SAND2006-7248C

Bordered Matrix Methods for Large-Scale Stability Analysis using LOCA

Eric Phipps and Andy Salinger

Sandia National Laboratories

Trilinos User's Group Meeting

November 8, 2006

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

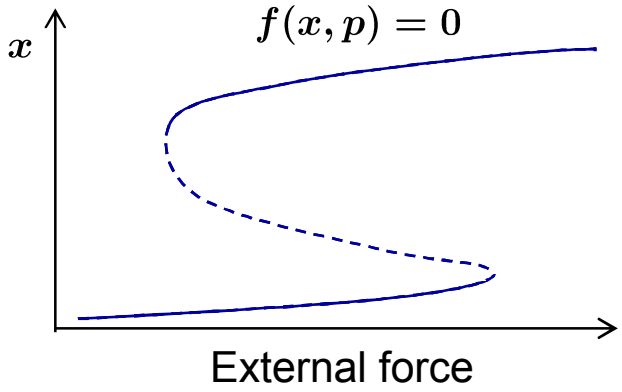
LOCA: Library of Continuation Algorithms

Application code provides:

- Nonlinear steady-state residual and Jacobian fill:

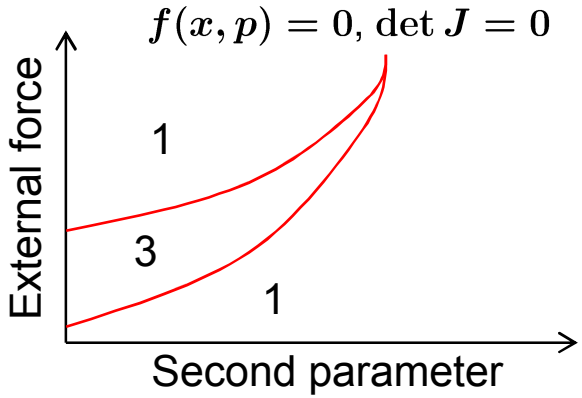
$$f(x,p) = \text{internal} - \text{external force}, \quad J = \frac{\partial f}{\partial x}$$

- Newton-like linear solves: $J\Delta x = -f$



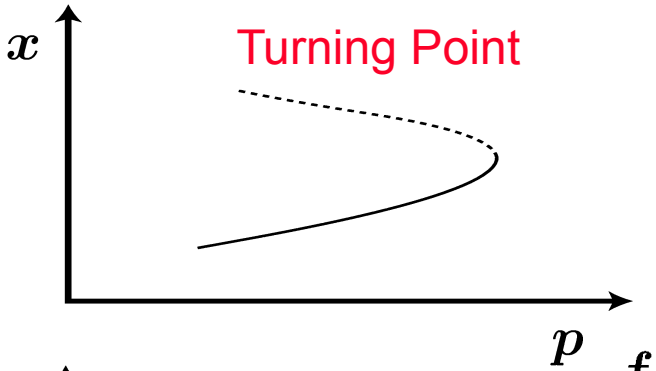
LOCA provides:

- Parameter Continuation: Tracks a family of steady state solutions with parameter
- Linear Stability Analysis: Calculates leading eigenvalues via Anasazi (Thornquist, Lehoucq)
- Bifurcation Tracking: Locates neutral stability point (x,p) and tracks as a function of a second parameter



