Linear Solver Performance Analysis of MFiX integrated with a Next Generation Computational Framework

V. Kotteda, PostDoc,
V. Kumar, Associate Professor, Mechanical
A Rodriguez, UG RA,
A Schiaffino, Grad. Student, Mechanical,
A Chattopadhyay, Grad. Student, Computational Science
The University of Texas at El Paso

W. Spotz, Senior Research Staff, Sandia National Labs – NM

NETL Workshop on Multiphase Flow Science
August 8-10, 2017
Morgantown, WV
Outline

- Goals and Objectives
- Technical approach
  - What is Trilinos?
  - Scalability and Portability
  - Sandia’s Next-Gen Computational Framework - MPI-X, UQ, etc
  - Trilinos-MFiX Framework
- Results & Discussion
- Concluding remarks
- Future work
The technical goal of this project is to develop, validate and implement advanced linear solvers to replace MFiX’s existing linear solvers. This goal will be achieved by integrating Trilinos, a publicly available open-source linear equation solver library developed by Sandia National Laboratory. The project will demonstrate scalability of the Trilinos- MFiX interface on various high-performance computing (HPC) facilities including the ones funded by the Department of Energy (DOE).

The expected results of the project will be reduction of computational time when solving complex gas-solid flow and reaction problems in MFiX, and reduction in time and cost of adding new algorithms and physics based models into MFiX

**Objectives**

- Create a framework to integrate MFiX with Trilinos linear solver packages
- Validate MFiX suites of problems on HPC systems with and without GPU acceleration
- Evaluate the performance
MFiX: Challenges and Opportunities

MFiX (developed by NETL)
- Model multiphase physics
- Widely used by the fossil fuel reactor communities and beyond
- Can significantly reduce time & cost to design a reactor

However
- **Computational expense** for most **practical applications** can make it **impractical**
- Limited software capabilities
  - Linear solver, MPI-X, UQ, etc.
- Can result in poor convergence especially in complex non-linear problems

But, could be made more practical if we could significantly reduce time-to-solution by
- Effectively exploiting HPC systems (massively parallel computers, GPUs, multithreading..)
- **Leveraging** state-of-the-art **preconditions and linear solver libraries**
- Providing a long-term portable and scalable software development framework
Supercomputers

https://www.top500.org/
What is Trilinos?

- Object-oriented software framework for…
- Solving big complex science & engineering problems
- More like LEGO™ bricks than Matlab™

Trilinos provides the state-of-the-art preconditions and linear solver libraries
- demonstrate scalability on current HPC systems
- illustrate plans for continued maintenance
- include support for new hardware technologies
Next-generation Computational Framework

- High fidelity modeling and simulation is a **critical enabling technology** to understand physical systems
- Exascale computing
  - provides potential for **high fidelity solutions to complex multiscale multiphysics problems**
  - requires exploiting **homogeneous or heterogeneous** mixtures CPUs and specialized processing units (modern day supercomputers!)
  - requires **next generation of computational framework and models** that are portable and scalable to 100,000s of processors

\[ L(u) = f \]

**Math. model**

\[ L_h(u_h) = f_h \]

**Numerical model**

\[ u_h = L_h^{-1} f_h \]

**Algorithms**

Find faster and more efficient ways to solve numerical models

### Numerical math
Convert to models that can be solved on digital computers

### Algorithms
Find faster and more efficient ways to solve numerical models

- **Beyond a “solvers” framework**
- Natural expansion of capabilities to satisfy application and research needs
- Discretization methods, Automatic Differentiation, Mortar methods, …
Trilinos Package Categories

- Tools
- Linear Algebra Services
- Iterative Linear Solvers
- Direct Linear Solvers
- Preconditioners
- Nonlinear Solvers
- Eigensolvers
- Optimization & Analysis
- Meshing & Load Balancing
- Discretizations
- Performance Portability
# Trilinos Package Advancement

