General error propagation in the RKnGLm method

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

ABSTRACT

The RKnGLm method is a numerical method for solving initial value problems in ordinary differential equations of the form \( y' = f(x, y) \) 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 RKnGLm. We show that, for a suitable choice of \( r \) and \( m \), the global order of RKnGLm 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 \( f(x, y) \) 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 \( y' = f(x, y) \) 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 \( O(h^{r+1}) \) local errors results in a global error of \( O(h^r) \), 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 \cite{Prentice2006}, designated RKnGLm, 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 \( O(h^{r+1}) \) in its local error, then the associated RKnGLm method is \( O(h^{r+1}) \) in its global error, i.e. the global error in RKnGLm has the same order as the local RK error (for the benefit of the reader, a brief description of RKnGLm 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 RKnGLm method, (b) we show how the global error of RKnGLm achieves \( O(h^{r+1}) \) 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 \( f(x, y) \) 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 RKnGLm 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 \( \mathbf{M} \), denotes a \( q \times q \) matrix.

E-mail address: jprentice@uj.ac.za.
2.1. Explicit Runge–Kutta methods

We denote an explicit RK method for solving the \( q \)-dimensional system
\[
y' = f(x, y) \quad y(x_0) = y_0 \quad a \leq x \leq b
\]
by
\[
\mathbf{w}_{i+1} = \mathbf{w}_i + h_i \mathbf{F}(x_i, \mathbf{w}_i)
\]
where \( h_i = x_{i+1} - x_i \) is a stepsize, \( \mathbf{w}_i \) denotes the numerical approximation to \( y(x_i) \) and \( \mathbf{F}(x, \mathbf{y}) \) is a function associated with the particular RK method. Here and throughout the symbol \( \hat{\cdot} \) 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
\[
\Delta_i = \mathbf{w}_i - y_i,
\]
and the local error at \( x_i \) by
\[
e_{i+1} = \left[ y_i + h_i \mathbf{F}(x_i, y_i) \right] - y_{i+1}.
\]
In the above, \( y_i \) denotes the true solution \( y(x_i) \). Note the use of the exact value \( y_i \) 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 \( \Delta_0 \) exists in the initial value, although in most practical cases \( \Delta_0 = 0 \). We have
\[
\mathbf{w}_1 = y_0 + \Delta_0 + h_0 \mathbf{F}(x_0, y_0 + \Delta_0)
\Rightarrow \Delta_1 = \left[ y_0 + h_0 \mathbf{F}(x_0, y_0) - y_1 \right] + \left[ \hat{\mathbf{I}} + h_0 \hat{\mathbf{F}}_y(x_0, \xi_0) \right] \Delta_0
= \epsilon_1 + \alpha_0 \Delta_0
\]
where \( \alpha_0 \) has been implicitly defined. In the above we use the symbol \( \xi_0 \) in \( \hat{\mathbf{F}}_y(x_0, \xi_0) \) to simply denote an appropriate set of constants such that \( \hat{\mathbf{F}}_y(x_0, \xi_0) \Delta_0 \) is the residual term in the first-order Taylor expansion of \( \mathbf{F}(x, y_0 + \Delta_0) \). Moreover, \( \hat{\mathbf{F}}_y \) is the Jacobian
\[
\hat{\mathbf{F}}_y = \begin{bmatrix}
\frac{\partial F_1}{\partial y_1} & \cdots & \frac{\partial F_1}{\partial y_q} \\
\vdots & \ddots & \vdots \\
\frac{\partial F_q}{\partial y_1} & \cdots & \frac{\partial F_q}{\partial y_q}
\end{bmatrix}
\]
where \( \{F_1, F_2, \ldots, F_q\} \) are the components of \( \mathbf{F} \). The matrix \( \hat{\mathbf{I}} \) is the identity matrix.

