

2.5 Exponentially fitted methods

A special exponentially fitted method - cf. exponentially fitted Runge-Kutta methods - is given by Fowler and Warten (1967). Their algorithm is explicit and there is no need for the user to specify the large negative eigenvalue. However, their algorithm is only efficient in the case of one real cluster of eigenvalues.

Another family of exponentially fitted method (where, again, one selects the parameters based on some judgement about the solution) is given in the work of Liniger and Willoughby (1970). A special feature is the A-stability of these exponentially fitted methods. They consider the schemes

$$y_{n+1} = y_n + h(\mu f(t_n,y_n) + (1-\mu)f(t_{n+1},y_{n+1})) \tag{2.9}$$

$$0 \leqslant \mu \leqslant \tfrac{1}{2}$$

and

$$\begin{aligned} y_{n+1} = y_n &+ h((1-a)f(t_n,y_n) + (1+a)f(t_{n+1},y_{n+1}))/2 \\ &-h^2((b-a)\dot{f}(t_n,y_n) + (b+a)\dot{f}(t_{n+1},y_{n+1}))/24 \end{aligned} \tag{2.10}$$

$$0 \leqslant b-a \leqslant 1/3 \;, \quad 1/3 \leqslant b+a \leqslant 2 \;.$$

It should be noted that in the first scheme $\mu = 0$ gives the backward-Euler and $\mu = \frac{1}{2}$ gives the trapezoidal rule. Thus, the choice of μ allows a selection either of these extremes or an intermediate scheme at any point during the integration.

These implicit schemes allow the user to combine the approximate integration of the slowly varying component with the exact integration of some component with a previous known large eigenvalue λ. The schemes are A-stable, hence, some careless estimation of the eigenvalue will not harm but only nullify the labour of exponential fitting.

3. Parameter estimation

3.1 Introduction

A mathematical representation of a biochemical system will often be given by a set of differential equations in which some parameters are unknown. On basis of data obtained in experiments these parameters have to be determined. In order for the parameters to make sense, it is necessary for the equations to be a fair enough representation of the situation in vitro. On the other hand, when the best set of parameters apparently are not compatible with the outcome of the experiments, the mathematical representation is unlikely to be valid. So we are left with two problems

a. The *qualitative problem* of how well the situation in vitro is described by a given set of equations - or, in biochemical terms, what is the mechanism of the reaction -, and
b. The *quantitative problem* of how the parameters can be estimated from experimental data given a likely set of equations.

The first problem is not a mathematical one. It very much concerns the "art" in biochemical research. The quantitative problem, however, is a mathematical one, and its results may serve as a feedback to the biochemist, who has to solve the qualitative problem. In this section we will confine ourselves to the solution of the quantitative problem.

Mathematically stated, the problem is this: a set of n differential equations is given [*)]

$$\frac{d}{dt} y = f(t,y,p) \tag{3.1}$$

where p represents an m-vector of parameters. In the process considered, p has the value p^*, but p^* is not known. Some components of the vector y can be measured for different values of t, but these measurements are affected by some random errors. It is assumed that the form of f is known, together with some statistical properties of the measurement errors. The problem is to deduce an estimate $\hat{p}$ of the vector p^*.

With y_i($1 \le i \le N$) we denote the observed value of some component y at time t_i. Thus the index i identifies an observation and also determines what component of y has been observed. So we have a set of observations $\{y_i\}$, a corresponding set $\{t_i\}$ $(t_1 \le t_2 \le \ldots \le t_N)$ and, for some p, we can compute a set of theoretical values $y(t_i,p)$. The problem now seems to be quite simple: we define the N-vector

*) In this section we use vector notation throughout, so $p \in R^m$, $y \in R \times R^m \to R^n$, $f \in R \times R^n \times R^m \to R^n$ etc..

$$Y(p) = (y(t_i,p) - y_i) \quad 1 \le i \le N \tag{3.2}$$

and we define

$$S(p) = ||Y(p)||^2 = \Sigma_{i=1}^{N} (y(t_i,p)-y_i)^2 \tag{3.3}$$

the sum of the squares of the discrepancies. Using an integration procedure to solve $y(t_i,p)$, we can solve the problem stated by minimizing $S(p)$ using standard techniques. Even when we assume that the minimum is unique and that the function $S(p)$ is the best one to minimize (this can be justified under certain conditions), the question still remains as to how badly conditioned the problem is. I.e., how small a perturbation in some values of y_i will cause how large a variation in the minimizing vector $\bar{p}$. In relation to this question it is clear that not only an estimate of p^* has to be determined but also an estimate of its reliability.

