ELSEVIER

Journal of Computational and Applied Mathematics 111 (1999) 93–111

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.nl/locate/cam

Stiff differential equations solved by Radau methods

Ernst Hairer [*,1], Gerhard Wanner [2]

Dept. de mathématiques, Université de Genève, CH-1211 Genève 24, Switzerland

Received 28 April 1998; received in revised form 26 December 1998

Abstract

Radau IIA methods are successful algorithms for the numerical solution of stiff differential equations. This article describes RADAU, a new implementation of these methods with a variable order strategy. The paper starts with a survey on the historical development of the methods and the discoveries of their theoretical properties. Numerical experiments illustrate the behaviour of the code. © 1999 Elsevier Science B.V. All rights reserved.

MSC: 65L05

Keywords: Stiff differential equations; Implicit Runge–Kutta methods; Radau IIA methods; Historical notes; Order stars; Order selection strategy

The methods described in this paper emerged around 1969 as the fruit of two independent developments, on the one side the study of stiff differential equations principally in the light of multistep methods, and on the other side the theory of implicit Runge–Kutta methods. This is outlined in the first two sections of this paper. Sections 3 and 4 collect properties of the Radau IIA methods and the last two sections are devoted to their implementation and to a new order selection strategy for implicit Runge–Kutta methods. Several numerical experiments are presented.

1. Stiff equations and stability analysis

Stiff problems are characterized by the fact that the numerical solution of *slow* smooth movements is considerably perturbed by nearby *rapid* solutions. A typical example is the equation

$$y' = -50(y - \cos x). \tag{1}$$

* Corresponding author. Tel.: +41-22-702-6953; fax: +41-22-300-2064.

E-mail addresses: ernst.hairer@math.unige.ch (E. Hairer), gerhard.wanner@math.unige.ch (G. Wanner)

[1] URL: http://www.unige.ch/math/folks/hairer/

[2] http://www.unige.ch/math/folks/wanner/

0377-0427/99/$ - see front matter © 1999 Elsevier Science B.V. All rights reserved.
PII: S0377-0427(99)00134-X

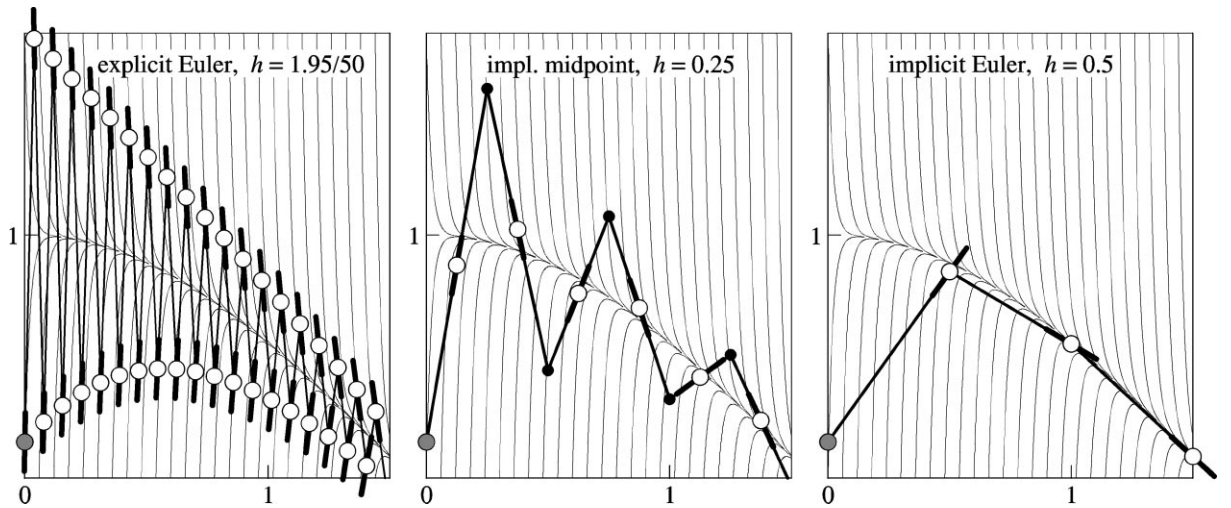

Fig. 1. Solution curves of (1) with explicit Euler, implicit midpoint rule, and implicit Euler solution.

Its solution curves are shown in Fig. 1. We see that the 'smooth' solution close to $y \approx \cos x$ is reached by all other solutions after a rapid 'transient phase'. The three figures present, in addition, for the initial value $y_0 = 0.15$, the numerical solutions for the explicit Euler method (left), the implicit midpoint rule (middle, where else?), and the implicit Euler method (right). The chosen step size for the explicit Euler method is tightly below the stability boundary. With larger step sizes, this method would overshoot the solution and produce serious numerical instability. The two other methods are stable for all h, but do not possess the same smoothing property.

1.1. First stiff problems

Stiff differential equations appeared half a century ago scattered here and there in the literature, and some ten years later one could say, in the words of G. Dahlquist, that 'around 1960 ... everyone became aware that the world was full of stiff problems' [28, p. 2].

The famous paper by Crank and Nicolson [10] treated a heat equation problem with nonlinear internal heat generation. For a numerical treatment, this problem is then reduced to a set of ordinary differential equations by 'replacing the space derivative'. This procedure, attributed by Crank and Nicolson to D.R. Hartree and named 'method II', is today known as 'method of lines'. If we omit the nonlinear term, the equation is $\partial\theta/\partial t = \partial^2\theta/\partial x^2$ and becomes, after partial discretization, the system

$$\theta_i' = \frac{1}{\Delta x^2}(\theta_{i+1} - 2\theta_i + \theta_{i-1})$$

($\theta_0 = \theta_{n+1} = 0, (n+1)\Delta x = 1$). It can be treated by the trapezoidal rule (as did Crank and Nicolson), or with the explicit or implicit Euler method. The results are shown in Fig. 2 and show precisely the same phenomena as for problem (1).

The first appearence of the term 'stiff' is in the paper by Curtiss and Hirschfelder [12] on problems in chemical kinetics. Without giving a precise definition, they call a differential equation stiff, if the

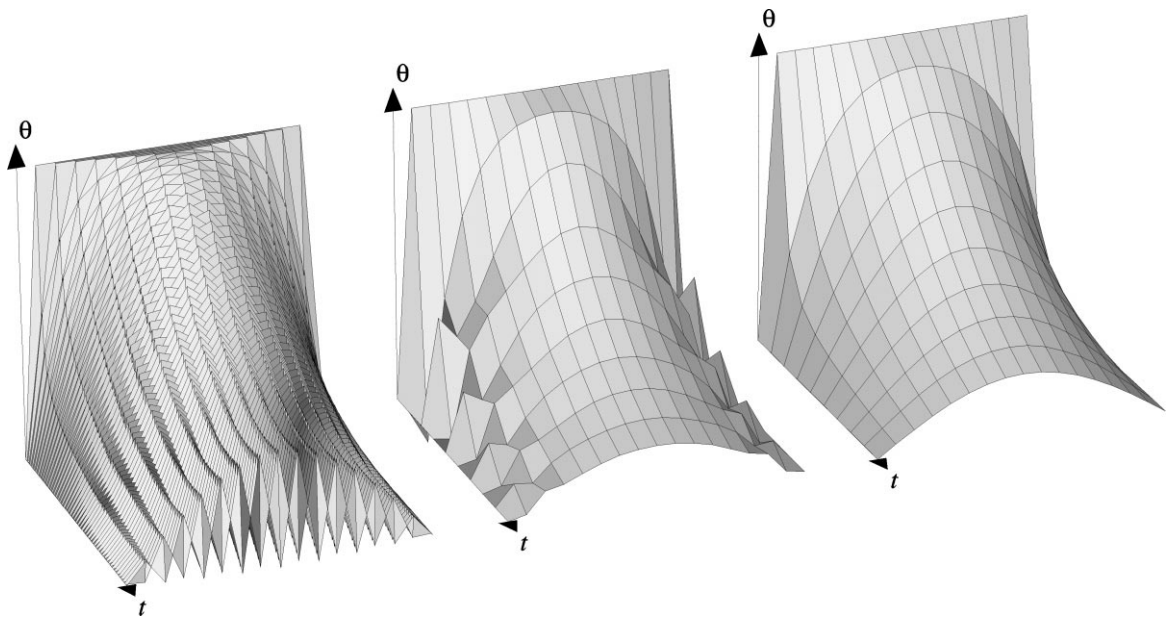

Fig. 2. Explicit Euler (left), trapezoidal rule (middle), implicit Euler (right).

