The Dynamics of Runge–Kutta Methods

Julyan H. E. Cartwright & Oreste Piro *
School of Mathematical Sciences
Queen Mary and Westfield College
University of London
Mile End Road
London E1 4NS
U.K.

The first step in investigating the dynamics of a continuous-time system described by an ordinary differential equation is to integrate to obtain trajectories. In this paper, we attempt to elucidate the dynamics of the most commonly used family of numerical integration schemes, Runge–Kutta methods, by the application of the techniques of dynamical systems theory to the maps produced in the numerical analysis.

QMW preprint DYN #91-9, Int. J. Bifurcation and Chaos, **2**, 427–449, 1992

*A CONICET (Consejo Nacional de Investigaciones Cientificas y Técnicas de Argentina) Fellow. Present address: Institut Nonlinéaire de Nice, Université de Nice—Sophia Antipolis, Parc Valrose, 06034 Nice Cédex, France.

1. Introduction

NUMERICAL solution of ordinary differential equations is the most important technique in continuous time dynamics. Since most ordinary differential equations are not soluble analytically, numerical integration is the only way to obtain information about the trajectory. Many different methods have been proposed and used in an attempt to solve accurately various types of ordinary differential equations. However there are a handful of methods known and used universally (i.e., *Runge–Kutta*, *Adams–Bashforth–Moulton* and *Backward Differentiation Formulae* methods). All these *discretize* the differential system to produce a *difference equation* or *map*. The methods obtain different maps from the same differential equation, but they have the same aim; that the dynamics of the map should correspond closely to the dynamics of the differential equation. From the Runge–Kutta family of algorithms come arguably the most well-known and used methods for numerical integration (see, for example, Henrici [1962], Gear [1971], Lambert [1973], Stetter [1973], Chua & Lin [1975], Hall & Watt [1976], Butcher [1987], Press *et al.* [1988], Parker & Chua [1989], or Lambert [1991]). Thus we choose to look at Runge–Kutta methods to investigate what pitfalls there may be in the integration of nonlinear and chaotic systems.

We examine here the initial-value problem; the conditions on the solution of the differential equation are all specified at the start of the trajectory—they are initial conditions. This is in contrast to other problems where conditions are specified both at the start and at the end of the trajectory, when we would have a (two-point) boundary-value problem.

Problems involving ordinary differential equations can always be reduced to a system of first-order ordinary differential equations by introducing new variables which are usually made to be derivatives of the original variables. Thus for generality, we consider the non-autonomous initial value problem

$$\begin{aligned} y' &= f(x,y), \\ y(a) &= \alpha, \qquad a \leqslant x, \end{aligned} \tag{1}$$

where y' represents dy/dx. This can be either a single equation $(x,y) \in (\mathbb{R},\mathbb{C})$ or, more generally, a coupled system of equations $(x,y) \in (\mathbb{R},\mathbb{C}^m)$ (often, we will have $y \in \mathbb{R}^m$):

$$\begin{aligned} {}^1y' &= {}^1f(x,{}^1y,{}^2y,\ldots,{}^my) \\ {}^2y' &= {}^2f(x,{}^1y,{}^2y,\ldots,{}^my) \\ &\;\;\vdots \\ {}^my' &= {}^mf(x,{}^1y,{}^2y,\ldots,{}^my). \end{aligned} \tag{2}$$

The variable x often represents time. Almost all numerical methods for the initial value problem are addressed to solving it in the above form.[1]

[1] We use the unorthodox notation ${}^{m+1}y$ etc. to avoid any confusion with the iterates of a map.

A non-autonomous system $y' = f(x, y)$ can always be transformed into an autonomous system $y' = f(y)$ of dimension one higher by letting $x \equiv {}^{m+1}y$, so that we add the equation ${}^{m+1}y' = 1$ to the system and ${}^{m+1}y(a) = a$ to the initial conditions. In this case, however, we will have unbounded solutions since ${}^{m+1}y \to \infty$ as $x \to \infty$. This can be prevented for non-autonomous systems that are periodic in x by identifying planes of constant ${}^{m+1}y$ separated by one period, so that the system is put onto a cylinder. We are usually interested in one of the two cases above: either an autonomous system, or a non-autonomous system that is periodic in x. In these cases, we can define the concepts of the limit sets of the system and their associated basins of attraction which are so useful in dynamics.

It is known that sufficient conditions for a unique, continuous, differentiable function $y(x)$ to exist as a solution to this problem are that $f(x, y)$ be defined and continuous and satisfy a *Lipschitz condition* in y in $\mathcal{R} = [a, b] \times (-\infty, \infty)^m$. The Lipschitz condition is that

$$\|f(x, y) - f(x, \tilde{y})\| \leqslant L\|y - \tilde{y}\|, \qquad \forall (x, y), (x, \tilde{y}) \in (\mathbb{R}, \mathbb{C}^m). \tag{3}$$

Here L is the Lipschitz constant which must exist for the condition to be satisfied. We shall always assume that such a unique solution exists.

Our aim is to investigate how well Runge–Kutta methods do at modelling ordinary differential equations by looking at the resulting maps as dynamical systems. Chaos in numerical analysis has been investigated before: the midpoint method in the papers by Yamaguti & Ushiki [1981] and Ushiki [1982], the Euler method by Gardini *et al.* [1987], the Euler method and the Heun method by Peitgen & Richter [1986], and the Adams–Bashforth–Moulton methods in a paper by Prüfer [1985]. These studies dealt with the chaotic dynamics of the maps produced in their own right, without relating them to the original differential equations.

In recent papers by Iserles [1990] and Yee *et al.* [1991], the connection is examined between a map and the differential equation that it models. Other studies by Kloeden & Lorenz [1986], and Beyn [1987a; 1987b], concentrate on showing how the limit sets of the map are related to those of the ordinary differential equations. Sauer & Yorke [1991] use shadowing theory to find orbits of the map which are shadowed by trajectories of the differential equation.

We bring together here all the strands in these different papers, and extend the examination of the connection between the map and the differential equation from our viewpoint as dynamicists. This topic has begun to catch the awareness of the scientific community lately (see for example Stewart [1992]), and several of the papers we discuss appeared after the initial submission of this work; we have included comments on them in this revised version.

2. Derivation of Runge–Kutta methods

RUNGE–KUTTA methods compute approximations Y_i to $y_i = y(x_i)$, with initial values $Y_0 = y_0 = \alpha$, where $x_i = a + ih$, $i \in \mathbb{Z}^+$, using the Taylor series expansion

$$y_{n+1} = y_n + hy'_n + \frac{1}{2}h^2 y''_n + \cdots + \frac{1}{p!}h^p y_n^{(p)} + O(h^{p+1}) \tag{4}$$

so if we term $f(x_n, y_n) = f_n$ etc. :

$$y_{n+1} = y_n + hf_n + \frac{1}{2}h^2 \left(\frac{df}{dx}\right)_n + \cdots + \frac{1}{p!}h^p \left(\frac{d^{p-1}f}{dx^{p-1}}\right)_n + O(h^{p+1}). \tag{5}$$

h is a non-negative real constant called the *step length* of the method.

To obtain a q-stage Runge–Kutta method (q function evaluations per step) we let

$$Y_{n+1} = Y_n + h\phi(x_n, Y_n; h), \tag{6}$$

where

$$\phi(x_n, Y_n; h) = \sum_{i=1}^{q} \omega_i k_i, \tag{7}$$

so that

$$Y_{n+1} = Y_n + h\sum_{i=1}^{q} \omega_i k_i, \tag{8}$$

with

$$k_i = f\left(x_n + h\alpha_i, Y_n + h\sum_{j=1}^{i-1} \beta_{ij} k_j\right) \tag{9}$$

and $\alpha_1 = 0$ for an explicit method, or

$$k_i = f\left(x_n + h\alpha_i, Y_n + h\sum_{j=1}^{q} \beta_{ij} k_j\right) \tag{10}$$

for an implicit method. For an explicit method, Eq.(9) can be solved for each k_i in turn, but for an implicit method, Eq.(10) requires the solution of a nonlinear system of k_is at each step. The set of explicit methods may be regarded as a subset of the set of implicit methods with $\beta_{ij} = 0$, $j \geqslant i$. Explicit methods are obviously more efficient to use, but we shall see that implicit methods do have advantages in certain circumstances.

For convenience, the coefficients α, β, and ω of the Runge–Kutta method can be written in the form of a *Butcher array*:

$$\begin{array}{c|c} \alpha & \mathrm{B} \\ \hline & \omega^T \end{array} \tag{11}$$

where $\alpha = [\alpha_1, \alpha_2 \ldots \alpha_q]^T$, $\omega = [\omega_1, \omega_2 \ldots \omega_q]^T$ and $\mathrm{B} = [\beta_{ij}]$.

Runge–Kutta schemes are *one-step* or *self-starting* methods; they give Y_{n+1} in terms of Y_n only, and thus they produce a one-dimensional map if they are dealing with a single differential equation. This may be contrasted with other popular schemes (the Adams–Bashforth–Moulton and Backward Differentiation Formulae methods), which are *multistep* methods; Y_{n+k} is given in terms of Y_{n+k-1} down to Y_n. Multistep methods give rise to multi-dimensional maps from single differential equations.

A method is said to have order p if p is the largest integer for which

$$y(x+h) - y(x) - h\phi(x, y(x); h) = O(h^{p+1}). \tag{12}$$

For a method of order p, we wish to find values for α_i, β_{ij} and ω_i with $1 \leqslant (i,j) \leqslant p$ so that Eq.(8) matches the first $p+1$ terms in Eq.(4). To do this we Taylor expand Eq.(8) about (x_n, Y_n) under the assumption that $Y_n = y_n$, so that all previous values are exact, and compare this with Eq.(4) in order to equate coefficients.

For example, the (unique) first-order explicit method is the well-known Euler scheme

$$Y_{n+1} = Y_n + hf(x_n, Y_n). \tag{13}$$

Let us derive an explicit method with $p = q = 2$, that is, a two-stage, second-order method. From Eq.(5) we have

$$y_{n+1} = y_n + hf_n + \frac{1}{2}h^2\left(\frac{df}{dx}\right)_n + O(h^3), \tag{14}$$

so expanding this,

$$y_{n+1} = y_n + hf_n + \frac{1}{2}h^2\left(\left(\frac{\partial f}{\partial x}\right)_n + \left(\frac{\partial f}{\partial y}\right)_n f_n\right) + O(h^3). \tag{15}$$

From Eq.(8), and assuming previous exactness,

$$Y_{n+1} = y_n + h\omega_1 k_1 + h\omega_2 k_2. \tag{16}$$

We can choose $\alpha_1 = 0$ so

$$\begin{aligned}
k_1 &= f(x_n, y_n) = f_n, && (17)\\
k_2 &= f(x_n + h\alpha_2, y_n + h\beta_{21}k_1) && (18)\\
&= f(x_n, y_n + h\beta_{21}k_1) + h\alpha_2\frac{\partial}{\partial x}f(x_n, y_n + h\beta_{21}k_1) + O(h^2) && (19)\\
&= f_n + h\alpha_2\left(\frac{\partial f}{\partial x}\right)_n + h\beta_{21}\left(\frac{\partial f}{\partial y}\right)_n f_n + O(h^2), && (20)
\end{aligned}$$

and Eq.(16) becomes

$$Y_{n+1} = y_n + h\omega_1 f_n + h\omega_2\left(f_n + h\alpha_2\left(\frac{\partial f}{\partial x}\right)_n + h\beta_{21}\left(\frac{\partial f}{\partial y}\right)_n f_n\right) + O(h^3). \tag{21}$$

We can now equate coefficients in Eqs.(15) and (21) to give:

$$
\begin{aligned}
[hf_n] &\ :\ \omega_1 + \omega_2 = 1, &(22)\\
\left[h^2\left(\frac{\partial f}{\partial x}\right)_n\right] &\ :\ \omega_2\alpha_2 = \frac{1}{2}, &(23)\\
\left[h^2\left(\frac{\partial f}{\partial y}\right)_n f_n\right] &\ :\ \omega_2\beta_{21} = \frac{1}{2}. &(24)
\end{aligned}
$$

This is a system with three equations in four unknowns, so we can solve in terms of (say) ω_2 to give a one-parameter family of explicit two-stage, second-order Runge–Kutta methods:

$$
\begin{aligned}
Y_{n+1} &= Y_n + h\left[(1-\omega_2)k_1 + \omega_2 k_2\right], &(25)\\
k_1 &= f(x_n, Y_n), &(26)\\
k_2 &= f\left(x_n + \frac{h}{2\omega_2}, Y_n + \frac{h}{2\omega_2}k_1\right). &(27)
\end{aligned}
$$

Well-known second-order methods are obtained with $\omega_2 = 1/2$, $3/4$ and 1. When $\omega_2 = 0$, the equation collapses to the first-order Euler method.

It is easy to see that we could not have obtained a third-order method with two stages, and in fact it is a general result that an explicit q-stage method cannot have order greater than q, but this is an upper bound that is realized only for $q \leqslant 4$. The minimum number of stages necessary for an explicit method to attain order p is still an open problem. Calling this $q_{\min}(p)$, the present knowledge [Butcher, 1987; Lambert, 1991] is:

p	1	2	3	4	5	6	7	8	9	10
$q_{\min}(p)$	1	2	3	4	6	7	9	11	$12 \leqslant q_{\min} \leqslant 17$	$13 \leqslant q_{\min} \leqslant 17$

One can see from the table above the reason why fourth-order methods are so popular, because after that, one has to add two more stages to the method to obtain any increase in the order. It is not known exactly how many stages are required to obtain a ninth-order or tenth-order explicit method. We only know that somewhere between twelve and seventeen stages will give us a ninth-order explicit method, and somewhere between that number and seventeen stages will give us a tenth-order explicit method. Nothing is known for explicit methods of order higher than ten. In contrast to explicit Runge–Kutta methods, it is known that for an implicit q-stage Runge–Kutta method, the maximum possible order $p_{\max}(q) = 2q$ for any q. It should be noted that the order of a method can change depending on whether it is being applied to a single equation or a system, and depending on whether or not the problem is autonomous (see, for example, Lambert [1991]).

