Developing Software for Investigating Mathematical Models described by Systems of ODEs [a]

W.H. Enright

Department of Computer Science

University of Toronto

BIRS Workshop on Control and Optimization with Differential-Algebraic Constraints

October 25 - 29, 2010

[a] This research is supported by the Natural Science and Engineering Research Council of Canada.

The Focus of this Talk

Accurate and Reliable Software for <u>Investigating</u> Mathematical Models described by Systems of <u>ODEs</u>

- “Investigating” and not only “Approximating the Solution”.
- “ODEs” includes IVPs, BVPs, DDEs, DAEs and VIDEs.

Acknowledgement

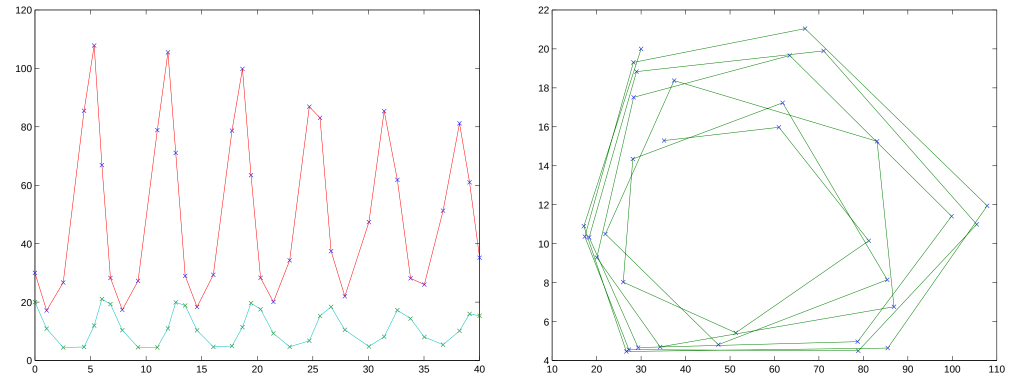
This work is part of an ongoing project and has benefited from numerous discussions and collaborations with

- Paul Muir
- Wayne Hayes
- Ken Jackson
- Silvana Ilie
- Hassan Goldani
- Hossein Zivari Piran
- Mohammad Shakourifar
- Li Yan

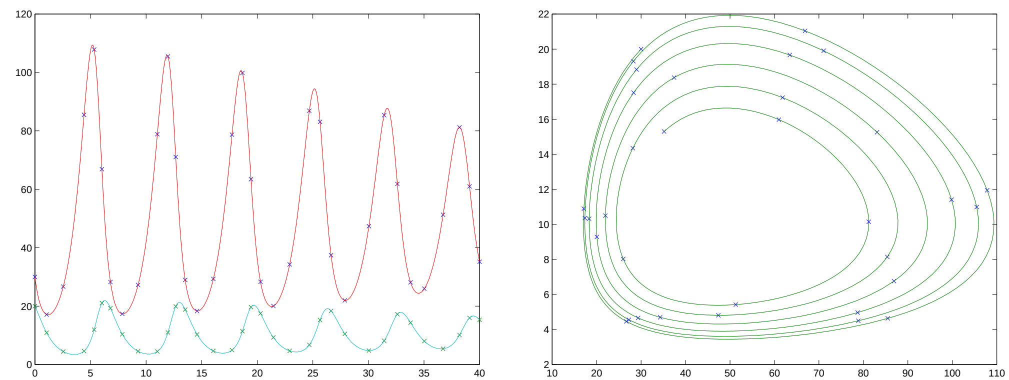
An Effective ODE Solver

Minimum Requirements:

- An Accurate Discrete Approximation is not Enough



- An Accurate Continuous Extension is Necessary



- A Reliable Error-Control and Stepsize-Selection Strategy

Outline of Talk

- Current Scientific Computing Paradigm and its implications:
  - Acceptability of an approximate solution
- Continuous RK Methods provide dense output for ODEs
- Defect Error Control for CRK Methods
- Measuring the Reliability of a CRK Method
- Classes of ODE problems that can be Investigated by CRK-based Methods (IVPs, BVPs, DDEs, DAEs, and VIDEs)
- Useful Software Tools for Investigating and quantifying Important Properties of the Mathematical Model and its Approximate Solution. (sensitivity analysis, global error estimation, parameter fitting and condition number estimation.)
- Some Numerical Examples
- Implementations and implications for DAEs

