Noname manuscript No.
(will be inserted by the editor)

The Reliability/Cost Trade-off for a Class of ODE Solvers

W.H. Enright · **Li Yan**

Received: date / Accepted: date

Abstract In the numerical solution of ODEs, it is now possible to develop efficient techniques that will deliver approximate solutions that are piecewise polynomials. The resulting methods can be designed so that the piecewise polynomial will satisfy a perturbed ODE with an associated defect (or residual) that is directly controlled in a consistent fashion. We will investigate the reliability/cost trade off that one faces when implementing and using such methods, when the methods are based on an underlying discrete Runge-Kutta formula.

In particular we will identify a new class of continuous Runge-Kutta methods with a very reliable defect estimator and a validity check that reflects the credibility of the estimate. We will introduce different measures of the "reliability" of an approximate solution that are based on the accuracy of the approximate solution; the maximum magnitude of the defect of the approximate solution; and how well the method is able to estimate the maximum magnitude of the defect of the approximate solution. We will also consider how methods can be implemented to detect and cope with special difficulties such as the effect of round-off error (on a single step) or the ability of a method to estimate the magnitude of the defect when the stepsize is large (as might happen when using a high-order method at relaxed accuracy requests).

Numerical results on a wide selection of problems will be summarized for methods of orders five, six and eight. It will be shown that a modest increase in the cost per step can lead to a significant improvement in the quality of the approximate solutions and the reliability of the method. For example, the numerical results demonstrate that, if one is willing to increase the cost per step by 50%, then a method can deliver approximate solutions where the reported estimated maximum defect is within 1% of its true value on 95% of the steps.

Subject Classification: 65L05, 65L10.
Keywords: Runge-Kutta Methods, initial value problems, defect, error control, continuous methods.

This work was supported by the Natural Sciences and Engineering Research Council of Canada.

Department of Computer Science, University of Toronto, Toronto, ON, Canada, M5S 3G4

1 Introduction and Motivation

Consider an initial value problem (IVP) defined by the system of ordinary differential equations (ODEs),

$$y' = f(x,y),\ \ y(x_0) = y_0,\ \ \text{on}\ \ [x_0,x_F]. \tag{1}$$

A numerical method when applied to (1) will introduce a partitioning $x_0 < x_1 < \cdots < x_N = x_F$ and corresponding discrete approximations $y_0, y_1 \cdots y_N$. The y_i's are usually determined sequentially. To analyze the convergence and accuracy of a numerical method it is convenient to introduce the local error associated with each step. On step i let $z_i(x)$ be the solution of the 'local' IVP,

$$z_i'(x) = f(x,z_i(x)),\ \ z_i(x_{i-1}) = y_{i-1},\ \ \text{on}\ \ [x_{i-1},x_i]. \tag{2}$$

The discrete local error associated with the i^{th} step is then defined to be $z_i(x_i) - y_i$.

A p^{th}-order, s-stage, discrete Runge-Kutta formula, when applied to (1), determines

$$y_i = y_{i-1} + h_i \sum_{j=1}^{s} \omega_j k_j \ \ = z_i(x_i) \ + \ O(h_i^{p+1}),$$

where $h_i = x_i - x_{i-1}$ and the j^{th} stage, k_j, is defined by,

$$k_j = f(x_{i-1} + h_i c_j, y_{i-1} + h_i \sum_{r=1}^{s} a_{jr} k_r).$$

This discrete Runge-Kutta formula is represented by its Butcher tableau:

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \dots & a_{1s} \\ c_2 & a_{21} & a_{22} & \dots & a_{2s} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss} \\ \hline & w_1 & w_2 & \dots & w_s \end{array}, \tag{3}$$

A continuous extension of this discrete formula (CRK) is determined by adding $(\bar{s} - s)$ additional stages to obtain an approximation for any $x \in (x_{i-1}, x_i)$

$$u_i(x) = y_{i-1} + h_i \sum_{j=1}^{\bar{s}} b_j(\tau) k_j, \tag{4}$$

where $\tau = \frac{x - x_{i-1}}{h_i}$ and $b_j(\tau)$ is a polynomial of degree at most $p+1$,

$$b_j(\tau) = \sum_{r=0}^{p+1} \beta_{jr} \tau^r. \tag{5}$$

The additional $(\bar{s} - s)$ stages and the polynomial coefficients β_{jr} are not uniquely determined by the underlying discrete formula and different criteria have been used when determining the most suitable interpolating scheme to define a continuous extension. (See [16] for a detailed discussion of this issue.) The interpolants that interest

us have optimal order (in that they agree with the local solution to $O(h_i^{p+1})$ and their derivatives agree with the derivative of the local solution to $O(h_i^p)$).

We will consider two types of optimal order interpolants, each of which satisfies an equation of the form,

$$u_i(x) = y_{i-1} + h_i \sum_{j=1}^{\bar{s}} b_j(\tau) k_j = z_i(x) + O(h_i^{p+1}).$$

The $[u_i(x)]_{i=1}^N$ define a piecewise polynomial $U(x)$ for $x \in [x_0, x_F]$ and it is this piecewise polynomial that will be considered the numerical solution generated by the CRK method when applied to (1). A simple set of constraints on the $b_j(\tau)$ and on the additional stages ($k_{s+1}, k_{s+2} \cdots k_{\bar{s}}$), (such as $k_{s+1} = f(x_i, y_i)$, $b_j(1) = w_j$ for $j = 1, 2, \cdots, s$ and $b_{s+1}(1) = b_{s+2}(1) = \cdots = b_{\bar{s}}(1) = 0$) will ensure that $u_i(x)$ interpolates y_{i-1}, y_i, $f(x_{i-1}, y_{i-1})$ and $f(x_i, y_i)$ and therefore $U(x) \in C^1[x_0, x_F]$.

The numerical solution $U(x)$ has an associated defect or residual,

$$\begin{aligned} \delta(x) &\equiv f(x, U(x)) - U'(x) \\ &= f(x, u_i(x)) - u_i'(x), \text{ for } x \in [x_{i-1}, x_i]. \end{aligned}$$

It can be shown that, for $U(x) \in C^1[x_0, x_F]$, defined in this way (see [3] for details),

$$\delta(x) = \bar{G}(\tau) h_i^p + O(h_i^{p+1}), \text{ for } x \in [x_{i-1}, x_i], \tag{6}$$

where

$$\bar{G}(\tau) = \bar{q}_1(\tau) F_1 + \bar{q}_2(\tau) F_2 + \cdots + \bar{q}_L(\tau) F_L, \tag{7}$$

and the $\bar{q}_j(\tau)$'s are polynomials in τ that depend only on the CRK formula while the F_j's are constants that depend only on the problem.

Methods can be designed and implemented with the objective of adjusting the stepsize h_i in an attempt to ensure that the maximum magnitude of $\delta(x)$ is bounded by a specified error tolerance, TOL, on each step. From (6) and (7) we see that as $h_i \to 0$ the defect will behave like a linear combination of the $\bar{q}_j(\tau)$ over each $[x_{i-1}, x_i]$. In the special case that $L = 1$ the shape of the defect will be the same (as $h_i \to 0$) for all problems and all steps. That is, the defect will almost always 'converge' to a multiple of $\bar{q}_1(\tau)$. The maximum defect should then occur (as $h_i \to 0$) at the location in $[0, 1]$ of the local extremum of $\bar{q}_1(\tau)$. In this case we can reliably estimate the maximum defect on a step using a single evaluation of the defect at a fixed sample point and we will refer to the defect control strategy as **Strict Defect Control (SDC)**.

