A Finite Volume Approach to Multiscale Elasticity

Paul Delgado
NSF Fellow (HRD-1139929)
Doctoral Candidate - Computational Science
University of Texas at El Paso

November 1, 2014

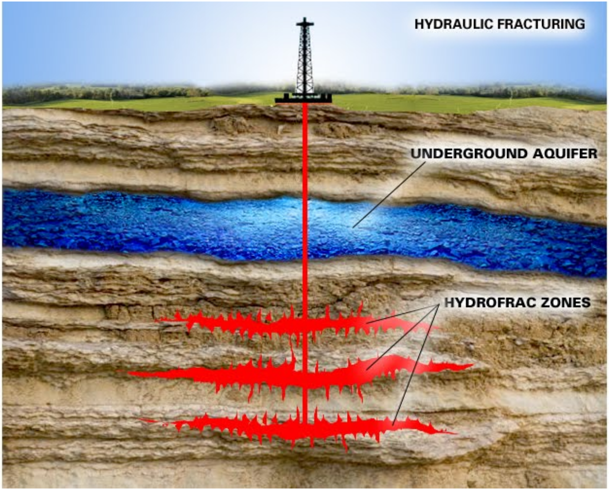
Inspiration

William Kamkwamba, South Africa

Definition

Poroelasticity

Applications



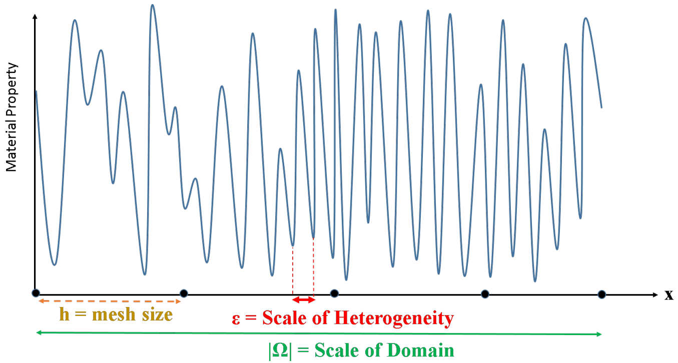
Fluid flow affects solid deformation!

The Challenge

Large variations in material parameters over small spatial scales.

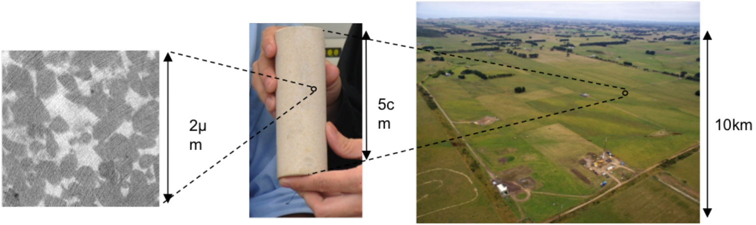
The Goldilocks Problem

Assume $\epsilon << |\Omega|$



- If $h > \epsilon$, then simulation is fast, but highly inaccurate.
- If $h < \epsilon$, then simulation is accurate, but extremely slow.

The Curse of Dimensionality

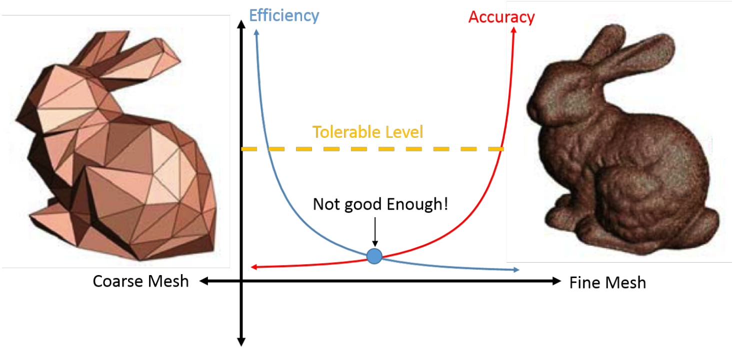


Assuming 10^3 nodes per μm, a Petascale computer solves the equations in

- In 2D $\Rightarrow \approx 3,000$ yrs
- In 3D $\Rightarrow \approx 31$ quadrillion yrs