Scientific Computing Paradigm

Mathematical Modelling in a Problem Solving Environment:

- Formulate the mathematical model of the system being investigated. (The model may involve parameters.)
- Approximate the exact solution of this model relative to a specified accuracy parameter, TOL.
- Visualize the <u>approximate</u> solution.
- Verify the approximate solution's consistency with the mathematical model (may involve parameter determination).
- Verify that mathematical model is well-posed and approximate solution is stable (may involve sensitivity analysis).

Implications for ODE Solvers

What is an acceptable approximate solution?

- The approximate solution must be easy to represent, display and manipulate.
- The accuracy (or quality) of the approximate solution must be easy to measure and interpret.

What are the implications for an ODE solver?

- It should use a generic calling sequence so it is easy to adopt in a PSE.
- Solver should be easy to invoke –(only need to specify those parameters necessary to define the problem).
- A discrete solution is not enough (as it is difficult to display and its accuracy difficult to interpret).

Continuous Runge-Kutta Methods

- Consider an IVP defined by the system

$$y' = f(x, y), \;\; y(a) = y_0, \;\; \text{on} \;\; [a, b].$$

- A numerical method will introduce a partitioning $a = x_0 < x_1 < \cdots < x_N = b$ and corresponding discrete approximations $y_0, y_1 \cdots y_N$. The y_i's are usually determined sequentially.

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

$$z_i' = f(x, z_i(x)), \;\; z_i(x_{i-1}) = y_{i-1}, \;\; \text{on} \;\; [x_{i-1}, x_i].$$

CRK methods (cont)

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

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

where $h_i = x_i - x_{i-1}$ and the j^{th} stage is defined by,

$$k_j = f(x_{i-1} + h_i c_j, y_{i-1} + h_i \sum_{r=1}^{s} a_{jr} k_r).$$

A Continuous extension (CRK) is determined by introducing $(\tilde{s} - s)$ additional stages to obtain an order p approximation for any $x \in (x_{i-1}, x_i)$

$$u_i(x) = y_{i-1} + h_i \sum_{j=1}^{\tilde{s}} b_j(\frac{x - x_{i-1}}{h_i}) k_j,$$

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

CRK methods (cont)

- We will consider $O(h^p)$ extensions, satisfying:

$$u_i(x) = y_{i-1} + h_i \sum_{j=1}^{\tilde{s}} b_j(\tau) k_j = z_i(x) + O(h_i^{p+1}).$$

- The $[u_i(x)]_{i=1}^{N}$ define a piecewise polynomial $U(x)$ for $x \in [x_0, x_F]$. This is the <u>approximate solution</u> generated by the CRK method.
- $U(x) \in C^0[x_0, x_F]$ and will interpolate the underlying discrete RK values, y_i, if $b_j(1) = \omega_j$ for $j = 1, 2 \cdots s$ and $b_{s+1}(1) = b_{s+2}(1) = \cdots b_{\tilde{s}}(1) = 0$.
- Similarly a simple set of constraints on the $\frac{d}{d\,\tau}(b_j(\tau))$, including $k_{s+1} = f(x_i, y_i)$, will ensure $U'(x)$ interpolates $f(x_i, y_i)$ and therefore $U(x) \in C^1[x_0, x_F]$.

Defect Error Control for CRKs

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

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

It can be shown that,

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

$$G(\tau) = \tilde{q}_1(\tau) F_1 + \tilde{q}_2(\tau) F_2 + \cdots + \tilde{q}_k(\tau) F_k,$$

where the $\tilde{q}_j$ are polynomials in τ that depend only on the CRK formula while the F_j are constants (the elementary differentials) that depend only on the problem.