In the general case, when $L > 1$, the associated $\bar{G}(\tau)$ defined in (7) can be very different for different problems and for different steps i, and this makes it difficult to choose a fixed sample point τ^* that would give a robust estimate of the maximum defect across the timestep. In this case we have used a value for τ^* that is not near any of the zeros of the $\bar{q}_1(\tau), \bar{q}_2(\tau) \cdots \bar{q}_L(\tau)$ [3] and we refer to this defect control strategy as **Relaxed Defect Control (RDC)**. The two types of continuous extensions

Formula	p	s	$\bar{s}$	$\tilde{s}$
CRK4	4	4	6	8
CRK5	5	6	9	12
CRK6	6	7	11	15
CRK7	7	9	15	20
CRK8	8	13	21	27

Table 1 Cost per step of some specific explicit RDC and SDC CRK formulas we have investgated. The formulas in bold are the ones which we have implemented and tested and which we will discuss in more detail in this paper.

that we consider in detail are $u_i(x)$ corresponding to RDC and $\tilde{u}_i(x)$ corresponding to SDC.

$$RDC: \quad u_i(x) = y_{i-1} + h_i \sum_{j=1}^{\bar{s}} b_j(\tau) k_j = z_i(x) + O(h_i^{p+1}),$$
$$\bar{G}(\tau) = \bar{q}_1(\tau)F_1 + \bar{q}_2(\tau)F_2 + \cdots + \bar{q}_L(\tau)F_L, \quad (8)$$

$$SDC: \quad \tilde{u}_i(x) = y_{i-1} + h_i \sum_{j=1}^{\tilde{s}} \tilde{b}_j(\tau) k_j = z_i(x) + O(h_i^{p+1}), \;\; \tilde{G}(\tau) = \tilde{q}_1(\tau)F_1 \, . \quad (9)$$

The RDC methods that we have implemented and will investigate in detail in subsequent sections correspond to CRK5, CRK6 and CRK8 (see table 1). These underlying CRK formulas have been derived and analysed previously (see [4]) and are investigated here in more detail to help quantify the potential cost and improvements in performance that can be realised when one implements an SDC interpolant for the same discrete RK formula. The SDC implementation of CRK5, CRK6 and CRK8 that we introduce and analyse in this paper are new. We will also discuss some potential difficulties which can arise with our SDC methods and we will propose a modified SDC CRK (denoted SDCV CRK) which, at a modest increase in cost, will detect and address this difficulty.

Note that the analysis, justification and implementation of local interpolants discussed in this investigation applies to general implicit Runge-Kutta formulas represented by (3). If the underlying discrete Runge-Kutta formula is explicit (i.e., $c_1 = 0$ and $a_{jr} = 0$ for $r \geq j$) then no nonlinear equations need be solved on each step and the "cost" of taking a step should be proportional to the number of stages ($\bar{s}$ or $\tilde{s}$) required in the definition of the associated local interpolant (8), (9). The numerical results we present and the extensive testing we have performed are for explicit CRK methods. We have also had limited experience with some implicit CRK methods (see [7] for example) where we have observed similar performance improvements with even less increase in cost.

For a p^{th} order discrete RK formula, we are primarily interested in optimal order (i.e., the corresponding defect is $O(h_i^p)$) local interpolants satisfying (8) or (9). Interpolants with lower order defects can be used effectively in some applications but, when defect error control is used, such methods will not generally be as efficient. Table 1 summarizes the relative costs of some known order p, explicit, RDC and SDC CRK's. It is interesting to observe that, for the orders we are considering, the cost of

an RDC CRK is about 50 % more than that of the underlying discrete formula while the cost of a SDC CRK is about double that of the underlying discrete formula.

For an SDC CRK (9) we know from (6) that the leading coefficient in the asymptotic expansion of the defect satisfies,

$$\tilde{G}(\tau) = \tilde{q}_1(\tau)F_1. \tag{10}$$

We will show that $F_1 h_i^p$ is a multiple of the discrete local error associated with y_i. That is, there is a direct asymptotic relationship between the local error of the underlying discrete RK formula and the continuous defect of any associated SDC CRK. This relationship will be analyzed in more detail in the next section (see also [14] for a discussion of this relationship for a different class of CRKs). Since $\tilde{q}_1(\tau)$ is independent of the problem we should observe as $h_i \to 0$ that the shape of $\tilde{G}$, and consequently the (norm of the) local maximum defect, should be proportional to the fixed polynomial $\tilde{q}_1(\tau)$, independent of the problem and the step i. The value to use for τ^* is then the location of the maximum of $|\tilde{q}_1(\tau)|, \tau \in [0,1]$. In the following sections we verify that this is indeed the case. Note that the "shape of the defect" over a step can be represented by a plot of $\delta(x_{i-1} + \tau h_i)/\delta(x_{i-1} + \tau^* h_i)$ for $0 \le \tau \le 1$ and this plot should approach $\tilde{q}_1(\tau)/\tilde{q}_1(\tau^*)$ for all steps and for all problems as $h_i \to 0$. The only exception would be on those special steps where F_1 is equal to zero or is very small in magnitude. To illustrate this expected behaviour, figure 1 presents an plot of $\delta(\tau)$ vs τ (scaled by its local extremum) for all steps required by a CRK to solve a typical problem with $TOL = 10^{-6}$.

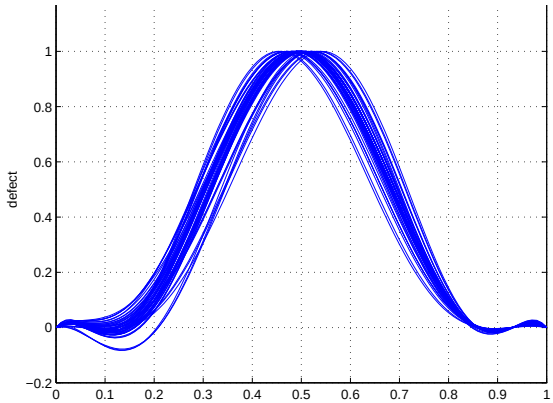

Fig. 1 Plot of defect vs τ (scaled by its local extremum) for each step of a typical $U(x)$ computed by CRK6.

2 Deriving an SDC CRK

One can analyze the error in any CRK by considering the local interpolant, $u_i(x)$, to be an approximation to the local solution $z_i(x)$ for $x \in [x_{i-1}, x_i]$. For this analysis it is

convenient to introduce the interpolant, $\tilde{z}_i(x)$, of degree at most $p+1$, that interpolates the exact local solution at x_{i-1}, x_i, and the derivative of the exact local solution at $x_{i-1}, x_i, x_{i-1}+\mu_r h_i$, for $r = 1,2\cdots p-2$. This approach allows one to decompose the error in $u_i(x)$ into two components – the error inherent in polynomial interpolation (the local interpolation error) and the error that arises as a consequence of "inexact" values being interpolated (the data error). This approach was first used for the analysis of CRKs by Shampine [15] and Gladwell et al. [9] to investigate the associated local errors. It was also subsequently used by Higham ([10] [11]) to investigate the defects and the quality of the defect estimates for a particular class of CRKs.

The polynomial $\tilde{z}_i(x)$ always exists and is unique (if the μ_r's are distinct and not equal to 0 or 1). This polynomial has an associated local interpolation error $LIE_i(x)$ that satisfies:

$$LIE_i(x) \equiv z_i(x) - \tilde{z}_i(x) = \frac{z_i^{(p+2)}(\eta)}{(p+2)!} h_i^{p+2} \tau^2 (\tau-1)^2 \prod_{r=1}^{p-2} (\tau - \mu_r), \qquad (11)$$

for some $\eta \in [x_{i-1}, x_i]$ with $\tau = (x - x_{i-1})/h_i$.

