ELSEVIER

Journal of Computational and Applied Mathematics 125 (2000) 1–29

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Numerical methods for ordinary differential equations in the 20th century

J.C. Butcher *

The University of Auckland, Department of Mathematics, Private Bag 92019, Auckland, New Zealand

Received 7 July 1999; received in revised form 18 November 1999

Abstract

Numerical methods for the solution of initial value problems in ordinary differential equations made enormous progress during the 20th century for several reasons. The first reasons lie in the impetus that was given to the subject in the concluding years of the previous century by the seminal papers of Bashforth and Adams for linear multistep methods and Runge for Runge–Kutta methods. Other reasons, which of course apply to numerical analysis in general, are in the invention of electronic computers half way through the century and the needs in mathematical modelling of efficient numerical algorithms as an alternative to classical methods of applied mathematics. This survey paper follows many of the main strands in the developments of these methods, both for general problems, stiff systems, and for many of the special problem types that have been gaining in significance as the century draws to an end.

Keywords: Initial value problems; Adams–Bashforth method; Adams–Moulton method; Runge–Kutta method; Consistency; Stability and convergence; Order of methods; Stiff problems; Differential equation software

1. Introduction

It is not possible to assess the history of this subject in the 20th century without first recognizing the legacy of the previous century on which it has been built. Notable are the 1883 paper of Bashforth and Adams [5] and the 1895 paper of Runge [57]. Not only did the former present the famous Adams–Bashforth method, which plays an essential part in much modern software, but it also looked ahead to the Adams–Moulton method and to the practical use of Taylor series methods. The paper by Runge is now recognized as the starting point for modern one-step methods. These early contributions, together with a brief introduction to the fundamental work of Euler, will form the subject matter of Section 2.

* Tel.: +64-9-373-7599; fax: +64-9-373-7457.
E-mail address: butcher@scitec.auckland.ac.nz (J.C. Butcher).

0377-0427/00/$ - see front matter
PII: S0377-0427(00)00455-6

These early papers each formulates the general initial value problem in much the same form. That is, given a function $f(x, y)$ and an "initial value" y_0, corresponding to a solution value at x_0, we seek to evaluate numerically the function y satisfying

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0. \tag{1}$$

The basic approach is to extend the set of x values for which an approximation to $y(x)$ is known, in a step-by-step fashion.

In the early writing on this problem, y is regarded as a scalar value function but the generalization to more general problems is suggested by a consideration of a pair of simultaneous equations

$$y'(x) = f(x, y(x), z(x)), \quad y(x_0) = y_0,$$

$$z'(x) = g(x, y(x), z(x)), \quad z(x_0) = z_0.$$

Today it is more natural to use formulation (1) but to interpret y as a vector-valued function. In this case, it is even possible to consider an autonomous system of differential equations

$$y'(x) = f(y(x)), \tag{2}$$

because, if necessary, x can be appended to $y(x)$ as an additional component satisfying the trivial differential equation $\mathrm{d}x/\mathrm{d}x = 1$.

After the section dealing with 19th century contributions, this review paper is divided into a number of further sections dealing either with specific periods of time or with contributions with a unifying theme. The development of algorithms based on linear multistep methods continued with the paper of Moulton [49] and to the predictor–corrector formulation together with local error estimation using Milne's device. This will be discussed in Section 2.6.

Sections follow on Runge–Kutta methods and on Taylor series methods. Special methods are needed for stiff problems, and we review some of the stability and other issues involved with the phenomenon of stiffness in Section 6. The development of software to solve initial value problems is discussed in Section 7. Finally, we discuss in Section 8 a number of identifiable problem classes that call for special techniques and special methods.

2. Early work on numerical ordinary differential equations

2.1. The Adams–Bashforth paper

The famous booklet by Bashforth and Adams [5] has a very long title but, when this is broken into two halves, as it appears on the title page, Fig. 1, the authorship of the two distinct aspects of the work is clearly ascribed to the separate authors. Thus, we may assume that the numerical component of this work is due to Mr Adams.

The numerical discussion begins by pointing out that given, for example, a second-order differential equation

$$\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} = f\left(\frac{\mathrm{d}y}{\mathrm{d}t}, y, t\right),$$

AN ATTEMPT

TO TEST

THE THEORIES OF CAPILLARY ACTION

BY COMPARING

THE THEORETICAL AND MEASURED FORMS
OF DROPS OF FLUID,

BY

FRANCIS BASHFORTH, B.D.

LATE PROFESSOR OF APPLIED MATHEMATICS TO THE ADVANCED CLASS
OF ROYAL ARTILLERY OFFICERS, WOOLWICH,
AND FORMERLY FELLOW OF ST JOHN'S COLLEGE, CAMBRIDGE.

WITH

AN EXPLANATION OF THE METHOD OF INTEGRATION
EMPLOYED IN CONSTRUCTING THE TABLES WHICH GIVE THE THEORETICAL
FORMS OF SUCH DROPS,

BY

J. C. ADAMS, M.A, F.R.S.

FELLOW OF PEMBROKE COLLEGE, AND LOWNDEAN PROFESSOR OF ASTRONOMY AND GEOMETRY
IN THE UNIVERSITY OF CAMBRIDGE.

Cambridge:
AT THE UNIVERSITY PRESS.
1883

Fig. 1. The title page of the Adams–Bashforth paper.

it is possible to find, by repeated differentiation and substitution of $\mathrm{d}^2y/\mathrm{d}t^2$ into the result, formulas for

$$\frac{\mathrm{d}^3y}{\mathrm{d}t^3}, \frac{\mathrm{d}^4y}{\mathrm{d}t^4}, \dots .$$

From these data evaluated at the initial value, the solution may then be advanced using the Taylor series. Hence, after a small time-step, values of y and of $\mathrm{d}y/\mathrm{d}t$ can be found. Further steps can then be taken in the same manner until a desired value of t is reached.

After these remarks, Adams goes on to derive the Adams–Bashforth method, as we know it today, in the form

$$y_1 - y_0 = \omega\left(q_0 + \tfrac{1}{2}\Delta q_0 + \tfrac{5}{12}\Delta^2 q_0 + \cdots\right), \tag{3}$$

where ω is the stepsize and $q_0, q_{-1}, \dots$ denote the derivatives computed at the points $t_0, t_{-1}, \dots$ where the solution values are $y_0, y_{-1}, \dots$. In the Adams notation, Δ denotes the backward difference $\Delta q_0 = q_0 - q_{-1}$, in contrast to the modern terminology of reserving Δ for the forward difference and using ∇ for the backward difference.

Adams goes on to discuss the relative merits of using, instead of (3), the formula

$$y_0 - y_{-1} = \omega\left(q_0 - \tfrac{1}{2}\Delta q_0 - \tfrac{1}{12}\Delta^2 q_0 + \cdots\right). \tag{4}$$

He correctly observes the advantages of (4) in terms of magnitudes of the error constants. The use of this implicit form of the Adams method was revisited and developed many years later by Moulton [49].

Eine ähnliche Ueberlegung führt nun auch für die Differentialgleichungen zu einer wesentlichen Verbesserung des Euler'schen Verfahrens. Ich will mich zunächst auf Differentialgleichungen erster Ordnung beschränken.

Statt

(1) $$\Delta y = f(x_0 y_0)\,\Delta x \text{ u. s. w.}$$

ist es schon viel besser wenn man

(2) $$\Delta y = f\left(x_0 + \tfrac{1}{2}\Delta x,\ y_0 + \tfrac{1}{2} f(x_0 y_0)\,\Delta x\right)\Delta x$$

u. s. w.

setzt. Diese Art der Berechnung entspricht dem aus der Summe der Tangententrapeze gebildeten Näherungswerthe eines Integrals und deckt sich völlig damit, wenn $f(xy)$ von y unabhängig vorausgesetzt wird.

Oder man kann der Summe der Sehnentrapeze entsprechend setzen:

(3) $$\Delta y = \frac{f(x_0 y_0) + f\left(x_0 + \Delta x,\ y_0 + f(x_0 y_0)\,\Delta x\right)}{2}\,\Delta x$$

u. s. w.

Fig. 2. An extract from the Runge paper.

2.2. The Runge paper

The second great legacy of the 19th century to numerical methods for ordinary differential equations was the work of Runge [57]. Whereas the Adams method was based on the approximation of the solution value for given x, in terms of a number of previously computed points, the approach of Runge was to restrict the algorithm to being "one step", in the sense that each approximation was based only on the most recent point already computed in a previous step. To achieve the required accuracy, approximations are found at a number of internal points within each step and the final result is computed in terms of these various stage values. The short extract from Runge's paper given in Fig. 2, includes the formulations of methods with two derivative calculations per step, based on the mid-point and trapezoidal quadrature rules, respectively.

2.3. The contributions of Heun and Kutta

Following the important and prophetic work of Adams and of Runge, the new century began with further contributions to what is now known as the Runge–Kutta method, by Heun [40] and Kutta [45]. In particular, the famous method in Kutta's paper is often known as *the* Runge–Kutta method. Heun's contribution was to raise the order of the method from two and three, as in Runge's paper, to four. This is an especially significant contribution because, for the first time, numerical methods for differential equations went beyond the use of what are essentially quadrature formulas. Even though second-order Runge methods can be looked at in this light, because the derivatives of the solution are computed from accurate enough approximations so as not to disturb the second-order behaviour, this is no longer true for orders greater than this. Write a three stage method in the form

$$Y_1 = y_0, \quad F_1 = f(x_0, Y_1),$$

$$Y_2 = y_0 + ha_{21}F_1, \quad F_2 = f(x_0 + hc_2, Y_2),$$

$$Y_3 = y_0 + h(a_{31}F_1 + a_{32}F_2), \quad F_3 = f(x_0 + hc_3, Y_3),$$

$$y_1 = y_0 + h(b_1F_1 + b_2F_2 + b_3F_3),$$

where a_{21}, a_{31}, a_{32}, b_1, b_2, b_3, c_2, c_3 are constants that characterize a particular method in this family. We can view computation of the stage values Y_1, identical to the initial value for the step, Y_2, which approximates the solution at x_0+hc_2 and Y_3, which approximates the solution at x_0+hc_3 as temporary steps, whose only purpose is to permit the evaluation of F_1, F_2 and F_3 as approximations to $y'(x_0)$, $y'(x_0+hc_2)$ and $y'(x_0+hc_3)$, respectively. From these derivative approximations, the result at the end of the step is found from the quadrature approximation

$$y(x_0+h) \approx y(x_0) + h(b_1y'(x_0) + b_2y'(x_0+hc_2) + b_3y'(x_0+hc_3)).$$

It is essential that this quadrature formula be sufficiently accurate to integrate polynomials of degree up to 2 exactly. This gives the conditions

$$b_1 + b_2 + b_3 = 1,$$

$$b_2c_2 + b_3c_3 = \tfrac{1}{2},$$

$$b_2c_2^2 + b_3c_3^3 = \tfrac{1}{3}.$$

However, because of possible inaccuracies in the computation of Y_2 and Y_3 as approximations to $y(x_0+hc_2)$ and $y(x_0+hc_3)$, respectively, the quadrature conditions are not enough and it is also necessary that

$$b_3a_{32}c_2 = \tfrac{1}{6},$$

to obtain third-order behaviour.