## Table

<table>
<thead>
<tr>
<th>Category</th>
<th><strong>1&lt;sup&gt;st&lt;/sup&gt; Generation</strong></th>
<th><strong>2&lt;sup&gt;nd&lt;/sup&gt; Generation</strong></th>
<th>**3&lt;sup&gt;rd&lt;/sup&gt; Gen</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Algebra Services</td>
<td>Epetra, EpetraExt, Komplex</td>
<td>Tpetra, Xpetra, Domi, RTOp, Thyra</td>
<td>Tpetra, Xpetra</td>
</tr>
<tr>
<td>Tools</td>
<td>Teuchos, Triutils, Galeri, Optika, Trios</td>
<td>Teuchos, Sacado, Trios</td>
<td></td>
</tr>
<tr>
<td>Direct Linear Solvers</td>
<td>Amesos, Pliris</td>
<td><strong>Amesos2</strong></td>
<td></td>
</tr>
<tr>
<td>Iterative Linear Solvers</td>
<td>AztecOO</td>
<td>Belos, Stratimikos</td>
<td>Belos</td>
</tr>
<tr>
<td>Preconditioners</td>
<td>IFPACK, ML</td>
<td>IFPACK2, MueLu, ShyLU</td>
<td>IFPACK2, MueLu</td>
</tr>
<tr>
<td>Nonlinear Solvers</td>
<td>NOX, LOCA</td>
<td>NOX, LOCA</td>
<td></td>
</tr>
<tr>
<td>Eigensolvers</td>
<td>Anasazi</td>
<td>Anasazi</td>
<td></td>
</tr>
<tr>
<td>Optimization &amp; Analysis</td>
<td>MOOCHO</td>
<td>MOOCHO, OptiPack, Phalanx, Piro, ROL</td>
<td></td>
</tr>
<tr>
<td>Meshing &amp; Load Balance</td>
<td>STK, Zoltan, Isorropia, Mesquite, Moertel</td>
<td>STK, Zoltan2, Pamgen</td>
<td></td>
</tr>
<tr>
<td>Discretizations</td>
<td>Intrepid, Shards, Rythmos</td>
<td>Intrepid, Shards, Tempus</td>
<td></td>
</tr>
<tr>
<td>Performance Portability</td>
<td></td>
<td></td>
<td><strong>Kokkos</strong></td>
</tr>
</tbody>
</table>
Trilinos + MFiX

Using Trilinos Linear Solver with MFiX
Trilinos-MFiX Flow chart

1. Decompose the domain based on NODESI, NODESJ and NODESK
   - Initialization of computations
   - Compute various terms/fluxes in equations for the fluid phase
   - Apply BC and solve the system of equations for fluid flow variables
   - Compute various terms/fluxes in equations for solids phase
   - Apply BC and solve the system of equations for solid phase variables
   - Output

   Trilinos
   - Yes
   - LEQ < 6
   - MFiX

   Yes
   - Finished time steps
   - Stop
   - No

   Trilinos

<table>
<thead>
<tr>
<th>LEQ</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SOR</td>
<td>Point Successive Over Relaxation</td>
</tr>
<tr>
<td>2</td>
<td>BiCGSTAB</td>
<td>Bi-Conjugate Gradient STABilized method</td>
</tr>
<tr>
<td>3</td>
<td>GMRES</td>
<td>Generalized Minimal RESidual algorithm</td>
</tr>
<tr>
<td>4</td>
<td>BICGSTAB + GMRES</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CG</td>
<td>Conjugate Gradient</td>
</tr>
<tr>
<td>6</td>
<td>BiCGSTAB/GMR ES/CG/Direct/…</td>
<td>Trilinos</td>
</tr>
</tbody>
</table>
A language independent interface to integrate legacy codes

Transfer A & b
Define solver attributes

MFiX wrapper
Interpret matrix structure, Implement CRS scheme

Fortran wrapper
Semantic for memory references

C wrapper
Communicator, Interpret Polymorphism representations

Cpp wrapper
Create Epetra_MAP, Fill A & b

Solution of Ax=b

LEQ = 6

Loop number of time steps
Loop number of nonlinear iteration
If LEQ=6
  Solve linear equations governing the fluid and solid/bubbles using Trilinos solver
Else
  Solve using MFiX solvers
End
End
End
Properties