Here we will assume that the measurement errors are statistically independent and that they have a Gaussian distribution with zero mean and variance η^2. Thus the covariance matrix of the vector of errors η is

$$E(\eta\eta^T) = \sigma^2 I \tag{3.4}$$

and the probability density of η is given by

$$p(\eta) = (2\pi\sigma)^{-N/2} \exp(-||\eta||^2/2\sigma^2)$$

3.2 The method

The dependence of $Y(p)$ on p

The solution of the differential equation (3.1) can be considered to be a function of t as well as a function of p. We consider the difference between two adjacent solutions $y_1(t,p)$ and $y_2(t,p+\delta)$ of equation (3.1), both starting at $y_1(0,p) = y_2(0,p+\delta) = c$. We compute the difference between y_1 and y_2 due to the small change in p. The functions y_1 and y_2 are defined by

$$\frac{d}{dt} y_1 = f(t,y_1,p) \qquad y_1(0) = c, \tag{3.5}$$

$$\frac{d}{dt} y_2 = f(t,y_2,p+\delta) \qquad y_2(0) = c. \tag{3.6}$$

Expanding (3.6) in a Taylorseries and keeping only first order terms in δ and in y_2-y_1, we obtain

$$\frac{d}{dt} y_2 = f(t,y_1,p) + FY(y_2-y_1) + FP\,\delta \tag{3.7}$$

where

$$FY = \left(\frac{\partial}{\partial y_1} f(t,y_1,p)\right) \tag{3.8}$$

is an n x n matrix, and

$$FP = (\frac{\partial}{\partial p} f(t,y_1,p)) \qquad (3.9)$$

is an n x p matrix, both matrices being functions of t, p, and y_1, but not of δ or $y_2 - y_1$.

It would be expedient to know how the computable values $y(t_i, p)$ depend upon small variations δ around p. Since equation (3.7) enables us to construct the differential equation which defines the n x m matrix

$$YP = \frac{\partial}{\partial p} y(t,p), \qquad (3.10)$$

we use (3.7) and write

$$\frac{\partial}{\partial p} \frac{d}{dt} y(t,p) = FP + FY.\frac{\partial}{\partial p} y(t,p) \qquad (3.11)$$

or, in shorthand,

$$\frac{d}{dt} YP = FP + FY.YP. \qquad (3.12)$$

This is a system of n x m differential equations. If we solve this system together with system (3.1), we are able to compute:

$$A(p) = \frac{\partial}{\partial p} y(t_i,p), \qquad (3.13)$$

an N x m matrix, giving the dependence of Y(p) (see equation 3.2)) upon variations to p.

Minimizing S(p)

Consider the function S(p) defined by equation (3.3). The value $\bar{p}$ that minimizes S(p) is an estimate of the true value p^*. In equation (3.3) y is a nonlinear function of p. Without some further assumptions the analysis would therefore be too involved to give hope of useful results. This difficulty is dealt with by assuming that p is a reasonably good approximation to $\bar{p}$. Using a generalized Newton-Raphson technique, we linearize the nonlinearity for small departures δp from $\bar{p}$. Suppose that p is a trial vector and δp is the required correction $(p + \delta p = \bar{p})$. The residual vector Y(p) is approximated by a linear function of the parameter

$$Y(p) = Y(\bar{p}-\delta p) = Y(\bar{p}) - A\,\delta p$$

and for the residual function

$$S(\bar{p}) = S(p+\delta p) = ||Y(p+\delta p)||^2$$

$$= ||Y(p) + A(p)\,\delta p\,||^2$$

$$= ||Y\,||^2 + 2\delta p^T A^T Y + \delta p^T A^T A \delta p$$

The approximating function to S(p) has a minimum at the point given by the normal equations

$$A^T(p)A(p)\delta p = -A^T(p)Y(p). \qquad (3.14)$$

If the matrix A^TA is nonsingular, this equation determines δp from Y(p).

In the linear theory p + δp so determined would be the required solution and the minimum value of S attained there would be

$$S(\bar{p}) = ||y(p)||^2 - \delta p^T A^T A \delta p \qquad (3.15)$$

In general, $S(\bar{p} + \delta p)$ will not be the minimal value of S and the whole process is repeated using p + δp as an approximation to $\bar{p}$ for the next iteration.