An example of a method due to Heun which satisfies the four conditions for this order uses the coefficients

$$c_2 = \tfrac{1}{3}, \quad c_3 = 1, \quad a_{21} = \tfrac{1}{3}, \quad a_{31} = 0, \quad a_{32} = \tfrac{2}{3}, \quad b_1 = \tfrac{1}{4}, \quad b_2 = 0, \quad b_3 = \tfrac{3}{4}.$$

Kutta took this investigation further and found a complete classification of the solutions to the eight conditions for four-stage methods with order 4. He also derived the 16 conditions for order 5.

The extract of Kutta's paper given in Fig. 3, includes the formulation of the method, together with the order conditions and the first line of the solution in the case that 0, c_2, c_3 and c_4 are all distinct numbers. In his notation we see that $\kappa = c_2$, $\lambda = c_3$ and $\mu = c_4$. It is an interesting consequence of these order conditions, that μ is necessarily equal to 1.

Of the various four stages, fourth-order methods derived by Kutta, the most famous, and also the most widely used, is

$$\begin{aligned}
Y_1 &= y_0, \quad F_1 = f(x_0, Y_1),\\
Y_2 &= y_0 + \tfrac{1}{2}hF_1, \quad F_2 = f(x_0 + \tfrac{h}{2}, Y_2),\\
Y_3 &= y_0 + \tfrac{1}{2}hF_2, \quad F_3 = f(x_0 + \tfrac{h}{2}, Y_3),\\
Y_4 &= y_0 + hF_3, \quad F_4 = f(x_0 + h, Y_4),\\
y_1 &= y_0 + h(\tfrac{1}{6}F_1 + \tfrac{1}{3}F_2 + \tfrac{1}{3}F_3 + \tfrac{1}{6}F_4).
\end{aligned}$$

The set of conditions for fifth-order methods is actually a little more complicated than Kutta realised, because there are actually 17 conditions. The reason for the discrepancy is that he was dealing with scalar differential equations, rather than vector-valued differential equations, and for orders five

Zu den Näherungen vierter Ordnung übergehend finden wir die Berechnung von vier Funktionswerten nötig, und die Vergleichung des Taylorschen Satzes ergiebt die folgenden acht Bedingungsgleichungen für die Koeffizienten:

$$\begin{aligned}
a + b + c + d &= 1,\\
b\varkappa + c\lambda + d\mu &= \tfrac{1}{2},\\
b\varkappa^2 + c\lambda^2 + d\mu^2 &= \tfrac{1}{3},\\
c\varrho\varkappa + d(\sigma\lambda + \tau\varkappa) &= \tfrac{1}{6},\\
b\varkappa^3 + c\lambda^3 + d\mu^3 &= \tfrac{1}{4},\\
c\varrho\varkappa\lambda + d(\sigma\lambda + \tau\varkappa)\mu &= \tfrac{1}{8},\\
c\varrho\varkappa^2 + d(\sigma\lambda^2 + \tau\varkappa^2) &= \tfrac{1}{12},\\
d\varrho\sigma\varkappa &= \tfrac{1}{24},
\end{aligned}$$

wenn als gewünschte Näherung angesetzt ist:

$$\begin{aligned}
\Delta y &= a\Delta' + b\Delta'' + c\Delta''' + d\Delta'''',\\
\Delta' &= f(x, y)\Delta x,\\
\Delta'' &= f(x + \varkappa\Delta x,\ y + \varkappa\Delta')\Delta x,\\
\Delta''' &= f(x + \lambda\Delta x,\ y + \varrho\Delta'' + (\lambda - \varrho)\Delta')\Delta x,\\
\Delta'''' &= f(x + \mu\Delta x,\ y + \sigma\Delta''' + \tau\Delta'' + (\mu - \sigma - \tau)\Delta')\Delta x.
\end{aligned}$$

Hier läfst sich das Lösungssystem auch noch niederschreiben. Wenn man $\varkappa$ und λ willkürlich läfst, erhält man nach einiger Rechnung:

$$c = \frac{1 - 2\varkappa}{12\lambda(\lambda - \varkappa)(1 - \lambda)},\quad b = \frac{1 - 2\lambda}{12\varkappa(\varkappa - \lambda)(1 - \varkappa)},\quad d = \frac{6\varkappa\lambda - 4(\varkappa + \lambda) + 3}{12(1 - \lambda)(1 - \varkappa)};$$

Fig. 3. An extract from the Kutta paper.

or greater the conditions become different. Another difficulty is in actually finding solutions to the algebraic conditions and Kutta presented methods that are slightly incorrect. It is interesting that once the correction is made, the additional condition, to make the method applicable to high-dimensional problems, happens to be satisfied.

2.4. The contributions of E.J. Nyström

The early history of Runge–Kutta methods culminated in the work of Nyström [53] in 1925. He was able to correct some of the fifth-order methods of Kutta and he also showed how to apply the Runge–Kutta method to second-order differential equation systems.

At first sight this is quite straightforward, because every second-order system can be re-formulated as a first-order system with additional dependent variables. However, solving such a problem directly may be much more efficient and the great prevalence of second-order problems in physical modelling makes this sort of gain in efficiency of considerable practical significance.

2.5. Moulton's paper and predictor–corrector methods

Implicit versions of Adams methods were first suggested in the Adams–Bashforth paper, but not studied in their own right until the paper of Moulton [49]. These so-called Adams–Moulton methods

have two great advantages over the original explicit methods. The first is that they do not need to use so many past values to obtain the same order and they have smaller error constants. To use them in practice, however, one first has to overcome the difficulty associated with their implicit nature. This difficulty hinges on the fact that y_n is not given in terms of rational operations on known data, but as the solution to an algebraic equation. For example, consider the third-order Adams–Bashforth and Adams–Moulton methods given by

$$y_n = y_{n-1} + h(\tfrac{23}{12} f(x_{n-1}, y_{n-1}) - \tfrac{4}{3} f(x_{n-2}, y_{n-2}) + \tfrac{5}{12} f(x_{n-3}, y_{n-3})), \tag{5}$$

$$y_n = y_{n-1} + h(\tfrac{5}{12} f(x_n, y_n) + \tfrac{2}{3} f(x_{n-1}, y_{n-1}) - \tfrac{1}{12} f(x_{n-2}, y_{n-2})). \tag{6}$$

It is known that the error introduced into the result in a single step is $-\frac{3}{8} y^{(4)} h^4 + \mathrm{O}(h^5)$ for the Adams–Bashforth method and $\frac{1}{24} y^{(4)} h^4 + \mathrm{O}(h^5)$ for the Adams–Moulton method. The way that advantage is gained from the desirable properties of each of the methods is to use them in "predictor–corrector mode". This means that a predicted value of y_n is first found using the explicit form of the method. The implicit or Moulton form of the method is then used with the term $f(x_n, y_n)$ replaced by the value calculated using the *predicted* value of y_n. There are many variants of this method in common use, but the most popular is the so-called PECE mode. In this mode, $f(x_n, y_n)$ is re-evaluated for use in later steps using y_n found from the Adams–Moulton method. Thus each step requires two evaluations of the function f and is thus twice as expensive as the simple use of the Adams–Bashforth formula alone. However, the advantages in terms of stability and accuracy resulting from the use of this PECE predictor–corrector mode are usually regarded as well worth the additional computing cost.

2.6. The Milne device

Although Milne preferred methods based on Newton–Cotes quadrature formulas, methods which are largely abandoned today in favour of Adams methods, a proposal he made [47] has been adapted to other situations and widely used. In the context of the predictor pair (5) and (6), implemented, for example in PECE mode, there are two approximations to $y(x_n)$ computed in each step. Since the local truncation errors of the two approximations are in the ratio – 9 to 1, it is proposed that the *difference* of the two approximations divided by 10 should be used as an estimate of the error in the corrected formula.

Milne, of course, intended this device to be used to check the accuracy of hand-computed results, but today it is used in automatic solvers, not just to verify the accuracy of any completed step, but also to adjust the size of a subsequent step in the interests both of efficiency and robustness.

Many modern computer codes implement predictor–corrector methods in a different manner than we have described. Specifically, the step number k is chosen to be the same for both the predictor and corrector formulas. This means that the order of the predictor will be k and the order of the corrector, which becomes the overall order of the combined method, will be $p = k + 1$. Even though the difference between the predicted and corrected solutions is no longer asymptotically equal to a multiple of the local truncation error, this difference is still used as the basis for stepsize control.

3. The modern theory of linear multistep methods

The modern analysis of linear multistep methods is intimately bound up with the work of Dahlquist [21,22]. This large body of work is in several parts, of which the first deals with the concepts of consistency, stability and convergence, expressed in terms of generating functions for the coefficients of the method. The key result in this phase of the work, is that consistency and stability are together equivalent to convergence. The second principle phase relates order of accuracy to stability and culminates in the famous "Dahlquist barrier" result, which limits the order of a convergent linear k-step method to $k+1$ (if k is odd) and to $k+2$ (if k is even). The remaining phase of Dahlquist's work is more appropriately discussed in Section 6.

3.1. Generating functions

Consider a linear multistep method of the form

$$\alpha_k y_n + \alpha_{k-1} y_{n-1} + \alpha_{k-2} y_{n-2} + \cdots + \alpha_0 y_{n-k} = h(\beta_k f(x_n, y_n) + \beta_{k-1} f(x_{n-1}, y_{n-1}) + \beta_{k-2} f(x_{n-2}, y_{n-2}) + \cdots + \beta_0 f(x_{n-k}, y_{n-k})),$$

assuming that $\alpha_k \neq 0$ and that α_0 and β_0 are not both zero (otherwise the value of k could be reduced). Such a method is known as a "linear k-step method" because the solution at step number n depends on exactly k previous step values. Dahlquist introduced polynomials ρ and σ to characterize the method as follows:

$$\rho(z) = \alpha_k z^k + \alpha_{k-1} z^{k-1} + \alpha_{k-2} z^{k-2} + \cdots + \alpha_0,$$
$$\sigma(z) = \beta_k z^k + \beta_{k-1} z^{k-1} + \beta_{k-2} z^{k-2} + \cdots + \beta_0.$$

Although Dahlquist allowed for the generality of allowing the coefficient of z^k to take on any non-zero value, in an actual computation with the method, the value of α_k has to be cancelled out from both polynomials.

