Journal of Applied Mathematics and Physics (ZAMP)
Vol. 30, 1979 Birkhäuser Verlag Basel

Runge–Kutta Algorithms for Oscillatory Problems

By D. G. Bettis, Texas Institute for Computational Mechanics, University of Texas at Austin, Texas 78712, U.S.A.

This paper is dedicated to the memory of Prof. Dr. E. Stiefel.

I. Introduction

The solution of the initial value problem for the classical Runge–Kutta algorithms is based on the condition that they are equivalent to a Taylor series approximation of a specified order. For problems that are characterized by solutions that are of an oscillatory nature, the coefficients of a Runge–Kutta algorithm can be selected such that this particular problem is solved without truncation errors. For oscillatory problems with complex eigenvalues that possess small real components, these methods can be more efficient than the classical Runge–Kutta methods. A similar approach for modifying the finite difference methods of Störmer–Cowell has been developed by Stiefel and Bettis [1969], and for Adams–Bashforth–Moulton methods by Bettis [1969, 1970]. This paper presents modified Runge–Kutta algorithms of 3 stages, 4 stages, and an embedded 3 stage–4 stage pair.

Consider an s stage Runge–Kutta algorithm for the solution of the system of equations

$$\frac{dY}{dt} = f(t, Y), \qquad Y(t_0) = Y_0.$$

The algorithm is

$$Y(t_0 + h) = Y_0 + h \sum_{i=0}^{r} C_i f_i,$$

$$f_0 = f(t_0, Y_0)$$

$$f_j = f\left(t_0 + \alpha_j h,\ Y_0 + h \sum_{k=0}^{j-1} \beta_{jk} f_k\right), \quad j = 1, \ldots, r,$$

with $s = r + 1$, and where h is the step size.

For a linear differential equation,

$$\frac{dY}{dt} = \lambda Y,$$

with λ complex, the expansions of the Runge–Kutta algorithm becomes

$$Y(t_0 + h) = Y_0\left[1 + h\lambda \sum_{i=0}^{r} C_i + (h\lambda)^2 \sum_{i=1}^{r} C_i \sum_{j=0}^{i-1} \beta_{ij} + (h\lambda)^3 \sum_{i=2}^{r} C_i \sum_{j=1}^{i-1} \beta_{ij} \sum_{k=0}^{j-1} \beta_{jk} + \cdots\right]. \tag{1}$$

The conditions that must be satisfied for the Runge–Kutta solution to be equivalent to that of a Taylor series may be obtained by a direct comparison of similar terms of the above expansion with a Taylor series. It is important to note that for a system of nonlinear equations, additional terms are involved, and the resulting comparison of series results in the usual complete set of equations of condition.

For the linear equation $Y(t_0 + h) = e^{\lambda h}$. Let the initial conditions be normalized with $t_0 = 0$, $Y(t_0) = 1$. For an s stage Runge–Kutta algorithm to solve the linear equation, the coefficients must be selected so that the normalized Eqn. (1) is satisfied.

Now consider the particular case where λ is purely imaginary, $\lambda = i\omega$, and let $\omega h = \sigma$. Let all but the last pair of nonvanishing coefficients of the $(\lambda h)^p$ multiples assume the value $1/p!$, their Taylor series equivalent. The two conditions necessary for (1) to be satisfied are that the remaining coefficients of the $(\lambda h)^i$, $i = p + 1, p + 2$, terms be equal to μ_i, where

$$\mu_i = \frac{1}{i!} + \sum_{j=1}^{\infty} \frac{(-1)^j \sigma^{2j}}{(i + 2j)!}.$$

This expansion is useful for computing the μ_i for small σ. Otherwise, they may be computed by the recurrence relation

$$\mu_i = \left[\frac{1}{(i - 2)!} - \mu_{i-2}\right]\Big/\sigma^2, \qquad \mu_0 = \cos\sigma, \qquad \mu_1 = \sin\sigma/\sigma.$$

It is noted that in the limit, as σ becomes zero, the μ_i become $1/i!$, and the conditions become the same as the standard equations of condition.