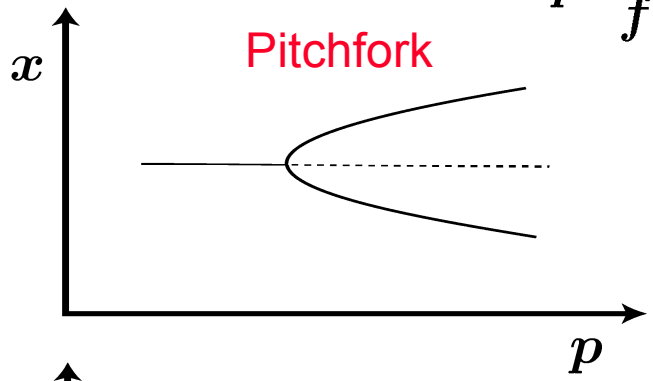
Sandia National Laboratories

Codimension 1 Bifurcations



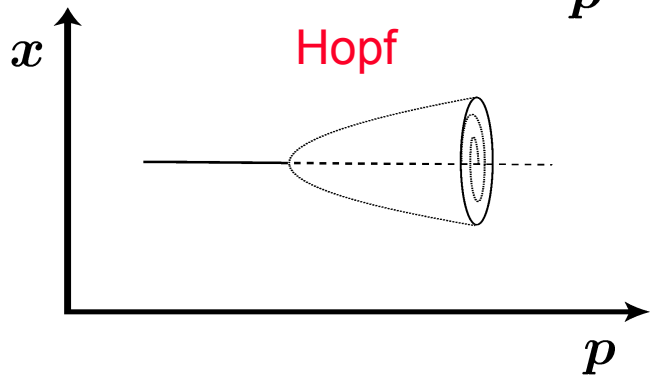
$$f(x,p) = 0$$
$$Jv = 0$$
$$\phi^T v = 1$$

- Combustion
- Buckling of an Arch



$$f(x,p) + s\psi = 0$$
$$Jv = 0$$
$$\langle x, \psi \rangle = 0$$
$$\phi^T v = 1$$

- Buckling of a Beam
- Pattern formation
- Cell differentiation (morphogenesis)



$$f(x,p) = 0$$
$$Jy - \omega M z = 0$$
$$Jz + \omega M y = 0$$
$$\phi^T y = 1$$
$$\phi^T z = 0$$

- Vortex Shedding
- Predator-Prey models
- Flutter
- El Niño

Sandia National Laboratories

New Features for Trilinos 7

- Conversion to “new” LOCA framework complete, old framework removed
  - Strategies/factories
  - Multi-vector support
  - `LOCA::NewStepper -> LOCA::Stepper`
- Epetra support for Hopf bifurcations
  - Leverages `EpetraExt::Block{Graph, CrsMatrix, MultiVector}`
- Minimally augmented turning point formulation
  - Faster, more robust turning point method
  - Uses Householder bordered solver method

Bordered Systems of Equations

Solving bordered systems of equations is a ubiquitous computation:

$$\begin{array}{c} \begin{matrix} & n & m \end{matrix} \\ \begin{matrix} n \\ m \end{matrix}\begin{bmatrix} J & A \\ B^T & C \end{bmatrix} \end{array} \begin{bmatrix} X \\ Y \end{bmatrix}\begin{matrix} n \\ m \end{matrix} = \begin{bmatrix} F \\ G \end{bmatrix}\begin{matrix} n \\ m \end{matrix}$$

$$n \gg m, l$$

Bordering Algorithm

$$JX_1 = F$$
$$JX_2 = A$$
$$(C - B^T X_2)Y = G - B^T X_1$$
$$X = X_1 - X_2 Y$$

Only requires solves of J but

- Requires m+l linear solves
- Has difficulty when J is nearly singular

Pseudo-Arclength Continuation

$$f(x,p) = 0$$
$$V_x^T(x - x_i) + V_p(p - p_i) = \Delta s$$

Constraint Following

$$f(x,p) = 0$$
$$g(x,p) = 0$$

Turning Point Identification

$$f(x,p) = 0$$
$$\sigma(x,p) = 0$$

Sandia National Laboratories

Solving Bordered Systems via QR

Extension of Householder pseudo-arclength technique by Homer Walker[1]

Rearranged Bordered System

$$\begin{bmatrix} C & B^T \\ A & J \end{bmatrix} \begin{bmatrix} Y \\ X \end{bmatrix} = \begin{bmatrix} G \\ F \end{bmatrix}$$

QR Factorization

$$\begin{bmatrix} C^T \\ B \end{bmatrix} = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$$