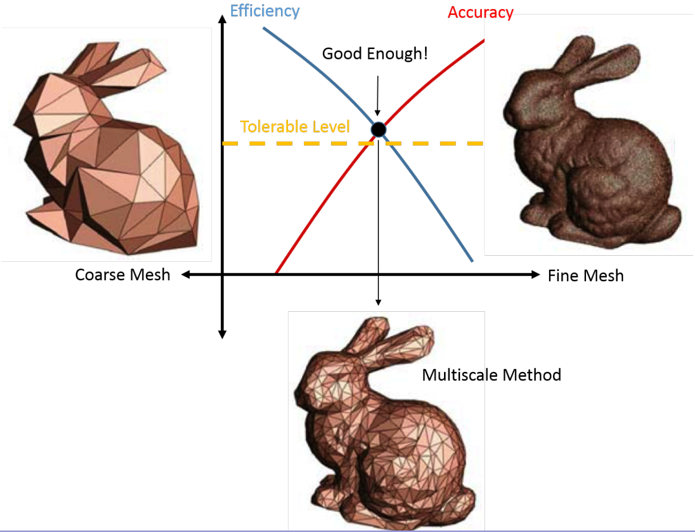
Moral: Parallelization, alone, will not solve this problem!!!

Conventional Methods



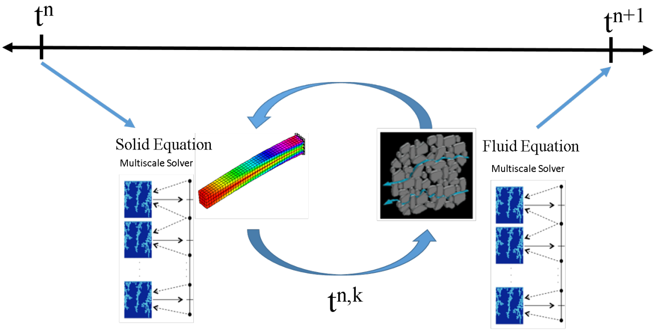
How can we balance the need for accuracy with the need for efficiency?

Multiscale Method



Our approach

Toward a multiscale method for poroelasticity



- Decouple solid & fluid equations
- Develop multiscale methods for each equation

Progress

- Developed & Verified Operator Splitting Method
- Developed 1D Multiscale Flow & Deformation Methods
- Improved methods for neumann conditions & source terms
- **Higher Dimensional method for Fluid Flow**

Today, we demonstrate our multiscale method for the solid equation in higher dimensions

Solid Equation

$$-\nabla \cdot \sigma = \vec{F} \text{ in } \Omega$$

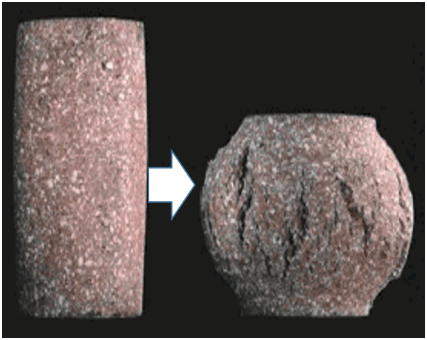
$$\sigma = \sigma(\epsilon)$$

$$\epsilon = \epsilon(\nabla \vec{u})$$

$$\nabla \vec{u} = \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix}$$

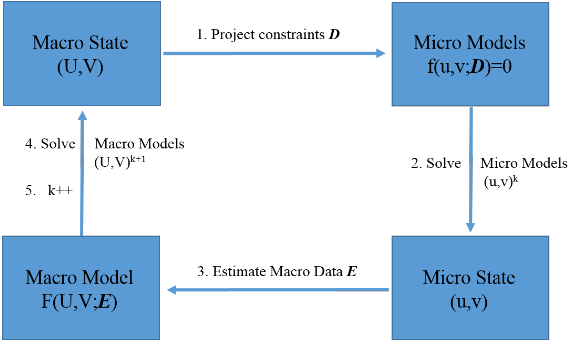
$$u = d(x, y) \text{ on } \partial\Omega_d$$

$$\sigma \cdot n = t(x, y) \text{ on } \partial\Omega_t$$



Momentum balance relates stress to displacement

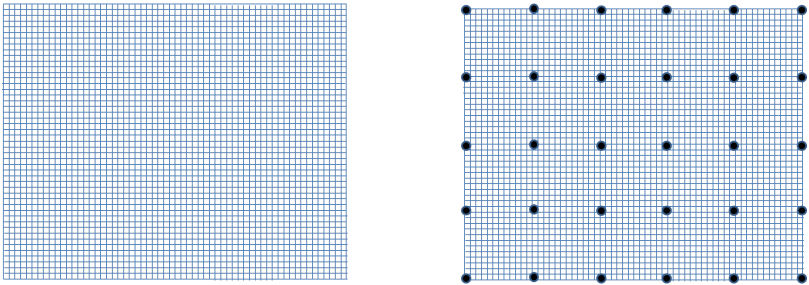
Methodology



Heterogeneous Multiscale Framework (E & Engquist 2003).