For \( \Delta_2 \) we have
\[
\mathbf{w}_2 = \mathbf{w}_1 + h_1 \mathbf{F}(x_1, \mathbf{w}_1)
\Rightarrow \mathbf{y}_2 + \Delta_2 = \left[ y_1 + \Delta_1 \right] + h_1 \mathbf{F}(x_1, y_1 + \Delta_1)
= \left[ y_1 + \Delta_1 \right] + h_1 \mathbf{F}(x_1, y_1) + h_1 \hat{\mathbf{F}}_y(x_1, \xi_1) \Delta_1
\Rightarrow \Delta_2 = \left[ y_1 + h_1 \mathbf{F}(x_1, y_1) - y_2 \right] + \left[ \hat{\mathbf{I}} + h_1 \hat{\mathbf{F}}_y(x_1, \xi_1) \right] \Delta_1
= \epsilon_2 + \alpha_1 \Delta_1
= \epsilon_2 + \alpha_1 \epsilon_1 + \alpha_1 \alpha_0 \Delta_0.
\]
It is easy to show that
\[
\Delta_3 = \epsilon_3 + \alpha_2 \epsilon_2 + \alpha_2 \alpha_1 \epsilon_1 + \alpha_2 \alpha_0 \Delta_0
\Delta_4 = \epsilon_4 + \alpha_3 \epsilon_3 + \alpha_3 \alpha_2 \epsilon_2 + \alpha_3 \alpha_1 \epsilon_1 + \alpha_3 \alpha_0 \Delta_0
\]
and, in general,
\[
\Delta_n = \epsilon_n + \alpha_{n-1} \epsilon_{n-1} + \cdots + \alpha_{n-1} \alpha_{n-2} \cdots \alpha_2 \alpha_1 \epsilon_1 + \alpha_{n-1} \alpha_{n-2} \cdots \alpha_2 \alpha_1 \alpha_0 \Delta_0
\]
where
\[
\alpha_k = \hat{\mathbf{I}} + h_k \hat{\mathbf{F}}_y(x_k, \xi_k)
\]
in which, for each \( k, \xi_k \) is an appropriate set of constants (as explained above).
If \( \| h_{t_i} \tilde{F}_y (x_k, \xi_k) \| \) is small then \( \tilde{a}_k \approx \hat{a}_k \), and so

\[
\Delta_n \approx \Delta_0 + \sum_{j=1}^n \varepsilon_j
\]

but this is generally not expected to be the case, particularly if \( \tilde{F}_y (x_k, \xi_k) \) has large norm. Furthermore, if the \( \hat{a} \)'s have norm greater than one, then the term in \( \varepsilon_1 \) (or \( \Delta \) if it is nonzero) could make the most significant contribution to the global error.

The global error \( \Delta_n \) is the accumulation of these local errors, as in

\[
\Delta_n = \sum_{j=1}^n \beta_j h^{r+1} = \left( \frac{1}{n} \sum_{j=1}^n \beta_j \right) (nh') = \bar{\beta} (b - a) h'
\]

where \( \bar{\beta} \) is the vector of coefficients of the terms in \( \varepsilon_j \) (assuming that each component of \( \varepsilon_j \) is proportional to \( h^{r+1} \)).

2.4. Gauss–Legendre quadrature

Gauss–Legendre (GL) quadrature on the interval \([-1, 1]\) is given by [5]

\[
\int_{-1}^1 f(x) \, dx = \sum_{i=1}^m W_i f(\tilde{x}_i) + \frac{f^{(2m)}(\eta(x))}{(2m)!} \int_{-1}^1 \left( \prod_{i=1}^m (x - x_i) \right)^2 \, dx.
\]

(5)

Here, the \( m \) nodes \( \tilde{x}_i \) are the roots of the \( m \)th degree Legendre polynomial on the interval \([-1, 1]\), and \( W_i \) are appropriate weights. In the error term, \(-1 < \eta(x) < 1\). On an arbitrary interval \([u, v]\) GL quadrature is

\[
\int_u^v f(x) \, dx \approx \frac{(v - u)}{2} \sum_{i=1}^m W_i f(x_i) = h \sum_{i=1}^m C_i f(x_i)
\]

where

\[
C_i \approx \frac{(m + 1) W_i}{2}
\]

and \( h \) denotes the average length of the subintervals into which \([u, v]\) is subdivided by the nodes \( x_i \). We have used the symbol \( \tilde{x}_i \) for the nodes on \([u, v]\) to differentiate from the nodes \( x_i \) on \([-1, 1]\); indeed,

\[
x_i = \frac{1}{2} \left[ (v - u) \tilde{x}_i + u + v \right].
\]

In the remainder of this paper \( x_i \) will be used as a generic symbol for the nodes. We have referred to the interval \([-1, 1]\) above because the nodes \( \tilde{x}_i \) on this interval are extensively tabulated.

Consequently, GL quadrature for the vector function \( \mathbf{f}(x, y) \) is given by

\[
\int_u^v \mathbf{f}(x, y) \, dx = h \sum_{i=1}^m C_i \mathbf{f}(x_i, y) + O(h^{2m+1}).
\]

(6)

Regarding the order of the error term in (6), simply assume that \( x_i = u + \theta_i h \), where \( \theta_i \) is an appropriate constant associated with each \( x_i \), and that \( x = u + sh \) where \( s \) is a continuous variable (so that \( dx = hds \)). If we make these substitutions in the error term in (5), we find that the error in GL quadrature is \( O(h^{2m+1}) \).

