Roundtable: What is Wrong with Automatic Mathematical Modeling

Matthew G Knepley

Mathematics and Computer Science Division
Argonne National Laboratory

March 5, 2007
Workshop on Automating the Development of Scientific Computing Software
LSU, Baton Rouge, LA

Challenges

- Can we establish good interfaces for all levels of the hierarchy?
- Do we need language extensions for more sophisticated problems?
- What information is required from each component?
- Is inter-language programming effective?
- Can we develop a general framework for boundary conditions?

Outline

Interface Hierarchy

- Coupling
  - Semi-implicit
  - Dynamic generation of Jacobian blocks
- Domain decomposition
  - Parallelism
  - BDDC, FETI
  - Assembly

Necessary Information

- Discretization (FIAT)
  - # of dof/dimension
  - Jet tabulation
  - Action of dual basis
  - Dof *kind*
- Functions (Section)
  - `restrict/update`
  - `complete`
- Topology (Sieve)
  - `cone/support`
  - `closure/star`
  - `meet/join`

- Equations (FFC)
  - Unknowns
    - Tensors
  - Knowns
    - Interpolants
    - Arbitrary functions
  - Arithmetic, Matrix arithmetic
  - Derivatives
    - div, grad, curl
    - exterior derivative/coderivative
  - Domain
    - Link mesh to equation
  - Boundary conditions
    - Operator for Dirichlet condition?
  - `jump`, `avg`

Outline

Language Extensions

- Optimization
  - *min*, *s.t.*
- Sensitivity
- Control

Inter-language Programming

I am now convinced that this causes more problems than it solves:

- Debugging
- Build systems
- Passing complex structures (objects)
- Top-level control

Outline

Dirichlet Values

- Topological boundary is marked during generation
- Cells bordering boundary are marked using `markBoundaryCells()`
- To set values:
  1. Loop over boundary cells
  2. Loop over the element closure
  3. For each boundary point i, apply the functional N_i to the function g
- The functionals are generated with the quadrature information
- Section allocation applies Dirichlet conditions automatically
  - Values are stored in the Section
  - `restrict()` behaves normally, `update()` ignores constraints

Boundary Condition Implementation

- Associate a transform $\mathcal{T}$ to each sieve point
    - Could extend to the cell closure
- Constrained points also have a rotation $\mathcal{C}$
    - Rotates to frame in which constrained dof are last
    - Easy to mix fields on a point
    - Applies for all values (get rotated equations)
- `update()` ignores constrained values
    - `restrict()` always retrieves all data