If it appears that S(p + δp) > S(p), some other techniques can be applied. Firstly the method of steepest descent is recommended with p as a point of departure. For this purpose the gradient vector $r=-A^T(p)\ Y(p)$ is calculated and a new trial step is executed with

$$\delta p = r||r||^2/||Ar||^2$$

If even with this δp it appears that S(p + δp) > S(p), the direction of the step should not be changed, but a relaxation factor can be used, e.g. the step δp may be multiplied by S(p)/(S(p) + S(p + δp)) and a new trial step executed from p.

3.3 Statistics

Let $\bar{p}$ be the final estimate of p so that $S(p) \geq S(\bar{p})$ for all p; we assume that the linear theory holds in a sufficiently large neighbourhood of $\bar{p}$.

For the perturbations η_i of the observed values y_i we assume an $N(0,\sigma^2)$ distribution and so it follows from equation (3.14) that the estimated value $\bar{p}$ will also be normally distributed. We define $\delta p = \bar{p}-p^*$, hence the expectation of δp will be zero when $p=\bar{p}$. We are also interested in the covariance matrix of δp, i.e. the expected value of $\delta p \delta p^T$.

$$E(\delta p \delta p^T) = E((A^TA)^{-1}\ A^TY\ Y^TA\ (A^TA)^{-1}) =$$
$$= (A^TA)^{-1}\ A^T\ E(YY^T)\ A\ (A^TA)^{-1} = \sigma^2\ (A^TA)^{-1}.$$

From this covariance matrix we derive r_{ij}, the correlations between the estimates δp_i and δp_j.

$$r_{ij} = \frac{q_{ij}}{\sqrt{q_{ii}q_{jj}}} \qquad \text{with } q_{ij} = (A^TA)_{ij}\ . \qquad (3.16)$$

By equation (3.14) p is a linear function of Y. Hence its probability density will be Gaussian and will be given by

$$P(\delta p) = ((2\pi\sigma)^m \det((A^TA)^{-1}))^{-\frac{1}{2}} \exp(-\delta p^T A^T A \delta p/2\sigma^2).$$

From (3.15) follows immediately

$$||Y(\underline{\bar{p}} + \underline{\delta p})||^2 = S(\underline{\bar{p}}) + \underline{\delta p}^T A^T A \underline{\delta p}.$$

Now it is clear that $||\underline{Y}||^2/\sigma^2$, $\underline{\delta p}^T A^T A \underline{\delta p}/\sigma^2$, and $S(\underline{p})/\sigma^2$ have a χ^2 distribution with N, m, and $N-m$ degrees of freedom, respectively. An estimate of σ^2 is given by

$$s^2 = S(\bar{p})/(N-m) = ||Y(\bar{p})||^2/(N-m) \tag{3.17}$$

The confidence region at level α is the ellipsoidal region

$$\delta p^T A^T A \delta p \le \frac{m}{N-m} S(\bar{p}) F_\alpha(m,N-m), \tag{3.18}$$

where $F\alpha(n,N-m)$ is the α-point of the F-distribution with m and N-m degrees of freedom. The principal axes of the ellipsoidal region are given by the eigenvectors of A^TA and the length of the axes is $\lambda_i^{-\frac{1}{2}}$ (λ_i is the eigenvalue of the corresponding eigenvector). The confidence limits for each estimate, supposing that the other estimates are exact, are

$$\bar{p}_i \pm \delta p_i$$

where

$$\delta p_i = \sqrt{\frac{m}{N-m} S(\bar{p}) F_\alpha/(A^TA)_{ii}}$$

Other confidence limits for the individual estimates (independently) are

$$\bar{p}_i \pm \delta p_i^*$$

where

$$\delta p_i^* = \sqrt{\frac{m}{N-m} S(\bar{p}) F_\alpha(A^TA)^{-1}_{ii}} \,. \tag{3.20}$$

The geometrical interpretation is that the tangent planes to the ellipsoid with normals to the direction i are at a distance δp_i^* from the centre of the ellipsoid and that the axis i intercepts the ellipsoid at points δp_i from the centre.

3.4 Integration of the differential equations

The system of the differential equations which we have to solve in each iteration step of the optimizing process is, in general, a rather large one. In the system we distinguish two parts
1. (see equation (3.1))

$$\frac{d}{dt} y(t,p) = f(t,y,p) \tag{3.21}$$

a coupled system of n differential equations.

2. (see equation (3.12))

$$\frac{d}{dt} YP = FP + FY.YP. \tag{3.22}$$

This is a set of m systems; each system consists of n differential equations and is coupled with system (3.21).