2.5. The RKrGLm algorithm

We briefly describe the general RKrGLm algorithm on the interval \([a, b]\), with reference to Fig. 1.

Subdivide \([a, b]\) into \( n \) subintervals \( H_j \), where \( j = 1, 2, \ldots, n \). At the RK nodes we use RKr:

\[
\mathbf{w}_{i+1} = \mathbf{w}_i + h \mathbf{F}(x_i, \mathbf{w}_i)
\]

where \( i = (j - 1)p + (j - 1) + 1, \ldots, (j - 1)p + m - 1 \). At the GL nodes we use \( m \)-point GL quadrature:

\[
\mathbf{w}_{jp} = \mathbf{w}_{(j-1)p} + h \sum_{i=(j-1)p+1}^{(j-1)p+m} C_i \mathbf{f}(x_i, \mathbf{w}_i).
\]

Note that \( p \equiv m + 1 \).
The GL component is motivated by
\[
\int_{x_{(j-1)p}}^{x_j} f(x, y) \, dx = y_j - y_{(j-1)p} \approx h \sum_{i=(j-1)p+1}^{(j-1)p+m} C_i (x_i, y_i)
\]
\[
\Rightarrow y_j \approx y_{(j-1)p} + h \sum_{i=(j-1)p+1}^{(j-1)p+m} C_i (x_i, y_i).
\]

The RKrGLm algorithm has been shown to be consistent, convergent and zero-stable [6].

2.6. Implementation of RKrGLm

There are a few points regarding the implementation of RKrGLm that need to be discussed:

- If we merely sample the solutions at the GL nodes, treating the computations at the RK nodes as if they were the stages of an ordinary RK method, then RKrGLm would be reduced to an inefficient one-step method, involving at least \( rm \) stage evaluations. This is not the intention behind the development of RKrGLm; rather, RKrGLm represents an attempt to improve the efficiency of any RK method, simply by replacing the computation at every \((m + 1)\)th node by a quadrature formula which does not require evaluation of any of the stages in the underlying RK method.

- Of course, it is clear from the above that on \( H_1 \) the RK nodes are required to be consistent with the nodes necessary for GL quadrature. If, however, the RK nodes are located differently (perhaps due to a local error control mechanism, for example) then it is a simple matter to construct a Hermite interpolating polynomial of degree \( 2m + 1 \) (which has an error of order \( 2m + 2 \)) using the solutions at the nodes \( \{x_0, \ldots, x_m\} \). Then, assuming \( x_0 \) maps to \(-1\) and \( x_m \) maps to the largest Legendre polynomial root \( x \) on \([-1, 1] \), the position of the other nodes \( \{x_1, \ldots, x_{m-1}\} \) suitable for GL quadrature can be determined, and the Hermite polynomial can be used to find approximate solutions of order \( r + 1 \) at these nodes, thus facilitating the GL component of RKrGLm. A similar procedure is carried out on the next subinterval \( H_2 \), and so on.

- If the underlying RK method possesses a continuous extension it would not be necessary to construct the Hermite polynomial described above. However, there is no guarantee that a continuous extension of appropriate order (at least \( 2m = r + 1 \)) will be available, and it is generally true that determining a continuous extension for a RK method requires additional stages in the RK method, which would most likely compromise the gain in efficiency offered by RKrGLm. Note that the construction of the Hermite polynomial only requires one additional evaluation of \( f(x, y) \), at \( x_m \).

2.7. Local error at the GL nodes

The local error at the GL nodes is defined in a similar way to that for a one-step method:
\[
\int_{x_{(j-1)p}}^{x_j} f(x, y) \, dx = y_j - y_{(j-1)p} = h \sum_{i=(j-1)p+1}^{(j-1)p+m} C_i (x_i, y_i) + O\left(h^{2m+1}\right)
\]
\[
\Rightarrow e_j = \left[ y_{(j-1)p} + h \sum_{i=(j-1)p+1}^{(j-1)p+m} C_i (x_i, y_i) \right] - y_j = O\left(h^{2m+1}\right).
\]

where \( j = 1, 2, \ldots, n \).

3. Error propagation in the RKrGLm method

3.1. Error propagation

In this section we present a theorem that describes the global error at the GL nodes in terms of accumulated local errors. Before doing so, however, we introduce a convenient notation: consider the term
where $\delta_{i,p-1}$ is the Kronecker delta. Of course, when $i = p - 1$ the Kronecker delta ensures that $\hat{\psi}_{j,p-1} = \hat{1}$. The ‘underarrow’ in (7) means that the product is right-to-left with increasing $k$. This order of multiplication is important because the $\hat{\alpha}$’s are matrices. When $i = 1$ we have

$$\hat{\psi}_{jp-p+1} = \hat{1} + \sum_{t=1}^{p-2} \left( \sum_{k=1}^{p-2} \left( \prod_{k=1}^{p-2} \hat{\alpha}_k \right) \right), \quad i = 1, 2, \ldots, p - 1$$

when $i = 2$ we have

$$\hat{\psi}_{jp-p+2} = \hat{1} + \sum_{t=2}^{p-2} \left( \sum_{k=1}^{p-2} \left( \prod_{k=1}^{p-2} \hat{\alpha}_k \right) \right) = \hat{1} + \hat{\alpha}_{jp-p+2} + \hat{\alpha}_{jp-p+1} + \hat{\alpha}_{jp-p+2}$$

and when $i = p - 2$ we have

$$\hat{\psi}_{jp-2} = \hat{1} + \sum_{t=p-2}^{p-2} \left( \sum_{k=1}^{p-2} \left( \prod_{k=1}^{p-2} \hat{\alpha}_k \right) \right) = \hat{1} + \hat{\alpha}_{jp-2},$$

and similarly for other values of $i$.

We may now state and prove a theorem describing the global error at the GL nodes in terms of accumulated local errors. We assume here that the RK nodes are located as required for GL quadrature. The effect of the Hermite polynomial described previously will be considered later.

**Theorem 1.** For RKrGLm we have

$$\Delta_{np} = \sum_{j=1}^{n} \epsilon_{jp} + hA_{jp-p+1,jp-1} + h\hat{B}_{jp}\Delta_{jp-1}$$

where

$$A_{jp-p+1,jp-1} = \sum_{i=1}^{p-1} \hat{y}_{jp-p+i} e_i$$

$$\hat{y}_{jp-p+i} = C_{jp-p+i} \hat{F}_x (x_{jp-p+i}, \zeta_{jp-p+i}) \hat{\psi}_{jp-p+i}, \quad i = 1, \ldots, p - 1$$

$$\hat{B}_{jp} = C_{jp-p+1} \hat{F}_x (x_{jp-p+1}, \zeta_{jp-p+1}) \hat{\psi}_{jp-p+1} + \cdots + C_{jp-1} \hat{F}_x (x_{jp-1}, \zeta_{jp-1}) \hat{\alpha}_{jp-p} \hat{\alpha}_{jp-p-1} \cdots \hat{\alpha}_{jp-p}$$

$$= \sum_{t=1}^{p-1} \sum_{k=1}^{p-2} \hat{F}_x (x_{jp-p+t}, \zeta_{jp-p+t}) \left( \prod_{k=1}^{p-2} \hat{\alpha}_k \right)$$

$p = m + 1$

and all other symbols have been defined previously (the Jacobian $\hat{F}_x$ is defined in the same way as $\hat{F}_x$ in (3), and $\zeta$ is analogous to $\xi$ in (4)).

**Proof.** The proof is by induction on $n$, with $p$ fixed.

Consider $n = 1$. For the first $m (= p - 1)$ nodes we have

$$\Delta_1 = \epsilon_1 + \hat{\alpha}_0 \Delta_0, \quad \Delta_2 = \epsilon_2 + \hat{\alpha}_1 \epsilon_1 + \hat{\alpha}_1 \hat{\alpha}_0 \Delta_0$$

$$\vdots$$

$$\Delta_{p-1} = \epsilon_{p-1} + \hat{\alpha}_{p-2} \epsilon_{p-2} + \cdots + \hat{\alpha}_{p-2} \hat{\alpha}_{p-3} \cdots \hat{\alpha}_2 \hat{\alpha}_1 \epsilon_1 + \hat{\alpha}_{p-2} \hat{\alpha}_{p-3} \cdots \hat{\alpha}_2 \hat{\alpha}_1 \hat{\alpha}_0 \Delta_0$$

