Nonlinear and Krylov Solvers

Presented to
ATPESC 2017 Participants

Barry Smith
Argonne Distinguished Fellow
Mathematics and Computer Science Division
Argonne National Laboratory

Q Center, St. Charles, IL (USA)
Date 08/07/2017
\[ Ax = b \quad x \in \mathbb{R}^N \quad (N = 10^3 \cdots 10^{12}) \]

Why iterative? For three dimensional PDEs (except for special cases) direct methods require

\[ \text{work} \geq CN^\alpha \geq 2 \]
\[ \text{memory} \geq CN^\beta \geq 4/3 \]

Richardson (simple) iteration

\[ x^{n+1} = x^n + (b - Ax^n) \]
\[ e^{n+1} = (I - A)e^n \]

Introduce a preconditioner \( B \)

\[ x^{n+1} = x^n + B(b - Ax^n) \]
\[ e^{n+1} = (I - BA)e^n \]

Damped Richardson \( B = \lambda I \), Jacobi preconditioning \( B = \text{diag}^{-1}(A) \)
Accelerating Simple Iteration

\[ x^{n+1} = x^n + B(b - Ax^n) \]

Note that

\[ x^{n+1} = \sum_{i=0}^{n} \alpha_i (BA)^i Bb \]

That is

\[ x^{n+1} \in K^n = \text{span}\{Bb, (BA)Bb, (BA)^2 Bb, \ldots\} \]

Why not instead define \( x^{n+1} \) by

\[ \min_{x^{n+1} \in K^n} ||B(Ax^{n+1} - b)|| \]
Generalized Minimal Residual (GMRES)

Implemented by constructing an orthonormal basis for $K^n$

$$\{ q_0 = \frac{Bb}{||Bb||}, q_1, q_2, \ldots \}$$

and cleverly solving the minimization in the new basis. The $q_i$ could be called search directions.

Operations required

- inner products and norms (global reductions)
- vector updates (embarrassingly parallel)
- $Ax$ (nearest neighbor operations)
- $B$ (anything goes from no communication to huge amounts of communication and synchronization)

Restarted GMRES - after $n$ iterations throw out the $q_i$ and start again.

Pipelined GMRES - overlap the global reductions with the $Ax$ and $B$. 
Other Commonly Used Krylov Methods

Conjugate gradient (CG) method
- for symmetric, positive-definite matrices
- has a three-term recurrence relation so does not require restart
- requires only two inner products and (optionally) a norm at each iteration

Bi-conjugate gradient stabilized (Bi-CG-stab) method
- uses a short recurrence relation so does not require restart
- requires several inner products and a norm at each iteration
- slower convergence than GMRES
Iterative Solution of Nonlinear Systems

\[
F(x) = b \quad x \in \mathbb{R}^N (N = 10^3 \cdots 10^{12})
\]

**Nonlinear Richardson (simple) iteration**

\[
x^{n+1} = x^n + \lambda (b - F(x^n))
\]

At best

\[
||e^{n+1}|| \leq C ||e^n||.
\]

**Nonlinear CG** - mimic CG to force each new search direction to be orthogonal to the previous directions.

**Anderson mixing (nonlinear GMRES)** - minimize

\[
||F(x^{n+1}) - b||
\]

by using \(x^{n+1}\) as a linear combination of previous solutions and solving a linear least squares problem.

At best

\[
||e^{n+1}|| \leq C ||e^n||^{\alpha \geq 1}.
\]
Newton’s Method

\[ x^{n+1} = x^n - \lambda J^{-1}(x^n)F(x^n) \]

At best

\[ ||e^{n+1}|| \leq C||e^n||^2 \]

Operations required

- inner products and norms (global reductions)
- vector updates (embarrassingly parallel)
- compute \( F() \) and \( J() \) (nearest neighbor operations)
- approximate linear solves (for Newton’s method)
Takeaways for Iterative Solvers for PDEs

- Krylov methods accelerate the convergence of simple iterative schemes
- Most commonly used Krylov methods: GMRES, CG, Bi-CG-stab
- Nonlinear solvers range from simple iteration (weak) to Newton’s method (strong)
- Components of linear and nonlinear solvers are similar
  - Embarrassingly parallel vector operations
  - Global reduction based inner products and norms
  - Nearest neighbor matrix-vector products and function evaluation
  - Nearest neighbor Jacobian evaluations
  - From embarrassingly parallel (weak) to strongly coupled (strong) preconditioners

