

Effective Condition Number Estimates for a Class of Initial Value Solvers ^a

W.H. Enright

Department of Computer Science

University of Toronto

SCICADE 2007, ST. Malo

^a This research was supported by the Natural Science and Engineering Research Council of Canada.

Acknowledgement

This work is part of an ongoing project and has benefited from numerous discussions and collaborations with

■ Colleagues:

- Paul Muir
- Wayne Hayes
- John Pryce
- Ned Nedialkov

■ Students:

- Hossein Zivari Piran
- Li Yan

Outline of Talk

- Condition Number of an IVP
 - Global Error Estimate
 - Iterative Improvement
- Continuous Runge-Kutta Methods (CRKs)
 - Relaxed Defect Control (RDC)
 - Strict Defect Control (SDC)
- Implementation and Numerical Results
- Conclusions and Future/Ongoing Related Work

Condition Number of an IVP

Assume $y(x)$ satisfies the IVP,

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

and $U(x)$ satisfies the perturbed IVP,

$$U' = f(x, U) + \delta(x), \quad U(x_0) = y_0 \quad \text{on } [x_0, x_F], \quad \text{with } \|\delta(x)\| \leq TOL.$$

Let $\epsilon(x) = y(x) - U(x)$. From the variation of constants formula,

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

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

$$\bar{K} \equiv \max_{x \in [x_0, x_F]} K(x),$$

can be viewed as the condition number of this IVP.

An Approximation to \bar{K}

- From the definition of \bar{K} we can determine a lower bound, \hat{K} ,

$$\hat{K} \equiv \max_{x \in [x_0, x_F]} \|\epsilon(x)\| / TOL.$$

- If one can compute an accurate and inexpensive approximation $E(x)$ to $\epsilon(x)$, and $U(x)$ is a numerical solution satisfying the perturbed IVP, then an effective estimate of the conditioning of the IVP is available as,

$$\tilde{K} \equiv \max_{x \in [x_0, x_F]} \|E(x)\| / TOL.$$

Computing $E(x) \approx \epsilon(x)$

■ With $\epsilon(x) = y(x) - U(x)$, we know it satisfies the IVP,

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

■ If we solve this 'companion' IVP using the same method used to determine $U(x)$, we can determine an inexpensive estimate $E(x)$ satisfying the IVP,

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

Iterative Improvement

- We can also use this estimate of the global error to 'improve' our 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 $\|\delta_2(x)\| \leq TOL_2$.

Continuous Runge-Kutta Methods

- Consider an IVP defined by the system

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

A numerical method will introduce a partitioning $x_0 < x_1 < \cdots < x_N = x_F$ and corresponding discrete approximations $y_0, y_1 \cdots y_N$.

- On step $(i + 1)$ let $z_i(x)$ be the solution of the local IVP:

$$z'_i = f(x, z_i(x)), \quad z_i(x_i) = y_i, \quad \text{on } [x_i, x_{i+1}].$$

CRK methods

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

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

where the j^{th} stage is defined by,

$$k_j = f\left(x_i + h_{i+1}c_j, y_i + h_{i+1} \sum_{r=1}^s a_{jr}k_r\right).$$

A Continuous extension (CRK) is determined by adding $(\bar{s} - s)$ extra stages to obtain an order p approximation for $x \in (x_i, x_{i+1})$

$$u_i(x) = y_i + h_{i+1} \sum_{j=1}^{\bar{s}} b_j \left(\frac{x - x_i}{h_{i+1}} \right) k_j,$$

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

CRK Methods (cont)

$$u_i(x) = y_i + h_{i+1} \sum_{j=1}^{\bar{s}} b_j(\tau) k_j,$$

- The $[u_i(x)]_{i=1}^N$ define a piecewise polynomial $U(x)$ for $x \in [x_0, x_F]$. $U(x)$ is the numerical solution generated by the CRK method.
- We will consider two types of $O(h^p)$ extensions, satisfying:

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

Defect Error Control

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

$$\delta(x) = f(x, U(x)) - U'(x) = f(x, u_i(x)) - u_i'(x), \quad \text{for } x \in [x_i, x_{i+1}].$$

It can be shown that, for such a CRK,

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

$$G(\tau) = q_1(\tau)F_1 + q_2(\tau)F_2 + \cdots + q_k(\tau)F_k,$$

where the q_j s are polynomials in τ that depend only on the method while the F_j s are constants that depend only on the problem.