Derivation of higher-order Runge–Kutta methods using the technique above is a process involving a large amount of tedious algebraic manipulation which is both time consuming and error prone. Using computer algebra removes the latter problem, but not the former, since finding higher-order methods involves solving larger and larger coupled systems of polynomial equations. This defeats Maple running on a modern workstation at $q = 5$. To overcome this problem a very elegant theory has been developed by Butcher which enables one to establish the conditions for a Runge–Kutta method, either explicit or implicit, to have a given order (for example the conditions given in Eqs.(22)–(24)). We shall merely mention here that the theory is based on the algebraic concept of *rooted trees*, and we refer you to books by Butcher [1987], and Lambert [1991] for further details.[2]

[2] A Mathematica package implementing Butcher's method for obtaining order conditions is now distributed as standard with version 2 of Mathematica.

3. Accuracy

THERE are two types of error involved in a Runge–Kutta step: *round-off* error and *truncation* error (also known as *discretization* error). Round-off error is due to the finite-precision (floating-point) arithmetic usually used when the method is implemented on a computer. It depends on the number and type of arithmetical operations used in a step. Round-off error thus increases in proportion to the total number of integration steps used, and so prevents one from taking a very small step length. Normally, round-off error is not considered in the numerical analysis of the algorithm, since it depends on the computer on which the algorithm is implemented, and thus is external to the numerical algorithm. Truncation error is present even with infinite-precision arithmetic, because it is caused by truncation of the infinite Taylor series to form the algorithm. It depends on the step size used, the order of the method, and the problem being solved.

An obvious requirement for a successful numerical algorithm is that it be possible to make the truncation error involved as small as is desired by using a sufficiently small step length: this concept is known as *convergence*. A method is said to be convergent if

$$\lim_{\substack{h\to 0\\ nh=x-a}} Y_n = y_n. \tag{28}$$

Notice that nh is kept constant, so that x_n is always the same point and a sequence of approximations Y_n converges to the analytic answer y_n as the step length is successively decreased. This is called a *fixed-station* limit. A concept closely related to convergence is known as *consistency*; a method is said to be consistent (with the initial value problem) if

$$\phi(x_n, y_n; 0) = f(x_n, y_n), \tag{29}$$

where $\phi(x, y; h)$ is as defined in Eq.(7). Inserting the consistency condition of Eq.(29) into Eq.(7) we obtain

$$\sum_{i=1}^{q} \omega_i = 1 \tag{30}$$

as the necessary and sufficient condition for Runge–Kutta methods to be consistent. Looking back at Eq.(22), we can see that we satisfied this condition in deriving the family of second-order explicit methods, and in fact it turns out to be automatically satisfied when the method has order one or higher. It is known that consistency is necessary and sufficient for convergence of Runge–Kutta methods, so all Runge–Kutta methods are convergent. We provide a proof of this in Appendix A.1.

The two crucial concepts in the analysis of numerical error are *local* error and *global* error. Local error is the error introduced in a single step of the integration routine, while global error is the overall error caused by repeated application of the integration formula. It is obviously the global error that we wish to know about when integrating a trajectory, however it is not possible to estimate anything other than bounds which are usually orders of magnitude too large, and so we must content ourselves with estimating the local error. Local and global error are sometimes defined to include round-off error and sometimes not. We do not include round-off error and to avoid any ambiguity we term the local and global error thus defined local and global truncation error.

Global truncation error at x_{n+1} is

$$e_{n+1} = \|y_{n+1} - Y_{n+1}\|, \tag{31}$$

while local truncation error is

$$T_{n+1} = \|y_{n+1} - y_n - h\phi(x_n, y_n; h)\|. \tag{32}$$

If we assume that $Y_n = y_n$, i.e., no previous truncation errors have occurred, then $T_{n+1} = \|y_{n+1} - Y_{n+1}\|$. So if the previous truncation error is zero, the local truncation error and the global truncation error are the same. Comparing Eq.(32) with Eq.(12), we can see that a pth-order method has local truncation error $O(h^{p+1})$. We can write Eq.(31) as

$$\begin{aligned}
e_{n+1} &= \|y_{n+1} - Y_{n+1}\| & (33)\\
&= \|y_{n+1} - Y_n - h\phi(x_n, Y_n; h)\| & (34)\\
&\leqslant \|y_{n+1} - y_n - h\phi(x_n, y_n; h)\| + \|y_n - Y_n\| & \\
&\quad + h\|\phi(x_n, y_n; h) - \phi(x_n, Y_n; h)\| & (35)\\
&\leqslant T_{n+1} + e_n + hLe_n. & (36)
\end{aligned}$$

Thus

$$\begin{aligned}
e_n &\leqslant (1 + hL)^n e_0 + \frac{(1 + hL)^n - 1}{hL} T_{n+1} & (37)\\
&\leqslant \left(\frac{(1 + hL)^n - 1}{L}\right) \frac{T_{n+1}}{h}, & (38)
\end{aligned}$$

since $e_0 = 0$. As $T_{n+1} = O(h^{p+1})$, we can now see that the global truncation error is $O(h^p)$.

We can write the local truncation error as

$$T_{n+1} = \Psi(y_n) h^{p+1} + O(h^{p+2}). \tag{39}$$

The term in h^{p+1} is the *principal* local truncation error, and $\Psi(y_n)$ is a function of the *elementary differentials* of order $p + 1$, evaluated at y_n. Elementary differentials are the building blocks of the Butcher theory mentioned earlier. (The coefficients of h^p in Eq.(15) are elementary differentials of order p.)

Practical codes based on Runge–Kutta or other numerical techniques rely on error estimates to exercise control of the step length in order to produce good results. This *step control* to maintain accuracy requirements may be based on *step doubling*, otherwise known as *Richardson extrapolation*. Each step is taken twice, once with step length h, and then again with two steps of step length $\frac{1}{2}h$. The difference between the two new Ys gives an estimate of the principal local truncation error of the form

$$\frac{Y_{n+1} - \tilde{Y}_{n+1}}{2^{p+1} - 1}. \tag{40}$$

A new step length can then be used for the next step to keep the principal local truncation error within the required bounds. A better technique, because it is less computationally expensive, is to use a Runge–Kutta method that has been specially developed to provide an estimate of the principal local truncation error at each step. This may be done by embedding a q-stage, pth-order method within a $(q+1)$-stage, $(p+1)$th-order method. Runge–Kutta–Merson and Runge–Kutta–Fehlberg are examples of algorithms using this *embedding estimate* technique. It should however be noted that the error estimates provided by some of the commonly used algorithms are not valid when integrating nonlinear equations. For example, Runge–Kutta–Merson was constructed for the special case of a linear differential system with constant coefficients, and the error estimates it provides are only valid in that rare case. It usually overestimates the error, which is safe but inefficient, but sometimes it underestimates the error, which could be disastrous. Thus some care has to be taken to ensure that the embedding algorithm used will provide suitable error estimates. As well as varying the step length, some codes based on Runge–Kutta methods may also change between methods of different orders depending on the error estimates being obtained. These *variable-step, variable-order* (VSVO) Runge–Kutta based codes are at present the last word in numerical integration.

The fact that codes based on Runge–Kutta methods use estimates of the principal local truncation error, which is proportional to h^{p+1}, rather than estimates of the local truncation error, which is $O(h^{p+1})$, can be significant. The principal local truncation error is usually large in comparison with the other parts of the local truncation error, in which case we are justified in using principal local truncation error estimates to set the step length. However, this is not always the case, and so one should be wary. The phenomenon of *B-convergence* [Lambert, 1991] shows that the other elements in the local truncation error can sometimes overwhelm the principal local truncation error, and the code could then produce incorrect results without informing the user.

4. Absolute Stability

IF the step length used is too small, excessive computation time and round-off error result. We should also consider the opposite case, and ask whether there is any upper bound on step length. Often there is such a bound, and it is reached when the method becomes numerically unstable: the numerical solution produced no longer corresponds qualitatively with the exact solution because some bifurcation has occurred.

The traditional criterion for ensuring that a numerical method is stable is called *absolute stability*. Absolute-stability analysis of Runge–Kutta and other numerical methods is carried out using the linear model problem

$$\begin{aligned} y' &= \lambda y, \\ y(a) &= \alpha, \qquad a \leqslant x, \end{aligned} \tag{41}$$

where λ is complex. This has the analytical solution

$$y(x) = \alpha e^{\lambda(x-a)}. \tag{42}$$

The problem has a stable fixed point at $y = 0$ for $\mathrm{Re}(\lambda) < 0$.

For systems of equations we generalize to the problem

$$\begin{aligned} y' &= Ay, \\ y(a) &= \alpha, \qquad a \leqslant x, \end{aligned} \tag{43}$$

where A is a matrix with distinct eigenvalues all lying in the negative half-plane so that again we have a stable fixed point at $y = 0$.

We are interested in these linear model problems since they underlie the theory of the classification of fixed points. Since its eigenvalues are distinct, A is diagonalizable and we can reduce Eq.(43) to a set of independent equations of the form of Eq.(41). λ_i are then the eigenvalues of the Jacobian matrix. This is why we allow λ to be complex above; higher-dimensional systems may have complex eigenvalues, so we need to know the behaviour of the numerical solutions in complex $h\lambda$-space.

The region of absolute stability for a method is then the set of values of h (real and non-negative) and λ (complex) for which $Y_n \to 0$ as $n \to \infty$, i.e., for which the fixed point at the origin is stable. Thus we want the set of values of h and λ for which $|S| \leqslant 1$ where S, the stability function, is the eigenvalue of the Jacobian of the Runge–Kutta map evaluated at the fixed point. In the case of systems, the modulus of the stability function is given by the *spectral radius* of the Jacobian of the Runge–Kutta map (the absolute value of the largest eigenvalue of the Jacobian) at the fixed point.

For example, let us derive the equation of the region of absolute stability for the family of Runge–Kutta methods derived earlier and presented in Eq.(25). Using the linear model problem we have

$$Y_{n+1} = Y_n + h\left[(1-\omega_2)\lambda Y_n + \omega_2\lambda\left(Y_n + \frac{h}{2\omega_2}\lambda Y_n\right)\right] \tag{44}$$

so that

$$\frac{dY_{n+1}}{dY_n} = 1 + h\lambda + \frac{(h\lambda)^2}{2}, \tag{45}$$

and the stability function is

$$S = \left.\frac{dY_{n+1}}{dY_n}\right|_{Y_n = Y_{\text{fixed point}}} \tag{46}$$

$$= 1 + h\lambda + \frac{(h\lambda)^2}{2}. \tag{47}$$

In general, recall that for an explicit method of order p, we wished to have Eq.(8) matching Eq.(4) up to terms of $O(h^p)$. Substituting $y' = f_n = \lambda y$ and $[(d^{p-1}f)/(dx^{p-1})]_n = \lambda^p y_n$ into Eq.(5):

$$y_{n+1} = y_n + h\lambda y_n + \frac{1}{2}h^2\lambda^2 y_n + \cdots + \frac{1}{p!}h^p\lambda^p y_n + O(h^{p+1}). \tag{48}$$

Thus for an explicit p-stage method of order p (which is only possible for $p \leqslant 4$), the stability function is

$$S = 1 + h\lambda + \frac{(h\lambda)^2}{2!} + \cdots + \frac{(h\lambda)^p}{p!}. \tag{49}$$

For an explicit q-stage method of order $p < q$,

$$S = 1 + h\lambda + \frac{(h\lambda)^2}{2!} + \cdots + \frac{(h\lambda)^p}{p!} + \sum_{i=p+1}^{q} \gamma_i (h\lambda)^i, \tag{50}$$

where the γ_is are determined by the particular Runge–Kutta method. In this case the free parameters may be used to maximize the area of the absolute-stability region. In practice, however, one does not get large increases in the area by doing this. It is interesting that in the optimal case $p = q$, the free parameters do not come into the expression for S, so all optimal methods of a given order will have the same absolute-stability region. We show the stability boundaries given by $|S| = 1$ for $1 \leqslant p = q \leqslant 4$ in Fig. 1. Notice that the size of the absolute-stability region increases with the order of the method. Note also that h and λ will always be found as the pair $h\lambda$ in this model problem, due to the method of construction of S. It is clear that for a q-stage method, S is a polynomial of degree q in h. Since $|S| \to \infty$ as $|h\lambda| \to \infty$, explicit methods all have bounded absolute-stability regions. Implicit Runge–Kutta methods have absolute-stability regions which can be unbounded. This is a great advantage in some situations, as we shall see below.

