Bridging the Gap: Harnessing advanced Computational Libraries for coupled multi-physics problems

Marc Spiegelman\textsuperscript{1,2} and Cian R. Wilson\textsuperscript{1}
Plus enormous contributions from PETSc, FEniCS and AMCG groups

\textsuperscript{1}Lamont-Doherty Earth Observatory, Columbia University, New York

\textsuperscript{2}DEES/DAPAM, Columbia University, New York

BTG, Princeton, 1–2 October 2012
Introduction

Motivation

- Computation is essential for exploring non-linear multi-physics problems in Geosciences
- Major advances in hardware, software and algorithms make complex problems more accessible.
- However, still a considerable gap between Geosciences and Computational Sciences/Math.
- Some Existing Barriers:
  - Overall Complexity of both software libraries and models
  - Lack of transparency/reproducibility/reusability of current models
  - Language/Training issues between computation and Earth Sciences
Introduction

Motivation

- Computation is essential for exploring non-linear multi-physics problems in Geosciences
- Major advances in hardware, software and algorithms make complex problems more accessible.
- However, still a considerable gap between Geosciences and Computational Sciences/Math.
- Some Existing Barriers:
  - Overall Complexity of both software libraries and models
  - Lack of transparency/reproducibility/reusability of current models
  - Language/Training issues between computation and Earth Sciences
- We need better tools for exploring complex (and simple) models and making advanced computation accessible to more general users.
Central to Solid Earth Geosciences: Essential for understanding
- Seismic, Volcanic and Tsunamigenic Natural Hazards
- Global tectonics and geochemical cycling
Target Application: multi-physics models of magmatic plate boundaries

![Cross-Section](image courtesy of the Neptune project (www.neptune.washington.edu))

- **Central to Solid Earth Geosciences:** Essential for understanding
  - Seismic, Volcanic and Tsunamagetic Natural Hazards
  - global tectonics and geochemical cycling
- **Multi-physics:** Tightly coupled non-linear systems
  - Coupled fluid-solid mechanics
  - Complex solid rheologies
  - Coupled thermodynamics/geodynamics
Target Application: multi-physics models of magmatic plate boundaries

Central to Solid Earth Geosciences: Essential for understanding
- Seismic, Volcanic and Tsunamigenic Natural Hazards
- Global tectonics and geochemical cycling

Multi-physics: Tightly coupled non-linear systems
- Coupled fluid-solid mechanics
- Complex solid rheologies
- Coupled thermodynamics/geodynamics

Considerable uncertainty in equations, constitutive relations and coupling
User Flexibility:

- Choice of equations, geometry, elements, constitutive relations, coupling
- Wide choice of solvers/preconditioners for coupled non-linear problems
- Ability to rapidly compose a range of models from simple process models to regional geodynamic models
- All choices available at or near run time
Model Requirements

**Infrastructure:**

- Residual monitoring for the *full coupled problem*
- Model reproducibility:
  - Transparent options system
  - Regression tested
- Generic model services:
  - Standardized I/O
  - Checkpointing
  - Monitoring and detectors
- Parallel, scaleable, open source, version controlled, regression tested... free

Spiegelman and Wilson

TerraFERMA

BTG 2012
## Generic Structure of PDE Based Numerical Models

<table>
<thead>
<tr>
<th>Options System</th>
<th>PDEs</th>
<th>Constitutive Eqs.</th>
<th>Meshes/Geometry</th>
<th>Discrete Eqs.</th>
<th>Solvers</th>
<th>Visualization</th>
<th>Error Est. Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Au = f</td>
<td>F(u) = 0</td>
<td>u^{n+1} = f(u^n)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PETSc, FEniCS, Gmsh, SPuD, TerraFERMA**

*Spiegelman and Wilson BTG 2012 6*
Generic Structure of PDE Based Numerical Models

\[ Au = f \]
\[ F(u) = 0 \]
\[ u^{n+1} = f(u^n) \]
Generic Structure of PDE Based Numerical Models

\[
Au = f \\
F(u) = 0 \\
\text{Visualization}
\]

\[
u^{n+1} = f(u^n)
\]

Options System
PDEs
Constitutive Eqs.
Meshes/Geometry
Discrete Eqs.
Solvers
Visualization
Error Est. Validation

PETSc
FEniCS
Gmsh
SPuD
TerraFERMA
Transparent Finite Element Rapid Model Assembler

