# Stepwise Global Error Control in an Explicit Runge-Kutta Method Using Local Extrapolation with High-Order Selective Quenching

J.S.C. Prentice

Department of Applied Mathematics, University of Johannesburg P.O. Box 524, Auckland Park, 2006, South Africa Tel: 271-1559-3145 E-mail: jprentice@uj.ac.za

Received: December 9, 2010 Accepted: January 6, 2011 doi:10.5539/jmr.v3n2p126

## Abstract

Stepwise local error control using local extrapolation in Runge-Kutta methods is well-known. In this paper, we introduce an algorithm, designated RK*rv*Q*z*, that is capable of controlling local and global errors in a stepwise manner. The algorithm utilizes three Runge-Kutta methods, of orders *r*, *v* and *z*, with  $r < v \ll z$ . Local error is controlled in the usual way using local extrapolation, whereas global error is controlled using a technique we have termed 'quenching', which exploits the availability of a very high order solution and the use of a 'safety factor', often present in local extrapolation methods. An example using RK34Q8 gives a clear indication of the effectiveness of the method.

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

## 1. Introduction

Initial-value problems (IVPs) of the form

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

$$x \in [x_0, x_f]$$
  

$$y(x_0) = y_0$$

are often solved numerically using a Runge-Kutta method. Usually, some form of local error control is implemented in a step-by-step manner. However, local error control does not guarantee global error control. Global error could still accumulate and become unacceptably large. Global error control, if implemented, requires a reintegration process solving the problem again using a smaller stepsize - and is not a stepwise process.

In this paper, we introduce an idea for achieving both local and global error control in a stepwise fashion, so that reintegration is not needed. For simplicity, we restrict our considerations to a nonstiff scalar problem (as opposed to a system of differential equations), for which the solution does not vary significantly in magnitude on  $[x_0, x_f]$  (this allows us to consider absolute error control only, rather than relative error control).

We describe relevant concepts in Section 2; we show how global error can grow, despite local error control, in Section 3; in Section 4 we discuss our approach for stepwise global error control; we describe this approach in algorithmic form in Section 5; and in Section 6 we give a numerical example, as a demonstration of our algorithm.

## 2. Relevant Concepts, Terminology and Notation

In this section, we discuss concepts relevant to the rest of paper, including appropriate terminology and notation. The reader is referred to Hairer et al (2000), Butcher (2003), Iserles (2009), Kincaid & Cheney (2002), LeVeque (2007), and many references therein, for discussions of Runge-Kutta methods and local error control in such methods.

## 2.1 Runge-Kutta Methods

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

$$k_{p} = f\left(x_{i} + c_{p}h_{i}, w_{i} + h_{i}\sum_{q=1}^{m} a_{pq}k_{q}\right) \qquad p = 1, 2, ..., m$$

$$w_{i+1} = w_{i} + h_{i}\sum_{p=1}^{m} b_{p}k_{p} \equiv w_{i} + h_{i}F\left(x_{i}, w_{i}; h_{i}\right).$$
(1)

Such a method is said to have *m* stages (the  $k_q$ ). We note that if  $a_{pq} = 0$  for all  $p \leq q$ , then the method is said to be *explicit*; otherwise, it is known as an *implicit* RK method. We will restrict our considerations here to explicit methods. The number of stages is related to the order *r* of the method, and for explicit methods we always have  $r \leq m$ . In the second line of (1), we have implicitly defined the function *F*. The symbol *w* is used here and throughout to indicate the

approximate numerical solution, whereas the symbol 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

$$w_{i+1}^r = w_i^r + h_i F^r \left( x_i, w_i^r; h_i \right).$$
<sup>(2)</sup>

In a sense, RKr is defined by  $F^r$ , although it is understood that, for any r, there are, generally speaking, numerous possible choices for  $F^r$ . 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.

2.2 Error Propagation

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

$$\Delta_{i+1}^r \equiv w_{i+1}^r - y_{i+1}$$

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

$$\varepsilon_{i+1}^{r} \equiv \left[ y_{i} + h_{i} F^{r} \left( x_{i}, y_{i}; h_{i} \right) \right] - y_{i+1}.$$
(3)

Note the use of the exact value  $y_i$  in the bracketed term in (3). Again, the superscripts are labels.

We have previously shown (Prentice, 2009) that

$$\Delta_{i+1}^r = \varepsilon_{i+1}^r + \alpha_i^r \Delta_i^r$$

$$\alpha_i^r \equiv 1 + h_i F_v^r (x_i, \xi_i; h_i),$$
(4)

where  $\xi_i \in (y_i, y_i + \Delta_i^r)$ . Equation (4) provides the relationship between local and global errors in RK*r*. We will assume that  $\Delta_0 = 0$  (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 RK*r*, it is known that

