Reducing the Uncertainty when Approximating the Solution of ODEs

W.H. Enright *

Department of Computer Science
University of Toronto
Canada

Abstract. One can reduce the uncertainty in the quality of an approximate solution of an ordinary differential equation (ODE) by implementing methods which have a more rigorous error control strategy and which deliver an approximate solution that is much more likely to satisfy the expectations of the user. We have developed such a class of ODE methods as well as a collection of software tools that will deliver a piecewise polynomial as the approximate solution and facilitate the investigation of various aspects of the problem that are often of as much interest as the approximate solution itself. We will introduce measures that can be used to quantify the reliability of an approximate solution and discuss how one can implement methods that, at some extra cost, can produce very reliable approximate solutions and therefore significantly reduce the uncertainty in the computed results.

Keywords: (Numerical methods, initial value problems, ODEs, reliable methods, defect control)

1 Introduction

In the numerical solution of ODEs, it is now possible to develop efficient techniques that compute approximate solutions that are more convenient to interpret and understand when used by practitioners who are interested in accurate and reliable simulations of their mathematical models. When implementing numerical methods for ODEs, there is inevitably a trade-off between efficiency and reliability that must be considered and most methods that are widely used are designed to provide reliable results most of the time. The methods we develop in this paper are designed so that the resulting piecewise polynomial will satisfy a perturbed ODE with an associated defect (or residual) that is *reliably* controlled. We also show how the these methods can be the basis for implementing effective tools for visualizing an approximate solution, and for performing key tasks such as sensitivity analysis, global error estimation and parameter fitting. Software implementing this approach will be described for systems of IVPs, BVPs, DDEs, and VIEs.

* This research is supported by the Natural Science and Engineering Research Council of Canada.

Numerical results will be presented which quantify the improvement in reliability that can be expected with the methods we have developed. We will also show an example of qthe use of a related software tool for estimation of the underlying mathematical conditioning of a problem and the global error of the approximate solution.

Consider an IVP defined by the system

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

When approximating the solution of this problem, a numerical method will introduce a partitioning $a = x_0 < x_1 < \cdots < x_N = b$ and determine corresponding discrete approximations $y_0, y_1 \cdots y_N$ where $y_i \approx y(x_i)$. The number of and the distribution of the meshpoints, x_i, are determined adaptively as the method attempts to satisfy an accuracy that is consistent with an accuracy parameter, TOL, that is specified as part of the numerical problem associated with (1).

For many applications it is now recognized that an accurate discrete approximation is not enough and most numerical methods now provide an accurate approximation to the solution of (1) that can be evaluated at any value of $x \in [a, b]$. For a discussion of how this is done and how such methods are used see [10], [3] and [7]. In particular Figures 1 and 2 show the advantage such a method has when it is used to display (or visualize) the solution of an IVP. [Note that the particular problem visualized here will be defined and investigated in more detail in section 3.2.1.] These methods are often called continuous methods (in contrast to the more traditional discrete methods discussed above). [This name can be confusing as the approximate solution provided by a continuous method may not produce an approximate solution $S(x)$ that is in $C^0[a, b]$.] In this investigation we will consider a class of numerical methods which produce a computeable approximation $S(x) \approx y(x)$ for any $x \in [a, b]$ and where the reliability and accuracy of such methods will be quantified in terms of how accurately and reliably $S(x)$ agrees with $y(x)$.

In the next section we will introduce and justify a class of continuous explicit Runge-Kutta methods (SDC-CRKs) that have a rigorously justified error control strategy and are designed to be very reliable when applied to non-stiff IVPs. We will introduce suitable measures that can be used to quantify the reliability of the performance of a CRK method when applied to a particular problem. We will then use these measures to assess the performance of three methods we have implemented (of orders five, six and eight) on a standard collection of 25 non-stiff test problems.

