Extended Finite Element Methods for Crustal Deformation Simulation

Ethan Coon

Columbia University

Collaborators:
Bruce Shaw (LDEO)
Marc Spiegelman (LDEO/APAM)

June 18, 2010

Introduction

Finite Element Methods

Extended Finite Element Methods

Earthquake rupture

Verification

Fault System Simulation

Conclusions and Discussion

Introduction

Finite Element Methods

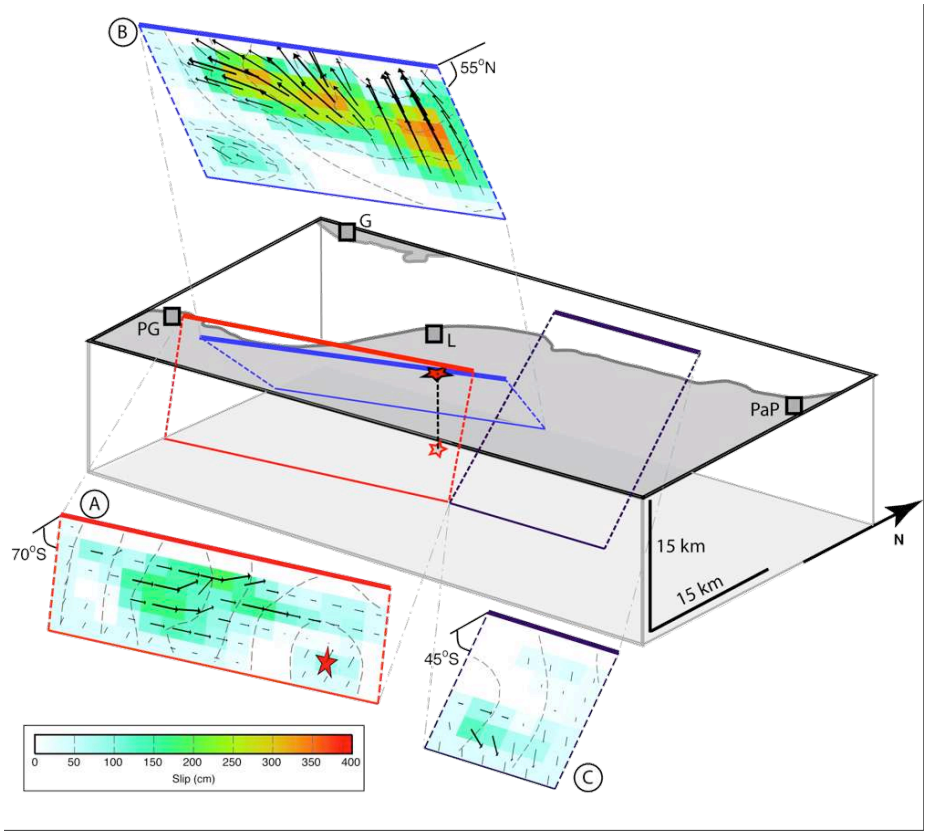
Extended Finite Element Methods

Earthquake rupture

Verification

Fault System Simulation

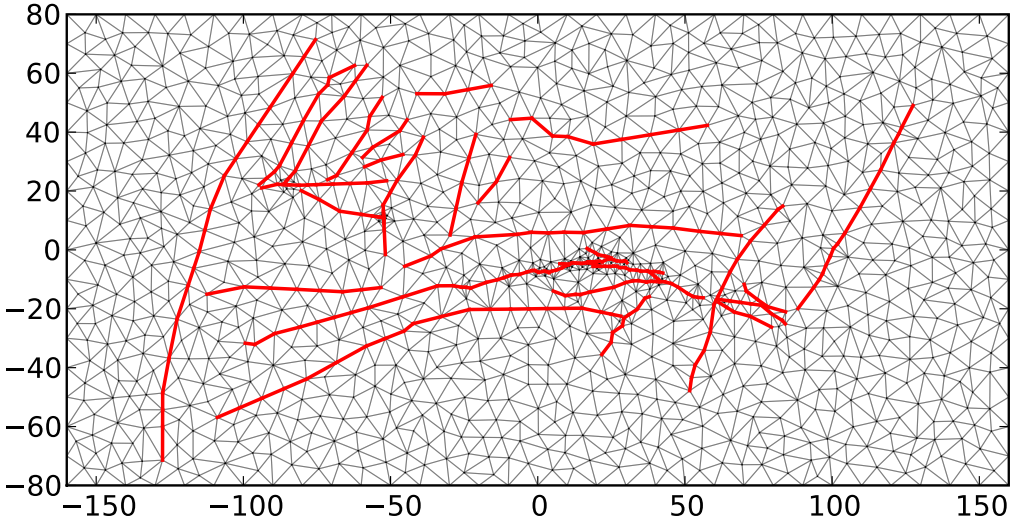
Conclusions and Discussion



Haiti Earthquake
Hayes et al, Wednesday morning

The biggest issue inhibiting the inclusion of fault system geometry in crustal deformation and earthquake simulations is fault meshing.