It is clear that given any linear multistep method, the corresponding pair of polynomials (ρ, σ) can be written down automatically and, given the polynomials, the method is completely specified. Hence, it has become customary to identify the methods with the pair of polynomials and we can speak of "the method (ρ, σ)". It is convenient to assume that ρ and σ have no common polynomial factor, since it would be possible to describe most aspects of the computational behaviour of the method in terms of simpler polynomials. Following Dahlquist, we will make this assumption.

3.2. Consistency, stability and convergence

There are some natural assumptions that can be made about linear multistep methods to guarantee that they can at least solve certain specific problems. We will consider these one, by one.

The first problem is $y'(x) = 0$, with initial value $y(0) = 0$. Since we are given only this single initial value we will need an algorithm to generate y_0, y_1, $y_2, \ldots, y_{k-1}$ which is, in the limit as $h \to 0$, consistent with the given initial data. Choose some $x > 0$, for example $x = 1$, as the point where the numerical result approximating the solution is supposed to be found. We would like our method to be able to compute $y(1)$ exactly in the limiting case as $x \to 0$.

This requirement is equivalent to the "stability condition": A linear multistep method (ρ, σ) is stable if all zeros of ρ lie in the unit disc and all zeros on the boundary are simple.

The second initial value problem is also based on the equation $y'(x) = 0$ but with $y(0) = 1$. To compute the correct result $y(1) = 1$, in the limit, it is necessary that $\rho(1) = 0$. We will refer to this as the "pre-consistency condition".

Finally, consider the initial value problem $y'(x)=1,\ y(0)=0$. If a method is stable and pre-consistent, then its ability to solve this problem in the limit hinges on the requirement that $\rho'(1) = \sigma(1)$. This condition, when combined with the pre-consistency condition, is known as the "consistency condition".

The definition of convergence is rather technical but deals with the ability of the linear multistep method to solve *any* differential equation system on condition only that f is continuous in its first variable and satisfies a Lipschitz condition in its second variable. The k initial approximations required to start the numerical process must converge to the given initial value as the stepsize tends to zero. This class of problems might seem restrictive but it is easy to extend it to many situations where the Lipschitz condition is replaced by a *local* Lipschitz condition.

The basic theorem connecting these concepts is that a method is convergent if and only if it is both stable and consistent. Of course convergence is not enough to ensure that the method is computationally efficient. In the next section we look at the criteria for the method to have some specific order of accuracy and we review a famous result of Dahlquist which imposes a barrier on what order is really achievable.

3.3. The order of linear multistep methods

Given a linear multistep method characterized by the polynomials ρ and σ, define the operator L on the continuously differentiable functions $I \to \mathbb{R}^N$ by the formula

$$L(y)(x) = \sum_{i=0}^{k} \alpha_{k-i} y(x - ih) - h \sum_{i=0}^{k} \beta_{k-i} y'(x - ih). \tag{7}$$

A method is said to be of order p if $L(P) = 0$ for P any polynomial of degree not exceeding p.

To understand the significance of this definition, assume that y is continuously differentiable at least $p + 1$ times and expand the right-hand side of (7) in a Taylor series about x_n. We have

$$L(y)(x_n) = \sum_{i=0}^{p+1} C_i h^i y^{(i)}(x_n) + \mathrm{O}(h^{p+2}),$$

where

$$C_0 = \sum_{i=0}^{k} \alpha_i,$$

$$C_1 = -\sum_{i=1}^{k} i\alpha_{k-i} - \sum_{i=0}^{k} \beta_{k-i},$$

$$C_j = \frac{(-1)^j}{j!} \left(\sum_{i=1}^{k} i^j \alpha_{k-i} + j \sum_{i=1}^{k} i^{j-1} \beta_{k-i} \right), \quad j = 2, 3, \ldots, p + 1.$$

If y is replaced by a polynomial of degree p, then

$$L(P)(x_n) = \sum_{i=0}^{p} C_i h^i P^{(i)}(x_n)$$

and, for this to vanish for all such polynomials, it is necessary and sufficient that

$$C_0 = C_1 = C_2 = \cdots = C_p = 0.$$

We derive the two methods (5) and (6) using these expressions. The Adams–Bashforth method of order 3 requires $k = 3$ and assumes that $\alpha_3 = 1$, $\alpha_2 = -1$, $\alpha_1 = \alpha_0 = \beta_3 = 0$. We have

$$C_0 = \alpha_3 + \alpha_2 + \alpha_1 + \alpha_0 \quad = 0,$$

$$C_1 = -\alpha_2 - 2\alpha_1 - 3\alpha_0 - \beta_2 - \beta_1 - \beta_0 \quad = 1 - \beta_2 - \beta_1 - \beta_0,$$

$$C_2 = \tfrac{1}{2}(\alpha_2 + 4\alpha_1 + 9\alpha_0 + 2(\beta_2 + 2\beta_1 + 3\beta_0)) \quad = \beta_2 + 2\beta_1 + 3\beta_0 - \tfrac{1}{2},$$

$$C_3 = -\tfrac{1}{6}(\alpha_2 + 8\alpha_1 + 27\alpha_0 - 3(\beta_2 + 4\beta_1 + 9\beta_0)) \quad = \tfrac{1}{6} - \tfrac{1}{2}(\beta_2 + 4\beta_1 + 9\beta_0).$$

The solution of $C_1 = C_2 = C_3 = 0$ is $\beta_2 = \frac{23}{12}$, $\beta_1 = -\frac{4}{3}$, $\beta_0 = \frac{5}{12}$, with the first nonzero coefficient in the Taylor expansion of $L(y)(x_n)$ given by

$$C_4 = -\tfrac{1}{24}(1 - 4(\beta_2 + 8\beta_1 + 27\beta_0)) = \tfrac{3}{8}.$$

The value of this quantity is closely related to the "error constant" for the method which is actually given by $C_4/\rho'(1)$. Note that, in this case, and also for the Adams–Moulton method which we will discuss next, $\rho'(1) = 1$.

For the Adams–Moulton method of order 3, an additional nonzero parameter β_k is available and $k = 2$ is sufficient for this order. We find $C_0 = 0$, $C_1 = 1 - \beta_2 - \beta_1 - \beta_0$, $C_2 = \beta_1 + 2\beta_0 - \frac{1}{2}$ and $C_3 = \frac{1}{6} - \frac{1}{2}(\beta_1 + 4\beta_0)$ and $C_1 = C_2 = C_3 = 0$ implies $\beta_2 = \frac{5}{12}$, $\beta_1 = \frac{2}{3}$, $\beta_0 = -\frac{1}{12}$, with

$$C_4 = -\tfrac{1}{24}(1 - 4(\beta_1 + 8\beta_0)) = -\tfrac{1}{24}.$$

To investigate the order conditions further, it is convenient to consider the expression $(\rho(\exp(z)) - z\sigma(\exp(z)))\exp(-kz)$ which can be expanded by Taylor series

$$\begin{aligned}(\rho(\mathrm{e}^z) - z\sigma(\mathrm{e}^z))\,\mathrm{e}^{-kz} &= \alpha_k + \alpha_{k-1}\mathrm{e}^{-z} + \alpha_{k-2}\mathrm{e}^{-2z} + \cdots \\ &\quad + z(\beta_k + \beta_{k-1}\mathrm{e}^{-z} + \beta_{k-2}\mathrm{e}^{-2z} - \cdots) \\ &= C_0 + C_1 z + C_2 z^2 + \cdots \\ &= C_{p+1} z^{p+1} + \mathrm{O}(z^{p+2}),\end{aligned}$$

if the order is p. The number C_{p+1} does not vanish unless the order is actually higher than p. Hence

$$\rho(\mathrm{e}^z) - z\sigma(\mathrm{e}^z) = \mathrm{O}(z^{p+1}).$$

Because $\rho(1) = 0$ for a consistent method, we can divide by z and we find

$$\frac{\rho(\mathrm{e}^z)}{z} - \sigma(\mathrm{e}^z) = \mathrm{O}(z^p)$$

and substituting $\exp(z)$ by $1 + z$

$$\frac{\rho(1+z)/z}{\ln(1+z)/z} - \sigma(1+z) = \mathrm{O}(z^p), \tag{8}$$

where $\ln(1+z)/z$ is defined in a neighbourhood of 0 by the series

$$\frac{\ln(1+z)}{z} = 1 - \frac{z}{2} + \frac{z^2}{3} - \cdots$$

so that

$$\left(\frac{\ln(1+z)}{z}\right)^{-1} = 1 + \frac{z}{2} - \frac{z^2}{12} + \frac{z^3}{24} - \frac{19z^4}{720} + \frac{3z^5}{160} + \mathrm{O}(z^6).$$

Using this expression, (8) can be used to derive methods with specific choices of ρ. Rewriting in the form

$$\rho(1+z) = \sigma(1+z)\left(z - \frac{z^2}{2} + \frac{z^3}{3} - \cdots\right)$$

enables coefficients to be found for the backward difference and similar methods in which the form of σ is prescribed.

3.4. The Dahlquist barrier

Even though it is possible, in principle, for linear multistep methods to have order as high as $2k$, this does not yield stable methods if $k > 2$. This is a consequence of the so-called "Dahlquist barrier" [21], which states that

Theorem 1. *The order of a stable linear k-step method is bounded by*

$$p \leqslant \begin{cases} k+2, & k \text{ even}, \\ k+1, & k \text{ odd}. \end{cases}$$

Proof. We will give here a vastly abbreviated proof, along the same lines as originally given by Dahlquist. Let

$$r(z) = \rho\left(\frac{1+z}{1-z}\right)\left(\frac{1-z}{2}\right)^k,$$

$$s(z) = \sigma\left(\frac{1+z}{1-z}\right)\left(\frac{1-z}{2}\right)^k,$$

where we note that the order conditions can be rewritten in the form

$$\frac{r(z)/z}{\ln((1+z)/(1-z))/z} - s(z) = \mathrm{O}(z^p). \tag{9}$$

Let $r(z) = a_0 + a_1 z + a_2 z^2 + \cdots + a_k z^k$ and $s(z) = b_0 + b_1 z + b_2 z^2 + \cdots + b_k z^k$, where $a_0 = 0$ by the consistency condition. By the stability condition, $a_1 \neq 0$ and no two of the coefficients in r can have opposite signs. If

$$\frac{1}{\ln((1+z)/(1-z))/z} = c_0 + c_2 z^2 + c_4 z^4 + \cdots,$$

it can be shown that $c_2, c_4, \ldots$ are all negative [21,39]. If (9) is to hold for $p > k+1$, then the coefficient of z^{p+1} in

$$(c_0 + c_2 z^2 + c_4 z^4 + \cdots)(a_1 + a_2 z + \cdots + a_k z^{k-1}), \tag{10}$$

must vanish. If the order is $p > k+2$, then the coefficient of z^{k+2} in (10) must also vanish. The two coefficients are respectively

$$a_k c_2 + a_{k-2} c_4 + \cdots, \tag{11}$$

$$a_{k-1} c_4 + a_{k-3} c_6 + \cdots. \tag{12}$$

If k is odd, (11) cannot vanish because this would imply that

$$a_k = a_{k-2} = \cdots = a_1 = 0.$$

On the other hand, if k is even, then (12) cannot vanish because we would then have

$$a_{k-1} = a_{k-3} = \cdots = a_1 = 0.$$

4. The modern theory of Runge–Kutta methods

The meaning of order looks quite different and is relatively complicated for one-step methods, for the very good reason that the result computed in a step is built up from the derivatives evaluated sequentially from the stages values and, at least for the early stages, these have low accuracy. In contrast, the result computed in linear multistep methods makes use of derivatives evaluated from a number of step values, which themselves have been evaluated in previous steps and all share the same order.

