Journal of Computational and Applied Mathematics 228 (2009) 344–354

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

General error propagation in the RK*r*GL*m* method

J.S.C. Prentice

Department of Applied Mathematics, University of Johannesburg, P.O. Box 524, Auckland Park, 2006, South Africa

ARTICLE INFO

Article history:
Received 4 February 2008
Received in revised form 19 September 2008

Keywords:
RK*r*GL*m*
Runge–Kutta
Gauss–Legendre
Initial value problem
Order
Local error
Global error

ABSTRACT

The RK*r*GL*m* method is a numerical method for solving initial value problems in ordinary differential equations of the form $\mathbf{y}' = \mathbf{f}(x, \mathbf{y})$ and is based on a combination of a Runge–Kutta method of order r and m-point Gauss–Legendre quadrature. In this paper we describe the propagation of local errors in this method, and we give an inductive proof of the form of the global error in RK*r*GL*m*. We show that, for a suitable choice of r and m, the global order of RK*r*GL*m* is expected to be $r + 1$, one better than the underlying Runge–Kutta method. We show that this gain in order is due to a reduction or "quenching" of the accumulated local error at every $(m + 1)$th node. We also show how a Hermite interpolating polynomial of degree $2m + 1$ may be employed to estimate $\mathbf{f}(x, \mathbf{y})$ if the nodes to be used for the Gauss–Legendre quadrature component are not suitably placed.

1. Introduction

One-step methods, such as Runge–Kutta (RK) methods, are popular methods for solving initial value problems in ordinary differential equations of the form $\mathbf{y}' = \mathbf{f}(x, \mathbf{y})$ numerically. Of interest in such methods is the propagation of approximation error, and the cumulative effect of this propagation. In an RK method, the accumulation of $O\left(h^{r+1}\right)$ local errors results in a global error of $O\left(h^r\right)$, where h is the stepsize. In other words, the global order of an RK method is one less than its local order. We have developed a method [6], designated RK*r*GL*m*, which is a combination of an RK method of global order r, and m-point Gauss–Legendre (GL) quadrature, that has the interesting property that if the underlying RK method is $O\left(h^{r+1}\right)$ in its local error, then the associated RK*r*GL*m* method is $O\left(h^{r+1}\right)$ in its *global* error, i.e. the global error in RK*r*GL*m* has the same order as the local RK error (for the benefit of the reader, a brief description of RK*r*GL*m* will be given in the next section). In this paper we add to the results of our previous work by means of the following: (a) we describe in detail the propagation of error in the RK*r*GL*m* method, (b) we show how the global error of RK*r*GL*m* achieves $O\left(h^{r+1}\right)$ by considering the accumulation of local error, (c) we give an inductive proof of the general structure of the global error, and (d) we clearly demonstrate by means of numerical examples how the GL component slows the accumulation of error (an effect we refer to as *error quenching*). We also consider the use of a Hermite interpolating polynomial for estimating the derivative $\mathbf{f}(x, \mathbf{y})$ if the RK nodes are not suitably placed for GL quadrature. Furthermore, our discussion is general, in the sense that we consider the application of RK*r*GL*m* to *systems* of differential equations, unlike our previous work where we considered only the scalar case.

2. Terminology and relevant concepts

In this section we describe notation, terminology and concepts relevant to the rest of the paper. Note that, throughout this paper, boldface type, as in $\mathbf{v}$, indicates a $q \times 1$ vector, and boldface type with caret, as in $\hat{\mathbf{M}}$, denotes a $q \times q$ matrix.

E-mail address: jprentice@uj.ac.za.