Spiegelman and Wilson
TerraFERMA
BTG 2012
Generic Structure of PDE Based Numerical Models

\[ Au = f \]
\[ F(u) = 0 \]
\[ u^{n+1} = f(u^n) \]
Generic Structure of PDE Based Numerical Models

\[ Au = f \]
\[ F(u) = 0 \]
\[ u^{n+1} = f(u^n) \]
Generic Structure of PDE Based Numerical Models

\begin{align*}
Au &= f \\
F(u) &= 0 \\
u^{n+1} &= f(u^n)
\end{align*}
Generic Structure of PDE Based Numerical Models

\[ Au = f \]
\[ F(u) = 0 \]
\[ u^{n+1} = f(u^n) \]
Generic Structure of PDE Based Numerical Models

\[ \mathbf{A}u = f \]
\[ \mathbf{F}(u) = 0 \]
\[ u^{n+1} = f(u^n) \]
Generic Structure of PDE Based Numerical Models

\[ Au = f \]
\[ F(u) = 0 \]
\[ u^{n+1} = f(u^n) \]

**Options System**
- PETSc
- FEniCS
- Gmsh
- SPuD
- TerraFERMA

**Discrete Eqs.**
- Visualization
- Error Est. Validation

**Meshes/Geometry**

**Constitutive Eqs.**

**PDEs**

**Discrete Eqs.**

**Solvers**

**Visualization**

**Error Est. Validation**

TerraFERMA

Transparent Finite Element Rapid Model Assembler

Spiegelman and Wilson  TerraFERMA  BTG 2012
Application: Fluid Migration in Subduction Zones

\[ \nabla \cdot \left( 2\eta \left( \frac{\nabla v_s + \nabla v_s^T}{2} \right) \right) = \nabla p^* + \varphi_0 \varphi k \]

\[ \nabla \cdot v_s = \varphi_0 \frac{p}{\zeta} \]

\[ -\nabla \cdot \frac{K}{\mu} \nabla p + \frac{p}{\zeta} = -\nabla \cdot \left( \frac{K}{\mu} k - \nabla p^* \right) + \Gamma \frac{\Delta \rho}{\rho_f \varphi_0} \]

\[ \frac{D\varphi}{Dt} = (1 - \varphi_0 \varphi) \frac{p}{\zeta} + \frac{\Gamma}{\varphi_0} \]
Example: Infinite Prandtl Thermal Convection

\[-\nabla \cdot \left[ 2 \mu \left( \frac{\nabla \mathbf{v} + (\nabla \mathbf{v})^T}{2} \right) \right] + \nabla p - T k = 0,\]

\[\nabla \cdot \mathbf{v} = 0,\]

\[\frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T + \frac{1}{Ra} \nabla^2 T = 0\]
Example: Infinite Prandtl Thermal Convection

\[- \nabla \cdot \left[ 2 \mu \left( \frac{\nabla v + \nabla v^T}{2} \right) \right] + \nabla p - T k = 0, \]

\[\nabla \cdot v = 0,\]

\[\frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T + \frac{1}{\text{Ra}} \nabla^2 T = 0\]
Example: Infinite Prandtl Thermal Convection

\[-\nabla \cdot \left[ 2\mu \left( \frac{\nabla v + \nabla v^T}{2} \right) \right] + \nabla p - Tk = 0,\]

\[\nabla \cdot v = 0,\]

\[\frac{\partial T}{\partial t} + \nabla \cdot \nabla T + \frac{1}{Ra} \nabla^2 T = 0\]
Example: Infinite Prandtl Thermal Convection

\[-\nabla \cdot \left[ 2\mu \left( \frac{\nabla \mathbf{v} + \nabla \mathbf{v}^T}{2} \right) \right] + \nabla p - T k = 0,\]

\[\nabla \cdot \mathbf{v} = 0,\]

\[\frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T + \frac{1}{Ra} \nabla^2 T = 0\]

Let \( u = (\mathbf{v}, p, T) \)

\[u_{i+1} = u_i - \alpha J(u_i)^{-1}r(u_i)\]
Example: FEniCS Equation Description

Given function \( u = (v, p, T) \), test function \( u_t = (v_t, p_t, T_t) \) and trial function \( u_a = (v_a, p_a, T_a) \):

