Rigorous Defect Control and the Numerical Solution of ODEs

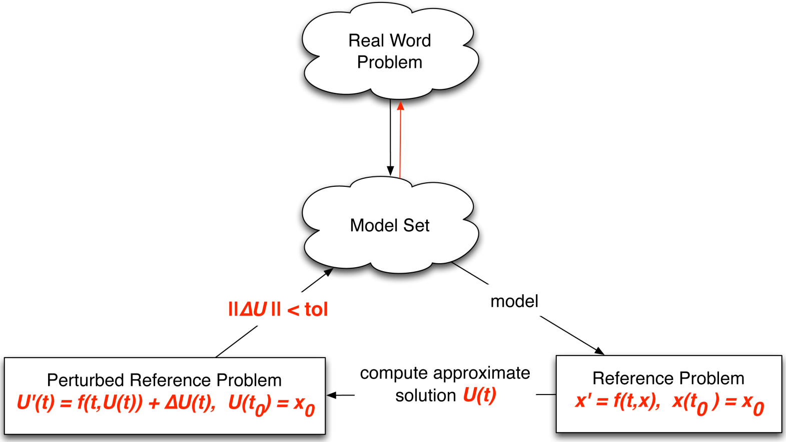
John Ernsthausen

McMaster University

johnernsthausen.com/sonad-talk.pdf

SONAD 2017
May 19, 2017

Problem statement



Defect control literature

- Enright advocates asymptotic defect control
  Enright and Coworkers and Students (1989-2012)
- Defect control and ODE boundary value problem
  Enright and Muir, Shampine and Muir (1993-2004)
- Corless and Corliss proposed rigorous defect control
  Corless and Corliss (1991)

Numerical problem

Given

$$x'(t) = f(t, x(t)) \quad x(t_0) = x_0 \quad t_{\text{end}} > t_0 \quad \text{tol} > 0$$

compute approximate u on $[t_i, t_{i+1}]$ near x_i and compute defect

$$\Delta u(t) \stackrel{\text{def}}{=} u'(t) - f(t, u(t))$$

Find stepsize so that u satisfies on $[t_i, t_{i+1}]$

$$u'(t) = f(t, u(t)) + \Delta u(t) \quad u(t_i) = x_i \quad \|\Delta u\|_\infty \le \text{tol}$$

Then u exactly solves “nearby” problem on $[t_0, t_{\text{end}}]$

$$u'(t) = f(t, u(t)) + \Delta u(t) \quad u(t_0) = x_0 \quad \|\Delta u\|_\infty \le \text{tol}$$

How to do it?

- Construct approximate solution u
- Bound $\|\Delta u(t)\|_\infty \leq$ tol rigorously on $[t_0, t_{\text{end}}]$
- Find good stepsize

Approximate solution

Good numerical ODE solvers for the initial value problem

$$x'(t) = f\big(t, x(t)\big) \quad x(t_0) = x_0$$

- control local error on each step
- return skeletal solution $\big(t_j, x_j\big)$
- return a continuously differentiable approximation u to x

Defect control (DC) methods

- monitor and control the maximum magnitude of the defect
- Asymptotic DC estimates $\|\Delta u\|_\infty$ by evaluating it at carefully selected points in each integration interval
- Rigorous DC ensures $\|\Delta u(t)\| \leq$ tol on $[t_0, t_{\text{end}}]$

Taylor series method

Computation often regarded as expensive

This is not the case

Computing defect inexpensive

Compared to cost of Taylor series method itself

$$u(t) = \sum_{k=0}^{n} (u)_k (t - t_i)^k \quad \text{where} \quad (u)_k = \frac{1}{k} (f)_{k-1}$$

Data management: ApproximateSolution class

Automatic differentiation via operator overloading

From f to its Computational Graph, a DAG
Bendtsen and Stauning [FADBAD++, TADIFF] (1997)

Idea: Taylor arithmetic

- Assume user equations are elementary functions
- Construct an efficient computational graph
- Nodes (basic functions): sin, asin, sqrt, pow, log, exp
- Edges (basic operators): add, sub, mul, div, composition

Interface to TADIFF: TaylorExpansion class

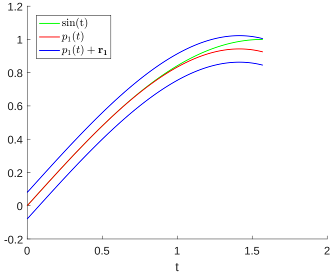
RPA based on Taylor Models (TMs)