The basic approach to the analysis of Runge–Kutta methods is to obtain the Taylor expansions for the exact and computed solutions at the end of a single step and to compare these series term by term. This idea dates back to Runge, Heun, Kutta and Nyström and we will give as an example the derivation of the conditions for order 3.

For the scalar differential equation

$$y'(x) = f(x, y(x)), \tag{13}$$

we calculate in turn

$$y'' = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} f, \tag{14}$$

$$y''' = \frac{\partial^2 f}{\partial x^2} + 2\frac{\partial^2 f}{\partial x \partial y} f + \frac{\partial^2 f}{\partial y^2} f^2 + \frac{\partial f}{\partial x}\frac{\partial f}{\partial y} + \left(\frac{\partial f}{\partial y}\right)^2 f, \tag{15}$$

where we have substituted $y' = f$ in the formula $y'' = (\partial f/\partial x) + (\partial f/\partial y) y'$ to obtain (14) and made similar substitutions in the derivation of (15). From these expressions we can write down the first few terms of the Taylor expansion $y(x_0 + h) = y(x_0) + h y'(x_0) + \frac{1}{2} h^2 y''(x_0) + \frac{1}{6} h^3 y'''(x_0) + \mathrm{O}(h^4)$.

Complicated though these expressions are, they are simple in comparison with the corresponding formulas for the fourth and higher derivatives. To obtain conditions for order 3 we also need the

Table 1
Details of Taylor expansions up to order 3

	Y_1	hF_1	Y_2	hF_2	Y_3	hF_3	y_1	$y(x_0+h)$
y	1	0	1	0	1	1	1	1
hf	0	1	a_{21}	1	$a_{31}+a_{32}$	1	$b_1+b_2+b_3$	1
$h^2\frac{\partial f}{\partial x}$	0	0	0	c_2	$a_{32}c_2$	c_3	$b_2c_2+b_3c_3$	$\frac{1}{2}$
$h^2\frac{\partial f}{\partial y}f$	0	0	0	a_{21}	$a_{32}a_{21}$	$a_{31}+a_{32}$	$b_2a_{21}+b_3(a_{31}+a_{32})$	$\frac{1}{2}$
$h^3\frac{\partial^2 f}{\partial x^2}$	0	0	0	$\frac{1}{2}c_2^2$	$\frac{1}{2}a_{32}c_2^2$	$\frac{1}{2}c_3^2$	$\frac{1}{2}(b_2c_2^2+b_3c_3^2)$	$\frac{1}{6}$
$h^3\frac{\partial^2 f}{\partial x\partial y}f$	0	0	0	c_2a_{21}	$a_{32}c_2a_{21}$	$c_3(a_{31}+a_{32})$	$b_2c_2a_{21}+b_3c_3(a_{31}+a_{32})$	$\frac{1}{3}$
$h^3\frac{\partial^2 f}{\partial y^2}f^2$	0	0	0	$\frac{1}{2}a_{21}^2$	$\frac{1}{2}a_{32}a_{21}^2$	$\frac{1}{2}(a_{31}+a_{32})^2$	$\frac{1}{2}(b_2a_{21}^2+b_3(a_{31}+a_{32})^2)$	$\frac{1}{6}$
$h^3\frac{\partial f}{\partial x}\frac{\partial f}{\partial y}$	0	0	0	0	0	$a_{32}c_2$	$b_3a_{32}c_2$	$\frac{1}{6}$
$h^3\left(\frac{\partial f}{\partial y}\right)^2 f$	0	0	0	0	0	$a_{32}a_{21}$	$b_3a_{32}a_{21}$	$\frac{1}{6}$

formulas for the first, second and third derivatives of the approximation computed by a Runge–Kutta method, which we will assume is explicit and has exactly 3 stages.

To simplify notation we will denote x, y, f and the various partial derivatives, as being evaluated at the initial point (x_0, y_0) in a step and we will then find Taylor expansions in turn for $Y_1, hF_1, \ldots, Y_2$, hF_2, Y_3, hF_3 and finally y_1. We will express the sequence of calculations in tabular form in Table 1, where the coefficients of y, hf, etc. are shown. In addition to the coefficients in the expansion of y_1, we append the corresponding coefficients for the exact solution at x_0+h.

By equating the last two columns of this table, we obtain conditions for order 3. These imply that

$$a_{21} = c_2, \tag{16}$$

$$a_{31} + a_{32} = c_3 \tag{17}$$

and that

$$b_1 + b_2 + b_3 = 1,$$

$$b_2c_2 + b_3c_3 = \tfrac{1}{2},$$

$$b_2c_2^2 + b_3c_3^3 = \tfrac{1}{3},$$

$$b_3a_{32}c_2 = \tfrac{1}{6}.$$

If $s = p$, which turns out to be possible for orders up to 4, conditions such as (16) and (17) always hold. Even for higher orders, where the argument is a little more complicated, there is never any reason for not assuming that

$$\sum_{j=1}^{s} a_{ij} = c_i, \quad i = 1, 2, \ldots, s, \tag{18}$$

where we adopt a convention that $a_{ij} = 0$ for $j \geqslant i$ in explicit methods. For the more general implicit methods, we will continue to assume (18).

There are three reasons for abandoning (1) as the standard problem and replacing it instead by (2), where the values of $y(x)$ are now in a finite-dimensional vector space rather than scalars. The first reason for the change to a high-dimensional *autonomous* problem is that there is no need to retain x as an argument of f in the vector case, because nonautonomous problems can always be transformed into equivalent autonomous problems by adding an additional component which always has a value exactly the same as x. A consideration of this formal re-formulation can be used to justify the assumption (18). The second reason is that the analysis is actually more straightforward in the autonomous vector case. Finally, it is found that the conditions for order as derived using the scalar first-order problem (13) are inadequate for specifying the order requirements for the general vector case. The two theories do not diverge until the fifth-order case is reached but after that the families of order conditions for the scalar and vector cases become increasingly different.

4.1. The order of Runge–Kutta methods

The analysis of order for the vector case that we present here is due to the present author [9] and is related to earlier work by Gill [33] and Merson [46]. Since it relates the various terms in the Taylor series expansion of both the exact solution and the approximation computed by a Runge–Kutta method, to the graphs known as "rooted trees" or arborescences, we briefly review rooted trees.

A rooted tree is simply a connected directed graph for which each vertex, except the root, has a single predecessor (or parent). The root has no predecessor. The order of a rooted tree t is the number of vertices. Denote this by $r(t)$. Clearly, the number of arcs in the rooted tree is $r(t) - 1$. Let a_n denote the number of distinct rooted trees with order n. Table 2 gives the first few values of a_n together with the sums $\sum_{i=1}^{n} a_i$.

The eight rooted trees for which $r(t) \leqslant 4$ are shown in Table 3, together with the values of $\sigma(t)$, the "symmetry" of t and $\gamma(t)$ the "density" of t. The quantity $\sigma(t)$ is the order of the group of permutations of the vertices which leave the structure unchanged, while $\gamma(t)$ is the product over all vertices of t of the total number of descendents (including the vertex itself) of this vertex. Also

Table 2
Numbers of trees and accumulated sums up to order 8

n	1	2	3	4	5	6	7	8
a_n	1	1	2	4	9	20	48	115
$\sum_{i=1}^{n} a_i$	1	2	4	8	17	37	85	200

Table 3
Various functions on trees

t	$\bullet$	$[\tau]$	$[\tau^2]$	$[[\tau]]$	$[\tau^3]$	$[\tau[\tau]]$	$[[\tau^2]]$	$[[[\tau]]]$
$r(t)$	1	2	3	3	4	4	4	4
$\sigma(t)$	1	1	2	1	6	1	2	1
$\gamma(t)$	1	2	3	6	4	8	12	24
$\alpha(t)$	1	1	1	1	1	3	1	1
$\beta(t)$	1	2	3	6	4	24	12	24

shown are the values of $\alpha(t)=r(t)!/\gamma(t)\sigma(t)$ and $\beta(t)=r(t)!/\sigma(t)$. The values of α and β have simple interpretations in terms of possible labellings of the vertices of a tree under various restrictions.

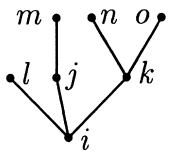
It can be shown that the Taylor expansion of the exact solution has the form

$$y(x_0+h)=y(x_0)+\sum_{t\in T}\frac{\alpha(t)h^{r(t)}}{r(t)!}F(t)(y_0) \tag{19}$$

and that the corresponding expansion for the solution computed using a Runge–Kutta method is

$$y(x_0)+\sum_{t\in T}\frac{\beta(t)\Phi(t)h^{r(t)}}{r(t)!}F(t)(y_0). \tag{20}$$

In each of these formulas, $F(t)$ is the "elementary differential" which we will define below and $\Phi(t)$ is the "elementary weight". The formula for $F(t)(y)$ is defined in terms of the differential equation and $\Phi(t)$ in terms of the Runge–Kutta method being used. Each of these quantities can be defined recursively but, for our present purposes, it will be enough to present one example, using a tree of order 7 and to list these quantities for all trees of order up to 4. In the special example, the tree t is shown with labels i, j, k, l, m, n, o attached to the vertices. The formula for $F(t)$, is given in terms of an expression for component number i, written as a superscript. The summation convention is assumed and f^i_{jkl} denotes the third partial derivative, $\partial^3 f^i/\partial y^j\partial y^k\partial y^l$, of f^i, evaluated at y, with similar meanings for f^j_m, f^k_{no}. The summations in the formula for $\Phi(t)$ are over all subscripts running from 1 to s. Note that the formula is simplified from $\Phi(t)=\sum b_i a_{il}a_{ij}a_{jm}a_{ik}a_{kn}a_{ko}$ by summing over l, m, n and o.