implicit Euler method performs much better than the explicit Euler method. They discuss the concept of stiffness by considering the equation

$$\frac{\mathrm{d}y}{\mathrm{d}x} = [y - G(x)]/a(x, y).$$

"If Δx is the desired resolution of x or the interval which will be used in the numerical integration, the equation is 'stiff' if

$$\left|\frac{a(x, y)}{\Delta x}\right| \ll 1$$

and $G(x)$ is well behaved". Eq. (1) above is just a special case of such an equation. The article by Curtiss and Hirschfelder is also famous because it introduces the backward differentiation formulas (BDF). Today's readers of this classical paper are often surprised by the fact that Curtiss and Hirschfelder were thinking of equations with *positive* $a(x, y)$, thus were seeking stable numerical solutions of *unstable* problems.

Another early contribution to stiff differential equations is the article by Fox and Goodwin [21]. Hidden as the last section in a paper on various methods for ODEs, the authors consider the problem

$$y' = -10y + 6z, \quad z' = 13.5y - 10z, \tag{5.1}$$

with exact solution $y(x) = 2\mathrm{e}(\mathrm{e}^{-x} + \mathrm{e}^{-19x})/3$, $z(x) = \mathrm{e}(\mathrm{e}^{-x} - \mathrm{e}^{-19x})$, in order to explain phenomena of 'building-up errors': 'For values of x greater than unity the second exponential term is completely negligible, and it would be expected that the equation (5.1) could be integrated with confidence at a fairly large interval, say $h = 0.2$'. Fig. 3 presents some solutions of this equation together with a numerical solution computed by their 'Method II', which is the trapezoidal rule. We again observe precisely the same phenomena as before.

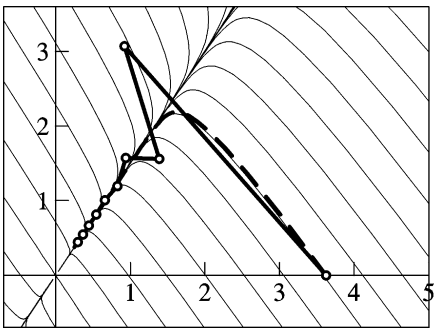

Fig. 3. Example of Fox and Goodwin [21], trapezoidal rule, $h = 0.2$.

1.2. Linear stability analysis

The explanation of the behaviours of the errors observed above in the neighbourhood of a smooth solution of $y'(x) = f(x, y(x))$ is done by linearization and leads to the so-called variational equation

$$y' = Jy \quad \text{where} \quad J \approx \partial f/\partial y \tag{2}$$

[40,13]. Other authors [10,21] were analysing linear equations anyway, or Loud [34] started from the beginning with a linear constant-coefficient case by saying that 'if a numerical method is to be of value in solving general differential equations, it should be extremely reliable with the simplest types ....' In higher dimensions, this equation is then diagonalized and leads to

$$y' = \lambda y, \quad \text{where} \quad \lambda \text{ represents eigenvalues of } J. \tag{3}$$

This equation is called the 'Dahlquist test equation', since Dahlquist stressed its importance in his famous paper [14]. Crank and Nicolson had analysed the errors in the *Fourier modes* as 'proposed to the authors by Prof. D.R. Hartree, following a suggestion by Prof. J. von Neumann'. This is equivalent to the foregoing diagonalization, since for the heat equation the Fourier base functions $\sin k\pi x$ coincide with the eigenvectors of J.

The above mentioned numerical methods, when applied to (3), lead to the following amplifications of the errors

$$y_{m+1} = R(z)y_m \quad \text{where} \quad z = h\lambda \tag{4}$$

and the so-called 'stability function' $R(z)$ is

$$R(z) = 1 + z \text{ (explicit Euler method)},$$

$$R(z) = \tfrac{1+z/2}{1-z/2} \text{ (trapezoidal and implicit midpoint rule)},$$

$$R(z) = \tfrac{1}{1-z} \text{ (implicit Euler method)}. \tag{5}$$

Instability appears if for an eigenvalue λ the modulus $|R(z)| > 1$. This happens for the explicit Euler method if, in the case of real eigenvalues, $z = h\lambda < -2$. The other two methods do not have such a restriction for negative λ. Their different smoothing properties are explained by the fact that $\lim_{z\to\infty} R(z) = -1$ for the trapezoidal rule, and $\lim_{z\to\infty} R(z) = 0$ for the implicit Euler method.

$$\mathrm{a} = -\mathrm{I}, \quad \frac{\mathrm{I} \pm \sqrt{6}}{5}, \quad \mathrm{A} = \frac{2}{9}, \quad \frac{\mathrm{I6} \mp \sqrt{6}}{\mathrm{I8}},$$

Fig. 4. Facsimile of Radau's formula of order 5 [38, p. 303].

1.3. A-stability

We quote from [14]: 'A k-step method is called A-stable, if all its solutions tend to zero, as $n \to \infty$, when the method is applied with fixed positive h to any differential equation of the form

$$\mathrm{d}x/\mathrm{d}t = qx,$$

where q is a complex constant with negative real part'. This famous definition can also be applied to one-step methods. The method is *A-stable* if the *stability domain*

$$S := \{z; |R(z)| \leqslant 1\} \tag{6}$$

covers the entire left half plane $\mathscr{C}^-$.

Immediately after this definition Dahlquist writes 'In most applications A-stability is not a necessary property. For certain classes of differential equations, however, it would be desirable to have an A-stable method …' and mentions applications in control engineering and chemical engineering. Dahlquist then proves his famous order barrier ($p \leqslant 2$) for A-stable multistep methods and discusses the stability for nonlinear problems. Mainly as a consequence of this severe order barrier, the search for A-stable, high-order methods attracted many numerical analysts during many years.

2. Early implicit Runge–Kutta methods

2.1. Radau quadrature

Rodolphe Radau[3] published in 1880 an extensive memoir [38] on quadrature formulas, with main emphasis, naturally, on Gauss, Lobatto, and Chebyshev methods. What we now call 'Radau formulas', formulas of maximal order with *one* end point as a prescribed node, occur very briefly and incidentally, merely for the sake of completeness. See in Fig. 4 Radau's publication of 'his' formula of order 5 for the interval $[-1, 1]$ with prescribed node -1. The Radau methods which will be most interesting *to us* are those with a fixed *right* endpoint. In today's notation, the formulas for the interval $[0, 1]$ with s stages have nodes c_i $(i = 1, \ldots, s)$, which are zeros of

$$\frac{\mathrm{d}^{s-1}}{\mathrm{d}x^{s-1}}(x^{s-1}(x-1)^s), \tag{7}$$

and the weights b_i are determined by the quadrature conditions

$$\sum_{i=1}^{s} b_i c_i^{q-1} = \frac{1}{q} \quad \text{for } q = 1, \ldots, s. \tag{8}$$

[3] Born 1835 in Prussia, studied Astronomy in Königsberg, moved 1858 to Paris, was a highly educated man (languages, music) and became a very fertile and successful author in Astronomy, Physics, Geodesy, Meteorology, and Applied Mathematics, died in 1911.