- For extreme scale (millions of cores) global reductions are best avoided or mitigated with pipelining in iterative solvers
High-performance iterative solvers available via

- PETSc
- SUNDIALS
- Trilinos

Complementary capabilities that meet the needs of different HPC apps

The remaining slides focus on PETSc.
PETSc/TAO:
Portable, Extensible Toolkit for Scientific Computation / Toolkit for Advanced Optimization

- **Easy customization and composability of solvers at runtime**
  - Enables optimality via flexible combinations of physics, algorithmics, architectures
  - Try new algorithms by composing new/existing algorithms (multilevel, domain decomposition, splitting, etc.)

- **Portability & performance**
  - Largest DOE machines, also clusters, laptops
  - Thousands of users worldwide

**Scalable algebraic solvers for PDEs.** Encapsulate parallelism in high-level objects. Active & supported user community. Full API from Fortran, C/C++, Python.

PETSc provides the backbone of diverse scientific applications. Clockwise from upper left: hydrology, cardiology, fusion, multiphase steel, relativistic matter, ice sheet modeling

https://www.mcs.anl.gov/petsc
PETSc: Platform for experimentation

- **No optimality without interplay among physics, algorithmics, and architectures**

- **Need algebraic solvers to be:**
  - **Composable:** Separately developed solvers may be easily combined, by non-experts, to form a more powerful solver.
  - **Hierarchical:** Outer solvers may iterate over all variables for a global problem, while inner solvers handle smaller subsets of physics, smaller physical subdomains, or coarser meshes.
  - **Nested:** Outer solvers call nested inner solvers.
  - **Extensible:** Users can easily customize/extend.

- Many solver configurations can be set at runtime to avoid needing to recompile.
PETSc/TAO capabilities

### Functionality

- **Optimization**
- **Time Integrators**
- **Nonlinear Algebraic Solvers**
- **Krylov Subspace Solvers**
- **Preconditioners**
- **Domain-Specific Interfaces**
  - Networks
  - Quadtree / Octree
  - Unstructured Mesh
  - Structured Mesh
- **Vectors**
- **Index Sets**
- **Matrices**
- **Computation & Communication Kernels**

### More Details (Algorithms, Data Structures, etc.)

<table>
<thead>
<tr>
<th>PDE Constrained</th>
<th>Adjoint Based</th>
<th>Derivative Free</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pseudo-transient General Linear</td>
<td>Runge-Kutta IMEX</td>
<td>Strong Stability Preserving Rosenbrock-W</td>
<td>Others</td>
</tr>
<tr>
<td>Line Search Newton Trust Region Newton</td>
<td>Quasi-Newton (BFGS) Nonlinear Multigrid (FAS)</td>
<td>Nonlinear Gauss Seidel Successive Substitutions</td>
<td>Nonlinear CG Active Set VI</td>
</tr>
</tbody>
</table>

- **Pipeline methods**
  - Hierarchical Krylov
  - Additive Schwarz
  - Geometric Multigrid
  - ILU/ICC
  - App-specific
  - Schur Complement
- **Schur Complement**
  - Others

- **Infrastructure networks, e.g., electrical, gas, water**
- **Structured mesh refinement**
- **Complex domains with finite element and finite volume discretizations**
- **Simple domains and discretizations, e.g., finite difference methods**
- **Compressed Sparse Row (AIJ)**
  - Block AIJ
  - Matrix Blocks (MatNest)
  - GPU and Phread Matrices
- **Block AIJ**
  - Dense
  - GPU
- **Matrices**
  - MPI, OpenMP, MPI-IO, CUDA, Pthreads, BLAS, LAPACK, etc.

---

*Note: The table and diagram are placeholders for the actual content provided in the image.*
Multiphase steel modeling

Computational scale bridging: coupled microscopic-macroscopic steel simulation

- Uses nonlinear and linear FETI-DP domain decomposition methods (in PETSc) and algebraic multigrid (in hypre)
- Demonstrates excellent performance on the entire Blue Gene/Q the Argonne Leadership Computing Facility (Mira: 1,572K MPI processes).

References: