On the Ilie–Corless Polynomial Complexity Proof

Solving ODEs or DAEs by computing the Taylor series is numerically stable

by John Pryce
with Ned Nedialkov and Wayne Enright

Given at University of Gent, Dept of Mathematics and Computer Science
28 August 2007

This work was supported in part by grants from
Leverhulme Trust
and
UK Engineering and Physical Sciences Research Council

Aim of this work

This is about methods for ODEs and DAEs based on Taylor series expansion to some order p at each step — "order p Taylor"

Ilie–Corless have an asymptotic complexity proof. Suppose:

- ODE/DAE is (piecewise) analytic;
- Problem is an IVP, and solution $y(t)$ exists on $I = [0, T]$, say.

Then the time-complexity of computing y is polynomial in the number of digits of accuracy

They use defect-based error control

My aim is to point out a gap in the proof and how I believe it can be filled

Some definitions

Differential Algebraic Equation (DAE): System of equations for $x_j(t)$, $j = 1, \ldots, n$, of perhaps fully implicit form

$$f_i(\, t, \text{ the } x_j \text{ and derivatives of them}\,) = 0, \quad 1 \leq i \leq n.$$

that is SA-friendly in the sense that Pryce's Structural Analysis (SA) approach succeeds on it.

SA-friendly includes explicit ODEs and many standard DAE classes e.g. semi-explicit index 1 or Hessenberg or index 3 mechanical systems.

Polynomial Time: Over a given finite interval I, one can construct a function that approximates the true solution correct to N (binary) digits, uniformly on I, in time bounded by some power of N

History

- ∼2003. Rob Corless (U. W Ontario): this complexity result for ODEs.
- ∼2005. Silvana Ilie gives me her extension to semi-explicit index 1 DAE.
- 3/07. Enright, Nedialkov, Pryce. DAETS code — defect control approach? I start extending Ilie proof to general DAE case.
- 4,5/07. I outline proof at AD-Hatfield: comments by Uwe Naumann.
- 5/07. Jacques Carette points out roundoff analysis gap: "but I don't do roundoff".
- 5/07. I get Ilie's proof for general DAE. But still has gap!
- 6/07. Andreas Griewank: "I think this is still an unsolved problem".
- 6,7/07. I reckon I can plug gap.

Polynomial complexity requires unbounded order

Methods of fixed (or variable but bounded) order won't do, e.g. suppose we use a 4th order Runge–Kutta method. Then

$$\text{Work } W \propto 1/(\text{Average step size } H)$$
$$\text{Global Error } E \propto H^4$$
$$\text{No. of digits accurate } N = -\log_2(E)$$

whence

$$W \propto 2^{N/4}$$

i.e. exponential complexity!

Variable order is more powerful

Corless–Ilie use elegant defect-equidistribution argument: implicitly assumes # of steps $\to \infty$ as accuracy $\to \infty$. My argument seems simpler, as follows:

By assumptions, solution is analytic vector function on $I : 0 \le t \le T$. So extends to analytic function in complex t-plane, whence Radius of Convergence function

$$\rho(t) = (\text{Distance from } t \text{ to nearest singularity})$$

is continuous and > 0 on $[0, T]$, so bounded above 0

So for any (small!) $\theta > 0$ a "$\rho(t)$ oracle" can give me a priori a mesh $0 = t_0 < t_1 < \ldots < t_m = T$ such that each step is $< \theta\times$(local radius of convergence) whence on each step $s = 1, \ldots, m$

$$(p\text{th TS term}) < C\,\theta^p \qquad (p \to \infty), \text{ where } C = C_s \text{ depends on } s$$

Resulting complexity . . . ?

Vary Taylor Series order p. On each step

$$\text{local error} = (\text{sum of dropped TS terms}) = \text{roughly } \propto \theta^p$$

which accumulates over our fixed mesh to give

$$\text{Global error is roughly } \propto \theta^p, \quad (p \to \infty)$$

So p-order TS gives $N \approx p$ (binary) digits of accuracy

Cost of p-order TS on a fixed function $\mathbf{f}$, using AD, is $\leq C\,L\,p^2$ arithmetic operations where L is length of $\mathbf{f}$'s code list and C a modest constant

One arithmetic op in N-digit arithmetic is at most $O(N^2)$ time

So with $p = N$ get about N binary digits of accuracy in

$$O(p^2 N^2) = O(N^4) \text{ time}$$

Polynomial complexity! but not so fast . . .