**Weak form of residual, \( r(u) \):**

\[
rv = \int_{\Omega} \left[ \left( \frac{\nabla v_t + \nabla v_t^T}{2} \right) : 2\mu \left( \frac{\nabla v + \nabla v^T}{2} \right) - \nabla \cdot v_t p - (v_t)_z T \right],
\]

\[
rf = \int_{\Omega} p_t \nabla \cdot v,
\]

\[
rT = \int_{\Omega} \left[ T_t ((T - T_n) + \Delta t v_\theta \cdot \nabla T_\theta) + \frac{\Delta t}{Ra} \nabla T_t \cdot \nabla T_\theta \right]
\]

\[
r = rv + rp + rT
\]

**Weak form of Jacobian, \( J(u) \):**

\[
J = r'(u)
\]
Example: FEniCS Equation Description

Given function \( u = (v, p, T) \), test function \( u_t = (v_t, p_t, T_t) \) and trial function \( u_a = (v_a, p_a, T_a) \):

**Weak form of residual, \( r(u) \):**

\[
\begin{align*}
    r_v &= \left( \text{inner}(\text{sym}(\text{grad}(v_t)), 2.*\mu*\text{sym}(\text{grad}(v))) - \text{div}(v_t)*p - T*v_t[1] \right) dx \\
    r_p &= p_t*\text{div}(v)*dx \\
    r_T &= (T_t*((T - T_n) + dt*\text{inner}(v_{\theta}, \text{grad}(T_{\theta}))) + (dt/Ra)*\text{inner}(\text{grad}(T_t), \text{grad}(T_{\theta}))) dx \\
    r &= r_v + r_p + r_T
\end{align*}
\]

**Weak form of Jacobian, \( J(u) \):**

\[
J = \text{derivative}(r, u, u_a)
\]
$J(u_i)\delta u = -r(u_i)$

$$J = \begin{pmatrix}
K_{11} & K_{12} & G_1 & C_1 \\
K_{21} & K_{22} & G_2 & C_2 \\
G_1^T & G_2^T & 0 & 0 \\
B_1 & B_2 & 0 & A
\end{pmatrix}$$
Example: PETSc Block Preconditioning & Solvers

\[ J(u_i)\delta u = -r(u_i) \]

\[
J = \begin{pmatrix}
K_{11} & K_{12} & G_1 \\
K_{21} & K_{22} & G_2 \\
G_1^T & G_2^T & 0 \\
B_1 & B_2 & 0
\end{pmatrix}
\begin{pmatrix}
C_1 \\
C_2 \\
0 \\
A
\end{pmatrix}
\]

Stokes
Advection-Diffusion
Example: PETSc Block Preconditioning & Solvers

\[ \tilde{J}_{PC}(u_i)^{-1} J(u_i) \delta u = -\tilde{J}_{PC}(u_i)^{-1} r(u_i) \]

\[ J = \begin{pmatrix} K_{11} & K_{12} & G_1 & C_1 \\ K_{21} & K_{22} & G_2 & C_2 \\ G_1^T & G_2^T & 0 & 0 \\ B_1 & B_2 & 0 & A \end{pmatrix} \]

\[ J_{PC} = \begin{pmatrix} K_{11} & K_{12} & G_1 & C_1 \\ K_{21} & K_{22} & G_2 & C_2 \\ G_1^T & G_2^T & M & 0 \\ B_1 & B_2 & 0 & A \end{pmatrix} \]
Example: PETSc Block Preconditioning & Solvers

\[ \tilde{J}_{PC}(u_i)^{-1} J(u_i) \delta u = -\tilde{J}_{PC}(u_i)^{-1} r(u_i) \]

\[
J = \begin{pmatrix}
K_{11} & K_{12} & G_1 & C_1 \\
K_{21} & K_{22} & G_2 & C_2 \\
G_1^T & G_2^T & 0 & 0 \\
B_1 & B_2 & 0 & A
\end{pmatrix}
\]

\[
J_{PC} = \begin{pmatrix}
K_{11} & K_{12} & G_1 & C_1 \\
K_{21} & K_{22} & G_2 & C_2 \\
G_1^T & G_2^T & M & 0 \\
B_1 & B_2 & 0 & A
\end{pmatrix}
\]

System (fgmres, fieldsplit)
\[ \tilde{J}_{PC}(u_i)^{-1} J(u_i) \delta u = -\tilde{J}_{PC}(u_i)^{-1} r(u_i) \]

\[ J = \begin{pmatrix}
    K_{11} & K_{12} & G_1 & C_1 \\
    K_{21} & K_{22} & G_2 & C_2 \\
    G_1^T & G_2^T & 0 & 0 \\
    B_1 & B_2 & 0 & A \\
\end{pmatrix} \]

\[ J_{PC} = \begin{pmatrix}
    K_{11} & K_{12} & G_1 & C_1 \\
    K_{21} & K_{22} & G_2 & C_2 \\
    G_1^T & G_2^T & M & 0 \\
    B_1 & B_2 & 0 & A \\
\end{pmatrix} \]