In the third section we will discuss how the approach we have introduced for IVPs has been used to develop reliable CRK-based methods for boundary value problems (BVPs), delay differential equations (DDEs) and Volterra integro-differential equations (VIDES). We also discuss how these CRK method can be used to develop effective software tools to investigate important properties of the problem and/or its approximate solution when the problem belongs to one of these classes. As an example we will show how this approach can be used to develop an effective technique to estimate the mathematical conditioning of

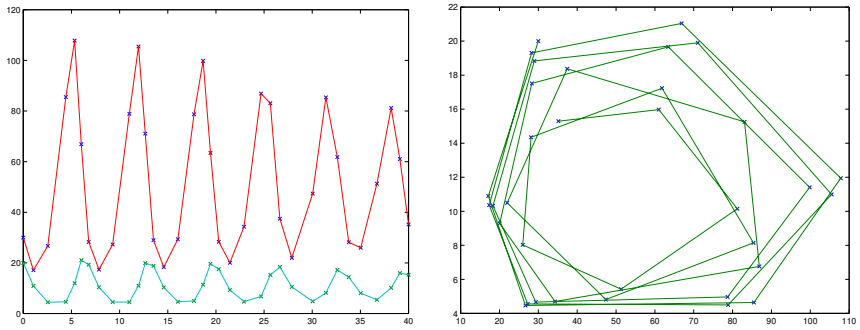

Fig. 1. Visualizing the approximate solution using an accurate discrete approximation. A standard solution plot of each component is displayed on the left while a phase plot of $y_1(t)$ vs $y_2(t)$ is displayed on the right

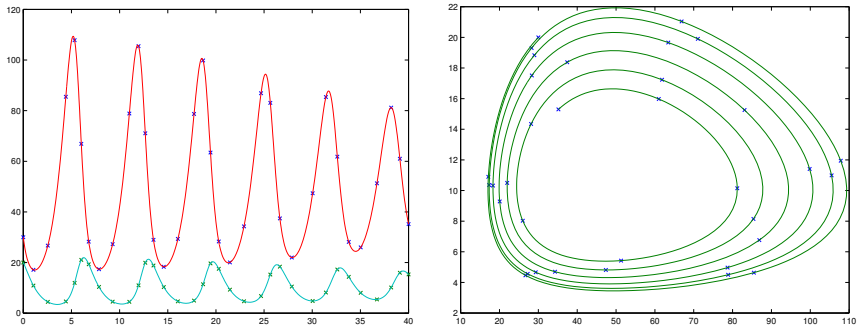

Fig. 2. Visualizing the approximate solution using an accurate continuous approximation. A standard solution plot of each component is displayed on the left while a phase plot of $y_1(t)$ vs $y_2(t)$ is displayed on the right

an IVP as well as an estimate of the global error of an approximate solution. This technique will be illustrated by applying it to two problems.

qIn the final section we will make some general observations and discuss some ongoing and future work that extends the techniques discussed in this paper to other classes of problems.

2 Continuous Runge-Kutta Methods

A classical, explicit, p^{th}-order, s-stage, discrete Runge-Kutta formula is defined by the vectors $(c_1, c_2, \ldots c_s), (w_1, w_2, \ldots w_s)$ and the lower triangular matrix $(a_{i,j}), i = 1, 2 \ldots s, j = 1, 2 \ldots i - 1$. When approximating the solution of (1),

after $y_0, y_1, \ldots y_{i-1}$ have been generated, the formula determines,

$$y_i = y_{i-1} + h_i \sum_{j=1}^{s} \omega_j k_j,$$

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

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

Let $z_i(x)$ be the solution of the local IVP associated with the i^{th} step,

$$z_i' = f(x, z_i(x)), \;\; z_i(x_{i-1}) = y_{i-1}, \;\; \text{for} \;\; x \in [x_{i-1}, x_i].$$

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

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

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

The set of polynomials, $[u_i(x)]_{i=1}^{N}$, define a piecewise polynomial $U(x)$ for $x \in [a, b]$. We consider $U(x)$ to be the numerical solution generated by the CRK method. The particular class of $O(h^p)$ extensions considered here, were introduced in [4].They satisfy,

