Relative Global Error Control in the RKQ Algorithm for Systems of Ordinary Differential Equations

J. S. C. Prentice

Department of Applied Mathematics, University of Johannesburg

P.O. Box 524, Auckland Park, 2006, South Africa

Tel: 27-115-593-145 E-mail: jprentice@uj.ac.za

Received: May 5, 2011 Accepted: June 8, 2011 Published: November 1, 2011

doi:10.5539/jmr.v3n4p59 URL: http://dx.doi.org/10.5539/jmr.v3n4p59

Abstract

We generalize the RKrvQz algorithm to solve nonstiff initial-value problems in ordinary differential equations. The algorithm can now be applied to systems of nonstiff initial-value problems (IVPs) in ordinary differential equations, and both relative error and absolute error can be controlled, locally and globally. We demonstrate the algorithm by solving the simple harmonic oscillator for moderate and strict tolerances.

Keywords: Runge-Kutta, Initial-value problem, Local error, Global error, Local extrapolation, Quenching, Relative error

1. Introduction

Recently, we described the RKrvQz algorithm (Prentice, 2011) for solving nonstiff initial-value problems (IVPs) in ordinary differential equations (ODEs). This algorithm uses three explicit Runge-Kutta (RK) methods, of orders r, v and z, to control both local and global errors in a stepwise manner. In that paper, we considered control of absolute error only, neglecting relative error control, and we considered the application of the algorithm to scalar problems, rather than systems of ODEs. The motivation for developing RKrvQz is that, in local extrapolation, the RKv solution is not only used to estimate the local error in the RKr solution, but the RKv solution is propagated in the RKr method. Any global error in the RKv solution is thus also propagated in the resulting RKr solution. This global error can accumulate to the point where the RKr solution is globally inaccurate (relative to some desired level of accuracy), even though its local error has been controlled. RKrvQz represents an attempt to control the global error in the RKr solution in a stepwise manner, i.e. as the RK iteration proceeds, rather than using an 'after-the-fact' reintegration procedure.

In the current paper, we consider RKrvQz applied to problems of the form

$$\begin{aligned}
\mathbf{y}' &\equiv \begin{bmatrix} \frac{dy_1}{dx} \\ \frac{dy_2}{dx} \\ \vdots \\ \frac{dy_n}{dx} \end{bmatrix} = \begin{bmatrix} f_1(x, y_1, y_2, \ldots, y_n) \\ f_2(x, y_1, y_2, \ldots, y_n) \\ \vdots \\ f_n(x, y_1, y_2, \ldots, y_n) \end{bmatrix} \equiv \mathbf{f}(x, \mathbf{y}) \\
x &\in \left[x_0, x_f\right] \\
\mathbf{y}(x_0) &= \mathbf{y}_0,
\end{aligned}$$

that is to say, systems of ODEs, and we discuss the inclusion of relative error control in the algorithm.

2. Relevant Concepts, Terminology and Notation

The current paper is based almost entirely on our previous work (Prentice, 2011), but for ease of reference we will present here concepts, notation and terminology relevant to our discussion as it pertains to systems of ODEs. Also, we will designate our previous paper as PR1, since we will refer to it several more times in the text. To a large extent, the remainder of this section is excerpted from PR1, with appropriate modifications. Throughout the remainder of the paper, quantities in normal font are scalars, and quantities in boldface font are $n \times 1$ vectors, except α and $\mathbf{F}_y^r$, which are $n \times n$ matrices. Additionally, we refer the reader to Hairer et al (2000), Butcher (2003), Iserles (2009), Kincaid & Cheney (2002), LeVeque (2007), and many references therein, for discussions of Runge-Kutta methods.

2.1 Runge-Kutta methods

The most general definition of a Runge-Kutta (RK) method for systems is

$$\begin{aligned} \mathbf{k}_p &= \mathbf{f}\left(x_i + c_p h_i, \mathbf{w}_i + h_i \sum_{q=1}^{m} a_{pq}\mathbf{k}_q\right) \qquad p = 1, 2, ..., m \\ \mathbf{w}_{i+1} &= \mathbf{w}_i + h_i \sum_{p=1}^{m} b_p \mathbf{k}_p \equiv \mathbf{w}_i + h_i \mathbf{F}\left(x_i, \mathbf{w}_i\right). \end{aligned} \tag{1}$$

