Applications of Krylov Subspace Methods in Large-Scale Continuation

Homer Walker
Worcester Polytechnic Instititute
June 24, 2004

Joint work with Eric Phipps and Roger Pawlowski. Thanks also to Andy Salinger and Matt Lasater.

General Continuation Problem

Problem: Given $F : I\!R^n \times I\!R^1 \longrightarrow I\!R^n$, solve $F(x, \lambda) = 0$ over a range of (x, λ) values.

Notation: $(x, \lambda) = \bar{x} \in I\!R^{n+1}$, $F(x, \lambda) = F(\bar{x})$, etc.

Typical method framework:

- *Predict* using some extrapolation procedure.
- *Correct* to return approximately to the curve.

Newton Corrector Iterations

Suppose we have "predicted" $\bar{x}_{pred}$.

Newton Corrector Iterations: Set $\bar{x}_0 = \bar{x}_{pred}$, and ...

$$\bar{x}_{k+1} = \bar{x}_k + \bar{s}_k, \quad \textit{where} \quad F'(\bar{x}_k)\,\bar{s}_k = -F(\bar{x}_k).$$

Require ...

$$\boxed{\begin{aligned} F'(\bar{x}_k)\,\bar{s}_k &= -F(\bar{x}_k) \\ \bar{t}^{\,T}\bar{s}_k &= 0 \end{aligned}}$$

...where $\bar{t}$ is an *approximate tangent*.

Newton–Krylov Iterations

Determine $\bar{s}_k$ using a *Krylov subspace method*, e.g., GMRES, BiCGSTAB, TFQMR, ...

Appeal in Newton iterations:

- Some require only Jacobian-vector products.
- Some monotonically decrease the linear residual norm = local linear model norm.

How to adapt to solve the $n \times (n+1)$ constrained system?

Bordering Approach

Bordering solution (H. Keller, 1977; T. Chan, SISSC 1984, ...,):

Write $F'(\bar{x}_k) = (F_x, F_\lambda)$ and $\bar{t} = \begin{pmatrix} \bar{t}_x \\ \bar{t}_\lambda \end{pmatrix}$, and ...

- Solve $F_x\, a = -F(\bar{x}_k)$ and $F_x\, b = -F_\lambda$.
- Set $\Delta\lambda = -(\bar{t}_x^T a)/(\bar{t}_x^T b + \bar{t}_\lambda)$ and $\Delta x = a + \Delta\lambda b$.
- Set $\bar{s}_k = \begin{pmatrix} \Delta x \\ \Delta\lambda \end{pmatrix}$.

Features:

- $\bar{t}^{\,T} \bar{s}_k = 0$ even if solves are inexact.
- Can use codes, preconditioners for $n \times n$ F_x.
- *Two* solves with F_x.
- F_x is *singular at turning points*.

Alternative Approach

From (W., SISC 2000) ...

Abstract procedure:

- Determine $Q \in \mathbb{R}^{(n+1)\times n}$ such that Range $Q = \{\,\bar{t}\,\}^{\perp}$ and $\|Qy\|_2 = \|y\|_2$ for $y \in \mathbb{R}^n$.
- Apply the Krylov subspace method to $F'(\bar{x}_k)Qy_k = -F(\bar{x}_k)$.
- Set $\bar{s}_k = Qy_k$.

Features:

- $\bar{t}^{\,T}\bar{s}_k = 0$ even if solves are inexact.
- *One* solve with $F'(\bar{x}_k)Q$.
- If $\bar{t}$ is an accurate tangent, then $\mathrm{cond}\,(F'(\bar{x}_k)Q) = \mathrm{cond}\,(F'(\bar{x}_k))$.
- Can use codes, preconditioners for $n \times n$ F_x.

Preconditioning

Suppose $M \in I\!R^{n \times n}$ is a preconditioner.

Possibilities:

- Left: $M^{-1} F'(\bar{x}_k) Q y_k = -M^{-1} F(\bar{x}_k), \quad \bar{s}_k = Q y_k.$
- Right: $F'(\bar{x}_k) Q M^{-1} z_k = -F(\bar{x}_k), \quad \bar{s}_k = Q M^{-1} z_k.$
- Etc.

Householder Implementation

- Choose Householder $P = I - 2ww^T \in I\!R^{(n+1)\times(n+1)}$ such that $P\bar{t} = (0, \cdots, 0, 1)^T$.
- Define $Q \in I\!R^{(n+1)\times n}$ by $Qy = P\begin{pmatrix} y \\ 0 \end{pmatrix}$ for $y \in I\!R^n$.
- Apply the Krylov subspace method to $F'(\bar{x}_k)Qy_k = -F(\bar{x}_k)$.
- Set $\bar{s}_k = Qy_k$.

Computing an Accurate $\bar{t}$.

Important ancillary task: computing a unit $\bar{t}$ such that $F'(\bar{x})\bar{t} = 0$.

Given an initial $\bar{t}_0$, ...

- Solve $F'(\bar{x})\bar{s} = -F'(\bar{x})\bar{t}_0$ subject to $\bar{t}_0{}^T\bar{s} = 0$.
- Set $\bar{t} = \dfrac{\bar{t}_0 + \bar{s}}{\|\bar{t}_0 + \bar{s}\|_2}$.

Illustrative Experiments with the Householder Approach

- Rudimentary method, Matlab implementation ...

Step:

  - Given $\bar{x}$, $\bar{t}$, h.
  - Predict: $\bar{x}_{pred} = \bar{x} + h\bar{t}$.
  - Correct to obtain $\bar{x}_{next}$:

    Apply Newton–Krylov corrector iterations.

    Convergence failure $\Rightarrow$ reduce h, repeat.

    Rapid convergence $\Rightarrow$ increase h for next step.
  - Update: $\bar{t} \leftarrow (\bar{x}_{next} - \bar{x})/\|\bar{x}_{next} - \bar{x}\|_2, \quad \bar{x} \leftarrow \bar{x}_{next}$.

- Krylov solvers: GMRES(40), BiCGSTAB.

Illustrative Experiments *(cont.)*

Bratu (Gelfand) problem:

$$\Delta u + \lambda e^{u} = 0 \text{ in } \Omega, \qquad u = 0 \text{ on } \partial\Omega$$

T. Chan (SISSC 1984) problem:

$$\Delta u + \lambda \left(1 + \frac{u + u^2/2}{1 + u^2/100} \right) \text{ in } \Omega, \qquad u = 0 \text{ on } \partial\Omega$$

- $\Omega = [0,1] \times [0,1]$, centered-differences, $m \times m$ grid.
- Fast Poisson left preconditioning.

Illustrative Experiments *(cont.)*

Look at mesh-independence of fast Poisson preconditioning.

Tables show geometric means of linear residual ratios $\|r_{k+1}\|_2 / \|r_k\|_2$.

Bratu problem, $0 \leq$ arclength ≤ 3.

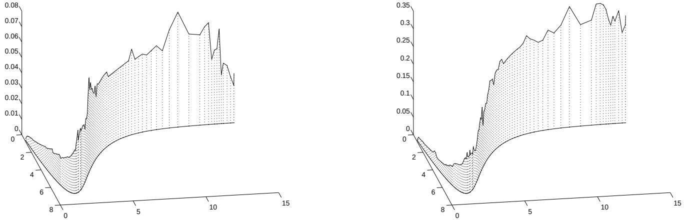
Grid Size	16×16	32×32	64×64	128×128
GMRES(40)	.0291	.0294	.0282	.0285
BiCGSTAB	.0681	.0961	.1091	.1278

Chan problem, $0 \leq$ arclength ≤ 20.