(10)
and so
\[
\mathbf{w}_p = y_p + \Delta_p \\
= y_0 + h \sum_{i=1}^{p-1} C_i f(x_i, w_i) \\
= y_0 + h \sum_{i=1}^{p-1} C_i f(x_i, y_i + \Delta_i) \\
= y_0 + h \sum_{i=1}^{p-1} C_i f(x_i, y_i) + h \sum_{i=1}^{p-1} C_i \hat{f}_j(x_i, \zeta_j) \Delta_i \\
= y_0 + h \sum_{i=1}^{p-1} C_i f(x_i, y_i) + h \sum_{i=1} \hat{y}_i e_i + h \hat{B}_p \Delta_0 \\
= y_0 + h \sum_{i=1} C_i f(x_i, y_i) + h \mathbf{A}_{1,p-1} + h \hat{B}_p \Delta_0
\]
\[
\Rightarrow \Delta_p = \left[ y_0 + h \sum_{i=1}^{p-1} C_i f(x_i, y_i) - y_4 \right] + h \mathbf{A}_{1,p-1} + h \hat{B}_p \Delta_0 \\
= \mathbf{e}_p + h \mathbf{A}_{1,p-1} + h \hat{B}_p \Delta_0.
\]

For the sake of detail, we also consider \( n = 2 \). We have, at the nodes \( \{x_{p+1}, \ldots, x_{2p-1}\} \)
\[
\Delta_{p+1} = \mathbf{e}_{p+1} + \hat{\alpha}_p \Delta_p \\
\vdots \\
\Delta_{2p-1} = \mathbf{e}_{2p-1} + \hat{\alpha}_{2p-2} \mathbf{e}_{2p-2} + \cdots + \hat{\alpha}_{2p-3} \mathbf{e}_{2p-3} + \hat{\alpha}_{p+1} \mathbf{e}_{p+1} + \hat{\alpha}_{2p-2} \hat{\alpha}_{2p-3} \cdots \hat{\alpha}_{p+1} \hat{\alpha}_p \Delta_p
\]
so that
\[
\mathbf{w}_{2p} = y_{2p} + \Delta_{2p} \\
= \mathbf{w}_p + h \sum_{i=p+1}^{2p-1} C_i f(x_i, w_i) \\
= y_p + \Delta_p + h \sum_{i=p+1}^{2p-1} C_i f(x_i, y_i + \Delta_i) \\
= y_p + \Delta_p + h \sum_{i=p+1}^{2p-1} \left[ C_i f(x_i, y_i) + C_i \hat{f}_j(x_i, \zeta_j) \Delta_i \right] \\
= y_p + h \sum_{i=p+1}^{2p-1} C_i f(x_i, y_i) + h \mathbf{A}_{p+1,2p-1} + h \hat{B}_{2p} \Delta_p + \Delta_p
\]
\[
\Rightarrow \Delta_{2p} = \left[ y_p + h \sum_{i=p+1}^{2p-1} C_i f(x_i, y_i) - y_{2p} \right] + h \mathbf{A}_{p+1,2p-1} + h \hat{B}_{2p} \Delta_p + \mathbf{e}_p + h \mathbf{A}_{1,3} + h \hat{B}_p \Delta_0 \\
= (\mathbf{e}_{2p} + \mathbf{e}_p) + \left( h \mathbf{A}_{p+1,2p-1} + h \mathbf{A}_{1,p-1} \right) + \left( h \hat{B}_{2p} \Delta_p + h \hat{B}_p \Delta_0 \right) \\
= \sum_{j=1}^{2p-1} \mathbf{e}_j + \sum_{j=p+1}^{2p-1} \mathbf{e}_j + h \mathbf{A}_{p+1,p-1} + h \hat{B}_{2p} \Delta_p + h \hat{B}_p \Delta_0.
\]

Now assume that (8) is true for \( n = N \), and consider \( n = N + 1 \):
\[
\mathbf{w}_{(N+1)p} = y_{(N+1)p} + \Delta_{(N+1)p} \\
= \mathbf{w}_{np} + h \sum_{i=np+1}^{(N+1)p-1} C_i f(x_i, w_i) \\
= y_{np} + \Delta_{np} + h \sum_{i=np+1}^{(N+1)p-1} C_i f(x_i, y_i + \Delta_i)
\[ \Delta_{(N+1)p} = \varepsilon_{(N+1)p} + hA_{(N+1)p} + h\hat{B}_{(N+1)p}\Delta_{Np} + \Delta_{Np} \]
\[ = \varepsilon_{(N+1)p} + hA_{(N+1)p} + h\hat{B}_{(N+1)p}\Delta_{Np} + \sum_{j=1}^{N} \varepsilon_{jp} + hA_{jp} + h\hat{B}_{jp}\Delta_{(j-1)p} \]
\[ = \sum_{j=1}^{N+1} \varepsilon_{jp} + hA_{jp} + r_{jp} + h\hat{B}_{jp}\Delta_{(j-1)p} \]

This completes the proof. \( \blacksquare \)