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

$U(x) \in C^0[a, b]$ and will interpolate the underlying discrete RK values, y_i, if $b_j(1) = \omega_j$ for $j = 1, 2 \cdots s$ and $b_{s+1}(1) = b_{s+2}(1) = \cdots b_{\bar{s}}(1) = 0$. If $k_1 = f(x_{i-1}, y_{i-1})$ and $k_{s+1} = f(x_i, y_i)$, a similar set of constraints on the $\frac{d}{d\,\tau}(b_j(\tau))$ will ensure $U'(x)$ interpolates $f(x_i, y_i), f(x_{i-1}, y_{i-1})$ and therefore $U(x) \in C^1[a, b]$. All the CRK extensions we consider in this investigation are in $C^1[a, b]$.

2.1 Defect Error Control for CRK Methods

When applied to (1) a CRK method will determine an approximate solution, $U(x)$. This approximate solution has a defect (or residual) defined by,

$$\delta(x) = f(x, U(x)) \; - \; U'(x). \tag{2}$$

It can be shown (see [1] for details) that, for such a CRK and $x \in (x_{i-1}, x_i))$,

$$\delta(x) = G(\tau) h_i^p + O(h_i^{p+1}),$$

$$G(\tau) = \tilde{q}_1(\tau)F_1 + \tilde{q}_2(\tau)F_2 + \cdots + \tilde{q}_k(\tau)F_k, \tag{3}$$

where $k \geq 1$ depends on the particular CRK formula and the $\tilde{q}_j's$ are polynomials in τ that depend only on the coefficients defining the CRK formula, while the $F_j's$ are constants (elementary differentials) that depend only on the problem.

CRK Methods can be implemented to adjust h_i in an attempt to ensure that the maximum magnitude of $\delta(x)$ is bounded by TOL on each step (see [11] and [2] for details). The quality of an approximate solution can then be described in terms of the maximum value of $\|\delta(x)\|/TOL$. From (3) it is clear that, as $h_i \to 0$, the defect will look like a linear combination of the $\tilde{q}_j(\tau)$ over $[x_{i-1}, x_i]$. Then the maximum defect will be easier to estimate if $k = 1$, in which case the maximum should occur (as $h_i \to 0$) at $\tau = \tau^*$ where τ^*, is the location in $[0, 1]$ of the local maximum of $\tilde{q}_1(\tau)$. In this case we call the defect control strategy **Strict Defect Control (SDC)** and CRK methods that implement this strategy are called SDC CRK methods. Figure 3 shows how the defect of an SDC method has a consistent shape when applied to a typical non-stiff IVP. We will consider only SDC extensions, $u_i(x)$,

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

In the next section we will discuss how, for a given discrete RK formula, we can identify a suitable continuous extension. SDC methods SDC5, SDC6 and SDC8 have been implemented at a cost per step that is given in table 1. We will report on how well these methods are able to provide reliable and consistent control of the size of the defect on non-stiff problems over a range of prescribed values of TOL.

Table 1. Cost per step of the explicit SDC CRK formulas we have implemented

Formula	p	s	$\bar{s}$
SDC5	5	6	12
SDC6	6	7	15
SDC8	8	13	27

2.2 Optimal SDC Extensions of a Discrete RK Formula

For a particular discrete explicit RK formula, we generally have a family of possible continuous extensions and we are interested in a continuous extension with the lowest cost per step (the smallest value of $\bar{s}$).

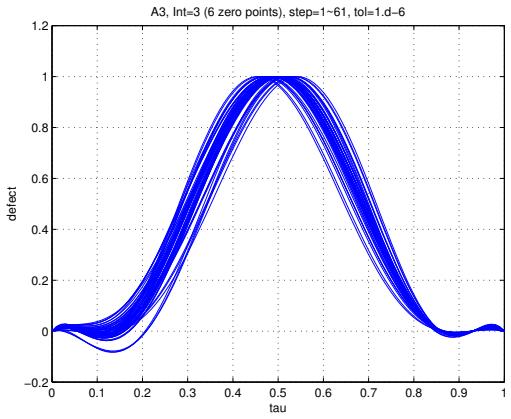

Fig. 3. Plot of scaled defect vs τ (ie. $\delta(\tau)/\delta(\tau^*)$ vs τ) for each step required to solve a typical problem with SDC CRK6 and $TOL = 10^{-6}$. Note that all components of the defect have a similar "shape" on each problem

