Sensitivity Capabilities in SUNDIALS

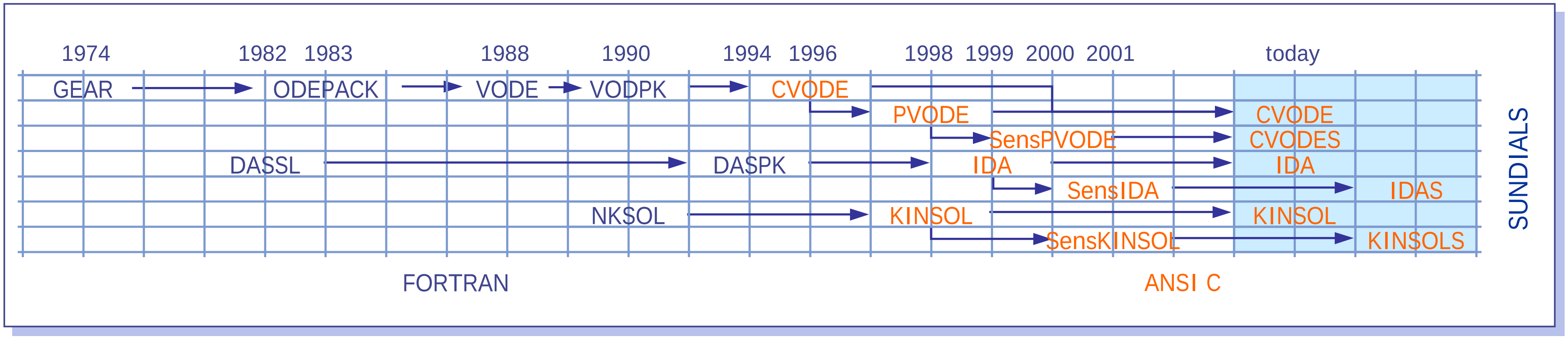
Radu Şerban

Center for Applied Scientific Computing
Lawrence Livermore National Laboratory

7th U.S. National Congress on Computational Mechanics
28 July 2003

Background

- LLNL has a long history of R&D in ODE and DAE methods and software, and closely related areas, with emphasis on applications to PDEs.



- Focus on recent years:
  - Parallel solution of large-scale problems
  - Sensitivity analysis

Background (cont.)

- Starting in 1993, the push to solve large systems in parallel motivated work to write or rewrite solvers in C:
  - CVODE: a C rewrite of VODE/VODPK [Cohen and Hindmarsh, 1994]
  - PVODE: parallel extension of CVODE [Byrne and Hindmarsh, 1998]
  - KINSOL: C rewrite of NKSOL [Taylor and Hindmarsh, 1998]
  - IDA: C rewrite of DASPK [Hindmarsh and Taylor, 1999]
- Preliminary sensitivity variants:
  - SensPVODE, SensIDA, SensKINSOL [Brown, Hindmarsh, Lee, 2000-2001]
- After the reorganization into SUNDIALS, there is one ODE solver, CVODE, in two versions – serial and parallel (through the NVECTOR module)
- New sensitivity capable solvers in SUNDIALS:
  - CVODES [Hindmarsh and Serban, 2002]
  - IDAS [Serban, 2003] – in development

Structure of SUNDIALS

User main routine
User problem-defining function
User preconditioner function

Solvers

- $x' = f(t,x)$, $x(t_0) = x_0$ — `CVODE`
- $F(t,x,x') = 0$, $x(t_0) = x_0$ — `IDA`
- $F(x) = 0$ — `KINSOL`

CVODE ODE Integrator

IDA DAE Integrator

KINSOL Nonlinear Solver

Band Linear Solver

Dense Linear Solver

Preconditioned GMRES Linear Solver

General Preconditioner Modules

Vector Kernels

The SUNDIALS Basic Solvers

- CVODE
  - Variable-order, variable-step BDF (stiff) or implicit Adams (nonstiff)
  - Nonlinear systems solved by Newton or functional iteration
  - Linear systems solved by direct (dense or band) or SPGMR solvers
- IDA
  - Variable-order, variable-step BDF
  - Nonlinear system solved by Newton
  - Linear systems solved by direct or SPGMR solvers
- KINSOL
  - Inexact Newton method
  - Krylov solver: SPGMR (Scaled Preconditioned GMRES)
- Preconditioners
  - Band preconditioner (CVODE)
  - Band-Block-Diagonal preconditioner (CVODE, IDA, KINSOL)
  - User-defined (setup and solve user routines)