System (fgmres, fieldsplit)

Temperature (gmres, ilu)  Stokes (fgmres, fieldsplit)
Example: PETSc Block Preconditioning & Solvers

\[
\tilde{J}_{PC}(u_i)^{-1} J(u_i) \delta u = -\tilde{J}_{PC}(u_i)^{-1} r(u_i)
\]

\[
J = \begin{pmatrix}
  K_{11} & K_{12} & G_1 & C_1 \\
  K_{21} & K_{22} & G_2 & C_2 \\
  G_1^T & G_2^T & 0 & 0 \\
  B_1 & B_2 & 0 & A \\
\end{pmatrix}
\]

\[
J_{PC} = \begin{pmatrix}
  K_{11} & K_{12} & G_1 & C_1 \\
  K_{21} & K_{22} & G_2 & C_2 \\
  G_1^T & G_2^T & M & 0 \\
  B_1 & B_2 & 0 & A \\
\end{pmatrix}
\]

System (fgmres, fieldsplit)

Temperature (gmres, ilu)

Stokes (fgmres, fieldsplit)

Pressure (cg,sor)

Velocity (preonly, fieldsplit)
Example: PETSc Block Preconditioning & Solvers

\[ \tilde{J}_{PC}(u_i)^{-1} J(u_i) \delta u = -\tilde{J}_{PC}(u_i)^{-1} r(u_i) \]

\[ J = \begin{pmatrix} K_{11} & K_{12} & G_1 & C_1 \\ K_{21} & K_{22} & G_2 & C_2 \\ G_1^T & G_2^T & 0 & 0 \\ B_1 & B_2 & 0 & A \end{pmatrix} \]

\[ J_{PC} = \begin{pmatrix} K_{11} & K_{12} & G_1 & C_1 \\ K_{21} & K_{22} & G_2 & C_2 \\ G_1^T & G_2^T & M & 0 \\ B_1 & B_2 & 0 & A \end{pmatrix} \]
Generic Structure of PDE Based Numerical Models

\[ Au = f \]
\[ F(u) = 0 \]
\[ u^{n+1} = f(u^n) \]

TerraFERMA

Transparent Finite Element Rapid Model Assembler
Example: TerraFERMA Options

Node
- bucket_options
  - geometry
  - io
  - timestepping
    - nonlinear_systems
- global_parameters
- system (Convection)
  - system

Option Properties

Description
Options describing a system.
A system consists of a DOLFIN functionspace, the fields on this functionspace, the forms describing the solvers and preconditioners that act on that functionspace and the coefficients that appear in those forms.
The system name must be unique amongst any other systems.

Attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>Convection</td>
</tr>
</tbody>
</table>

Data
No data

Comment
No comment

---

Spiegelman and Wilson  TerraFERMA  BTG 2012  15
Example: TerraFERMA Options

Option Properties

Description
Options describing a system.
A system consists of a DOLFIN functionspace, the fields on this functionspace, the forms describing the solvers and preconditioners that act on that functionspace and the coefficients that appear in those forms.
The system name must be unique amongst any other systems.

Attributes

Name | Value
name | Convection

Data
No data

Comment
No comment
Example: Setting Equations in TerraFERMA

```plaintext
1 b = 6.9077552789821368
2 mu = exp(-b*T_i)
3
4 rv = (inner(sym(grad(v_t)), 2.*mu*sym(grad(v_i))) - div(v_t)*p_i - T_i*v_t[1]) *dx
5
6 rp = p_t*div(v_i)*dx
7 rT = (T_t*((T_i - T_n) + dt*inner(v_theta, grad(T_theta))) + (dt/Ra)*inner(grad(T_t), grad(T_theta))) *dx
8
10 r = rv + rp + rT
```

Option Properties

- **Description**
  - UFL code form describing a linear residual form (must return a linear form). Any system, field or coefficient UFL symbols defined in this options file may be used in this form as well as any symbols defined in the preamble above.