Key Idea

Microgrid　　　　Macrogrid



- A fully coupled microscopic model on the entire computational domain Ω
- Seek a solution at a small subset of the microgrid.
- **Key to Efficiency:** Use less info than what is available!

Macro Model

Incomplete Finite Volume Method

$$-\int_{\partial CV} \sigma \cdot \vec{n} = \int_{CV} \vec{F}$$

$$-\int_{CV^E} \sigma_x + \int_{CV^W} \sigma_x - \int_{CV^N} \tau_{xy} + \int_{CV^S} \tau_{xy} = \int_{CV} f \quad (1)$$

$$-\int_{CV^N} \sigma_y + \int_{CV^S} \sigma_y - \int_{CV^E} \tau_{xy} + \int_{CV^W} \tau_{xy} = \int_{CV} g \quad (2)$$

- $\sigma = \begin{bmatrix} \sigma_x & \tau_{x_y} \\ \tau_{xy} & \sigma_y \end{bmatrix}$
- No explicit constitutive relation $\sigma = \sigma(\epsilon(\nabla \vec{u}))$

Micro Model

Linear Heterogeneous Isotropic Model

$$\nabla \cdot \sigma + \vec{F} = 0$$

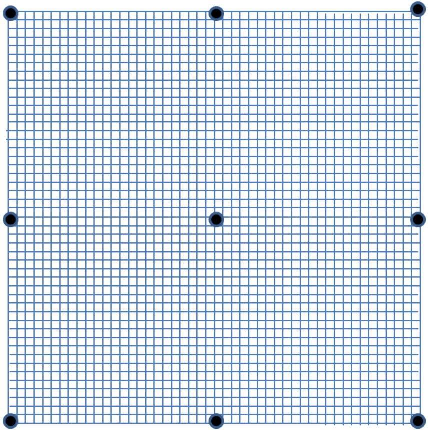
$$\sigma(\epsilon) = \begin{bmatrix} \sigma_x & \tau_{xy} \\ \tau_{xy} & \sigma_y \end{bmatrix} = 2\mu(\vec{x})\epsilon + \lambda(\vec{x}) tr(\epsilon) I$$

$$\epsilon(\nabla \vec{u}) = \frac{1}{2}\left(\nabla \vec{u} + \nabla \vec{u}^T\right)$$

$$\nabla \vec{u} = \begin{bmatrix} u_x & u_y \\ v_x & v_y \end{bmatrix}$$

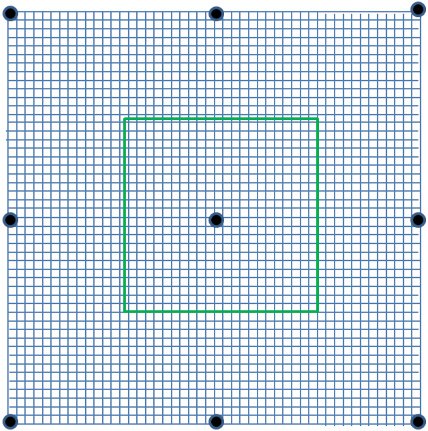
Other models are also possible (molecular dynamics, lattice structures, etc...)

Step 1: Initial Guess



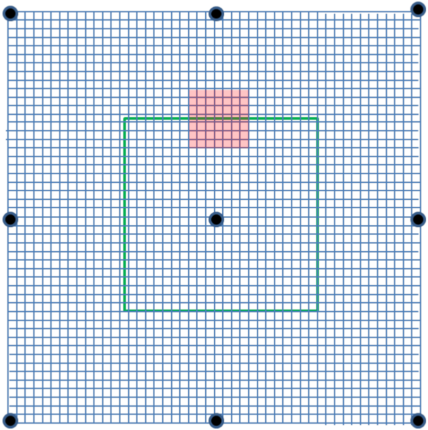
Old field variables $(u_{ij}, v_{ij})^K$

