Whats Next in the Evolution of 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 some advances in software development that are likely to occur in the near future. We will also identify some critical issues that must be resolved in order to make software more effective for DAEs, BVPs and DDEs.

We will focus on the class of CRK methods where we have the most experience but the issues we identify and address are, in most cases, also relevant to other software.

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

1. Some improvements for DAE software.

2. Some improvements for BVP software.

3. Some issues to be addressed in DDE software.

4. Related tools that could be used effectively in ODE software.

5. Extension of approach to 2D and 3D (PDEs).

$\boxed{\text{I) DAE Software:}}$

In recent years there has been considerable progress in the development of methods for special classes of DAEs. Nevertheless, few can be applied directly to systems of DAEs in the general form

$$F(t, y, y') = 0, \;\; y(t_0) = y_0, \;\; t \in [t_0, t_F],$$

with $\frac{\partial F}{\partial y'}$ known to be singular.

The 'index' of a problem refers to the minimum number of times one has to differentiate the equation in order to derive an equivalent IVP.

The higher the index the more sensitive the solution can be to perturbations in the data and the more difficult it becomes to develop reliable numerical methods.

Currently there are reliable methods for index 1 problems and other reliable methods for special classes of index 2 and 3 problems.

When one considers the development of defect-based error control for DAE methods two key questions must first be answered:

1. How does one define a sufficiently accurate continuous extension, $u_i(t)$, of the discrete approximation (for $t \in [t_{i-1}, t_i]$)?

2. What measure of the size of the defect is appropriate to control? That is, can one introduce a measure $\mu_i(\delta)$ such that for $t \in [t_{i-1}, t_i]$ the condition that $\mu_i(\delta(t)) \leq TOL$ will ensure that the global error will be proportional to TOL and $\mu_i(\delta)$ will be inexpensive to estimate on each step?

One approach has been introduced in the PhD thesis of C. Macdonald. To determine $u(t)$ one begins with a discrete IRK and an associated implicit CRK. One then introduces an associated overdetermined NLLSQ problem on each time step by requiring that the corresponding $\tilde{u}_i(t)$ satisfy (in a least squares sense) the defining equations of the underlying CRK as well as additional 'collocation' equations. The defect, $\tilde{\delta}(t)$, is then defined by

$$\tilde{\delta}(t) = F(t, \tilde{u}(t), \tilde{u}'(t)). \tag{1}$$

Conditions on the CRK formulas and on the collocation points are identified which result in $\|\tilde{\delta}(t)\|$ being $O(h^p)$. Estimates of $\|\tilde{\delta}(t)\|$ are justified and an experimental code introduced. A general purpose numerical method is planned.

Other critical issues to be addressed are the solution of the implicit equations and the automatic recognition of the index.

$\boxed{\text{II) BVP Software:}}$

New more accurate interpolants have been developed for Collocation methods. These methods (such as the COLSYS/COLNEW package) can be applied directly to high order systems:

$$y^{(m)} = f(t, y, y' \cdots y^{(m-1)}),$$

or to mixed order systems:

$$y_j^{(m_j)} = f_j(t, z(y)), \quad j = 1 \cdots d,$$

where

$$z(y) = [y_1, y_1^{'}, \cdots y_d^{(m_d-1)}].$$

It is well-known that collocation at the k 'Gauss points' is equivalent to the use of a k-stage IRK formula of order $2k$, while the associated collocation polynomial is only $O(h^{k+1})$.

In recent work we (with Muir and Sivasothinathan) and others (Pruess and Jin) have developed more accurate off-mesh interpolants. In particular we have used the ideas of explicit CRKs (applied to the equivalent IRK) to derive $O(h^{2k}), u(t)$.

- The approach has been extended to mixed order systems where $u_j(t) \in C^{m_j}[t_0, t_F]$,

- We obtain optimal off-mesh accuracy, $u^{(r)}(t) = y^{(r)}(t) + O(h^{2k-r})$, for $r = 0 \cdots m_j$.

- The associated continuous local error can be of order $2k$ or $2k+1$.

Consider the well-known ‘swirling flow’ test problem.

$$\epsilon f^{''''} + f f^{'''} + g g^{'} = 0,$$

$$\epsilon g^{''} + f g^{'} - f^{'} g = 0,$$

with $\epsilon = .075, f(0) = f(1) = f'(0) = f'(1) = 0, g(0) = -1, g(1) = 1$.

It is a mixed order, [4,2], system which can be reformulated as an equivalent 1^{st} order system $(d = 6)$ or as a 2^{nd} order system $(d = 3)$.

The cost and accuracy of our new interpolants when solving this problem using COLNEW is summarized in the following tables.