- Particle diameter = 0.04 cm
- Particle density = 2.0 g/cm³
- Restitution coefficient = 0.80
- Angle of internal friction = 30°
- Fluid viscosity = 0.00018 g/cm s
- Fluid density = 0.0012 g/cm³

Boundary conditions

- Inlet: constant mass inflow
  - 124.6 cm/s for 4.3 < x < 5.7; 25.9 cm/s for 0 < x < 4.3, 5.7 < x < 10
- Sidewalls: slip condition
- Outlet: pressure outflow condition (p = 0)

Dimensions: 10 cm X 100 cm
Flow in a fluidized bed (3D)

**Bubbling bed**
Dimensions: 10cm x 10cm x 100cm

- Sand particle diameter = 0.04cm
- Sand particle density = 2.0 g/cm³
- Restitution co-efficient = 0.80
- Angle of internal friction = 30
- Fluid (gas) viscosity = 0.00018 g/cm s
- Fluid density = 0.0012 g/cm³

**Boundary conditions**
- Inlet: constant mass inflow (124.6 cm/s for 4.3<x<5.7, 4.3<z<5.7)
- Sidewalls: slip condition
- Outlet: pressure outflow condition
Flow in a fluidized bed

MFIX

MFIX-TRILINOS

DEM

PIC
3D Bubbling flow Problem
Case 1: Mesh Size = 10M
Case 2: Mesh size = 200M

Computers:
- **Stampede**: Texas Advanced Computing Center (TACC)
- **Bridges**: Pittsburgh Supercomputing Center (PSC)
- **Comet**: San Diego Supercomputer Center (SDSC)

Various computer architectures used for the performance analysis study

<table>
<thead>
<tr>
<th></th>
<th>Stampede (AS)</th>
<th>Comet (AC)</th>
<th>Bridges (AB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model (Intel Xeon)</td>
<td>E5-2680 2.7GHz</td>
<td>E5-2695 2.5GHz</td>
<td>E5-2695 2.30 GHz</td>
</tr>
<tr>
<td>Cores per socket</td>
<td>8</td>
<td>12</td>
<td>14</td>
</tr>
<tr>
<td>Sockets</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>L1 cache (KB)</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>L2 cache (KB)</td>
<td>256</td>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>L3 cache (KB)</td>
<td>20480</td>
<td>30720</td>
<td>35840</td>
</tr>
<tr>
<td>RAM (GB)</td>
<td>32</td>
<td>128</td>
<td>130</td>
</tr>
</tbody>
</table>
Performance of MFiX-Trilinos

Trilinos

Solvers: BiCGStab
Packages: Tpetra (obj), Belos (solver), MueLu (pre)

PC: Jacobi
Linear iterative solvers

**Trilinos**

Solvers: BiCGStab, TFQMR, Pseudoblock TFQMR, GMRES, Flexible GMRES, Recycling GMRES, Hybrid GMRES

Packages: Tpetra (obj), Belos (solver), MueLu (pre)

Mesh M1 (10M)  Mesh M2 (200M)
First and second generation solver stacks

**First Generation**
Iterative Solvers: BiCGStab, GMRES
Packages: Epetra (obj), Aztec(solver), ML
PC: Smoothed Aggregation

**Second generation**
Iterative Solvers: BiCGStab, GMRES
Packages: Tpetra (obj), Belos (solver), MueLu
PC: Smoothed Aggregation
Concluding remarks and future work

- Presented MFiX linear solver integration framework with Trilinos’s Linear solver
- Analyzed solver performance for various problem sizes and HPC systems -- a speed of ~5 times was noticed

Future
- Portability – Kokkos
- Efficient memory and data transfer management

Thank you!

Q&A ?

Acknowledgments: DOE/NETL – UCR, XSEDE, Sandia, ME & Computational Depts