Thus, by requiring the last two 'linear' equations of condition to be equal to μ_i, the resulting Runge–Kutta algorithm will solve the equation $dY/dt = i\omega Y$ without truncation error. That is, the simple harmonic oscillator can be solved exactly.

II. A Three-Stage Algorithm

For a three-stage Runge–Kutta algorithm, the 'linear' equations of condition become

$$\begin{aligned} &\sum_{i=0}^{2} C_i = 1 \\ &\sum_{i=1}^{2} C_i \sum_{j=0}^{i-1} \beta_{ij} = \mu_2 \\ &C_2\beta_{21}\beta_{10} = \mu_3 \end{aligned} \tag{2}$$

The first equation of (2) determines the value of C_0. The two coefficients, β_{20} and β_{21}, can be expressed as

$$\beta_{21} = \mu_3/C_2\beta_{10}$$

$$\beta_{20} = (\mu_2 - C_1\beta_{10} - \mu_3/\beta_{10})/C_2.$$

The coefficients β_{10}, C_1, and C_2 remain as free parameters, and they may be selected so that the remaining 'nonlinear' equation of condition for a Runge–Kutta algorithm of order three is satisfied. That is,

$$\tfrac{1}{2}[C_1\beta_{10}^2 + C_2(\beta_{20} + \beta_{21})^2] - \tfrac{1}{6}$$

should vanish. Therefore, let C_2 be given by

$$C_2 = \frac{(\mu_2 - C_1\beta_{10})^2}{\frac{1}{3} - C_1\beta_{10}^2}.$$

Selecting the remaining coefficients $\beta_{10} = \frac{1}{2}$ and $C_1 = \frac{1}{3}$, will result in relatively small truncation error terms of order four when σ is small. For a nonautonomous system of differential equations, the α_i are defined as

$$\alpha_i = \sum_{j=0}^{i-1} \beta_{ij}. \tag{3}$$

Finally, the coefficients for a three-stage algorithm are

$$\beta_{10} = \tfrac{1}{2}$$

$$\beta_{20} = \frac{3(6\mu_2 - 12\mu_3 - 1)}{2(6\mu_2 - 1)^2}$$

$$\beta_{21} = \frac{18\mu_3}{(6\mu_2 - 1)^2}$$

$$C_1 = \tfrac{1}{3}$$

$$C_2 = \frac{(6\mu_2 - 1)^2}{9}.$$

III. A Four-Stage Algorithm

A four-stage algorithm may be determined so that, as σ becomes zero, the method will be of order four, in the classical sense. This method will have two free parameters.

Let the two free parameters be

$$\alpha_2 = \frac{1 - \sigma}{2}$$

$$\alpha_3 = 1 - \sigma$$

There results a restrictive condition between α_1 and α_3:

$$\alpha_1\alpha_3 A + \alpha_1 B + \alpha_3 C + D = 0, \tag{4}$$

where

$$A = \frac{\mu_4}{2} - \mu_3^2$$

$$B = \frac{\mu_3}{8} - \frac{\mu_4}{3}$$

$$C = \frac{\mu_3}{12} - \frac{\mu_4}{3}$$

$$D = \frac{\mu_4}{4} - \frac{1}{96}.$$

Using (4) to determine α_1, the C coefficients may be determined from

$$C_1 = \frac{3 - 4(\alpha_2 + \alpha_3) + 6\alpha_2\alpha_3}{12\alpha_1(\alpha_1 - \alpha_2)(\alpha_1 - \alpha_3)}$$

$$C_2 = \frac{3 - 4(\alpha_1 + \alpha_3) + 6\alpha_1\alpha_3}{12\alpha_3(\alpha_2 - \alpha_1)(\alpha_2 - \alpha_3)}$$

$$C_3 = \frac{3 - 4(\alpha_1 + \alpha_2) + 6\alpha_1\alpha_2}{12\alpha_3(\alpha_3 - \alpha_1)(\alpha_3 - \alpha_2)}$$

$$C_0 = 1 - C_1 - C_2 - C_3.$$

Then

$$\beta_{21} = (\tfrac{1}{8} - \alpha_3\mu_3)/[C_2\alpha_1(\alpha_2 - \alpha_3)]$$

$$\beta_{32} = \mu_4/C_3\beta_{21}\alpha_1$$

$$\beta_{31} = (\mu_3 - C_2\beta_{21}\alpha_1 - C_3\alpha_2\beta_{32})/C_3\alpha_1.$$

The β_{i0} coefficients, $i = 1, 2, 3$, are determined from (3).