$$\varepsilon_{i+1}^r \propto h_i^{r+1}$$
  
 $\Delta_{i+1}^r \propto h^r.$ 

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. RK*r* and RK*v*, with r < v. Let  $w_{i+1}^r$  denote the approximate solution at  $x_{i+1}$  obtained with RK*r*, and similarly for  $w_{i+1}^v$ . Let the local error at  $x_{i+1}$  in the RK*r* method be given by  $\varepsilon_{i+1}^r = \beta_{i+1}^r h_i^{r+1}$ , 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  $w_i^r, w_i^v = y_i$ , which means that  $\Delta_i^r, \Delta_i^v = 0$ , we have

$$\begin{split} w_{i+1}^{r} - w_{i+1}^{v} &= y_{i+1} + \Delta_{i+1}^{r} - \left(y_{i+1} + \Delta_{i+1}^{v}\right) \\ &= \varepsilon_{i+1}^{r} + \alpha_{i}^{r} \Delta_{i}^{r} - \left(\varepsilon_{i+1}^{v} + \alpha_{i}^{v} \Delta_{i}^{v}\right) \\ &= \varepsilon_{i+1}^{r} - \varepsilon_{i+1}^{v} \\ &= \beta_{i+1}^{r} h_{i}^{r+1} - \beta_{i+1}^{v} h_{i}^{v+1} \\ &\approx \beta_{i+1}^{r} h_{i}^{r+1} \end{split}$$

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

$$\beta_{i+1}^r \approx \frac{w_{i+1}^r - w_{i+1}^v}{h^{r+1}}.$$
(5)

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  $\delta$ . Moreover, assume that, using stepsize  $h_i$ , we find

$$\left|\varepsilon_{i+1}^{r}\right| = \left|\beta_{i+1}^{r}h_{i}^{r+1}\right| > \delta.$$

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

$$\left|\beta_{i+1}^{r}\left(h_{i}^{*}\right)^{r+1}\right| = \delta \Rightarrow h_{i}^{*} = \left(\frac{\delta}{\left|\beta_{i+1}^{r}\right|}\right)^{\frac{1}{r+1}}$$
(6)

Published by Canadian Center of Science and Education

and we repeat the RK computation with this new stepsize. This, of course, gives

$$x_{i+1} = x_i + h_i^*.$$

This procedure is then carried out on the next step, and so on. If the estimated error does not exceed the tolerance, then no stepsize adjustment is necessary, and we proceed to the next step.

Often, we introduce a so-called 'safety factor'  $\sigma$ , as in

$$h_i^* = \sigma \left( \frac{\delta}{\left| \beta_{i+1}^r \right|} \right)^{\frac{1}{r+1}},$$

where  $\sigma < 1$ , so that the new stepsize is slightly smaller than that given by (6). This is an attempt to cater for the possibility that  $\beta_{i+1}^r$  may have been underestimated, due to the assumptions made in deriving (5). The choice of the value of  $\sigma$  is subjective, although a representative value is 0.85.

There is a very important point to be noted. Our procedure for determining  $\beta_{i+1}^r$  hinged on the requirement  $w_i^r, w_i^v = y_i$ . However, we only have the exact solution at the initial point  $x_0$ ; at all subsequent nodes, the solution is approximate. How do we meet the requirement  $w_i^r, w_i^v = y_i$ ?

Since the higher-order solution  $w_i^v$  is available, we simply use  $w_i^v$  as input to generate both  $w_{i+1}^r$  (using RKr), and  $w_{i+1}^v$  (using RKv). In other words, we are assuming that  $w_i^v$  is accurate enough, relative to  $w_i^r$ , to be regarded as the exact value - an assumption entirely consistent with the assumption made in deriving (5). 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*.

#### 3. The Problem

We assume that  $w_i^v$  is used to generate  $w_{i+1}^r$  and  $w_{i+1}^v$ . Such value of  $w_{i+1}^r$  (and associated quantities) will be denoted  $w_{i+1}^{rv}$ . Hence, we have

$$\begin{split} \Delta_{i+1}^{rv} &= \beta_{i+1}^{r} h_{i}^{r+1} + \alpha_{i}^{rv} \Delta_{i}^{v} \\ &= \beta_{i+1}^{r} h_{i}^{r+1} + \Delta_{i}^{v} + h_{i} F_{y}^{rv} \Delta_{i}^{v} \\ \Delta_{i+1}^{v} &= \beta_{i+1}^{v} h_{i}^{v+1} + \alpha_{i}^{v} \Delta_{i}^{v} \\ &= \beta_{i+1}^{v} h_{i}^{v+1} + \Delta_{i}^{v} + h_{i} F_{y}^{v} \Delta_{i}^{v}. \end{split}$$

Hence,

$$w_{i+1}^{rv} - w_{i+1}^{v} = \beta_{i+1}^{r} h_{i}^{r+1} + \Delta_{i}^{v} + h_{i} F_{y}^{rv} \Delta_{i}^{v} - \left(\beta_{i+1}^{v} h_{i}^{v+1} + \Delta_{i}^{v} + h_{i} F_{y}^{v} \Delta_{i}^{v}\right)$$
  
$$= \beta_{i+1}^{r} h_{i}^{r+1} - \beta_{i+1}^{v} h_{i}^{v+1} + \left(F_{y}^{rv} - F_{y}^{v}\right) h_{i} \Delta_{i}^{v}$$
  
$$\approx \beta_{i+1}^{r} h_{i}^{r+1}$$
(7)

for small  $h_i$ , because  $h_i \Delta_i^v = O(h^{v+1})$ . We see that the presence of global error in the higher-order solution does not affect the expression for  $\beta_{i+1}^r$  obtained under the assumption  $w_i^r, w_i^v = y_i$ , particularly if  $r \ll v$ . Usually, though, r = v - 1 is effective for local extrapolation.