0377-0427/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2008.09.032

2.1. Explicit Runge–Kutta methods

We denote an explicit RK method for solving the q-dimensional system

$$\mathbf{y}' = \mathbf{f}(x, \mathbf{y}) \qquad \mathbf{y}(x_0) = \mathbf{y}_0 \quad a \leqslant x \leqslant b \tag{1}$$

by

$$\mathbf{w}_{i+1} = \mathbf{w}_i + h_i \mathbf{F}(x_i, \mathbf{w}_i)$$

where $h_i \doteqdot x_{i+1} - x_i$ is a stepsize, $\mathbf{w}_i$ denotes the numerical approximation to $\mathbf{y}(x_i)$ and $\mathbf{F}(x, \mathbf{y})$ is a function associated with the particular RK method. Here and throughout the symbol $\doteqdot$ is used to indicate a definition.

2.2. Local and global errors

We define the global error in a numerical solution at x_i by

$$\boldsymbol{\Delta}_i \doteqdot \mathbf{w}_i - \mathbf{y}_i,$$

and the local error at x_i by

$$\boldsymbol{\varepsilon}_{i+1} \doteqdot \left[\mathbf{y}_i + h_i \mathbf{F}(x_i, \mathbf{y}_i)\right] - \mathbf{y}_{i+1}. \tag{2}$$

In the above, $\mathbf{y}_i$ denotes the true solution $\mathbf{y}(x_i)$. Note the use of the exact value $\mathbf{y}_i$ in the bracketed term in (2).

2.3. Error propagation in a Runge–Kutta method

We describe a known result that is useful in our later discussion. For the sake of generality we will assume an error $\boldsymbol{\Delta}_0$ exists in the initial value, although in most practical cases $\boldsymbol{\Delta}_0 = \mathbf{0}$. We have

$$\begin{aligned}
&\mathbf{w}_1 = \mathbf{y}_0 + \boldsymbol{\Delta}_0 + h_0 \mathbf{F}(x_0, \mathbf{y}_0 + \boldsymbol{\Delta}_0) \\
&\Rightarrow \boldsymbol{\Delta}_1 = \left[\mathbf{y}_0 + h_0 \mathbf{F}(x_0, \mathbf{y}_0) - \mathbf{y}_1\right] + \left[\hat{\mathbf{I}} + h_0 \hat{\mathbf{F}}_y\left(x_0, \boldsymbol{\xi}_0\right)\right] \boldsymbol{\Delta}_0 \\
&\qquad\qquad = \boldsymbol{\varepsilon}_1 + \hat{\boldsymbol{\alpha}}_0 \boldsymbol{\Delta}_0
\end{aligned}$$

where $\hat{\boldsymbol{\alpha}}_0$ has been implicitly defined. In the above we use the symbol $\boldsymbol{\xi}_0$ in $\hat{\mathbf{F}}_y\left(x_0, \boldsymbol{\xi}_0\right) \boldsymbol{\Delta}_0$ simply to denote an appropriate set of constants such that $\hat{\mathbf{F}}_y\left(x_0, \boldsymbol{\xi}_0\right) \boldsymbol{\Delta}_0$ is the residual term in the first-order Taylor expansion of $\mathbf{F}(x_0, \mathbf{y}_0 + \boldsymbol{\Delta}_0)$. Moreover, $\hat{\mathbf{F}}_y$ is the Jacobian

$$\hat{\mathbf{F}}_y = \begin{bmatrix} \dfrac{\partial F_1}{\partial y_1} & \cdots & \dfrac{\partial F_1}{\partial y_q} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial F_q}{\partial y_1} & \cdots & \dfrac{\partial F_q}{\partial y_q} \end{bmatrix} \tag{3}$$

where $\left\{F_1, F_2, \ldots, F_q\right\}$ are the components of $\mathbf{F}$. The matrix $\hat{\mathbf{I}}$ is the identity matrix.