Data

```
1 b = 6.9077552789821368
2 mu = exp(-b*T_i)
3
4 rv = (inner(sym(grad(v_t)), 2.*mu*sym(grad(v_i))) - div(v_t)*p_i - T_i*v_t[1]) *dx
5
6 rp = p_t*div(v_i)*dx
7 rT = (T_t*((T_i - T_n) + dt*inner(v_theta, grad(T_theta))) + (dt/Ra)*inner(grad(T_t), grad(T_theta))) *dx
8
10 r = rv + rp + rT
```
Example: Setting Equations in TerraFERMA

**Option Properties**

**Description**

ufl code form describing a jacobian bilinear form (must return a bilinear form). Any system, field or coefficient ufl symbols defined in this options file may be used in this form as well as any symbols defined in the preamble and residual linear form above.

**Data**

```
a = derivative(r, u_i, u_a)
```
Example: Solver Options in TerraFERMA

- **Temperature**
  - iterative_method (gmres)
  - preconditioner (ilu)

- **Stokes**
  - iterative_method (fgmres)
  - preconditioner (fieldsplit)

- **Pressure**
  - cg, sor

- **Velocity**
  - preonly, fieldsplit

System (fgmres, fieldsplit)
Example: Solver Options in TerraFERMA

- System (fgmres, fieldsplit)
  - Temperature (gmres, ilu)
  - Stokes (fgmres, fieldsplit)
Example: Solver Options in TerraFERMA

System (fgmres, fieldsplit)

- Temperature (gmres, ilu)
- Stokes (fgmres, fieldsplit)

- Pressure (cg,sor)
- Velocity (preonly,fieldsplit)
<table>
<thead>
<tr>
<th>Ra = 10^4</th>
<th>( \text{isoviscous} )</th>
<th>( 32 \times 32 )</th>
<th>( 64 \times 64 )</th>
<th>( 128 \times 128 )</th>
<th>( \text{Benchmark} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nu</td>
<td>4.887</td>
<td>4.885</td>
<td>4.885</td>
<td>4.884</td>
<td></td>
</tr>
<tr>
<td>( v_{rms} )</td>
<td>42.865</td>
<td>42.865</td>
<td>42.865</td>
<td>42.865</td>
<td></td>
</tr>
<tr>
<td>( q_1 )</td>
<td>8.060</td>
<td>8.059</td>
<td>8.059</td>
<td>8.059</td>
<td></td>
</tr>
<tr>
<td>( q_2 )</td>
<td>0.589</td>
<td>0.589</td>
<td>0.589</td>
<td>0.589</td>
<td></td>
</tr>
<tr>
<td>( T_e )</td>
<td>0.422</td>
<td>0.422</td>
<td>0.422</td>
<td>0.422</td>
<td></td>
</tr>
<tr>
<td>( z_e )</td>
<td>0.226</td>
<td>0.226</td>
<td>0.225</td>
<td>0.225</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ra = 10^5</th>
<th>( \text{isoviscous} )</th>
<th>( 32 \times 32 )</th>
<th>( 64 \times 64 )</th>
<th>( 128 \times 128 )</th>
<th>( \text{Benchmark} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nu</td>
<td>10.539</td>
<td>10.535</td>
<td>10.534</td>
<td>10.534</td>
<td></td>
</tr>
<tr>
<td>( v_{rms} )</td>
<td>193.222</td>
<td>193.215</td>
<td>193.215</td>
<td>193.214</td>
<td></td>
</tr>
<tr>
<td>( q_1 )</td>
<td>19.081</td>
<td>19.080</td>
<td>19.080</td>
<td>19.079</td>
<td></td>
</tr>
<tr>
<td>( q_2 )</td>
<td>0.722</td>
<td>0.723</td>
<td>0.723</td>
<td>0.723</td>
<td></td>
</tr>
<tr>
<td>( T_e )</td>
<td>0.428</td>
<td>0.428</td>
<td>0.428</td>
<td>0.428</td>
<td></td>
</tr>
<tr>
<td>( z_e )</td>
<td>0.114</td>
<td>0.111</td>
<td>0.112</td>
<td>0.112</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ra = 10^6</th>
<th>( \text{isoviscous} )</th>
<th>( 32 \times 32 )</th>
<th>( 64 \times 64 )</th>
<th>( 128 \times 128 )</th>
<th>( \text{Benchmark} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_{rms} )</td>
<td>834.024</td>
<td>833.990</td>
<td>833.989</td>
<td>833.990</td>
<td></td>
</tr>
<tr>
<td>( q_1 )</td>
<td>46.008</td>
<td>45.972</td>
<td>45.967</td>
<td>45.964</td>
<td></td>
</tr>
<tr>
<td>( q_2 )</td>
<td>0.877</td>
<td>0.877</td>
<td>0.877</td>
<td>0.877</td>
<td></td>
</tr>
<tr>
<td>( T_e )</td>
<td>0.432</td>
<td>0.432</td>
<td>0.432</td>
<td>0.432</td>
<td></td>
</tr>
<tr>
<td>( z_e )</td>
<td>0.059</td>
<td>0.058</td>
<td>0.058</td>
<td>0.058</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ra = 10^4</th>
<th>( T\text{-dep viscosity} )</th>
<th>( 32 \times 32 )</th>
<th>( 64 \times 64 )</th>
<th>( 128 \times 128 )</th>
<th>( \text{Benchmark} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nu</td>
<td>10.069</td>
<td>10.066</td>
<td>10.064</td>
<td>10.066</td>
<td></td>
</tr>
<tr>
<td>( v_{rms} )</td>
<td>479.951</td>
<td>480.385</td>
<td>480.257</td>
<td>480.433</td>
<td></td>
</tr>
<tr>
<td>( q_1 )</td>
<td>17.533</td>
<td>17.531</td>
<td>17.528</td>
<td>17.531</td>
<td></td>
</tr>
<tr>
<td>( q_2 )</td>
<td>1.007</td>
<td>1.008</td>
<td>1.008</td>
<td>1.009</td>
<td></td>
</tr>
<tr>
<td>( T_e )</td>
<td>0.739</td>
<td>0.740</td>
<td>0.740</td>
<td>0.741</td>
<td></td>
</tr>
<tr>
<td>( z_e )</td>
<td>0.062</td>
<td>0.063</td>
<td>0.063</td>
<td>0.062</td>
<td></td>
</tr>
</tbody>
</table>
Current Benchmarks

