Interfacing MFIX with PETSc and HYPRE Linear Solver libraries

Award #: DE-FE0026191

PI: Gautham Krishnamoorthy (UND)
Co-PI: Jeremy Thornock (U.Utah)
October 28th, 2015
Presentation at:
DOE-NETL
Objective/Vision

• Build a robust, well-abstracted, interface to the PETSc, HYPRE linear solver libraries from MFIX

• Code verification against established MFIX solutions and code to code comparisons

PETSc

HYPRE

MFIX
Input and make files

MFIX
Code

Improved speed and scaling performance

UND

THE UNIVERSITY OF UTAH
Objective/Vision (continued)

- Identification of optimum solvers and pre-conditioners
Examine the scaling of MFIx when invoking PETSc and HYPRE.
Background
Software Abstraction

The problem:
• MFIX already has linear solver options
• Interfacing with the linear solver packages is not universal (different stencil setup operations)
• Fortran (MFIX is written in F90) isn’t an object-oriented programming language

Our approach:
• To enable programmers and users, a well abstracted ‘linear solver interface’ is required
• Operations to setup a general linear solve (Ax=b), is easily abstracted
  ➢ Compute matrix and vector elements (local to global mapping in PETSc and HYPRE)
• Object orientation can be simulated in F90 with some clever use of existing F90 features
• Define a common interface (a.k.a base class) and derive specific solver interface for existing MFIX solvers, HYPRE, and PETSC, etc.
Background

Hypre

- Hypre (LLNL) is a linear solver package for the solution of preconditioned, sparse linear systems (including multigrid)
- Hypre includes native support for Fortran codes (MFIX)
- U.Utah and UND have extensive experience using Hypre for septadagonal matrix systems (Pressure-Poisson and P-1 radiation model)
- Hypre is current production linear solver for the U. of Utah’s combustion (including PC) Arches code (Thornock; current software architect)

Arches/Hypre weak scaling up to 512K cores for a PC combustion simulation (simulations performed on Mira)
Background

PETSc

- PETSc (ANL) is a linear solver package for the solution of preconditioned, sparse linear systems (KSP)
- PETSc includes native support for Fortran codes (MFIx)
- U. Utah and UND have extensive experience using PETSc (non-symmetric matrices resulting from the discrete ordinates radiation model)
Team Description

PROJECT MANAGER
University of North Dakota
UND
Gautham Krishnamoorthy

HYPRE

UU Principal Investigator
Jeremy Thornock

UU Graduate Student

PETSc

UND Principal Investigator
Gautham Krishnamoorthy

UND Graduate Student

1 visit/year to partner University

Collaboration, Virtual meetings and exchanges
Task 1 (Description): Project Management

- UND and U.Utah have discussed and concurred on a plan to move forward on this project
- MFIX, PETSc and HYPRE have been downloaded, compiled and tutorials have been run at both centers
- A graduate student has been recruited at UND, recruitment efforts ongoing at Utah
Task 2 (Description): Interfacing MFIX with PETSc and HYPRE

- **Problem Setup**: Solver parameters (solver tolerances, maximum number of iterations, solver types, pre-conditioners etc...)

- **Solver Setup**: Solver object creation (allocation of A, x, and b) and initialization methods.

- **Communication Linear System**: Handshake (or “mapping”) function for passing the linear system coefficients (A) and right-hand-side values (b) in the current native MFIX data-structure to the solver-specific types.

- **Solve System**: Compute the solution (x) to the linear system

- **Return/Copy Solution**: Conversion of the solver type solution (x) to the current, native MFIX type

- **Cleanup**: De-allocation and destruction of solver objects

*one-time costs during simulation start-up during a transient calculation
Task 3 (Description): Code verification

- Established solutions from MFIX tutorials on serial and parallel machines
- Code-to-code comparisons (ANSYS FLUENT, ARCHES)

Task 4 (Description): Solver optimization

- PETSc (GMRES, BiCGSTAB)
- HYPRE (Pre-conditioned multigrid)
Task 5 (Description): Scaling studies

U. Of Utah’s Ash cluster will be used for scaling tests

Ash Cluster Hardware Overview

- 253 12 Core Nodes and 164 20 Core Nodes (6316 total cores)
- 2.8 GHz Intel Xeon (Westmere X5660) processors
- 24 Gbytes memory per node on the 12 core nodes
- 64 Gbytes memory per node on the 20 core nodes
- Mellanox FDR Infiniband interconnect
- Gigabit Ethernet interconnect for management

(photo credit: Sam T. Liston)
# GANTT chart

<table>
<thead>
<tr>
<th>Task ID</th>
<th>Task Name</th>
<th>2015</th>
<th>2016</th>
<th>2017</th>
<th>2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Project Management Planning</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2a</td>
<td>Interfacing PETSc with MIFIX (UND)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2b</td>
<td>Interfacing HYPRE with MIFIX (UU)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Verification and testing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4, 5</td>
<td>Solver optimization and scaling studies</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The chart indicates the progress of tasks across different quarters from 2015 to 2018.
Milestones

Year 1 (9/1/2015 – 8/31/2016)
• Successful mapping of “A” and “b” elements from native MFIIX data structure to PETSc and HYPRE solver objects
• MFIIX runs with PETSc and HYPRE linear solver options on serial machines. Solutions verified against existing MFIIX tutorials
• Solver options are specified within the source code

Year 2 (9/1/2016 – 8/31/2017)
• Refinement of the solver interface
• Extension of MFIIX-PETSc-HYPRE coupling to parallel machines and solution verification

Year 3 (9/1/2017 – 8/31/2018)
• Identification of optimum solvers (explore dependency on flow regimes)
• Scaling performance
• Rigorously assess improvement in time to solution compared to native MFIIX linear solvers