Compact WY Representation[2]

$$Q = I + WTW^T \in \mathbf{R}^{(n+m)\times(n+m)}$$

$$\begin{bmatrix} Z_Y \\ Z_X \end{bmatrix} = Q^T \begin{bmatrix} Y \\ X \end{bmatrix} \Rightarrow \begin{bmatrix} R^T & 0 \\ [A & J]Q \end{bmatrix} \begin{bmatrix} Z_Y \\ Z_X \end{bmatrix} = \begin{bmatrix} G \\ F \end{bmatrix} \Rightarrow \begin{cases} Z_Y = R^{-T}G \\ PZ_X = \tilde{F} \end{cases}$$

where $\begin{cases} PZ_X \equiv [A \;\; J]Q \begin{bmatrix} 0 \\ Z_X \end{bmatrix} \\ \tilde{F} = F - [A \;\; J]Q \begin{bmatrix} Z_Y \\ 0 \end{bmatrix} \end{cases}$. Write $W = \begin{bmatrix} W_1 \\ W_2 \end{bmatrix}$ then

$$P = J + (AW_1 + JW_2)(W_2T)^T = J + UV^T.$$

P is nxn, nonsingular, rank m update to J

[1]H.F Walker, SIAM J. Sci. Comput., 1999

[2]R. Schreiber, SIAM J. Stat. Comput., 1989

Sandia National Laboratories

Implementing Householder Bordered Solver Method in LOCA

- All steps in bordered solver algorithm can be done generically except solve

$$PZ_X = \tilde{F}, \quad P = J + UV^T$$

- Solving P is linear algebra dependent

- Two Epetra implementations of P
  - J is an Epetra_Operator, P is an Epetra_Operator
  - J is an Epetra_RowMatrix, P is an Epetra_RowMatrix
    - Include UV^T terms for nonzeros of J for preconditioning
    - Appropriate for Epetra_RowMatrix based preconditioners

- If J is an Epetra_CrsMatrix, also can overwrite J with $J + UV^T$ for Epetra_CrsMatrix preconditioners

Calling the Householder Bordered Solver Method in LOCA

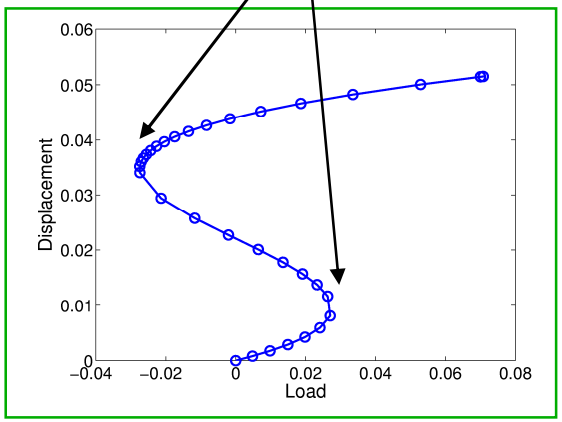
- Methods for solving bordered systems encapsulated in
  - `LOCA::BorderedSolver::AbstractStrategy`
  - `LOCA::BorderedSolver::Factory`
  - Selected by parameter lists
- Householder requires LOCA Epetra Factory

```
// Create parameter list
Teuchos::RefCountPtr<Teuchos::ParameterList> paramList = Teuchos::rcp(new Teuchos::ParameterList);
Teuchos::ParameterList& locaParamsList = paramList->sublist("LOCA");
Teuchos::ParameterList& locaStepperList = locaParamsList.sublist("Stepper");
locaStepperList.set("Continuation Method", "Arc Length");      // Default
locaStepperList.set("Bordered Solver Method", "Householder"); // Householder QR method
locaStepperList.set("Include UV In Preconditioner", true);    // Include U*V in preconditioner for P = J+U*V
locaStepperList.set("Use P For Preconditioner", true);        // true for RowMatrix-based preconditioners,
                                                              // false for CrsMatrix precs (Ifpack_CrsRiluk)

// Create Epetra factory
Teuchos::RefCountPtr<LOCA::Abstract::Factory> epetraFactory = Teuchos::rcp(new LOCA::Epetra::Factory);

// Create global data object
Teuchos::RefCountPtr<LOCA::GlobalData> globalData = LOCA::createGlobalData(paramList, epetraFactory);

// Create the stepper
LOCA::Stepper stepper(globalData, grp, combo, paramList);
```