For more details about the old literature on Gaussian quadrature see the contribution by Runge and Willers in the *Enzyklopädie der Math. Wiss.*, Bd. 2, Teil 3, erste Hälfte, pp. 49ff. There, it can be seen that not only French (Radau), but also German (Gauss, Jacobi, Christoffel, Grunert), Dutch (Lobatto) and Russian (A. Markov) mathematicians have their heroes in this subject.

2.2. Runge–Kutta methods

The long story of the extension of quadrature formulas to methods which solve systems of ordinary differential equations

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

is, for example, outlined in [9]. This results in the formulas

$$Y_i = y_0 + h \sum_{j=1}^{s} a_{ij} f(x_0 + c_j h, Y_j), \quad i = 1, \ldots, s, \tag{9}$$

$$y_1 = y_0 + h \sum_{j=1}^{s} b_j f(x_0 + c_j h, Y_j). \tag{10}$$

Whenever there are nonzero coefficients a_{ij} with $i \leqslant j$, the method is called *implicit* and relation (9) constitutes a nonlinear system of equations for the unknowns $Y_1, \ldots, Y_s$. It is interesting to notice, that Butcher's important publication on implicit Runge–Kutta methods [5] was not allowed to be published in a 'computational' journal, unless an appendix on the solution of these implicit equations with fixed-point iterations was added. It turned out later that, in the case of stiff differential equations, Newton-type iterations are necessary (Liniger and Willoughby [33], see Section 5 below).

The a priori unknown coefficients a_{ij} are determined by the requirement that the expansion in powers of h of the numerical solution coincides with that of the true solution up to and including a certain order p. This requirement turns out to be a very complicated set of algebraic equations. The construction of higher order Runge–Kutta methods became only accessible after the discovery (in particular cases by A. Huťa [31], and in full clarity by J. Butcher [5]) of the so-called 'simplifying assumptions'

$$C(\eta): \quad \sum_{j=1}^{s} a_{ij} c_j^{q-1} = \frac{c_i^q}{q}, \quad i = 1, \ldots, s, \ \ q = 1, \ldots, \eta; \tag{11}$$

$$D(\zeta): \quad \sum_{i=1}^{s} b_i c_i^{q-1} a_{ij} = \frac{b_j}{q}(1 - c_j^q), \quad j = 1, \ldots, s, \ \ q = 1, \ldots, \zeta. \tag{12}$$

With the help of these conditions, Butcher was able to construct implicit Runge–Kutta methods of order $p = 2s$ of *arbitrarily* high order (Gauss methods). They are generalizations of the implicit midpoint rule.

2.3. Butcher's Radau methods

Soon after this famous paper, Butcher [6] published a paper on implicit Runge–Kutta methods based on Radau quadrature formulas. These quadratures made it possible to derive Runge–Kutta

processes that were not as 'terribly' implicit as the Gauss processes, because the fixed left or right node as endpoint allowed the first or last stage to be explicit. Butcher named his methods which correspond to the fixed *right* endpoint, with an explicit last stage, as 'II-processes'.

3. Radau IIA methods for stiff problems

3.1. Stability analysis for Runge–Kutta methods

Applying a Runge–Kutta method to Dahlquist's test equation $y' = \lambda y$ gives a numerical approximation $y_1 = R(\lambda h)y_0$, where $R(z)$ is a polynomial in the case of explicit one-step methods, and a rational function in general. It is called stability function of the method. For the Runge–Kutta method (9)–(10) one obtains

$$R(z) = 1 + zb^{\mathrm{T}}(I - zA)^{-1}\mathbb{1}, \tag{13}$$

where $b^{\mathrm{T}} = (b_1, \ldots, b_s)$, $A = (a_{ij})_{i,j=1}^{s}$, $\mathbb{1} = (1, \ldots, 1)^{\mathrm{T}}$.

The stability functions for the above mentioned Gauss methods were computed by Ehle [19], who obtained the *diagonal* Padé approximations R_{ss} to the exponential function e^z. The general formula is $R_{kj} = P_{kj}/Q_{kj}$ where

$$P_{kj}(z) = 1 + \frac{k}{j+k}z + \frac{k(k-1)}{(j+k)(j+k-1)} \cdot \frac{z^2}{2!} + \cdots + \frac{k(k-1)\ldots 1}{(j+k)\ldots(j+1)} \cdot \frac{z^k}{k!} \tag{14}$$

and $Q_{kj}(z) = P_{jk}(-z)$. The diagonal approximations were known to be A-stable (Birkhoff and Varga [3]), but possess a similar bad damping property as the implicit midpoint rule in Figs. 1–3 above.

Still more disappointing were the resulting stability functions for Butcher's Radau II methods, which led to Padé approximations *above* the diagonal ($j = k - 1$). These tend to ∞ for $z \to \infty$ and therefore only have a bounded stability domain.

Ehle [19] (and independently Axelsson [1]) thus undertook the search for other extensions of the Radau formulas, which he named 'Radau IIA' methods, by working with Eqs. (11) and (12) in such a way that the stability functions appear to be *below* the diagonal so that $\lim_{z\to\infty} R(z) = 0$. The resulting methods of orders 1, 3, and 5 are (for $s = 1$ the implicit Euler method)

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array} \qquad \begin{array}{c|cc} \frac{1}{3} & \frac{5}{12} & -\frac{1}{12} \\ 1 & \frac{3}{4} & \frac{1}{4} \\ \hline & \frac{3}{4} & \frac{1}{4} \end{array}$$

$$\begin{array}{c|ccc} \frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\ \frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\ 1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\ \hline & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \end{array} \tag{15}$$

with stability functions $R_{s-1,s}$

$$\frac{1}{1-z}, \quad \frac{1+\frac{1}{3}z}{1-\frac{2}{3}z+\frac{1}{3}\frac{z^2}{2!}}, \quad \frac{1+\frac{2}{5}z+\frac{1}{10}\frac{z^2}{2!}}{1-\frac{3}{5}z+\frac{3}{10}\frac{z^2}{2!}-\frac{1}{10}\frac{z^3}{3!}}. \tag{16}$$

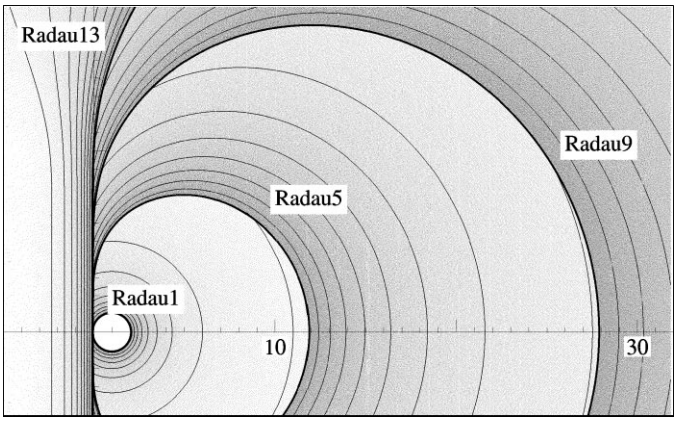

Fig. 5. Stability domains for RADAU methods of orders 1,5,9,13.

