Solvers in PyLith

Brad Aagaard, Matthew Knepley*, Charles Williams

Computation Institute
University of Chicago

Department of Molecular Biology and Physiology
Rush University Medical Center

Crustal Deformation Modeling Tutorial
Cyberspace, Jun. 19–24, 2011

How Do We Solve a System of Linear Equations?

For a set of linear equations,

$$\left[\begin{array}{c} \\ A \\ \\ \end{array}\right] \left[\begin{array}{c} \\ x \\ \\ \end{array}\right] = \left[\begin{array}{c} \\ b \\ \\ \end{array}\right]$$

we can get the solution x using

- Gaussian elimination (LU)
- QR factorization and backsolve (QR)
- SVD factorization (SVD)

However, these methods use lots of memory and time.

Always start with LU

For any new problem, always begin with a small, serial version which can be solved by LU factorization using the options below:

```
--petsc.ksp_type preonly --petsc.pc_type lu
```

This eliminates the solver as a variable, so that non-convergent or inaccurate solutions do not complicate interpretation of the model. You can move to slightly larger and parallel problems by adding

```
--petsc.pc_factor_mat_solver_package mumps
```

How Do We Solve a System of Linear Equations?

We can reduce the problem size using subspace projection, where V is the basis for a small subspace. We solve a small system, $V^T A V$, and then project that solution into the full solution space.

$$\begin{bmatrix} V^T \end{bmatrix} \begin{bmatrix} A \end{bmatrix} \begin{bmatrix} V \end{bmatrix} \begin{bmatrix} y \end{bmatrix} = \begin{bmatrix} V^T \end{bmatrix} \begin{bmatrix} b \end{bmatrix}$$

$$\begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} V \end{bmatrix} \begin{bmatrix} y \end{bmatrix}$$

What is a Krylov Method?

A Krylov Method uses matrix vector products with the system matrix A to generate a subspace V in which the problem is solved.

$$V = \text{span}\left\{b, Ab, A^2b, \ldots, A^{k-1}b\right\}$$

GMRES is a Krylov method which generates an approximate solution $\hat{x}$ that minimizes the residual, $||b - A\hat{x}||$, over all choices $\hat{x} \in V$.

What is a Krylov Method?

Krylov methods are not robust solvers. They are generally used to accelerate other methods, called preconditioners,

- Relaxation (Jacobi, Gauss-Seidel, Successive Over-Relaxation)
- Additive Schwarz Method and Block-Jacobi
- Multigrid (MG)
- Incomplete Factorization (Cholesky, LU)

which can stagnate. Thus Krylov solvers are used to make existing solvers more robust.

What are Schwarz Methods?

Schwarz Methods, a particular kind of Domain Decomposition, divide the domain into overlapping subdomains and repeatedly solve the equations over each subdomain. In terms of linear algebra, using only two subdomains, we have

$$\begin{bmatrix} R_0^T A R_0 & \\ & R_1^T A R_1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} R_0^T b \\ R_1^T b \end{bmatrix}$$

The solution is repeatedly updated using $x = x_0 + x_1$. The Block-Jacobi method is the degenerate case where we have no overlap between subdomains.

Why are these Methods Inadequate?

The traditional preconditioners we have described perform poorly for PyLith. They are designed for

scalar problems,

which model a single physics,

and are dominated by local interactions.

These three limitations must be overcome to achieve efficient, scalable solutions in PyLith.

Why is PyLith Nonlocal?

Elliptic equations, the kind we solve for the quasistatic problem in PyLith, have nonlocal behavior. For example, a small perturbation to a boundary condition changes not just the solution around that part of the boundary, but the entire solution. Prototypical elliptic equations are electrostatics

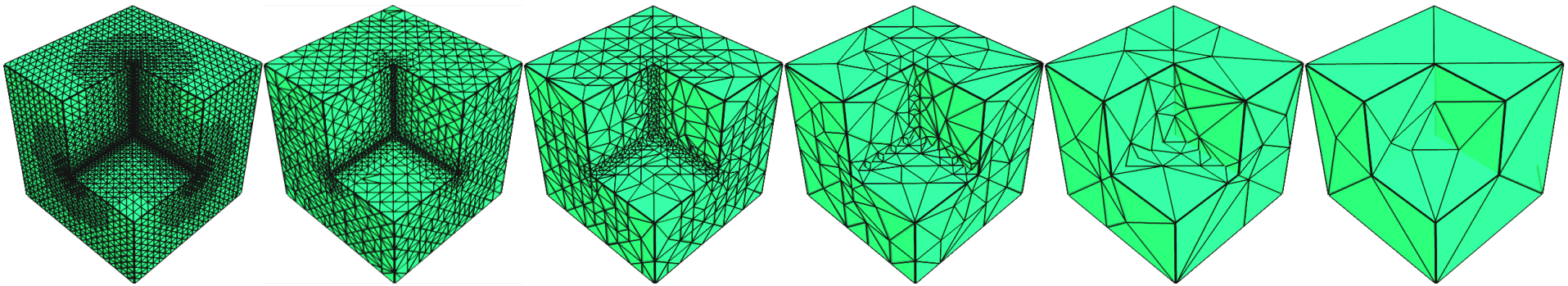
$$\Delta\phi = -\rho$$

and linear elasticity

$$\nabla \cdot \left(\nabla \mathbf{u} + \nabla^T \mathbf{u}\right) = -\mathbf{f}$$

What is Multigrid?