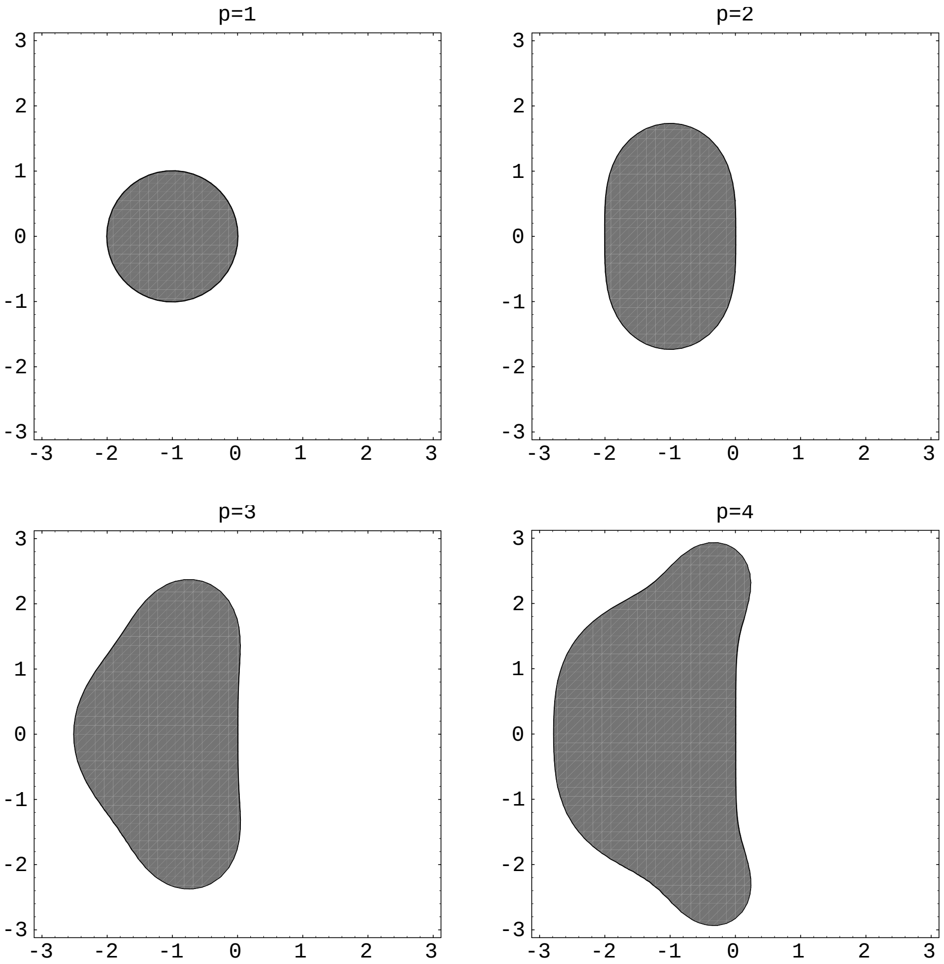

Figure 1: The absolute stability regions of explicit p-stage, pth-order Runge–Kutta methods for $1 \leqslant p \leqslant 4$ are plotted in complex $h\lambda$-space. The absolute stability regions are shown in grey. The ordinate and abscissa are $\mathrm{Im}(h\lambda)$ and $\mathrm{Re}(h\lambda)$ respectively. Notice that the size of the regions increases with the order of the method.

5. Nonlinear Absolute Stability

USING a linear model problem, the Runge–Kutta map is also linear. This means that the Runge–Kutta method is bound to have only one fixed point, as has the model problem. The basin of attraction is bound to be infinite if the fixed point is attractive, and merely the point itself otherwise, as in the model problem. This need not be the case with a nonlinear model problem; a Runge–Kutta method which has a certain absolute-stability region with the linear model problem could have quite a different region of stability with a nonlinear problem. The conventional absolute-stability analysis can be extended to nonlinear model problems as long as they have a stable fixed point. In this case a *nonlinear* absolute-stability test can be carried out in the same way as the linear absolute-stability test, by finding values of $h\lambda$ in complex space at which the fixed point loses stability. For example, let us look at the simplest nonlinear equation, the logistic equation

$$\begin{aligned} y' &= \lambda y(1-y), \\ y(a) &= \alpha, \qquad a \leqslant x. \end{aligned} \tag{51}$$

This has the analytical solution

$$y(x) = \frac{\alpha}{\alpha + (1-\alpha)e^{-\lambda(x-a)}}. \tag{52}$$

The problem has a stable fixed point at $y = 0$ for $\mathrm{Re}(\lambda) < 0$, and a stable fixed point at $y = 1$ for $\mathrm{Re}(\lambda) > 0$. It turns out that for this nonlinear model problem, the nonlinear absolute-stability function is the same as Eq.(49) for the fixed point at $y = 0$, so that the nonlinear absolute-stability regions for Runge–Kutta methods of orders one to four for this fixed point are the same as those shown in Fig. 1. They are merely the reflections of these regions in the imaginary axis of the $h\lambda$-plane for the other fixed point at $y = 1$.

The breakdown of stability in the two cases though is very different. In the linear model problem one merely has a change of stability of a fixed point. After the fixed point has become unstable, all orbits diverge to infinity, and so integration of the system on a computer rapidly leads to overflow. In the case of the nonlinear model problem from the logistic equation, things are far more interesting. Let us for a moment look at the Euler method for this problem. One obtains the map

$$Y_{n+1} = Y_n + h\lambda Y_n(1 - Y_n). \tag{53}$$

This can be put in the form

$$z_{n+1} = z_n^2 + c \tag{54}$$

by a linear coordinate transformation $z_n = -h\lambda Y_n + (1 + h\lambda)/2$ with $c = (1 - (h\lambda)^2)/4$. (Any complex quadratic map can be transformed into Eq.(54)

with a similar linear coordinate transformation.) Now Eq.(54) is immediately recognizable as giving the Mandelbrot set when iterated with z and c complex and the initial z value, z_0, set to the critical point of the map, $z_0 = 0$. The Mandelbrot set is then the set of complex c values for which the z orbit remains bounded. From Julia and Fatou, it is known that the basin of attraction of any finite attractor will contain the critical point (see, for example, Devaney [1989]), so the Mandelbrot set catalogues the parameter values for which a finite attractor exists. Other initial conditions may not fall in the basin of attraction of a finite attractor even if one exists; thus the Mandelbrot set is the maximum region in parameter space for which orbits can remain bounded. That is to say that using other initial conditions will lead to subsets of the Mandelbrot set.

The Mandelbrot set for Eq.(53) is shown in Fig. 2. The set itself is shown in red and the different coloured regions around it indicate the speed of escape to infinity. One can see the two nonlinear absolute-stability regions mentioned earlier; the circle of radius 1 and centre -1 which contains all parameter values for which the fixed point at 0 is stable, and the circle of radius 1 and centre 1 containing the parameter values for which the fixed point at 1 is stable. These circles map to the cardioid of the well-known Mandelbrot set of Eq.(54) under the coordinate transformation given above. The successively smaller circles further along the real axis in both directions are of periods 2, 4, 8, ... ; this is the well-known period-doubling cascade of the logistic map. Off the real axis the largest buds on the main circles are of period 3. Periods 4 and 5 are the next most prominent. We can see that the breakdown of nonlinear absolute stability on moving from inside to outside the main circles will not necessarily immediately lead to divergence and overflow in the computer. The result will depend on the point at which $h\lambda$ crosses the boundary in complex space, but it might well enter one of the buds surrounding the main circles for which the attractor has a higher period. The attracting set for initial conditions other than the critical point (which is $Y_0 = (1 + h\lambda)/(2h\lambda)$ in these coordinates) is a subset of the Mandelbrot set since even if $h\lambda$ lies inside the boundary of the Mandelbrot set, there is no guarantee that the orbit will converge to an attractor other than infinity, because the basin boundaries of the attractors are finite. The basin boundary at the point $h\lambda$ is known as the Julia set of $h\lambda$.

Here we start to see the big difference between this model problem and the previous linear one. We have finite basins of attraction in this nonlinear problem, so to arrive at the required fixed point solution, not only must one use a sufficiently small step length, but one must also be within the Julia set at that value of $h\lambda$. That is not all; it is possible for the Runge–Kutta map of an autonomous problem to have a set of fixed points that is larger than the set of fixed points of the differential equation [Iserles, 1990; Yee *et al.*, 1991]. This is obvious for an explicit method if in $y' = f(y)$, $f(y)$ is a polynomial, since the Runge–Kutta map will be a higher-degree polynomial than $f(y)$ due to the construction of the Runge–Kutta method, and so must have more fixed points. In fact, the fixed-point set of the

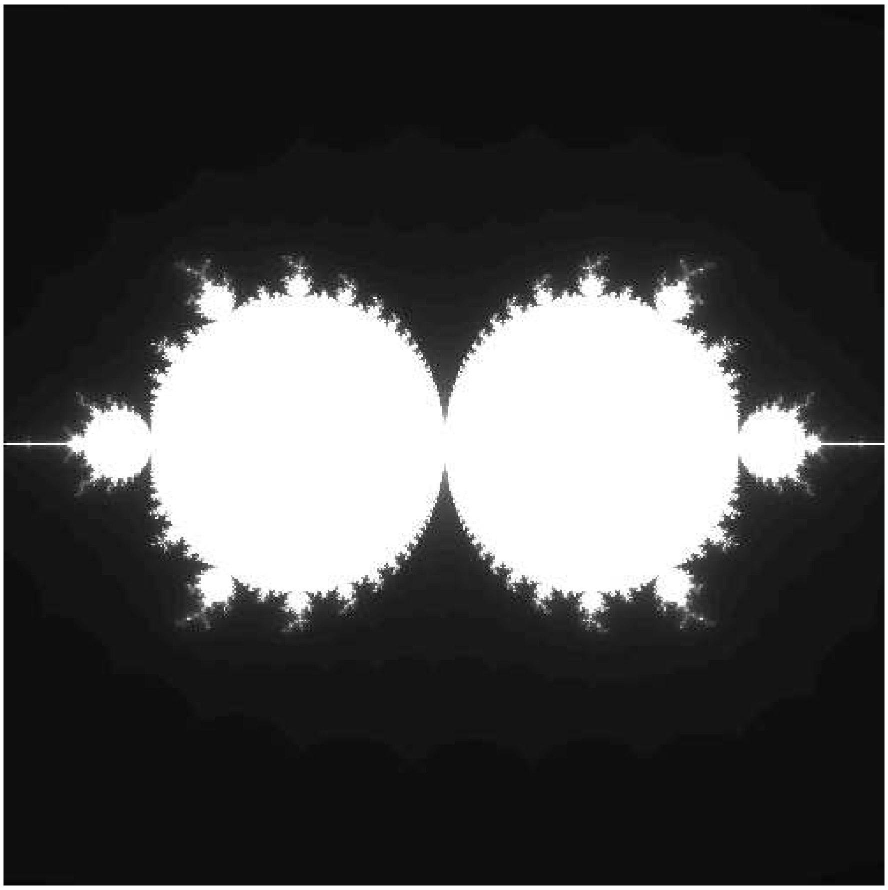

Figure 2: The Mandelbrot set for the map $Y_{n+1} = Y_n + h\lambda Y_n(1 - Y_n)$, which arises from applying the Euler method to the logistic equation, is shown in red. The different coloured regions surrounding it indicate the speed of escape to infinity at that point of complex $h\lambda$-space. The two large circles in this Mandelbrot set map to the prominent cardioid seen in the normal parameterization $z_{n+1} = z_n^2 + c$ of the Mandelbrot set.

Runge–Kutta map contains the fixed-point set of the differential equation as a subset. If Δ is a fixed point of $y' = f(y)$ then $f(\Delta) = 0$. The Runge–Kutta map

$$Y_{n+1} = Y_n + h \sum_{i=1}^{q} \omega_i k_i \tag{55}$$

has fixed points given by

$$\phi = \sum_{i=1}^{q} \omega_i k_i = 0, \tag{56}$$

where

$$k_i = f\left(Y_n + h \sum_{j=1}^{i-1} \beta_{ij} k_j\right) \tag{57}$$

for an explicit method. Now if $Y_n = \Delta$, $k_1 = f(\Delta) = 0$ and $k_i = f(\Delta) = 0$ for all i. For an implicit method

$$k_i = f\left(Y_n + h \sum_{j=1}^{q} \beta_{ij} k_j\right), \tag{58}$$

so if $Y_n = \Delta$, $k_i = 0$ for all i is again a solution. Thus $\phi = 0$ and so the fixed points of the differential equation are also fixed points of the map, but they are not necessarily the only fixed points of the map.

The Euler method is an exception: since the Euler map is $Y_{n+1} = Y_n + hf(Y_n)$, the fixed points will be given by $f(Y_n) = 0$, like those of the differential equation. Iserles [1990] terms methods like the Euler method, for which the fixed-point sets of the differential equation and the map are the same, *regular* methods. He gives an example of an implicit two-stage method that is regular. The commonly-used Runge–Kutta schemes are not implicit and are not regular, so additional *ghost* fixed points occur in the map that are not present in the differential equation. For instance, consider the integration of the logistic equation with a second-order explicit method. In this case, the fixed points of the differential equation are given by a quadratic

$$\lambda Y_n(1 - Y_n) = 0, \tag{59}$$

and those of the Runge–Kutta method are given by a quartic

$$\begin{aligned} \frac{1}{\omega_2} Y_n(1 - Y_n) &\left(2h\lambda Y_n - h\lambda - 4\omega_2 - \sqrt{h^2\lambda^2 + 16\omega_2(\omega_2 - 1)}\right) \\ &\times \left(2h\lambda Y_n - h\lambda - 4\omega_2 + \sqrt{h^2\lambda^2 + 16\omega_2(\omega_2 - 1)}\right) = 0. \end{aligned} \tag{60}$$

Thus we get two extra fixed points, and it turns out that one of these ghost fixed points remains stable as the step length tends to zero. In Fig. 3 we show the ghost fixed points plotted against $h\lambda$ and ω_2 with all variables real. Notice that the ghost roots are dependent on the step length and the Runge–Kutta parameter whereas the real roots are not.