However, the expression for  $\Delta_{i+1}^{rv}$  informs of a potential problem: we have

$$\Delta_{i+1}^{rv} = \beta_{i+1}^r h_i^{r+1} + \alpha_i^{rv} \Delta_i^v,$$

where  $\Delta_i^v$  is the global error in  $w_i^v$ . In (7), we see that a subtractive cancellation ensures that the  $\Delta_i^v$  term does not enter directly into the estimate for  $\beta_{i+1}^r$ . Nevertheless, even if  $|\beta_{i+1}^r h_i^{r+1}| \le \delta$ , we could still have  $|\Delta_{i+1}^{rv}| > \delta$ , perhaps substantially so, if  $|\Delta_i^v|$  is large. Moreover, we should certainly expect that  $|\Delta_i^v|$  could become large under iteration (i.e. as *i* increases), since global error is essentially an accumulation of local errors. The point here is that, even if local error control is effective, the global error  $\Delta_{i+1}^{rv}$  could become large, and could grow in an uncontrolled fashion.

Usually, the approach to controlling global error involves estimating the global error after the RK solution has been obtained on the entire interval of integration  $[x_0, x_f]$ , and then repeating the RK computation on  $[x_0, x_f]$ , using a suitably reduced stepsize. Such an approach is termed *reintegration*.

Our ambition is to develop an algorithm through which the global error in  $w_{i+1}^{rv}$  can be estimated and controlled in a stepby-step manner, without the need for reintegration, while at the same time controlling local error in the usual manner of local extrapolation. To achieve this end, we will use a third RK method of very high order, and we will exploit the safety factor mentioned previously. Our motivation for developing this algorithm is both academic and practical: academically speaking, it is natural to ask whether or not global error can be controlled in a stepwise manner, given that local error is controlled in such fashion; practically speaking, real-time computations which require the use of an accurate solution, such as a feedback loop in a control system, cannot make use of a reintegration process, and so are reliant on the quality of output generated by stepwise algorithms.

#### 4. The Solution

Say we have three explicit methods available (RKr, RKv and RKz), with

 $r < v \ll z$ ,

so that RKz is of much higher order than RKr and RKv. We would suggest v = r + 1 and z = v + 2, at least.

Let  $h_i^*$  denote the stepsize for which

$$\beta_{i+1}^r (h_i^*)^{r+1} = \delta$$

where  $\delta$  is a user-imposed tolerance. Of course, since the safety factor  $\sigma$  is less than unity, we have

$$\beta_{i+1}^r \left(\sigma h_i^*\right)^{r+1} < \delta.$$

The quantity  $\sigma h_i^*$  is the *de facto* stepsize used, as and when required, in local extrapolation.

We implement local extrapolation in the usual way, using RKr and RKv, propagating  $w_i^v$  at each step. Simultaneously, we implement RKz at the same nodes. At each node we have

$$\begin{split} w_{i+1}^{rv} &= y_{i+1} + \varepsilon_{i+1}^{r} + \alpha_{i}^{rv} \Delta_{i}^{v} \\ w_{i+1}^{v} &= y_{i+1} + \varepsilon_{i+1}^{v} + \alpha_{i}^{v} \Delta_{i}^{v} \\ w_{i+1}^{z} &= y_{i+1} + \varepsilon_{i+1}^{z} + \alpha_{i}^{z} \Delta_{i}^{z}. \end{split}$$

Additionally, we can propagate  $w_i^z$  in RKr, which gives

$$w_{i+1}^{rz} = y_{i+1} + \varepsilon_{i+1}^r + \alpha_i^{rz} \Delta_i^z,$$

where  $w_{i+1}^{r_z}$  is the solution obtained using RKr with  $w_i^z$  as input.