For a prescribed set of interpolation points $(\mu_1, \mu_2 \cdots \mu_{p-2})$, we can represent $\tilde{z}_i(x)$ in terms of the corresponding 'generalized Lagrange basis', $\hat{Q}_j(x),\ j = 1,2\cdots(p+2)$,

$$\begin{aligned}\tilde{z}_i(x) &= \hat{Q}_0(x) z_i(x_{i-1}) + \hat{Q}_1(x) z_i(x_i) + \hat{Q}_2(x) z_i'(x_{i-1}) + \hat{Q}_3(x) z_i'(x_i) \\ &\qquad + \sum_{j=1}^{p-2} \hat{Q}_{3+j}(x) z_i'(x_{i-1} + \mu_j h_i). \\ &= \hat{Q}_0(x) y_{i-1} + \hat{Q}_1(x) z_i(x_i) + \hat{Q}_2(x) f(x_{i-1}, y_{i-1}) + \hat{Q}_3(x) z_i'(x_i) \\ &\qquad + \sum_{j=1}^{p-2} \hat{Q}_{3+j}(x) z_i'(x_{i-1} + \mu_j h_i). \qquad (12)\end{aligned}$$

$\hat{Q}_0(x)$ is the unique polynomial of degree at most $p+1$ satisfying the $p+2$ equations,

$$\begin{aligned}&\hat{Q}_0(x_{i-1}) = 1,\ \hat{Q}_0(x_i) = \hat{Q}_0'(x_{i-1}) = \hat{Q}_0'(x_i) = 0 \\ &\qquad \text{and } \hat{Q}_0'(x_{i-1} + \mu_r h_i) = 0 \text{ for } r = 1,2\cdots(p-2). \qquad (13)\end{aligned}$$

$\hat{Q}_1(x)$ is the unique polynomial of degree at most $p+1$ satisfying the $p+2$ equations,

$$\begin{aligned}&\hat{Q}_1(x_{i-1}) = \hat{Q}_1'(x_{i-1}) = \hat{Q}_1'(x_i) = 0,\ \hat{Q}_1(x_i) = 1 \\ &\qquad \text{and } \hat{Q}_1'(x_{i-1} + \mu_r h_i) = 0 \text{ for } r = 1,2\cdots(p-2). \qquad (14)\end{aligned}$$

$\hat{Q}_2(x)$ is the unique polynomial of degree at most $p+1$ satisfying the $p+2$ equations,

$$\begin{aligned}&\hat{Q}_2(x_{i-1}) = \hat{Q}_2(x_i) = \hat{Q}_2'(x_i) = 0, \hat{Q}_2'(x_{i-1}) = 1, \\ &\qquad \text{and } \hat{Q}_2'(x_{i-1} + \mu_r h_i) = 0 \text{ for } r = 1,2\cdots(p-2). \qquad (15)\end{aligned}$$

$\hat{Q}_3(x)$ is the unique polynomial of degree at most $p+1$ satisfying the $p+2$ equations,

$$\hat{Q}_3(x_{i-1}) = \hat{Q}_3(x_i) = \hat{Q}_3'(x_{i-1}) = 0, \hat{Q}_3'(x_i) \; = \; 1,$$
$$\text{and } \hat{Q}_3'(x_{i-1} + \mu_r h_i) = 0 \text{ for } r = 1, 2 \cdots (p-2). \quad (16)$$

Similarly, for $j = 1, 2 \cdots (p-2)$, $\hat{Q}_{j+3}(x)$ is the unique polynomial of degree at most $p+1$ satisfying,

$$\hat{Q}_{j+3}(x_{i-1}) = \hat{Q}_{j+3}(x_i) = \hat{Q}_{j+3}'(x_{i-1}) = \hat{Q}_{j+3}'(x_i) = 0, \;\; \hat{Q}_{j+3}'(x_{i-1} + \mu_j h_i) = 1,$$
$$\text{and } \; \hat{Q}_{j+3}'(x_{i-1} + \mu_r h_i) = 0 \text{ for } \; r = 1, 2 \cdots (p-2), \;\; r \neq j. \quad (17)$$

In this investigation we will find it convenient to introduce a scaled version of the generalized Lagrange basis to represent $\tilde{z}_i(x)$ and the local interpolants $u_i(x)$ and $\tilde{u}_i(x)$ (where the sequence $(\mu_1, \cdots \mu_{p-2})$ corresponds to a subset of $(c_{s+1}, c_{s+2} \cdots c_{\tilde{s}})$). This representation will be used in the construction, analysis and implementation of the specific SDC CRK formulas we develop. We also find it more convenient to work with the variable τ (rather than x) when defining and implementing the local interpolants. With this change of variable the $\hat{Q}_j(x)$ will be represented as a suitably scaled polynomial in τ and will be independent of i and h_i. To see this let $Q_0(\tau) = \hat{Q}_0(x), Q_1(\tau) = \hat{Q}_1(x)$ and $Q_j(\tau) = (1/h_i)\hat{Q}_j(x)$ for $j = 2, 3 \cdots (p-2)$. The $\hat{Q}_j(x)$ will satisfy equations ((13) - (17)) if the $Q_j(\tau)$ are defined by the following sets of equations (where, in these equations, the symbol $'$ represents $\frac{d}{d\tau}$ rather than $\frac{d}{dx}$) and we know $\frac{d}{dx} Q_j(\tau) = \frac{1}{h_i} Q_j'(\tau)$). Note that each of these sets of equations can be solved in Maple using a short script (see ([5] or [12]) for details).

$Q_0(\tau)$ is defined by,

$$Q_0(0) = 1, Q_0(1) = Q_0'(0) = Q_0'(1) = 0$$
$$\text{and } \; Q_0'(\mu_r) = 0 \text{ for } r = 1, 2 \cdots (p-2). \quad (18)$$

$Q_1(\tau)$ is defined by,

$$Q_1(0) = 0, Q_1(1) = 1, Q_1'(0) = Q_1'(1) = 0$$
$$\text{and } \; Q_1'(\mu_r) = 0 \text{ for } r = 1, 2 \cdots (p-2). \quad (19)$$

$Q_2(\tau)$ is defined by,

$$Q_2(0) = Q_2(1) = Q_2'(1) = 0, Q_2'(0) = 1,$$
$$\text{and } \; Q_2'(\mu_r) = 0 \text{ for } r = 1, 2 \cdots (p-2). \quad (20)$$

$Q_3(\tau)$ is defined by,

$$Q_3(0) = Q_3(1) = Q_3'(0) = 0, Q_3'(1) = 1,$$
$$\text{and } \; Q_3'(\mu_r) = 0 \text{ for } r = 1, 2 \cdots (p-2). \quad (21)$$

For $j = 1,2\cdots(p-2)$, $Q_{j+3}(\tau)$ is defined by,

$$Q_{3+j}(0) = Q_{3+j}(1) = Q'_{3+j}(0) = Q'_{3+j}(1) = 0,\ \ Q'_{3+j}(\mu_j) = 1,$$
$$\text{and } Q'_{3+j}(\mu_r) = 0 \text{ for } r = 1,2\cdots(p-2),\ \ r \neq j. \quad (22)$$

We can then write $\tilde{z}_i(x)$ from (12) as,

$$\tilde{z}_i(x) = Q_0(\tau)y_{i-1} + Q_1(\tau)z_i(x_i) + h_iQ_2(\tau)f(x_{i-1},y_{i-1}) + h_iQ_3(\tau)z'_i(x_i)$$
$$+ h_i\sum_{j=1}^{p-2} Q_{3+j}(\tau)z'_i(x_{i-1}+\mu_j h_i). \quad (23)$$

Typically we will ask that the local interpolants introduced on step i interpolate some of the approximate solution and derivative values introduced on the step. In particular, for the local interpolants we are considering, we assume the first stage satisfies $k_1 = y'_{i-1} = f(x_{i-1},y_{i-1})$ and the first additional stage satisfies $k_{s+1} = y'_i = f(x_i,y_i)$. As an example, consider the polynomial $\bar{u}_i(x)$ of degree at most $p+1$, that interpolates the four values $y_{i-1},y_i,y'_{i-1},y'_i$ as well as the additional $p-2$ derivative approximations,

$$k_r \approx y'(x_{i-1}+c_rh_i),\ \ r = \bar{s},\bar{s}-1\cdots\bar{s}-p+3.$$

When written as a polynomial in τ, we see (from (23)),

$$\bar{u}_i(x_{i-1}+\tau h_i) = Q_0(\tau)y_{i-1} + Q_1(\tau)y_i + h_iQ_2(\tau)k_1 + h_iQ_3(\tau)k_{s+1}$$
$$+ h_i\sum_{r=1}^{p-2} Q_{3+r}(\tau)k_{\bar{s}-p+2+r}\ . \quad (24)$$