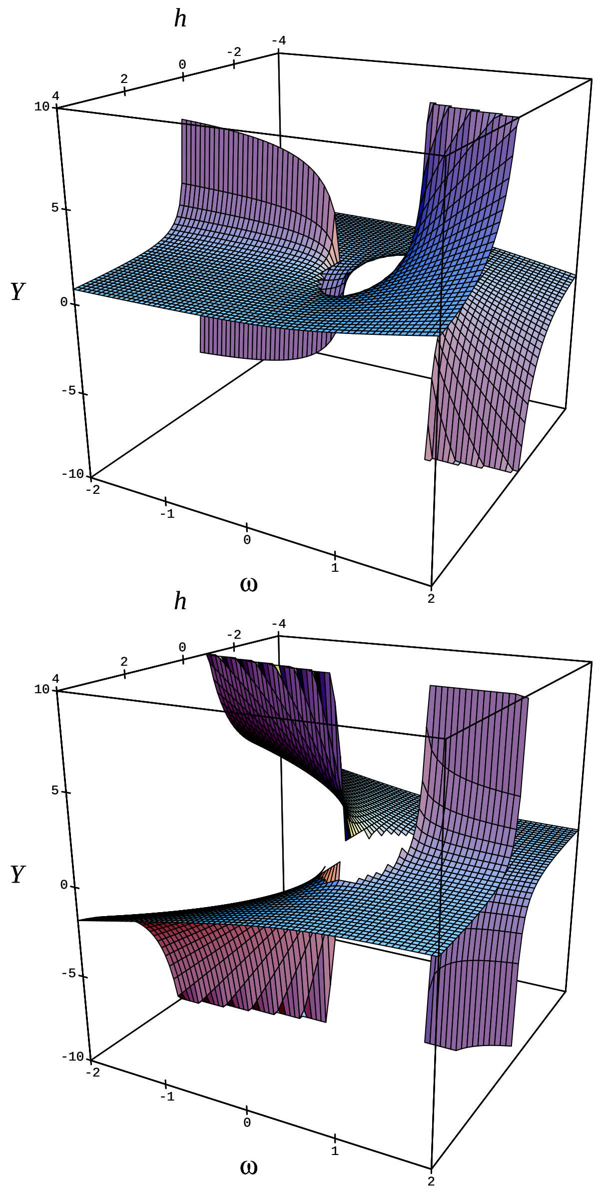

Figure 3: (a) & (b). The two ghost fixed points of two-stage explicit Runge–Kutta maps of the logistic equation are shown as functions of $h\lambda$ and the Runge-Kutta parameter ω_2. Notice that they tend to infinity as $h \to 0$, and that they are in general dependent on $h\lambda$ and ω_2, whereas the real fixed points of the logistic equation, 0 and 1, are independent of these parameters.

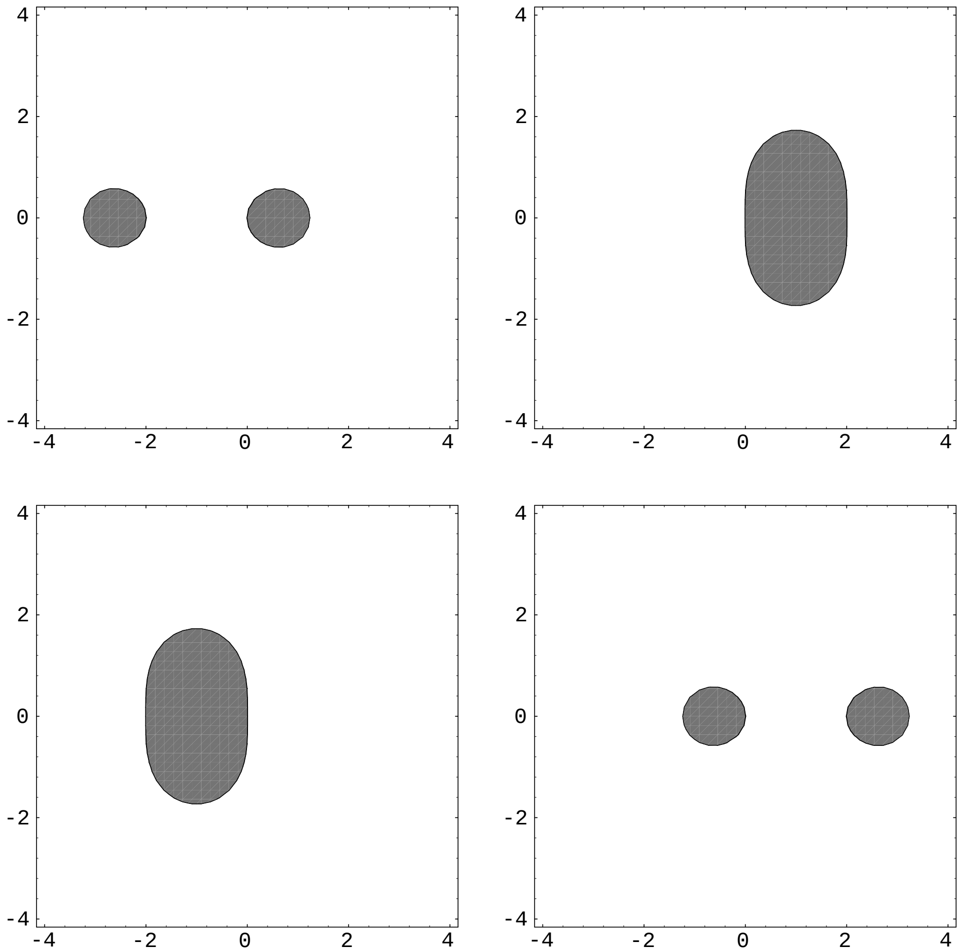

Figure 4: Nonlinear absolute-stability regions of the four fixed points of the two-stage, second-order explicit Runge–Kutta method with $\omega_2 = 1$ applied to the logistic equation are plotted in complex $h\lambda$-space. The absolute stability regions are shown in grey. The ordinate and abscissa are $\text{Im}(h\lambda)$ and $\text{Re}(h\lambda)$ respectively. The two absolute-stability regions from the real fixed points of the logistic equation, the large single regions, are independent of ω_2, but those of the ghost fixed points, the two disconnected circles, are not, so we have chosen $\omega_2 = 1$ here. The absolute-stability regions of the real and ghost fixed points overlap in some areas. In these cases, which fixed point is found depends on the initial conditions.

Since consistency tells us that $\phi(Y_n; 0) = f(Y_n)$, we know that there must, for an irregular method, be fewer fixed points when the step length is zero than when it is nonzero. One can ask what happens to the ghost fixed points as the step length tends to zero. Figure 3 shows that in this case the ghost fixed points tend to infinity as the step length decreases. In Fig. 4 we show the nonlinear absolute-stability regions of all four fixed points for the Runge–Kutta method with $\omega_2 = 1$. (It is interesting that whereas the two absolute-stability regions of the real fixed points are independent of ω_2, the two regions of the ghost fixed points are not; we choose $\omega_2 = 1$ as our example.) The union of these four regions is the part of the Mandelbrot set for this map in which iterates tend to a fixed point. In addition to these regions, the Mandelbrot set will have further regions where periodic orbits of period greater than one are stable, similar to the buds off the most prominent circles in the Mandelbrot set shown in Fig. 2.

In Fig. 5, we show a bifurcation diagram for a fourth-order Runge–Kutta scheme integrating the logistic equation. What we are doing here is just looking along the real axis of the Mandelbrot set for this case; we are keeping $h\lambda$ real. The complete Mandelbrot set would be much more difficult to compute than the quadratic case of Fig. 2, since one would have to follow fifteen critical points. One can however say that it would have the fourth-order absolute-stability region of Fig. 1 and its mirror image in the imaginary axis as subsets, in a similar fashion to the circles of Fig. 2. The map is a sextodecic (sixteenth degree) polynomial in Y_n, but one can see that period doubling leading to chaos and eventually escape to infinity occurs along the real axis in a similar way to the cascade in the logistic map. Since the fourth-order absolute-stability regions are larger than the first-order ones, the behaviour remains stable up to larger step lengths here than in the logistic-map case.

For another example of the appearance of ghost fixed points, we integrate the equation $y' = \cos y$, which has real fixed points at $y = (2m+1)\pi/2$ where m is an integer, with a second-order explicit Runge–Kutta method. In addition to the real fixed points, we also get ghost fixed points which are the roots of

$$1 - \omega_2 + \omega_2 \frac{\cos\left(Y_n + \dfrac{h}{2\omega_2}\cos Y_n\right)}{\cos Y_n} = 0. \tag{61}$$

We plot a pair of these ghost fixed points against h and ω_2 in Fig. 6. The ghost fixed points, which are stable, come together and coalesce at a nonzero h. At smaller values of h, the ghost fixed points are imaginary even when the other variables are real. The pattern shown in Fig. 6 is repeated periodically in Y_n and for all values of ω_2. (There are no ghost fixed points for $\omega_2 = 0$, since in this case we have the first-order Euler method.)

Although ghost fixed points have been known about for some time [Yamaguti & Ushiki, 1981; Ushiki, 1982; Prüfer, 1985], it is only recently that it has been appreciated that, in some cases, they exist for all step lengths, i.e., at step lengths below the linear absolute-stability boundary [Yee *et al.*, 1991]. Thus we can see that irregularity in the numerical method can

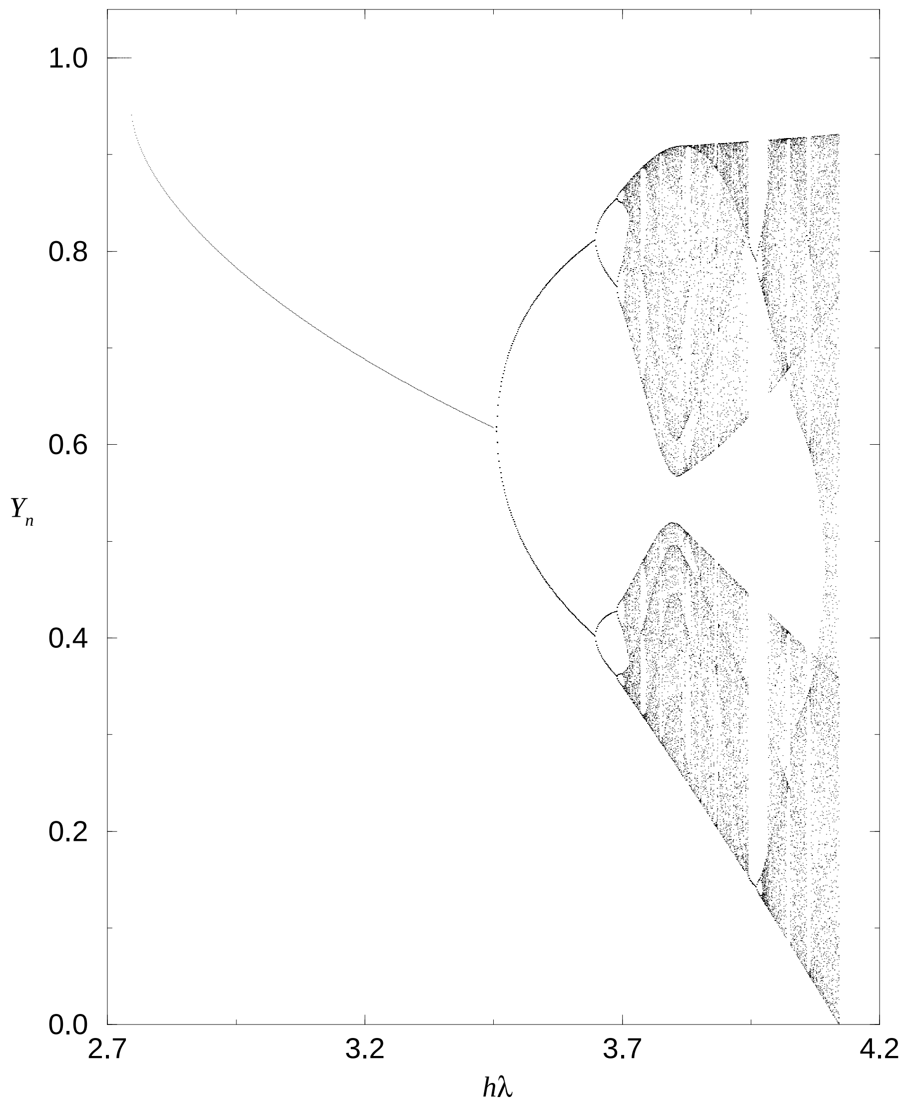

Figure 5: Bifurcation diagram for the well-known fourth-order Runge–Kutta method (often called *the* Runge–Kutta Method), applied to the model problem $y' = \lambda y(1-y)$, the logistic equation, with $y(0) = 1/2$. Notice that the first bifurcation occurs at $h\lambda = 2.78$. Comparing this with the picture for $p = 4$ in Fig. 1, and remembering that the nonlinear absolute-stability regions for the logistic equation are the same as the linear regions of Fig. 1, we conclude that this first bifurcation, a transcritical bifurcation where the real fixed point at 1 and a ghost fixed point meet and exchange stability, occurs as $h\lambda$ crosses the absolute-stability boundary. At higher values of $h\lambda$, we see the panoply of period-doubling bifurcations leading to chaos and finally escape to infinity.

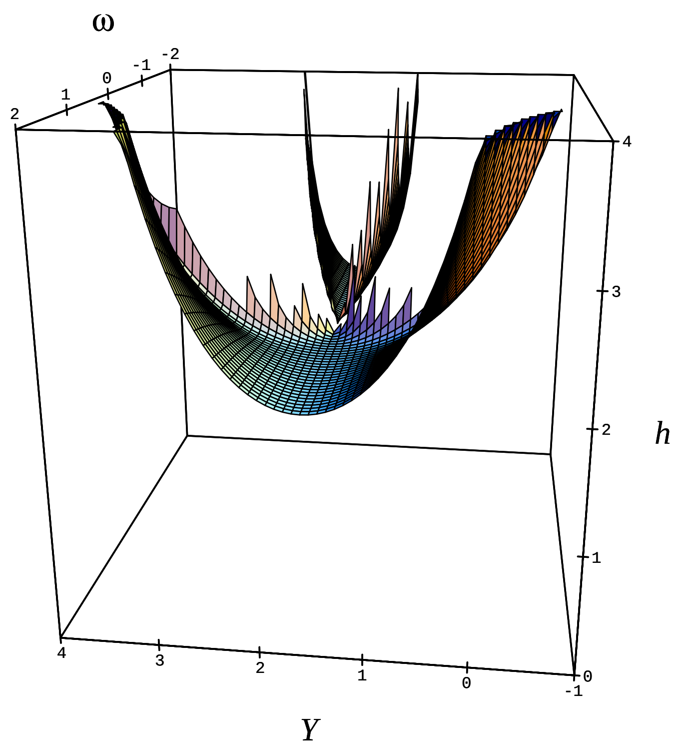

Figure 6: The ghost fixed points of two-stage explicit Runge–Kutta maps of $y' = \cos y$ are shown here as a function of the step length h, and the Runge–Kutta parameter ω_2. They coalesce and disappear at a nonzero value of h. Below this limit the ghost fixed points are imaginary. This picture is repeated periodically in Y_n. The ghost fixed points disappear for $\omega_2 = 0$, when we have the regular first-order Euler method.

be a serious problem. Convergence, because it is a limit concept, ties the dynamics of the map from the numerical method only loosely to that of the differential equation, leaving room for major differences to occur. These differences manifest themselves in ghost fixed points. As we have seen, the ghost fixed points must disappear when the step length is zero, but they may be present for all nonzero step lengths. They can be stable for arbitrarily small step lengths, in which case a trajectory may converge to a fixed point which does not exist in the original system. Even if they are unstable, they still greatly affect the dynamics of the discrete system compared to the continuous original. The difference between linear and nonlinear absolute-stability regions is that basin boundaries are infinite in the linear case, but finite in the nonlinear case. Thus convergence to the fixed point is guaranteed if $h\lambda$ is within the linear absolute-stability region, whereas this is not true in the nonlinear case since, in addition, Y_0 must be inside the Julia set.