The structure of the system (3.21 - 3.22) as a whole can be clarified by writing:

1. the system (3.21 - 3.22) as

$$\begin{aligned} \dot{y} &= f \\ \dot{y}_{p1} &= f_{p1} + f_y\, y_{p1} \\ &\vdots \\ \dot{y}_{pm} &= f_{pm} + f_y\, y_{pm} \end{aligned} \tag{3.23}$$

where $y_{pi} = \partial y/\partial p_i$, $f_{pi} = \partial f/\partial p_i$ and

$f_y = \partial f/\partial y$ the Jacobian matrix of the system (3.21), and by writing

2. the Jacobian matrix of the system (3.21 - 3.22) as

$$J = \begin{pmatrix} f_y & 0 & \cdots & 0 \\ f_{py1} & f_y & & 0 \\ \vdots & & \ddots & \\ f_{p} & 0 & & f_y \end{pmatrix},$$

where $f_{pyi} = \partial(\partial f/\partial p_i)/\partial y$.

In this Jacobian matrix the one way coupling of the system is clearly demonstrated. Besides we notice that the eigenvalues of J are all the same as the eigenvalues of f_y, and so the stability behaviours of system (3.23) and system (3.24) are similar.

In order to solve the system of differential equations efficiently, we apply implicit linear multistep methods and we make use of the particular structure mentioned. In each step of the integrating process, equation (3.21) is solved as an independent system. When this part of the integration has been succesfully completed, the m systems of equations (3.22) can be solved with only a little work. We will show this in more detail.

Since we only use implicit linear multistep methods, the result of one integration step during the solution of

$$\dot{y} = f(y)$$

corresponds to the solution of the nonlinear equation

$$y_n = h\beta\, f(y_n) + \phi_n, \tag{3.25}$$

where Φ_n contains the information about a number of completed steps. After the choice of a suitable starting value ${}_0y_n$, this equation is solved with a modified Newton-Raphson method

$${}_{r+1}y_n = {}_ry_n - (I-h\beta f_y)^{-1}({}_ry_m - \Phi_n - h\beta f({}_ry_n)). \tag{3.26}$$

When we solve the system of differential equations

$$\begin{aligned} \dot{y} &= f(y) \\ \dot{w} &= g(y) + f_y\, w \end{aligned}$$

we make use of the one-way coupling of the system. In each step, we have to solve the nonlinear system

$$y_n = h\beta\, f(y_n) + \phi_n \tag{3.27}$$

$$w_n = h\beta\, g(y_n) + h\beta\, f_y(y_n).w_n + \Psi_n \tag{3.28}$$

We do not iterate this system simultaneously, but we solve the nonlinear equation (3.27) by the iteration process (3.26), we substitute the computed value of y_n in (3.28), and we solve the linear equation (3.28) directly. For the solution of this linear equation one needs $(I-h\beta f_y(y_n))^{-1}$: the same factor that will be used in (3.26).

The solution of the system (3.23) is obtained in the same way. In each step of the integration process, the first system of n equations (3.21) is solved by iteration. When this iteration has been completed, each of the m systems of the n equations (3.22) is solved directly. Each one of these m systems needs the L U-decomposition of one and the same matrix $I-h\beta f_y(y_n)$. Moreover, this L U-decomposition can be used again in the next modified Newton-Raphson iteration.

We notice that the possibility of coupling the integration of (3.22) with the integration of (3.21) with this ease, depends crucially on the form of the linear integration formula (3.25). It cannot be done, for instance, with Runge-Kutta methods.

We can use another feature of the integration method. On an interval containing some meshpoints, the linear multistep methods approximate the solution of the differential equation to a polynomial of a certain degree. As a consequence, there is no need to take the meshpoints of our integrating procedure together with the points $\{t_i\}$ where the solution is wanted. The solution is obtained by interpolating the approximating polynomial.

The method is effectively used in a number of problems. Results and an ALGOL 60 program are available in Hemker (1972).

Acknowledgements

I wish to thank Messrs. P.J.v.d.Houwen and G.M.Willems (Mathematical Centre,Amsterdam) and H.C.Hemker (Laboratory for Cardiovascular and Blood Coagulation Biochemistry,Leyden) for reading the manuscript and providing much helpful editorial criticism.

References

Beentjes, P.A. (1972). Report NR 23/72
(Mathematical Centre,Amsterdam).

Briggs, G.E., Haldane, H.B.S. (1925). Bioch.J. 19, 338.

Cole, J.D. (1968). Perturbation methods in applied mathematics.
(Blaisdell Publ. Comp., Waltham, Mass .).

Dahlquist, G. (1963). BIT 3, 27.

Dekker, K. (1972). Report NR 25/72
(Mathematical Centre,Amsterdam).