- Generating meshes which both conform to fault geometry and have good properties for simulation is hard.
- Current process is iterated by hand.
- Fault geometry is uncertain – are solutions robust to fault geometry variation?
- Fault geometry is uncertain – what is the definition of fault?


80
60
40
20
0
−20
−40
−60
−80
−150
−100
−50
0
50
100
150

Accuracy in FEM simulations

In order for FEM solutions to converge, we need the following (approximability):

$$\inf_{v \in \mathcal{V}^h} ||u - v|| \to 0 \text{ as } h \to 0$$

So, to decrease error, either:

Accuracy in FEM simulations

In order for FEM solutions to converge, we need the following (approximability):

$$\inf_{v \in \mathcal{V}^h} ||u - v|| \to 0 \text{ as } h \to 0$$

So, to decrease error, either:

- $h \to 0$ (*h-refinement*)

Accuracy in FEM simulations

In order for FEM solutions to converge, we need the following (approximability):

$$\inf_{v \in \mathcal{V}^h} ||u - v|| \to 0 \text{ as } h \to 0$$

So, to decrease error, either:

- $h \to 0$ (*h-refinement*)
- decrease $\inf_{v \in \mathcal{V}^h} ||u - v||$ *pick better basis functions* $\mathcal{V}$

Introduction

Finite Element Methods

Extended Finite Element Methods

Earthquake rupture

Verification

Fault System Simulation

Conclusions and Discussion

Accuracy in FEM simulations

Introduction

Finite Element Methods

Extended Finite Element Methods

Earthquake rupture

Verification

Fault System Simulation

Conclusions and Discussion

In order for FEM solutions to converge, we need the following (approximability):

$$\inf_{v \in \mathcal{V}^h} ||u - v|| \to 0 \text{ as } h \to 0$$

So, to decrease error, either:

- $h \to 0$ (*h-refinement*)
- decrease $\inf_{v \in \mathcal{V}^h} ||u - v||$ *pick better basis functions* $\mathcal{V}$

Take advantage of *a priori* knowledge of fault geometry and the resulting solutions to construct basis functions which better represent our expected solution u.

Partition of Unity FEM (Melenk and Babuska '96)

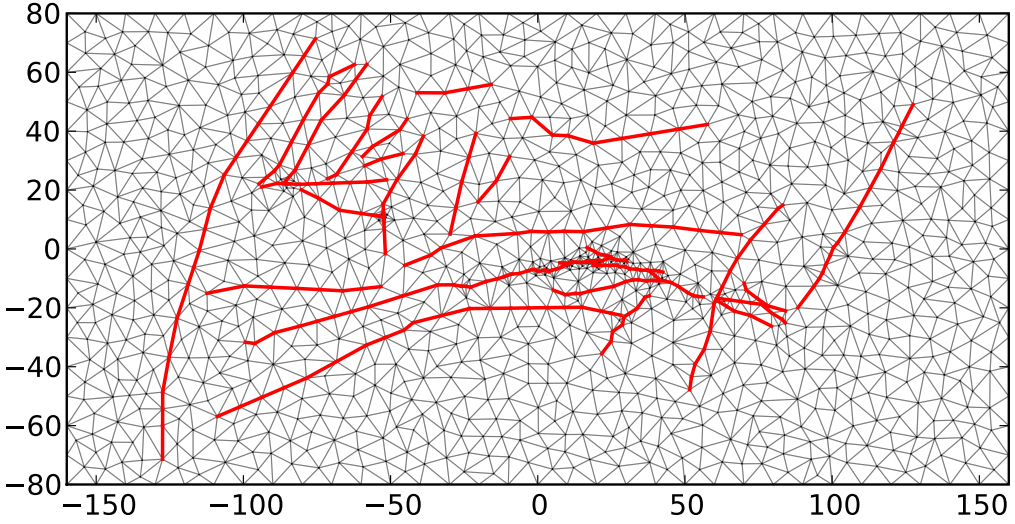
Given:

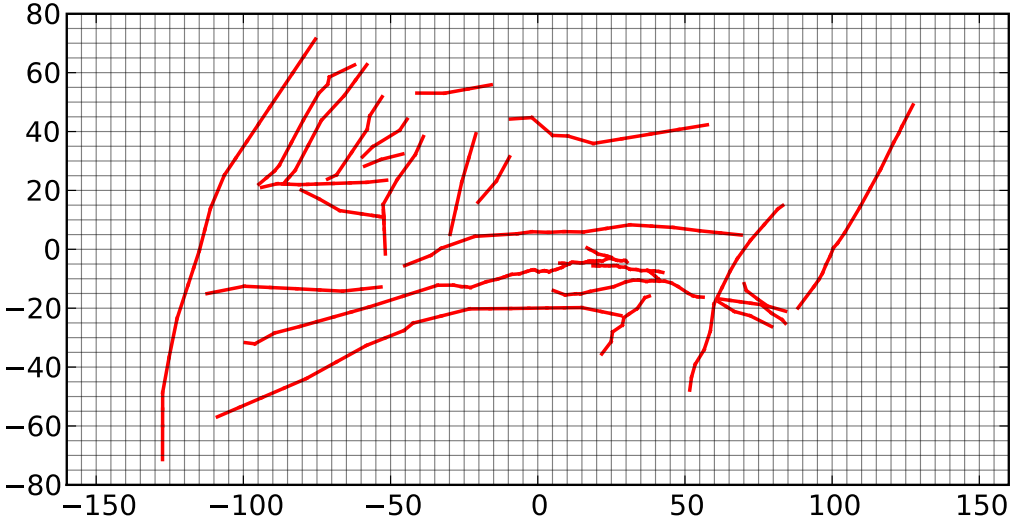
- Partition of unity function space: $\{\phi_i\} \quad s.t. \quad \sum_i \phi_i = 1$ on entire domain
- Local neighborhoods $\Omega_i = support(\phi_i)$

Introduce:

- Local extension functions F_j which incorporate properties of the physics
- Local function spaces $\mathcal{V}^i = \{1\phi_i\} \bigcup \{F_j\phi_i\}_j$
- Approximation space: $\mathcal{V} = \bigcup_j \mathcal{V}^i$

Can show that, if the $\{F_j\}_j$ functions well-approximate the solution on Ω^i, then the solution for $\inf_{v \in \mathcal{V}} ||v - u||$ converges as quickly as approximating a smooth solution with the $\{\phi_i\}$.