- Incompressible 2D Convection (Blankenbach et al, 1989)
- Incompressible 3D Convection (Busse et al., 1994)
- Cylindrical Laminar Plumes (Vatteville et al., 2009)
- Compressible 2D Convection (King et al., 2009)
- Kinematic Subduction Zones (van Keken et al., 2008)
- Linearized Free Surface Evolution (Kramer et al., 2012)
- Non-linear magma waves (Simpson and Spiegelman, 2011)
- Cylindrical Convection (Rhodri Davies: rhodri.davies@imperial.ac.uk)
Fluid Migration

Buoyancy

Buoyancy and Pore Pressure

Time: 30

Spiegelman and Wilson

TerraFERMA

BTG 2012 23
Fluid Migration

Buoyancy

Buoyancy and Pore Pressure

Time: 302

Spiegelman and Wilson

TerraFERMA

BTG 2012
Fluid Migration

Buoyancy

Buoyancy and Pore Pressure

Time: 751

T (C)

Porosity

0.02

0.01

0.001

0.0001

2e-5
Fluid Migration

Buoyancy

Buoyancy and Pore Pressure

Time: 1440

Time: 1440
Fluid Migration

Buoyancy

Buoyancy and Pore Pressure

Time: 2400

T (°C)
1300
1250
1000
750
500
250
100

Porosity
0.02
0.01
0.001
2e-5
Conclusions

- Many important problems in Earth Sciences can be described by non-linear coupled systems of partial differential equations.
- Much of the complexity modelling these systems stems from the nature of multi-physics where small changes in the coupling between fields or constitutive relations can lead to radical changes in behavior and the resulting demands on discretizations and solvers.
- Many established "single-physics” models have locked in solution strategies that are unsuited or difficult to extend to the more challenging non-linear behaviour of a multi-physics system.
- We require a significant increase in flexibility for users for both composing problems (from simple model problems to more ‘realistic’ coupled simulations) and changing solver strategies.
- Several advanced, open-source computational libraries mean that it is now possible to provide much greater flexibility to the user.
- TerraFERMA is a model building framework that gives access to these libraries while also providing a transparent description of the model.
Equations (Katz et al., 2007)

Solid flow for solid velocity, \(v_s\), and dynamic pressure, \(p^*\):

\[
\nabla \cdot \left( 2 \eta \left( \frac{\nabla v_s + \nabla v_s^T}{2} \right) \right) = \nabla p^* + \varphi_0 \varphi \ k
\]

\[
\nabla \cdot v_s = \varphi_0 \frac{\mathcal{P}}{\zeta}
\]