Sensitivity Analysis

- Sensitivity Analysis (SA) is the study of how the variation in the output of a model (numerical or otherwise) can be apportioned, qualitatively or quantitatively, to different sources of variation.
- Applications:
  - Model evaluation (most and/or least influential parameters)
  - Model reduction
  - Data assimilation
  - Uncertainty quantification
  - Optimization (parameter estimation, design optimization, optimal control, …)
- Approaches:
  - Forward sensitivity analysis
  - Adjoint sensitivity analysis

Sensitivity Analysis Approaches

Parameter dependent system

$$\begin{cases} F(x, \dot{x}, t, p) = 0 \\ x(0) = x_0(p) \end{cases}$$

Forward sensitivity

$$\begin{cases} F_{\dot{x}} \dot{s}_i + F_x s_i + F_{p_i} = 0 \\ s_i(0) = x_{0\,p_i} \end{cases}, \quad i = 1, \ldots, N_p$$

$$g(t, x, p)$$

$$\frac{\mathrm{d}g}{\mathrm{d}p} = g_x s + g_p$$

Computational cost: $(1+N_p)N_x$

increases with N_p

Adjoint sensitivity

$$\begin{cases} (\lambda^* F_{\dot{x}})' - \lambda^* F_x = -g_x \\ \lambda^* F_{\dot{x}} x_p = \ldots \quad \text{at } t = T \end{cases}$$

$$G(x, p) = \int_0^T g(t, x, p)\,dt$$

$$\frac{\mathrm{d}G}{\mathrm{d}p} = \int_0^T (g_p - \lambda^* F_p)\,dt - (\lambda^* F_{\dot{x}} x_p)\Big|_0^T$$

Computational cost: $(1+N_G)N_x$

increases with N_G

Forward Sensitivity Analysis

- For a parameter dependent system

$$\begin{cases} F(x, \dot{x}, t, p) = 0 \\ x(0) = x_0(p) \end{cases}$$

find $s_i = dx/dp_i$ by simultaneously solving the original system with the N_p sensitivity systems obtained by differentiating the original system with respect to each parameter in turn:

$$\begin{cases} F_{\dot{x}} \dot{s}_i + F_x s_i + F_{p_i} = 0 \\ s_i(0) = x_{0\,p_i} \end{cases} \quad , \quad i = 1, \ldots, N_p$$

- Gradient of a derived function

$$g(t, x, p) \quad \rightarrow \frac{\mathrm{d}g}{\mathrm{d}p} = g_x s + g_p$$

- Can obtain gradients with respect to p for any derived function
- Computational cost - $(1+N_p)N_x$ - increases with N_p

Adjoint Sensitivity Analysis

- index-0 and index-1 DAE

$$(\lambda^* F_{\dot{x}})\big|_{t=T} = 0 \qquad \frac{dG}{dp} = \int_0^T \left(g_p - \lambda^* F_p\right)dt + (\lambda^* F_{\dot{x}})\big|_{t=0} x_{0p}$$

$$\begin{cases} (\lambda^* F_{\dot{x}})' - \lambda^* F_x = -g_x \\ \lambda^* F_{\dot{x}} x_p = \dots \quad \text{at } t = T \end{cases}$$

$$G(x,p) = \int_0^T g(t,x,p)\,dt$$

$$\frac{\mathrm{d}G}{\mathrm{d}p} = \int_0^T (g_p - \lambda^* F_p)\,dt - (\lambda^* F_{\dot{x}} x_p)\big|_0^T$$