These expressions give

$$w_{i+1}^{r\nu} - w_{i+1}^{z} = \varepsilon_{i+1}^{r} + \alpha_{i}^{r\nu}\Delta_{i}^{\nu} - \left(\varepsilon_{i+1}^{z} + \alpha_{i}^{z}\Delta_{i}^{z}\right) \approx \varepsilon_{i+1}^{r} + \alpha_{i}^{r\nu}\Delta_{i}^{\nu}$$

$$w_{i+1}^{r\nu} - w_{i+1}^{rz} = \varepsilon_{i+1}^{r} + \alpha_{i}^{r\nu}\Delta_{i}^{\nu} - \left(\varepsilon_{i+1}^{r} + \alpha_{i}^{rz}\Delta_{i}^{z}\right) \approx \alpha_{i}^{r\nu}\Delta_{i}^{\nu}$$

$$w_{i+1}^{rz} - w_{i+1}^{z} = \varepsilon_{i+1}^{r} + \alpha_{i}^{rz}\Delta_{i}^{z} - \left(\varepsilon_{i+1}^{z} + \alpha_{i}^{z}\Delta_{i}^{z}\right) \approx \varepsilon_{i+1}^{r}$$
(8)

since  $r \ll z$ . We thus have a reliable estimate of the components of  $\Delta_{i+1}^{rv} = \varepsilon_{i+1}^r + \alpha_i^{rv} \Delta_i^v$ .

Now, say a suitable stepsize adjustment has been made, and the solutions  $\{w_{i+1}^{rv}, w_{i+1}^{rz}, w_{i+1}^{v}, w_{i+1}^{z}\}$  at  $x_{i+1} = x_i + \sigma h_i^*$  have been computed. We have

$$\left|\varepsilon_{i+1}^{r}\right| = \left|\beta_{i+1}^{r}\left(\sigma h_{i}^{*}\right)^{r+1}\right| < \delta,$$

so it is certainly possible that

$$\left|\Delta_{i+1}^{r\nu}\right| = \left|\beta_{i+1}^{r} \left(\sigma h_{i}^{*}\right)^{r+1} + \alpha_{i}^{r\nu} \Delta_{i}^{\nu}\right| \leqslant \delta.$$

$$\tag{9}$$

If this is the case, we proceed to the next step.

If, however, we find

$$\left|\beta_{i+1}^{r}\left(\sigma h_{i}^{*}\right)^{r+1}+\alpha_{i}^{r\nu}\Delta_{i}^{\nu}\right|>\delta,\tag{10}$$

we must conclude that  $\Delta_i^{\nu}$  has become too large. We then set

$$w_i^v = w_i^z$$

since

$$v_i^z = y_i + \varepsilon_i^z + \alpha_{i-1}^z \Delta_{i-1}^z \approx y_i$$

because RKz is of much higher order than RKv, and we recalculate  $w_{i+1}^{rv}$  and  $w_{i+1}^{v}$ . In other words,  $w_i^{v}$  is replaced with a much more accurate value. This will yield

$$\begin{aligned} & w_{i+1}^{rv} &= \varepsilon_{i+1}^r + \alpha_i^{rz} \Delta_i^z \approx \varepsilon_{i+1}^r \\ & w_{i+1}^v &= \varepsilon_{i+1}^v + \alpha_i^{vz} \Delta_i^z \approx \varepsilon_{i+1}^v , \end{aligned}$$

Published by Canadian Center of Science and Education

so that  $w_{i+1}^{rv}$  and  $w_{i+1}^{v}$  will now have relatively small global error  $(\propto \Delta_i^z)$  accumulated from previous iterations. Effectively, we have greatly reduced, or *quenched*, the global error present in  $w_{i+1}^{rv}$  and  $w_{i+1}^{v}$ , due to RKv. We only select to perform a quench when we encounter the case in (10); the case (9) does not require a modification of  $w_i^v$  (and, hence,  $w_{i+1}^{rv}$  and  $w_{i+1}^v$ ).

It may occur, and often does, that a stepsize adjustment via local extrapolation is not required, simply because the stepsize is already small enough. In such case we must, nevertheless, still test the inequality (10) and perform a quench, if necessary.

Usually, in local extrapolation, we would use  $w_{i+1}^{rv}$  and  $w_{i+1}^{v}$  to estimate the local error  $\varepsilon_{i+1}^{r}$  but, if  $w_{i+1}^{rz}$  is available, we should rather use (8) to estimate  $\varepsilon_{i+1}^{r}$ , because such estimate is more reliable.

The importance of the safety factor is apparent:  $\sigma$  ensures that  $|\varepsilon_{i+1}^r|$  is *strictly* less than  $\delta$ , so that the global error component  $\alpha_i^{rv}\Delta_i^v$  can be accommodated to some degree. The extent of this accommodation is determined by  $\sigma^{r+1}$ . Say  $\sigma = 0.85$  and r = 3, so that  $\sigma^{r+1} = 0.522$ . Then, assuming both components of  $\Delta_{i+1}^{rv}$  have the same sign,  $\alpha_i^{rv}\Delta_i^v$  can be accommodated up to a magnitude of  $0.478\delta$ , before quenching is needed.

Lastly, we emphasize that, at each node  $x_{i+1}$ , it is  $w_{i+1}^{rv}$  that is presented as the solution to the IVP - this is the numerical solution for which both local and global error control has been performed.