3.2. Error accumulation and the choice of \( r \) and \( m \)

In (8), the term \( \sum_{j=1}^{n} \varepsilon_{jp} \) is the sum of the local errors at the GL nodes, while \( \sum_{j=1}^{n} hA_{jp} + p_{j} - 1 \) is composed of local errors at the RK nodes. Hence, we have

\[ \sum_{j=1}^{n} \varepsilon_{jp} \propto \sum_{j=1}^{n} h^{2m+1} = nh^{2m+1} \]

\[ \sum_{j=1}^{n} hA_{jp} + p_{j} - 1 \propto \sum_{j=1}^{n} h^{r+2} = nh^{r+2} \]

and so

\[ \sum_{j=1}^{n} \varepsilon_{jp} + hA_{jp} + p_{j} - 1 = nD_{1}h^{2m+1} + nD_{2}h^{r+2} \]

\[ = \left( \frac{D_{1}}{p+1} \right) h^{2m} \left( (p+1) nh \right) + \left( \frac{D_{2}}{p+1} \right) h^{r+1} \left( (p+1) nh \right) \]

\[ = \left[ \frac{D_{1}(b-a)}{p+1} \right] h^{2m} + \left[ \frac{D_{2}(b-a)}{p+1} \right] h^{r+1} \]

\[ = O \left( h^{\min(2m, r+1)} \right) \]

where \( D_{1} \) and \( D_{2} \) are vectors of appropriate coefficients. If we choose \( r \) and \( m \) such that \( 2m \geq r+1 \), then the first two terms in (8) are proportional to \( h^{r+1} \). Consequently, the third term, which contains global errors at previous GL nodes, is \( O \left( h^{r+2} \right) \).

Hence, we have, for a suitable choice of \( r \) and \( m \),

\[ \Delta_{np} = O \left( h^{r+1} \right) . \]

As for writing the global error explicitly in terms of the local errors consider, for example, \( \Delta_{12} \) in RK5GL3 (\( r = 5, m = 3 \)). Using the above expressions, we have, in terms of the local errors \( \varepsilon_{i} \),

\[ \Delta_{12} = \sum_{i=1}^{12} \hat{G}_{i} \varepsilon_{i} \]  

(11)

where

\[ \hat{G}_{1} = \hat{\gamma}_{1} h + \left( \hat{B}_{12}\hat{\gamma}_{1} + \hat{B}_{8}\hat{\gamma}_{1} \right) h^{2} + \hat{B}_{12}\hat{B}_{8}\hat{\gamma}_{1} h^{3} \text{ for } i = 1, 2, 3 \]

\[ \hat{G}_{4} = \hat{I} + \left( \hat{B}_{12} + \hat{B}_{8} \right) h + \hat{B}_{8} h^{2} \]

\[ \hat{G}_{5} = \hat{\gamma}_{1} h + \hat{B}_{12}\hat{\gamma}_{1} h^{2} \text{ for } i = 5, 6, 7 \]

\[ \hat{G}_{8} = \hat{I} + \hat{B}_{12} h \]

\[ \hat{G}_{9} = \hat{\gamma}_{2} h \text{ for } i = 9, 10, 11 \]

\[ \hat{G}_{12} = \hat{I} . \]  

(12)

4. Comments

The mechanism for the \( O \left( h^{r+1} \right) \) global error in RKrGLm is shown in the first two terms on the rhs of (8). The first of these is the sum of the GL local errors. The second term is a linear combination of the RK local errors, multiplied by a factor \( h \). The effect of the GL component, then, is to increase the order of the accumulated RK local errors by one. We refer to this as a
“quenching” effect that occurs at the GL nodes, and it serves to prevent the accumulation of the RK local errors. The third term in (8) contains terms of higher order, as shown, for example, in the expansion of $\Delta_{12}$ via (11) and (12).

5. Effect of the Hermite interpolating polynomial

The use of a Hermite interpolating polynomial, as described previously, will not affect the order of RKGLm, but will affect the nature of the coefficients in (8). In this section we consider how these coefficients are affected by the use of a Hermite polynomial.

5.1. The Hermite interpolating polynomial

We give a brief description of Hermite interpolation. If the data $\{x_i, y_i, y_i': i = 0, \ldots, m\}$ are available, then a polynomial $\mathcal{H}(x)$, of degree at most $2m + 1$, with the interpolatory properties

$$\mathcal{H}(x_i) = y_i, \quad \mathcal{H}'(x_i) = y_i'$$

for each $i$, may be constructed. If the nodes $x_i$ are distinct, then $\mathcal{H}(x)$ is unique. This approximating polynomial is known as the Hermite interpolating polynomial [1], and has an approximation error given by

$$y(x) - \mathcal{H}(x) = \frac{y^{(2m+2)}(\xi(x))}{(2m+2)!} \prod_{i=0}^{m} (x-x_i)^2,$$