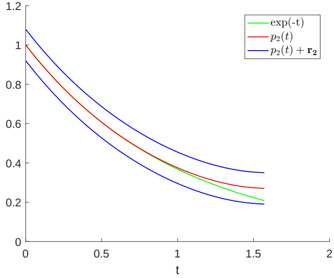
TMs: Berz & Makino, … RPA: Joldes, …

- Represent a function on $[a, b]$ as a Taylor polynomial + interval error bound:

$$(p, \boldsymbol{r}) \quad \text{means} \quad f(t) - p(t) \in \boldsymbol{r} = [\,\underline{\boldsymbol{r}}, \bar{\boldsymbol{r}}\,] \quad \text{for all } t \in [a, b]$$

- TMs of degree 4 for $\sin(t)$ and $\exp(-t)$ on $[0, \pi/2]$





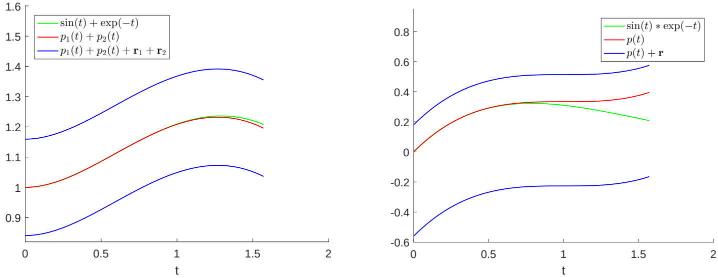
Arithmetic operations with TMs

- E.g. Addition $f(t) - p_1(t) \in \mathbf{r}_1, \quad g(t) - p_2(t) \in \mathbf{r}_2$:

$$f(t) + g(t) - \big(p_1(t) + p_2(t)\big) \in \mathbf{r}_1 + \mathbf{r}_2$$

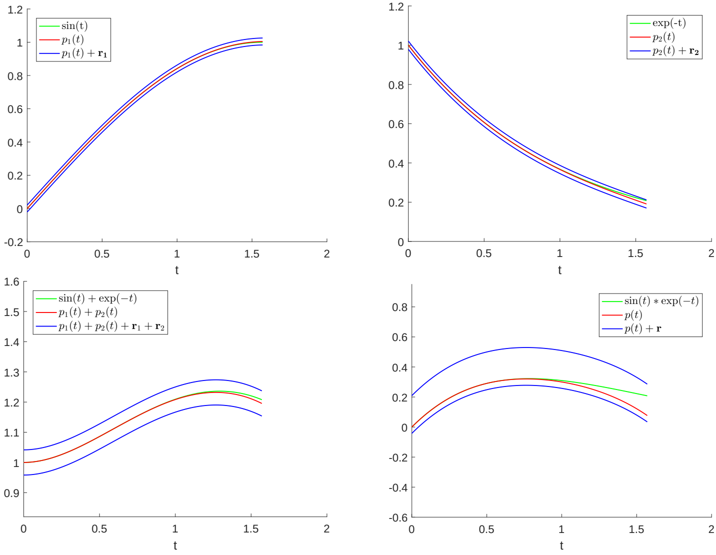
- Multiplication, division, elementary function: construct polynomial part and bound remainder terms