That is, $\bar{u}_i(x)$ will interpolate the stages k_1 and k_{s+1} as well as the last $p-2$ stages, $k_{\bar{s}-p+3},k_{\bar{s}-p+4}\cdots k_{\bar{s}}$. Subtracting (24) from (23) we see that, for $x \in (x_{i-1},x_i)$,

$$\tilde{z}_i(x) - \bar{u}_i(x) = Q_1(\tau)(z_i(x_i)-y_i) + Q_3(\tau)(h_iz'_i(x_i) - h_ik_{s+1})$$
$$+ \sum_{r=1}^{p-2} Q_{3+r}(\tau)(h_iz'_i(x_{i-1}+h_ic_{\bar{s}-p+2+r}) - h_ik_{\bar{s}-p+2+r}). \quad (25)$$

Differentiating both sides of (25) with respect to x we obtain,

$$\tilde{z}'_i(x) - \bar{u}'_i(x) = \frac{1}{h_i}q_1(\tau)(z_i(x_i)-y_i) + q_3(\tau)(z'_i(x_i)-k_{s+1})$$
$$+ \sum_{r=1}^{p-2} q_{3+r}(\tau)(z'_i(x_{i-1}+h_ic_{\bar{s}-p+2+r}) - k_{\bar{s}-p+2+r}), \quad (26)$$

where $q_j(\tau) = \frac{d}{d\tau}Q_j(\tau)$ for $j = 1,2\cdots(p+1)$.

There are different approaches that can be used to derive SDC CRK formulas. We will use a "bootstrapping" approach similar to that introduced in [6] to derive higher order local interpolants. Other approaches, such as the direct solution of the

continuous order conditions, are also possible and could lead to a wider class of suitable SDC CRK formulas.

To derive an appropriate SDC CRK formula we will assume we have an RDC CRK formula, $u_i(x)$ satisfying (8). For example, for the particular fifth, sixth and eighth order CRKs we will implement and investigate in the next section, the corresponding RDC CRKs we begin with have been identified and discussed in [5]. For $\tilde{s} = \bar{s} + p - 2$ and any choice of $(p-2)$ abscissa in $(0,1)$, $(c_{\bar{s}+1}, c_{\bar{s}+2}, \cdots c_{\tilde{s}})$ we then define $k_{\bar{s}+j}$ for $j = 1, 2, \cdots (p-2)$ by,

$$k_{\bar{s}+j} \;=\; f(x_{i-1} + c_{\bar{s}+j}h_i, u_i(x_{i-1} + c_{\bar{s}+j}h_i)).$$

Note that $k_{\bar{s}+j}$ will then be an $O(h_i^{p+1})$ approximation to $z_i'(x_{i-1} + c_{\bar{s}+j}h_i)$ if $f(x,y)$ satisfies a Lipschitz condition with respect to y. Therefore $h_i k_{\bar{s}+j}$ will be an $O(h_i^{p+2})$ approximation to $h_i z_i'(x_{i-1} + c_{\bar{s}+j}h_i)$. This is sufficient to ensure that data errors associated with the $k_{\bar{s}+j}$ terms will not contribute to the leading coefficient in the expansion of the defect of $\bar{u}_i(x)$. To see this consider the defect $\bar{\delta}(x)$ defined by

$$f(x,\bar{u}_i(x)) - \bar{u}_i'(x) = [f(x,\bar{u}_i(x)) - f(x,z_i(x))] + [f(x,z_i(x)) - z_i'(x)] \;+\; [z_i'(x) - \bar{u}_i'(x)]. \quad (27)$$

Since the second term in the RHS of (27) is zero, we can expand each of the other two terms to obtain,

$$\bar{\delta}(x) = [f(x,\bar{u}_i(x)) - f(x,\tilde{z}_i(x))] \;+\; [f(x,\tilde{z}_i(x)) - f(x,z_i(x))] + [z_i'(x) - \tilde{z}_i'(x)] \;+\; [\tilde{z}_i'(x) - \bar{u}_i'(x)]. \quad (28)$$

Of the four terms comprising the RHS of (28), the first term is $O(h_i^{p+1})$ (from (25)), the second term is $O(h_i^{p+2})$ (from (11)), and the third term is $O(h_i^{p+1})$ (from (11)) and standard interpolation theory. The fourth term (from (26)) can be written as,

$$[\tilde{z}_i'(x) - \bar{u}_i'(x)] = q_1(\tau)\frac{z_i(x_i) - y_i}{h_i} + O(h_i^{p+1}),$$

and we observe that the $O(h_i^{p+1})$ contribution arising from this term will be directly related to the $q_j(\tau), j = 3, 4 \cdots p+1$, but there will be other contributions as well that arise from the first and third terms. We then have that the defect $\bar{\delta}(x)$ will satisfy,

$$\bar{\delta}(x) = q_1(\tau)F_1 h_i^p \;+\; (\hat{q}_1(\tau)\hat{F}_1 + \hat{q}_2(\tau)\hat{F}_2 + \cdots\cdots \hat{q}_{\hat{L}}(\tau)\hat{F}_{\hat{L}})h_i^{p+1} + O(h_i^{p+2}), \quad (29)$$

where $F_1 h_i^p = z_i(x_i) \; - y_i$, the discrete local error associated with the current step, and the set of polynomial coefficients $[q_2(\tau), q_3(\tau) \cdots q_{p+1}(\tau)]$, (defined in (26), are contained in the set $[\hat{q}_1(\tau), \hat{q}_2(\tau) \cdots \hat{q}_{\hat{L}}(\tau)]$. This will be true for any choice of the parameters $c_{\bar{s}+1}, c_{\bar{s}+2} \cdots c_{\tilde{s}}$. The particular choice we make in our development of a suitable SDC CRK is motivated by an attempt to avoid some of the potential difficulties identified in the next section.

2.1 Modified Defect Control Strategy : SDCV

Let $\tilde{u}_i(x)$ be a given SDC CRK defined by (9) and satisfying (29). There are two potential deficiencies of this formula that could affect the cost of the method and/or the reliability of the estimate $\|\bar{\delta}(\tau^*)\|$,

- $q_1(\tau)$ may have a large maximum value. (Note that its 'average' value must be one since $q_1(0) = q_1(1) = 0$ and $Q_1(1) = 1 = \int_0^1 q_1(\tau)d\tau$.)
- The $\hat{q}_j(\tau)$ may be large in magnitude relative to $q_1(\tau)$ (and therefore h_i would have to be small before the estimate is justified). (That is, before $|h_i\hat{q}_j(\tau)| << |q_1(\tau)|$.)

These two potential difficulties were addressed by performing a search (for each value of p) over the $p-2$ free parameters to identify a suitable SDC CRK. For each set of free parameters we determined

$$D_j = \max_{\tau\in[0,1]} |q_j(\tau)|, j = 1,2\cdots p+1,$$

where the $q_j(\tau)$ are defined in (26). The corresponding SDC CRK was considered to be suitable if D_1 was not much larger than 2 and $D_j/D_1 < 1$ for $j = 3,4\cdots p+1$. Such a requirement will at least ensure that some of the $|\hat{q}(\tau)|$ in (29) will be small relative to D_1.

There are two other situations that can arise on some problems on isolated steps for any SDC CRK, which can result in an unreliable defect estimate. The first situation arises when $|F_1|$ is zero or very small in magnitude. Extensive testing of suitable SDC CRKs revealed that, on the very few steps where the estimate was too small, $|F_1|$ was inevitably near zero. In this case the first term of (29) is nearly zero and the contributions to the defect arise from a combination of $O(h_i^{p+1})$ terms. On these isolated steps, the actual maximum defect was usually smaller than TOL, but the estimated value was not close to the true maximum. We introduced a validity check (or confirmation check) to detect this situation. This validity check, as we explain below, involves two additional defect samples. If this check is satisfied on all steps, one can have increased confidence in the reliability of the integration. We then modified the underlying SDC defect control strategy so that, when this check is not satisfied, two extra sampled defect evaluations are performed to determine a more appropriate estimate of the maximum defect. These two extra samples, together with the first sample at τ^* and the two from the validity check, give us five samples of the defect, and we choose the largest of these to be our estimate of the maximum defect. We call this modified defect control strategy **SDCV**.