6. Stiff Problems

OFTEN, accuracy requirements that set a bound on the local truncation error keep the step length well within the region of stability. When this is not the case, and maximum step length is dictated by the boundary of the stability region, the problem is said to be *stiff*.

Traditionally, a linear stiff system of size n was defined by

$$\mathrm{Re}(\lambda_i) < 0, \qquad 1 \leqslant i \leqslant n, \tag{62}$$

with

$$\max_{1\leqslant i\leqslant n} |\mathrm{Re}(\lambda_i)| \gg \min_{1\leqslant i\leqslant n} |\mathrm{Re}(\lambda_i)|. \tag{63}$$

The *stiffness ratio* R provided a measure of stiffness:

$$R = \frac{\max\limits_{1\leqslant i\leqslant n} |\mathrm{Re}(\lambda_i)|}{\min\limits_{1\leqslant i\leqslant n} |\mathrm{Re}(\lambda_i)|}. \tag{64}$$

λ_i are the eigenvalues of the Jacobian of the system. By this definition, a stiff problem has a stable fixed point with eigenvalues of greatly different magnitudes. Remember that large negative eigenvalues correspond to fast-decaying transients $e^{\lambda x}$ in the solution. (Large positive eigenvalues may also lie outside the region of absolute stability, but traditionally we are not so interested in them, because the solution is exponentially growing here anyway.) This definition of stiffness is not valid for nonlinear systems. It is based on the linear model problem in Eq.(43), and the eigenvalues λ_i above pertain to a linear system. One should note that the stiffness ratio is often not a good measure of stiffness even in linear systems, since if the minimum eigenvalue is zero, the problem has infinite stiffness ratio, but may not be stiff at all if the other eigenvalues are of moderate size.

We shall adopt a verbal definition of stiffness, valid for both linear and nonlinear problems, that is similar to Lambert's [Lambert, 1991]:

> If a numerical method is forced to use, in a certain interval of integration, a step length which is excessively small in relation to the smoothness[3] of the exact solution in that interval, then the problem is said to be *stiff* in that interval.

Unlike the linear definition of stiffness, our definition allows a single equation, not just a system of equations, to be stiff. It also allows a problem to be stiff 'in parts': a nonlinear problem may start off nonstiff and become stiff, or vice versa. It may even have alternating stiff and nonstiff intervals.

As an example of a linear stiff problem, consider the equation

$$y'' - 1001y' + 1000y = 0. \tag{65}$$

[3] n.b. The word 'smoothness' is used here in its intuitive, nontechnical sense.

We can write this as $y' = Ay$ where

$$A = \begin{pmatrix} 0 & 1 \\ -1000 & -1001 \end{pmatrix}, \tag{66}$$

so the eigenvalues are $\lambda_1 = -1$ and $\lambda_2 = -1000$. The equation has solution

$$y = Ae^{-x} + Be^{-1000x}, \tag{67}$$

so we would expect to be able to use a large step length after the e^{-1000x} transient term had become insignificant in size, but in fact the presence of the large negative eigenvalue λ_2 prevents this, since $h\lambda_2$ would then lie outside the absolute-stability region. With appropriate initial conditions, one could even remove the e^{-1000x} term from the solution entirely, but this would not change the fact that step length is dictated here by the size of $h\lambda_2$; one would still have to use a very small step length throughout the calculation.

Now let us look at a nonlinear stiff problem. Take the equation

$$y' = \lambda y + g'(x) - \lambda g(x). \tag{68}$$

This has solution

$$y = Ae^{\lambda x} + g(x). \tag{69}$$

Here the size of $h\lambda$ controls stability, so if $g(x)$ is a fairly smooth function and λ is large, we have stiffness. Again we can even remove the transient $Ae^{\lambda x}$ term by setting $y_0 = g(x_0)$ so that $A = 0$, but if λ is large we still have a stiff problem, even though λ does not appear in the solution. For example (from Lambert [1991]), choosing $g(x) = 10 - (10 + x)e^{-x}$ and $y_0 = 0$, we have a problem whose solution is $y = 10 - (10 + x)e^{-x}$, i.e., not involving λ. However, $h\lambda$ is still controlling the stability of the numerical integration. With this nonlinear problem, we have started to move beyond absolute-stability theory. If we had chosen $g(x)$ so that the problem did not have a fixed-point solution, but some other fairly smooth motion (for example $g(x) = \sin x$), we would still have a stiff problem, but would no longer be able to analyze it using absolute stability. This is the case with the next example.

In Fig. 7, we show the results of integrating the van der Pol equation

$$y'' - \lambda(1 - y^2)y' + y = 0 \tag{70}$$

using a variable-step fourth-order Runge–Kutta code, for the cases $\lambda = 1$ and $\lambda = 100$. We have requested the same bound on the principal local truncation error estimate in both cases. We can see from the far greater number of steps needed in the latter case, that at large λ the van der Pol equation becomes very stiff. At these large λ values, the equation describes a relaxation oscillator. These have fast and slow states in their cycle which characterizes the 'jerky' motion displayed in Fig. 7. Chaotic systems can also be stiff, as we can see if we introduce forcing into the van der Pol equation:

$$y'' - \lambda(1 - y^2)y' + y = A\cos\omega x. \tag{71}$$

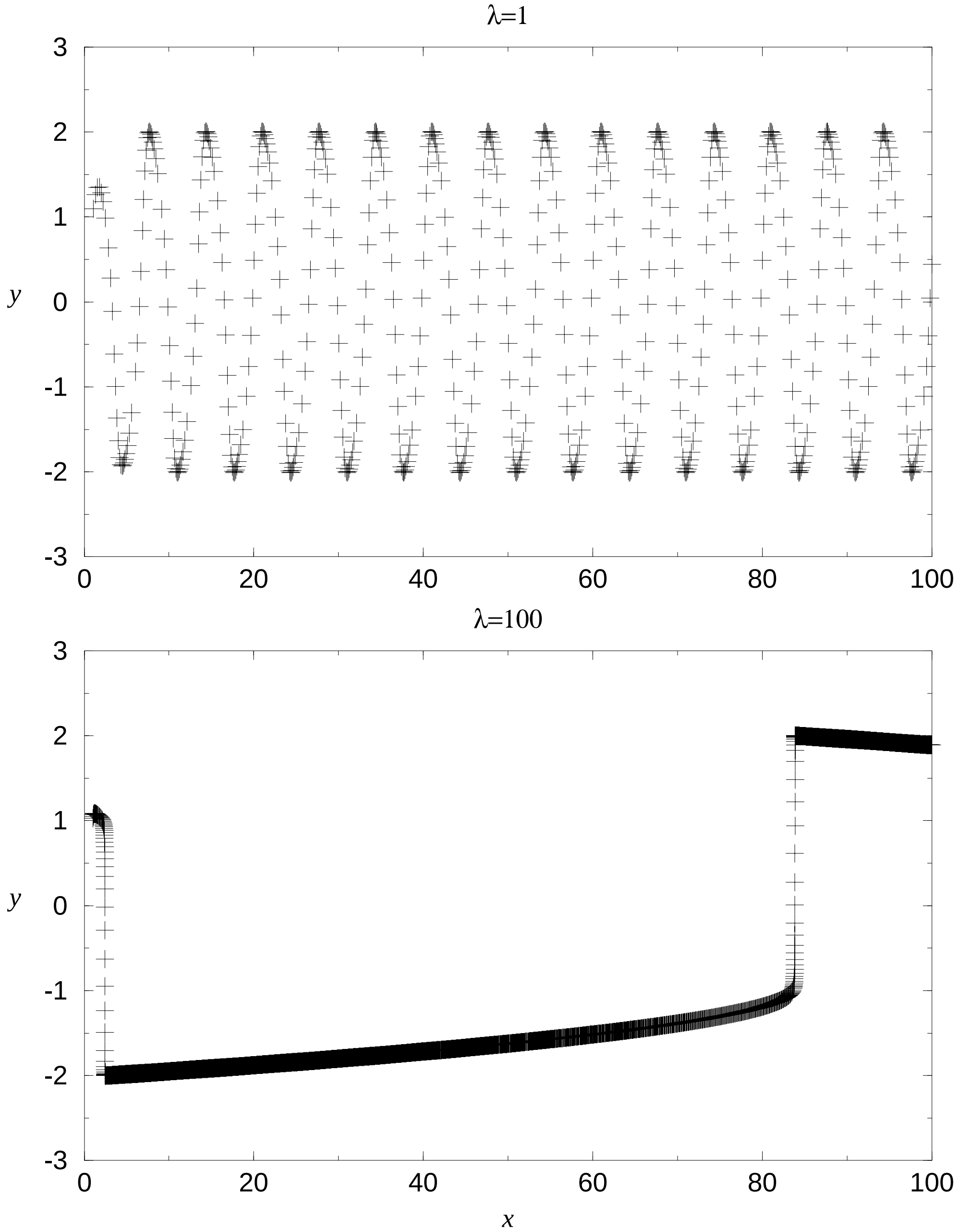

Figure 7: Results of numerical integration of the van der Pol equation $y'' - \lambda(1 - y^2)y' + y = 0$ with $\lambda = 1$ and $\lambda = 100$ using a variable-step fourth-order Runge-Kutta method. Each step is represented by a cross. The far greater number of steps taken in the latter case, despite the greater smoothness of the computed solution, shows the presence of stiffness. The steps are so small at $\lambda = 100$ that the individual crosses merge to form a continuous broad line on the graph.

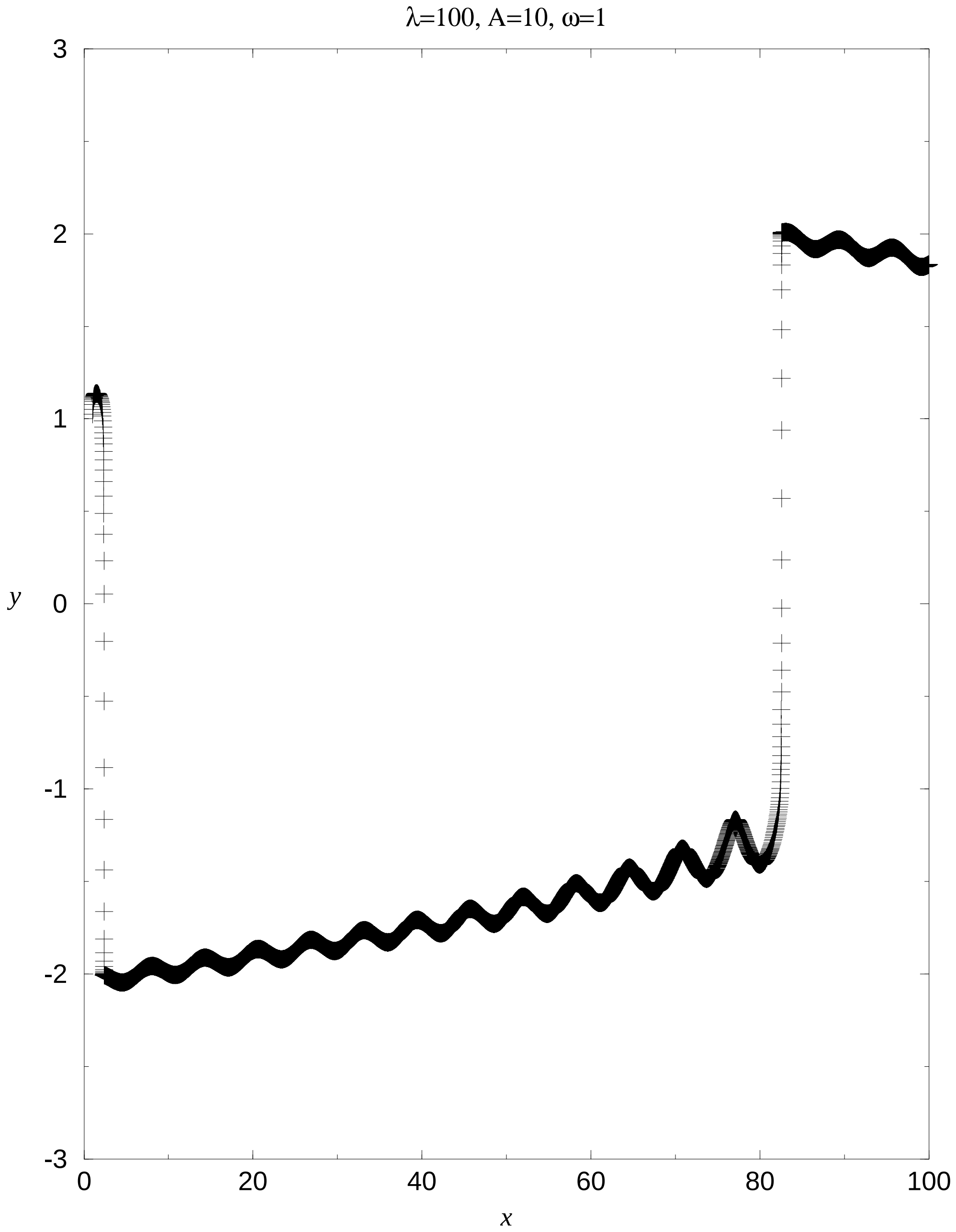

Figure 8: Results of numerical integration of the forced van der Pol equation $y'' - \lambda(1 - y^2)y' + y = A\cos\omega x$ with $\lambda = 100$, $A = 10$, and $\omega = 1$ using a variable-step fourth-order Runge-Kutta method, with each step represented by a cross. It is obviously stiff, since as in the $\lambda = 100$ case in Fig. 7, the steps are so small that they have merged together in the picture. The chaotic nature of the forced van der Pol equation is not apparent at this timescale; the manifestation of chaos in this system lies in the random selection of one of two possible periods for each relaxation oscillation, so this picture, which shows only part of one cycle, cannot display chaos.

