Robust and Reliable Error Control for ODE Software

W.H. Enright
Department of Computer Science
University of Toronto
Toronto, Canada M5S 3G4

enright@cs.utoronto.ca

ANODE 2001

Auckland, January 2001

$\boxed{\text{Background/motivation:}}$

We will consider continuous methods which are designed to directly monitor and control the maximum magnitude of an estimate of the defect of a piecewise polynomial $u(t)$ that is delivered as the approximate solution to an ODE.

We will focus on methods based on an order p CRK formula but most of the discussion and analysis will apply to other classes of continuous methods such as those based on multistep formulas or collocation.

$\boxed{\text{Outline of Presentation:}}$

1. Continuous Runge-Kutta methods (CRK).

2. The defect (residual) of a numerical solution and the associated error-control and mesh selection strategies.

3. Cost/reliability/robustness issues and trade-offs.

4. Survey of existing software: for IVPs, for BVPs and for DDEs.

5. What needs to be done ?

$\boxed{\text{I) Continuous Runge-Kutta formulas (CRKs):}}$

For $t \in [a, b]$ consider the 1^{st}-order system

$$y' = f(t, y).$$

A p^{th}-order, s-stage RK formula determines

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

where

$$k_j = f(t_{i-1} + hc_j, y_{i-1} + h \sum_{r=1}^{s} a_{jr} k_r).$$

A Continuous extension (CRK) is determined by adding extra stages to obtain an order p approximation for $t \in (t_{i-1}, t_i)$

$$u_i(t) = y_{i-1} + h \sum_{j=1}^{\bar{s}} b_j(\frac{t - t_{i-1}}{h}) k_j,$$

where $b_j(\tau)$ is a polynomial of degree p.

Use of $u_i(t)$ for $t \in (t_{i-1}, t_i)$:

- As a 'post-processing' option for plotting or event location ('gstops').

- in the strategies/heuristics of the method.

  - defect control

  - discontinuity detection

  - mesh refinement

  - software for IVPs, BVPs, DDEs, DAEs

- This approach has been extended to collocation formulas for mixed-order systems.

  - leads to 'superconvergent interpolants'

- The $[u_i(t)]_{i=1}^{N}$ define a piecewise polynomial $u(t)$ for $t \in [a,b]$. This will be considered the numerical solution generated by the CRK method.

- $u(t) \in C^0[a,b]$ and will interpolate the underlying discrete RK values, y_i, if $b_j(1) = \omega_j$ for $j = 1, 2 \cdots s$ and $b_{s+1}(1) = b_{s+2}(1) = \cdots b_{\bar{s}}(1) = 0$.

- Similarly a simple set of constraints on the $b_j^{'}(\tau)$, will ensure $u^{'}(t)$ interpolates $f(t_i, y_i)$ and therefore $u(t) \in C^1[a,b]$.

Formula	p	s	$\bar{s}$
CRK4	4	4	6
CRK5	5	6	9
CVSS6B	6	7	11
CVSS7	7	9	15
CVSS8	8	13	21

Table 1: Cost per step of some CRK formulas

$\boxed{\text{II) The defect of the numerical solution:}}$

For a piecewise polynomial approximation $u(t)$ we define the defect, $\delta(t)$, for $t \in (t_0, t_F)$ by

$$\delta(t) = u'(t) - f(t, u(t)).$$

Note that, in deriving CRK formulas, our assumption that $u(t)$ be order p implies, for sufficiently smooth problems,

$$u'(t) = y'(t) + O(h^{p-1}).$$

Furthermore, since

$$\begin{aligned}\delta(t) &= u'(t) - f(t, u(t)) - y'(t) + f(t, y(t)) \\ &= (u'(t) - y'(t)) + (f(t, y(t)) - f(t, u(t))),\end{aligned}$$

for differential equations that are Lipschitz continuous, $\|\delta(t)\|$ will be at worst $O(h^{p-1})$.