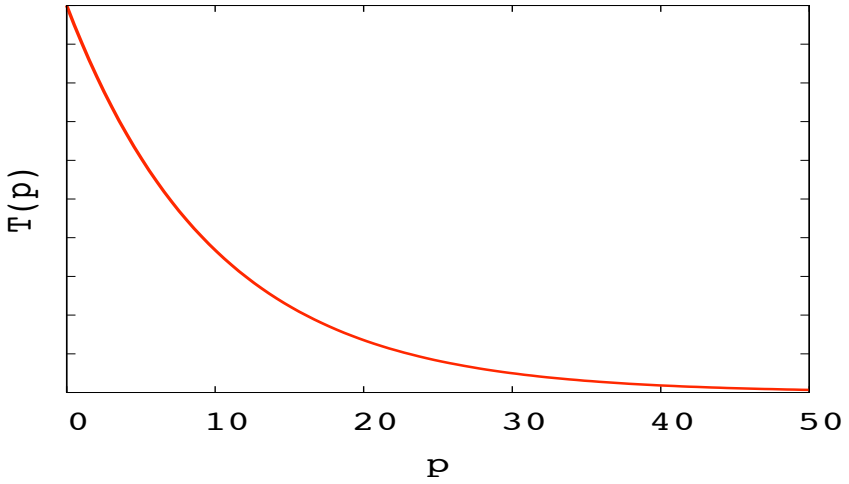
Why there's a real difficulty

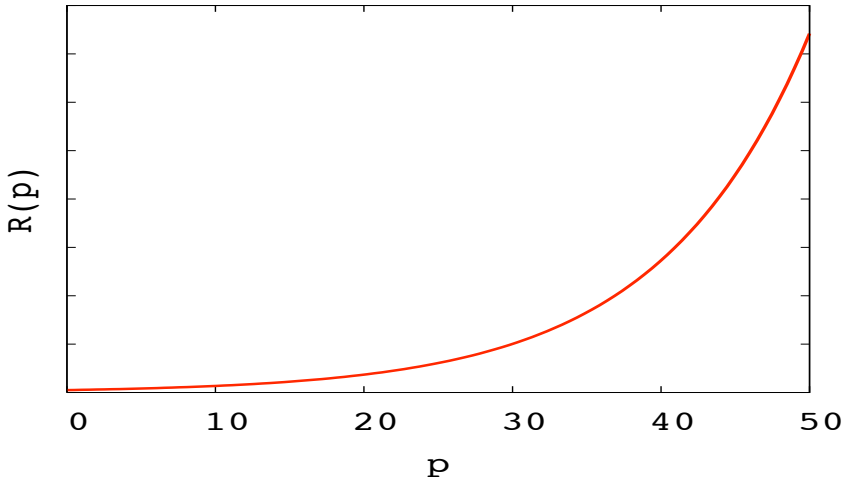
The question: Can the effects of roundoff grow very fast as $p \to \infty$?

Rough model: For given step h within radius of convergence let

$$T(p)\ (\to 0) = \text{ truncation error of Taylor}(p) \text{ with exact arithmetic}$$
$$R(p)\ (\to \infty?) = \text{ s.t. total roundoff error of Taylor}(p) \text{ is } \sim R(p)u$$

where u is roundoff unit: $u = 2^{-N}$ where N is #bits of precision





Then total error at order p with N-bit arith is $\approx \epsilon(p, N) = (T(p) + R(p)2^{-N})$

Result: If $R(p)$ grows horribly — e.g. as $p!$ — work needed to get

$$M = -\log_2(\epsilon(p, N)) \text{ bits accurate}$$

(even just on one step) cannot be polynomial in M

Possible sources: complex code list and/or cancellation in summing

I aim to show this cannot happen

Tackling this

Still technical snags for general DAE so will outline for explicit ODE case $\mathbf{x}' = \mathbf{f}(\mathbf{x})$.
Method in outline:

1. Convert $\mathbf{f}$ to basic code list involving only $+$ $-$ $\times$ $\div$
2. Remove $-$ and $\div$, now only $+$ and $\times$
3. Regard result as a DAE (always SA-friendly), apply Pryce method to it
4. Now System Jacobian $\mathbf{J}$ encapsulates much of bad roundoff behaviour
5. Scale independent variable so TCs are same as TS terms, i.e. $h = 1$
6. Regard TS term recurrences as infinite block-triangular system
7. Inverse of its block-triangular Jacobian gives bound on roundoff
8. I prove a technical result that bounds this inverse