  Berz & Makino, Joldes

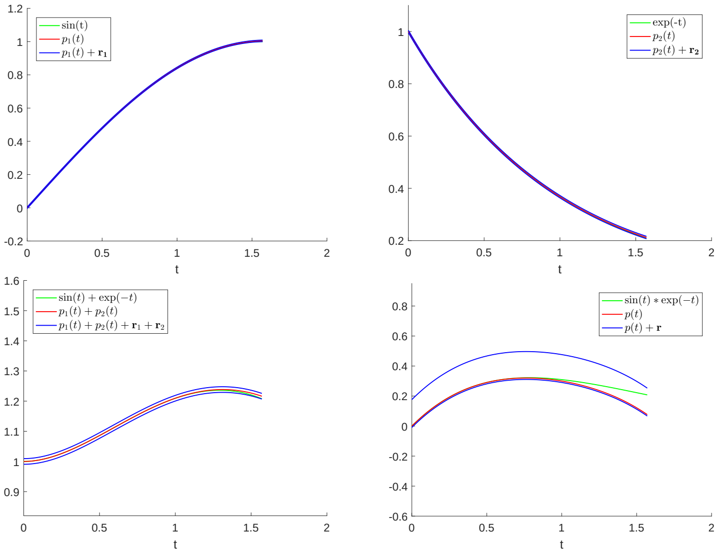


(a) addition (b) muliplication

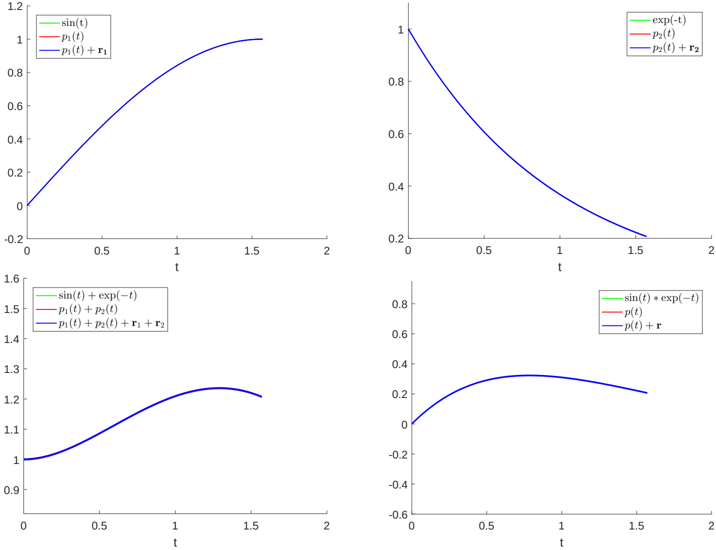
Example: TMs of degree 5



Example: TMs of degree 6



Example: TMs of degree 7



Our process

On each integration interval $[t_j, t_{j+1}]$

Phase I: Compute approximate (polynomial) solution
We use Taylor series $h_j = t_{j+1} - t_j$

$$u(t) = u_0 + u_1(t - t_j) + \cdots + u_p(t - t_j)^k \quad t \in [0, h_j]$$

- u_0 initial condition at t_j
- u_i Taylor coefficients at t_j
- computed using automatic differentiation and FADBAD++
  Bendtsen and Stauning

Interpolate $f(t_{j+1}, u(t_{j+1}))$:

$$U(t) = u(t) + \frac{\Delta u(t_{j+1})}{h_j^k}(t - t_j)^{k+1} - \frac{\Delta u(t_{j+1})}{h_j^{k+1}}(t - t_i)^{k+2}$$

Our process cont.

Phase II: Bound the defect

- Evaluate code list of $x' - f(t,x)$ with input $\big(U,[0,0]\big)$ in TM arithmetic using SOLLYA package
  Chevillard, Lauter, Joldes

  For each component of the solution, the result is a polynomial p and a remainder bound $\boldsymbol{r}$:

$$\Delta U(t) - p(t) = [U'(t) - f(t, U(t))] - p(t) \in \boldsymbol{r} \text{ on } [t_j, t_{j+1}]$$

- Compute using SOLLYA package

$$\text{rigorous bound} \quad \bar{p} \geq \|p\|_\infty = \sup_{t\in[t_j,t_{j+1}]} |p(t)|$$

- Then

$$\|\Delta U\|_\infty \leq \delta := \overline{p} + |\boldsymbol{r}|, \qquad |\boldsymbol{r}| = \max\{|\underline{\boldsymbol{r}}|, |\overline{\boldsymbol{r}}|\}$$

Example

Consider

$$x'(t) = f(t, x(t)) = x(t) - x(t)^2 \qquad x(0) = 0.2$$

and

$$u(t) = 0.2 + 0.16t + 0.048t^2 + 1.0667\times10^{-3}t^3 \qquad [t_0, t_1] = [0, 0.4]$$

First three coefficients exact; last rounded to 4 digits

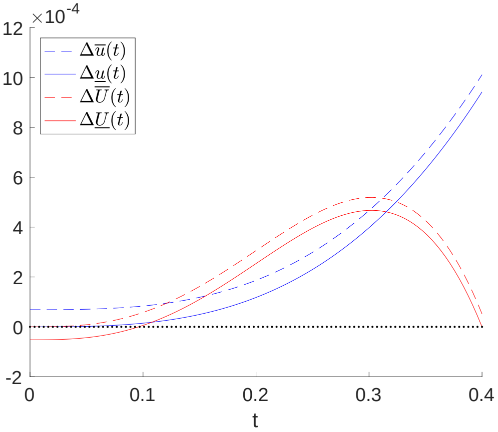
Interpolating $f(t_1, u(t_1))$ (4 digits)