In selecting an optimal continuous extension, one should also attempt to avoid potential difficulties which can arise. Each SDC extension satisfies,

$$\delta(x) = \tilde{q}_1(\tau)F_1 h_i^p \; + \; (\hat{q}_1(\tau)\hat{F}_1 + \hat{q}_2(\tau)\hat{F}_2 + \cdots\cdots \hat{q}_{\hat{k}}(\tau)\hat{F}_{\hat{k}})h_i^{p+1} + O(h_i^{p+2})$$

and a particular extension might be inappropriate for two reasons,

- $\tilde{q}_1(\tau)$ may have a large maximum (It is straightforward to show that, for the SDC extensions we are considering, $\tilde{q}_1(0) = \tilde{q}_1(1) = 0$ and its 'average' value must be one, for $\tau \in (0, 1)$).
- The $\hat{q}_j(\tau)$ may be large in magnitude relative to $\tilde{q}_1(\tau)$ (and therefore h_i would have to be small before the estimate is justified). (That is, before $|h_i\hat{q}_j(\tau)| << |\tilde{q}_1(\tau)|$.)

For each p we have identified a particular SDC-CRK that minimizes these difficulties and uses the fewest number of additional stages, $\bar{s}$. Note that if $|F_1|$ is zero or very small on isolated steps then the associated error control may still be unreliable. Figure 4 shows plots of the polynomials $\tilde{q}_1(\tau)$ and $\hat{q}_j(\tau), \ldots \hat{q}_{\hat{k}}(\tau)$ for the particular order 6 SDC extension we have chosen to implement.

2.3 Quantifying Reliability of a SDC Method

Consider two measures of reliability of a CRK method:

- How well does the **Method** control the maximum magnitude of the defect? We can measure the ratio of the max defect to TOL on each step (DMAX) and the fraction of steps where this ratio is greater than 1 (Frac-D).

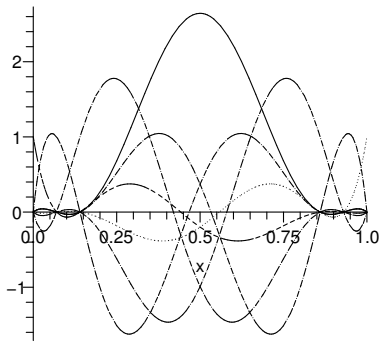

Fig. 4. Plots of $\tilde{q}_1$ and $\hat{q}_2 \cdots \hat{q}_7$ for SDC CRK6. $\tilde{q}_1$ is represented by the solid line and has the highest magnitude.

– How well does the **Estimate** of the max defect reflect its true value? We can measure both the ratio of the true maximum defect (on a successful step) to its estimated value (R-Max) and the fraction of attempted steps where the estimated maximum is within one percent of the true maximum (Frac-G).

We will use these measures of reliability to demonstrate that SDC error control significantly reduces the uncertainty of approximate solutions to ODE problems.

We have implemented SDC RK methods of orders five, six and eight (SDC5, SDC6 and SDC8) and have run each of these methods on the 25 IVP test problems of DETEST [6] (all non-stiff), at 9 tolerances from 10^{-1} to 10^{-9}. The performance of the methods on the 25 test problems on a subset of the tolerances is summarized in Table 2, where we report the above reliability measures, the total number of steps (NSTP) and the total number of function evaluations (NFCN) for all the problems.

