Finite Element Assembly on Arbitrary Meshes

Matthew Knepley

Computation Institute
University of Chicago
Department of Molecular Biology and Physiology
Rush University Medical Center

Department of Applied Mathematics and Computational Science
King Abdullah University of Science and Technology
Apr 5, 2010
Collaborators

- **Automated FEM**
  - Andy Terrel (UT Austin)
  - Ridgway Scott (UChicago)
  - Rob Kirby (Texas Tech)

- **Sieve**
  - Dmitry Karpeev (ANL)
  - Peter Brune (UChicago)
  - Anders Logg (Simula)

- **PyLith**
  - Brad Aagaard (USGS)
  - Charles Williams (NZ)
Rethinking meshes produces a simple FEM interface and good code reuse.
Rethinking meshes produces a simple FEM interface and good code reuse.
Rethinking meshes produces a simple FEM interface and good code reuse.
The biggest problem in scientific computing is programmability:
- Lack of usable implementations of modern algorithms
  - Unstructured Multigrid
  - Fast Multipole Method
- Lack of comparison among classes of algorithms
  - Meshes
  - Discretizations

We should reorient thinking from
- characterizing the solution (FEM)
  - “what is the convergence rate (in $h$) of this finite element?”

to
- characterizing the computation (FErari)
  - “how many digits of accuracy per flop for this finite element?”
The biggest problem in scientific computing is programmability:

- Lack of widespread implementations of modern algorithms
  - Unstructured Multigrid
  - Fast Multipole Method
- Lack of comparison among classes of algorithms
  - Meshes
  - Discretizations

We should reorient thinking from

- characterizing the solution (FEM)
  - “what is the convergence rate (in $h$) of this finite element?”

to

- characterizing the computation (FErari)
  - “how many digits of accuracy per flop for this finite element?”
Sieve is an interface for
- general topologies
- functions over these topologies (bundles)
- traversals

One relation handles all hierarchy
- Vast reduction in complexity
  - Dimension independent code
  - A single communication routine to optimize
- Expansion of capabilities
  - Partitioning and distribution
  - Hybrid meshes
  - Complicated structures and embedded boundaries
  - Unstructured multigrid
Mesh Databases

“Most” Mesh Libraries
- Specific geometry
- Strange constraints
- Complex query model

Topological Mesh DB
- Single model
- Simple query model
- Can tune implementation

Lawler, Kalé

\(^a\) M. Knepley (UC) / KAUST
Mesh Databases

“Most” Mesh Libraries
- Specific geometry
- Strange constraints
- Complex query model

Topological Mesh DB
- Single model
- Simple query model
- Can tune implementation

\textsuperscript{a}Aagaard, Knepley, Williams

\textsuperscript{a}M. Knepley (UC)