Step 2: Loop



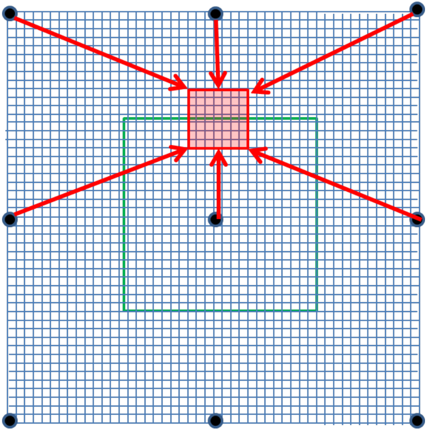
For each control volume boundary *D*

Step 3: Sample



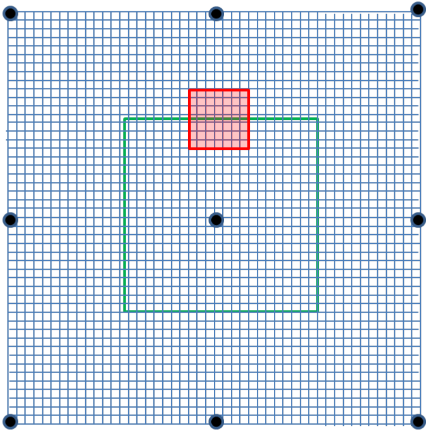
Micro data near CV^D boundary midpoint

Step 4: Constraint Projection



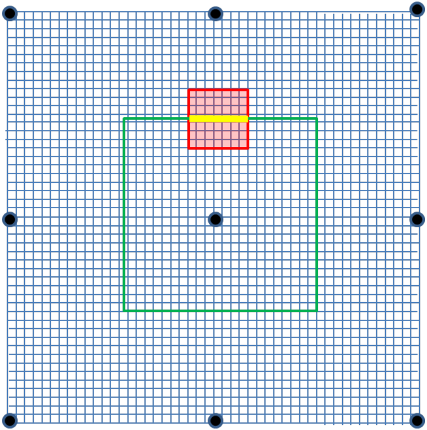
Interpolate BC's from local macro field $(u_{ij}, v_{ij})^K$

Step 5: Solve Micromodel



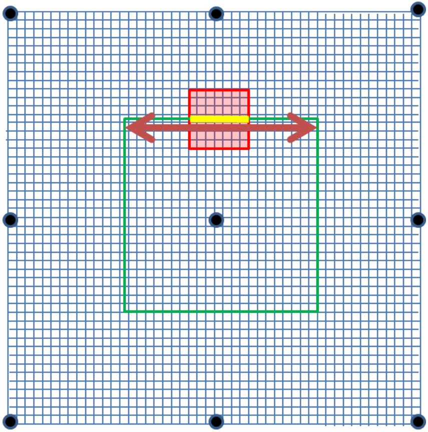
Obtain local micro field in B_δ^D

Step 6: Data Estimation (1)



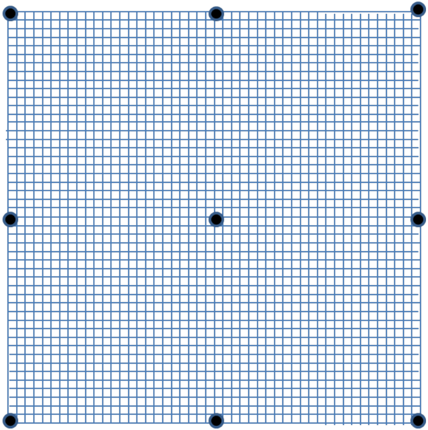
Calculate total normal & shear force along mid cross-section.

Step 6: Data Estimation (2)



Rescale total forces to entire control volume boundary *D*

Step 7: Solve Macro Model



Obtain updated field variables $(u_{ij}, v_{ij})^{K+1}$

Key to Micro-Macro Iterations

- Assume $\sigma = 0$ when $\nabla \vec{u} = 0$.
- Assume u_x, u_y, v_x and v_y are *independent variables*.
- Taylor series expansion of σ_y, σ_x, and τ_{xy}
- Fixed Point Iteration over K

Then

$$\int_{\partial CV^D} \nu^{K+1} = \sum_{i=1}^{4} \frac{\int_{\partial CV^D} \nu^{D,K}\left(G_i^{D,K}\right)}{G_i^{D,K} \cdot e_i} G_i^{D,K+1} \cdot e_i \tag{3}$$

- **Stress Component:** $\nu = \sigma_y, \sigma_x$, and τ_{xy}
- **Boundary:** $D = N, S, E, W$
- **Subgradient:**[1] $G_i^{D,K} \equiv vec\left(\nabla \vec{u}^{D,K}\right) \circ e_i$ (i=1,...,4)