#### 5. The RKrvQz Algorithm

We describe the sequential execution of the algorithm, which we designate RKrvQz, on a generic subinterval  $[x_i, x_{i+1}]$ :

- 1. Use  $w_i^v$  and  $w_i^z$  in RK*r*, RK*v* and RK*z* to generate  $w_{i+1}^{rv}$ ,  $w_{i+1}^{rz}$ ,  $w_{i+1}^v$  and  $w_{i+1}^z$ . Use  $h_i = h_{i-1}$  as the stepsize (we discuss the case of  $h_0$  in the Appendix).
- 2. Estimate  $\varepsilon_{i+1}^r$  using  $w_{i+1}^{rv} w_{i+1}^v$  or  $w_{i+1}^{rz} w_{i+1}^z$ . The former is the usual local extrapolation approach, whereas the latter is more reliable.
- 3. If  $|\varepsilon_{i+1}^r| > \delta$ , determine a new stepsize  $h_i = \sigma h_i^*$ , and repeat steps 1 and 2 using this new stepsize. Then go to step 5.
- 4. If  $|\varepsilon_{i+1}^r| \leq \delta$ , go to step 5.
- 5. Estimate  $\Delta_{i+1}^{rv} = \varepsilon_{i+1}^r + \alpha_i^{rv} \Delta_i^v$  using  $w_{i+1}^{rv} w_{i+1}^z$ .
- 6. If  $\left|\Delta_{i+1}^{rv}\right| > \delta$ , set  $w_i^v = w_i^z$  (quenching) and repeat steps 1 and 2 using the stepsize determined in step 3. Then go to step 8.
- 7. If  $\left|\Delta_{i+1}^{rv}\right| \leq \delta$ , go to step 8.
- 8. We now have the numerical solutions  $w_{i+1}^{rv}$ ,  $w_{i+1}^{r}$ ,  $w_{i+1}^{rz}$  and  $w_{i+1}^{z}$ , at  $x_{i+1} = x_i + h_i$ , with the magnitude of both the local and global error in  $w_{i+1}^{rv}$  less than the tolerance  $\delta$ .

#### 6. Numerical Example

By way of example, we apply RK34Q8 (r = 3, v = 4, z = 8) to the IVP

$$y' = \left(\frac{\ln 1000}{100}\right) y$$
$$x \in [0, 100]$$
$$y(0) = 1$$

The coefficient in the differential equation has been chosen so that *y* does not exceed 1000 on the interval of integration, i.e. *y* does not vary substantially, so that absolute error control is suitable. The exact solution is, of course,

$$y(x) = e^{\frac{\ln 1000}{100}x}.$$

As we shall see, the RK3 global error in this problem is a rapidly increasing function of x, and so it is an ideal problem for demonstrating the capabilities of RKrvQz. The RK methods used in RK34Q8 are RK3 (Kincaid & Cheney, 2002), the 'classical' RK4 (LeVeque, 2007) and Fehlberg's RK8 (Butcher, 2003). The tableaux for these methods are given in Tables 1 and 2.

In Figure 1 we show  $\varepsilon_i^3$  and  $\alpha_i^{34}\Delta_i^3$  for  $\delta = 10^{-4}$ ,  $10^{-8}$ . The local errors are always less than  $\delta$ , with the 'zigzag' shape of the curves reflecting stepsize adjustments. Nevertheless, we see that  $\alpha_i^{34}\Delta_i^3$  increases monotonically and, at x = 100, it is about 100 times larger than  $\delta$ .

Global errors obtained with local error control only (the sum of the errors shown in Figure 1), and with RK34Q8, are shown in Figure 2. For the former, the global error increases significantly, even though the local error has been controlled - a potent example of the problem discussed in Section 3. The RK34Q8 global error, however, remains bounded by  $\delta$  on the entire interval of integration - a vivid demonstration of the capability of RK34Q8. Here, the zigzag nature of the error curves is due to quenching (the propagation of  $w_i^8$ ). In all calculations shown in these figures, we used  $\sigma = 0.85$ , and the initial stepsize  $h_0$  was determined using the procedure described in the Appendix.

It may be prudent to keep track of the global error in RK8. For this, we estimate  $\varepsilon_{i+1}^8$  using Richardson extrapolation (Butcher, 2003)

$$\varepsilon_{i+1}^8 \approx \frac{w_{i+1}^8 - w_{i+1}^8 \left(\frac{h_i}{2}\right)}{1 - 2^{-8}},$$

where  $w_{i+1}^8\left(\frac{h_i}{2}\right)$  is determined from

$$\theta = w_i^8 + \frac{h_i}{2} F^8 \left( x_i, w_i^8; \frac{h_i}{2} \right)$$
$$w_{i+1}^8 \left( \frac{h_i}{2} \right) = \theta + \frac{h_i}{2} F^8 \left( x_i + \frac{h_i}{2}, \theta; \frac{h_i}{2} \right).$$
(11)

We then assume

$$F_{y}^{8}\left(x_{i},\xi_{i};h_{i}\right)\approx f_{y}\left(x_{i},w_{i}^{8}\right),$$

so that

$$\Delta_{i+1}^8 = \varepsilon_{i+1}^8 + \alpha_i^8 \Delta_i^8 \approx \varepsilon_{i+1}^8 + \left(1 + h_i f_y\left(x_i, w_i^8\right)\right) \Delta_i^8$$