- Hessenberg index-2 DAE

$$\begin{cases} \dot{x}^d = f^d(x^d, x^a, p) \\ 0 = f^a(x^d, p) \end{cases} \rightarrow \begin{cases} \dot{\lambda}^d + A^* \lambda^d + C^* \lambda^a = -g^*_{x^d} \\ B^* \lambda^d = -g^*_{x^a} \end{cases}$$

$$A = \frac{\partial f^d}{\partial x^d},\ B = \frac{\partial f^d}{\partial x^a},\ C = \frac{\partial f^a}{\partial x^d},\ \exists (CB)^{-1}$$

search for final conditions of the form $\lambda^{d*}(T) = \xi^* C\big|_{t=T}$

At $t = T$:

$$\lambda^{d*} B = -g_{x^a} \Rightarrow \xi^* CB = -g_{x^a} \Rightarrow \xi^* = -g_{x^a}(CB)^{-1}$$

$$f^a(x^d, p) = 0 \Rightarrow Cx^d_p = -f^a_p \Rightarrow \lambda^{d*} x^d_p = -\xi^* f^a_p$$

$$\lambda^{d*}(T) = \left(-g_{x^a}(CB)^{-1} C\right)\big|_{t=T} \qquad \frac{dG}{dp} = \int_0^T \left(g_p + \lambda^{d*} f^d_p + \lambda^{a*} f^a_p\right)dt + \left(\lambda^{d*}\right)\big|_{t=0} x^d_{0p} - \left(g_{x^a}(CB)^{-1} f^a_p\right)\big|_{t=T}$$

Adjoint Sensitivity - Sensitivity of g(x,T,p)

- Sensitivity of objective function

$$\left.\frac{dg}{dp}\right|_{t=T} = \frac{d}{dT}\frac{dG}{dp} = \left.\left(g_p - \lambda^* F_p\right)\right|_{t=T} + \int_0^T \mu^* F_p \, dt - \left.\left(\mu^* F_{\dot{x}} x_p\right)\right|_{t=0} - \left.\left(\frac{d(\lambda^* F_{\dot{x}} x_p)}{dT}\right)\right|_{t=T}$$

- Adjoint system

$$\begin{cases} (\mu^* F_{\dot{x}})' - \mu^* F_x = 0 \\ \mu^* = \mathrm{K} \quad \text{at} \quad t = T \end{cases}$$

Implicit ODE

$$F(x, \dot{x}) = 0$$

$$A = \frac{\partial F}{\partial \dot{x}}, \; B = \frac{\partial F}{\partial x}, \; \exists A^{-1}$$

$$\begin{cases} (A^* \mu)' - B^* \mu = 0 \\ A^* \mu = g_x^* \quad @T \end{cases}$$

Semi-explicit index-1 DAE

$$\begin{cases} \dot{x}^d = f^d(x^d, x^a) \\ 0 = f^a(x^d, x^a) \end{cases}$$

$$A = \frac{\partial f^d}{\partial x^d}, \; B = \frac{\partial f^d}{\partial x^a}, \; C = \frac{\partial f^a}{\partial x^d}, \; D = \frac{\partial f^a}{\partial x^a}, \; \exists D^{-1}$$

$$\begin{cases} \dot{\mu}^d = -A^* \mu^d - C^* \mu^a \\ 0 = B^* \mu^d + D^* \mu^a \\ \mu^d = g^*_{x^d} - C^* (D^*)^{-1} g^*_{x^a} \quad @T \end{cases}$$

Hessenberg index-2 DAE

$$\begin{cases} \dot{x}^d = f^d(x^d, x^a) \\ 0 = f^a(x^d) \end{cases}$$

$$A = \frac{\partial f^d}{\partial x^d}, \; B = \frac{\partial f^d}{\partial x^a}, \; C = \frac{\partial f^a}{\partial x^d}, \; \exists (CB)^{-1}$$

$$\begin{cases} \dot{\mu}^d = -A^* \mu^d - C^* \mu^a \\ 0 = B^* \mu^d \\ \mu^d = P^* \left[g^*_{x^d} - A^* C^* (B^* C^*)^{-1} g^*_{x^a} - \frac{dC^*}{dt} (B^* C^*)^{-1} g^*_{x^a} \right] \quad @T \end{cases}$$

$$P = I - B(CB)^{-1} C$$

Stability of the adjoint system

- Explicit ODE: proof using Green's function;

$$\dot{x} = Ax \longrightarrow \dot{\mu} = -A^* \mu$$

- Semi-explicit index-1 and Hessenberg index-2 DAE: the EUODE of the adjoint system is the adjoint of the EUODE of the original system;

Example: Semi-explicit index-1 DAE

$$\begin{cases} \dot{x}^d = Ax^d + Bx^a \\ 0 = Cx^d + Dx^a \end{cases} \longrightarrow \begin{cases} \dot{\mu}^d = -A^* \mu^d - C^* \mu^a \\ 0 = B^* \mu^d + D^* \mu^a \end{cases}$$

$$\downarrow \qquad\qquad\qquad\qquad \downarrow$$

$$\dot{x}^d = Ax^d - B(D)^{-1} Cx^d \longrightarrow \dot{\mu}^d = -A^* \mu^d + C^* (D^*)^{-1} B^* \mu^d$$

Stability of the adjoint system (contd.)