Methods can be implemented to adjust h_{i+1} in an attempt to ensure that the maximum magnitude of $\delta(x)$ is bounded by TOL on each step.

Defect Error Control (cont)

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

$$G(\tau) = q_1(\tau)F_1 + q_2(\tau)F_2 + \cdots + q_k(\tau)F_k.$$

- As $h \rightarrow 0$ the defect will then look like a linear combination of the $q_j(\tau)$ over $[x_i, x_{i+1}]$.
- In the special case where $k = 1$ the shape of the defect will be the same (as $h \rightarrow 0$) for all problems and all steps. That is, the defect will almost always 'converge' to a multiple of $q_1(\tau)$.

Defect Error Control (cont)

- When $k > 1$ one can estimate the maximum defect by evaluating $\delta(x)$ at a carefully chosen set of sample points. We will call this defect control strategy, **Relaxed Defect Control (RDC)**.
- The maximum defect will be easier to estimate if $k = 1$, in which case the maximum should occur (as $h \rightarrow 0$) at $\tau = \tau^*$ where τ^* , the location in $[0, 1]$ of the local maximum of $q_1(\tau)$. In this case we will refer to the defect control strategy as **Strict Defect Control (SDC)**.
- We will consider 2 types of continuous extensions: $u_i(x)$ corresponding to RDC and $\tilde{u}_i(x)$ corresponding to SDC.

Defect Control (cont)

$$RDC : u_i(x) = y_i + h_{i+1} \sum_{j=1}^{\bar{s}} b_j(\tau) k_j = z_i(x) + O(h^{p+1}),$$

$$SDC : \tilde{u}_i(x) = y_i + h_{i+1} \sum_{j=1}^{\tilde{s}} \tilde{b}_j(\tau) k_j = z_i(x) + O(h^{p+1}).$$

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 CRK formulas

Implementation of this Approach

- CRK5, CRK6 and CRK8 implemented with RDC or SDC selected as an option.
- The computation of $E(x)$ is done with the same method used to generate $U(x)$ with $TOL_2 = TOL/100$ (in most cases).
- We will report on how well the methods are able to provide consistent accuracy and a consistent indication of the conditioning for two test problems over a range of accuracies.

Parallel implementation of SDC

For CRKs the cost of computing y_{i+1} is almost the same as forming the interpolant $\tilde{u}_i(x)$. If there are 2 or 3 processors available, the following organization of tasks should lead to a very effective way to implement this approach and compute $U(x)$, $E(x)$, $U_1(x)$, and $\max[||\delta_2(x)||]$ in total time that is comparable to computing $U(x)$ on a single processor.

For each step from x_i to x_{i+1} do the following:

Processor 1: Compute y_{i+1} and $u_i(x)$ for $x \in [x_i, x_{i+1}]$.

Processor 2: Apply the discrete formula only, taking 2 steps of size $h_{i+1}/2$ to compute $E_{i+1/2}, E_{i+1}$.

Processor 3: Compute the interpolant $E(x)$ for $x \in [x_{i-1/2}, x_{i+1/2}]$ using the discrete solution produced by Processor 2. (This processor will be a half step behind the other processors.)

Test Problems

Predator – Prey Problem:

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

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

with $y_1(0) = 30$, $y_2(0) = 20$, and $x \in [0, 40]$.

Lorenz Problem:

$$y_1' = 10(y_2 - y_1),$$

$$y_2' = y_1(28 - y_3) - y_2,$$

$$y_3' = y_1y_2 - \frac{8}{3}y_3,$$

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

Performance of the Methods

For each method we monitor performance over a range of tolerances and report 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 several sample points per step.

Performance of the Approach

For each problem and method we report the following:

- K-ESTM – The estimate of the conditioning corresponding to the maximum observed value of $\|E(x)\|/TOL$ measured over several 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 .