When σ vanishes, the above coefficients become identical to the coefficients of the classical Runge–Kutta algorithm of order four.

IV. An Embedded 3 Stage–4 Stage Pair

For the case in which σ is zero, contradictions in the equations of condition preclude the existence of a 3, 4 stage algorithm of orders 3, 4 [Fehlberg, 1970]. However, when σ is not zero, a 3, 4 stage algorithm is feasible, with the advantage that the difference between the two solutions offers an estimate of the truncation errors. Specifically, this provides an estimate of the contributions of the nonlinear truncation errors of order four, when the coefficients of the 3-stage solution are chosen so that they satisfy (1).

In order to embed a 3-stage algorithm into the previously developed 4-stage algorithm, the free parameter α_3 has to be selected so that the following restrictive condition is satisfied

$$\alpha_1\alpha_3 A + \alpha_1 B + C = 0, \tag{5}$$

where

$$A = 12\mu_2\mu_3 - 6\mu_3$$

$$B = 4\mu_3 - \tfrac{3}{2}\mu_2$$

$$C = \tfrac{1}{2} - 3\mu_3$$

Thus, α_1 and α_3 must satisfy both (4) and (5).

The coefficients for the 3-stage algorithm will be denoted as C_i^*, $i = 0, 1, 2$ (the α and β coefficients are equivalent to those of the 4-stage algorithm). C_1^* is a free parameter, but α_2 is not. Selecting $C_1^* = \frac{1}{3}$,

$$\alpha_2 = \frac{\frac{1}{3} - C_1^*\alpha_1^2}{\mu_2 - C_1^*\alpha_1}$$

Then

$$C_2^* = \frac{\frac{1}{3} - \alpha_1\mu_2}{\alpha_2(\alpha_2 - \alpha_1)}$$

$$C_0^* = 1 - C_1^* - C_2^*.$$

V. Conclusions

Runge–Kutta algorithms with three stages and four stages have been presented that possess modified coefficients so that the new algorithm is characterized by the ability to solve without truncation error an equation whose solution is the exponential function $e^{i\omega t}$. Numerical experiments indicate that these algorithms are efficient for the class of differential equations which possess highly-oscillatory components due to eigenvalues that have small real parts.

An embedded 3, 4 stage algorithm is given which provides an error estimate of the nonlinear truncation error terms of order four. Results with this method indicate that when σ is small, numerical difficulties can occur. This is to be expected, since a 3, 4 stage third-, fourth-order algorithm is not possible.

Acknowledgements

Support from the National Science Foundation (Grant MCS76–20022) is greatly appreciated.

References

[1] D. G. Bettis, *Numerical Integration of Fourier and Ordinary Polynomials*, Num. Math. *14* (1970).
[2] D. G. Bettis, *Stabilization of Finite Difference Methods of Numerical Integration*, Cel. Mech. *2* (1970).
[3] E. Fehlberg, *Klassische Runge–Kutta–Formeln vierter und niedrigerer Ordnung mit Schrittweiten-Kontrolle und ihre Anwendung auf Wärmeleitungsprobleme*, Computing *6* (1970).
[4] E. Stiefel and D. G. Bettis, *Stabilization of Cowell's Method*, Num. Math. *13* (1969).

Abstract

For the numerical integration of differential equations with oscillatory solutions adapted Runge–Kutta algorithms of up to 4 stages are presented. The coefficients of these methods are chosen such that certain particular oscillatory solutions are computed without truncation errors.

Zusammenfassung

Für die numerische Integration von Differentialgleichungen mit rasch oszillierenden Lösungen werden angepasste Runge–Kutta–Algorithmen mit bis zu 4 Stufen konstruiert. Die Koeffizienten dieser Verfahren werden so gewählt, dass gewisse spezielle oszillierende Lösungen ohne Abbrechfehler berechnet werden.

(Received: January 31, 1979)