Their stability regions (for $s = 1, 3, 5, 7$ and $p = 1, 5, 9, 13$, respectively) are presented in Fig. 5. For the proof of their A-stability, one first studies the behaviour of $|R(\mathrm{i}y)|$ on the imaginary axis. The condition $|R(\mathrm{i}y)|^2 \leqslant 1$ becomes $E(y) := |Q(\mathrm{i}y)|^2 - |P(\mathrm{i}y)|^2 \geqslant 0$. For methods with $k < j$ and order $\geqslant 2s - 2$ this polynomial is of the form $C^2 y^{2s}$ and stability on the imaginary axis is clear. Next one inspects the location of the poles, which must be all in $\mathscr{C}^+$. Then, A-stability follows from the maximum principle. Ehle (as well as Axelsson) proved this property *for all s*. Ehle still proved that the whole second sub-diagonal $j = k + 2$ was A-stable too, and stated the *conjecture* that all other Padé approximations were not A-stable.

Two later elegant discoveries shed new light on Ehle's methods: the *order stars* and the interpretation of these formulas as *collocation methods*.

3.2. Order stars

The crucial idea is to replace the stability domain (6) by (see [42])

$$A := \{z; |R(z)| > |\mathrm{e}^z|\} \tag{17}$$

which compares the stability fuction to the exponential function, i.e., to the *true* solution (Fig. 6). With the help of this idea one is able to give an elegant proof for the A-stability of the Gauss and Radau IIA methods, as well as for Ehle's conjecture.

It came as a surprise that other outstanding problems could be solved too (restricted Padé approximations, multistep methods and the Daniel–Moore conjecture). For details we refer to [28, Sections IV.4, V.4].

3.3. Collocation methods

Hammer and Hollingsworth [29] discovered that the trapezoidal rule can be interpreted as generated by parabolas 'pieced together' in such a way that they 'agree in direction with that indicated by the differential equation at two points'. These authors conclude that the extension of this idea 'to higher order integration methods is straightforward'. It was then the elegant paper of Wright

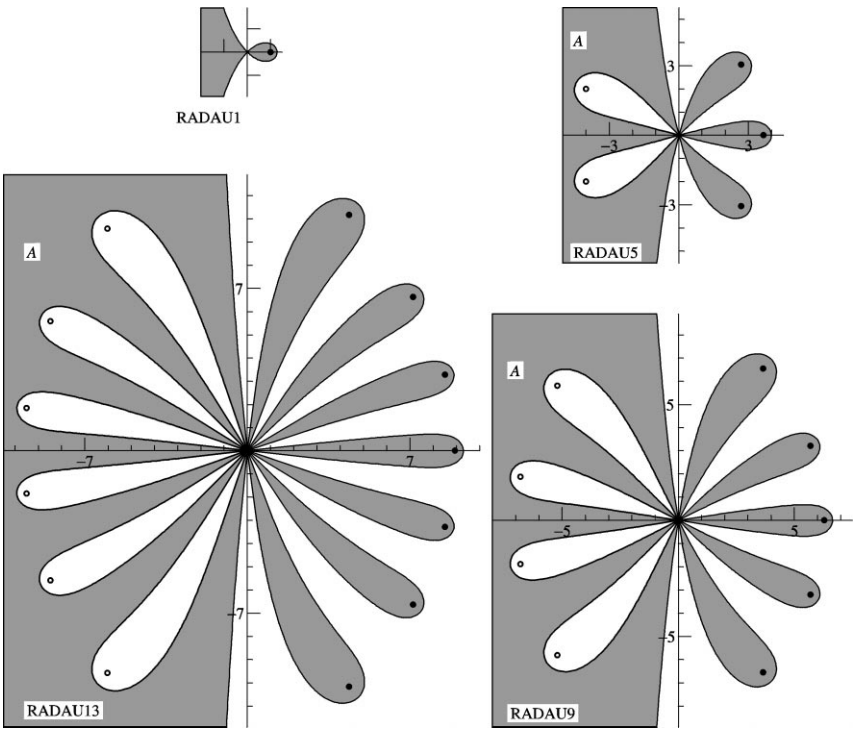

Fig. 6. Order stars for RADAU methods of orders 1,5,9,13.

[43] (see also Guillou and Soulé [24]) who identified many known classes of implicit Runge–Kutta methods as collocation methods. They are defined in the following way: Search for a polynomial $u(x)$ of degree s, whose derivative coincides at s given points $x_0 + c_i h$ $(i = 1, \ldots, s)$ with the vector field of the differential equation, i.e.,

$$u(x_0) = y_0 \quad \text{(initial value)},$$

$$u'(x_0 + c_i h) = f(x_0 + c_i h, u(x_0 + c_i h)), \quad i = 1, \ldots, s.$$

The numerical solution after one step is then given by $y_1 = u(x_0 + h)$. Radau IIA methods are precisely the collocation methods with the nodes given by (7). Fig. 7 illustrates the collocation methods of Gauss (order 4) and Radau IIA (order 3) at a nonstiff problem (above), as well as a stiff problem (below). The fourth-order method appears to be better in the nonstiff case only.

The equivalence of collocation methods with Runge–Kutta methods is established by applying Lagrange's interpolation formula to $u'(x_0 + th)$ with $u'(x_0 + c_i h) = k_i$. Then

$$a_{ij} = \int_0^{c_i} \ell_j(t)\,\mathrm{d}t, \quad b_j = \int_0^1 \ell_j(t)\,\mathrm{d}t \quad (i, j = 1, \ldots, s) \quad \text{where } \ell_j(t) = \prod_{k \neq j} \frac{(t - c_k)}{(c_j - c_k)}.$$

These coefficients satisfy (8) and (11) for $q = 1, \ldots, s$.

Another interesting consequence of the collocation idea is that an approximation to the solution is available on the *whole interval* $[x_0, x_0 + h]$ and not only at the endpoint ('dense output').

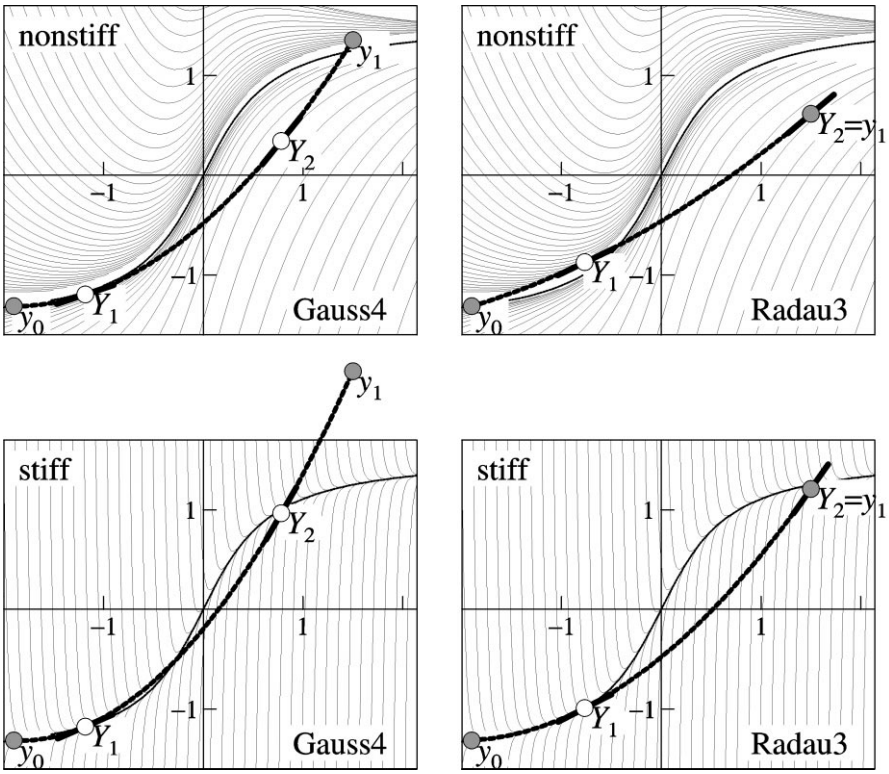

Fig. 7. Gauss method, $s = 2$ (left), Method Radau IIA, $s = 2$ (right).