Fluid flow for pore pressure, \(\mathcal{P}\), and porosity, \(\varphi\):

\[
-\nabla \cdot \left( \frac{K}{\mu} \nabla \mathcal{P} + \frac{\mathcal{P}}{\zeta} \right) = -\nabla \cdot \left( \frac{K}{\mu} k - \nabla p^* \right) + \Gamma \frac{\Delta \rho}{\rho_f \varphi_0}
\]

\[
\frac{D\varphi}{Dt} = \left(1 - \varphi_0 \varphi\right) \frac{\mathcal{P}}{\zeta} + \frac{\Gamma}{\varphi_0}
\]

Constitutive relations:

\[
\eta = \eta(T, \dot{\varepsilon}, \varphi) , \quad \zeta = \eta \varphi^{-m}, \quad K = \varphi^n
\]
Equations (Katz et al., 2007)

Solid flow for solid velocity, \( \mathbf{v}_s \), and dynamic pressure, \( p^* \):

\[
\nabla \cdot \left( 2 \eta \left( \frac{\nabla \mathbf{v}_s + \nabla \mathbf{v}_s^T}{2} \right) \right) = \nabla p^* + \varphi_0 \varphi \textbf{k} \\
\n\nabla \cdot \mathbf{v}_s = \varphi_0 \frac{\mathcal{P}}{\zeta} 
\]

Fluid flow for pore pressure, \( \mathcal{P} \), and porosity, \( \varphi \):

\[
- \nabla \cdot \frac{K}{\mu} \nabla \mathcal{P} + \frac{\mathcal{P}}{\zeta} = - \nabla \cdot \left( \frac{K}{\mu} \textbf{k} - \nabla p^* \right) + \Gamma \frac{\Delta \rho}{\rho_f \varphi_0} \\
D \frac{\varphi}{Dt} = \left( 1 - \varphi_0 \varphi \right) \frac{\mathcal{P}}{\zeta} + \frac{\Gamma}{\varphi_0} 
\]

Constitutive relations:

\[
\eta = \eta(T, \dot{\varepsilon}, \varphi), \quad \zeta = \eta \varphi^{-m}, \quad K = \varphi^n 
\]
Equations (Katz et al., 2007)

Solid flow for solid velocity, $\mathbf{v}_s$, and dynamic pressure, $p^*$:

$$\nabla \cdot \left( 2\eta \left( \frac{\nabla \mathbf{v}_s + \nabla \mathbf{v}_s^T}{2} \right) \right) = \nabla p^* + \varphi_0 \varphi \mathbf{k}$$

$$\nabla \cdot \mathbf{v}_s = \varphi_0 \frac{\mathcal{P}}{\zeta}$$

Fluid flow for pore pressure, $\mathcal{P}$, and porosity, $\varphi$:

$$-\nabla \cdot \left( \frac{K}{\mu} \nabla \mathcal{P} + \frac{\mathcal{P}}{\zeta} \right) = -\nabla \cdot \left( \frac{K}{\mu} \mathbf{k} - \nabla p^* \right) + \Gamma \frac{\Delta \rho}{\rho_f \varphi_0}$$

$$\frac{D \varphi}{Dt} = \left( 1 - \varphi_0 \varphi \right) \frac{\mathcal{P}}{\zeta} + \frac{\Gamma}{\varphi_0}$$

Constitutive relations:

$$\eta = \eta(T, \dot{\epsilon}, \varphi), \quad \zeta = \eta \varphi^{-m}, \quad K = \varphi^n$$
Equations (Katz et al., 2007)

**Solid flow for solid velocity, \( \mathbf{v}_s \), and dynamic pressure, \( p^* \):**

\[
\nabla \cdot \left( 2 \eta \left( \frac{\nabla \mathbf{v}_s + \nabla \mathbf{v}_s^T}{2} \right) \right) = \nabla p^* + \varphi_0 \varphi \mathbf{k}
\]

\[
\nabla \cdot \mathbf{v}_s = \varphi_0 \frac{\mathbf{P}}{\zeta}
\]

**Fluid flow for pore pressure, \( \mathbf{P} \), and porosity, \( \varphi \):**

\[
-\nabla \cdot \frac{K}{\mu} \nabla \mathbf{P} + \frac{\mathbf{P}}{\zeta} = -\nabla \cdot \left( \frac{K}{\mu} \mathbf{k} - \nabla p^* \right) + \Gamma \frac{\Delta \rho}{\rho_f \varphi_0}
\]

\[
\frac{D \varphi}{Dt} = \left( 1 - \varphi_0 \varphi \right) \frac{\mathbf{P}}{\zeta} + \frac{\Gamma}{\varphi_0}
\]