To justify the particular validity check we have adopted, we observe that $q_1(\tau)$ is zero at $\tau = 0, \tau = 1$ and attains its maximum magnitude at τ^*. Let $\tau_1 < \tau^*$ and $\tau_2 > \tau^*$ satisfy

$$q_1(\tau_1) = q_1(\tau_2) = q_1(\tau^*)/2.$$

If we then let $R_1 = \frac{\delta(x_{i-1}+\tau_1 h)}{\delta(x_{i-1}+\tau^* h)}$ and $R_2 = \frac{\delta(x_{i-1}+\tau_2 h)}{\delta(x_{i-1}+\tau^* h)}$, then as $h \to 0$ we expect both R_1 and R_2 to approach $1/2$. We compute these two ratios and consider the validity check

to be satisfied if both are close to $1/2$. In our tests we interpreted "close to $1/2$" to mean "in the range $[.3,.7]$".

The second situation, where the estimate of the maximum defect for a suitable SDC CRK may be unreliable, arises on problems where round-off errors are not dominated by the local errors in the various computations associated with a single step. The polynomial coefficients defining a CRK (the $b_j(\tau)$ and $\tilde{b}_j(\tau)$ in (8) and (9)) can be large in magnitude, particularly for $p \geq 6$. In addition, the defect is defined and estimated as the difference between quantities that must be "almost equal". This can lead to large relative errors (due to the accumulated affects of round-off error) when evaluating the interpolant and when evaluating the defect. This issue is investigated in detail in [5] where it is observed that, if we know in advance the values $0 \leq \tau_1 \leq \tau_2 \cdots \tau_k \leq 1$ where the method will be evaluating either the interpolant, its derivative, or the associated defect, then the polynomial coefficients (evaluated at these special points) can be computed to full precision and stored internally in the method. It is easy to do this for the CRKs we have implemented as the defect is only evaluated at one point (for a RDC CRK or an SDC CRK) or at three points (for an SDCV CRK).

On the other hand, when a user is interested in sampling the interpolant, its derivative, or the defect at an arbitrary point $x \in [x_{i-1}, x_i]$, the total computation must be performed at the working precision and the contribution of round-off error to the returned approximation will be greater. As an example, figure 2 displays a plot of the computed scaled defect for a sixth order SDC CRK evaluated at a fine mesh of 100 sampled points over a single step when $TOL = 10^{-12}$. The plot on the left corresponds to the case where the evaluation of the polynomial coefficients is performed in double precision, while the plot on the right corresponds to the case where these coefficients are evaluated in extended precision (but the remainder of the computation performed in double precision). As expected, the effect of round-off error is greater as $\tau \rightarrow 1$ and the contribution to the error that is due to round-off tends to oscillate in sign and grow in magnitude as τ increases. This suggests that a credible indication or signal that round-off errors cannot be ignored would be if a sampled defect value close to $\tau = 1$ has a magnitude that is comparable to $|\delta(x_{i-1} + \tau^* h_i)|$. What is not clear is what action should be taken when this signal is raised. Alternatives would include halting the integration with a warning suggesting that a higher precision implementation of the method or a lower order method be used. Note that this difficulty, arising from round-off errors introduced on a single step, is relatively easy to detect and will usually only arise at stringent accuracy requests (for example, when $TOL < 10^{-10}$ in double precision).

3 Implementation and testing

We derived new SDC CRKs and implemented the resulting methods for several underlying RDC CRKs including SDC CRK5 which is based on the discrete 5^{th}-order Runge-Kutta formula that is implemented as `ode45` in *Matlab*. We also implemented SDC CRK6 and SDC CRK8 based on RDC CRK6 and RDC CRK8 which were introduced and investigated in [5]. These particular RDC CRKs are themselves based on underlying discrete Runge-Kutta formulas derived by Verner [16] as the higher

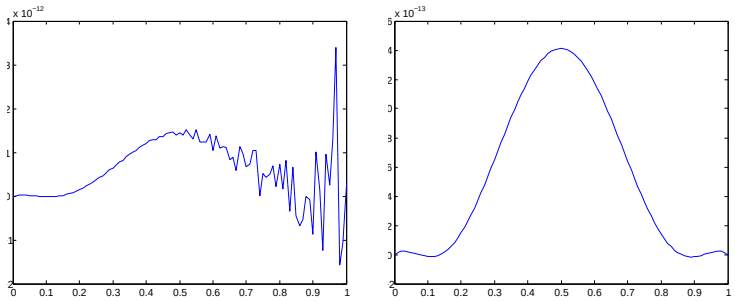

Fig. 2 Plot of defect vs τ (scaled by its local extremum) for SDC CRK6 on a typical step where round-off error is comparable to truncation error as $\tau \rightarrow 1.0$. Plot on left is when the defect is evaluated in double precision while the plot on right is when the same defect is evaluated using extended precision.

order formula of an order $(5,6)$ formula pair and the higher order formula of an order $(7,8)$ formula pair. Each method was extensively tested using the DETEST package [8], which assesses the performance of a method on a suite of 25 problems. We modified the testing package so that it would determine and report measures of the cost and reliability of an approximate solution, as well as more direct measures of the reliability of the estimate of the maximum defect. The performance assessment summaries that we report for each method then has two components:

- Two measures of how well a **Method** controls the maximum magnitude of the defect. We report the ratio of the maximum defect to TOL over all steps and the fraction of steps where this ratio is greater than 1.
- Two measures of how well the **Estimate** of the maximum defect reflects its true value. We determine the ratio of the true maximum defect to the estimated value on each step and report the maximum of this ratio over all steps. We also report the the fraction of steps where the estimated maximum is within one percent of the true maximum.

We have implemented and tested three versions of each CRKp: corresponding to RDC CRKp, SDC CRKp and SDCV CRKp (where $p = 5,6,8$). The user selects the defect control strategy by setting an integer parameter. We have run all versions on the 25 test problems of DETEST (all non-stiff), at 9 tolerances from 10^{-1} to 10^{-9}. We report performance summaries over all problems (for a given TOL), but detailed results are available for each method on each problem. We report two measures of cost: NSTP (the number of steps) and NFCN (the number of derivative evaluations), two measures of the reliability of the method: DMAX and Frac-D (maximum ratio over all steps of the true maximum defect to TOL and the fraction of steps where this ratio exceeded one), and two measures of the reliability of the estimate: R-Max and Frac-G (maximum ratio over all steps of the true maximum defect to the estimate and the fraction of steps where this ratio was bounded by 1.01).

0	0						
1/5	1/5	0					
3/10	3/40	9/40	0				
4/5	44/45	−56/15	32/9	0			
8/9	19372/6561	−25360/2187	64448/6561	−212/729	0		
1	9017/3168	−355/33	46732/5247	49/176	−5103/18656	0	
1	35/384	0	500/1113	125/192	−2187/6784	11/84	0
	35/384	0	500/1113	125/192	−2187/6784	11/84	**0**

Table 2 The tableau corresponding to *Matlab*'s `ode45`. The first six rows correspond to the discrete tableau. The last row defines the extra stage from which *Matlab* builds an interpolant with associated local error that is O(h^5) and defect that is $O(h^4)$.

3.1 CRK5

The tableau for the well-known discrete order $(4,5)$ Runge-Kutta formula pair used by *Matlab*'s `ode45` is shown in Table 2. The first six rows represent the tableau and define $k_1,\ldots,k_6$ for the standard, discrete solution.