Simple example

ODE is	Code list	As DAE
$x_1' = x_2 + x_1/x_2$	$v_1 = x_1/x_2$	$0 = V_1 = -x_1 + v_1 x_2$
$x_2' = x_1 x_2 - x_1$	$v_2 = x_2 + v_1$	$0 = V_2 = -v_2 + x_2 + v_1$
	$v_3 = x_1 x_2$	$0 = V_3 = -v_3 + x_1 x_2$
	$v_4 = v_3 - x_1$	$0 = V_4 = -v_3 + v_4 + x_1$
	$x_1' = v_2$	$0 = F_1 = x_1' - v_2$
	$x_2' = v_4$	$0 = F_2 = x_2' - v_4$

DAE's Signature Tableau

	v_1	v_2	v_3	v_4	x_1	x_2	c_i
V_1	0^*	$-$	$-$	$-$	0	0	0
V_2	0	0^*	$-$	$-$	$-$	0	0
V_3	$-$	$-$	0^*	$-$	0	0	0
V_4	$-$	$-$	0	0^*	0	$-$	0
F_1	$-$	0	$-$	$-$	1^*	$-$	0
F_2	$-$	$-$	$-$	0	$-$	1^*	0
d_j	0	0	0	0	1	1	

($-$ means $-\infty$)

J = System Jacobian

	v_1	v_2	v_3	v_4	x_1	x_2
V_1	$x_{2,0}$					
V_2	1	-1				
V_3			-1			
V_4				1		
F_1		-1			1	
F_2				-1		1

(blank means zero)

Simple example, cont.

Denote Taylor coefficients of v_1 by $(v_{1,0}\,, v_{1,1}\,, v_{1,2}\,, \ldots)$ and so on

By Pryce method, solution scheme is specified by offsets thus:

Stage $k = -1$: Take $x_{1,0}\,, x_{2,0}$ as initial values

Stages $k = 0, 1, \ldots$: Solve for the highlit items in

$$\left.\begin{array}{rcl}
0 = V_{1,k} &=& -x_{1,k} + (v_{1,k}\, x_{2,0} + \cdots + v_{1,0}\, x_{2,k}\,) \\
0 = V_{2,k} &=& -v_{2,k} + x_{2,k} + v_{1,k} \\
0 = V_{3,k} &=& -v_{3,k} + (x_{1,k}\, x_{2,0} + \cdots + x_{1,0}\, x_{2,k}\,) \\
0 = V_{4,k} &=& -v_{3,k} + v_{4,k} + x_{1,k} \\
0 = F_{1,k} &=& (k+1) x_{1,k+1} - v_{2,k} \\
0 = F_{2,k} &=& (k+1) x_{2,k+1} - v_{4,k}
\end{array}\right\} \text{ for } \begin{bmatrix} v_{1,k} \\ v_{2,k} \\ v_{3,k} \\ v_{4,k} \\ x_{1,k+1} \\ x_{2,k+1} \end{bmatrix} = \mathbf{x}_k, \text{ say}$$

— items in black known from previous stages

The block-triangular system

These equations for the Taylor coefficients have the form

$$\mathbf{F}_k\Big(\ldots, \mathbf{x}_{k-1}, \mathbf{x}_k\Big) = \mathbf{J} D_k \mathbf{x}_k + \mathbf{G}\Big(\ldots, \mathbf{x}_{k-1}\Big) = \mathbf{0} \qquad (k = 0, 1, 2, \ldots)$$