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

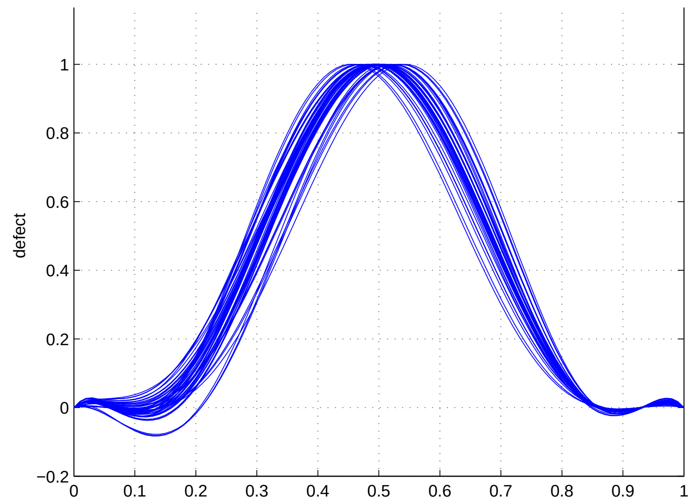
Defect Error Control (cont)

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

$$G(\tau) = \tilde{q}_1(\tau)F_1 + \tilde{q}_2(\tau)F_2 + \cdots + \tilde{q}_k(\tau)F_k.$$

- As $h_i \to 0$ the defect will then look like a linear combination of the known polynomials $\tilde{q}_j(\tau)$ over $[x_{i-1}, x_i]$.
- In the special case where $k = 1$ the shape of the defect will be the same (as $h_i \to 0$) for all problems and all steps. That is, the defect will almost always 'converge' to a multiple of $\tilde{q}_1(\tau)$, in which case the maximum should occur (as $h_i \to 0$) at $\tau = \tau^*$ where τ^* is the location of the local extremum of $\tilde{q}_1(\tau)$. In this case we will refer to the defect control strategy as **Strict Defect Control (SDC)**.

Typical Shape of SDC Defects



Plot of scaled defect vs τ (ie. $\delta(\tau)/\delta(\tau^*)$ vs τ) for each step required to solve a typical problem with SDC CRK6 and $TOL = 10^{-6}$.

Defect Control (cont)

$$p^{th} - \text{order discrete RK} : y_i = y_{i-1} + h_i \sum_{j=1}^{s} \omega_j k_j,$$

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

Formula	p	s	$\tilde{s}$
CRK4	4	4	8
CRK5	5	6	12
CRK6	6	7	15
CRK7	7	9	20
CRK8	8	13	27

Table 1: Cost per step of some SDC-CRK formulas
(Note that $\tilde{s} \approx 2\ s$.)

Strict Defect Control

SDC CRKs are not unique (for a given discrete RK formula).
Each SDC-CRK satisfies,

$$\delta(x) = \tilde{q}_1(\tau)F_1 h_i^p \; + \; (\hat{q}_1(\tau)\hat{F}_1 + \hat{q}_2(\tau)\hat{F}_2 + \cdots\cdots \hat{q}_{\hat{k}}(\tau)\hat{F}_{\hat{k}})h_i^{p+1} + O(h_i^{p+2})$$

Potential Difficulties:

- $\tilde{q}_1(\tau)$ may have a large maximum ($\tilde{q}_1(0) = \tilde{q}_1(1) = 0$ and its 'average' value must be one).
- The $\hat{q}_j(\tau)$ may be large in magnitude relative to $\tilde{q}_1(\tau)$ (and therefore h_i would have to be small before the estimate is justified). (That is, before $|h_i\hat{q}_j(\tau)| << |\tilde{q}_1(\tau)|$.)
- $|F_1|$ may be zero (or very small) on isolated steps.

For each p we have identified a particular SDC-CRK that minimizes these difficulties.