Turning Point Identification
Moore-Spence Formulation

Turning Point Bifurcation

$$f(x,p) = 0$$
$$Jv = 0$$
$$\phi^T v = 1$$



Full Newton Algorithm

$$\begin{bmatrix} J & 0 & f_p \\ (Jv)_x & J & (Jv)_p \\ 0 & \phi^T & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta v \\ \Delta p \end{bmatrix} = - \begin{bmatrix} f \\ Jv \\ \phi^T v - 1 \end{bmatrix}$$

Bordering Algorithm

$$Ja = -f \qquad \Delta p = \frac{1 - \phi^T v - \phi^T c}{\phi^T d}$$
$$Jb = -f_p$$
$$Jc = -(Jv)_x a - Jv \qquad \Delta x = a + \Delta p b$$
$$Jd = -(Jv)_x b - (Jv)_p \qquad \Delta v = c + \Delta p d$$

… but 4 solves of J per Newton iteration are used to drive J singular!

Sandia National Laboratories

Minimally Augmented Turning Point Algorithm

- Widely used algorithm for small systems:

$$\begin{bmatrix} J & a \\ b^T & 0 \end{bmatrix} \begin{bmatrix} v \\ s \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} J^T & b \\ a^T & 0 \end{bmatrix} \begin{bmatrix} u \\ t \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\Longrightarrow s = t = -u^T J v$$

- J is singular if and only if s = 0

- Turning point formulation:

$$f(x,p) = 0$$
$$s(x,p) = 0$$

- Newton's method:

$$\begin{bmatrix} J & f_p \\ s_x & s_p \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta p \end{bmatrix} = - \begin{bmatrix} f \\ s \end{bmatrix},$$

$$s_x = -(u^T J v)_x = -(u^T J)_x v.$$

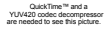
- 3 linear solves per Newton iteration
  - 4 for bordering
  - Symmetric problems reduces to 2 solves.

Sandia National Laboratories

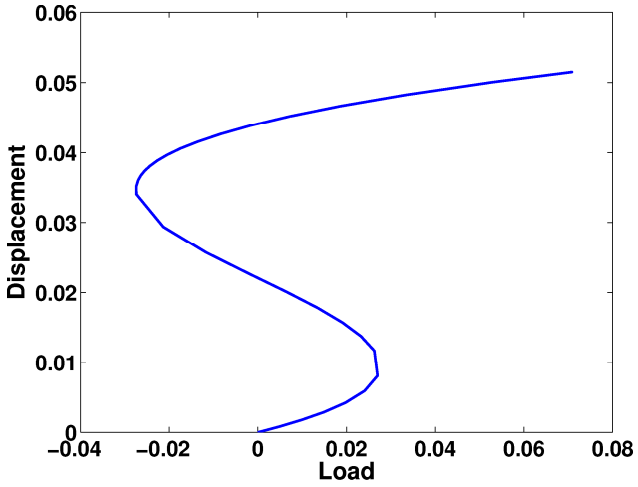
Calling the Householder Bordered Solver Method in LOCA

```
// Create parameter list
Teuchos::RefCountPtr<Teuchos::ParameterList> paramList = Teuchos::rcp(new Teuchos::ParameterList);
Teuchos::ParameterList& locaParamsList = paramList->sublist("LOCA");
Teuchos::ParameterList& locaStepperList = locaParamsList.sublist("Stepper");
locaStepperList.set("Continuation Method", "Arc Length");
locaStepperList.set("Bordered Solver Method", "Nested");

Teuchos::ParameterList& nestedList = locaStepperList.sublist("Nested Bordered Solver");
nestedList.set("Bordered Solver Method", "Householder");
nestedList.set("Include UV In Preconditioner", true);
nestedList.set("Use P For Preconditioner", true);

// Create bifurcation sublist
Teuchos::ParameterList& bifurcationList = locaParamsList.sublist("Bifurcation");
bifurcationList.set("Type", "Turning Point");
bifurcationList.set("Bifurcation Parameter", "Right BC");
bifurcationList.set("Formulation", "Minimally Augmented");
bifurcationList.set("Symmetric Jacobian", false);
bifurcationList.set("Update Null Vectors Every Continuation Step", true);
bifurcationList.set("Transpose Solver Method","Explicit Transpose");
//bifurcationList.set("Transpose Solver Method","Transpose Preconditioner");
//bifurcationList.set("Transpose Solver Method","Left Preconditioning");
bifurcationList.set("Initial Null Vector Computation", "Solve df/dp");

bifurcationList.set("Bordered Solver Method", "Householder");
bifurcationList.set("Include UV In Preconditioner", true);
bifurcationList.set("Use P For Preconditioner", true);
```