3.4. B-stability

After the first success of the linear stability theory, people started to feel that the stability analysis based on linearization of the differential system which leads to the variational Eq. (2) and the subsequent suppression of the dependence of J on x is lacking rigour. It was finally Dahlquist [15] who found a satisfying frame for handling general nonlinear problems, and Butcher [7] who transferred these ideas to Runge–Kutta methods. The main idea is the following: two neighbouring solutions of a nonlinear system are approaching in the Euclidean norm if

$$\langle f(x, y) - f(x, z), y - z \rangle \leqslant 0. \tag{18}$$

The requirement for what is then called *B-stability* of a method is that then the same property must also hold for two neighbouring *numerical* solutions

$$\|y_1 - z_1\| \leqslant \|y_0 - z_0\|.$$

Butcher proved, among others, that the Radau IIA methods are B-stable. An elegant derivation of this property has been found in [41] by using the collocation idea.

For further references on the subsequent very rich development of this theory we refer especially to [16]; see also [9, Section 7] and [28, Chapter IV.12].

4. Convergence analysis for stiff problems

Convergence proofs for stiff differential equations are much more difficult than for nonstiff problems. The reason is that the factor $\exp((x_n - x_0)L)$, which is present in standard convergence estimates, is very large so that these estimates become useless (L is a Lipschitz constant of the problem).

4.1. Stiff accuracy

Prothero and Robinson [37] proposed the problem

$$y' = \lambda(y - \varphi(x)) + \varphi'(x), \quad y(x_0) = \varphi(x_0), \quad \Re\lambda \leqslant 0,$$

which allows explicit formulas for the local and global errors and provides much new insight. For $\lambda \to -\infty$ one can verify that the internal stages of a Runge–Kutta method (with invertible matrix a_{ij}) are very close to the exact solution $Y_i \approx \varphi(x_0 + c_i h)$, but the numerical approximation y_1 may be far away (see Fig. 7, lower pictures). This suggests to consider methods, for which y_1 is already one of the internal stages, say Y_s. This means that $a_{si} = b_i$ (all i) and is known as 'stiff accuracy', a property that is satisfied by Radau IIA methods.

4.2. Convergence for singular perturbation problems

An important class of stiff differential equations are of singular perturbation type:

$$\begin{aligned} y' &= f(y,z), \\ \varepsilon z' &= g(y,z), \end{aligned} \tag{19}$$

where $\varepsilon > 0$ is small and the eigenvalues of $\partial g/\partial z$ satisfy $\Re\lambda \leqslant -1$ along the solution. The problems considered by Fox and Godwin and Curtiss and Hirschfelder (1) as well as (29) and (30) below are of this type or can be brought to this form.

A typical convergence result is the following [26]: assume that the Runge–Kutta method is A-stable, is of classical order p, has a nonsingular coefficient matrix, statisfies $|R(\infty)| < 1$, and has *stage order* q (condition $C(q)$ of (11)). Then the global error satisfies

$$y_n - y(x_n) = \mathcal{O}(h^p) + \mathcal{O}(\varepsilon h^{q+1}), \quad z_n - z(x_n) = \mathcal{O}(h^{q+1}). \tag{20}$$

If in addition $a_{si} = b_i$ for all i, we have

$$z_n - z(x_n) = \mathcal{O}(h^p) + \mathcal{O}(\varepsilon h^q). \tag{21}$$

For the s-stage Radau IIA methods, for which $p = 2s - 1$ and $q = s$, we have (21) and the even sharper estimate $y_n - y(x_n) = \mathcal{O}(h^{2s-1}) + \mathcal{O}(\varepsilon^2 h^s)$ for the y-component.

4.3. B-convergence

Another type of convergence results can be obtained for stiff differential equations $y' = f(x,y)$ satisfying a *one-sided Lipschitz condition* (compare with (18) above)

$$\langle f(x,y) - f(x,z), y - z \rangle \leqslant \nu \|y - z\|^2,$$

where v is of moderate size. The analysis was developed by Frank et al. [22,23] and Dekker and Verwer [16]. As in the convergence results for singular perturbation problems, the stage order q plays an important role (see also [28, Chapter IV.15]).

4.4. Differential-algebraic equations

In the limit $\varepsilon \to 0$, the problem (19) becomes a differential equation for y coupled with an algebraic relation:

$$\begin{aligned} y' &= f(y,z), \\ 0 &= g(y,z). \end{aligned} \tag{22}$$

Implicit Runge–Kutta methods can be applied directly to (22) [36]. The idea is to apply the method to (19) and to consider in the resulting formulas the limit $\varepsilon \to 0$. Obviously, the numerical solution of stiffly accurate methods satisfies exactly the algebraic relation of (22). In the case that this algebraic relation can be solved for z (index 1), the investigation of convergence is easy, and it is an essential ingredient for the convergence results of singular perturbation problems. If the algebraic relation cannot be solved for z (higher index), the study of the global error is more complicated [27]. Typically an order reduction takes place, and methods with high stage order (such as Radau IIA) have favourable convergence properties.

4.5. Nonlinear parabolic differential equations

The study of Runge–Kutta methods applied to abstract differential equations in a Hilbert space (including parabolic problems) has been initiated by Crouzeix [11]. For B-stable methods energy estimates can be established which then allow elegant stability and convergence proofs [35]. It is interesting to note that the so-called 'discontinuous Galerkin methods' are, after a suitable discretization of the occurring integrals, equivalent to the Radau IIA methods [30].

5. Implementation

Runge–Kutta methods have been developed for ordinary differential equations $y' = f(x,y)$, but they can easily be adapted to problems of the form

$$My' = f(x,y), \tag{23}$$

where M is a constant (possibly singular) matrix. If we formally replace (23) by $y' = M^{-1}f(x,y)$, apply the Runge–Kutta method, and then multiply the formulas by M, we obtain the nonlinear system

$$M(Y_i - y_0) = h\sum_{j=1}^{s} a_{ij} f(x_0 + c_j h, Y_j) \tag{24}$$

instead of (9). For stiffly accurate methods (such as the Radau IIA methods) the numerical solution after one step is given by $y_1 = Y_s$.

Remark. More general problems than (23) can be handled by introducing new variables for the derivatives. For example, the general implicit differential equation $F(y', y) = 0$ is equivalent to the system

$$\begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} u' \\ v' \end{pmatrix} = \begin{pmatrix} v \\ F(v,u) \end{pmatrix},$$

by using the new variables $u = y$ and $v = y'$. The special structure of the right-hand side of this system can be exploited when solving linear systems. This compensates the doubling of the dimension of the system.