Optimal SDC CRK6

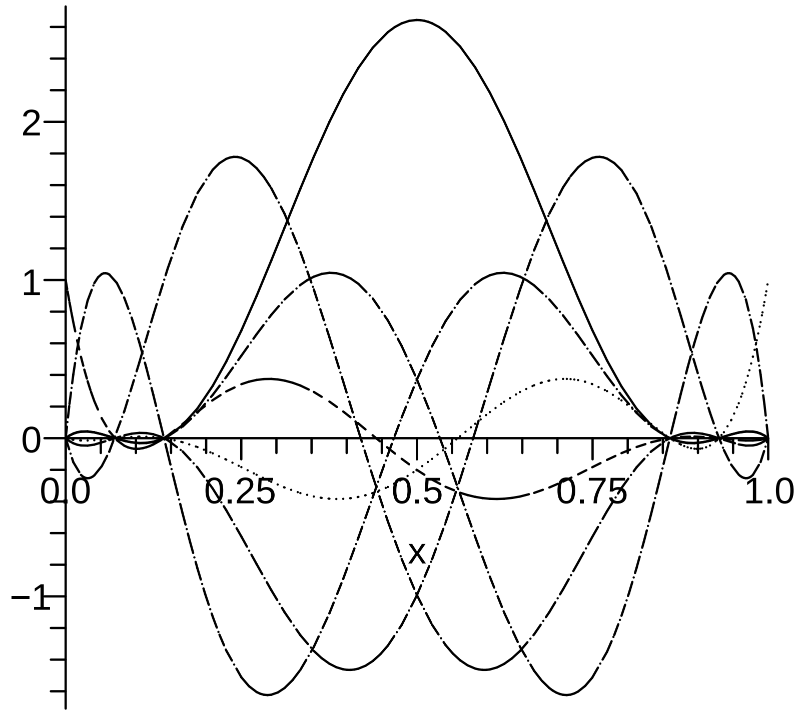


Figure 1: Plots of $\tilde{q}_1$ and $\hat{q}_2 \cdots \hat{q}_7$ for SDC CRK6. $\tilde{q}_1$ is represented by the solid line and has the highest magnitude.

Optimal SDC CRK8

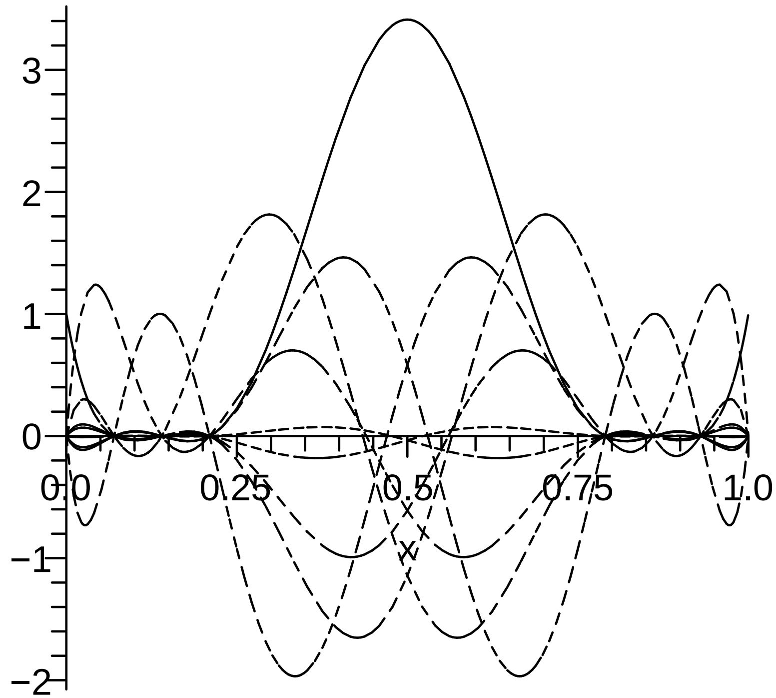


Figure 2: Plots of $\tilde{q}_1$ and $\hat{q}_2 \cdots \hat{q}_9$ for SDC CRK8. $\tilde{q}_1$ is represented by the solid line and has the highest magnitude.

Quantifying Reliability

Consider two measures of reliability of a CRK method:

- How well does the **Method** control the maximum magnitude of the defect? We can measure the ratio of the max defect to TOL on each step and the fraction of steps where this ratio is greater than 1 ?
- How well does the **Estimate** of the max defect reflect its true value? We can measure both the ratio of the true maximum defect (on a successful step) to its estimated value and the fraction of attempted steps where the estimated maximum is within <u>one percent</u> of the true maximum.