Such a method is said to have m stages (the $\mathbf{k}_q$), and each stage is an $n \times 1$ vector. If $a_{pq} = 0$ for all $p \leqslant q$, then the method is said to be *explicit*; otherwise, it is known as an *implicit* RK method. As indicated earlier, we will focus our attention on explicit methods. The number of stages is related to the order r of the method, and for explicit methods we always have $r \leqslant m$. In the second line of (1), we have implicitly defined the function $\mathbf{F}$ (of course, $\mathbf{F}(x_i, \mathbf{w}_i)$ is an $n \times 1$ vector). The symbol $\mathbf{w}$ is used here and throughout to indicate the approximate numerical solution, whereas the symbol $\mathbf{y}$ will be used to denote the exact solution. As a refinement to our notation, we will denote a Runge-Kutta method of order r as RKr and, for such a method, we write

$$\mathbf{w}^r_{i+1} = \mathbf{w}^r_i + h_i \mathbf{F}^r\left(x_i, \mathbf{w}^r_i\right). \tag{2}$$

We may regard RKr as being defined by $\mathbf{F}^r$, although it is understood that, for any r, there are, generally speaking, numerous possible choices for $\mathbf{F}^r$. We denote the jth component of $\mathbf{F}^r$ by F^r_j. The superscripts in (2) are labels, not exponents. The stepsize h_i is given by

$$h_i \equiv x_{i+1} - x_i$$

and carries the subscript because it may vary from step to step.

Note that

$$\mathbf{w}^r_i \equiv \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,n} \end{bmatrix},$$

wherein the first index in the subscript indicates iteration number, and the second index indicates component.

2.2 Error propagation

We define the *global error* in a numerical solution generated by RKr at x_{i+1} by

$$\mathbf{\Delta}^r_{i+1} \equiv \mathbf{w}^r_{i+1} - \mathbf{y}_{i+1}, \tag{3}$$

and the *local error* at x_{i+1} by

$$\varepsilon^r_{i+1} \equiv \left[\mathbf{y}_i + h_i \mathbf{F}^r\left(x_i, \mathbf{y}_i\right)\right] - \mathbf{y}_{i+1}. \tag{4}$$

Note the use of the exact value $\mathbf{y}_i$ in the bracketed term in (4). Again, the superscripts are labels. The errors $\mathbf{\Delta}^r_{i+1}$ and ε^r_{i+1} are $n \times 1$ vectors.

We have previously shown (Prentice, 2009) that

$$\begin{aligned} \mathbf{\Delta}^r_{i+1} &= \varepsilon^r_{i+1} + \alpha^r_i \mathbf{\Delta}^r_i && (5) \\ \alpha^r_i &\equiv \mathbf{I}_n + h_i \mathbf{F}^r_y\left(x_i, \xi_i\right), && (6) \end{aligned}$$

where

$$\mathbf{F}^r_y\left(x_i, \xi_i\right) \equiv \begin{bmatrix} \frac{\partial F^r_1}{\partial y_1} & \cdots & \frac{\partial F^r_1}{\partial y_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F^r_n}{\partial y_1} & \cdots & \frac{\partial F^r_n}{\partial y_n} \end{bmatrix}_{(x_i,\xi_i)}$$

is the Jacobian of $\mathbf{F}^r(x, \mathbf{y})$ evaluated at (x_i, ξ_i), and ξ_i is a vector of constants arising in the residual term of a first-order Taylor expansion of $\mathbf{F}^r\left(x_i, \mathbf{w}^r_i\right)$ about the point $(x_i, \mathbf{y}_i)$; see (Prentice, 2009) for detail. Equation (5) provides a "master" relationship between local and global errors in RKr. We will assume that $\mathbf{\Delta}_0 = 0$ (i.e. the initial value is known exactly). We see that the global error at any node x_{i+1} is the sum of a local error term and a term incorporating the global error at the previous node. For RKr, it is known that

$$\begin{aligned} \varepsilon^r_{i+1} &\propto h^{r+1}_i \\ \mathbf{\Delta}^r_{i+1} &\propto h^r. \end{aligned}$$

On the RHS of these expressions, the superscripts are exponents, and h is a parameter representative of the stepsizes h_i.

2.3 Local error control via local extrapolation

Consider two RK methods of order r and order v, i.e. RKr and RKv, with $r < v$. Let $\mathbf{w}_{i+1}^r$ denote the approximate solution at x_{i+1} obtained with RKr, and similarly for $\mathbf{w}_{i+1}^v$. Let the local error at x_{i+1} in the RKr method be given by the $n \times 1$ vector $\varepsilon_{i+1}^r = \beta_{i+1}^r h_i^{r+1}$, and similarly for $\varepsilon_{i+1}^v = \beta_{i+1}^v h_i^{v+1}$ (which defines the *local error coefficients* $\beta_{i+1}^r, \beta_{i+1}^v$). Now, if $\mathbf{w}_i^r, \mathbf{w}_i^v = \mathbf{y}_i$, which means that $\mathbf{\Delta}_i^r, \mathbf{\Delta}_i^v = 0$, we have

$$\begin{aligned}
\mathbf{w}_{i+1}^r - \mathbf{w}_{i+1}^v &= \mathbf{y}_{i+1} + \mathbf{\Delta}_{i+1}^r - \left(\mathbf{y}_{i+1} + \mathbf{\Delta}_{i+1}^v\right) \\
&= \varepsilon_{i+1}^r + \alpha_i^r \mathbf{\Delta}_i^r - \left(\varepsilon_{i+1}^v + \alpha_i^v \mathbf{\Delta}_i^v\right) \\
&= \varepsilon_{i+1}^r - \varepsilon_{i+1}^v \\
&= \beta_{i+1}^r h_i^{r+1} - \beta_{i+1}^v h_i^{v+1} \\
&\approx \beta_{i+1}^r h_i^{r+1}
\end{aligned}$$

if h_i is sufficiently small (since $r < v$). This gives

$$\beta_{i+1}^r \approx \frac{\mathbf{w}_{i+1}^r - \mathbf{w}_{i+1}^v}{h^{r+1}}. \tag{7}$$

Once we have estimated the local error, we can perform error control. Assume that we require that the local error at each step must be less than a user-defined tolerance δ (we will say more about the nature of this tolerance later). Moreover, assume that, using stepsize h_i, we find for the jth component of ε_{i+1}^r,

$$\left|\varepsilon_{i+1,j}^r\right| = \left|\beta_{i+1,j}^r h_i^{r+1}\right| > \delta. \tag{8}$$

In other words, the magnitude of the local error $\varepsilon_{i+1,j}^r$ exceeds the desired tolerance. We remedy the situation by determining a new stepsize $h_{i,j}^*$ from

$$\left|\beta_{i+1,j}^r \left(h_{i,j}^*\right)^{r+1}\right| = \delta \Rightarrow h_{i,j}^* = \left(\frac{\delta}{\left|\beta_{i+1,j}^r\right|}\right)^{\frac{1}{r+1}} \tag{9}$$

and we repeat the RK computation with this new stepsize.

We would carry out this process of determining a new stepsize for each component of ε_{i+1}^r that exceeds δ in magnitude; each such process would yield a stepsize $h_{i,j}^*$; we would then choose

$$h_i^* = \min\left\{h_{i,j}^*\right\}.$$

Often, we introduce a so-called ‘safety factor’ σ, as in

$$h_i^* \to \sigma h_i^*,$$

where $\sigma < 1$, so that the new stepsize is slightly smaller than that given by (9). This is an attempt to cater for the possibility that β_{i+1}^r may have been underestimated, due to the assumptions made in deriving (7). The choice of the value of σ is subjective, although a representative value is 0.8.

Hence, we have

$$\begin{aligned}
x_{i+1} &= x_i + h_i^* \\
\mathbf{w}_{i+1}^r &= \mathbf{w}_i^r + h_i^* \mathbf{F}^r\left(x_i, \mathbf{w}_i^r\right).
\end{aligned} \tag{10}$$

On the other hand, if we find that the estimated error does not exceed the tolerance in any component, then no stepsize adjustment is necessary, and we proceed directly to the next step, using the already existing value of the stepsize.

Furthermore, since the higher-order solution $\mathbf{w}_i^v$ is available, we use $\mathbf{w}_i^v$ (in place of $\mathbf{w}_i^r$ in (10)) as input to generate both $\mathbf{w}_{i+1}^r$ (using RKr), and $\mathbf{w}_{i+1}^v$ (using RKv). In other words, we are assuming that $\mathbf{w}_i^v$ is accurate enough, relative to $\mathbf{w}_i^r$, to be regarded as the exact value - an assumption entirely consistent with the assumption made in deriving (7). This means that we determine the higher-order solution at each node, and this solution is used as input for both RK methods in computing solutions at the next node. This form of local error control is known as *local extrapolation,* and we denote this algorithm by RKrv.

2.4 Absolute and relative error control

If the tolerance δ in (8) is a constant, then the form of error control is absolute, i.e. we are demanding that $\left|\varepsilon^r_{i+1,j}\right|$ must be less than some absolute limit δ. Alternatively, we could demand

$$\frac{\left|\varepsilon^r_{i+1,j}\right|}{\left|w_{i+1,j}\right|} \leqslant \delta \Rightarrow \left|\varepsilon^r_{i+1,j}\right| \leqslant \delta \left|w_{i+1,j}\right|.$$

This means that we require $\left|\varepsilon^r_{i+1,j}\right|$ to be less than some limit, *relative* to the magnitude of $w_{i+1,j} \approx y_{i+1,j}$. Of course, if $\left|w_{i+1,j}\right|$ is close to zero, then the corresponding stepsize $h^*_{i,j}$ will be very small, and if $\left|w_{i+1,j}\right| = 0$, then $h^*_{i,j}$ cannot be computed at all. To counteract this possibility, we actually demand

$$\left|\varepsilon^r_{i+1,j}\right| \leqslant \max\left\{\delta_A, \delta_R \left|w_{i+1,j}\right|\right\} \equiv \delta_{i+1,j} \tag{11}$$

where δ_A and δ_R are known as the *absolute* and *relative* tolerances, respectively, and δ_{i+1} denotes a *node-dependent* tolerance. We then have two cases:

$$\begin{aligned} \delta_A &< \delta_R \left|w_{i+1,j}\right| \Rightarrow \left|\varepsilon^r_{i+1,j}\right| \leqslant \delta_R \left|w_{i+1,j}\right| \\ \delta_A &> \delta_R \left|w_{i+1,j}\right| \Rightarrow \left|\varepsilon^r_{i+1,j}\right| \leqslant \delta_A \end{aligned}$$

Obviously, if $\delta_A = \delta_R \left|w_{i+1,j}\right|$, then the two cases are equivalent. Often, we use $\delta_A = \delta_R$.

There is a good practical reason for implementing relative error control. If $\left|w_{i+1,j}\right| > 1$, then (9) gives, with $\delta_A = \delta_R$,

$$h^*_{i,j} = \left(\frac{\delta_R \left|w_{i+1,j}\right|}{\left|\beta^r_{i+1,j}\right|}\right)^{\frac{1}{r+1}} > \left(\frac{\delta_R}{\left|\beta^r_{i+1,j}\right|}\right)^{\frac{1}{r+1}}.$$

So $h^*_{i,j}$ determined in the relative sense is larger than $h^*_{i,j}$ determined in the absolute sense, particularly if $\left|w_{i+1,j}\right| \gg 1$. This implies that fewer nodes x_i would be required on $\left[x_0, x_f\right]$, resulting in a more efficient algorithm.

In (11), we have defined the node-dependent tolerance $\delta_{i+1,j}$. The local error control algorithm described in section 2.3 is easily adapted to cater for relative error control simply by replacing δ with $\delta_{i+1,j}$. Since $\delta_{i+1,j}$ can differ for each component of $\mathbf{w}_{i+1}$, we can write this tolerance as a vector

$$\delta_{i+1} = \begin{bmatrix} \max\left\{\delta_A, \delta_R \left|w_{i+1,1}\right|\right\} \\ \max\left\{\delta_A, \delta_R \left|w_{i+1,2}\right|\right\} \\ \vdots \\ \max\left\{\delta_A, \delta_R \left|w_{i+1,n}\right|\right\} \end{bmatrix}.$$

Consequently, in the next section, when we write

$$\left|\varepsilon^r_{i+1}\right| <_c \delta_{i+1}, \tag{12}$$

we mean that each component of $\left|\varepsilon^r_{i+1}\right|$ is less than the corresponding component of δ_{i+1}, and the notation

$$\left|\varepsilon^r_{i+1}\right| >_c \delta_{i+1} \tag{13}$$

means that there is at least one component of $\left|\varepsilon^r_{i+1}\right|$ that is greater than its corresponding component of δ_{i+1}.

3. The RK*rv*Q*z* Algorithm, with Absolute and Relative Error Control, for Systems of ODEs

We now describe the RK*rv*Q*z* algorithm for systems, incorporating relative error control. This is a generalization of the algorithm presented in PR1, and we will be economical in our discussion; the reader is referred to PR1 for detail.

We have three RK methods (RK*r*, RK*v* and RK*z*) at our disposal, with $r < v \ll z$. Let $\mathbf{w}^v_{i+1}$ denote the approximate solution at x_{i+1} obtained with RK*v*, and similarly for $\mathbf{w}^z_{i+1}$. Let $\mathbf{w}^{rv}_{i+1}$ denote the approximate solution at x_{i+1} obtained with RK*r*, using $\mathbf{w}^v_i$ as its input.

We have, using (3) and (5),

$$\begin{aligned}
\mathbf{w}^{rv}_{i+1} &= \mathbf{y}_{i+1} + \varepsilon^{r}_{i+1} + \alpha^{rv}_{i}\mathbf{\Delta}^{v}_{i} \\
\mathbf{w}^{v}_{i+1} &= \mathbf{y}_{i+1} + \varepsilon^{v}_{i+1} + \alpha^{v}_{i}\mathbf{\Delta}^{v}_{i} \\
\mathbf{w}^{z}_{i+1} &= \mathbf{y}_{i+1} + \varepsilon^{z}_{i+1} + \alpha^{z}_{i}\mathbf{\Delta}^{z}_{i}
\end{aligned}$$

which gives

$$\mathbf{w}^{rv}_{i+1} - \mathbf{w}^{z}_{i+1} = \varepsilon^{r}_{i+1} + \alpha^{rv}_{i}\mathbf{\Delta}^{v}_{i} - \left(\varepsilon^{z}_{i+1} + \alpha^{z}_{i}\mathbf{\Delta}^{z}_{i}\right) \approx \varepsilon^{r}_{i+1} + \alpha^{rv}_{i}\mathbf{\Delta}^{v}_{i} \tag{14}$$

since $r \ll z$. The notation α^{rv}_{i} in the above simply indicates that $\mathbf{w}^{v}_{i}$ has been used as input for RKr, and the form of α^{rv}_{i} is no different from α^{r}_{i} in (6).

Local error control via local extrapolation, as described in sections 2.3 and 2.4, is carried out using $\mathbf{w}^{rv}_{i+1}$ and $\mathbf{w}^{v}_{i+1}$ (i.e. with RKrv), yielding a stepsize h^{*}_{i}. Note that here h^{*}_{i} is determined with (9), so there is at least one component j such that

$$\left|\beta^{r}_{i+1,j}\left(h^{*}_{i}\right)^{r+1}\right| = \delta_{i+1,j}.$$

If it so happens that a new stepsize is not necessary then, for the purposes of what follows, we set $h^{*}_{i} = h_{i-1}$. Assuming now that a safety factor σ has been used, so that the new stepsize is σh^{*}_{i}, we have the solutions $\left\{\mathbf{w}^{rv}_{i+1}, \mathbf{w}^{rz}_{i+1}, \mathbf{w}^{v}_{i+1}, \mathbf{w}^{z}_{i+1}\right\}$ at $x_{i+1} = x_{i} + \sigma h^{*}_{i}$, and

$$\left|\varepsilon^{r}_{i+1}\right| = \left|\beta^{r}_{i+1}\left(\sigma h^{*}_{i}\right)^{r+1}\right| <_{c} \delta_{i+1}. \tag{15}$$

The LHS of (15) is the 'new' local error, arising from the use of σh^{*}_{i} as the stepsize. Now, if

$$\left|\mathbf{\Delta}^{rv}_{i+1}\right| \equiv \left|\varepsilon^{r}_{i+1} + \alpha^{rv}_{i}\mathbf{\Delta}^{v}_{i}\right| \leqslant_{c} \delta_{i+1}, \tag{16}$$

where the LHS is known from (14), then global error control is not necessary, and we proceed with the next RK iteration. However, if

$$\left|\mathbf{\Delta}^{rv}_{i+1}\right| >_{c} \delta_{i+1}, \tag{17}$$

then it means that at least one component of $\mathbf{\Delta}^{v}_{i}$ has become unacceptably large. We respond by setting

$$\mathbf{w}^{v}_{i} = \mathbf{w}^{z}_{i}$$

since

$$\mathbf{w}^{z}_{i} = \mathbf{y}_{i} + \varepsilon^{z}_{i} + \alpha^{z}_{i-1}\mathbf{\Delta}^{z}_{i-1} \approx \mathbf{y}_{i},$$

because RKz is of much higher order than RKv, and we recalculate $\mathbf{w}^{rv}_{i+1}$ and $\mathbf{w}^{v}_{i+1}$ using h^{*}_{i}. In other words, $\mathbf{w}^{v}_{i}$ is replaced with a much more accurate value. This will yield

$$\begin{aligned}
\mathbf{w}^{rv}_{i+1} &= \varepsilon^{r}_{i+1} + \alpha^{rz}_{i}\mathbf{\Delta}^{z}_{i} \approx \varepsilon^{r}_{i+1} \\
\mathbf{w}^{v}_{i+1} &= \varepsilon^{v}_{i+1} + \alpha^{vz}_{i}\mathbf{\Delta}^{z}_{i} \approx \varepsilon^{v}_{i+1},
\end{aligned}$$

so that $\mathbf{w}^{rv}_{i+1}$ and $\mathbf{w}^{v}_{i+1}$ will now have relatively small global error $\left(\propto \mathbf{\Delta}^{z}_{i}\right)$ accumulated from previous iterations. We have referred to this process as *quenching* in PR1.

Note that the safety factor σ ensures that $\left|\varepsilon^{r}_{i+1}\right|$ is *strictly* less than δ_{i+1} in (15), so that the global error component $\alpha^{rv}_{i}\mathbf{\Delta}^{v}_{i}$ can be accommodated somewhat. The extent of this accommodation is determined by σ^{r+1}. Say $\sigma = 0.8$ and $r = 3$, so that $\sigma^{r+1} = 0.41$. Then, assuming that all components of $\beta^{r}_{i+1}\left(\sigma h^{*}_{i}\right)^{r+1}$ and $\alpha^{rv}_{i}\mathbf{\Delta}^{v}_{i}$ have the same sign, $\alpha^{rv}_{i}\mathbf{\Delta}^{v}_{i}$ can be accommodated up to a magnitude of $0.59\delta_{i+1}$, before quenching is needed.

Finally, we emphasize that, at each node x_{i+1}, it is $\mathbf{w}^{rv}_{i+1}$ that is presented as the solution to the IVP, since this is the numerical solution for which both local and global error control has been implemented.

4. Comments

(1) In PR1 we also considered the computation of

$$\mathbf{w}^{rz}_{i+1} = \mathbf{y}_{i+1} + \varepsilon^{r}_{i+1} + \alpha^{rz}_{i}\mathbf{\Delta}^{v}_{i},$$

which enables

$$\mathbf{w}^{rv}_{i+1} - \mathbf{w}^{rz}_{i+1} = \mathbf{y}_{i+1} + \varepsilon^{r}_{i+1} + \alpha^{rv}_{i}\mathbf{\Delta}^{v}_{i} - \left(\mathbf{y}_{i+1} + \varepsilon^{r}_{i+1} + \alpha^{rz}_{i}\mathbf{\Delta}^{v}_{i}\right) \approx \alpha^{rv}_{i}\mathbf{\Delta}^{v}_{i}$$

to be estimated. This is merely a refinement, and is not necessary for the implementation of RKrvQz; furthermore, it does require additional computational effort. However, such an estimate of $\alpha_i^{rv}\mathbf{\Delta}_i^v$ is likely to be very good, since $v \ll z$, and together with (14), will provide a very good estimate of ε_{i+1}^r, which may be more reliable than the estimate of ε_{i+1}^r obtained from $\mathbf{w}_{i+1}^{rv} - \mathbf{w}_{i+1}^v$ (the estimate of ε_{i+1}^r is, of course, required for local error control via RKrv).

(2) We have used the same tolerance δ_{i+1} for both local and global error control. It is not necessary to do so although, since global error is, roughly speaking, an accumulation of local error, it is probably wise to ensure that the global tolerance is not smaller than the local tolerance.

(3) The algorithm presented in PR1 is obtained from the current algorithm when $\delta_R = 0$ and $n = 1$.

5. Numerical Example

A particularly suitable example is the simple harmonic oscillator

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= -y_1 \\ \mathbf{y}(0) &= \begin{bmatrix} 0 \\ 1000 \end{bmatrix} \end{aligned} \tag{18}$$

which has solution

$$\begin{aligned} y_1(x) &= 1000 \sin x \\ y_2(x) &= 1000 \cos x. \end{aligned}$$

Since the solution is oscillatory, there will be occasions when the solution is close to zero, which requires absolute error control, as per (14). Also, since the maximum magnitude of the solution is 1000, there will be occasions when relative error control is required.

We use the explicit methods RK3, RK4 and RK8, as referenced in PR1. We solve (18) on $x \in [0, 20]$ with RK34 (local extrapolation only), and RK34Q8 (local extrapolation with global quenching), for the cases $\delta_A = \delta_R = 10^{-5}$ and $\delta_A = \delta_R = 10^{-10}$. We refer to these cases as Case I and Case II, respectively. For both cases we use $\delta = 0.8$. We show some performance parameters in Table 1. In this table we show the maximum global error in each component of (18) and the number of quenches required in each case. Note that here the magnitude of the global error $|\Delta_i|$ is calculated as

$$|\Delta_i| = \begin{cases} \left|\frac{y_i - w_i}{y_i}\right| & \text{if} \quad |y_i| > 1 \\ |y_i - w_i| & \text{if} \quad |y_i| \leqslant 1 \end{cases}.$$

It is clear that RK34, despite the implementation of local extrapolation, does not satisfy the imposed tolerances, and in one instance has a maximum error almost 700 times larger than the desired tolerance. On the other hand, RK34Q8 always achieves the desired level of accuracy. Error curves are shown in Figures 1 and 2, wherein the effects of the quenching procedure are clear. In each figure, there are two plots; it is understood that these plots share a common legend and a common x-axis. In the RK34Q8 plots, the quenches occur at those values of x where the error exhibits a sharp decrease.