3 SDC RK Methods for other Classes of ODEs

In addition to reliable methods for IVPs, we have developed (or are actively developing) effective and very reliable SDC methods for other important classes of differential equations. These include,

– BVPs ([5]):

$$y' = f(x, y), \quad x \in [a, b],$$

with

$$g(y(a), y(b)) = 0, \quad g : \Re^n \times \Re^n \to \Re^n.$$

Table 2. Numerical Results for SDC CRKs on the 25 problems of DETEST

TOL	CRK	NSTP	NFCN	DMAX	Frac-D	R-Max	Frac-G
10^{-2}	SDC5	625	11709	0.97	.000	1.05	.67
	SDC6	549	12300	1.00	.000	1.43	.71
	SDC8	333	12793	1.01	.003	1.65	.35
10^{-4}	SDC5	1065	19033	1.01	.001	1.12	.78
	SDC6	931	19819	1.00	.001	1.08	.87
	SDC8	465	17319	1.05	.004	1.47	.45
10^{-6}	SDC5	2099	35703	1.01	.002	1.08	.86
	SDC6	1748	35073	1.01	.001	1.08	.96
	SDC8	712	26253	1.02	.001	1.34	.59
10^{-8}	SDC5	4566	66937	1.01	.001	1.07	.95
	SDC6	3547	65148	1.01	.001	1.07	.98
	SDC8	1081	38251	1.12	.007	2.60	.62

– DDEs (both retarded and neutral problems) ([12]):

$$y' = f(x, y(x), y(x-\sigma_1) \cdots y(x-\sigma_k), y'(x-\sigma_{k+1}), \\ \cdots y'(x-\sigma_{k+\ell})), \quad \text{for } x \in [a,b],$$

where $y(x) \in \Re^n$ and,

$$y(x) = \phi(x), \quad y'(x) = \phi'(x), \quad \text{for } \; x \le a,$$

$$\sigma_i \equiv \sigma_i(x, y(x)) \ge 0 \;\text{ for }\; i = 1, 2 \cdots k+\ell.$$

– VIDEs (with a time dependent delay) ([9]):

$$y'(x) = f(x, y(x)) + \int_{x-\sigma(x)}^{x} K(x, s, y(s), y'(s)) ds, \tag{4}$$

for $x \in [a,b]$, $f : \Re \times \Re^n \to \Re^n$ and $K : \Re \times \Re \times \Re^n \times \Re^n \to \Re^n$ and $y(x) = \phi(x)$ for $x \le a$.

For each Class of ODEs we are not only interested in providing effective SDC methods to approximate the solution of the ODE, but we are also developing effective software tools to investigating important properties of the problem and its approximate solution. For example:

– Detecting, Locating and Coping with Discontinuous Problems
– Estimating the Global Error and the Mathematical Conditioning of the Problem
– Computing a sensitivity analysis of the solution (eg., $\frac{\partial y_i(x)}{\partial p_j}$).
– Solving Problems which depend on parameters and parameter determination.

3.1 Global Error Estimates and Condition Number of an IVP

Assume $y(x)$ satisfies (1) and the computed approximate solution $U(x)$ satisfies (from (2)) the perturbed IVP,

$$U' = f(x,U) - \delta(x), \;\; U(x_0) = y_0 \;\text{ on }\; [a,b], \;\; \text{with } \; \|\delta(x)\| \leq TOL.$$

Let $\epsilon(x) = y(x) - U(x)$. From the variation of constants formula, (see for example [8]), one can show,

$$\|\epsilon(x)\| \leq K(x)\, TOL,$$

where $K(x)$ reflects the sensitivity of $y(x)$ to perturbations. Then

$$\bar{K} \equiv \max_{x\in[a,b]} K(x),$$

can be viewed as the condition number of this IVP. From the definition of $\bar{K}$ we can determine a lower bound, $\hat{K}$,

$$\hat{K} \equiv \max_{x\in[a,b]} \|\epsilon(x)\|/TOL.$$

If we compute an accurate approximation $E(x)$, to $\epsilon(x)$, (and the inequality $\|\delta(x)\| \leq TOL$ is almost sharp), then an effective estimate of the conditioning of the IVP is,

$$\tilde{K} \equiv \max_{x\in[a,b]} \|E(x)\|/TOL. \tag{5}$$

We know that, $\epsilon(x) = y(x) - U(x)$, is the exact solution of the IVP,

$$\begin{aligned}
\epsilon' &= f(x,y) - f(x,U) - \delta(x), \\
&= f(x, U(x) + \epsilon(x)) - f(x,U) - \delta(x), \\
&= f(x, U(x) + \epsilon(x)) - U'(x), \\
&\equiv g(x,\epsilon).
\end{aligned}$$