Reliability of SDC Methods

- We have implemented SDC versions of CRK5, CRK6 and CRK8.
- We have run these three methods on the 25 IVP test problems of DETEST (all non-stiff), at 9 tolerances from 10^{-1} to 10^{-9}.
- We report summaries only. We report two measures of cost: NSTP and NFCN, two measures of the reliability of the method : DMAX and Frac-D (max defect and fraction of steps where this exceeded TOL), and two measures of the reliability of the estimate: R-Max and Frac-G (maximum ratio of the true maximum defect to the estimate and the fraction of steps where this was bounded by 1.01).

Numerical Results for SDC CRKs

Results on the 25 DETEST Problems for SDC5, SDC6 and SDC8

TOL	CRK	NSTP	NFCN	DMAX	Frac-D	R-Max	Frac-G
10^{-2}	SDC5	625	11709	0.97	.000	1.05	.67
	SDC6	549	12300	1.00	.000	1.43	.71
	SDC8	333	12793	1.01	.003	1.65	.35
10^{-4}	SDC5	1065	19033	1.01	.001	1.12	.78
	SDC6	931	19819	1.00	.001	1.08	.87
	SDC8	465	17319	1.05	.004	1.47	.45
10^{-6}	SDC5	2099	35703	1.01	.002	1.08	.86
	SDC6	1748	35073	1.01	.001	1.08	.96
	SDC8	712	26253	1.02	.001	1.34	.59
10^{-8}	SDC5	4566	66937	1.01	.001	1.07	.95
	SDC6	3547	65148	1.01	.001	1.07	.98
	SDC8	1081	38251	1.12	.007	2.60	.62

SDC-CRK based methods developed for

- IVPs:

$$y' = f(x, y), \;\; y(a) = y_0, \;\; x \in [a, b],$$

where $y, y_0 \in \Re^n$ and $f : \Re \times \Re^n \to \Re^n$.

- BVPs:

$$y' = f(x, y), \;\; x \in [a, b],$$

with

$$g(y(a), y(b)) = 0, \;\; g : \Re^n \times \Re^n \to \Re^n.$$

- DAEs (with low index):

$$F(x, y, y') = 0, \;\; y(x) \in \Re^n, \;\; y(a) = y_0,$$

for $x \in [a, \; b]$. With $\frac{\partial F}{\partial y'}$ singular but of constant rank in some neighborhood of $y(x)$.

Classes of ODEs (cont)

- DDEs (both retarded and neutral problems):

$$\begin{aligned} y' &= f(x, y(x), y(x-\sigma_1)\cdots y(x-\sigma_k), y'(x-\sigma_{k+1}), \\ &\quad \cdots y'(x-\sigma_{k+\ell})), \text{ for } x \in [a,b], \end{aligned}$$

  where $y(x) \in \Re^n$ and,

$$y(x) = \phi(x), \;\; y'(x) = \phi'(x), \;\; \text{for } \; x \le a,$$

$$\sigma_i \equiv \sigma_i(x, y(x)) \ge 0 \; \text{ for } \; i = 1, 2 \cdots k + \ell.$$

- VIDEs (with a time dependent delay):

(1) $$y'(x) = f(x, y(x)) + \int_{x-\sigma(x)}^{x} K(x, s, y(s), y'(s))ds,$$

  for $x \in [a,b]$, $f : \Re \times \Re^n \rightarrow \Re^n$ and $K : \Re \times \Re \times \Re^n \times \Re^n \rightarrow \Re^n$ and $y(x) = \phi(x)$ for $x \le a$.

Effective Tools for Investigating ODEs

For each Class of ODEs we are implementing effective tools for:

- Estimating the Global Error
- Detecting, Locating and Coping with Discontinuous Problems
- Estimating the Conditioning of the Problem
- Sensitivity analysis of the Problem (eg., $\frac{\partial y_i(x)}{\partial p_j}$)
- Solving Problems which depend on parameters (parameter continuation and/or parameter fitting – an inverse problem)

Implementation for DDEs

In his PhD thesis, Hossein Zivaripiran [University of Toronto, 2009] began the implementation of a PSE (DDEM) for the investigation of DDEs. (see `http://www.cs.utoronto.ca/~hzp`).