5.1. Solving the nonlinear Runge–Kutta system

For nonlinear differential equations (23) the system (24) has to be solved iteratively. Newton's method needs for each iteration the solution of a linear system with matrix

$$\begin{pmatrix} M - ha_{11}\dfrac{\partial f}{\partial y}(x_0 + c_1 h, Y_1) & \dots & -ha_{1s}\dfrac{\partial f}{\partial y}(x_0 + c_s h, Y_s) \\ \vdots & & \vdots \\ -ha_{s1}\dfrac{\partial f}{\partial y}(x_0 + c_1 h, Y_1) & \dots & M - ha_{ss}\dfrac{\partial f}{\partial y}(x_0 + c_s h, Y_s) \end{pmatrix}.$$

In order to simplify this, we replace all Jacobians $(\partial f/\partial y)(x_0 + c_i h, Y_i)$ by an approximation $J \approx (\partial f/\partial y)(x_0, y_0)$. Then, the simplified Newton iterations for (24) become, in the variables $Z_i := Y_i - y_0$,

$$(I \otimes M - hA \otimes J)\Delta Z^k = \dots, \qquad Z^{k+1} = Z^k + \Delta Z^k. \tag{25}$$

The supervector Z collects the stage values $(Z_1, \dots, Z_s)$ and the upper index in Z^k indicates the iteration number. Every iteration requires s evaluations of f and the solution of a $n \cdot s$-dimensional linear system, where the matrix $(I \otimes M - hA \otimes J)$ is the same for all iterations. Its LU-decomposition would be too costly. Therefore, Butcher [8] and Bickart [2] independently introduced an algorithm that exploits the special structure of the matrix $I \otimes M - hA \otimes J$ in (25) and thus reduces considerably the numerical work. The idea is to premultiply (25) by $(hA)^{-1} \otimes I$ (assuming that A is invertible) and to transform A^{-1} to a simple matrix (diagonal, block diagonal, triangular or Jordan canonical form)

$$T^{-1}A^{-1}T = \Lambda.$$

With the transformed variables $W^k = (T^{-1} \otimes I)Z^k$, iteration (25) becomes equivalent to

$$(h^{-1}\Lambda \otimes M - I \otimes J)\Delta W^k = \dots, \qquad W^{k+1} = W^k + \Delta W^k, \tag{26}$$

and the huge linear system is split into s linear systems of dimension n (for complex eigenvalues of A we have to deal with complex matrices).

5.2. Step size selection

Due to the superconvergence of the Radau IIA methods (classical order $p = 2s-1$) it is not possible to have an embedded method of order $p-1$ without extra cost. By taking a linear combination of $hf(x_0, y_0)$ and the internal stage values $Y_1, \ldots, Y_s$ it is however possible to get an approximation $\hat{y}_1$ of order s. The expression $\mathrm{err} = \|(M - h\gamma_0 J)^{-1}(\hat{y}_1 - y_1)\|$, where γ_0 is chosen such that the LU-factors of $M - h\gamma_0 J$ are already available from the solution of the nonlinear system, can be used for step size selection. The assumption $\mathrm{err}_{n+1} \approx C_n h_n^{s+1}$ (error in the nth step) together with $C_{n+1} \approx C_n$ leads to the standard strategy

$$h_{\mathrm{new}} = \mathrm{fac} \cdot h_n \left(\frac{1}{\mathrm{err}_{n+1}} \right)^{1/(s+1)}. \tag{27}$$

Here the user prescribed tolerance is incorporated in the norm (see [28, p. 124]).

A more sophisticated step size strategy ('step size control with memory') is based on the assumption $C_{n+1}/C_n \approx C_n/C_{n-1}$. It leads to the formula [25, 44]

$$h_{\mathrm{new}} = \mathrm{fac} \cdot h_n \left(\frac{1}{\mathrm{err}_{n+1}} \right)^{1/(s+1)} \frac{h_n}{h_{n-1}} \left(\frac{\mathrm{err}_n}{\mathrm{err}_{n+1}} \right)^{1/(s+1)}. \tag{28}$$

Our experience has shown that taking for h_{n+1} the minimum of the step sizes proposed by (27) and (28) results in a robust strategy. It automatically selects the step size of (27) in regions where the steps increase, and that of (28) where the steps decrease. In this way many step rejections are avoided that would appear by considering the strategy (27) only.

6. New order selection strategy and numerical results

If a class of methods with various orders is available, it is natural to search for an algorithm of a variable order implementation. The standard strategy (as used in extrapolation codes [18], in BDF codes, in STRIDE [4] and GAM [32]) is to choose the order p in such a way that the *error per unit step* is minimal, i.e.,

$$C^{(p)}/h^{(p)} \to \min,$$

where $C^{(p)}$ and $h^{(p)}$ denote the cost factors and the proposed step sizes for the method of order p. For implicit Runge–Kutta methods (such as Radau IIA) it is difficult to estimate the cost factors (the number of Newton iterations may depend strongly on the order), and it is also difficult to get reliable predictions for the optimal step sizes ('... a disadvantage of codes based on RK-formulas is that an order variation strategy is a hard task to handle because of the difficulty of having a convenient representation of the local truncation errors'. [32]).

During the preparation of the second edition of our monograph [28], we had written a code RADAUP which implements the Radau IIA methods of orders 5, 9, and 13, but only in fixed order mode. Many numerical experiments with this code have shown an interesting phenomenon which is explained in the following example.

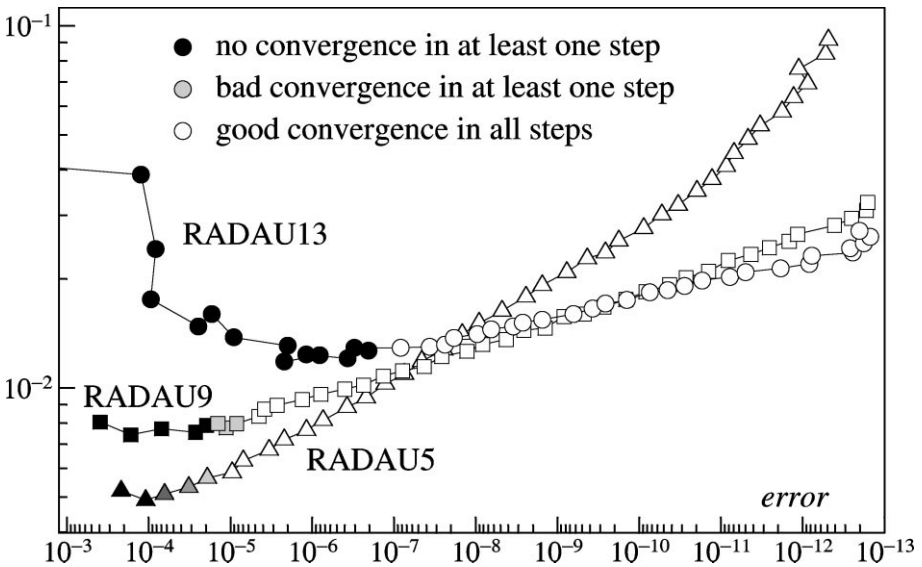

Fig. 8. Convergence of fixed-order Radau IIA codes.

6.1. Example

We consider the classical problem due to Robertson [39] which models a chemical reaction (problem ROBER in [28]). The equations and initial values are given by

$$\begin{aligned} y_1' &= -0.04y_1 + 10^4 y_2 y_3, & y_1(0) &= 1, \\ y_2' &= 0.04y_1 - 10^4 y_2 y_3 - 3 \times 10^7 y_2^2, & y_2(0) &= 0, \\ y_3' &= 3 \times 10^7 y_2^2, & y_3(0) &= 0, \end{aligned} \tag{29}$$

and the integration interval is $[0, 10^{11}]$. We apply our fixed-order code RADAUP, named as RADAU5 for order 5, RADAU9 for order 9, and RADAU13 for order 13, with many different tolerances $\mathrm{Rtol} = 10^{-2-m/4}$ $(m = 0, 1, \ldots, 40)$ and $\mathrm{Atol} = 10^{-6}\,\mathrm{Rtol}$. In Fig. 8 we plot the computing time as a function of the global error (the maximum error at $x = 1, 10, 10^2, \ldots, 10^{11}$) in a double logarithmic scale. The interesting observation is that for low tolerances the high-order methods perform much worse than as might be expected. Demanding less accuracy even increases the computer time. This is due to the fact that for the high-order methods the local error is very small, so that the step size strategy proposes very large steps. With these large step sizes the simplified Newton method for the nonlinear Runge–Kutta system has difficulties to converge. In Fig. 8 we have indicated by a black symbol the situations where convergence failed in at least one step. A grey symbol stands for slow convergence. As a conclusion we can say that high-order methods perform better than low-order methods as soon as the convergence of the simplified Newton iterations is sufficiently fast. For the Robertson problem this phenomenon can be observed very clearly, and experiments with many other problems indicate that this conclusion