$t=$ 

$$F^i(t)=f^i_{jkl}f^l f^j_m f^m f^k_{no}f^n f^o,$$
$$\Phi(t)=\sum b_i c_i a_{ij}c_j a_{ik}c_k^2.$$

Because the elementary differentials are independent, in the sense that, given any set of n rooted trees, $t_1,t_2,\ldots,t_n$ and any sequence of real numbers $q_1,q_2,\ldots,q_n$, it is possible to find a function f such that for some specific value of y and some specific coordinate direction, say e_1^{T}, all the equations

$$e_1^{\mathrm{T}}F(t_i)(y)=q_i,\quad i=1,2,\ldots,n,$$

Table 4
Elementary differential and weights up to order 4

t	$\bullet$	$\vert$	$\vee$	$\vert$	$\vee$	$\vee$	$\vee$	$\vert$
$F(t)$	f^i	$f^i_j f^j$	$f^i_{jk} f^j f^k$	$f^i_j f^j_k f^k$	f^i_{jkl}	$f^i_{jk} f^j f^k_l f^l$	$f^i_j f^j_{kl} f^k f^l$	$f^i_j f^j_k f^k_l f^l$
$\Phi(t)$	$\sum b_i$	$\sum b_i c_i$	$b_i c_i^2$	$\sum b_i a_{ij} c_k$	$\sum b_i c_i^3$	$\sum b_i c_i a_{ij} c_j$	$\sum b_i a_{ij} c_j^2$	$\sum b_i a_{ij} a_{jk} c_k$

can be satisfied simultaneously, it is only possible that (19) and (20) agree to within $O(h^{p+1})$ if

$$\alpha(t) = \beta(t)\Phi(t)$$

for every tree t with no more than p vertices.

Inserting the formulas for α and β, we find that

$$\Phi(t) = \frac{1}{\gamma(t)} \tag{21}$$

as the condition corresponding to this tree (Table 4).

It is interesting that, for the single first-order scalar differential equation (13), the independence of the elementary differentials breaks down and it turns out to be possible to obtain methods that have some specified order in this case, but a lower order for the more general system of equations given by (2). This effect occurs for order 5 and higher orders.

Other interpretations of order are of course possible. An alternative derivation of the order conditions, due to Albrecht [1], is based on expressions arising from the Taylor series for

$$y(x_0) + h\sum_{j=1}^{s} a_{ij} y'(x_0 + hc_j) - y(x_0 + hc_i) = \varepsilon^{(2)}h^2 + \varepsilon^{(3)}h^3 + \cdots,$$

where

$$\varepsilon_i^{(2)} = \sum_j a_{ij}c_j - \frac{1}{2}c_j^2,$$

$$\varepsilon_i^{(3)} = \sum_j a_{ij}c_j^2 - \frac{1}{3}c_j^3,$$

$$\vdots \quad \vdots$$

For order 4 for example, it is found to be necessary that

$$\sum_i b_i\varepsilon_i^{(2)} = 0, \quad \sum_i b_i c_i \varepsilon_i^{(2)} = 0, \quad \sum_i b_i a_{ij}\varepsilon_j^{(2)} = 0, \quad \sum_i b_i \varepsilon_i^{(3)} = 0,$$

which, together with the quadrature conditions

$$\sum_i b_i c_i^{k-1} = \frac{1}{k}, \quad k = 1,2,3,4,$$

are equivalent to (21), up to order four. A third approach, due to Hairer and Wanner [36], is based on the use of B-series. This theory, used to study compositions of Runge–Kutta methods, is related to [13], and has applications also to more general problems and methods.

Table 5
Minimum s to obtain order p

p	1	2	3	4	5	6	7	8
s	1	2	3	4	6	7	9	11

4.2. Attainable order of Runge–Kutta methods

For explicit Runge–Kutta methods with s stages, there are $s(s+1)/2$ free parameters to choose. It is easy to show that an order p is possible only if $s \geqslant p$. Up to order 4, $s = p$ is actually possible. However, for $p > 4$, the relationship between the minimum s to obtain order p is very complicated but is partly given in Table 5. The results given for $p > 4$ were proved in [11,15].

For implicit Runge–Kutta methods, which we will discuss below, the relationship is much simpler. In fact, order p can be obtained with s stages if and only if $p \leqslant 2s$.

4.3. Implicit Runge–Kutta methods

One of the earliest references to implicitness, as applied to Runge–Kutta methods, was in the book by Kunz [44] where the method of Clippinger and Dimsdale was quoted. This method with tableau

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{5}{24} & \frac{1}{3} & -\frac{1}{24} \\ 1 & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \hline & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \end{array}$$

is the forerunner both of Lobatto methods and of block methods [59].

The method of Hammer and Hollingsworth [38] will be explored in some detail. It is the forerunner of Gauss and other important classes of methods. The coefficients for the method are

$$\begin{array}{c|cc} \frac{1}{2}-\frac{\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4}-\frac{\sqrt{3}}{6} \\ \frac{1}{2}+\frac{\sqrt{3}}{6} & \frac{1}{4}+\frac{\sqrt{3}}{6} & \frac{1}{4} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

This method has order 4. This is a little surprising because the eight conditions for this order have been seemingly satisfied using only the six free parameters in A and b^{T}. Although the order conditions are trivial to check, we will verify them below using an argument that illustrates what happens much more generally.

If the coefficient matrix A is allowed to be fully implicit, that is any element on or above the diagonal may have a non-zero value, then there are clearly more free parameters available to satisfy the order conditions. The advantages, in terms of order, are even greater than might be expected from a mere comparison of the number of conditions with the number of free parameters, because various simplifying assumptions are easier to satisfy. These simplifying assumptions bring about a drastic lowering of the number of further conditions required for order; furthermore they interact and reinforce each other.

The simplifying assumptions we will use, are denoted by $C(\eta)$, $D(\xi)$ and $E(\xi,\eta)$, where we have used the notation of [10]. In each case ξ and η are positive integers and the assumptions refer to some equalities involving the coefficients of a specific method. The specific meanings are

$$C(\eta)\colon \sum_{j=1}^{s} a_{ij}c_j^{l-1} = \frac{1}{k}c_i^l, \quad i = 1,2,\ldots,s,\ l = 1,2,\ldots,\eta,$$

$$D(\xi)\colon \sum_{i=1}^{s} b_i c_i^{k-1} a_{ij} = \frac{1}{k}b_j(1-c_j^k), \quad j = 1,2,\ldots,s,\ k = 1,2,\ldots,\xi,$$

$$E(\xi,\eta)\colon \sum_{i=1}^{s} b_i c_i^{k-1} a_{ij} c_j^{l-1} = \frac{1}{l(k+1)}, \quad k = 1,2,\ldots,\xi,\ l = 1,2,\ldots,\eta.$$

Let us consider the relationship between these assumptions in the case that $\eta = \xi = s$ and the further assumption that c and b^{T} are chosen, as in the two-stage method we are considering, so that the c_i are the zeros of the degree s Legendre polynomial, shifted to the interval $[0,1]$, and b_i are the corresponding Gaussian weights. These assumptions on c and b^{T} will guarantee that $\sum_i b_i c_i^{k-1} = 1/k$ for $k=1,2,\ldots,2s$. Under this condition, $E(s,s)$ follows from $D(s)$ and because the linear combinations used to verify this have coefficients in a nonsingular (Vandermonde) matrix, the argument can be reversed. Similarly, $C(s)$ is also equivalent to $E(s,s)$.

In what has come to be referred to as a Gauss method, the b^{T} and c vectors are chosen to satisfy the requirements of Gaussian quadrature and the elements in each row of A are chosen so that $C(s)$ is satisfied. It then follows that $D(s)$ also holds. The method formed in this way always has order $2s$ and we will verify this for $s = 2$ in Table 6. Where no reason is given, the result is because of Gaussian quadrature. In other cases the manipulations are based on $C(2)$ or $D(2)$ and make use of order conditions already verified earlier in Table 6. Gauss methods for arbitrary $s \geqslant 1$ were introduced in [10,17].

Methods also exist with order $2s-1$ based on Radau quadrature of type I ($c_1 = 0$) or type II ($c_s = 1$). The most important of these are the Radau IIA methods. Some variants of Lobatto methods ($c_1 = 0$ and $c_s = 1$) with order $2s-2$, were once considered attractive for practical computation but have been superseded by other implicit methods.

It is now generally believed that the proper role of implicit Runge–Kutta methods is in the solution of stiff problems (see Section 6). There is a conflict between the three aims of high accuracy, good stability, and low implementation cost. Gauss methods seem to be ideal from the stability and accuracy points of views but they are very expensive, because of the fully implicit structure of the coefficient matrix. The accuracy is not as good as might be expected from order considerations alone because of an "order reduction" phenomenon [55,29], but the cost alone is enough to make alternative methods more attractive.

4.4. DIRK and SIRK methods

An obvious alternative to fully implicit methods, is to insist that the coefficient matrix have a lower triangular structure, because in this case the stages can be evaluated sequentially and the cost of each is relatively low. It turns out to be an advantage to have the diagonal elements equal and this

Table 6
Verification of order conditions for 2 stage Gauss method

•	$b_1 + b_2 = 1$
I	$b_1c_1 + b_2c_2 = \frac{1}{2}$
V	$b_1c_1^2 + b_2c_2^2 = \frac{1}{3}$
⋮	$b_1(a_{11}c_1 + a_{12}c_2) + b_2c_2(a_{11}c_1 + a_{12}c_2)$ $= \frac{1}{2}(b_1c_1^2 + b_2c_2^2) = \frac{1}{6}$
W	$b_1c_1^3 + b_2c_2^3 = \frac{1}{4}$
V⋮	$b_1c_1(a_{11}c_1 + a_{12}c_2) + b_2c_2(a_{11}c_1 + a_{12}c_2)$ $= \frac{1}{2}(b_1c_1^3 + b_2c_2^3) = \frac{1}{8}$
Y	$(b_1a_{11} + b_2a_{21})c_1^2 + (b_1a_{12} + b_2a_{22})c_2^2$ $= b_1(1 - c_1)c_1^2 + b_2(1 - c_2)c_2^2 = \frac{1}{12}$
⋮	$\sum b_ia_{i1}(a_{11}c_1 + a_{12}c_2) + \sum b_ia_{i2}(a_{21}c_1 + a_{22}c_2)$ $= \frac{1}{2}(\sum b_ia_{i1}c_1^2 + \sum b_ia_{i2}c_2^2) = \frac{1}{24}$

additional requirement has little impact on the availability of methods of a required order with good stability. Methods of this type have been variously named "semi-implicit" [10], "semi-explicit" [51] and "diagonally implicit" or "DIRK" [2]. Although equal diagonals were originally built into the DIRK formulation, common usage today favours using this name more widely and using "SDIRK" (or "singly diagonally implicit") in the more restricted sense. Other key references concerning these methods are [3,18].