---

[1] $(\circ)$denotes the **Hadamard Product** and e_i denotes **standard basis** in $\mathbb{R}^4$

Numerical Experiments

Unit Square Domain $\Omega = [0, 1]^2$
Cases w/ Analytical Solutions

- Prescribed displacement u, v functions
- Smooth material functions λ, μ
- Derived source terms f, g

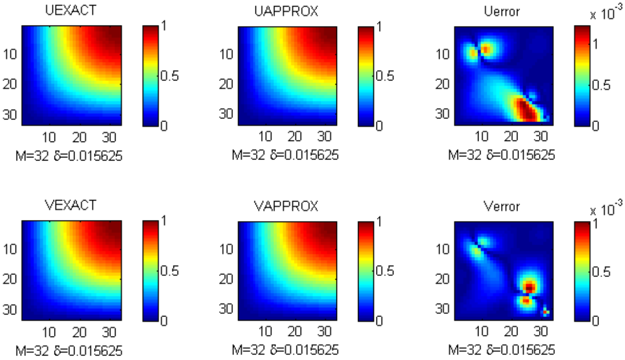
Cases w/o Analytical Solutions

- Random material parameters λ, μ
- Prescribed source terms f, g
- Reference Solution obtained numerically

Analyze convergence as the total sampling area $\to |\Omega|$

Results - Analytical Case

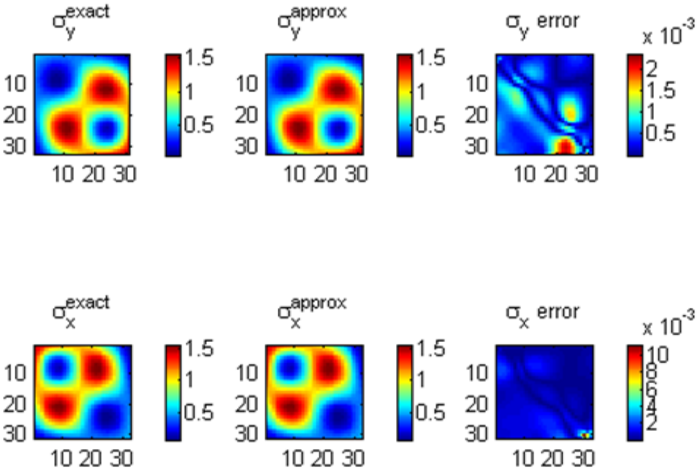
Displacement



$$u = v = sin(\frac{\pi x}{2})sin(\frac{\pi y}{2}),\ \lambda = \mu = 11 + sin(2\pi x)sin(2\pi y)$$