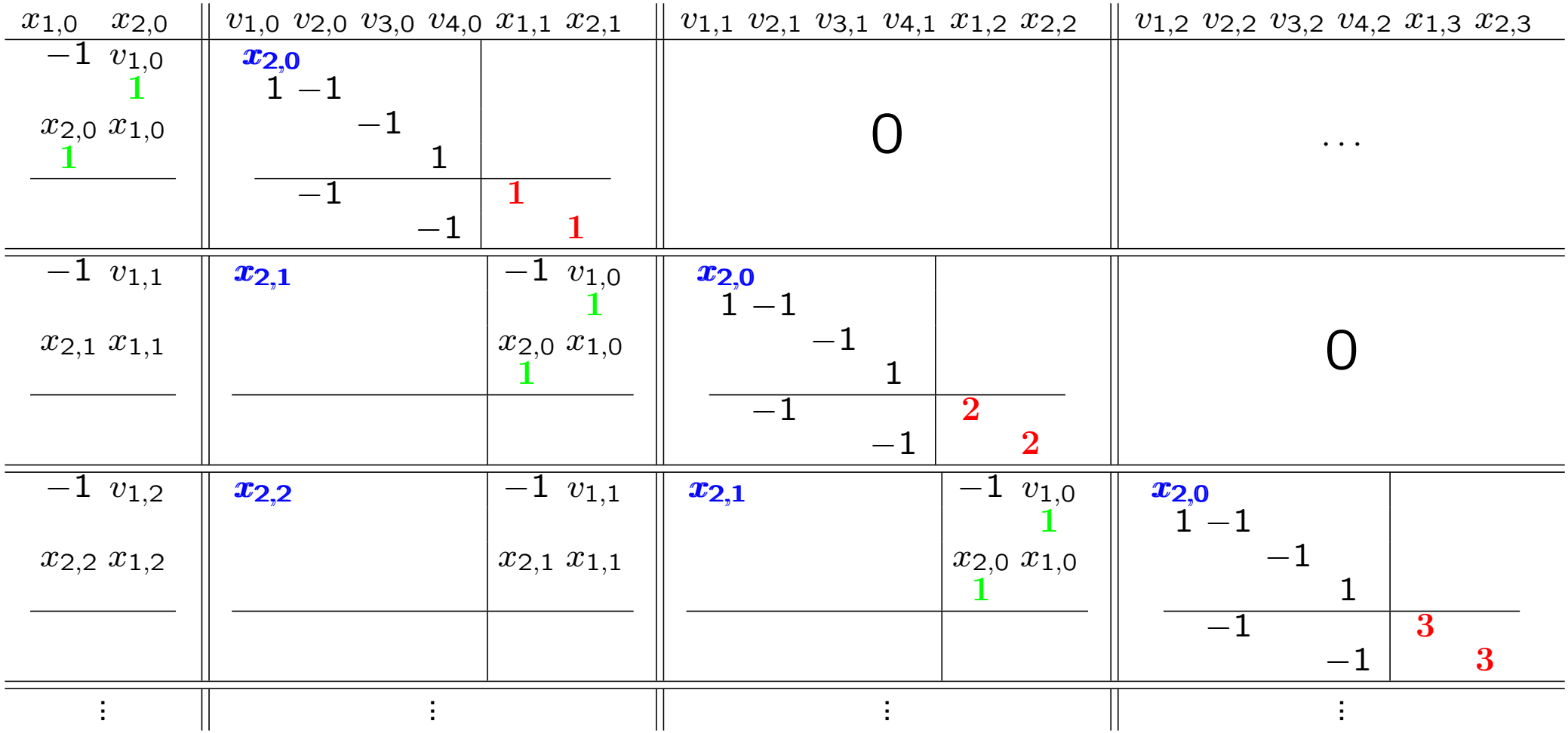
where $\mathbf{J}$ is System Jacobian and $D_k = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & k{+}1 & & \\ & & & & & k{+}1 & \end{bmatrix}$

These form an infinite block triangular system

$$\mathbf{0} = \begin{pmatrix} \mathbf{F}_0\left(\mathbf{x}_{-1}, \mathbf{x}_0\right) \\ \mathbf{F}_1\left(\mathbf{x}_{-1}, \mathbf{x}_0, \mathbf{x}_1\right) \\ \mathbf{F}_2\left(\mathbf{x}_{-1}, \mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2\right) \\ \cdots \end{pmatrix} = \mathbf{F}(\mathbf{x}), \quad \text{where } \mathbf{x} = \begin{pmatrix} \mathbf{x}_{-1} \\ \mathbf{x}_0 \\ \mathbf{x}_1 \\ \cdots \end{pmatrix}$$

Key to roundoff analysis is its "big Jacobian" $\dfrac{\partial \mathbf{F}}{\partial \mathbf{x}} = \left(\dfrac{\partial \mathbf{F}_i}{\partial \mathbf{x}_j}\right)_{i \geq 0,\, j \geq -1}$

Big Jac $\partial \mathbf{F}/\partial \mathbf{x}$



This typifies the pattern for a general ODE

Big Jac cont.

Omitting left column, which shows sensitivity to initial values, "big Jac" is

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}} = \mathbf{A} = \begin{bmatrix} \mathbf{J}D_0 & 0 & \cdots & \\ A_{1,0} & \mathbf{J}D_1 & 0 & \cdots \\ A_{2,0} & A_{2,1} & \mathbf{J}D_2 & 0 \\ A_{3,0} & A_{3,1} & A_{3,1} & \mathbf{J}D_3 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

where $\mathbf{J}$ is nonsingular, assuming initial value $x_{2,0}$ of x_2 is $\neq 0$. So are D_k.