$$U(t) = v(t) + 1.5795\times10^{-2}t^4 - 3.9486\times10^{-2}t^5$$

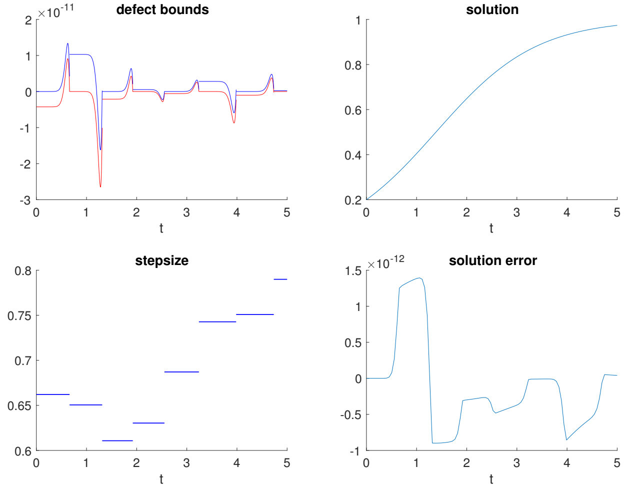
Evaluating $x' - (x - x^2)$ with $(U, [0, 0])$ on $[0, 0.4]$

$$p(t) = 1.3878\times10^{-17}t + 10^{-10}t^2 + 7.7898\times10^{-2}t^3 - 2.0426\times10^{-1}t^4 + 2.8849\times10^{-2}t^5$$

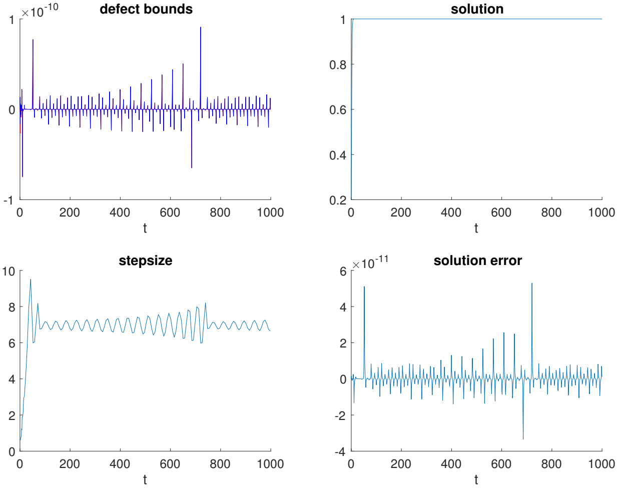
$$\boldsymbol{r} = [-5.1923\times10^{-5}, 1.8090\times10^{-17}]$$



Enclosures of $\Delta u(t)$ (blue) and $\Delta U(t)$ (red)



$$x' = x - x^2,\ x(0) = 0.2,\ \text{order } 15,\ \text{tol} = 10^{-10}$$



$$x' = x - x^2,\ x(0) = 0.2,\ \text{order } 15,\ \text{tol} = 10^{-10}$$

Stepsize control

We use an "elementary" stepsize controller

- Set $\delta_{\max} = \max_j \delta_j$
  δ_j bounds jth solution component defect
- If $\delta_{\max} \le \text{tol}$, accept h_j and

$$h_{j+1} = 0.9\, h \left(\frac{0.5\ \text{tol}}{\delta_{max}} \right)^{1/k}$$

- else reject step and recompute $\delta_{\max}$ with

$$h_j \leftarrow h_j \left(\frac{0.25\ \text{tol}}{\delta_{max}} \right)^{1/k}$$

Note coefficients are not recomputed, just $\delta_{\max}$

It appears very challenging to find a good controller

ODETS: Putting it all together

Guaranteed ODE defect control

Corless and Corliss (1991), Nedialkov (1999)

- Evaluate computational graph
  TaylorExpansion class
- Compute approximate solution using taylor arithemetic
  ApproximateSolution class
- Compute defect TM and bound it
  Tmodel class
- Apply stepsize control to rigorously control defect
  ODETS class