This forced van der Pol equation exhibits chaotic behaviour (see, for example, Tomita [1986], Thompson & Stewart [1986], or Jackson [1989]) and is also stiff, as can be seen in Fig. 8. The presence of fast and slow time scales in a problem is a characteristic of stiffness. Stiff problems are not mere curiosities, but are common in dynamics and elsewhere [Aiken, 1985].

When integrating a stiff problem with a variable-step Runge–Kutta code, the initial step length chosen, which often causes the method to be at or near numerical instability, generally leads to a large local truncation error estimate. This then causes the routine to reduce the step length, often substantially, until the principal local truncation error is brought back within its prescribed bound. The routine then integrates the problem successfully, but uses a far greater number of steps than seems reasonable, given the smoothness of the solution. Because of this, round-off error and computation time are a problem when using conventional techniques to integrate stiff problems.

It would seem to be especially desirable then, for methods of integration for stiff problems, that the method be stable for all step lengths for the parameter values where the original system is stable. For example, the linear model problem Eq.(41) is stable for $\mathrm{Re}(\lambda) < 0$, so the numerical method should be stable for all h for $\mathrm{Re}(\lambda) < 0$ i.e., the absolute-stability region should be the left half-plane. The concept of *A-stability* was introduced for this reason. A method is A-stable if its linear absolute-stability region contains the whole of the left half-plane. This being the case, a numerical method integrating the linear model problem will converge to the fixed point for all values of λ that the model problem itself does, and for all values of h. A-stability is a very severe requirement for a numerical method: we already know that explicit Runge–Kutta methods cannot fulfill this requirement since their absolute-stability regions are finite. It is known, however, that some implicit Runge–Kutta methods are A-stable [Butcher, 1987; Lambert, 1991]. The drawback with implicit methods is that at each step a system of nonlinear equations must be solved. This is usually achieved using a Newton–Raphson algorithm, but at the expense of many more function evaluations than are necessary in the explicit case. Consequently, implicit Runge–Kutta methods are uneconomical compared to rival methods for integrating stiff problems. Usually, stiff problems are instead solved using Backward Differentiation Formulae (also known as *Gear*) methods. A-stability is not quite what we required above however, since it is based on linear absolute stability, and also because it allows regions in addition to the left half-plane to be in the absolute-stability region, so that the numerical method may give a convergent solution when the exact solution is diverging. Better then is what has been called *precise* A-stability, which holds that the absolute-stability region should be just the left half-plane. Precise A-stability though is still based on linear absolute-stability theory.

7. A Nonlinear Stability Theory

IN the past few years numerical analysts have come to realize that linear stability theory cannot be applied to nonlinear systems. One cannot say that the Jacobian represents the local behaviour of the solutions except at a fixed point. This had not previously been appreciated in numerical analysis, and there was a tendency to believe that looking at the Jacobian at one point as a constant, the solutions nearby would behave like the linearized system produced from this 'frozen' Jacobian. Numerical analysts have now recognized the failings of linear stability theory when applied to nonlinear systems, and have constructed a new theory of nonlinear stability.

The theory looks at systems that have a property termed *contractivity*; if $y(x)$ and $\tilde{y}(x)$ are any two solutions of the system $y' = f(x, y)$ satisfying different initial conditions, then if

$$\|y(x_2) - \tilde{y}(x_2)\| \leqslant \|y(x_1) - \tilde{y}(x_1)\| \tag{72}$$

for all x_1, x_2 where $a \leqslant x_1 \leqslant x_2 \leqslant b$, the system is said to be contractive. An analogous definition may be framed for a discrete system; if

$$\|Y_{n+1} - \tilde{Y}_{n+1}\| \leqslant \|Y_n - \tilde{Y}_n\|, \tag{73}$$

the discrete system is said to be contractive. Now we must define another property of the system which numerical analysts have termed *dissipativity*. Since we have already used this word in dynamics, we shall call this new property *NA-dissipativity* (numerico-analytic dissipativity). The system $y' = f(x, y)$ is said to be NA-dissipative in $[a, b]$ if

$$\langle f(x, y) - f(x, \tilde{y}), y - \tilde{y} \rangle \leqslant 0, \tag{74}$$

where $\langle \cdot, \cdot \rangle$ is an inner product, holds for all y, $\tilde{y}$ in M_x and for $a \leqslant x \leqslant b$, where M_x is the domain of $f(x, y)$ regarded as a function of y. NA-dissipativity can be shown to imply contractivity. We needed to define NA-dissipativity because we can obtain a usable test for it; a system is NA-dissipative if

$$\mu\,[J] \leqslant 0, \tag{75}$$

where $\mu\,[J]$ is the *logarithmic norm* of the Jacobian $J = \partial f / \partial y$. If σ_i, $i = 1 \ldots m$ are the eigenvalues of $\frac{1}{2}\left(J + J^T\right)$, Eq.(75) can be shown to be equivalent to

$$\max_i \sigma_i \leqslant 0. \tag{76}$$

We now have a practical sufficient condition for contractivity.

We further define: if a Runge–Kutta method applied with any step length to a NA-dissipative autonomous system is contractive, then the method is said to be *B-stable*. If we have the same situation with a nonautonomous system, then the method is said to be *BN-stable*. A sufficient condition for both of these properties is given by *algebraic stability*. A Runge–Kutta method is said to be algebraically stable if $\Omega = \mathrm{diag}(\omega_1, \omega_2 \ldots \omega_q)$ and

$M = \Omega\mathrm{B} + \mathrm{B}^T\Omega - \omega\omega^T$ are both nonnegative definite. B and ω here come from Eq.(11), the Butcher array of the Runge–Kutta method. Algebraic stability implies A-stability, but the reverse is not true.

Let us look at this theory from a dynamics viewpoint. The problem is that contractivity is a very severe requirement to impose; in fact it precludes the possibility of chaos occurring in the system. Is is easy to see that this must be so, since chaos demands that neighbouring trajectories be divergent, whereas contractivity demands that they be convergent. We show in Appendix A.2 that contractivity is sufficient to give nonpositive Lyapunov exponents; positive Lyapunov exponents are a necessary condition for chaos to occur. Unfortunately then, one can only investigate the stability of contractive, nonchaotic systems with the nonlinear stability theory as it stands. However, as we have remarked previously, practical numerical codes do not use any stability theory to evaluate their accuracy, so this is a theoretical, rather than a practical, problem. It should be pointed out that what numerical analysts call dissipativity, what we have termed NA-dissipativity, is a much stronger requirement even than strict dissipativity ($\nabla.f < 0$) in dynamics. The latter merely requires that $\sum_i \lambda_i < 0$, whereas the former insists that $\max_i \lambda_i \leqslant 0$ (λ_i being the Lyapunov exponents of the system).

8. Towards a Comprehensive Stability Theory

WE are still seeking a comprehensive nonlinear stability theory. The theory of the previous section deals only with the special case of contractive systems. Nonlinear absolute stability shows that regularity in the numerical method is obviously a good thing, but it is only concerned with fixed-point behaviour. Other studies have been made on the link between the fixed points in the differential equation and those in the map produced by the numerical analysis. Stetter [1973] has shown that hyperbolic stable fixed points in the continuous system remain as hyperbolic stable fixed points in the discrete system for sufficiently small step lengths. Beyn [1987b] shows that hyperbolic unstable fixed points are also correctly represented in the discrete system for sufficiently small step lengths, and that the local stable and unstable manifolds converge to those of the continuous system in the limit as the step length tends to zero. We need to look at other sorts of asymptotic behaviour apart from fixed points: invariant circles (limit cycles) and strange attractors.

Peitgen & Richter [1986] use two different Runge–Kutta methods, the Euler method and the two-stage, second-order Heun method, to discretize the Lotka–Volterra equations

$$\begin{aligned} {}^1y' &= {}^1y - {}^1y\,{}^2y, \\ {}^2y' &= -{}^2y + {}^1y\,{}^2y. \end{aligned} \tag{77}$$

This system has a centre-type (elliptic) fixed point at $(1, 1)$ surrounded by an infinity of invariant circles filling that quadrant of the plane. Instead of this continuum of invariant circles, Runge–Kutta maps of this system have an unstable fixed point with only one attracting invariant circle around it. (Fig. 51 in Peitgen & Richter [1986] is incorrect in showing the Euler method as having all orbits tending to infinity at any step length.) Peitgen and Richter describe more interesting dynamics occurring in the Heun method. For small step lengths, there is just the unstable fixed point and the attracting invariant circle. As the step length is increased, the invariant circle comes into resonance with various periodic orbits and is eventually transformed into a strange attractor. They also observe periodic orbits remote from the invariant circle and coexisting with it. Discretizing the nongeneric situation of a continuum of invariant circles that occurs in the Lotka–Volterra equations leads to a restoration of genericity; we arrive at the structurally stable configuration of just one invariant circle around the fixed point. The periodic orbits described on and remote from the invariant circle, and the transition to chaos that occurs, are consistent with investigations of one-parameter families of maps embedded in a two-parameter family which has a Hopf bifurcation [Aronson *et al.*, 1983; Arrowsmith *et al.*, 1992]. Gardini *et al.* [1987] study the Euler map of a three-dimensional version of the Lotka–Volterra system, in which they find Hopf bifurcations and a transition to strange attractors following a similar pattern to the two-dimensional case.

In the previous example, we have seen that a non-structurally-stable configuration of invariant circles in a system of ordinary differential equations collapses to a system with one invariant circle under the discretization imposed. Beyn [1987a] shows that for a continuous system with a hyperbolic invariant circle, the invariant circle is retained in the discretization if the step length is small enough. He demonstrates that the continuous and discrete invariant circles run out of phase. The relative phase shift per revolution depends on the step length, and the global error oscillates as the discrete and continuous systems move into and out of phase. Beyn gives examples of integrating systems that have invariant circles with the Euler method and with a fourth-order Runge–Kutta method. He shows that when the step length is increased, the invariant circles in the discrete system are transformed in the former case into a strange attractor, and in the latter, into a stable fixed point via a Hopf bifurcation.

The work reviewed in the previous paragraphs shows that under certain reasonable conditions, the numerical method can correctly reproduce different kinds of limit sets that are present in the differential equation. The conditions include asking that the behaviour in the continuous system be structurally stable. This condition is not satisfied for the Lotka–Volterra example above, which is why the numerical methods used do not correctly reproduce the behaviour of that system.

The problem with discretization, however, is that it introduces new limit-set behaviour in addition to that already existing in the continuous system. This is highlighted by the work of Kloeden & Lorenz [1986] who show that if a continuous system has a stable attracting set then, for sufficiently small step lengths, the discrete system has an attracting set which contains the continuous one. (An attracting set need not contain a dense orbit, which is what distinguishes it from an attractor.) In particular, this is shown by the demonstration that the fixed-point set of the continuous system is a subset of the fixed-point set of the discrete system, with extra ghost fixed points appearing in the discretization.

In general, a discretized dynamical system may possess fixed points, invariant circles, and strange attractors which are either fewer in number, or are entirely absent in the continuous system from which it arose. For instance, strange attractors are generic in systems of dimensionality three or more. Discretizing a system with a strange attractor will lead to a strange attractor. The question then to be asked is whether the properties of the discrete strange attractor are similar to those of the continuous one. Discretizing a system without a strange attractor may also lead to a strange attractor. If a structurally-stable feature is present in the continuous system, it will also be found in the discrete version, but the converse is not true. Note that non-structurally-stable behaviour will not in general persist under the perturbation of the system introduced by discretization.

Shadowing theory was first used to demonstrate that an orbit of a floating-point map produced by a computer using real arithmetic can be shadowed by a real orbit of the exact map [Hammel *et al.*, 1988]. More recently it has been shown that an orbit of a floating-point map can be shadowed by a real trajectory of an ordinary differential equation [Sauer & Yorke, 1991]. In the former case, we are investigating the effect of round-off error, in the latter case, global error. The last result would seem to contradict what has been stated before about the defects of numerical methods; in fact, it does not. The reason is that although shadowing theory is able to put a bound on the global error, it is only possible to do this if the numerical method satisfies

$$\|Y_{n+1} - g(Y_n)\| < \delta \qquad \forall n. \tag{78}$$

This is a form of local error where Y_{n+1} is a point on the numerical orbit and $g(Y_n)$ is the exact time-h map applied to the previous point on the numerical orbit. Compare this with our definition of local truncation error in Eq.(32), which instead takes the difference between a point on the orbit of the exact time-h map and the numerical method applied to the previous point on the exact orbit. It is not possible to get a good bound on δ with Runge–Kutta methods, but it is possible with the direct Taylor series method. (The pth order Taylor series method is just the Taylor series truncated at that order.) The disadvantage of the Taylor series method is that one has to do a lot of differentiation. However, if one can satisfy this bound, and another condition which is basically an assumption of hyperbolicity, then an orbit is $\sqrt{\delta}$-shadowed by an orbit of the exact time-h map:

$$\|y_n - Y_n\| < \sqrt{\delta} \qquad \forall n. \tag{79}$$

Comparing this with Eq.(31), we can see that it is giving us a bound on the global error in the method. Sauer & Yorke [1991], as an example, apply this shadowing method to prove that a chaotic numerical orbit of the forced, damped pendulum is shadowed by a chaotic real trajectory.