RDC on pred-prey problem

Method	TOL :	10^{-2}	10^{-4}	10^{-6}	10^{-8}
CRK50:	NS	73	155	345	825
	NSE	174	422	1018	2546
	DEFUM	2.1	2.7	8.1	5.6
	G-ERRM	4.0	6.5	11.3	9.7
CRK60:	NS	69	140	291	614
	NSE	133	280	582	1196
	DEFUM	1.4	1.0	4.5	3.2
	G-ERRM	2.9	3.1	2.6	2.3
CRK80:	NS	38	59	89	141
	NSE	70	115	192	360
	DEFUM	.76	1.1	2.5	8.6
	G-ERRM	1.9	1.5	3.7	8.5

Reliability of Error Control

RDC on pred-prey problem

Method	TOL :	10^{-2}	10^{-4}	10^{-6}	10^{-8}
CRK50:	G-ERRM	4.0	6.5	11.3	9.7
	K-ESTM	4.0	6.5	11.3	9.8
	DEFEM	.018	.017	.018	.019
	GE(U+E)	.02	.0005	.0005	.02
CRK60:	G-ERRM	2.9	3.1	2.6	2.3
	K-ESTM	2.8	3.1	2.6	2.3
	DEFEM	.016	.013	.015	.13
	GE(U+E)	.002	.004	.003	.012
CRK80:	G-ERRM	1.9	1.5	3.7	8.5
	K-ESTM	1.9	1.5	3.7	8.6
	DEFEM	.015	.015	.12	13.1
	GE(U+E)	.001	.0006	.009	.37

Validity of Condition Estimates

SDC on pred-prey problem

Method	TOL :	10^{-2}	10^{-4}	10^{-6}	10^{-8}
CRK52:	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
CRK52:	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
CRK82:	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

Reliability of Error Control

SDC on pred-prey problem

Method	TOL :	10^{-2}	10^{-4}	10^{-6}	10^{-8}
CRK52:	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
CRK62:	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
CRK82:	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

Validity of Condition Estimates

RDC on Lorenz problem

Method	TOL :	10^{-2}	10^{-4}	10^{-6}	10^{-8}
CRK50:	NS	339	706	1634	3996
	NSE	873	1988	4669	8206
	DEFUM	2.5	2.9	3.2	14.2
	G-ERRM	$4.4 \cdot 10^3$	$3.8 \cdot 10^5$	$2.7 \cdot 10^5$	$2.8 \cdot 10^5$
CRK60:	NS	327	666	1389	2962
	NSE	740	1345	2750	4606
	DEFUM	1.7	1.3	1.3	2.1
	G-ERRM	$4.3 \cdot 10^3$	$2.1 \cdot 10^5$	$1.0 \cdot 10^5$	$1.0 \cdot 10^5$
CRK80:	NS	147	223	351	614
	NSE	280	444	704	1222
	DEFUM	1.7	2.2	3.2	4.0
	G-ERRM	$4.6 \cdot 10^3$	$.29 \cdot 10^5$	$.38 \cdot 10^5$	$.5 \cdot 10^4$

Reliability of Error Control

RDC on Lorenz problem

Method	TOL :	10^{-2}	10^{-4}	10^{-6}	10^{-8}
CRK50:	G-ERRM	$4.4 \cdot 10^3$	$3.8 \cdot 10^5$	$2.7 \cdot 10^5$	$2.8 \cdot 10^5$
	K-ESTM	$4.4 \cdot 10^3$	$3.8 \cdot 10^5$	$2.7 \cdot 10^5$	$2.9 \cdot 10^5$
	DEFEM	.026	.038	.019	.24
	GE(U+E)	$.63 \cdot 10^1$	$.38 \cdot 10^2$	$.26 \cdot 10^2$	$.39 \cdot 10^4$
CRK60:	G-ERRM	$4.3 \cdot 10^3$	$2.1 \cdot 10^5$	$1.0 \cdot 10^5$	$1.0 \cdot 10^5$
	K-ESTM	$4.4 \cdot 10^3$	$2.1 \cdot 10^5$	$1.0 \cdot 10^5$	$1.1 \cdot 10^5$
	DEFEM	.016	.016	.017	.59
	GE(U+E)	$.76 \cdot 10^2$	$.24 \cdot 10^3$	$.67 \cdot 10^2$	$.46 \cdot 10^4$
CRK80:	G-ERRM	$4.6 \cdot 10^3$	$.29 \cdot 10^5$	$.38 \cdot 10^5$	$.5 \cdot 10^4$
	K-ESTM	$4.7 \cdot 10^3$	$.29 \cdot 10^5$	$.33 \cdot 10^5$	$1.9 \cdot 10^5$
	DEFEM	.047	.007	.52	60.0
	GE(U+E)	$.21 \cdot 10^3$	$.27 \cdot 10^2$	$.44 \cdot 10^4$	$1.9 \cdot 10^5$

