Nonlinear Solver Algorithms at the Exascale: Rethinking the Full Linearization Bottlenecks

Peter Brune, Barry Smith  
MCS Division, Argonne National Lab

Matt Knepley  
CI, University of Chicago

Examath13 Workshop  
Washington, D.C.

August 22, 2013
Newton-Krylov methods
- assembled Jacobian inexactely inverted by Krylov method
- line search globalization
- the preconditioner is where localization happens
  - multigrid, domain decomposition, fieldsplit
  - local inner work is a decomposition of a global linearization

This won’t work at the Exascale!!!
- low intensity: matrices are memory bandwidth-heavy
- high synchronization: line searches, Krylov convergence, etc.
New Nonlinear Solvers in PETSc

- **global** nonlinear iterative solvers:
  - nonlinear Krylov (NGMRES, NCG, NRICH)
  - quasi-Newton (QN)

- **decomposition** nonlinear solvers:
  - nonlinear additive-Schwarz (NASM)
  - full approximation scheme (FAS)
  - Gauss-Seidel-Newton (GSN)

- easily **interchangeable** from the command line
  - solver types
  - line searches
  - subsolvers

- PETSc’s **nonlinear** (SNES) solvers more like **linear** (KSP)

- but there’s more...
New Solver Framework in PETSc

1. **nonlinear preconditioning**
   - outer nonlinear globalization and inner nonlinear preconditioner
   - easy-to-access implementations of ASPIN, NGMRES-FAS, etc.
   - command-line customization of the hierarchy

2. **composite solvers**
   - multiplicative and additive combination of solvers
   - easy-to-use nonlinear elimination
These really work!

- high Rayleigh number (Ra = 2e4) flow
- time, iterations, V-cycles, **intensity** (GFLOPs), MPI reductions
- just a **demonstration**; 64 cores, 4k unknowns per core
- Newton-(GMRES-MG) with nonlinear elimination vs. NGMRES-FAS

<table>
<thead>
<tr>
<th></th>
<th>NK-MG</th>
<th>NASM*(NK-MG)</th>
<th>NGMRES-FAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>time (sec)</td>
<td>7</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>its.</td>
<td>24</td>
<td>12</td>
<td>22</td>
</tr>
<tr>
<td>V-Cycles</td>
<td>354</td>
<td>155</td>
<td>22</td>
</tr>
<tr>
<td>GFLOPs</td>
<td>11</td>
<td>14</td>
<td>32</td>
</tr>
<tr>
<td>MPIReduct</td>
<td>4129</td>
<td>2711</td>
<td>775</td>
</tr>
</tbody>
</table>
Conclusions

1. play to future machines’ strengths:
   - **algebraic intensity** esp. for heterogeneous machines
   - **hierarchy** is in the architecture
   - **synchronization** must be used sparingly

2. predictions:
   - local and **hierarchical** solver variants
     - **additive** variants (FAS, composite)
     - **adaptivity** and nonlinear load balancing
   - FAS as a workhorse
     - FAS has much **higher throughput** than matrix-based MG
     - FAS as preconditioner or composite subsolver
     - **low-communication** variants (segmental refinement FAS)

3. rethink design:
   - design space **underexplored**
   - suite of alternative solvers
   - needs mathematical and software effort