We have estimated the absolute local error ε_{i+1}^8 of the RK8 solution using Richardson extrapolation (see PR1 for details), even though this requires extra computational effort. We then estimate the global error in the RK8 solution using

$$\mathbf{\Delta}_{i+1}^8 = \varepsilon_{i+1}^8 + \alpha_i^8 \mathbf{\Delta}_i^8 \tag{19}$$

with $\mathbf{\Delta}_0^8 = \mathbf{0}$ and

$$\begin{aligned} \alpha_i^8 &= \mathbf{I}_2 + h_i \mathbf{F}_y^8 (x_i, \xi_i) \\ &\approx \mathbf{I}_2 + h_i \mathbf{F}_y^8 (x_i, \mathbf{y}_i) \\ &\approx \mathbf{I}_2 + h_i \mathbf{f}_y^8 (x_i, \mathbf{y}_i) \\ &= \begin{bmatrix} 1 & h_i \\ -h_i & 1 \end{bmatrix}. \end{aligned}$$

For Case I, we estimate that the maximum magnitude of either component in $\mathbf{\Delta}^8$ is 13×10^{-12}; the actual value is 9×10^{-12}. For Case I, we estimate the maximum magnitude of either component in $\mathbf{\Delta}^8$ to be 5×10^{-12}, while the actual value is

3×10^{-12}. In both cases, our estimate is good and slightly larger than the actual value. Of course, using (19), these estimates can be computed *in situ*, i.e. as the iteration proceeds. This global error is measure of the quality of the solution $\mathbf{w}^8$ which is used for the quenching process. If one or more of the components of the estimated value of $\mathbf{\Delta}^8$ is considered to be too large relative to δ_A and/or δ_R, we propose that δ_A and/or δ_R be increased relative to the estimated value of $\mathbf{\Delta}^8$. This amounts to reducing the level of accuracy imposed on the problem, but would still yield reliable, albeit less accurate, results. If one or more of the components of $\mathbf{\Delta}^8$ become comparable to δ_A and/or δ_R, then the entire quenching process will be compromised, because then $\mathbf{w}^8$ is no more accurate than $\mathbf{w}^3$, which defeats the purpose of quenching. This strategy can be summarized as

$$\max_{j=1,2}\left\{\frac{\left|\mathbf{\Delta}^8\right|_j}{\delta_A}, \frac{\left|\mathbf{\Delta}^8\right|_j}{\delta_R}\right\} > \gamma \Rightarrow \left\{\begin{array}{l} \delta_A \to \eta_A\delta_A \\ \delta_R \to \eta_R\delta_R \end{array}\right.$$

where $\gamma < 1, \eta_A > 1$ and $\eta_R > 1$ are user-defined, and the index j indicates component. For example, in Case II ($\delta_A = \delta_R = 10^{-10}$), if we had set $\gamma = 0.03$, we would have estimated $\left|\mathbf{\Delta}^8\right|_1 > 0.03\delta_A$ at $x \sim 10.4$. We could then have made the adjustments $\delta_A \to 2\delta_A, \delta_R \to 2\delta_R$, say. On the remainder of the interval, $\left|\mathbf{\Delta}^8\right|_j \not> \gamma\delta_A = 6 \times 10^{-12}$, and so no further adjustments to the tolerances would have been needed. It is reasonable to believe that a tolerance of $\delta_A = \delta_R = 2 \times 10^{-10}$ would have been considered acceptable, in these circumstances.

6. Conclusion

We have extended the functionality of the RK*rv*Q*z* algorithm, which now can be applied to IVPs in the form of systems of ODEs, and for which relative and/or absolute tolerances on local and global error can be imposed. The simple harmonic oscillator has served as a useful example for demonstrating this updated version of RK*rv*Q*z*.

References

Butcher, J. C. (2003). *Numerical Methods for Ordinary Differential Equations,* Chichester: Wiley.

Hairer, E., Norsett, S.P., and Wanner, G. (2000). *Solving Ordinary Differential Equations I: Nonstiff Problems,* Berlin: Springer.