For $\boldsymbol{\Delta}_2$ we have

$$\begin{aligned}
&\mathbf{w}_2 = \mathbf{w}_1 + h_1 \mathbf{F}(x_1, \mathbf{w}_1) \\
&\Rightarrow \mathbf{y}_2 + \boldsymbol{\Delta}_2 = [\mathbf{y}_1 + \boldsymbol{\Delta}_1] + h_1 \mathbf{F}(x_1, \mathbf{y}_1 + \boldsymbol{\Delta}_1) \\
&\qquad\qquad = [\mathbf{y}_1 + \boldsymbol{\Delta}_1] + h_1 \mathbf{F}(x_1, \mathbf{y}_1) + h_1 \hat{\mathbf{F}}_y\left(x_1, \boldsymbol{\xi}_1\right) \boldsymbol{\Delta}_1 \\
&\Rightarrow \boldsymbol{\Delta}_2 = [\mathbf{y}_1 + h_1 \mathbf{F}(x_1, \mathbf{y}_1) - \mathbf{y}_2] + \left[\hat{\mathbf{I}} + h_1 \hat{\mathbf{F}}_y\left(x_1, \boldsymbol{\xi}_1\right)\right] \boldsymbol{\Delta}_1 \\
&\qquad\qquad = \boldsymbol{\varepsilon}_2 + \hat{\boldsymbol{\alpha}}_1 \boldsymbol{\Delta}_1 \\
&\qquad\qquad = \boldsymbol{\varepsilon}_2 + \hat{\boldsymbol{\alpha}}_1 \boldsymbol{\varepsilon}_1 + \hat{\boldsymbol{\alpha}}_1 \hat{\boldsymbol{\alpha}}_0 \boldsymbol{\Delta}_0.
\end{aligned}$$

It is easy to show that

$$\begin{aligned}
\boldsymbol{\Delta}_3 &= \boldsymbol{\varepsilon}_3 + \hat{\boldsymbol{\alpha}}_2 \boldsymbol{\varepsilon}_2 + \hat{\boldsymbol{\alpha}}_2 \hat{\boldsymbol{\alpha}}_1 \boldsymbol{\varepsilon}_1 + \hat{\boldsymbol{\alpha}}_2 \hat{\boldsymbol{\alpha}}_1 \hat{\boldsymbol{\alpha}}_0 \boldsymbol{\Delta}_0 \\
\boldsymbol{\Delta}_4 &= \boldsymbol{\varepsilon}_4 + \hat{\boldsymbol{\alpha}}_3 \boldsymbol{\varepsilon}_3 + \hat{\boldsymbol{\alpha}}_3 \hat{\boldsymbol{\alpha}}_2 \boldsymbol{\varepsilon}_2 + \hat{\boldsymbol{\alpha}}_3 \hat{\boldsymbol{\alpha}}_2 \hat{\boldsymbol{\alpha}}_1 \boldsymbol{\varepsilon}_1 + \hat{\boldsymbol{\alpha}}_3 \hat{\boldsymbol{\alpha}}_2 \hat{\boldsymbol{\alpha}}_1 \hat{\boldsymbol{\alpha}}_0 \boldsymbol{\Delta}_0
\end{aligned}$$

and, in general,

$$\boldsymbol{\Delta}_n = \boldsymbol{\varepsilon}_n + \hat{\boldsymbol{\alpha}}_{n-1} \boldsymbol{\varepsilon}_{n-1} + \cdots + \hat{\boldsymbol{\alpha}}_{n-1} \hat{\boldsymbol{\alpha}}_{n-2} \cdots \hat{\boldsymbol{\alpha}}_2 \hat{\boldsymbol{\alpha}}_1 \boldsymbol{\varepsilon}_1 + \hat{\boldsymbol{\alpha}}_{n-1} \hat{\boldsymbol{\alpha}}_{n-2} \cdots \hat{\boldsymbol{\alpha}}_2 \hat{\boldsymbol{\alpha}}_1 \hat{\boldsymbol{\alpha}}_0 \boldsymbol{\Delta}_0$$

where

$$\hat{\boldsymbol{\alpha}}_k = \hat{\mathbf{I}} + h_k \hat{\mathbf{F}}_y\left(x_k, \boldsymbol{\xi}_k\right) \tag{4}$$

in which, for each k, $\boldsymbol{\xi}_k$ is an appropriate set of constants (as explained above).