The $O(h_i^5)$ interpolant (with a defect that is $O(h_i^4)$) used in `ode45` is defined by

$$\begin{pmatrix} \hat{b}_1(\tau) \\ \hat{b}_2(\tau) \\ \hat{b}_3(\tau) \\ \hat{b}_4(\tau) \\ \hat{b}_5(\tau) \\ \hat{b}_6(\tau) \\ \hat{b}_7(\tau) \end{pmatrix} = \begin{pmatrix} 1 & -183/64 & 37/12 & -145/128 \\ 0 & 0 & 0 & 0 \\ 0 & 1500/371 & -1000/159 & 1000/371 \\ 0 & -125/32 & 125/12 & -375/64 \\ 0 & 9477/3392 & -729/106 & 25515/6784 \\ 0 & -11/7 & 11/3 & -55/28 \\ 0 & 3/2 & -4 & 5/2 \end{pmatrix} \begin{pmatrix} \tau \\ \tau^2 \\ \tau^3 \\ \tau^4 \end{pmatrix}. \qquad (30)$$

This defines the standard local interpolant on step i,

$$\hat{u}_i(x_{i-1}+\tau h_i) = y_{i-1} + h_i \sum_{j=1}^{7} \hat{b}_j(\tau) k_j.$$

To construct an $O(h_i^6)$ RDC interpolant $u_i(t)$, we add 2 more stages (see [2] and [3] for a justification of these particular choices),

$$k_8 = f(x_{i-1}+0.86h_i, \hat{u}_i(x_{i-1}+0.86h_i)),$$

$$k_9 = f(x_{i-1}+0.93h_i, \hat{u}_i(x_{i-1}+0.93h_i)),$$

and then the RDC interpolant, $u_i(x)$ depends on 9 stages with coefficients defined by,

$$\begin{pmatrix} b_1(\tau) \\ b_2(\tau) \\ b_3(\tau) \\ b_4(\tau) \\ b_5(\tau) \\ b_6(\tau) \\ b_7(\tau) \\ b_8(\tau) \\ b_9(\tau) \end{pmatrix} = \begin{pmatrix} 1 & -\frac{1708582621}{524156928} & \frac{1232939669}{262078464} & -\frac{1663764925}{524156928} & \frac{208375}{253952} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{499875}{94976} & -\frac{1618625}{142464} & \frac{871875}{94976} & -\frac{15625}{5936} \\ 0 & \frac{499875}{65536} & -\frac{1618625}{98304} & \frac{871875}{65536} & -\frac{15625}{4096} \\ 0 & -\frac{26237439}{6946816} & \frac{28319463}{3473408} & -\frac{45762975}{6946816} & \frac{820125}{434176} \\ 0 & \frac{43989}{28672} & -\frac{142439}{43008} & \frac{76725}{28672} & -\frac{1375}{1792} \\ 0 & -\frac{2291427}{100352} & \frac{3838251}{50176} & -\frac{8579075}{100352} & \frac{199625}{6272} \\ 0 & -\frac{47953125}{1078784} & \frac{74828125}{539392} & -\frac{155453125}{1078784} & \frac{78125}{1568} \\ 0 & \frac{8734375}{145824} & -\frac{14359375}{72912} & \frac{31234375}{145824} & -\frac{234375}{3038} \end{pmatrix} \begin{pmatrix} \tau \\ \tau^2 \\ \tau^3 \\ \tau^4 \\ \tau^5 \end{pmatrix}, \quad (31)$$

with an associated defect that is $O(h_i^5)$,

$$u_i(x_{i-1}+\tau h_i) = y_{i-1} + h_i \sum_{j=1}^{9} b_j(\tau) k_j.$$

To determine a suitable SDC CRK from this RDC CRK we used the approach justified in section 2.1 to search for c_{10}, c_{11}, c_{12} such that the corresponding D_j satisfies the two constraints; D_1 is not much greater than 2 and $D_j/D_1 < 1$ for $j = 3, 4 \cdots (p+1)$. Figure 3 displays the plots of the polynomials $q_1(\tau), q_2(\tau) \cdots q_6(\tau)$ for $c_{10} = .10, c_{11} = .80, c_{12} = .90$ which is the choice we have used to define SDC CRK5. With this choice, we compute the three new stages,

$$\begin{aligned} k_{10} &= f(x_{i-1} + 0.1h_i, u_i(x_{i-1} + 0.1h_i)), \\ k_{11} &= f(x_{i-1} + 0.8h_i, u_i(x_{i-1} + 0.8h_i)), \\ k_{12} &= f(x_{i-1} + 0.9h_i, u_i(x_{i-1} + 0.9h_i)). \end{aligned}$$

The SDC interpolant $\tilde{u}_i(x_{i-1} + \tau h_i)$ is then,

$$\tilde{u}_i(x_{i-1}+\tau h_i) \;=\; y_{i-1} + h_i \sum_{j=1}^{12} \tilde{b}_j(\tau) k_j, \quad (32)$$

where the $\tilde{b}_j(\tau)$ can be explicitly expressed in terms of the generalized Lagrange coefficients introduced in the previous section. To see this, with $p = 5, \mu_1 = .1, \mu_2 = .8, \mu_3 = .9$ we can write $\tilde{u}_i(x)$ as,

$$\begin{aligned} \tilde{u}_i(x_{i-1}+\tau h_i) \;=\; & Q_0(\tau) y_{i-1} + Q_1(\tau) y_i + Q_2(\tau) h_i y'_{i-1} + Q_3(\tau) h_i y'_i \\ & + h_i \sum_{j=1}^{3} Q_{3+j}(\tau) k_{9+j}. \quad (33) \end{aligned}$$

Noting that $Q_0(\tau) = 1 - Q_1(\tau)$, $y'_{i-1} = k_1, y'_i = k_7$ and that $(w_1, w_2, \cdots w_6)$ are prescribed by the last row of Table 2, we have (from 33),

$$\tilde{u}_i(x_{i-1} + \tau h_i) \;=\; (1 - Q_1(\tau))y_{i-1} + Q_1(\tau)(y_{i-1} + h_i \sum_{j=1}^{6} w_j k_j) \\ +\; Q_2(\tau)h_i k_1 \;+\; Q_3(\tau)h_i k_7 \;+\; h_i \sum_{j=1}^{3} Q_{3+j}(\tau)k_{9+j}, \quad (34)$$

which after re-arranging terms is,

$$\tilde{u}_i(x_{i-1} + \tau h_i) \;=\; y_{i-1} + h_i(w_1 Q_1(\tau) + Q_2(\tau))k_1 + h_i \sum_{j=2}^{6} w_j Q_1(\tau)k_j + \\ h_i Q_3(\tau)k_7 + h_i \sum_{j=1}^{3} Q_{3+j}(\tau)k_{9+j}. \quad (35)$$

Since this interpolating polynomial is unique we can equate the coefficient polynomials in (35) and (32) and observe that,

$$\tilde{b}_1(\tau) = w_1 Q_1(\tau) + Q_2(\tau). \quad (36)$$
$$\tilde{b}_{j+1}(\tau) = w_{j+1} Q_1(\tau) \;\text{ for }\; j = 1, 2, \cdots 5, \quad (37)$$
$$\tilde{b}_7(\tau) = Q_3(\tau), \quad (38)$$
$$\tilde{b}_8(\tau) = \tilde{b}_9(\tau) \;=\; 0, \quad (39)$$
$$\tilde{b}_{9+j}(\tau) = Q_{9+j}(\tau) \;\text{ for }\; j = 1, 2, 3. \quad (40)$$

Solving for the $Q_j(\tau)$ by applying Maple to solve equations (18)- (22) (as described in section 2), we then use (36)-(40) to obtain,

$$\begin{pmatrix} \tilde{b}_1(\tau) \\ \tilde{b}_2(\tau) \\ \tilde{b}_3(\tau) \\ \tilde{b}_4(\tau) \\ \tilde{b}_5(\tau) \\ \tilde{b}_6(\tau) \\ \tilde{b}_7(\tau) \\ \tilde{b}_8(\tau) \\ \tilde{b}_9(\tau) \\ \tilde{b}_{10}(\tau) \\ \tilde{b}_{11}(\tau) \\ \tilde{b}_{12}(\tau) \end{pmatrix} = \begin{pmatrix} 1 & -\frac{13303}{1584} & \frac{791347}{28512} & -\frac{1589515}{38016} & \frac{35045}{1188} & -\frac{113375}{14256} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\frac{12000}{4081} & \frac{962000}{36729} & -\frac{6725000}{12243} & \frac{80000}{1749} & -\frac{500000}{36729} \\ 0 & -\frac{375}{88} & \frac{60125}{1584} & -\frac{168125}{2112} & \frac{4375}{66} & -\frac{15625}{792} \\ 0 & \frac{19683}{9328} & -\frac{350649}{18656} & \frac{2941515}{74624} & -\frac{76545}{2332} & \frac{91125}{9328} \\ 0 & -\frac{6}{7} & \frac{481}{63} & -\frac{1345}{84} & \frac{40}{3} & -\frac{250}{63} \\ 0 & \frac{62}{33} & -\frac{16099}{891} & \frac{14095}{297} & -\frac{14620}{297} & \frac{16000}{891} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{2500}{231} & -\frac{304250}{6237} & \frac{170750}{2079} & -\frac{127250}{2079} & \frac{106250}{6237} \\ 0 & \frac{375}{56} & -\frac{15875}{252} & \frac{26125}{168} & -\frac{3125}{21} & \frac{3125}{63} \\ 0 & -\frac{500}{99} & \frac{43750}{891} & -\frac{39250}{297} & \frac{40750}{297} & -\frac{43750}{891} \end{pmatrix} \begin{pmatrix} \tau \\ \tau^2 \\ \tau^3 \\ \tau^4 \\ \tau^5 \\ \tau^6 \end{pmatrix}.$$

Table 3 displays the summary statistics for the numerical tests of the three versions of this CRK5.