Iserles, A. (2009). *A First Course in the Numerical Analysis of Differential Equations,* Cambridge: CUP.

Kincaid, D., and Cheney, W. (2002). *Numerical Analysis: Mathematics of Scientific Computing 3rd ed.*, Pacific Grove: Brooks/Cole.

LeVeque, R. J. (2007). *Finite Difference Methods for Ordinary and Partial Differential Equations*, Philadelphia: SIAM.

Prentice, J. S. C. (2009). General error propagation in the RK*r*GL*m* method, *Journal of Computational and Applied Mathematics,* 228, 344-354.

Prentice, J. S. C. (2011). Stepwise Global Error Control in an Explicit Runge-Kutta Method using Local Extrapolation with High-Order Selective Quenching, *Journal of Mathematics Research,* 3, 2, 126-136.

Table 1. Performance parameters for RK34 and RK34Q8 applied to the oscillator problem

	Case I			Case II	
	RK34	RK34Q8		RK34	RK34Q8
y_1 : max $\lvert\Delta_i\rvert$	104×10^{-5}	0.95×10^{-5}		692×10^{-10}	0.98×10^{-10}
y_2 : max $\lvert\Delta_i\rvert$	32×10^{-5}	0.97×10^{-5}		641×10^{-10}	0.98×10^{-10}
# of quenches	0	7		0	13

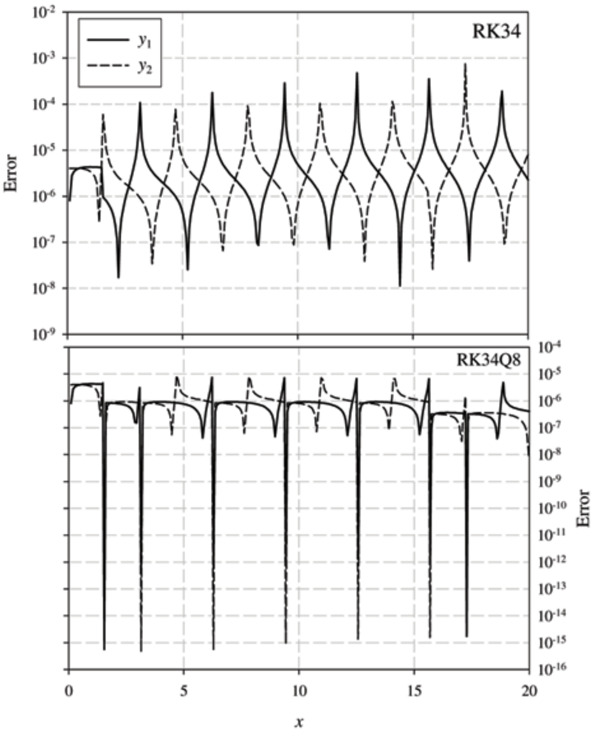

Figure 1. Global error curves for both components of the oscillator problem, using RK34 and RK34Q8, for Case I

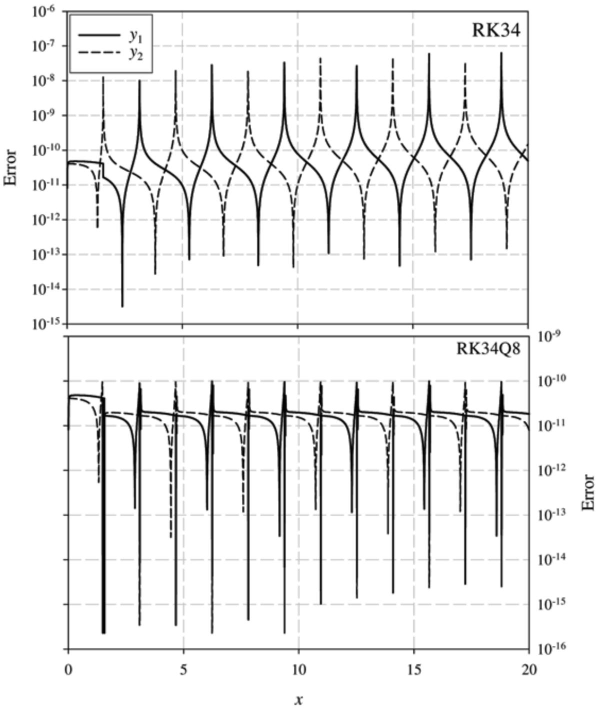

Figure 2. Global error curves for both components of the oscillator problem, using RK34 and RK34Q8, for Case II.