If we let $z_i(t)$ be the solution of the local initial value problem

$$z_i' = f(t, z_i), \quad z_i(t_{i-1}) = y_{i-1},$$

for $i = 1, 2 \cdots N$; then the local error, $le_i(t)$, associated with the continuous approximation on step i is defined for $t \in (t_{i-1}, t_i)$ to be

$$le_i(t) = z_i(t) - u_i(t).$$

it is well known that the discrete local error of a p^{th}-order Runge-Kutta formula must be $O(h_i^{p+1})$. That is

$$\begin{aligned} le_i(t_i) &= z_i(t_i) - u_i(t_i) \\ &= z_i(t_i) - y_i \\ &= O(h_i^{p+1}). \end{aligned}$$

If the continuous local error is $O(h_i^{p+1})$ then we will have

$$u_i'(t) = z_i'(t) + O(h_i^p)$$

and therefore, for $t \in (t_{i-1}, t_i)$,

$$\begin{aligned} \delta(t) &= (u_i'(t) - z_i'(t)) + (f(t, z_i(t)) - f(t, u_i(t))) \\ &= O(h_i^p). \end{aligned} \tag{1}$$

Furthermore, if we have the stronger additional constraint that, for $t \in (t_{i-1}, t_i)$,

$$le_i(t) = \psi_i(\tau)\ D(t_{i-1})\ h_i^{p+1} + O(h_i^{p+2}), \tag{2}$$

where $D(t)$ depends only on the problem, and $\psi_i(\tau)$ is a polynomial in τ whose coefficients are independent of the problem, it can be shown that

$$\delta(t) = \psi_i'(\tau)\ D(t_{i-1})\ h_i^p + O(h_i^{p+1}).$$

Note that an order p CRK method will always deliver $O(h^p)$ accuracy but, without additional constraints (such as (1) or (2)), these formulas may be more difficult to implement in an effective and reliable way (when the objective is to control the magnitude of $\|\delta(t)\|$) since:

- The magnitude of the associated defect will in general only be $O(h_i^{p-1})$,

- Although the defect can be sampled at any point in the interval of interest it may not be easy to justify a rigorous, inexpensive estimate of its maximum magnitude over each step.

The first of these difficulties can be overcome by considering only order p CRK formulas with local error that satisfies (1) and both difficulties can be overcome by considering only those order p CRK formulas that satisfy (2).

In this latter case as $h_i \to 0$, for any $t \in (t_{i-1}, t_i)$, $\delta(t)$ will satisfy (2) and therefore for $\bar{t} \in (t_{i-1}, t_i)$ (corresponding to a local maximum of $|\psi_i'(\frac{t-t_{i-1}}{h_i})|$), $\|\delta(\bar{t})\|$ will be an asymptotically correct estimate of the maximum magnitude of the defect on the i^{th} step.

Note that since $\psi_i'(\tau)$ is a polynomial which is independent of the problem and the stepsize, the location of its local maximum magnitude, $\bar{\tau}$, (for $\tau \in (0, 1)$) is known and the corresponding value for $\bar{t}$ is $\bar{t} = t_{i-1} + \bar{\tau} h_i$.

The following table identifies the number of stages required for some order p CRK methods we have developed. $\bar{s}$ is the number of stages necessary to produce a CRK satisfying (1), while $\tilde{s}$ is the number of stages required to produce a CRK satisfying (2). The underlying discrete formulas were selected from a family of formulas proposed by Verner and investigated by Sharp and Smart.

Formula	p	s	$\bar{s}$	$\tilde{s}$
CRK4	4	4	6	7
CRK5	5	6	9	11
CVSS6B	6	7	11	14
CVSS7	7	9	15	20
CVSS8	8	13	21	28

Table 2: Cost per step of some CRK formulas

$\boxed{\text{IV) Cost/reliability/robustness trade-offs:}}$

Let $\| \ \|$ be a vector norm that has been prescribed (as the default or part of the problem specification) for measuring the size of the error in the numerical approximation. Two norms that have been suggested for monitoring/controlling the size of the defect on each step are:

i) $\|\delta_i(t)\|_\infty = \max_{t_{i-1}<t<t_i}[\|\delta(t)\|]$,

ii) $\|\delta_i(t)\|_2 = 1/h_i \left(\int_{t_{i-1}}^{t_i} \|\delta_i(s)\|^2 \, ds\right)^{1/2}$.