	For $O(h^{2k})$ interp				For $O(h^{2k+1})$ interp			
$mmin =$	1	2	3	4	1	2	3	4
$k=1$	1	-	-	-	1	-	-	-
$k=2$	1	1	-	-	2	1	-	-
$k=3$	3	1	1	-	6	2	1	-
$k=4$	8	3	1	1	13	6	2	1
$k=5$	16	8	3	1	23	13	6	2
$k=6$	27	16	8	3	36	23	13	6
$k=7$	41	27	16	8	52	36	23	13

Table 3: Extra stages required for accurate interpolants

m	Err Tol	N	Execution Time			Global Error in approximation		
			Col-new	Boots		Colnew	Boots	
				w/o le	w le		w/o le	w le
1,1,1, 1,1,1	10^{-4}	6	0.125	0.000	0.000	$1.6 \cdot 10^{-5}$	$8.8 \cdot 10^{-8}$	$2.0 \cdot 10^{-8}$
	10^{-6}	12	0.219	0.000	0.008	$6.4 \cdot 10^{-7}$	$3.5 \cdot 10^{-10}$	$8.0 \cdot 10^{-11}$
	10^{-8}	24	0.359	0.008	0.016	$2.3 \cdot 10^{-8}$	$1.5 \cdot 10^{-12}$	$3.4 \cdot 10^{-12}$
	10^{-10}	96	1.203	0.023	0.047	$2.4 \cdot 10^{-11}$	$2.5 \cdot 10^{-13}$	$7.9 \cdot 10^{-13}$
2,2,2	10^{-4}	6	0.055	0.008	0.000	$3.2 \cdot 10^{-7}$	$3.6 \cdot 10^{-8}$	$1.1 \cdot 10^{-8}$
	10^{-6}	6	0.055	0.000	0.008	$3.2 \cdot 10^{-7}$	$3.6 \cdot 10^{-8}$	$1.1 \cdot 10^{-8}$
	10^{-8}	12	0.094	0.000	0.000	$6.7 \cdot 10^{-9}$	$1.6 \cdot 10^{-10}$	$4.4 \cdot 10^{-11}$
	10^{-10}	24	0.156	0.000	0.000	$1.2 \cdot 10^{-10}$	$7.1 \cdot 10^{-13}$	$1.9 \cdot 10^{-13}$
2,4	10^{-4}	6	0.031	0.000	0.000	$1.6 \cdot 10^{-7}$	$5.6 \cdot 10^{-10}$	$1.6 \cdot 10^{-10}$
	10^{-6}	6	0.031	0.000	0.000	$1.6 \cdot 10^{-7}$	$5.6 \cdot 10^{-10}$	$1.6 \cdot 10^{-10}$
	10^{-8}	12	0.062	0.000	0.008	$2.8 \cdot 10^{-9}$	$4.1 \cdot 10^{-12}$	$6.1 \cdot 10^{-13}$
	10^{-10}	24	0.141	0.000	0.008	$4.7 \cdot 10^{-11}$	$2.2 \cdot 10^{-14}$	$9.7 \cdot 10^{-15}$

Table 19. Adaptive mesh results of problem 1 for k=4 (Double precision)

These results show that these interpolants provide, at very little extra cost, more accurate approximations even at coarse meshes. The next step is to investigate how the internal strategies/heuristics of COLNEW can be modified to make direct use of these interpolants.

- Mesh selection,
- Global error estimation,
- Defect control,
- Starting values for the nonlinear equations.

$\boxed{\text{III) DDE Software:}}$

Several complications can arise when solving DDEs that are not handled well by current software. Some of these difficulties can be avoided by restricting the class of problems. (For example in developing a DDE solver, dde23, for MATLAB, Shampine and Thompson consider only constant delay problems.) Examples of such complications where better heuristics are needed are:

- Implicit equations for small delays.

- The choice of 'value' for y'_{i-j} when a discontinuity is crossed.

- On problems with multiple delays, different components may be signaling the same discontinuity and this must be recognized.

IV) Related tools that could prove useful.

- Technique to verify the consistency of the partial derivative and derivative routines.

- Automatic generation of code for analytic partial derivatives. This could be used by virtually all BV and DAE methods.

- The derivation of generic drivers for tasks such as continuation and parameter fitting. This is possible once a standard interface is adopted for ODE software.

V) Generalization to PDEs:

Visualization of approximate PDE solutions in two and three dimensions involves the use of surface plots, contour plots and/or animation. To do this efficiently we have generalized the idea of a continuous extension of a discrete ODE solution. That is we have derived high order multivariate piecewise polynomials that interpolate the underlying discrete solution defined on an arbitrary mesh.

The resulting accurate interpolant can be used by the visualization packages without requiring any further computation of the PDE solver.