where $x_1 < \xi(x) < x_m$. If $h$ is the average separation of the nodes on $[x_0, x_m]$, it is possible to write $x - x_i = \sigma_i h$, where $\sigma_i$ is a suitable constant, and hence

$$y(x) - \mathcal{H}(x) = O\left(h^{2m+2}\right).$$

The algorithm for determining the coefficients of $\mathcal{H}(x)$ is linear, as in

$$c = A^{-1} d$$

where $c$ is a vector of the coefficients of $\mathcal{H}(x)$, $A$ is the relevant interpolation matrix, and $d$ is a vector containing $y_i$ and $y_i'$. The details of these terms will be described in the next subsection. Nevertheless, if an error $O(\Delta)$ exists in each of $y_i$ and $y_i'$, then an error of $O(\Delta)$ will exist in each component of $c$. Moreover, since $\mathcal{H}(x)$ is linear in its coefficients, then an error of $O(\Delta)$ will also exist in any computed value of $\mathcal{H}(x)$. Consequently, we may write

$$y(x) - \mathcal{H}(x) = O\left(h^{2m+2}\right) + O(\Delta)$$

where the $O(\Delta)$ term arises from errors in $y_i$ and $y_i'$. We have assumed, of course, that the errors in $y_i$ and $y_i'$ are of the same order, which is the situation that we encounter here, since for the derivative $y' = f(x, y)$ we have

$$f(x_i, w_i) = f(x_i, y_i + \Delta_i) = f(x_i, y_i) + \hat{f}_y(x_i, \theta_i) \Delta_i.$$

In the above we use the symbol $\theta_i$ in $\hat{f}_y(x_i, \theta_i) \Delta_i$ simply to denote an appropriate set of constants such that $\hat{f}_y(x_i, \theta_i) \Delta_i$ is the residual term in the first-order Taylor expansion of $f(x_i, y_i + \Delta_i)$. $\hat{f}_y$ is the Jacobian

$$\hat{f}_y = \begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \cdots & \frac{\partial f_1}{\partial y_q} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_q}{\partial y_1} & \cdots & \frac{\partial f_q}{\partial y_q} \end{bmatrix}$$

where $\{f_1, f_2, \ldots, f_q\}$ are the components of $f$, and $q$ is the dimension of the system (1). Clearly, a global error of $\Delta_i$ in $w_i$ implies an error of $O(\Delta_i)$ in the derivative $f(x_i, w_i)$.

Also, it is understood that, when dealing with a $q$-dimensional system of differential equations, it is possible to determine a Hermite polynomial for each component of the system. In such a context, $\mathcal{H}(x)$ will denote a $q \times 1$ vector of Hermite polynomials.

5.2. Errors due to the Hermite polynomial

We now study the effect that the Hermite polynomial has on the error coefficients in (8). We assume that the nodes $\{x_0, \ldots, x_{m-1}\}$ are not appropriately located for GL quadrature, and that a Hermite polynomial on $[x_0, x_m]$ has been determined using the nodes $\{x_0, \ldots, x_m\}$ and the approximate solutions $\{w_0, \ldots, w_m\}$ at these nodes. We will restrict our discussion to the first component of the system (1); the other components are similar.
The coefficients of the Hermite polynomial are determined from
\[ c = A^{-1}d \]
where
\[
c = \begin{bmatrix} c_{2m+1} & c_{2m} & \cdots & c_1 & c_0 \end{bmatrix}^T
\]
\[
A = \begin{bmatrix}
x_0^{2m+1} & x_0^{2m} & \cdots & x_0^1 & 1 \\
\vdots & \vdots & & \vdots & \vdots \\
x_m^{2m+1} & x_m^{2m} & \cdots & x_m^1 & 1 \\
(2m+1)x_0^{2m} & (2m)x_0^{2m-1} & \cdots & 1 & 0 \\
\vdots & \vdots & & \vdots & \vdots \\
(2m+1)x_m^{2m} & (2m)x_m^{2m-1} & \cdots & 1 & 0
\end{bmatrix}
\]
and
\[
d = \begin{bmatrix}
w_{1,0} \\
w_{1,m} \\
f_1(x_0, w_{1,0}) \\
f_1(x_m, w_{1,m}) \\
\end{bmatrix} = \begin{bmatrix}
y_{1,0} \\
y_{1,m} \\
f_1(x_0, y_{1,0}) \\
f_1(x_m, y_{1,m}) \\
\end{bmatrix} + \begin{bmatrix}
\Delta_{1,0} \\
\Delta_{1,m} \\
O(\Delta_{1,0}) \\
O(\Delta_{1,m}) \\
\end{bmatrix} + \begin{bmatrix}
d_{\text{true}} \\
d_{\text{err}} \\
\end{bmatrix}
\]
where \( w_{1,0} \) is the approximate solution of the first component of the system at \( x_0 \), with similar definitions for \( w_{1,m}, y_{1,0}, y_{1,m}, \Delta_{1,0} \) and \( \Delta_{1,m} \). Since the solutions at \( \{x_0, \ldots, x_m\} \) have been determined by the RK method, the errors \( \{\Delta_{1,0}, \ldots, \Delta_{1,m}\} \) are given by the first component in each of (9) and (10). The second term on the rhs of (13), \( d_{\text{err}} \), when multiplied by \( A^{-1} \), constitutes the error in the Hermite polynomial coefficients due to errors in the approximate solutions at \( \{x_0, \ldots, x_m\} \).