Ehle, B.L. (1968). BIT 8, 276.

Fowler, M.E. and Warten, R.M. (1967). IBM J.Res.Develop. 11,537.

Gear, C.W. (1968). Proc.IFIP Congr. 1968, p. 187.

- (1971). Numerical initial value problems in ordinary differential equations (Prentice-Hall, Inc. Englewood Cliffs, N.J.).

Garfinkel, D. and Hess, B. (1964). J.Biol. Chem. 239, 971.

Gutfreund, H. (1965).An introduction to the study of enzymes
(Blackwell Scientific Publications, Oxford).

Heineken, F.G., Tsuchiya, M. and Aris, R. (1967). Math. Biosc. 1, 95.

Hemker, H.C. and Hemker, P.W. (1969). Proc.Roy.Soc. B 173, 411.

Hemker, P.W. (1971). Report MR 128/71.
(Mathematical Centre,Amsterdam).

- (1972). Report MR 134/72
(Mathematical Centre,Amsterdam).

Henrici, P. (1962). Discrete variable methods in ordinary differential equations (J.Wiley and Sons, New York).

Houwen, P.J.v.d. (1970a). Report TW 119
(Mathematical Centre,Amsterdam).

- (1970b). Report TW 122
(Mathematical Centre,Amsterdam).

- (1972a). Report TW 132
(Mathematical Centre,Amsterdam).

- (1972b). To appear in Numerische Mathematik.

Lapidus, L. and Seinfeld, J.H. (1971). Numerical solution of ordinary differential equations.
(Acad. Press, New York).

Lindberg, B. (1971). BIT 11, 29.

Liniger, W. and Willoughby, R.A. (1970). SIAM J.Num.Anal. 7,47.

Zonneveld, J.A. (1964). Automatic numerical integration.
(Mathematical Centre,Amsterdam).

BIFURCATION THEORY IN BIOCHEMICAL DYNAMICS

OKAN GUREL
IBM Corporation, White Plains, New York 10604
U. S. A.

INTRODUCTION

Biological molecules are building blocks of both static and dynamic aspects of the biological systems. This type of classification may not be quite proper for the reason that the same molecule may play both roles in a global sense. If it is necessary for a structural molecule m_i to take part in a dynamic activity, m_i might be expressed as being included in the dynamic model, thus m_i becomes a dynamic building block as well as a static one.

The dynamics of these molecules can be expressed as differential equations with some parameters. The singular and periodic solutions of these equations and their stability properties can be studied to determine the topological characteristics of the solutions. The theory of bifurcations is concerned with the changes which occur in the topological structure of a dynamic system in a particular region where the system itself, namely the set of parameters in the right-hand sides of the system equations is altered. The term bifurcation generally refers to those changes in topological structure that occur at the bifurcation values of the parameters.

The aim of this paper is to point out the relationship between the topology of dynamics of molecules and that of dynamics of cells by referring to various bifurcations.

BIOLOGICAL INTEGRITY

If is not conflicting to assume that the *space of molecules* M consists of m_i, the biochemical molecules of both static and dynamic aspects of the biological systems. The dynamics of these molecules in an abstract form is defined by the mapping $f:M \to M$. Therefore, f corresponds to a vector field over the molecular space M. It should also be noted that for a dynamic process defined by f, some other elements, a subset of biochemical molecules, such as enzymes or metalloproteins, or some physical entities, may form the *parameter space* Γ, thus the dynamics of molecules is denoted by $f_\Gamma: M \to M$, a mapping depending on the parameter set Γ.

In a similar fashion, the *space of cells*, at the cellular level, is denoted by C. The mapping $g:C \to C$ would correspond to the dynamics of the cell space. It may be the case that the biochemical or physical parameters forming the set Ω are involved in the vector field g over C such that the dynamics of the cell space is in an abstract form $g_\Omega: C \to C$. The parameter space at the cellular level may involve the parameter space Γ and some of the elements of the space of molecules such that the dynamics at the cellular level would be $g_{\Gamma\Omega M}: C \to C$. In general terms, what we have is that the mapping relating M and C spaces denoted by Π would complete the diagram

$$\begin{tikzcd}
M \arrow[r, "f"] \arrow[d, "\Pi"'] \arrow[dr, dashed] & M \arrow[d, "\Pi"] \\
C \arrow[r, "g"'] & C
\end{tikzcd}$$

such that $\Pi f: M \to C$ or $g\Pi: M \to C$, thus $\Pi f = g\Pi$. The existence of the mapping Π, which may be