Defect controlled Predator-Prey

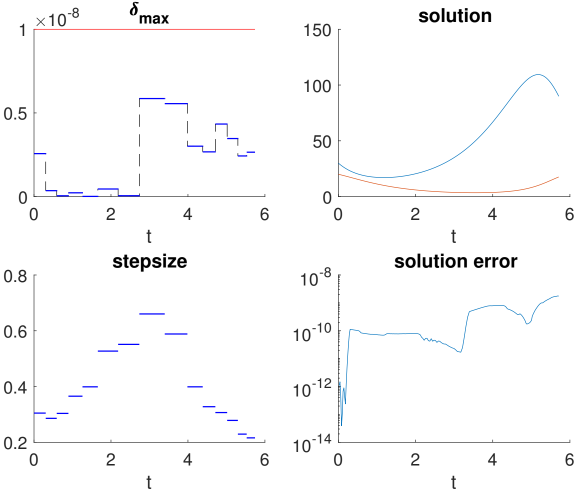


Figure: Predator-prey, order 14, tol $= 10^{-8}$

Defect controlled Predator-Prey

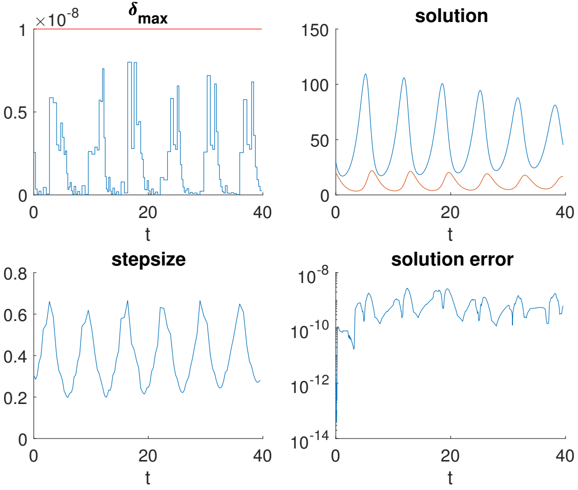


Figure: Predator-prey, order 14, tol $= 10^{-8}$

Defect controlled Lorenz

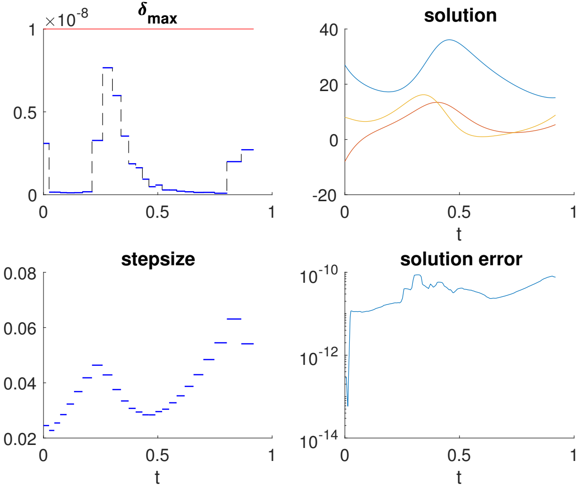


Figure: Lorenz, order 14, tol $= 10^{-8}$

Defect controlled Lorenz

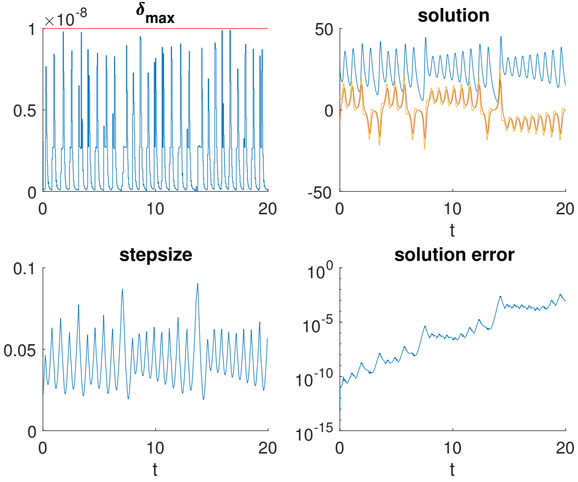


Figure: Lorenz, order 14, tol $= 10^{-8}$

Accepted/rejected steps

		Lorenz $t_{\text{end}} = 20$		pred. prey $t_{\text{end}} = 40$	
order	tol	acc	rej	acc	rej
15	10^{-6}	356	79	80	23
	10^{-8}	465	65	103	20
	10^{-10}	612	25	135	15
	10^{-12}	814	2	179	15
20	10^{-6}	266	70	62	19
	10^{-8}	325	80	76	22
	10^{-10}	399	81	92	25
	10^{-12}	508	57	114	28

Conclusions

- Corless and Corliss rigorous defect control implemented
- It appears very challenging to find a good step controller