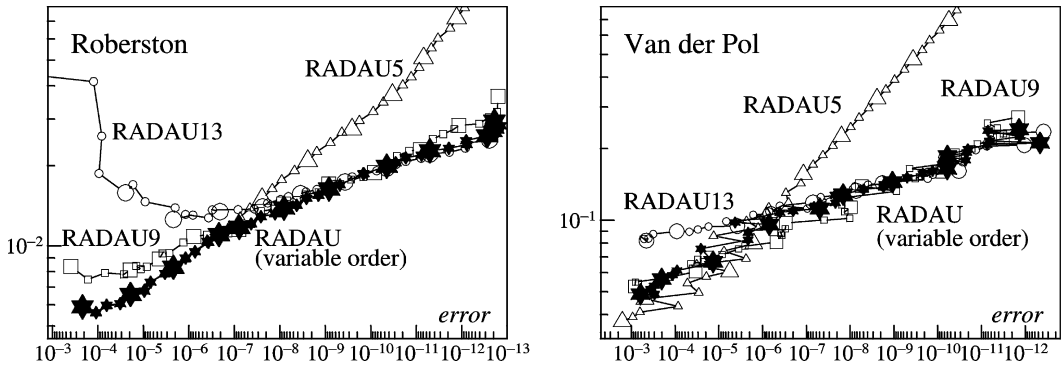

Fig. 9. Variable order code RADAU for Robertson's problem and for the Van der Pol oscillator.

applies in general. We are thus led to the following cheap and robust order selection strategy:

6.2. New order selection strategy (orders 5, 9, and 13)

We let $\Theta_k := \|\Delta W^k\|/\|\Delta W^{k-1}\|$ (for $k \geqslant 1$) be the quotient of two consecutive increments in the simplified Newton iteration (26), and denote

$$\Psi_1 := \Theta_1, \qquad \Psi_k := \sqrt{\Theta_k \cdot \Theta_{k-1}} \quad \text{for } k \geqslant 2.$$

The last Ψ_k in a step is called *contractivity factor*. We then select the orders as follows:

- start the computation with low order (say, $p = 5$), and do not change the order during the first 10 steps;
- increase the order by 4, if the contractivity factor is $\leqslant 0.002$ and if $p < 13$;
- decrease the order by 4, if the contractivity factor is $\geqslant 0.8$ or no convergence occurs and if $p > 5$;
- after a decrease of the order, an order increase is not allowed during 10 steps.

This order selection strategy is easy to implement and the choice of the threshold values prevents frequent oscillations in the order selection. Our variable order code, based on the Radau IIA methods of orders 5, 9, and 13 and on the above order selction strategy, is called RADAU. Its performance for the Robertson problem (29) is illustrated in Fig. 9 (the black stars). The larger symbols correspond to integer exponents in the tolerance $10^{-2}, 10^{-3}, \ldots, 10^{-12}$. We see that for a given tolerance this code performs exactly as the best code among RADAU5, RADAU9, RADAU13. We cannot hope for more. The number of steps taken with the different orders are given in Table 1. For low tolerances, the code does not switch to higher orders at all. Fort stringent tolerances, the code takes a few steps with order 5, then some steps with order 9, and soon switches to the optimal order 13. Since the solution of the Robertson problem tends to a steady-state solution, the code never decreases the order again.

Table 1
Number of steps of orders 5, 9, and 13 taken by RADAU

Rtol	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}	10^{-9}	10^{-10}	10^{-11}	10^{-12}
Order 5	87	111	144	195	10	10	10	10	10	10	10
Order 9	0	0	0	0	98	116	138	12	14	12	12
Order 13	0	0	0	0	0	0	0	90	102	117	134

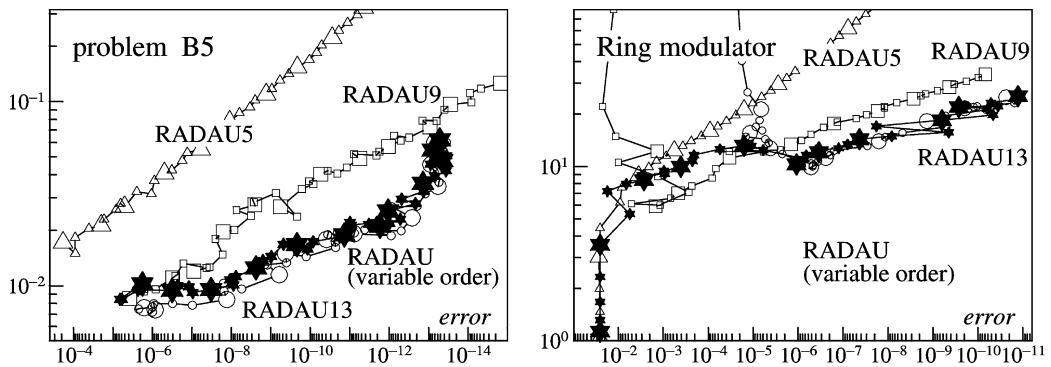

Fig. 10. RADAU for linear problem B5 and for the ring modulator.

6.3. Further examples

As a second example we consider Van der Pol's equation

$$\begin{aligned} y_1' &= y_2, & y_1(0) &= 2, \\ \varepsilon y_2' &= (1 - y_1^2)y_2 - y_1, & y_2(0) &= 0, \end{aligned} \tag{30}$$

with $\varepsilon = 10^{-6}$ on the interval $[0, 11]$ (problem VDPOL of [28]). Its solution tends to a limit cycle, in which transient parts in the solution alternate with intervals where the problem is very stiff. Fig. 9 (right picture) shows the computing times of the fixed-order codes together with those of the new code RADAU as a function of the global error (similar to the previous figure). For large tolerances ($\mathrm{Rtol} > 10^{-5}$) the code RADAU works with order 5 in the stiff regions, and it takes order 9 in the (nonstiff) transients, where small step sizes are used. For more stringent tolerances the order switches in a similar way between order 9 and order 13. For very stringent tolerances ($\mathrm{Rtol} \leqslant 10^{-8}$) the code quickly switches to order 13 and remains at this high order until the end of integration.

For linear differential equations with constant coefficients, the simplified Newton method gives the exact solution after one iteration already. Therefore, after a short initial phase the code RADAU will switch to the highest possible order 13. Fig. 10 (left picture) shows the results for the famous problem 'B5' of [20] with eigenvalues $-10 \pm 100\mathrm{i}$, -4, -1, -0.5 and -0.1. We have taken $\mathrm{Atol} = 10^{-6}\,\mathrm{Rtol}$. It turns out that already for low tolerances the high-order methods are most efficient.

As a last example we consider a more engaging problem of dimension $n=15$, the ring modulator. It is a model for a small circuit and has been introduced by Horneber in 1976. Promoted by [17,27] as a test problem for stiff integrators, it is included in the Amsterdam test set (CWI) at http://www.cwi.nl/cwi/projects/IVPtestset.shtml. We choose the parameter value $C_S=2\times10^{-12}$ and the interval $[0,10^{-3}]$. The Jacobian of the problem along the solution has complex conjugate eigenvalues (approximately $-2\times10^4\pm i3\times10^7$), so that some components of the solution are highly oscillatory with amplitude of size 0.1 (much larger than the tolerance). There are real positive eigenvalues of size $\approx10^5$ (nonstiff), and also real negative eigenvalues of size $\approx-10^{11}$ which make the problem very stiff. For this problem the high-order methods are very expensive at low tolerances, but are very efficient for Rtol $\leqslant10^{-5}$. The variable order code RADAU correctly selects a nearly optimal order at all tolerances (see Fig. 10, right picture).