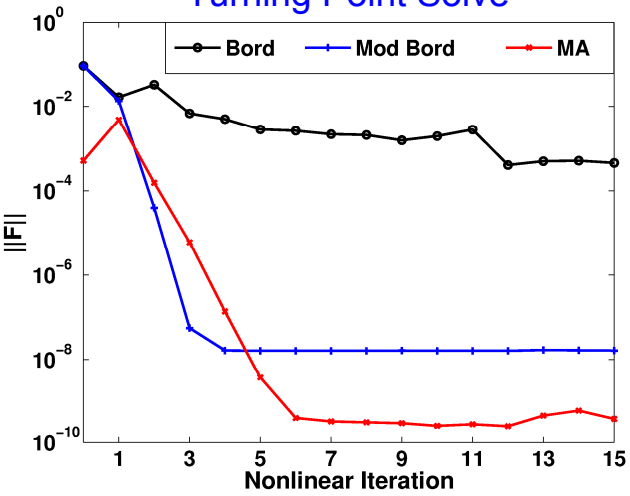
Snap-through Buckling of a Symmetric Cap
(Salinas/FEI, 200K unknowns, 16 procs)



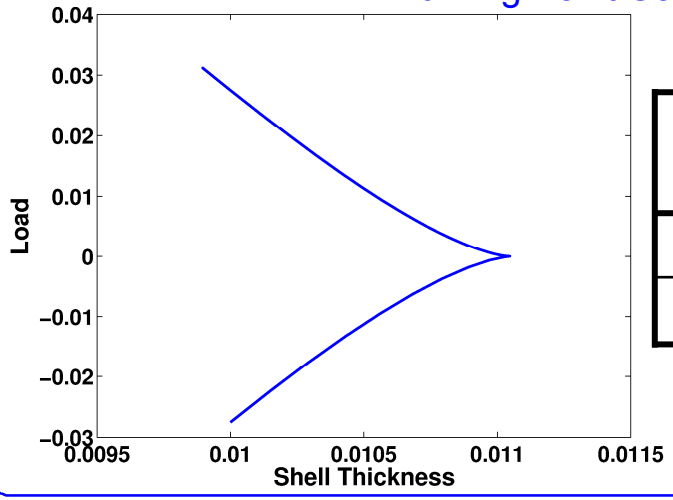
Load/Deflection Curve



Turning Point Solve



Turning Point Continuation



Method	Total Time (hrs)
Mod. Bordering	4.2
Min. Augmented	1.2

- Provides robust buckling info almost 4 times faster than previous best method
- Increases scalability
- Clear how to extend this to other bifurcations

Sandia National Laboratories

Summary

- QR approach provides a convenient way to solve bordered systems
  - Nonsingular
  - Only involves one linear solve
  - Only requires simple vector operations
  - Doesn't change dimension of the linear system
  - Become a workhorse tool in LOCA
- Highly encouraged by minimally augmented turning point formulation
  - No singular matrix solves
  - Improves robustness, scalability and accuracy
  - Requires linear algebra-specific implementation
- Future work
  - Minimally augmented pitchfork, Hopf bifurcations
  - Preconditioners for $\boldsymbol{J + UV^T}$.

Points of Contact

- LOCA: Trilinos continuation and bifurcation package
  - Sub-package of NOX
  - Andy Salinger (agsalin@sandia.gov)
  - Eric Phipps (etphipp@sandia.gov)
  - www.software.sandia.gov/nox
- NOX: Trilinos nonlinear solver package
  - Roger Pawlowski (rppawlo@sandia.gov)
  - Tammy Kolda (tgkolda@sandia.gov)
  - www.software.sandia.gov/nox

Sandia National Laboratories