Validity of Condition Estimates

SDC on Lorenz problem

Method	TOL :	10^{-2}	10^{-4}	10^{-6}	10^{-8}
CRK52:	NS	356	751	1738	4304
	NSE	834	1591	3470	4306
	DEFUM	1.3	1.4	1.4	1.4
	G-ERRM	$4.4 \cdot 10^3$	$1.9 \cdot 10^5$	$1.9 \cdot 10^5$	$1.8 \cdot 10^6$
CRK62:	NS	316	642	1339	2865
	NSE	731	1326	2678	2865
	DEFUM	1.4	1.3	1.3	1.2
	G-ERRM	$4.2 \cdot 10^3$	$2.8 \cdot 10^5$	$1.5 \cdot 10^5$	$1.5 \cdot 10^5$
CRK82:	NS	145	228	371	634
	NSE	292	454	803	1349
	DEFUM	1.3	1.4	1.6	1.4
	G-ERRM	$5.5 \cdot 10^3$	$.16 \cdot 10^5$	$.14 \cdot 10^5$	$.20 \cdot 10^5$

Reliability of Error Control

SDC on Lorenz problem

Method	TOL :	10^{-2}	10^{-4}	10^{-6}	10^{-8}
CRK52:	G-ERRM	$4.4 \cdot 10^3$	$1.9 \cdot 10^5$	$1.9 \cdot 10^5$	$1.8 \cdot 10^6$
	K-ESTM	$4.6 \cdot 10^3$	$1.9 \cdot 10^5$	$1.9 \cdot 10^5$	$1.9 \cdot 10^6$
	DEFEM	.016	.018	.020	.14
	GE(U+E)	$.47 \cdot 10^3$	$.50 \cdot 10^2$	$.17 \cdot 10^3$	$.68 \cdot 10^4$
CRK62:	G-ERRM	$4.2 \cdot 10^3$	$2.8 \cdot 10^5$	$1.5 \cdot 10^5$	$1.5 \cdot 10^5$
	K-ESTM	$4.2 \cdot 10^3$	$2.8 \cdot 10^5$	$1.5 \cdot 10^5$	$1.3 \cdot 10^5$
	DEFEM	.011	.011	.004	.20
	GE(U+E)	$.29 \cdot 10^3$	$.31 \cdot 10^3$	$.32 \cdot 10^3$	$.20 \cdot 10^5$
CRK82:	G-ERRM	$5.5 \cdot 10^3$	$.16 \cdot 10^5$	$.14 \cdot 10^5$	$.20 \cdot 10^5$
	K-ESTM	$5.5 \cdot 10^3$	$.16 \cdot 10^5$	$.15 \cdot 10^5$	$.70 \cdot 10^5$
	DEFEM	.013	.003	.076	9.0
	GE(U+E)	$.70 \cdot 10^2$	$.82 \cdot 10^1$	$.42 \cdot 10^3$	$.48 \cdot 10^5$

Validity of Condition Estimates

Conclusions and Observations

- Different CRKs with defect control compute numerical solutions with a consistent accuracy that is insensitive to the order or the number of steps.
- Generic global error estimates are available which provide a reliable indication of the conditioning of the problem at a cost that is comparable to that required to compute $U(x)$.
- On some problems (not those reported here) RDC may not be able to estimate the maximum defect well on all steps. SDC rarely has trouble with this.
- For high order methods, at severe values of TOL , round-off error may be significant and make it difficult to estimate the global error. This can easily be recognized and a warning flag raised.

Future Related Investigations

- Implement and test a parallel version of a CRK that uses the strategy outlined earlier.
- Investigate the use of this approach with multistep methods or with implicit methods designed for stiff problems. In these cases one must cope with an iteration error and/or a less smooth 'companion' error equation.
- Extend this approach to other classes of ODEs such as BVPs, DAEs or DDEs.
- Extend the approach to methods for PDEs.