Multigrid uses the traditional preconditioners to solve problems on a series of coarse grids. Since coarse grids are small, they can be solved much faster than the original problem. The coarse grid solutions provide global information, which is used to speed up the original solve. However, this requires knowledge of the user's mesh, and the ability to coarsen it.



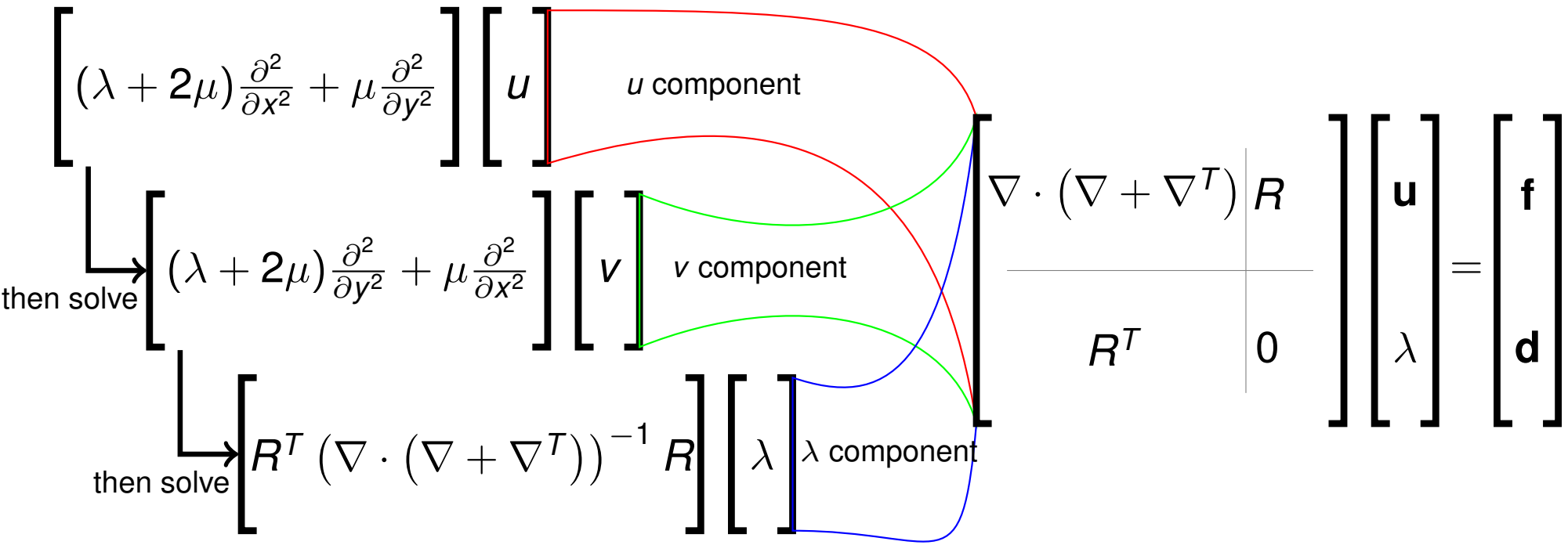
What is Algebraic Multigrid?

Algebraic Multigrid (AMG) also constructs coarse problems in order to generate global information. However, it does not use coarse meshes. Instead, AMG builds coarse linear systems using only information from the original system. Thus, it can be applied to any problem, although it is only effective on certain classes of problems.

In PyLith, we use AMG to solve single components of the momentum balance equation using the ML package from Sandia, which can be automatically installed and used through PETSc.

What is PCFIELDSPLIT?

The FieldSplit preconditioner in PETSc solves individual systems, such as different components of momentum balance, and then combines these solves to produce the overall solution.



By combining traditional preconditioners, effective for simple scalar equations, we can construct a very effective overall solver.

What is Recommended for PyLith?

Two examples, `examples/3d/tet4/step02.cfg` and `examples/3d/tet4/step04.cfg`, use the field-split solver.

Property	Value	Description
fs_pc_type	field_split	Precondition fields separately
fs_pc_fieldsplit_real_diagonal		Use diagonal blocks from the true operator, rather than the preconditioner
fs_pc_fieldsplit_type	multiplicative	Apply each field preconditioning in sequence, which is stronger than all-at-once (additive)
fs_fieldsplit_n_pc_type	ml	Multilevel algebraic multigrid preconditioning using Trilinos/ML via PETSc, n=0,. . . ,N+1
fs_fieldsplit_n_ksp_type	jacobi	Jacobi is sometimes the best option for fault preconditioning, however we often use ML, n=0,. . . ,N+1

Table: PETSc Options which activate PCFIELDSPLIT

More Information

The PyLith Manual Section on PETSc has a thorough discussion of PETSc solver options for PyLith, as well as default values.

The PETSc Homepage contains a user manual, FAQ, webpage documentation for each function, and hundreds of example codes.