Therefore if we solve this 'companion' IVP using the same SDC method used to determine $U(x)$, we can determine an inexpensive estimate $E(x)$ of the global error and use this to obtain (from (5)) an estimate of the conditioning of the IVP. Note that this computed $E(x)$ will satisfy the IVP,

$$E' = g(x,E) + \delta_2(x), \;\; \text{where } \; \|\delta_2(x)\| \leq TOL_2.$$

We can also use this estimate of the global error to improve the accuracy of the numerical solution since $U_1(x) = U(x) + E(x)$ satisfies the perturbed IVP:

$$\begin{aligned}
U_1'(x) &= U'(x) + E'(x), \\
&= f(x,U) + \delta(x) + g(x,E) + \delta_2(x), \\
&= f(x,U) + \delta(x) + f(x, U(x) + E(x)) - U'(x) + \delta_2(x), \\
&= f(x, U(x) + E(x)) + \delta_2(x), \\
&= f(x, U_1(x)) + \delta_2(x),
\end{aligned}$$

where $\|\delta_2(x)\| \leq TOL_2$ and TOL_2 can be determined by sampling $\|\delta_2(\tau^*)\|$ on each step.

3.2 Two Sample Problems

Predator – Prey Problem:.

This is a well known system that models (over time) the populations of two computing species in an isolated environment. It is a well conditioned problem.

$$y_1' = y_1 - 0.1y_1y_2 + 0.02x,$$
$$y_2' = -y_2 + 0.02y_1y_2 + 0.008x,$$

$$\text{with } y_1(0) = 30, \quad y_2(0) = 20, \quad \text{and } x \in [0, 4].$$

Lorenz Problem:

This is a standard example often cited in the literature on dynamical systems as a system which can exhibit chaotic behaviour. The condition number is exponential in the length of the integration interval.

$$y_1' = 10(y_2 - y_1),$$
$$y_2' = y_1(28 - y_3) - y_2,$$
$$y_3' = y_1y_2 - \frac{8}{3}y_3,$$

$$\text{with } y_1(0) = 15, \ y_2(0) = 15, \ y_3(0) = 36, \quad \text{and } x \in [0, 15].$$

For each method we monitor performance on these problems over a range of tolerances and report, in Table 3 and Table 4, the following:

- NS – The number of steps to determine $U(x)$.
- NSE – The number of steps to determine $E(x)$.
- DEFUM – The maximum magnitude of the defect $\delta(x)$, (associated with $U(x)$), in units of TOL. This is determined by evaluating the defect at several sample points per step.
- G-ERRM – The maximum global error associated with $U(x)$ in units of TOL. This is determined by computing the true global error at 100 sample points per step.
- K-ESTM – The estimate of the conditioning corresponding to the maximum observed value of $\|E(x)\|/TOL$ measured over 100 sample values per step.
- DEFEM – The maximum magnitude of the defect $\delta_2(x)$, (associated with $E(x)$), in units of TOL.
- GE(U+E) – The The maximum global error associated with the improved solution $U(x) + E(x)$ in units of TOL.

4 Observations and Future Work

The results presented in Table 2 demonstrate the strong reliability of the SDC IVP methods we have implemented. In particular these tables show, that over a wide range of non-stiff problems and accuracy requests, the computed approximate solution will almost always satisfy a perturbed ODE with the norm of the

Table 3. Reliability of Error Control and Validity of the Estimate of Conditioning for SDC on the pred-prey problem