TOL	CRK	NSTP	NFCN	DMAX	Frac-D	R-Max	Frac-G
10^{-2}	RDC	609	7153	2.37	.199	18.85	.18
	SDC	623	9853	1.02	.003	8.12	.63
	SDCV	625	11709	0.97	.000	1.05	.67
10^{-4}	RDC	1070	12130	5.89	.179	126.82	.14
	SDC	1065	16081	1.60	.005	7.12	.73
	SDCV	1065	19033	1.01	.001	1.12	.78
10^{-6}	RDC	2176	23146	5.44	.233	55.44	.09
	SDC	2095	30037	1.44	.007	11.49	.83
	SDCV	2099	35703	1.01	.002	1.08	.86
10^{-8}	RDC	4929	46051	21.28	.354	207.40	.07
	SDC	4562	56953	1.24	.003	32.80	.94
	SDCV	4566	66937	1.01	.001	1.07	.95

Table 3 Results on the 25 DETEST Problems for CRK5 for the three defect control strategies

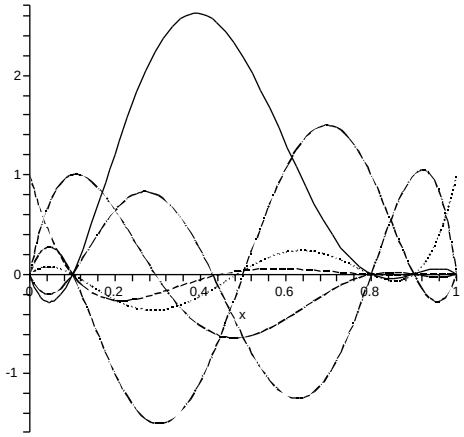

Fig. 3 Plots of q_1 through q_6 for SDC CRK5. q_1 is represented by the solid line and has the highest magnitude among all q_j. The abscissa vector is [.10, .80, .90]. The maximum ratio of $D_j(j = 3,4\cdots 6)$ to D_1 is 0.57. The maximum of q_1 occurs at $\tau^* = 0.389$, $\tau \in [0,1]$ and the values required to define the validity check for SDCV are $\tau_1 = .207, \tau_2 = .600$.

3.2 CRK6

The tableau for an effective 6^{th}-order discrete Runge-Kutta formula is presented in [16] and an associated RDC CRK is justified and implemented in [4]. There are $s = 7$ stages required to define the underlying discrete 6^{th}-order formula. One additional stage results in a non-optimal interpolant of $O(h_i^6)$ accuracy, an additional three stages are used to define the RDC CRK $u_i(x)$ with $\bar{s} = 11$, and $\tilde{s} = 15$ stages are then used to define the SDC CRK, $\tilde{u}_i(x)$.

A search for a suitable abcissa vector, $[c_{12}, c_{13}, c_{14}, c_{15}]$ led us to identify the choice $[.07, .14, .86, .93]$. Figure 4 displays the corresponding plots of the coefficients that form the leading terms in the expansion of the defect for this SDC CRK6. Table 4 displays the summary statistics for the numerical tests of the three versions of this CRK6.

3.3 CRK8

The tableau for an effective 8^{th}-order discrete Runge-Kutta formula is prescribed in [16] and an associated RDC CRK justified and implemented in [4]. There are $s = 13$ stages required to define the underlying discrete formula, $\bar{s} = 21$ stages to define the RDC CRK $u_i(x)$, and $\tilde{s} = 27$ stages to define the SDC CRK.

A search for a suitable abcissa vector led us to the choice $[.07, .14, .21, .79, .86, .93]$. Figure 5 displays the corresponding plots of the coefficients that form the leading

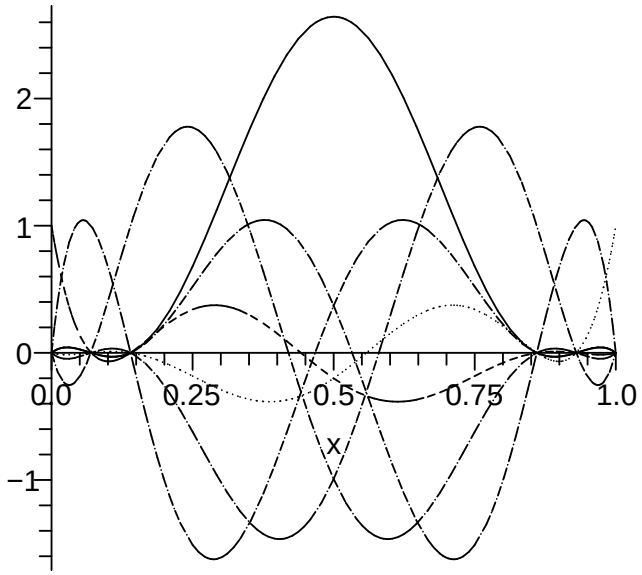

Fig. 4 Plots of q_1 through q_7 for SDC CRK6. q_1 is represented by the solid line and has the highest magnitude among all q_j. The abscissa vector is [.07, .14, .86, .93]. The maximum ratio of $D_j (j = 3, 4 \cdots 7)$ to D_1 is 0.67. The maximum of q_1 occurs at $\tau^* = .500, \tau \in [0,1]$ and the values required to define the validity check for SDCV are $\tau_1 = .311, \tau_2 = .689$.

TOL	CRK	NSTP	NFCN	DMAX	Frac-D	R-Max	Frac-G
10^{-2}	RDC	552	7879	5.27	.176	23.25	.50
	SDC	547	10585	1.00	.000	1.74	.70
	SDCV	549	12300	1.00	.000	1.43	.71
10^{-4}	RDC	955	13082	4.87	.144	15.34	.55
	SDC	929	17305	4.90	. 003	18.90	.87
	SDCV	931	19819	1.00	.001	1.08	.87
10^{-6}	RDC	1789	23499	10.75	.103	112.90	.59
	SDC	1748	30925	1.01	.001	1.81	.96
	SDCV	1748	35073	1.01	.001	1.08	.96
10^{-8}	RDC	3622	43288	6.48	.098	1286.90	.67
	SDC	3547	57460	1.01	.001	1.14	.98
	SDCV	3547	65148	1.01	.001	1.07	.98

Table 4 Results on the 25 DETEST Problems for CRK6 for the three defect control strategies

terms in the expansion of the defect for this SDC CRK8. Table 5 displays the summary statistics for the numerical tests of the three versions of this CRK8.