All our codes (including RADAU) are available on the Internet at the address
http://www.unige.ch/math/folks/hairer/
Experimentations with these codes are welcome.

Acknowledgements

The authors are grateful to E. Neuenschwander and V.I. Lebedev for useful historical informations, and to Ch. Lubich for his valuable comments on early drafts of this paper.

References

[1] O. Axelsson, A class of A-stable methods, BIT 9 (1969) 185–199.
[2] T.A. Bickart, An efficient solution process for implicit Runge–Kutta methods, SIAM J. Numer. Anal. 14 (1977) 1022–1027.
[3] G. Birkhoff, R.S. Varga, Discretization errors for well-set Cauchy problems, I, J. Math. Phys. 44 (1965) 1–23.
[4] K. Burrage, J.C. Butcher, F.H. Chipman, An implementation of singly-implicit Runge–Kutta methods, BIT 20 (1980) 326–340.
[5] J.C. Butcher, Implicit Runge–Kutta processes, Math. Comput. 18 (1964) 50–64.
[6] J.C. Butcher, Integration processes based on Radau quadrature formulas, Math. Comput. 18 (1964) 233–244.
[7] J.C. Butcher, A stability property of implicit Runge–Kutta methods, BIT 15 (1975) 358–361.
[8] J.C. Butcher, On the implementation of implicit Runge–Kutta methods, BIT 6 (1976) 237–240.
[9] J.C. Butcher, G. Wanner, Runge–Kutta methods: some historical notes, Appl. Numer. Math. 22 (1996) 113–151.
[10] J. Crank, P. Nicolson, A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type, Proc. Cambridge Philos. Soc. 43 (1947) 50–67.
[11] M. Crouzeix, Sur l'approximation des équations différentielles opérationnelles linéaires par des méthodes de Runge–Kutta, Thèse d'Etat, Univ. Paris 6, 1975.
[12] C.F. Curtiss, J.O. Hirschfelder, Integration of stiff equations, Proc. Natl. Acad. Sci. 38 (1952) 235–243.
[13] G. Dahlquist, Fehlerabschätzungen bei Differenzenmethoden zur numerischen Integration gewöhnlicher Differentialgleichungen, ZAMM 31 (1951) 239–240.
[14] G. Dahlquist, A special stability problem for linear multistep methods, BIT 3 (1963) 27–43.
[15] G. Dahlquist, Error analysis for a class of methods for stiff nonlinear initial value problems, Numerical Analysis, Dundee, Lecture Notes in Math., Vol. 506, 1975, pp. 60–74.
[16] K. Dekker, J.G. Verwer, Stability of Runge–Kutta Methods for Stiff Nonlinear Differential Equations, North-Holland, Amsterdam, 1984.
[17] G. Denk, P. Rentrop, Mathematical models in electric circuit simulation and their numerical treatment, in: Numerical Treatment of Differential Equations, Halle/DDR, Teubner-Texte zur Mathematik, 1989.

[18] P. Deuflhard, Order and stepsize control in extrapolation methods, Numer. Math. 41 (1983) 399–422.
[19] B.L. Ehle, On Padé approximations to the exponential function and A-stable methods for the numerical solution of initial value problems, Report CSRR 2010, Dept. AACS, Univ. of Waterloo, Ontario, Canada. See also: BIT 8 (1969) 276–278; SIAM J. Math. Anal. 4 (1969) 671–680.
[20] W.H. Enright, T.E. Hull, B. Lindberg, Comparing numerical methods for stiff systems of O.D.E:s, BIT 15 (1975) 10–48.
[21] L. Fox, E.T. Goodwin, Some new methods for the numerical integration of ordinary differential equations, Proc. Cambridge Philos. Soc. 45 (1949) 373–388.
[22] R. Frank, J. Schneid, C.W. Ueberhuber, The concept of B-convergence, SIAM J. Numer. Anal. 18 (1981) 753–780.
[23] R. Frank, J. Schneid, C.W. Ueberhuber, Order results for implicit Runge–Kutta methods applied to stiff systems, SIAM J. Numer. Anal. 22 (1985) 515–534.
[24] A. Guillou, J.L. Soulé, La résolution numérique des problèmes différentiels aux conditions initiales par des méthodes de collocation, R.I.R.O, No R-3, 1969, pp. 17–44.
[25] K. Gustafsson, Control-theoretic techniques for stepsize selection in implicit Runge–Kutta methods, ACM Trans. Math. Software 20 (1994) 496–517.
[26] E. Hairer, Ch. Lubich, M. Roche, Error of Runge–Kutta methods for stiff problems studied via differential algebraic equations, BIT 28 (1988) 678–700.
[27] E. Hairer, Ch. Lubich, M. Roche, The numerical solution of differential-algebraic systems by Runge–Kutta methods, Lecture Notes in Math., Vol. 1409, Springer, Berlin, 1989.
[28] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems, 2nd revised Edition, Springer Series in Comput. Math., Vol. 14, Springer, Berlin, 1996, 614pp.
[29] P.C. Hammer, J.W. Hollingsworth, Trapezoidal methods of approximating solutions of differential equations, MTAC 9 (1955) 92–96.
[30] B.L. Hulme, Discrete Galerkin and related one-step methods for ordinary differential equations, Math. Comput. 26 (1972) 881–891.
[31] A. Huťa, Une amélioration de la méthode de Runge–Kutta-Nyström pour la résolution numérique des équations différentielles du premier ordre, Acta Math. Univ. Comenian. 1 (1956) 201–224.
[32] F. Iavernaro, F. Mazzia, Solving ordinary differential equations by generalized Adams methods: properties and implementation techniques, Report n. 24/97, Università degli studi di Bari, Italy, 1997.
[33] W. Liniger, R.A. Willoughby, Efficient integration methods for stiff systems of ordinary differential equations, SIAM J. Numer. Anal. 7 (1970) 47–66.
[34] W.S. Loud, On the long-run error in the numerical solution of certain differential equations, J. Math. Phys. 28 (1) (1949) 45–49.
[35] Ch. Lubich, A. Ostermann, Runge–Kutta approximation of quasi-linear parabolic equations, Math. Comput. 64 (1995) 601–627.
[36] L. Petzold, Order results for implicit Runge–Kutta methods applied to differential/algebraic systems, SIAM J. Numer. Anal. 23 (1986) 837–852.
[37] A. Prothero, A. Robinson, On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations, Math. Comput. 28 (1974) 145–162.
[38] R. Radau, Étude sur les formules d'approximation qui servent à calculer la valeur numérique d'une intégrale définie, J. Math. Pures Appl. Sér. 3, 6 (1880) 283–336.
[39] H.H. Robertson, The solution of a set of reaction rate equations, in: J. Walsh (Ed.), Numerical Analysis, an Introduction, Academic Press, New York, 1966, pp. 178–182.
[40] H. Rutishauser, Über die Instabilität von Methoden zur Integration von gewöhnlichen Differentialgleichungen, Z. Angew. Math. Phys. 3 (1952) 65–74.
[41] G. Wanner, A short proof on nonlinear A-stability, BIT 16 (1976) 226–227.
[42] G. Wanner, E. Hairer, S.P. Nørsett, Order stars and stability theorems, BIT 18 (1978) 475–489.
[43] K. Wright, Some relationships between implicit Runge–Kutta collocation and Lanczos τ methods, and their stability properties, BIT 10 (1970) 217–227.
[44] H.A. Watts, Step size control in ordinary differential equation solvers, Trans. Soc. Computer Simulation 1 (1984) 15–25.