With  $\Delta_0^8 = 0$  and  $\Delta_1^8 = \varepsilon_1^8$ , we can estimate  $\Delta_{i+1}^8$  as RK34Q8 proceeds. If we detect that  $\Delta_{i+1}^8$  is approaching  $\delta$  in magnitude, then we could increase  $\delta$ . This would mean that the tolerance on the global error increases occasionally, as RK34Q8 proceeds, but this is better than incorrectly estimating  $\varepsilon_{i+1}^r$  and  $\alpha_i^{rv}\Delta_i^y$ . A possible condition for increasing  $\delta$  would be  $|\Delta_{i+1}^8| \sim 0.01\delta$ , with  $\delta$  being increased to  $2\delta$ , perhaps. We must state here that this is purely a speculation on our part; for the example considered here,  $|\Delta_{i+1}^8| < 0.01\delta$  on [0, 100] always. Indeed, in Figure 3 we show actual and estimated values of  $|\Delta_i^8|$  for  $\delta = 10^{-4}$ . It is clear that in both cases,  $|\Delta_{i+1}^8| \ll \delta$ . Also, the estimates are good, particularly for  $\delta = 10^{-4}$ .

In Figure 4, we show global errors in RK34Q8 with  $\sigma = 0.9$ . The increases in the safety factor results in more quenches, in comparison with the error curves in Figure 2.

#### 7. Comments

A few comments, pertaining mostly to possible future research, are appropriate:

- 1. The use of RKz means that RKrvQz requires greater computational effort than standard local error control via local extrapolation. This, of course, is the price we must pay for controlling global error, in addition to local error. However, in comparison with reintegration, the extra effort may not be all that significant. In reintegration, we would need to use RKr and RKv to obtain solutions on the interval of integration using a smaller stepsize (the RKv solution would be needed to confirm the quality of the RKr solution), after having performed local error control on the entire interval. We suspect that the additional computational effort in using RKr and RKv a second time in a reintegration process would probably not be all that different from the computational effort involved in using RKz in RKrvQz. In Section 3, we gave our motives for developing RKrvQz, and we are sure that any extra effort in using RKz is a small price to pay for achieving simultaneous stepwise local and global error control. We are quite sure that, whenever and wherever possible, accuracy must take precedence over efficiency. This notwithstanding, we discuss some possible improvements in the efficiency of RKrvQz in #2 and #3 below.
- 2. The safety factor  $\sigma$  can be used to control the magnitude of  $\alpha_i^{r\nu}\Delta_i^{\nu}$ . Instead of demanding (9), we rather demand

$$\left|\beta_{i+1}^{r}\left(\sigma h_{i}^{*}\right)^{r+1}\right|+\left|\alpha_{i}^{rv}\Delta_{i}^{v}\right|\leqslant\delta,$$

which is more stringent. This implies

$$\left|\alpha_{i}^{rv}\Delta_{i}^{v}\right| \leqslant \delta - \left|\beta_{i+1}^{r}\left(\sigma h_{i}^{*}\right)^{r+1}\right|$$

and if  $\sigma$  is close to unity,  $|\alpha_i^{r\nu}\Delta_i^{\nu}|$  must necessarily be small, otherwise quenching must be performed. Since  $|\alpha_i^{r\nu}\Delta_i^{\nu}|$  is small it is not unreasonable to assume that  $|\alpha_i^{\nu}\Delta_i^{\nu}|$  will also be small, so that the estimate of  $|\varepsilon_{i+1}^r|$  using  $w_{i+1}^{r\nu} - w_{i+1}^{\nu}$  will be reliable. This simply means that it would not be necessary to determine  $w_{i+1}^{rz}$  at each node, which might improve the efficiency of the algorithm.

- 3. Efficiency could also be improved by using an embedded RK pair for RK*r* and RK*v*. This would reduce the total number of stage evaluations at each step. A well-known example of such a method is Fehlberg's embedded RK(4,5) pair (Fehlberg, 1970). However, we must note that for any given *r* and *v* there is no guarantee that an embedded RK(*r*, *v*) pair exists.
- 4. We could use two tolerances  $\delta_1$  and  $\delta_2$ , imposed on the local and global error, respectively, with  $\delta_1 < \delta_2$ . In other words, we do not impose the same level of accuracy on the global error as we do on the local error. This might contradict the objectives in #2 above, though.
- 5. In our numerical example, we estimated  $\varepsilon_{i+1}^z$  by means of Richardson's extrapolation, and then used (4) with  $F_y^z(x_i, \xi_i; h_i) \approx f_y(x_i, w_i^z)$ . A more accurate, and more efficient, estimate might be obtained by using a higher-order method RKz', with z' > z. A high-order embedded RK(z, z') pair might be useful here, such as Fehlberg's RK(7,8), although error control via local extrapolation with this particular pair is not accurate when f is a function of x only (Hairer et al, 2000). Verner has offered an embedded RK(5,6) pair (Verner, 1978), but this would restrict r and v to 2 and 3, respectively, whereas we would probably prefer to have r = 3 or 4.
- 6. We have considered IVPs for which the solution does not vary considerably in magnitude on the interval of integration. This has allowed us to consider absolute error control only (wherein a uniform tolerance is used). For solutions that vary significantly in magnitude, we would need to implement relative error control. This requires using a node-dependent tolerance

$$\delta_i = \max\left\{\delta_A, \delta_R \left| y_i \right| \right\},\,$$

where the tolerances  $\delta_A$  and  $\delta_R$  are user-defined, and  $\delta_A$  is included to cater for those occasions when  $|y_i| \sim 0$ .