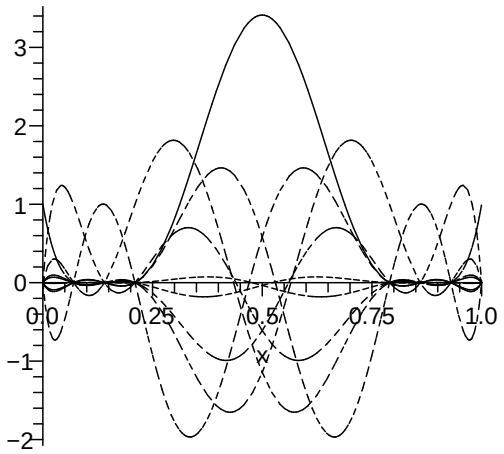

Fig. 5 Plots of q_1 through q_9 for SDC CRK8. q_1 is represented by the solid line and has the highest magnitude among all q_j. The abscissa vector is [0.07, 0.14, 0.21, 0.79, 0.86, 0.93]. The maximum ratio of $D_j(j = 3,4\cdots 9)$ to D_1 is 0.53. The maximum of q_1 occurs at $\tau^* = .500, \tau \in [0,1]$ and the values required to define the validity check for SDCV are $\tau_1 = .353, \tau_2 = .647$.

TOL	CRK	NSTP	NFCN	DMAX	Frac-D	R-Max	Frac-G
10^{-2}	RDC	337	8745	10.68	.213	36.71	.30
	SDC	332	11439	7.16	.009	30.50	.35
	SDCV	333	12793	1.01	.003	1.65	.35
10^{-4}	RDC	495	13285	7.70	.139	32.70	.17
	SDC	466	15781	1.02	. 002	4.34	.45
	SDCV	465	17319	1.05	.004	1.47	.45
10^{-6}	RDC	715	18245	6.09	.126	134.32	.10
	SDC	707	23425	3.01	.008	22.70	.58
	SDCV	712	26253	1.02	.001	1.34	.59
10^{-8}	RDC	1095	27065	31.12	.179	409.09	.08
	SDC	1081	34787	1.86	.005	20.80	.62
	SDCV	1081	38251	1.12	.007	2.60	.62

Table 5 Results on the 25 DETEST Problems for CRK8 for the three defect control strategies

4 Observations and Conclusions

In this paper, we have investigated several explicit, continuous Runge-Kutta methods in which the maximum defect across a timestep can be reliably and efficiently monitored and controlled. We have analyzed interpolants whose defect, in the limit of stringent values of TOL, has a predicted "shape" dependent only on the order of the discrete formula and not on the problem being integrated. Although methods with similar characteristics have been discussed in the past, the methods we have investigated are different in that the order of the defect is optimal relative to the order of the discrete formula. This results in a method in which the error control is both efficient and theoretically justified for all problems.

The numerical results presented in Section 3 are summaries (for three values of TOL) of the detailed performance assessment of each method on all 25 non-stiff test problems of the DETEST package [8]. The detailed statistics for each method on each problem are presented in [12] where several additional measures of performance that quantify the observed relationship between the global error and the prescribed TOL are presented. Note that some of these measures are not easy to report in meaningful summaries as the relationship between the global error and TOL is very sensitive to the problem. The detailed statistics do confirm that, from the point of view of returning an approximate solution whose error is bounded by a multiple of TOL, all the methods tested performed well. That is, the approximate solution generally satisfied,

$$||y(x) - U(x)|| \leq K(x)TOL,$$

where $K(x)$ depended primarily on the problem and was insensitive to the order of the method (or the number of steps used to compute the approximate solution).

We see from the summaries reported in Section 3 that, for a given TOL, the number of steps required to solve all 25 problems was not very sensitive to the local interpolant used (RDC, SDC or SDCV). The number of stages required per step for each of the defect control schemes is then a good predictor of the relative costs of computing $U(x)$ for a given TOL. Note that, when solving a given problem with the same underlying discrete RK formula, the RDC interpolant and the SDC interpolant will be different. The magnitude of the leading coefficient in the expansion of the respective defects will likely be smaller for the SDC interpolant than for the RDC interpolant and this can result in fewer steps for an SDC CRK method to solve the problem with an associated smaller maximum defect than that associated with the RDC CRK.

Looking closely at the summary results for the RDC methods we observe that we obtain a level of reliability that might well be considered acceptable for most applications at all tolerances. The maximum defect exceeds TOL on ten to twenty percent of the steps but it rarely exceeds 10 TOL. The SDC methods (without a validity check) are much more reliable with the maximum defect exceeding TOL on less than one percent of the steps and never exceeding 10 TOL. In addition, the defect estimate is within one percent of the true maximum defect (over each step) most of the time. The extra cost of the SDC methods (relative to the RDC versions of the same discrete formula) is generally no greater than twenty-five percent. Finally, with

the validity check, the SDCV methods were able to detect and deal with the very few steps where the observed maximum defect was not well estimated by the corresponding SDC methods. For the SDCV methods the maximum defect is never larger than 1.2 *TOL* and R-Max is never very large. (Note that on some problems, see table 5 for example, with the higher order CRKs, the fraction deceived was greater for the SDCV method than for the corresponding SDC method. One possible explanation for this is that the number of deceived steps is very small and any increase at all will be reported as a distractingly large increase when reported as a fraction of the total number of steps.) The extra cost of the SDCV methods (relative to the RDC versions of the same discrete formula) is generally no greater than fifty percent.

Now that we have developed a class of very reliable SDC CRKs for IVPs, we are investigating how effective this approach will be to develop improved methods for other classes of ODEs. For example we are currently investigating the use of SDC CRKs in methods for delay differential equations, boundary value problems and Volterra integral differential equations. We hope, in the future, to drive and implement asymptotically correct defect estimates for multistep methods (in particular for those based on Adams or BDF formulas).

Acknowledgements The authors would like to thank P. Muir, W. Hayes and M. Shakourifar for their helpful discussions during the preparation of this paper. The authors would also like to thank the referees whose comments and suggestions have improved the presentation.

References

1. J. R. Dormand and P. J. Prince. A family of embedded Runge-Kutta formulae. *J. Comp. Appl. Math.*, 6:19–26, 1980.
2. W. H. Enright. Analysis of error control stategies for continuous Runge-Kutta methods. *SIAM Journal on Numerical Analysis*, 26:588–599, 1989.
3. W. H. Enright. A new error-control for initial value solvers. *App. Math. Comp.*, 31:288–301, 1989.
4. W. H. Enright. The relative efficiency of alternative defect control schemes for high-order continuous Runge-Kutta formulas. *SIAM Journal on Numerical Analysis*, 30:1419–1445, 1993.
5. W. H. Enright and W. B. Hayes. Robust and reliable defect control for Runge-Kutta methods. *ACM Trans. Math. Soft.*, 33:1-19, 2007.
6. W. H. Enright, K. R. Jackson, S. P. Nϕrsett, and P. G. Thomsen. Interpolants for Runge-Kutta formulas. *ACM Trans. Math. Soft.*, 12:193–218, 1986.
7. W. H. Enright and P. H. Muir. New interpolants for asymptotically correct defect control of BVODEs. to appear in Num. Alg., accepted by eds., January 2008.
8. W. H. Enright and J. D. Pryce. Two FORTRAN packages for assessing initial value methods. *ACM Trans. Math. Soft.*, 13:1–27, 1987.
9. I. Gladwell, L. F. Shampine, L. S. Baca, and R. W. Brankin. Practical aspects of interpolation in Runge-Kutta codes. *SIAM J. Sci. Statist. Comput.*, 8:322-341, 1987.
10. D. J. Higham. Robust defect control with Runge-Kutta schemes. *SIAM J. Numer. Anal.*, 26:1175-1183, 1989.
11. D. J. Higham. Runge-Kutta defect control using Hermite-Birkhoff interpolation. *SIAM J. Sci. Statist. Comput.*, 12:991-999, 1991.
12. L. Yan. Robust and reliable defect control for Runge-Kutta methods. MSc Thesis, Department of Computer Science, University of Toronto, 2006.
13. Matlab 6. MathWorks. Natick, MA, 2000.
14. L. F. Shampine. Solving ODEs and DDEs with residual control. *Appl. Numer. Math.*, 52:113-127, 2005.

15. L. F. Shampine. Interpolation for Runge-Kutta methods. *SIAM J. Numer. Anal.*, 22:1014-1027, 1985.
16. J. H. Verner. Differentiable interpolants for high-order Runge-Kutta methods. *SIAM J. Numer. Anal.*, 30:1446-1466, 1993.