Crucial point (see pattern of entries in "big Jac"):

The A_{ij}, with exact computation, decrease geometrically off the diagonal — if step size h satisfies $0 < h < \theta \times$ (radius of convergence) then

$$\|A_{j+p,j}\| \leq \alpha\, \theta^p \qquad\qquad (i = 0, 1, \ldots;\ p = 1, 2, \ldots)$$

for some $\alpha \geq 0$. Clearly $\theta > 0$ can be as small as we like.

Roundoff analysis

Model roundoff by saying the actual computed values are $\bar{\mathbf{x}}_k$, that satisfy

$$\mathbf{F}_k\Big(\bar{\mathbf{x}}_k,\ \bar{\mathbf{x}}_{k-1},\ \ldots\Big) \ = \ \mathbf{J}D_k\,\bar{\mathbf{x}}_k + \mathbf{G}\Big(\bar{\mathbf{x}}_{k-1},\ \ldots\Big) \ = \ \boldsymbol{\delta}_k \qquad (k = 0, 1, 2, \ldots)$$

where $\boldsymbol{\delta}_k$ comes from roundoff in computing $\mathbf{G}$ and solving the linear system with $\mathbf{J}D_k$.

By MVT argument, errors $\boldsymbol{\xi}_k = \bar{\mathbf{x}}_k - \mathbf{x}_k$ satisfy the block triangular system

$$\bar{\mathbf{A}}\boldsymbol{\xi} = \begin{bmatrix} \mathbf{J}D_0 & 0 & \cdots & \\ \bar{A}_{1,0} & \mathbf{J}D_1 & 0 & \cdots \\ \bar{A}_{2,0} & \bar{A}_{2,1} & \mathbf{J}D_2 & 0 \\ \bar{A}_{3,0} & \bar{A}_{3,1} & \bar{A}_{3,1} & \mathbf{J}D_3 \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{pmatrix} \boldsymbol{\xi}_0 \\ \boldsymbol{\xi}_1 \\ \boldsymbol{\xi}_2 \\ \boldsymbol{\xi}_3 \\ \ldots \end{pmatrix} = \begin{pmatrix} \boldsymbol{\delta}_0 \\ \boldsymbol{\delta}_1 \\ \boldsymbol{\delta}_2 \\ \boldsymbol{\delta}_3 \\ \ldots \end{pmatrix} = \boldsymbol{\delta}$$

where the matrix is an average of $\partial\mathbf{F}/\partial\mathbf{x}$ values between $\mathbf{x}_k$ and $\bar{\mathbf{x}}_k = \mathbf{x}_k + \boldsymbol{\xi}_k$

Key bound

Theorem 1 *The inverse of (exact computation)* $\mathbf{A}$ *has the form*

$$\mathbf{A}^{-1} = \begin{bmatrix} (\mathbf{J}D_0)^{-1} & 0 & \cdots & \\ B_{1,0} & (\mathbf{J}D_1)^{-1} & 0 & \cdots \\ B_{2,0} & B_{2,1} & (\mathbf{J}D_2)^{-1} & 0 \\ B_{3,0} & B_{3,1} & B_{3,1} & (\mathbf{J}D_3)^{-1} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

where the B_{ij} *decrease geometrically, with a different constant. Namely*

$$\|B_{j+p,j}\| \le \beta\,\phi^p \qquad (i = 0, 1, \ldots;\ p = 1, 2, \ldots)$$

where

$$\phi = \left(1 + \alpha\|\mathbf{J}^{-1}\|\right)\theta.$$

In difficult cases $\|\mathbf{J}^{-1}\|$ may be large, and α may be astronomically large. (Solving $y' = -y$ on 0 to 1000 in one step gives $\alpha \approx e^{2000}$ I think.)

But this represents a fixed overhead of $\log_2\left(1 + \alpha\|\mathbf{J}^{-1}\|\right)$ extra bits of precision, so no difficulty in theory as required accuracy $\to \infty$.