- 7. Applying RKrvQz to a system of differential equations would require error control to be applied to each component of the system. This would lead to a value of  $h_i^*$  for each component we would choose the smallest. We would also test the inequality (9) for each component; if at least one of the components failed the test, we would perform a quench in all components.
- 8. We anticipate that it should not be difficult to modify RK*rv*Q*z* for implicit RK methods, which would be suitable for stiff problems. In this regard, RK*r*, RK*v* and RK*z* would all be implicit, A-stable RK methods. For example, we could use the well-known second-order Implicit Midpoint Rule, the fourth-order Kuntzmann-Butcher method, and the sixth-order Kuntzmann-Butcher method in place of RK*r*, RK*v* and RK*z*, respectively. We could denote such an algorithm by IRK24Q6, where the 'I' indicates 'implicit'.

## 8. Conclusion

We have developed a numerical algorithm for solving initial-value problems in ordinary differential equations, designated RKrvQz, that is capable of controlling both local and global errors in the numerical solution, in a stepwise manner. The algorithm uses local extrapolation to control the local error, and so-called quenching to retard the build-up of global error. Three Runge-Kutta methods are used in RKrvQz: RKr and RKv are used for local error control, and RKz is used for the estimation of global error and the quenching procedure. A numerical example with a rapidly increasing global error has demonstrated the effectiveness of the algorithm. In this exploratory paper, we have restricted our work to scalar problems, with absolute error control. The extension of RKrvQz to systems, and the incorporation of relative error control must be the subject of future research.

## References

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

Fehlberg, E. (1970). Low-order classical Runge-Kutta formulas with step size control and their application to some heat transfer problems, *Computing*, 6, 61-71.

Gladwell, I., Shampine, L.F., and Brankin, R.W. (1987). Automatic selection of the initial step size for an ODE solver, *Journal of Computational and Applied Mathematics*, 18, 175-192.

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 RKrGL*m* method, *Journal of Computational and Applied Mathematics*, 228, 344 354.

Verner, J.H. (1978). Explicit Runge-Kutta methods with estimates of the local truncation error, SIAM Journal on Numerical Analysis, 15, 772-790.

### Appendix

To implement RK*rv*Q*z*, as described in Section 5, we need an initial stepsize  $h_0$ . We can estimate  $h_0$  in the following way: we assume that, for RK*r*, the local error on  $[x_0, x_1]$  is given by

$$\varepsilon_{1} = \frac{y^{(r+1)}(\eta)}{(r+1)!} h_{0}^{r+1}$$

$$\eta \in (x_{0}, x_{1})$$
(12)

(similar to Gladwell et al, 1987), which is the local error in the Taylor method of order r, to which RKr is equivalent (by construction). We define the operator

$$\widehat{D} \equiv \frac{\partial}{\partial x} + f \frac{\partial}{\partial y}$$

and we determine, using computer algebra software,

$$y^{(r+1)} = \frac{d^r f}{dx^r} = \underbrace{\widehat{D}\widehat{D}\cdots\widehat{D}}_{r \text{ times}} f$$

to obtain a symbolic expression for  $y^{(r+1)}$ , in terms of x and y. We then choose N equispaced nodes on  $[x_0, \alpha]$ , where  $\alpha$  is user-defined, and we use Euler's method to obtain approximate solutions w at these N nodes, subject to the initial value  $y_0$ . The value of N is also user-defined (N = 10 should be sufficient) and  $\alpha$  should be chosen close to  $x_0$ , particularly for a strict tolerance  $\delta$ . We then substitute the values of x and w so obtained at these nodes into the symbolic expression for  $y^{(r+1)}(x, y)$ , and determine the average

$$\overline{\left|y^{(r+1)}\right|} \equiv \frac{\sum\limits_{i=1}^{N} \left|y^{(r+1)}\left(x_{i}, w_{i}\right)\right|}{N}$$

The stepsize  $h_0$  is then determined from

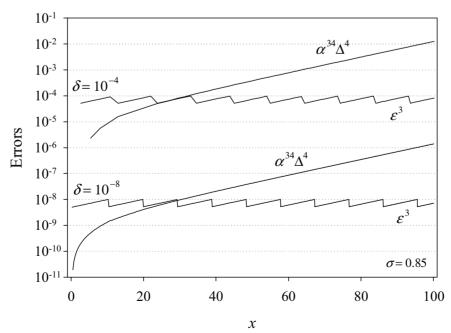
$$h_0 = \left(\frac{(r+1)!\delta}{\left|y^{(r+1)}\right|}\right)^{\frac{1}{r+1}},$$

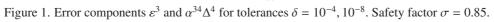
where the tolerance  $\delta$  replaces  $\varepsilon_1$  in (12). Note that we use Euler's method purely to keep computational effort to a minimum, but we could actually use any explicit RK method, including RK*r*.