DDEM includes modules for:

1. Accurate location of all significant discontinuities.
2. Reliable simulation and visualization of a problem.
3. Efficient solution of the discrete approximations when delay is small or the underlying discrete RK formula is implicit.
4. Reliable approximation of first order sensitivities. (No other method we know of can do this.)
5. Parameter fitting from noisy data (using a “nonsmooth Newton” approach to achieve superlinear convergence.

Example: Parameter Fitting for DDEs

Consider the Kermack-McKendrick model of an infectious disease with periodic outbreaks:

$$
\begin{aligned}
y_1' &= -y_1(x)y_2(x-\sigma) + y_2(x-\rho), \\
y_2' &= y_1(x)y_2(x-\sigma) - y_2(x), \\
y_3' &= y_2(x) - y_2(x-\rho),
\end{aligned}
$$

with $x \in [0, 55]$, and $y_1(x) = 5.0, y_2(x) = 0.1, y_3(x) = 1.0,$ for $x \leq 0$.

The exact solution to this problem is unknown. Each delay introduces a C^2 discontinuity in the objective function whenever it is evaluated at a multiple of σ or ρ. We generate the data to be "fit" by computing an accurate solution with parameter values, $\sigma^* = 1$ and $\rho^* = 10$. We perturb these values by up to a 10% random perturbation to determine our initial guess for each parameter and we use 10 equally spaced sample points to define the prescribed data to be fit.

Parameter Fitting Results

Newton Jac	FCN	ITER	TIME	OBJ
DivDiff	783092	393.2	54.9	$7.4 \cdot 10^{-13}$
SenJac	37344	13.8	2.3	$1.3 \cdot 10^{-9}$
ConSenJac	5293	2.1	0.31	$1.3 \cdot 10^{-9}$

We report the total number of derivative evaluations FCN, The number of Newton iterations ITER, and the CPU time TIME (each averaged over 10 runs) for solving this problem with standard divided differences used to approximate the Newton Jacobian (DivDiff); with the Newton Jacobian approximated using an accurate Sensitivity Analysis (SenJac); and with the Newton Jacobian approximated using a constrained Newton step (ConSenJac). We also report the value of the objective function OBJ at the computed optimum point.

Implementation for DAEs

Consider the case where the algebraic constraints can be explicitly identified and the problem decoupled, $y(x) = [y_1(x), y_2(x)]^T$, and written in the semi-explicit form,

$$
\begin{aligned}
y_1'(x) &= f(x, y_1(x), y_2(x)), \\
0 &= g(x, y_1(x), y_2(x)).
\end{aligned}
$$

When one considers defect-based error control for DAEs, two questions arise:

1. How does one define a suitable continuous extension, $[u_i(x), v_i(x)]^T \approx [y_1(x), y_2(x)]^T$ for $x \in [x_{i-1}, x_i]$.
2. What measure of the size of the associated defect should be monitored and controlled?

Defect Error Control for DAEs

Let $u(x)$ and $v(x)$ be the vector of piecewise polynomials associated with the $u_i(x), v_i(x)$. The approximate solution, defined by $[u(x), v(x)]^T$ satisfies,

$$\begin{aligned} \delta_1(x) &= u'(x) - f(x, u(x), v(x)), \\ \delta_2(x) &= g(x, u(x)v(x)). \end{aligned}$$

The global errors in the approximate solution, $\|y_1(x) - u(x)\|$ and $\|y_2(x) - v(x)\|$, will be bounded by a suitable multiple of TOL, provided $\|\delta_1(x)\|, \|\delta_2(x)\|, \|\delta_2'(x)\|$ are each suitably bounded. The bounds that each must satisfy are computable and a effective DAE solver will adjust h_i in an attempt to satisfy them on each step.

The Next Steps for DAEs

To develop a similar PSE for DAEs will require the development of modules for the following key components (of this PSE):

- Accurate location of all significant discontinuities.
- Efficient solution of the discrete approximations corresponding to the underlying implicit discrete RK formula.
- Reliable simulation and visualization of a problem.
- Reliable approximation of first order sensitivities.
- Parameter fitting from noisy data (using a “nonsmooth Newton” approach to achieve superlinear convergence.