**Constitutive relations:**

\[
\eta = \eta(T, \dot{\varepsilon}, \varphi) , \quad \zeta = \eta \varphi^{-m} , \quad K = \varphi^n
\]
Equations (Katz et al., 2007)

Solid flow for solid velocity, \(v_s\), and dynamic pressure, \(p^*\):

\[
\nabla \cdot \left( 2 \eta \left( \frac{\nabla v_s + \nabla v_s^T}{2} \right) \right) = \nabla p^* + \varphi_0 \varphi k
\]

\[
\nabla \cdot v_s = \varphi_0 \frac{\mathcal{P}}{\zeta}
\]

Fluid flow for pore pressure, \(\mathcal{P}\), and porosity, \(\varphi\):

\[
- \nabla \cdot \frac{K}{\mu} \nabla \mathcal{P} + \frac{\mathcal{P}}{\zeta} = - \nabla \cdot \left( \frac{K}{\mu} k - \nabla p^* \right) + \Gamma \frac{\Delta \rho}{\rho_f \varphi_0}
\]

\[
\frac{D\varphi}{Dt} = \left( 1 - \varphi_0 \varphi \right) \frac{\mathcal{P}}{\zeta} + \frac{\Gamma}{\varphi_0}
\]

Constitutive relations:

\[
\eta = \eta(T, \dot{\varepsilon}, \varphi), \quad \zeta = \eta \varphi^{-m}, \quad K = \varphi^n
\]
Equations (Katz et al., 2007)

Solid flow for solid velocity, \( \mathbf{v}_s \), and dynamic pressure, \( p^* \):

\[
\nabla \cdot \left( 2 \eta \left( \frac{\nabla \mathbf{v}_s + \nabla \mathbf{v}_s^T}{2} \right) \right) = \nabla p^* \\
\nabla \cdot \mathbf{v}_s = 0
\]

Fluid flow for pore pressure, \( P \), and porosity, \( \varphi \):

\[
-\nabla \cdot \frac{K}{\mu} \nabla P + \frac{P}{\zeta} = -\nabla \cdot \frac{K}{\mu} \mathbf{k} + \Gamma \frac{\Delta \rho}{\rho_f \varphi_0} \\

\frac{D \varphi}{Dt} = \frac{P}{\zeta} + \frac{\Gamma}{\varphi_0}
\]

Constitutive relations:

\[
\eta = \eta (T, \dot{\epsilon}, \varphi), \quad \zeta = \eta \varphi^{-m}, \quad K = \varphi^n
\]
Solid Flow

\[ T = T_s \]

\[ T = -zT_0/50 \]

\[ v_i = 0 \]

\[ |v| = v_0 \]

\[ T = T_0 \]

\[ T = T_s + (T_0 - T_s) \text{erf}(-z/\sqrt{\kappa t\text{age}}) \]

Velocity streamlines
Equations

Buoyancy and Pore Pressure:

\[-\nabla \cdot \frac{K}{\mu} \nabla P + \frac{P}{\zeta} = -\nabla \cdot \frac{K}{\mu} k + \Gamma \frac{\Delta \rho}{\rho_f \varphi_0}\]

\[\frac{D \varphi}{Dt} = \frac{P}{\zeta} + \frac{\Gamma}{\varphi_0}\]

Constitutive relations:

\[\eta = \eta(T, \dot{\epsilon}, \varphi), \quad \zeta = \eta \varphi^{-m}, \quad K = \varphi^n\]
Equations

Buoyancy (zero compaction length approximation):

\[
\frac{D\varphi}{Dt} = -\nabla \cdot \frac{K}{\mu} k + \frac{\Gamma}{\varphi_0} \left( \frac{\Delta \rho}{\rho_f} + 1 \right)
\]

Buoyancy and Pore Pressure:

\[
-\nabla \cdot \frac{K}{\mu} \nabla P + \frac{P}{\zeta} = -\nabla \cdot \frac{K}{\mu} k + \Gamma \frac{\Delta \rho}{\rho_f \varphi_0}
\]

\[
\frac{D\varphi}{Dt} = \frac{P}{\zeta} + \frac{\Gamma}{\varphi_0}
\]

Constitutive relations:

\[
\eta = \eta(T, \dot{\epsilon}, \varphi), \quad \zeta = \eta \varphi^{-m}, \quad K = \varphi^n
\]