9. Symplectic Methods for Hamiltonian Systems

IT is now well known that numerical methods such as the ordinary Runge–Kutta methods are not ideal for integrating Hamiltonian systems, because Hamiltonian systems are not generic in the set of all dynamical systems, in the sense that they are not structurally stable against non-Hamiltonian perturbations. The numerical approximation to a Hamiltonian system obtained from an ordinary numerical method does introduce a non-Hamiltonian perturbation. This means that a Hamiltonian system integrated using an ordinary numerical method will become a dissipative (non-Hamiltonian) system, with completely different long-term behaviour, since dissipative systems have attractors and Hamiltonian systems do not.

This problem has led to the introduction of methods of symplectic integration for Hamiltonian systems, which do preserve the features of the Hamiltonian structure by arranging that each step of the integration be a *canonical* or *symplectic* transformation [Menyuk, 1984; Feng, 1986; Sanz-Serna & Vadillo, 1987; Itoh & Abe, 1988; Lasagni, 1988; Sanz-Serna, 1988; Channell & Scovel, 1990; Forest & Ruth, 1990; MacKay, 1990; Yoshida, 1990; Auerbach & Friedmann, 1991; Candy & Rozmus, 1991; Feng & Qin, 1991; Miller, 1991; Marsden *et al.*, 1991; Sanz-Serna & Abia, 1991; Maclachlan & Atela, 1992].

A symplectic transformation satisfies

$$M^T J M = J, \tag{80}$$

where M is the Jacobian of the map for the integration step, and J is the matrix

$$\begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}, \tag{81}$$

with I being the identity matrix. Preservation of the symplectic form is equivalent to preservation of the Poisson bracket operation, and Louiville's theorem is a consequence of it.

Many different symplectic algorithms have been developed and discussed, and many of them are Runge–Kutta methods [Lasagni, 1988; Sanz-Serna, 1988; Channell & Scovel, 1990; Forest & Ruth, 1990; Yoshida, 1990; Candy & Rozmus, 1991; Sanz-Serna & Abia, 1991; Maclachlan & Atela, 1992]. Explicit symplectic Runge–Kutta methods have been introduced for separable Hamiltonians of the form $H(p,q) = T(p) + V(q)$. Fourth-order explicit symplectic Runge–Kutta methods for this case are discussed by Candy & Rozmus [1991], Forest & Ruth [1990], and Maclachlan & Atela [1992], and sixth-order and eighth-order methods are developed by Yoshida [1990]. In the special separable case where $T(p) = p^2/2$, and we have a Hamiltonian of potential form, even more accurate methods of fourth-order and fifth-order have been developed by Maclachlan & Atela [1992]. No explicit symplectic Runge–Kutta methods exist for general Hamiltonians which are not separable. Lasagni [1988] and Sanz-Serna [1988] both discovered that the implicit Gauss–Legendre Runge–Kutta methods are symplectic. Maclachlan & Atela [1992] find these Gauss–Legendre Runge–Kutta methods to be optimal for general Hamiltonians. Thus symplectic integration proves to be a situation where implicit Runge–Kutta methods find a use, despite the computational penalty involved in implementing them compared to explicit methods.

A positive experience with practical use of these methods in a problem from cosmology has been reported by Santillan Iturres *et al.* [1992]. They have used the methods described by Channell & Scovel [1990] to integrate a rather complex Hamiltonian, discovering a structure (suspected to be there from nonnumerical arguments) which nonsymplectic methods were unable to reveal.

Although symplectic methods of integration are undoubtedly to be preferred in dealing with Hamiltonian systems, it should not be supposed that they solve all the difficulties of integrating them; they are not perfect. Channell & Scovel [1990] give examples of local structure introduced by the discretization. For another example, integration of an integrable Hamiltonian system, where the solution of Newton's equations is reducible to the solution of a set of simultaneous equations, followed by integration over single variables, and trajectories lie on invariant tori, will cause a nonintegrable perturbation to the system. For a small perturbation, however, such as we should get from a good symplectic integrator, the KAM theorem tells us that most of the invariant tori will survive. Nevertheless, the dynamical behaviour of the symplectic map is qualitatively different to that of the original system, since in addition to invariant tori, the symplectic map will possess island chains surrounded by stochastic layers. Thus the numerical method perturbing the nongeneric integrable system restores genericity.

There is a more important reason why care is needed in integrating Hamiltonian systems, even with symplectic maps, and that is the lack of energy conservation in the map. It would seem to be an obvious goal for a Hamiltonian integration method both to preserve the symplectic structure and to conserve the energy, but it has been shown that this is in general impossible, because the symplectic map with step length h would then have to be the exact time-h map of the original Hamiltonian. Thus a symplectic map which only approximates a Hamiltonian cannot conserve energy [Zhong & Marsden, 1988; MacKay, 1990; Marsden *et al.*, 1991]. Algorithms have been given which are energy conserving at the expense of not being symplectic, but for most applications retaining the Hamiltonian structure is more important than energy conservation. Marsden *et al.* [1991] mention an example where using an energy-conserving algorithm to integrate the equations of motion of a rod which can both rotate and vibrate leads to the absurd conclusion that rotation will virtually cease almost immediately in favour of vibration.

In fact, the symplectic map with step length h is the exact time-h map of a time-dependent Hamiltonian $\hat{H}(p,q,t)$ with period h, and is near to the time-h map of the original Hamiltonian $H(p,q)$, so that the quantity $\|H - \hat{H}\|$, which is measuring energy conservation, is a good guide to the accuracy of the method. In many cases, the lack of energy conservation is not too much of a problem, because if the system is close to being integrable, and has less than two degrees of freedom, there will be invariant tori in the symplectic map which the orbits cannot cross, and so the energy can only undergo bounded oscillations. This is in contrast to integrating the same system with a nonsymplectic method, where there would be no bound on the energy, which could then increase without limit. This is a major advantage of symplectic methods. However, consider a system which has two degrees of freedom. The phase space of the symplectic map is extended compared to that of the original system, so that an N-degree-of-freedom system becomes an $(N+1)$-degree-of-freedom map. (The extra degree of freedom comes from t and $\hat{H}$.) Now in the case where $N=2$, the original system, if it were near integrable, would have two-dimensional invariant tori acting as boundaries to motion in the three-dimensional energy shell, but in the map the extra degree of freedom would mean that the three-dimensional invariant tori here would no longer be boundaries to motion in the five-dimensional energy shell, so Arnold diffusion would occur. This is a major qualitative difference between the original system and the numerical approximation. It has been shown to occur for two coupled pendulums by Maclachlan & Atela [1992], and proves that symplectic methods should not blindly be relied upon to provide predictions of long-time behaviour for Hamiltonian systems.

There is a further point about symplectic maps that affects all numerical methods using floating-point arithmetic, and that is round-off error. Round-off error is a particular problem for Hamiltonian systems, because it introduces non-Hamiltonian perturbations despite the use of symplectic integrators. The fact that symplectic methods do produce behaviour that looks Hamiltonian shows that the non-Hamiltonian perturbations are much smaller than those introduced by nonsymplectic methods. However, it is shown by Earn & Tremaine [1992] that round-off error does adversely affect the long-term behaviour of Hamiltonian maps like the standard map, by introducing dissipation. To iterate the map they instead use integer arithmetic with Hamiltonian maps on a lattice that they construct to be better and better approximations to the original map as the lattice spacing is decreased. They show that these lattice maps are superior to floating-point maps for Hamiltonian systems. Possibly a combination of the techniques of symplectic methods and lattice maps may lead to the numerical integration of Hamiltonian systems being possible without any non-Hamiltonian perturbations.

10. Conclusions

RUNGE–KUTTA integration schemes should be applied to nonlinear systems with knowledge of the caveats involved. The absolute-stability boundaries may be very different from the linear case, so a linear stability analysis may well be misleading. A problem may occur if a reduction in step length happens to take one outside the absolute-stability region due to the shape of the boundary. In this case, the usual step-control schemes would have disastrous results on the problem, as step-length reduction in an attempt to increase accuracy would have the opposite effect.

Even inside the absolute-stability boundary, all may not be well due to the existence of stable ghost fixed points in many problems. Since basin boundaries are finite, starting too far from the real solution may land one in the basin of attraction of a ghost fixed point. Contrary to expectation, this incorrect behaviour is not prevented by insisting that the method be convergent.

Stiffness needs a new and better definition for nonlinear systems. We have provided a verbal description, but a mathematical definition is still lacking. There is a lot of scope to investigate further the interaction between stiffness and chaos. Explicit Runge–Kutta schemes should not be used for stiff problems, due to their inefficiency: Backward Differentiation Formulae methods, or possibly implicit Runge–Kutta methods, should be used instead.

Dynamics is not only interested in problems with fixed point solutions, but also in periodic and chaotic behaviour. This is something that has not in the past been fully appreciated by some workers in numerical analysis who have tended to concentrate on obtaining results, such as those of nonlinear stability theory, that require properties like contractivity which are too restrictive for most dynamical systems.

There are results that tie the limit set of the Runge–Kutta map to that of the ordinary differential equation from which it came, but they are not as powerful as those which relate the dynamics of Poincaré maps to their differential equations. Structurally-stable behaviour in the ordinary differential equation is correctly portrayed in the Runge–Kutta map, but additional limit-set behaviour may be found in the map that is not present in the differential equation. Nonhyperbolic behaviour will probably not be correctly represented by the Runge–Kutta method.

Shadowing theory offers hope that it will be possible to produce numerical methods with built-in proofs of correctness of the orbits they produce, at least for hyperbolic orbits. However, it does not seem to be possible to do this with Runge–Kutta methods, and the Taylor series method, for which it is possible, has some severe disadvantages. More research needs to be done in this area.

Hamiltonian systems should be integrated with symplectic Runge–Kutta methods so that dissipative perturbations are not introduced. Even using symplectic integration, Hamiltonian systems still need to be handled with care. As in dissipative systems, nongeneric behaviour like integrability will not be reproduced in the numerical method. A more general problem is that approximate symplectic integrators cannot conserve energy. Round-off error is more of a problem for symplectic integration than in other cases, because it introduces dissipative perturbations to the system that one is trying to avoid.

A lot more work is needed on predicting the stability and accuracy of methods for integrating nonlinear and chaotic systems. At present, we must make do with Runge–Kutta and other methods, but be wary of the results they are giving us—caveat emptor!

Acknowledgements

WE should like to acknowledge the helpful suggestions of David Arrowsmith who has read previous versions of this paper. We would also like to thank Shaun Bullett and Chris Penrose with whom we have had useful discussions about complex maps, and Nik Buric who has helped greatly in the clarification of various points. Thanks also go to Carl Murray for help in producing some of the illustrations. JHEC would like to acknowledge the support of the Science and Engineering Research Council (SERC) and the AEJMC Foundation. OP would like to acknowledge the support of the Wolfson Foundation and CONICET.

References

Aiken, R. C., editor [1985] *Stiff Computation* (Oxford University Press).

Aronson, D. G., Chory, M. A., Hall, G. R. & McGehee, R. P. [1983] "Bifurcations from an invariant circle for two-parameter families of maps of the plane: A computer-assisted study," *Commun. Math. Phys.* **83**, 303.

Arrowsmith, D. K., Cartwright, J. H. E., Lansbury, A. N. & Place, C. M. [1992] "The Bogdanov map: Bifurcations, mode locking, and chaos in a dissipative system," *Int. J. Bifurcation and Chaos, submitted*.

Auerbach, S. P. & Friedmann, A. [1991] "Long-term behaviour of numerically computed orbits: Small and intermediate timestep analysis of one-dimensional systems," *J. Comput. Phys.* **93**, 189.

Beyn, W.-J. [1987a] "On invariant closed curves for one-step methods," *Numer. Math.* **51**, 103.

Beyn, W.-J. [1987b] "On the numerical approximation of phase portraits near stationary points," *SIAM J. Num. Anal.* **24**, 1095.

Butcher, J. C. [1987] *The Numerical Analysis of Ordinary Differential Equations: Runge–Kutta and General Linear Methods* (Wiley).

Candy, J. & Rozmus, W. [1991] "A symplectic integration algorithm for separable Hamiltonian systems," *J. Comput. Phys.* **92**, 230.

Channell, P. J. & Scovel, C. [1990] "Symplectic integration of Hamiltonian systems," *Nonlinearity* **3**, 231.

Chua, L. O. & Lin, P. M. [1975] *Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques* (Prentice-Hall).

Devaney, R. L. [1989] *An Introduction to Chaotic Dynamical Systems* (Addison–Wesley) second edition.

Earn, D. J. D. & Tremaine, S. [1992] "Exact numerical studies of Hamiltonian Maps: Iterating without roundoff error," *Physica* **56D**, 1.

Feng, K. [1986] "Difference schemes for Hamiltonian formalism and symplectic geometry," *J. Comput. Math.* **4**, 279.

Feng, K. & Qin, M.–z. [1991] "Hamiltonian algorithms for Hamiltonian systems and a comparative numerical study," *Comput. Phys. Commun.* **65**, 173.

Forest, E. & Ruth, R. D. [1990] "Fourth order symplectic integration," *Physica* **43D**, 105.