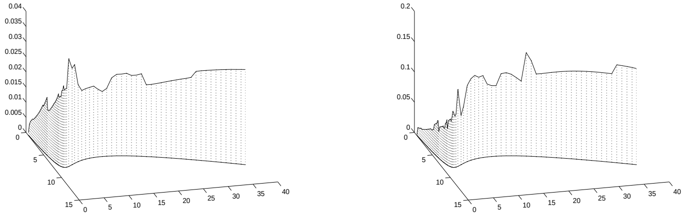
Grid Size	16×16	32×32	64×64	128×128
GMRES(40)	.0207	.0197	.0196	.0205
BiCGSTAB	.0575	.0655	.0789	.0935

Illustrative Experiments *(cont.)*

Look at linear solver convergence along the curves.



Bratu problem, GMRES (left) and BiCGSTAB (right), geometric mean of $\|r_{k+1}\|_2/\|r_k\|_2$, 64×64 *grid.*



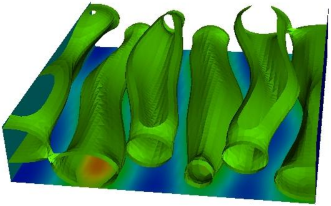
Chan problem, GMRES (left) and BiCGSTAB (right), geometric mean of $\|r_{k+1}\|_2/\|r_k\|_2$, 64×64 *grid.*

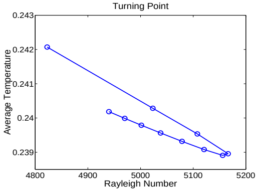
LOCA: **L**ibrary **o**f **C**ontinuation **A**lgorithms

Main developers: Andy Salinger, Eric Phipps, Roger Pawlowski.

- Bifurcation analysis library for large-scale applications on distributed-memory parallel platforms.
- Provides capabilities for
  - parameter continuation (including pseudo-arclength and multi-parameter),
  - bifurcation tracking (turning point, pitchfork, Hopf),
  - linear stability analysis (drivers for eigensolver).
- A Trilinos package (`software.sandia.gov/trilinos`) that wraps the NOX nonlinear solver package, also uses AztecOO linear solvers, Anasazi eigensolver, and Epetra data structures.
- Designed for easy linking to application codes employing Newton-based solvers.

3D Rayleigh–Benard Convection



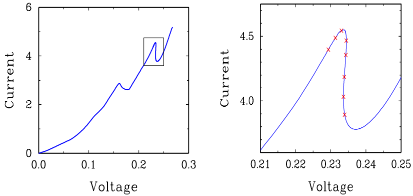


- $5 \times 5 \times 1$ box, 208K unknowns, 16 processors.
- Incompressible Navier–Stokes, heat transport, unstructured finite-element method (MPSalsa).
- 51 continuation steps, fixed step size.

	Newt. Its.	Func. Evals.	Jac. Evals.	GMRES Its.	Total GMRES Time (hrs.)	Total Time (hrs.)
Bordering	134	317	134	56,929	10.01	12.09
Householder	117	283	117	24,612	4.37	5.35

Resonant Tunneling Diode

- Problem described in Lasater talk.
- 1M unknowns, 20 processors.
- Seven continuation steps in the difficult region shown on the right.



	Bordering	Householder
Ave. GMRES Its. for First Linear Solve per Newton Step	214	202
Ave. GMRES Its. for Second Linear Solve per Newton Step	206	0
GMRES Failed to Meet Tolerance	11	0
Total Newton Iterations	21	15
Total GMRES Iterations	8826	3026
Total Newton Solve Time (mins.)	98.4	33.6

In Conclusion

Both the Householder and bordering approaches ...

- satisfy $\bar{t}^{\,T} \bar{s}_k = 0$ even if solves are inexact,
- allow easy re-use of codes, preconditioners for $n \times n$ F_x.

The Householder approach ...

- reduces the number of solves,
- maintains conditioning near turning points,
- improves behavior of the linear *and nonlinear* iterations,
- significantly reduces overall time to solution.