Table 1. Tableaux for RK3 (left) and RK4

## Table 2. Tableaux for RK8

$\frac{2}{27}$	$\frac{2}{27}$												
$\frac{1}{9}$	$\frac{1}{36}$	$\frac{1}{12}$											
$\frac{1}{6}$	$\frac{1}{24}$	0	$\frac{1}{8}$										
$\frac{5}{12}$	$\frac{5}{12}$	0	$-\frac{25}{16}$	$\frac{25}{16}$									
$\frac{1}{2}$	$\frac{1}{20}$	0	0	$\frac{1}{4}$	$\frac{1}{5}$								
$\frac{5}{6}$	$-\frac{25}{108}$	0	0	$\frac{125}{108}$	$-\frac{65}{27}$	$\frac{125}{54}$							
$\frac{1}{6}$	$\frac{31}{300}$	0	0	0	$\frac{61}{225}$	$-\frac{2}{9}$	$\frac{13}{900}$						
$\frac{2}{3}$	2	0	0	$-\frac{53}{6}$	$\frac{704}{45}$	$-\frac{107}{9}$	$\frac{67}{90}$	3					
$\frac{1}{3}$	$-\frac{91}{108}$	0	0	$\frac{23}{108}$	$-\frac{976}{135}$	$\frac{311}{54}$	$-\frac{19}{60}$	$\frac{17}{6}$	$-\frac{1}{12}$				
1	$\frac{2383}{4100}$	0	0	$-\frac{341}{164}$	$\frac{4496}{1025}$	$-\frac{301}{82}$	$\frac{2133}{4100}$	$\frac{45}{82}$	$\frac{45}{164}$	$\frac{18}{41}$			
0	$\frac{3}{205}$	0	0	0	0	$-\frac{6}{41}$	$-\frac{3}{205}$	$-\frac{3}{41}$	$\frac{3}{41}$	$\frac{6}{41}$	0		
1	$-\frac{1777}{4100}$	0	0	$-\frac{341}{164}$	$\frac{4496}{1025}$	$-\frac{289}{82}$	$\frac{2193}{4100}$	$\frac{51}{82}$	$\frac{33}{164}$	$\frac{12}{41}$	0	1	
	0	0	0	0	0	$\frac{34}{105}$	$\frac{9}{35}$	$\frac{9}{35}$	$\frac{9}{280}$	$\frac{9}{280}$	0	$\frac{41}{840}$	$\frac{41}{840}$





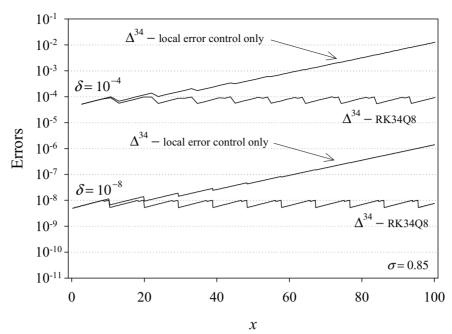


Figure 2. Global errors in RK3 (local error control only) and RK34Q8, for tolerances  $\delta = 10^{-4}$ ,  $10^{-8}$ . Safety factor  $\sigma = 0.85$ .

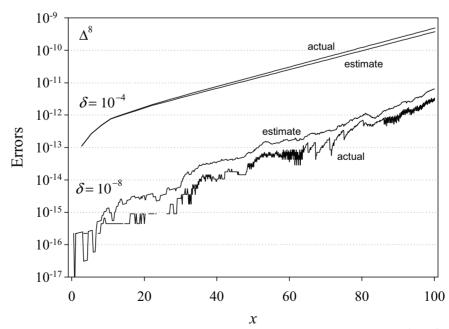


Figure 3. Global errors in RK8 (actual and estimated), for tolerances  $\delta = 10^{-4}$ ,  $10^{-8}$ .

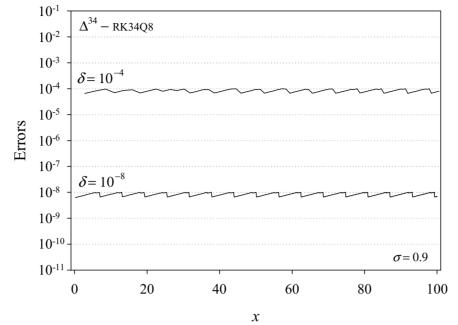


Figure 4. Global errors in RK34Q8, for tolerances  $\delta = 10^{-4}$ ,  $10^{-8}$ , with safety factor  $\sigma = 0.9$ .

136