Singly implicit methods, *without* necessarily possessing the DIRK structure are those for which A has only a single eigenvalue $\sigma(A) = \{\gamma\}$ [7]. If the stage order is s, it turns out that the abscissae for the method satisfy $c_i = \gamma\xi_i$, where $\xi_1, \xi_2, \ldots, \xi_s$ are the zeros of the Laguerre polynomial L_s. The advantage of these methods is that for many large problems, the component of the computer cost devoted to linear algebra is little more than for a DIRK method. Various improvements to the design of SIRK methods have been proposed.

5. Nontraditional methods

While the traditional methods, linear multistep and Runge–Kutta, are widely used and are generally regarded as satisfactory for solving a wide variety of problems, many attempts have been made to extend the range of available methods. Some of these will be discussed in this section.

5.1. Taylor series methods

Because the Euler method is based on approximations to the Taylor expansion

$$y(x_0 + h) \approx y(x_0) + hy'(x_0),$$

it is natural to ask if it is possible to take this expansion further by evaluating $y''(x_0)$, and possibly higher derivatives, by algebraic means. Algorithms for carrying out this process can be constructed using a recursive evaluation scheme. We mention two important early papers which exploit this idea, using a classical view of computer arithmetic but combined with this non-classical method of solution [4,32].

A second interesting and important contribution to Taylor series that has a further distinctive feature is the work of Moore [48]. The distinctive feature is that the work is carried out in the context of interval arithmetic. This means that it becomes possible, not only to advance the solution step-by-step in a relatively efficient manner, but it also becomes possible, owing to the standard bounds on the truncation error of a Taylor expansion, to obtain *rigorous* error bounds. Thus, in principle, it became possible to obtain intervals in which each component of the solution is *certain* to lie for any particular value of x. The difficulty is, of course, that the lengths of these intervals can grow rapidly as x increases.

5.2. Hybrid methods

These methods are similar to linear multistep methods in predictor–corrector mode, but with one essential modification: an additional predictor is introduced at an *offstep* point. This means that the final (corrector) stage has an additional derivative approximation to work from. This greater generality allows the consequences of the Dahlquist barrier to be avoided and it is actually possible to obtain convergent k-step methods with order $2k+1$ up to $k=7$. Even higher orders are available if two or more offstep points are used. The three independent discoveries of this approach were reported in [34,30,12]. Although a flurry of activity by other authors followed, these methods have never been developed to the extent that they have been implemented in general purpose software.

5.3. Cyclic composite methods

It is remarkable that even though a number of individual linear multistep methods may be unstable, it is possible to use them cyclically to obtain a method which, overall, is stable. An example of a fifth-order method given in the key paper on this subject [26] is as follows:

$$\begin{aligned} y_n = &-\tfrac{8}{11}y_{n-1} + \tfrac{19}{11}y_{n-2} \\ &+h(\tfrac{10}{33}f(x_n, y_n) + \tfrac{19}{11}f(x_{n-1}, y_{n-1}) + \tfrac{8}{11}f(x_{n-2}, y_{n-2}) - \tfrac{1}{33}f(x_{n-3}, y_{n-3})), \end{aligned} \tag{22}$$

$$\begin{aligned} y_n = &\tfrac{449}{240}y_{n-1} + \tfrac{19}{30}y_{n-2} - \tfrac{361}{240}y_{n-3} + h(\tfrac{251}{720}f(x_n, y_n) + \tfrac{19}{30}f(x_{n-1}, y_{n-1}) - \tfrac{449}{240}f(x_{n-2}, y_{n-2}) \\ &- \tfrac{35}{72}f(x_{n-3}, y_{n-3})), \end{aligned} \tag{23}$$

$$\begin{aligned} y_n = &-\tfrac{8}{11}y_{n-1} + \tfrac{19}{11}y_{n-2} \\ &+h(\tfrac{10}{33}f(x_n, y_n) + \tfrac{19}{11}f(x_{n-1}, y_{n-1}) + \tfrac{8}{11}f(x_{n-2}, y_{n-2}) - \tfrac{1}{33}f(x_{n-3}, y_{n-3})). \end{aligned} \tag{24}$$

The method is used cyclically in the sense that in each of three steps, the first uses (22), the second uses (23) and the third uses (24) (which in this particular example happens to be the same as the first member of the cycle). Methods of this family have been studied by a number of authors and have also been discovered for use with stiff problems.

5.4. Rosenbrock methods

Rosenbrock in his 1963 paper [56], discusses the problem of evaluating the stages of a diagonally implicit Runge–Kutta methods. Normally this is carried out by an iteration process based on Newton's method. For each stage and each iteration, an evaluation of f is carried out together with a solution of a linear equation system with matrix of coefficients of the form $I - h\lambda J$, where J denotes an approximation to the Jacobian matrix. The question was then asked if improved performance can be obtained by an alternative procedure in which exactly the same amount of work is performed but *only once* per stage, with the proviso that J is exactly the Jacobian evaluated at y_{n-1}. Amongst the examples of this type of "Rosenbrock method" given in the original paper, the following is identified as having order 2 and possessing L-stability:

$$
\begin{aligned}
F_1 &= (I - h(1 - \tfrac{\sqrt{2}}{2})J)^{-1} f(y_{n-1}),\\
F_2 &= (I - h(1 - \tfrac{\sqrt{2}}{2})J)^{-1} f(y_{n-1} + h(\tfrac{1}{2}\sqrt{2} - \tfrac{1}{2})F_1),\\
y_n &= y_{n-1} + hF_2.
\end{aligned}
$$

Amongst the many further contributions to the study of Rosenbrock methods, and their generalizations, we refer to [16,52,43].

6. Methods for stiff problems

The paper by Curtiss and Hirschfelder [20] is usually acknowledged as introducing numerical analysts to the phenomenon of stiffness. Much has been written about what "stiffness" really means but the property is generally understood in terms of what goes wrong when numerical methods not designed for such problems are used to try to solve them. For example, classical explicit Runge–Kutta methods were not intended to solve stiff problems but, when one attempts to use them, there is a severe restriction on stepsize that must be imposed, apparently because of stability rather than accuracy requirements. It is easy to see how this can come about for linear problems of the form

$$y'(x) = My(x),$$

if the matrix M happens to have all its eigenvalues close to zero or else in the left half complex plane and with a large magnitude. Assuming for simplicity that M can be diagonalized and that the problem is solved in its transformed form, the accuracy is determined by the ability of the numerical method to solve problems of the form

$$y'(x) = \mu y(x),$$

where $|\mu|$ is small. However, the stability of the numerical approximation is limited by the fact that we are simultaneously trying to solve a problem of the form

$$y'(x) = \lambda y(x),$$

where $|\lambda|$ is large. In the exact solution, terms of the second type correspond to rapidly decaying transients, whereas in the computed solution they represent unstable parasitic solutions, unless h is so small that $h\lambda$ lies in what is known as the "stability region".

To find the stability region for a numerical method it is necessary to consider the behaviour of the numerical method with a problem of just this type. For classical methods the behaviour depends on the product of h and λ which we will write as z. For the classical fourth-order Runge–Kutta method, the numerical solution for this problem satisfies

$$y_n = R(z)y_{n-1}, \tag{25}$$

where

$$R(z) = 1 + z + \frac{z^2}{2} + \frac{z^3}{6} + \frac{z^4}{24}$$

and the stability region is the set of points in the complex plane for which $|R(z)| \leqslant 1$.

If $z = h\lambda$ is outside this set, as it might well be, then selecting h to make $h\mu$ of a reasonable size will not be satisfactory because the unstable behaviour of the component of the solution associated with λ will swamp the solution and destroy its accuracy.

To analyse this type of possible instability, Dahlquist [23] introduced the concept of A-stability. A numerical method is said to be "A-stable" if its stability region includes all of the left half-plane. Even though the definition was first framed in the context of linear multistep methods, it was soon applied to Runge–Kutta methods, for which it takes a particularly simple form. Write Y for the vector of stage values, then this vector and the output approximation are related by

$$Y = y_{n-1}\mathrm{e} + zAY, \quad y_n = y_{n-1} + zb^{\mathrm{T}}Y,$$

leading to (25) with the stability function given by

$$R(z) = 1 + zb^{\mathrm{T}}(I - zA)^{-1}\mathrm{e}.$$

For explicit methods the stability region is always a bounded set and these methods cannot be A-stable. On the other hand for an implicit method, $R(z)$ has the form $N(z)/D(z)$ where the polynomials N and D can have degrees as high as s. Methods of arbitrarily high orders can be A-stable.

For a general linear multistep method, defined by polynomials ρ and σ, the stability region is the set of points z for which the polynomial $\rho(w) - z\sigma(w)$, of degree k in w satisfies the root condition. It was shown by Dahlquist [23] that for these methods, A-stability cannot exist for orders greater than 2.

6.1. Order stars

Runge–Kutta methods of Gauss type have stability functions of the form

$$R(z) = \frac{N(z)}{N(-z)},$$

where the polynomial N has degree s. Furthermore, $R(z)=\exp(z)+\mathrm{O}(z^{2s+1})$. This is an example of a "Padé approximation" to the exponential function, in the sense that the order of approximation is exactly the sum of the numerator and denominator degrees. Not only are the diagonal members of the Padé table significant, but the same can be said of the first subdiagonal (with degree $s-1$ for the numerator and s for the denominator, because these correspond to the stability functions for the Radau IA methods, and for the practically important Radau IIA methods. The second subdiagonals are also important because they are the stability functions for the Lobatto IIIC methods. It is known that the Padé approximations to the exponential function, in each of these three diagonals, correspond to A-stable methods. It is also clear that the approximations above the main diagonal cannot share this property but what can be said about approximations *below* the second subdiagonal? Considerable evidence existed for the "Ehle conjecture" [27] which claimed that *none* of these stability functions can correspond to an A-stable method or, in Ehle's terminology, that they are not A-acceptable.

In 1978 a new method was discovered for settling this, and many related questions. This approach introduced "order stars" [60], based on relative stability regions. Rather than study the regions of the complex plane for which $|R(z)|<1$, the regions are studied for which $|\exp(-z)R(z)|<1$. Since A-stable methods are those for which the stability function has no poles in the left half-plane and has its magnitude bounded by 1 on the imaginary axis, changing from the stability function $R(z)$ to the relative stability function $\exp(-z)R(z)$ leaves this criterion unchanged, but introduces much more structure, because $\exp(-z)R(z)=1+Cz^{p+1}+\mathrm{O}(z^{p+2})$, when z is small.