Gardini, L., Lupini, R., Mammana, C. & Messia, M. G. [1987] "Bifurcations and transition to chaos in the three-dimensional Lotka–Volterra map," *SIAM J. Appl. Math.* **47**, 455.

Gear, C. W. [1971] *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice-Hall).

Hall, G. & Watt, J. M. [1976] *Modern Numerical Methods for Ordinary Differential Equations* (Oxford University Press).

Hammel, S. M., Yorke, J. A. & Gregobi, C. [1988] "Numerical orbits of chaotic processes represent true orbits," *Bull. Am. Math. Soc.* **19**, 465.

Henrici, P. [1962] *Discrete Variable Methods in Ordinary Differential Equations* (Wiley).

Iserles, A. [1990] "Stability and dynamics of numerical methods for nonlinear ordinary differential equations," *IMA J. Num. Anal.* **10**, 1.

Itoh, T. & Abe, K. [1988] "Hamiltonian-conserving discrete canonical equations based on variational difference quotients," *J. Comput. Phys.* **76**, 85.

Jackson, E. A. [1989] *Perspectives of Nonlinear Dynamics*, vol. 1 (Cambridge University Press).

Kloeden, P. E. & Lorenz, J. [1986] "Stable attracting sets in dynamical systems and their one-step discretizations," *SIAM J. Num. Anal.* **23**, 986.

Lambert, J. D. [1973] *Computational Methods in Ordinary Differential Equations* (Wiley).

Lambert, J. D. [1991] *Numerical Methods for Ordinary Differential Systems* (Wiley).

Lasagni, F. M. [1988] "Canonical Runge–Kutta methods," *ZAMP* **39**, 952.

MacKay, R. S. [1990] "Some aspects of the dynamics and numerics of Hamiltonian systems," in *Dynamics of Numerics and Numerics of Dynamics* IMA.

Maclachlan, R. I. & Atela, P. [1992] "The accuracy of symplectic integrators," *Nonlinearity* **5**, 541.

Marsden, J. E., O'Reilly, O. M., Wicklin, F. W. & Zombro, B. W. [1991] "Symmetry, stability, geometric phases and mechanical integrators," *Preprint*.

Menyuk, C. R. [1984] "Some properties of the discrete Hamiltonian method," *Physica* **11D**, 109.

Miller, R. H. [1991] "A horror story about integration methods," *J. Comput. Phys.* **93**, 469.

Parker, T. S. & Chua, L. O. [1989] *Practical Numerical Algorithms for Chaotic Systems* (Springer).

Peitgen, H.-O. & Richter, P. H. [1986] *The Beauty of Fractals*, chapter 8 'A Discrete Volterra–Lotka System', ,p. 125 (Springer).

Press, W. H., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. A. [1988] *Numerical Recipes in C* (Cambridge University Press).

Prüfer, M. [1985] "Turbulence in multistep methods for initial value problems," *SIAM J. Appl. Math.* **45**, 32.

Santillan Iturres, A., Domenech, G., El Hasi, C., Vucetich, H. & Piro, O. [1992] *preprint*.

Sanz-Serna, J. M. [1988] "Runge–Kutta schemes for Hamiltonian systems," *BIT* **28**, 877.

Sanz-Serna, J. M. & Abia, L. [1991] "Order conditions for canonical Runge–Kutta schemes," *SIAM J. Num. Anal.* **28**, 1081.

Sanz-Serna, J. M. & Vadillo, F. [1987] "Studies in numerical nonlinear instability III: Augmented Hamiltonian systems," *SIAM J. Appl. Math.* **47**, 92.

Sauer, T. & Yorke, J. A. [1991] "Rigorous verification of trajectories for the computer simulation of dynamical systems," *Nonlinearity* **4**, 961.

Stetter, H. J. [1973] *Analysis of Discretization Methods for Ordinary Differential Equations* (Springer).

Stewart, I. [1992] "Numerical methods: Warning—handle with care!," *Nature* **355**, 16.

Thompson, J. M. T. & Stewart, H. B. [1986] *Nonlinear Dynamics and Chaos* (Wiley).

Tomita, K. [1986] "Periodically forced nonlinear oscillators," in Holden, A. V., editor, *Chaos* (Manchester University Press).

Ushiki, S. [1982] "Central difference scheme and chaos," *Physica* **4D**, 407.

Yamaguti, M. & Ushiki, S. [1981] "Chaos in numerical analysis of ordinary differential equations," *Physica* **3D**, 618.

Yee, H. C., Sweby, P. K. & Griffiths, D. F. [1991] "Dynamical approach study of spurious steady-state numerical solutions of nonlinear differential equations. I. The dynamics of time discretization and its implications for algorithmic development in computational fluid dynamics," *J. Comput. Phys.* **47**, 249.

Yoshida, H. [1990] "Construction of higher order symplectic integrators," *Phys. Lett. A* **150**, 262.

Zhong, G. & Marsden, J. [1988] "Lie–Poisson Hamilton–Jacobi theory and Lie–Poisson integrators," *Phys. Lett. A* **133**, 134.

A.1. Convergence of Runge–Kutta methods

To prove that consistency is necessary and sufficient for convergence of Runge–Kutta methods, we follow Henrici [1962]. Let $g(x, y) = \phi(x, y; 0)$ satisfy a Lipschitz condition as in Eq.(3) so that $u' = g(x, u)$ with $u(a) = \alpha$ has a unique solution $u(x)$. Using the mean value theorem:

$$\begin{aligned} u_{n+1} - u_n &= u(x_n + h) - u(x_n) && (82)\\ &= hu'(x_n + \theta h), \qquad \theta \in [0, 1] && (83)\\ &= hg(x_n + \theta h, u(x_n + \theta h)). && (84) \end{aligned}$$

Now define $U_0 = \alpha$ and $U_{n+1} = U_n + h\phi(x_n, U_n; h)$. Let $e_n = \|u_n - U_n\|$ then

$$\begin{aligned} e_{n+1} &= \|u_{n+1} - U_{n+1}\| && (85)\\ &= \|u_n + hg(x_n + \theta h, u(x_n + \theta h)) - U_n - h\phi(x_n, U_n; h)\| && (86)\\ &\leqslant e_n + h\left(\|g(x_n + \theta h, u(x_n + \theta h)) - \phi(x_n, U_n; h)\|\right). && (87) \end{aligned}$$

We can write the part in parentheses as

$$\begin{aligned} &\|g(x_n + \theta h, u(x_n + \theta h)) - \phi(x_n, U_n; h)\| = \\ &\qquad \|g(x_n + \theta h, u(x_n + \theta h)) - g(x_n, u_n) \\ &\qquad\qquad + \phi(x_n, u_n; 0) - \phi(x_n, u_n; h) \\ &\qquad\qquad + \phi(x_n, u_n; h) - \phi(x_n, U_n; h)\|. \qquad (88) \end{aligned}$$

Let

$$\chi(h) = \max_{\substack{x\in[a,b]\\ \theta\in[0,1]}} \|g(x + \theta h, u(x + \theta h)) - g(x, u)\| \qquad (89)$$

and

$$\zeta(h) = \max_{x\in[a,b]} \|\phi(x, u; 0) - \phi(x, u; h)\|. \qquad (90)$$

If $\phi(x, y; h)$ is continuous and satisfies a Lipschitz condition, then

$$\|\phi(x_n, u_n; h) - \phi(x_n, U_n; h)\| \leqslant L\|u_n - U_n\|. \qquad (91)$$

Thus

$$\begin{aligned} e_{n+1} &\leqslant e_n + h(\chi(h) + \zeta(h) + Le_n) && (92)\\ &\leqslant (1 + hL)e_n + h(\chi(h) + \zeta(h)), && (93) \end{aligned}$$

and so, since $1 + hL$ and $h(\chi(h) + \zeta(h))$ are both positive,

$$e_n \leqslant (1 + hL)^n e_0 + \frac{(1 + hL)^n - 1}{hL} h(\chi(h) + \zeta(h)). \qquad (94)$$

Now $u_0 = U_0$ so $e_0 = 0$. Thus

$$e_n \leqslant \frac{(1 + hL)^n - 1}{L}(\chi(h) + \zeta(h)). \qquad (95)$$

We now look at the fixed-station limit as in Eq.(28):

$$\lim_{\substack{h\to 0\\ nh=x-a}} e_n \leqslant \lim_{\substack{h\to 0\\ nh=x-a}} \frac{(1+hL)^n-1}{L}(\chi(h)+\zeta(h)). \tag{96}$$

Since $g(x,y)$ is continuous,

$$\lim_{\substack{h\to 0\\ nh=x-a}} \chi(h)=0. \tag{97}$$

For the same reason

$$\lim_{\substack{h\to 0\\ nh=x-a}} \zeta(h)=0. \tag{98}$$

Thus we obtain

$$\lim_{\substack{h\to 0\\ nh=x-a}} e_n=0, \tag{99}$$

which shows that consistency is sufficient for convergence since if $f(x,y)=g(x,y)$ then $u(x)=y(x)$. To establish its necessity, let us assume that the method is convergent but that $g(x,y)\neq f(x,y)$ at some point (x,y). There exists an α such that $y(w)$, the solution of the initial value problem, passes through (x,y). U_n as defined above then converges in the limit to $y(w)$, and also, as we proved above, to $u(w)$. If $u(x)\neq y(x)$, then immediately there is a contradiction. Otherwise, if $u(x)=y(x)$ then $u'(x)=y'(x)$ where $u'(x)=g(x,y)$ and $y'(x)=f(x,y)$, but $g(x,y)\neq f(x,y)$ which is again a contradiction. Thus we have proved that consistency is necessary and sufficient for convergence, and it follows from Eq.(30) that all Runge–Kutta methods are convergent.

A.2. Contractivity and Lyapunov Exponents

A.2(A). *Continuous systems*

The Lyapunov exponents of $y(x_0)$ are defined for continuous systems by

$$\lambda_i=\lim_{x\to\infty}\frac{1}{x}\ln|m_i(x)|, \qquad i=1,\ldots,m \tag{100}$$

whenever this limit exists. Here $m_i(x)$ are the eigenvalues of $\Phi_x(y(x_0),x_0)$, where Φ comes from the variational equation

$$\frac{d\Phi_x(x_0,y_0)}{dx}=J\Phi_x(x_0,y_0), \tag{101}$$

where J is the Jacobian matrix. One can show (see, for example, Parker & Chua [1989]) that a perturbation grows as

$$\delta y(x)=\Phi_x(y(x_0),x_0)\delta y(x_0). \tag{102}$$

Taking the norm of both sides,

$$\begin{aligned} \|\delta y(x)\| &= \|\Phi_x(y(x_0), x_0)\delta y(x_0)\| && (103) \\ &\leqslant \|\Phi_x(y(x_0), x_0)\|\|\delta y(x_0)\|. && (104) \end{aligned}$$

Now the perturbation $\delta y(x_0) = y(x_0) - \tilde{y}(x_0)$ and $\delta y(x) = y(x) - \tilde{y}(x)$ so

$$\|y(x) - \tilde{y}(x)\| \leqslant \|\Phi_x(y(x_0), x_0)\|\|y(x_0) - \tilde{y}(x_0)\|. \qquad (105)$$

We can see from Eq.(72) that contractivity is asking that $\|\Phi_x(y(x_0), x_0)\| \leqslant 1$. From properties of the matrix norm, we know that

$$|m_i(x)| \leqslant \|\Phi_x(y(x_0), x_0)\|, \qquad (106)$$

so contractivity is equivalent to $|m_i(x)| \leqslant 1$, or $(1/x)\ln|m_i(x)| \leqslant 0$. Thus from Eq.(100), contractivity is sufficient to give nonpositive Lyapunov exponents and thence regular motion. (Note that the reverse is not necessarily true.)

A.2(B). *Discrete systems*

The Lyapunov exponents of Y_0 are defined for discrete systems by

$$\lambda_i = \lim_{n\to\infty} \frac{1}{n} \ln |m_i(n)|, \qquad i = i, \ldots, m, \qquad (107)$$

whenever this limit exists. Here $m_i(n)$ are the eigenvalues of

$$\left\{\prod_{i=1}^{n} J(x_{n-i}, Y_{n-i})\right\}^{\frac{1}{n}} \qquad (108)$$

where J is the Jacobian matrix. One can show (see, for example, Parker & Chua [1989]) that a perturbation grows as

$$\delta Y_{n+1} = J(x_n, Y_n)\delta Y_n. \qquad (109)$$

Taking the norm of both sides,

$$\begin{aligned} \|\delta Y_{n+1}\| &= \|J(x_n, Y_n)\delta Y_n\| && (110) \\ &\leqslant \|J(x_n, Y_n)\|\|\delta Y_n\|. && (111) \end{aligned}$$

Now the perturbation $\delta Y_n = Y_n - \tilde{Y}_n$ and $\delta Y_{n+1} = Y_{n+1} - \tilde{Y}_{n+1}$ so

$$\|Y_{n+1} - \tilde{Y}_{n+1}\| \leqslant \|J(x_n, Y_n)\|\|Y_n - \tilde{Y}_n\|. \qquad (112)$$

We can see from Eq.(73) that contractivity is asking that $\|J(x_n, Y_n)\| \leqslant 1$. From properties of the matrix norm, we know that if $\|J(x_n, Y_n)\| \leqslant 1$ then

$$\left\|\left\{\prod_{i=1}^{n} J(x_{n-i}, Y_{n-i})\right\}^{\frac{1}{n}}\right\| \leqslant 1 \qquad (113)$$

and

$$|m_i(n)| \leqslant \left\| \left\{ \prod_{i=1}^{n} J(x_{n-i}, Y_{n-i}) \right\}^{\frac{1}{n}} \right\|, \tag{114}$$

so contractivity is equivalent to $|m_i(n)| \leqslant 1$, or $(1/n)\ln|m_i(n)| \leqslant 0$. Thus from Eq.(107), contractivity is sufficient to give nonpositive Lyapunov exponents and thence regular motion. (Note that the reverse is not necessarily true.)