Method	TOL :	10^{-2}	10^{-4}	10^{-6}	10^{-8}
SDC5:	NS	70	148	315	705
	NSE	147	307	644	1412
	DEFUM	1.8	1.1	1.2	1.2
	G-ERRM	3.7	7.3	11.4	14.4
	K-ESTM	3.7	7.3	11.4	14.6
	DEFEM	.009	.009	.011	.034
	GE(U+E)	.002	.009	.004	.041
SDC6:	NS	65	134	277	585
	NSE	132	265	551	1168
	DEFUM	1.3	1.0	1.0	1.2
	G-ERRM	2.2	4.6	2.5	3.5
	K-ESTM	2.2	4.6	2.5	3.6
	DEFEM	.009	.005	.007	.013
	GE(U+E)	.0006	.001	.001	.008
SDC8:	NS	34	53	83	127
	NSE	65	104	177	262
	DEFUM	1.3	1.1	0.9	2.1
	G-ERRM	9.5	6.1	6.1	14.4
	K-ESTM	9.5	6.1	6.1	13.9
	DEFEM	.012	.010	.018	1.9
	GE(U+E)	.0009	.002	.003	2.0

Table 4. Reliability of Error Control and Validity of the Estimate of Conditioning for SDC on the Lorenz problem

Method	TOL :	10^{-2}	10^{-4}	10^{-6}	10^{-8}
SDC5:	NS	356	751	1738	4304
	NSE	834	1591	3470	4306
	DEFUM	1.3	1.4	1.4	1.4
	G-ERRM	$4.4\cdot10^{3}$	$1.9\cdot10^{5}$	$1.9\cdot10^{5}$	$1.8\cdot10^{6}$
	K-ESTM	$4.6\cdot10^{3}$	$1.9\cdot10^{5}$	$1.9\cdot10^{5}$	$1.9\cdot10^{6}$
	DEFEM	.016	.018	.020	.14
	GE(U+E)	$.47\cdot10^{3}$	$.50\cdot10^{2}$	$.17\cdot10^{3}$	$.68\cdot10^{4}$
SDC6:	NS	316	642	1339	2865
	NSE	731	1326	2678	2865
	DEFUM	1.4	1.3	1.3	1.2
	G-ERRM	$4.2\cdot10^{3}$	$2.8\cdot10^{5}$	$1.5\cdot10^{5}$	$1.5\cdot10^{5}$
	K-ESTM	$4.2\cdot10^{3}$	$2.8\cdot10^{5}$	$1.5\cdot10^{5}$	$1.3\cdot10^{5}$
	DEFEM	.011	.011	.004	.20
	GE(U+E)	$.29\cdot10^{3}$	$.31\cdot10^{3}$	$.32\cdot10^{3}$	$.20\cdot10^{5}$
SDC8:	NS	145	228	371	634
	NSE	292	454	803	1349
	DEFUM	1.3	1.4	1.6	1.4
	G-ERRM	$5.5\cdot10^{3}$	$.16\cdot10^{5}$	$.14\cdot10^{5}$	$.20\cdot10^{5}$
	K-ESTM	$5.5\cdot10^{3}$	$.16\cdot10^{5}$	$.15\cdot10^{5}$	$.70\cdot10^{5}$
	DEFEM	.013	.003	.076	9.0
	GE(U+E)	$.70\cdot10^{2}$	$.82\cdot10^{1}$	$.42\cdot10^{3}$	$.48\cdot10^{5}$

perturbation bounded by the requested accuracy parameter, TOL. Furthermore, our analysis in the previous section shows that, as a result of this strong reliability property, the maximum global error will be proportional to the tolerance and the proportionality constant will be insensitive to the order of the SDC method. The results reported in Table 3 and Table 4 confirm that this is true for our two test problems. This allows us to implement and justify a rigorous and inexpensive measure of the underlying mathematical conditioning. For example, in the case of the Lorenz problem, which is known to be badly conditioned, Table 4 shows, that in order to compute an approximate solution with an accuracy of two significant figures, one must specify a value for TOL that is less than 10^{-7}.

It must be acknowledged that the analysis and methods developed in this paper apply to the usual case where truncation error of the RK formulas dominates the affects of rounding errors when approximating the solution of an ODE. If one is interested in satisfying severe accuracy requirements and using a high order SDC method then round-off error can become significant and reduce the reliability of the computed results. In such cases, an SDC method can (at a small amount of extra work) detect that the defect estimates are adversly affected by round-off error (see [4] for details) and signal that this is the case. The remedy, in this case, would be to use higher precision (if it is available) or to use a lower order SDC method which is not as sensitive to round-off errors.