Consider Fig. 4 taken from [60]. Shown in this figure are the order stars of four Padé approximation, with degrees j (denominator) and k (numerator). The shaded parts of the figures, known as the fingers and characterized by $|\exp(-z)R(z)|>1$ and the unshaded parts, the dual fingers characterized by $|\exp(-z)R(z)|<1$ meet at 0 in a pattern determined by the sign of the real part of Cz^{p+1}, for $|z|$ small. This means that there will be exactly $p+1$ fingers and the same number of dual fingers meeting at zero. Furthermore, the angles subtended by each finger is the same $\pi/(p+1)$. It can be shown that all the bounded fingers contain poles and the bounded dual fingers contain zeros. The two upper figures are for A-stable methods, in which all the poles are in the right half-plane and no finger crosses the imaginary axis. The two lower figures, for which $j-k>2$, cannot meet these requirements, because there are too many bounded fingers for it to be possible for them all to leave zero in the right-hand direction. Some of these fingers must leave zero in the left-hand direction and either reach poles in the left half-plane or cross the imaginary axis to reach poles in the right-hand half-plane. A rigorous form of this argument is used to prove the Ehle conjecture and to prove a number of other results concerning both one step and multistep methods.

A recent study of order stars, which reviews most of the work up to the present time, is available in the book by Iserles and Nørsett [42].

6.2. Nonlinear stability

To obtain a deeper understanding of the behaviour of stiff problems, and of the numerical methods used to solve them, Dahlquist in 1975 [24], studied nonlinear problems of the form

$$y'(x)=f(x,y(x)), \tag{26}$$

where f satisfies the dissipativity condition

$$\langle f(x,u)-f(x,v),u-v\rangle\leqslant 0 \tag{27}$$

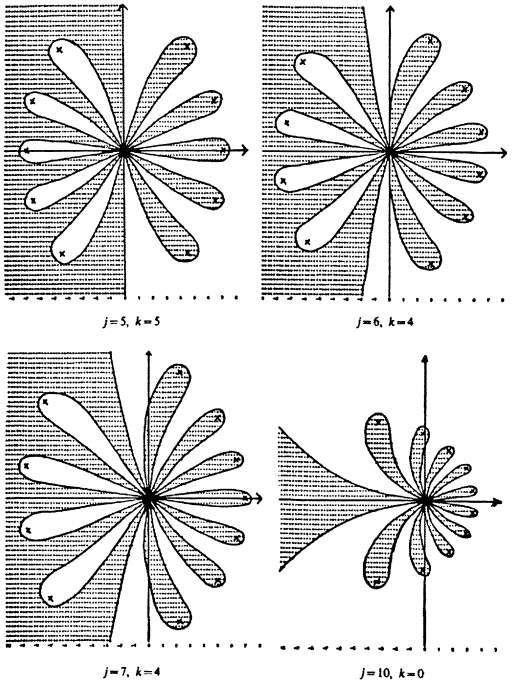


Figure 1. Order stars for Padé approximations.

Fig. 4. An extract from the order star paper.

and $\langle\cdot\rangle$ denotes an inner product, with $\|\cdot\|$ the corresponding norm. It is easy to see that two exact solutions to (26) possess the property that

$$\|y(x)-z(x)\| \leqslant \|y(x_0)-z(x_0)\|, \quad \text{for } x \geqslant x_0. \tag{28}$$

The aim is now to find conditions on a method such that a discrete analogue of (28) holds. It turns out to be more convenient to consider instead of a linear multistep method

$$\begin{aligned}&\alpha_k y_n + \alpha_{k-1} y_{n-1} + \alpha_{k-2} y_{n-2} + \cdots + \alpha_0 y_{n-k}\\ &\quad = h(\beta_k f(x_n, y_n) + \beta_{k-1} f(x_{n-1}, y_{n-1}) + \cdots + \beta_0 f(x_{n-k}, y_{n-k})),\end{aligned}$$

the corresponding "one-leg method"

$$\begin{aligned}&\alpha_k y_n + \alpha_{k-1} y_{n-1} + \alpha_{k-2} y_{n-2} + \cdots + \alpha_0 y_{n-k}\\ &\quad = h\left(\sum_{i=0}^{k} \beta_i\right) f\left(\frac{\beta_k}{\sum_{i=0}^{k}\beta_i} y_n + \frac{\beta_{k-1}}{\sum_{i=0}^{k}\beta_i} y_{n-1} + \cdots + \frac{\beta_0}{\sum_{i=0}^{k}\beta_i} y_{n-k}\right).\end{aligned}$$

For this type of method, Dahlquist considered contractivity in the sense that

$$\|Y_n\| \leqslant \|Y_{n-1}\|,$$

where

$$Y_n = \begin{bmatrix} y_n \\ y_{n-1} \\ \vdots \\ y_{n-k+1} \end{bmatrix}$$

and

$$\left\| \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_k \end{bmatrix} \right\| = \sum_{i,j=1}^{k} g_{ij} \langle \eta_i, \eta_j \rangle.$$

It is assumed that

$$G = \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1k} \\ g_{21} & g_{22} & \cdots & g_{2k} \\ \vdots & \vdots & & \vdots \\ g_{k1} & g_{k2} & \cdots & g_{kk} \end{bmatrix}$$

is a positive-definite matrix.

It is explained in Dahlquist's paper how results for one-leg methods can be interpreted as having a significance also for the corresponding linear multistep methods. He also found necessary and sufficient conditions for this property to hold. In a later paper [25], he showed that for one-leg methods, A-stability and G-stability are essentially equivalent properties.

The corresponding theory for Runge–Kutta methods [14,19,8], leads to a consideration of a matrix M with (i,j) element equal to $b_i a_{ij} + b_j a_{ji} - b_i b_j$. Assuming that this matrix is positive semi-definite, and the same is true for $\mathrm{diag}(b_1, b_2, \ldots, b_s)$, then a Runge–Kutta method applied to two distinct solutions of (26), satisfying (27), satisfies the contractivity property

$$\|y_n - z_n\| \leqslant \|y_{n-1} - z_{n-1}\|.$$

It is interesting that M has a more modern role in connection with symplectic methods for Hamiltonian problems.

A further development, initiated in the paper [29], is connected with the behaviour of truncation error for Runge–Kutta methods applied to stiff problems.

7. The beginnings of differential equation software

Programs to solve differential equations are as old as modern computers themselves. Today, a central aim in the design of differential equation software is the building of general purpose codes, specific only as regards stiffness versus nonstiffness, which adapt their behaviour to that of the computed solution dynamically. Variable stepsize is a characteristic feature of this software and usually variable order is used as well.

The most famous of the early codes in this tradition is the FORTRAN subroutine named by its designer, Gear, as "DIFSUB" [31]. Actually, this name was used generally by Gear for a range of

possible subroutines using a range of different methods. We will concentrate for the moment on the linear multistep version of DIFSUB.

As with all linear multistep implementations, the characteristic problems of starting values, local error estimation, change of stepsize and output interpolation have to be solved. A basic principle used in DIFSUB are the use of the Nordsieck representation of the data passed between steps, and this plays a crucial role in the solution of all these implementation questions, as well as the further problem of variable order.

The single paper of Nordsieck [50] explains how it is possible to rewrite a k-step Adams methods so that information on the values of $hf(x_{n-i}, y_{n-i})$ for $i=1,2,\ldots,k$ is organized as approximations to $hy'(x_{n-1}), \frac{1}{2!}h^2y''(x_{n-1}),\ldots,\frac{1}{k!}h^k y^{(k)}(x_{n-1})$. The rules for integration to the next step, are particularly simple in the case of the Adams–Bashforth method. The solution is first extrapolated using the approximation

$$\begin{bmatrix} y(x_n) \\ hy'(x_n) \\ \frac{1}{2!}h^2y''(x_n) \\ \vdots \\ \frac{1}{k!}h^k y^{(k)}(x_n) \end{bmatrix} \approx \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 2 & 3 & \cdots & k \\ 0 & 0 & 1 & 3 & \cdots & \binom{k}{2} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} y(x_{n-1}) \\ hy'(x_{n-1}) \\ \frac{1}{2!}h^2y''(x_{n-1}) \\ \vdots \\ \frac{1}{k!}h^k y^{(k)}(x_{n-1}) \end{bmatrix}$$

and a correction is then made to each component using a multiple of $hf(x_n, y_n) - hy'(x_n)$, so as to ensure that the method is equivalent to the Adams–Bashforth method. Adding an Adams–Moulton corrector to the scheme, is equivalent to adding further corrections.

Using the Nordsieck representation, it is possible to change stepsize cheaply, by simply rescaling the vector of derivative approximations. It is possible to estimate local truncation error using the appropriately transformed variant of the Milne device. It is also possible to measure the accuracy of lower and one higher-order alternative methods so that the appropriateness of order-changing can be assessed. Thus the ingredients are present to build a completely adaptive nonstiff solver. By adapting the backward difference methods to a similar form, it is possible to allow also for stiffness.

The DIFSUB program of Gear uses these techniques to obtain an efficient solver and many later programs are based on similar ideas. The first general purpose solver for differential–algebraic equations, the DASSL subroutine of Petzold [54], is also closely related to DIFSUB.

Early success was also achieved in the algorithm of Bulirsch and Stoer [6]. This used extrapolation in a similar manner to the quadrature algorithm of Romberg. The main difference between differential equations and quadrature is that significant efficiency gains are made by reusing some of the abscissae in a quadrature formula; this happens in the traditional version of the Romberg method because the mesh size is halved in each iteration. For differential equations there is no advantage in this, because reuse is not possible. Hence, in later developments of extrapolation methods, for both nonstiff and stiff problems, various sequences of stepsizes have been considered, where the aim is to balance computational cost against the quality of the convergence.

As many programs became available, using a variety of methods and variety of implementations of the same basic method, it became appropriate to consider what is really expected of these automatic solvers. Each person who develops this software needs to apply quality tests and to compare any new implementation against existing codes. In the interests of providing objective standards, a number of test sets have been developed. The earliest of these that has become widely adopted, and which in

fact serves as a de facto standard, is the DETEST problem set [41]. While this is appropriate for testing and comparing nonstiff solvers, a stiff counterpart, known as STIFF DETEST, [28] became available a little later.

8. Special problems

While the general aim of providing accurate and efficient general purpose numerical methods and algorithms has been a central activity in the 20th century, there has always been a realization that some problem types have such distinctive features that they will need their own special theory and techniques. Stiff problems were recognized approximately half way through the century as such a problem type and these have received considerable attention, especially in the last 30 years.