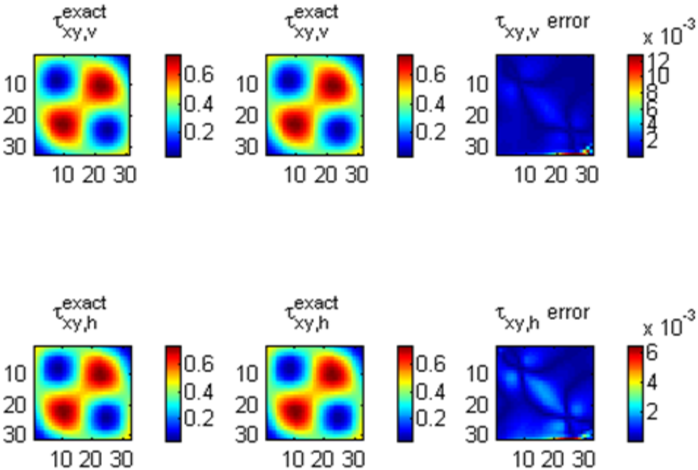
Results - Analytical Case

Normal Stress



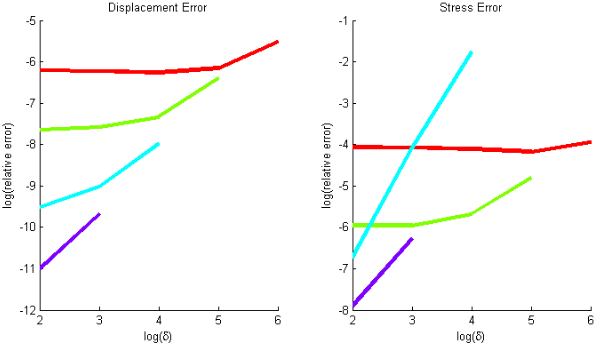
Results - Analytical Case

Shear Stress



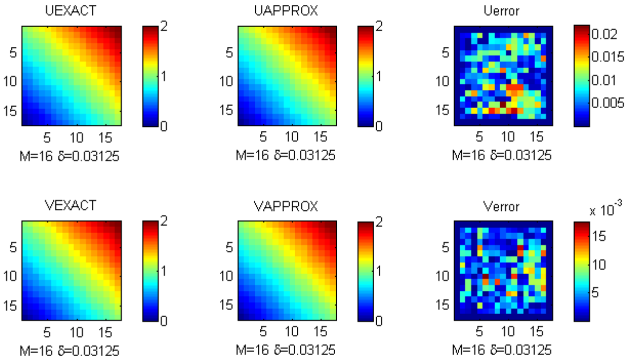
Results - Analytical Case

Convergence



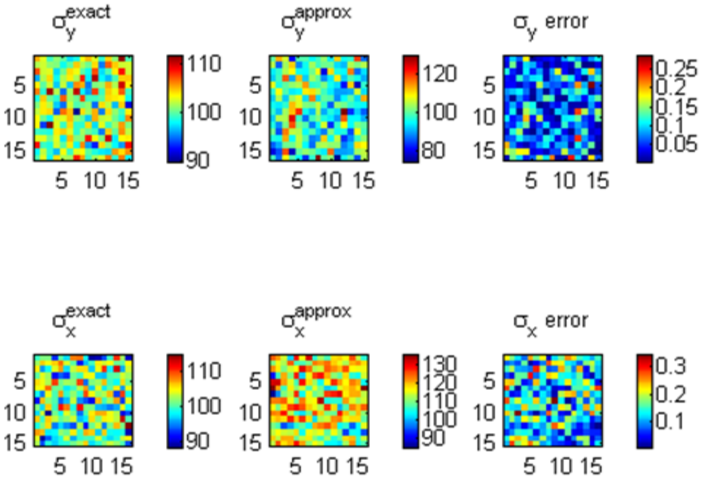
Results - Random Case

Displacement



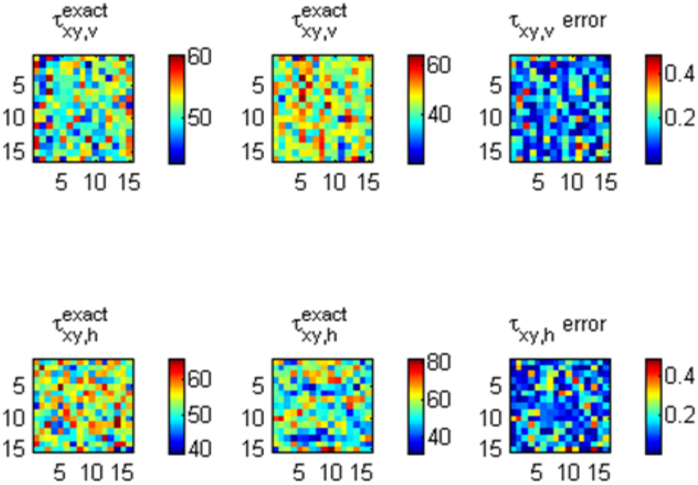
Results - Random Case

Normal Stress



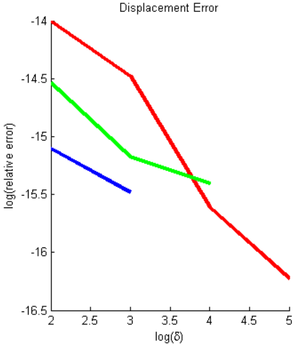
Results - Random Case

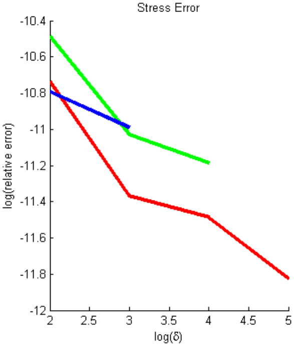
Shear Stress



Results - Random Case

Convergence





Conclusions

- Our method fails as a general purpose PDE solver
- Works best in the worse case scenario: ***random heterogeneity***
- Displacement is well approximated, but not stress.
- Algorithm is highly **parallelizable**
- Results are consistent with other implementations of HMM.

Future Work

- Multiphysics Simulation
- Parallelization
- Improve stress estimation
- Test with other micromodels