- Implicit ODE and index-1 DAE: use bounded transformation
- Lemma (Campbell, Bichols, Terrel)
  Given the time dependent linear DAE system

$$A(t)\dot{x} + B(t)x = f(t)$$

  and nonsingular time dependent differentiable matrices P(t) multiplying the equations of the DAE and Q(t) transforming the variables, the adjoint system of the transformed DAE is the transformed system of the adjoint DAE.
- Theorem
  For general index-0 and index-1 DAE systems, if the original DAE system is stable then the augmented DAE system is stable.

$$\begin{cases} \dot{\bar{\lambda}} - F_x^* \lambda = -g_x^* \\ \bar{\lambda} - F_{\dot{x}}^* \lambda = 0 \end{cases}$$

Forward Sensitivity Analysis in SUNDIALS

User main r
Specificatio
Activation
User proble
User precon

```
nvSpec = NV_SpecInit_Parallel(…);
y0 = N_VNew(nvSpec);
cvmem = CVodeCreate(BDF,NEWTON);
flag = CVodeSet*(…);
flag = CVodeMalloc(cvmem,rhs,t0,y0, …);
flag = CVSpgmr(cvmem,…);
y0S = N_VNewS(Ns,nvSpec);
flag = CVodeSetSens*(…);
flag = CVodeSensMalloc(cvmem,y0S,…);
for(tout = …) {
      flag = CVode(…,y,…);
      flag = CVodeGetSens(…,yS,…);
}
NV_SpecFree_Parallel(…);
CVodeFree(cvmem);
```

red)
, CS
les

Band Linear Solver

Solver

Linear Solver

Modules

or Kernels

Forward Sensitivity Analysis - Methods

For ODE/DAE implicit integrators

- Staggered Direct Method
  On each time step, converge Newton iteration for state variables, then solve linear sensitivity system
  - Requires formation and storage of Jacobian matrices
  - Not matrix-free
  - Errors in finite-difference Jacobians lead to errors in sensitivities
- Simultaneous Corrector Method ✓
  On each time step, solve the nonlinear system simultaneously for solution and sensitivity variables
  - Block-diagonal approximation of the combined system Jacobian
  - Requires formation of sensitivity R.H.S. at every iteration
- Staggered Corrector Method ✓
  On each time step, converge Newton for state variables, then iterate to solve sensitivity system
  - With SPGMR, sensitivity systems solved (theoretically) in 1 iteration

Adjoint Sensitivity Analysis in SUNDIALS

User main routine
Activation of sensitivity computation
User problem-defining function
User reverse function
User preconditioner function
User reverse preconditioner function

Implementation
- check point approach; total cost is 2 forward solutions + 1 backward solution
- integrate any system backwards in time
- may require modifications to some user-defined vector kernels