Another of the special problem types that has a separate claim for its own special methods, has been second-order differential equations and systems. These have a natural importance as arising in classical mechanical modelling and they were treated as a particular case by Nyström and others. While any problem of this type can be rewritten as a first-order system, it is found that treating them directly can lead to substantial gains in efficiency, especially if the second-order system takes the special form

$$y''(x) = f(x, y(x)),$$

where we note that $y'(x)$ does not occur as an argument of f. The Runge–Kutta approach to this type of problem was studied by Nyström [53] and has been of interest ever since. A modern theory of these methods is given in [37]. Linear multistep methods for this problem were studied as part of an investigation of a more general situation

$$y^{(n)}(x) = f(x, y(x)),$$

by Dahlquist [22].

It is interesting that one of the most active areas of modern research is closely related to this long-standing problem. Mechanical problems that can be expressed in a Hamiltonian formulation, rather than as a second-order system, can be studied in terms of the preservation of qualitative properties. It is found that the symplectic property can be preserved by the use of specially designed Runge–Kutta methods. The burgeoning subject of geometric integration, started from the study of Hamiltonian systems by Feng Kang, J.M. Sanz-Serna and others, and is now a central activity as the century closes. Although it is too early to view geometric integration from a historical perspective, it is at least possible to refer to a recent review of this subject [58].

There are several other evolutionary problems that can be solved by methods closely related to ordinary differential equation methods. Delay differential equations, and other types of functional differential equations can be solved using a combination of a differential equation solver, an interpolator and a means of handling discontinuities.

We have already noted that algebraic differential equations, especially those of low index, can be effectively solved using linear multistep methods. Implicit Runge–Kutta methods also have an important role in the numerical treatment of differential–algebraic equations. The theory of order of these methods can be extended to allow for the inclusion of algebraic constraints in the formulation, using generalizations of rooted trees [35].

References

[1] P. Albrecht, A new theoretical approach to Runge–Kutta methods, SIAM J. Numer. Anal. 24 (1987) 391–406.
[2] R. Alexander, Diagonally implicit Runge–Kutta methods for stiff ODEs, SIAM J. Numer. Anal. 14 (1977) 1006–1021.
[3] R. Alt, Deux théorèms sur la A-stabilité des schémas de Runge–Kutta simplement implicites, Rev. Francais d'Automat. Recherche Opérationelle Sér. R-3 6 (1972) 99–104.
[4] D. Barton, I.M. Willers, R.V.M. Zahar, The automatic solution of ordinary differential equations by the method of Taylor series, Comput. J. 14 (1971) 243–248.
[5] F. Bashforth, J.C. Adams, An attempt to test the theories of capillary action by comparing the theoretical and measured forms of drops of fluid, with an explanation of the method of integration employed in constructing the tables which give the theoretical forms of such drops, Cambridge University Press, Cambridge, 1883.
[6] R. Bulirsch, J. Stoer, Numerical treatment of ordinary differential equations by extrapolation methods, Numer. Math. 8 (1966) 1–13.
[7] K. Burrage, A special family of Runge–Kutta methods for solving stiff differential equations, BIT 18 (1978) 22–41.
[8] K. Burrage, J.C. Butcher, Stability criteria for implicit Runge–Kutta methods, SIAM J. Numer. Anal. 16 (1979) 46–57.
[9] J.C. Butcher, Coefficients for the study of Runge–Kutta integration processes, J. Austral. Math. Soc. 3 (1963) 185–201.
[10] J.C. Butcher, Implicit Runge–Kutta processes, Math. Comp. 18 (1964) 50–64.
[11] J.C. Butcher, On the attainable order of Runge–Kutta methods, Math. Comp. 19 (1965) 408–417.
[12] J.C. Butcher, A modified multistep method for the numerical integration of ordinary differential equations, J. Assoc. Comput. Mach. 12 (1965) 124–135.
[13] J.C. Butcher, An algebraic theory of integration methods, Math. Comp. 26 (1972) 79–106.
[14] J.C. Butcher, The non-existence of ten stage eighth order explicit Runge–Kutta methods, BIT 25 (1985) 521–540.
[15] J.C. Butcher, A stability property of implicit Runge–Kutta methods, BIT 15 (1975) 358–361.
[16] D.A. Calahan, A stable, accurate method of numerical integration for nonlinear systems, Proc. IEEE 56 (1968) 744.
[17] F. Ceschino, J. Kuntzmann, Problèmes Différentiels de Conditions Initiales, Dunod, Paris, 1963.
[18] M. Crouzeix, Sur les méthodes de Runge–Kutta pour l'approximation des problèmes d'évolution, Lecture Notes in Economics and Mathematical Systems, Vol. 134, Springer, Berlin, 1974, pp. 206–223.
[19] M. Crouzeix, Sur la B-stabilité des méthodes de Runge–Kutta, Numer. Math. 32 (1979) 75–82.
[20] C.F. Curtiss, J.O. Hirschfelder, Integration of stiff equations, Proc. Nat. Acad. Sci. 38 (1952) 235–243.
[21] G. Dahlquist, Convergence and stability in the numerical integration of ordinary differential equations, Math. Scand. 4 (1956) 33–53.
[22] G. Dahlquist, Stability and error bounds in the numerical integration of ordinary differential equations, Kungl. Tekn. Högsk. Handl. Stockholm 130 (1959) 1–87.
[23] G. Dahlquist, A special stability property for linear multistep methods, BIT 3 (1963) 27–43.
[24] G. Dahlquist, Error Analysis for a Class of Methods for Stiff Non-Linear Initial Value Problems, Lecture Notes in Mathematics, Vol. 506, Springer, Berlin, 1976, pp. 60–72.
[25] G. Dahlquist, G-stability is equivalent to A-stability, BIT 18 (1978) 384–401.
[26] J. Donelson, E. Hansen, Cyclic composite multistep predictor-corrector methods, SIAM J. Numer. Anal. 8 (1971) 137–157.
[27] B.L. Ehle, A-stable methods and Padé approximations to the exponential, SIAM J. Numer. Anal. 4 (1973) 671–680.
[28] W.H. Enright, T.E. Hull, B. Lindberg, Comparing numerical methods for stiff systems of ODEs, BIT 15 (1975) 10–48.
[29] R. Frank, J. Schneid, C.W. Ueberhuber, The concept of B-convergence, SIAM J. Numer. Anal. 18 (1981) 753–780.
[30] C.W. Gear, Hybrid methods for initial value problems in ordinary differential equations, SIAM J. Numer. Anal. 2 (1965) 69–86.
[31] C.W. Gear, Algorithm 407, DIFSUB for solution of ordinary differential equations, Comm. ACM 14 (1971) 447–451.
[32] A. Gibbons, A program for the automatic integration of differential equations using the method of Taylor series, Comput. J. 3 (1960) 108–111.

[33] S. Gill, A process for the step-by-step integration of differential equations in an automatic digital computing machine, Proc. Cambridge Philos. Soc. 47 (1951) 96–108.
[34] W.W. Gragg, H.J. Stetter, Generalized multistep predictor-corrector methods, J. Assoc. Comput. Mach. 11 (1964) 188–209.
[35] E. Hairer, C. Lubich, M. Roche, in: The Numerical Solution of Differential–Algebraic Systems by Runge–Kutta Methods, Lecture Notes in Mathematics, Vol. 1409, Springer, Berlin, 1989.
[36] E. Hairer, G. Wanner, On the Butcher group and general multi-value methods, Computing 13 (1974) 1–15.
[37] E. Hairer, G. Wanner, A theory for Nyström methods, Numer. Math. 25 (1976) 383–400.
[38] P.C. Hammer, J.W. Hollingsworth, Trapezoidal methods of approximating solutions of differential equations, MTAC 9 (1955) 269–272.
[39] P. Henrici, Discrete Variable Methods in Ordinary Differential Equations, Wiley, New York, 1962.
[40] K. Heun, Neue Methoden zur approximativen Integration der Differentialgleichungen einer unabhängigen Veränderlichen, Z. Math. Phys. 45 (1900) 23–38.
[41] T.E. Hull, W.H. Enright, B.M. Fellen, A.E. Sedgwick, Comparing numerical methods for ordinary differential equations, SIAM J. Numer. Anal. 9 (1972) 603–637.
[42] A. Iserles, S.P. Nørsett, Order Stars, Chapman & Hall, London, 1991.
[43] P. Kaps, G. Wanner, A study of Rosenbrock type methods of high order, Numer. Math. 38 (1981) 279–298.
[44] K.S. Kunz, Numerical solution of ordinary differential equations: methods of starting the solution, in: Numerical Analysis, McGraw-Hill, New York, 1957.
[45] W. Kutta, Beitrag zur näherungsweisen Integration totaler Differentialgleichungen, Z. Math. Phys. 46 (1901) 435–453.
[46] R.H. Merson, An operational method for the study of integration processes, Proceedings of the Symposium on Data Processing, Weapons Research Establishment, Salisbury, South Australia, 1957.
[47] W.E. Milne, A note on the numerical integration of differential equations, J. Res. Nat. Bur. Standards 43 (1949) 537–542.
[48] R.E. Moore, The automatic anlyasis and control of error in digital computation based on the use of interval numbers, Error in Digital Computation, Vol. 1, Wiley, New York, 1964, pp. 61–130.
[49] F.R. Moulton, New Methods in Exterior Balistics, University of Chicago, Chicago, 1926.
[50] A. Nordsieck, On numerical integration of ordinary differential equations, Math. Comp. 16 (1962) 22–49.
[51] S.P. Nørsett, One-step methods of Hermite type for numerical integration of stiff systems, BIT 14 (1974) 63–77.
[52] S.P. Nørsett, A. Wolfbrandt, Order conditions for Rosenbrock type methods, Numer. Math. 32 (1979) 1–15.
[53] E.J. Nyström, Über die numerische Integration von Differentialgleichungen, Acta Soc. Sci. Fennicae 50 (13) (1925).
[54] L.R. Petzold, A description of DASSL: a differential/algebraic system solver, in: Scientific Computing, Montreal, Quebec, North-Holland, Amsterdam, 1983, 65–68.
[55] A. Prothero, A. Robinson, On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations, Math Comp. 28 (1974) 145–162.
[56] H.H. Rosenbrock, Some general implicit processes for the numerical solution of differential equations, Comput. J. 5 (1963) 329–330.
[57] C. Runge, Über die numerische Auflösung von Differentialgleichungen, Math. Ann. 46 (1895) 167–178.
[58] J.M. Sanz-Serna, Geometric integration, The State of the Art in Numerical Analysis, Oxford University Press, Oxford, 1997, pp. 121–143.
[59] L.F. Shampine, H.A. Watts, Block implicit one-step methods, Math. Comp. 23 (1969) 731–740.
[60] G. Wanner, E. Hairer, S.P. Nørsett, Order stars and stability theorems, BIT 18 (1978) 475–489.