The SDC methods investigated in this paper are suitable for non-stiff problems. We are currently implementing and testing continuous extensions of implicit RK methods that could be suitable for stiff problems. The derivation of these extensions is straightforward, but the development of an effective adaptive stepsize control strategy for stiff problems remains a challenge. We are considering some alternative techniques related to defect control for use on these problems. We are also considering how to best develop accurate continuous extensions and reliable defect control for multistep methods.

References

1. Enright, W.: A new error-control for initial value solvers. App. Math. Comp. 31, 288–301 (1989)
2. Enright, W.: The relative efficiency of alternative defect control schemes for high-order continuous Runge-Kutta formulas. SIAM Journal on Numerical Analysis 30(5), 1419–1445 (1993)
3. Enright, W., Jackson, W., Nϕrsett, S., Thomsen, P.: Interpolants for Runge-Kutta formulas. ACM Transactions on Mathematical Software 12(3), 193–218 (1986)
4. Enright, W., Yan, L.: The Reliability/Cost trade-off for a class of ODE solvers. Numerical Algorithms 53(2), 239-260 (2009)
5. Enright, W., Muir, P.: New Interpolants for Asymptotically Correct Defect Control of BVODEs. Numerical Algorithms (53)2, 219-238 (2009)
6. Enright, W., Pryce, J.: Two FORTRAN packages for assessing initial value methods. ACM Transactions on Mathematical Software 13(1), 1–27 (1987)
7. Gladwell, I., Shampine, L.,Baca, L., Brankin, R.: Practical aspects of interpolation in Runge-Kutta codes. SIAM Journal of Scientific and Statistical Computing (8), 322-341 (1987)

8. Hairer, E., Nϕrsett, S., Wanner, G.: Solving Ordinary Differential Equations I: Nonstiff Problems. Springer-Verlag, Berlin (1987)
9. Shakourifar, M. Enright, W.: Reliable Approximate Solution of Systems of Volterra Integro-Differential Equations with Time Dependent Delays. to appear in SIAM Journal of Scientific Computing, 2011.
10. Shampine, L.: Interpolation for Runge-Kutta methods. SIAM Journal of Numererical Analysis (22), 1014-1027 (1985)
11. Shampine, L.: Solving ODEs and DDEs with residual control. Applied Numererical Mathematics (52), 113-127 (2005)
12. Zivaripiran, H., Enright, W.: An Efficient Unified Approach for the Numerical Solution of Delay Differential Equations. Numerical Algorithms (53)2, 397-417 (2009)

DISCUSSION

Speaker: Wayne Enright

Bill Oberkampf : Is the advantage of continuous Runge-Kutta methods over traditional Runge-Kutta methods that you can relax the assumption on the solution from C^1 to C^0?

Wayne Enright : No. If the solution is not not differentiable at $\bar{x} \in [a, b]$ then, for any numerical method to be effective, it must locate all such points and force these points to be meshpoints. This can be done automatically by CRK method which detect such points by observing sudden increases in the magnitude of the defect. The main advantage of CRK methods is that they provide accurate approximations to the solution for any value of $x \in [a, b]$, (not just at the meshpoints, x_i).

The term continuous Runge Kutta method can be misleading. It would perhaps be better to refer to this class of Runge Kutta methods as continuous-output Runge Kutta methods (CORK), or dense-output Runge Kutta methods (DORK).

Van Snyder : Can the ideas underlying continuous Runge-Kutta methods be applied to Adams method?

Wayne Enright : This is an extension that we have thought about for some time. The main difficulty in extending the approach is that, for the most natural piecewise polynomial approximations, the associated defect would depend on past stepsizes as well as on the current stepsizes. This would make it particularly challenging to define local interpolants that permit an asymptotically correct estimate of the maximum defect.