CVODES
ODE Integrator

IDAS
DAE Integrator

KINSOLS
Nonlinear Solver

Band Linear Solver

Dense Linear Solver

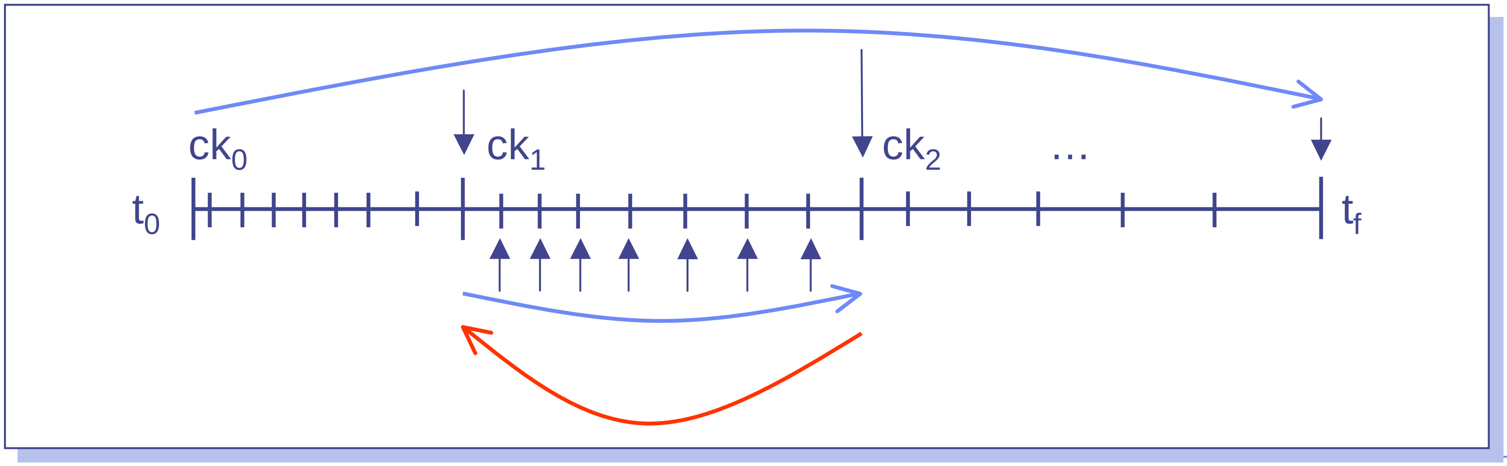
Preconditioned GMRES Linear Solver

General Preconditioner Modules

(Modified) Vector Kernels

Adjoint Sensitivity – Implementation

- Solution of the forward problem is needed in the backward integration phase → need predictable and compact storage of solution values for the solution of the adjoint system
- Checkpointing:
  - Cubic Hermite interpolation
  - Simulations are reproducible from each checkpoint
  - Force Jacobian evaluation at checkpoints to avoid storing it
  - Store solution and first derivative at all intermediate steps between two consecutive checkpoints
  - Computational cost: 2 forward and 1 backward integrations



Applications

- Parallel CVODE is being used in a 3D tokamak turbulence model in LLNL's Magnetic Fusion Energy Division. A typical run has 7 unknowns on a 64x64x40 mesh, with up to 60 processors
- KINSOL with a HYPRE multigrid preconditioner is being applied within CASC to solve a nonlinear Richards equation for pressure in porous media flows. Fully scalable performance was obtained on up to 225 processors on ASCI Blue.
- CVODE, KINSOL, IDA, with MG preconditioner, are being used to solve 3D neutral particle transport problems in CASC. Scalable performance obtained on up to 5800 processors on ASCI Red.
- SensPVODE, SensKINSOL, SensIDA have been used to determine solution sensitivities in neutral particle transport applications.
- IDA and SensIDA are being used in a cloud and aerosol microphysics model at LLNL to study cloud formation processes.
- CVODES is used for sensitivity analysis of chemically reacting flows (SciDAC collaboration with Sandia Livermore)
- CVODES is used for sensitivity analysis of radiation transport (diffusion approximation)

Current and Future Work

- Software development
  - IDAS (forward and adjoint sensitivity variant of IDA)
  - Automatic generation of sensitivity systems
    - Complex-step tools for forward sensitivity and/or Jacobian data
    - Incorporation of AD tools as they become available (forward/reverse)
  - Solvers as CCA components
    - Classic ccaffeine components for CVODE and CVODES exist
  - BABEL-ize SUNDIALS solvers
- Adjoint sensitivity for parameter identification
  - POD-based reduced model to replace checkpointing
  - Treatment of discontinuous adjoint variables (observations at discrete times)
- Sensitivity-based error analysis
  - Error estimates for reduced models
  - Global error control for ODE/DAE systems using adjoint sensitivities
- Multiple right hand side linear solvers
  - Efficiency improvements in forward sensitivity analysis

Availability

- Open source BSD license
  www.llnl.gov/CASC/sundials

- Publications
  www.llnl.gov/CASC/nsde

- The SUNDIALS Team

  Peter Brown

  Keith Grant

  Alan Hindmarsh

  Steven Lee

  Radu Serban

  Dan Shumaker

  Carol Woodward

- Past contributors

  Scott Cohen and Allan Taylor

UCRL-PRES-

This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore National Laboratory under Contract No. W-7405-Eng-48.