Bounding the effect of roundoff errors

In bounding $\bar{\mathbf{A}}^{-1}$ the snag is the feedback between $\boldsymbol{\xi}$, $\boldsymbol{\delta}$ and $\bar{\mathbf{A}}$

When one tries to apply Theorem 1 to $\bar{\mathbf{A}}$, the bounds on $B_{j+p,j}$ are gradually degraded by roundoff

The smaller is the roundoff unit u, the larger p can become before this happens

The key point is to show the needed u for a given p is sufficiently large that the "real difficulty" in Slide 8 is overcome

Bounding the effect of roundoff errors, cont

The worst case in the kth block of $\boldsymbol{\delta}$ comes from convolutions from a "multiply" operation like

$$c_k = a_0 b_k + a_{k-1} b_1 + \cdots + a_0 b_k$$

If everything up to here had been done exactly, roundoff error in doing the RHS in floating point would be bounded by

$$\begin{aligned} & 2\left(|a_0| \cdot |b_k| + |a_{k-1}| \cdot |b_1| + \cdots + |a_0| \cdot |b_k|\right) u \\ \leq\ & 2\left(\alpha \cdot \alpha\theta^k + \alpha\theta \cdot \alpha\theta^{k-1} + \cdots + \alpha\theta^k \cdot \alpha\right) u \\ =\ & 2\,\alpha^2\,(k+1)\,\theta^k\,u \end{aligned}$$

Bounding the effect of roundoff errors, cont

Assume (inductively on k) roundoff has contaminated the RHS values by at most 0.4$\times$ the bounds on their true values, then this bound is increased by a factor

$$\leq (1 + 0.4)^2 < 2$$

so the error in the actual computed value

$$\bar{c}_k = \bar{a}_0\bar{b}_k + \bar{a}_{k-1}\bar{b}_1 + \cdots + \bar{a}_0\bar{b}_k$$

is at most twice the above bound.

Doing the solve with $(\mathbf{J}D_k)$ multiplies the bound by a factor involving the condition number $\kappa(\mathbf{J}) = \|\mathbf{J}^{-1}\|.\|\mathbf{J}\|$. Overall this gives

$$\|\boldsymbol{\xi}_k\| \leq C\, k\theta^k\, u$$

with a possibly huge C depending only on the problem, provided k is small enough.

Bounding the effect of roundoff errors, cont

The ξ_k feed back to make $\|\bar{A}_{j+p,j}\|$ at most twice the exact-computation value, provided $j+p$ is small enough, whence the inverse of actual $\bar{\mathbf{A}}$ has the form

$$\bar{\mathbf{A}}^{-1} = \begin{bmatrix} (\mathbf{J}D_0)^{-1} & 0 & \cdots & \\ \bar{B}_{1,0} & (\mathbf{J}D_1)^{-1} & 0 & \cdots \\ \bar{B}_{2,0} & \bar{B}_{2,1} & (\mathbf{J}D_2)^{-1} & 0 \\ \bar{B}_{3,0} & \bar{B}_{3,1} & \bar{B}_{3,1} & (\mathbf{J}D_3)^{-1} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

where

$$\|\bar{B}_{j+p,j}\| \le \bar{\beta}\,\bar{\phi}^{\,p} \qquad\qquad (i = 0, 1, \ldots;\ p = 1, 2, \ldots)$$

for small enough $j+p$, with

$$\bar{\phi} = \left(1 + 2\alpha\|\mathbf{J}^{-1}\|\right)\theta.$$

Bounding the effect of roundoff errors, cont

These bounds apply to a single step from t to $t+h$ where θ is essentially $h/$(local radius of convergence $\rho(t)$)

But all values involved are continuously functions of t, on interval $[0, T]$.

So a compactness argument shows there is a finite mesh (t_i), with associated θ_i and corresponding $\bar{\phi}_i$ on ith subinterval, such that

$$\bar{\phi}_i < \frac{1}{2} \quad \text{for all } i$$

and that "small enough k" means

$$(\text{problem-dependent const}) \times \left(\frac{\bar{\phi}_i}{\theta_i}\right)^k \times u < 1$$

which with order $p =$ (largest k) and $N = -\log_2 u$ bits of precision means one can take

$$N = C_1 \times p + C_2 \quad \text{for all } p$$

where C_1, C_2 depend purely on the problem

Bounding the effect of roundoff errors, cont

The condition $\bar{\phi}_i < \frac{1}{2}$ is used to ensure that the computed Taylor terms $\bar{x}_k$, even with roundoff contamination, decay at least like 2^{-k}, which bounds the roundoff in the final process of summing them.

Hence Taylor order p, with $(C_1 p + C_2)$-bit floating point, delivers about p correct bits in the solution, in the absolute error sense . . .

. . . and does so in time bounded by $O(p^4)$.

Summary

Taylor coefficient computation is numerically stable, for sufficiently small stepsize, in an asymptotic complexity sense

The gap in the Ilie–Corless proof is plugged, at least for ODEs

DAEs to follow