The important point to note in (14) is that the entries in \( d_{\text{err}} \) are all linear combinations of local errors \( \varepsilon \), so that \( \Delta^*_{1,1} \) is also a linear combination of local errors. Due to the choice of \( r \) and \( m \), the \( O(h^{2m+2}) \) term is of higher order than the RK local order \( (r+1) \). The result is similar for the other nodes \( \{x^*_2, \ldots, x^*_m\} \), and also for the other components of the system. The only effect that the use of a Hermite polynomial has on \( \Delta_{\text{up}} \) in (8) is to alter the coefficients \( A_{jp-p+1,jp-1} \) and \( B_{jp} \); however, the results pertaining to error accumulation are unchanged, and so we still expect \( \Delta_{\text{up}} \) to be \( O(h^{r+1}) \), even when Hermite polynomials are used.

6. Numerical examples

By way of an example, we consider RK5GL3 (i.e. \( r = 5, m = 3 \) with \( r = 2m \)) and solve the one-dimensional test problem
\[
y' = \frac{y}{4} \left(1 - \frac{y}{20}\right)
\]
on $[0, 5]$ with $y(0) = 1$. This equation has solution

$$y(x) = \frac{20}{1 + 19e^{-x/4}}$$

and is one of the test problems used in [4]. The fifth-order Runge–Kutta method (RK5) used here is due to Fehlberg, as described in [3]. The global error is shown in Fig. 2. In the upper plot the RK nodes are consistent with the nodes required for GL quadrature. In the lower plot, the RK nodes are equispaced and the RK5GL3 employs a Hermite polynomial, as described previously. The accumulation of error in RK5 is clear, whereas the error quenching in RK5GL3 is also apparent. The quenching effect occurs at each of the GL nodes, where there is clearly a sharp reduction in the magnitude of the error. In between the GL nodes the error accumulates, as expected of the RK5 method. When the RK nodes are equispaced, the RK error on the first subinterval $H_1$ is less than that of RK5GL3, but thereafter the superior order of RK5GL3 becomes apparent.

For an example of a system, we use RK2GL2 ($r = 2$, $m = 2$ and $r + 1 < 2m$) to solve

$$\begin{align*}
y_1' &= y_2 \\
y_2' &= e^{2x} \sin x - 2y_1 + 2y_2 \\
y_1(0) &= -\frac{2}{5}, \quad y_2(0) = -\frac{3}{5}
\end{align*}$$

on $[0, 1]$. Although this system is linear, it provides a clear demonstration of the error quenching effect. The solution to this system is

$$\begin{align*}
y_1 &= \frac{1}{5} e^{2x} (\sin x - 2 \cos x) \\
y_2 &= \frac{1}{5} e^{2x} (4 \sin x + 3 \cos x).
\end{align*}$$

The second-order RK method used in RK2GL2 is the well-known trapezoidal method [2]. Global error curves for $y_1$ and $y_2$ are shown in Fig. 3. They are qualitatively similar to those in Fig. 2, and the error quenching effect is apparent. As in the upper plot in Fig. 2, for the sake of comparison, we have used the RK2GL2 node distribution in obtaining the RK2 solution.

### 7. Conclusion

We have considered the propagation of local error in the RK$r$GL$m$ method for IVPs, and presented expressions for the global error in terms of accumulated local errors. We have shown that, for a suitable choice of $r$ and $m$, the global error in RK$r$GL$m$ is of order $r + 1$. The mechanism for this increased order is seen to be a “quenching” effect introduced by the GL component of the algorithm. Two numerical examples have demonstrated this effect.
Fig. 3. Error curves for RK2GL2 and RK2 for the components $y_1$ and $y_2$ of the test system.

References