In the software we have developed we have used i) but others (eg. Shampine and Kierzenka) have used ii) and argued that it can more credibly estimated for coarse meshes.

Whichever norm is used, one can use two types of estimates. A 'strict' defect control where the underlying CRK satisfies a condition similar to (2) and a 'relaxed' defect control where only (1) is satisfied and the corresponding estimate is based on a small number of sampled defect evaluations. We have implemented both types of error control and also implemented methods which leave this decision as an option.

There is clearly a cost/reliability trade off to be considered when selecting the 'right' combination for a particular application.

To illustrate this we assessed the performance of two versions of the CRK method CVSS6B using DETEST. Table 3 shows results for problem B4 which is typical for all problems.

Typical performance of strict vs relaxed error control. Note that with the different interpolants, but identical underlying discrete formula, the actual stepsizes used may be quite different.

Strategy	TOL	Time	FCN	Steps	GL Err	Max Def	% Succ
Relaxed defect control	10^{-2}	.010	166	15	4.7	1.9	73
	10^{-3}	.015	254	22	18.0	1.2	91
	10^{-4}	.023	375	32	28.4	1.3	84
	10^{-5}	.031	507	46	34.5	1.7	74
	10^{-6}	.046	749	68	37.6	1.5	62
	10^{-7}	.067	1101	100	39.0	1.5	49
	10^{-8}	.099	1618	147	40.5	1.5	45
	10^{-9}	.145	2377	216	41.1	1.5	45
Strict defect control	10^{-2}	.014	253	17	1.2	.88	100
	10^{-3}	.021	379	24	9.5	.94	100
	10^{-4}	.027	491	35	15.6	.98	100
	10^{-5}	.041	743	53	16.6	.76	100
	10^{-6}	.061	1093	78	16.6	.66	100
	10^{-7}	.090	1625	116	16.5	.61	100
	10^{-8}	.131	2381	170	16.5	.58	100
	10^{-9}	.193	3501	250	16.5	.56	100

Table 3: CVSS6B on problem B4 of DETEST

Advantages of CRK based IVP methods:

- event detection (gstops)
- discontinuity detection and accurate crossing
- solutions generated by different methods can be validated and compared generically

$\boxed{\text{Software for BVPs:}}$

Consider the two-point BVP:

$$y' = f(t, y), t \in [t_0, t_F], g(y(t_0), y(t_F)) = 0. \quad (3)$$

MIRKDC and bvp4c directly attempt to produce an approximate solution that satisfies:

$$u'(t) = f(t, u(t)) + \delta(t),$$

with

$$\|\delta(t)\| \leq \ TOL,$$

and

$$g((u(t_0), u(t_F)) = 0.$$

Advantages of CRK based BVP methods:

- cost of determining $u(t)$ is small relative to that of determining the discrete solution,

- size of the defect can be used in the mesh refinement strategy (even before convergence achieved on an initial coarse mesh),

- $u(t)$ can be used to generate an initial guess for use with continuation.

$\boxed{\text{Software for DDEs:}}$

A class of numerical methods based on CRK formulas with defect control has been developed for systems of retarded and neutral DDEs of the form,

$$
\begin{aligned}
y' &= f(t, y, y(t - \sigma_1(t, y)), y'(t - \sigma_2(t, y))), \\
y(t) &= \phi(t), \;\; t \le t_0,
\end{aligned}
$$

where $\sigma_1(t, y)$ and $\sigma_2(t, y)$ are positive scalar functions. One particular formula from this class, CVSS6B, has been implemented in DDVERK (see also dde23 of Shampine and Thompson).

Advantages of CRK based DDE methods:

- automatic handling of discontinuities
- can easily handle multiple delays
- can easily cope with vanishing delays

V) What needs to be done next?

- DAEs (experimental code to software)
- BVPs (improvements to key strategies)
- DDEs (recognizing difficulties)
- IVPs (consistent interface for all widely used software)