80
60
40
20
0
−20
−40
−60
−80
−150
−100
−50
0
50
100
150

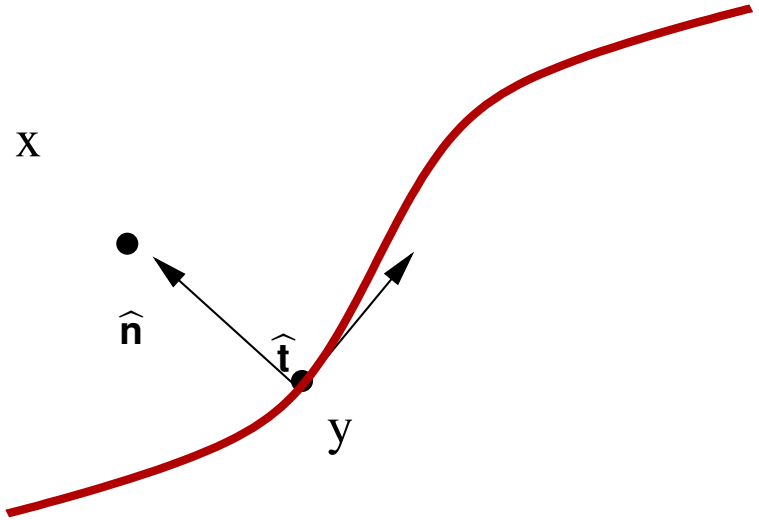

80
60
40
20
0
−20
−40
−60
−80
−150
−100
−50
0
50
100
150

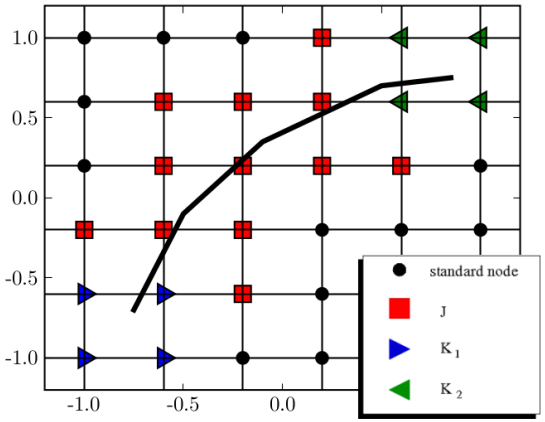
Heaviside extensions

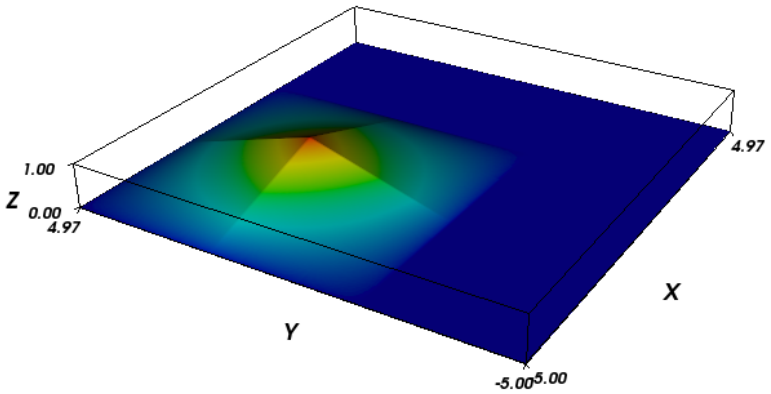
For each fault Γ^k, introduce a Haar function extension.

$$F^k(x) \equiv \begin{cases} 1 & \text{if } (x-y)\cdot\widehat{\mathbf{n}}(y) > 0 \\ -1 & \text{otherwise} \end{cases}$$

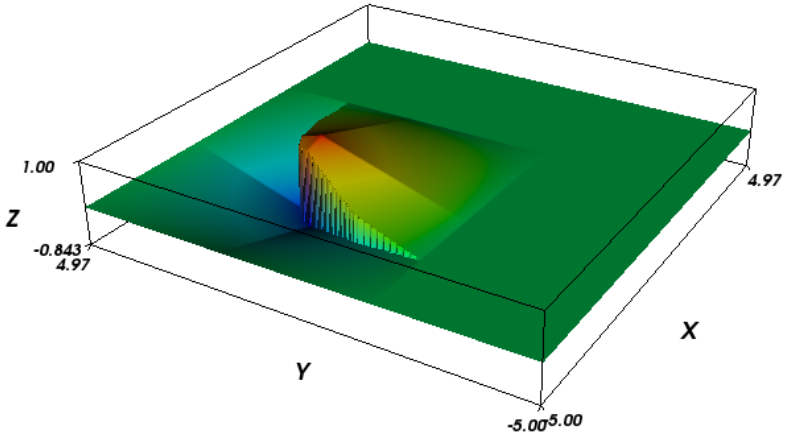
where $y \equiv \arg\min_{z\in\Gamma} |x-z|$







Standard Basis Function



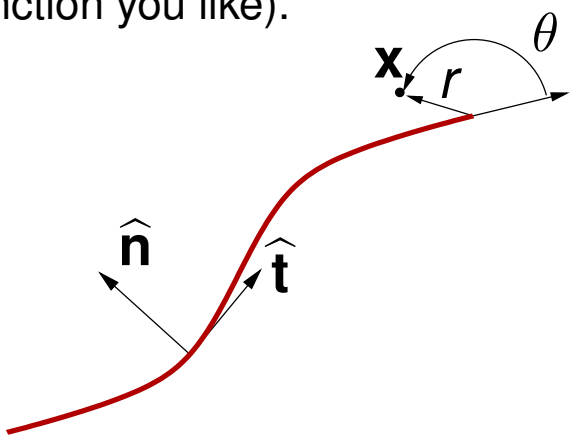
Extended Basis Function

(Belytschko & Black, 1999)
(Dolbow, Moes, Belytschko 2000)

Tip extensions

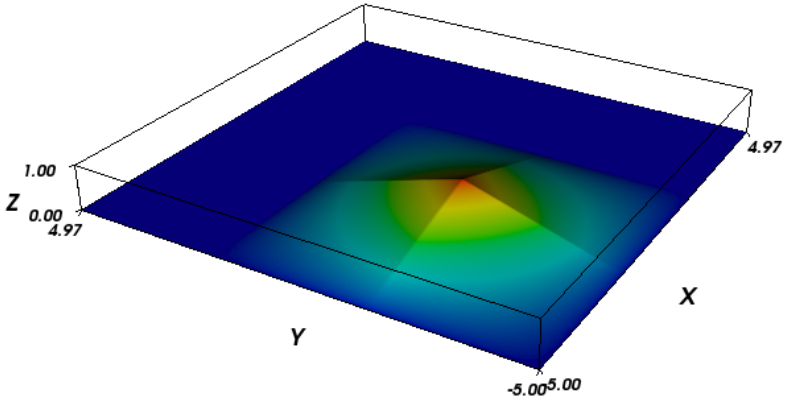
At the tip of crack k, x^k_{tip}, introduce tip functions which match the stress singularity of elasticity (or whatever tip function you like).

$$F^k_{1-4}(x) \equiv \{\sqrt{r}\sin(\theta/2)$$
$$\sqrt{r}\cos(\theta/2)$$
$$\sqrt{r}\sin(\theta/2)\sin(\theta)$$
$$\sqrt{r}\cos(\theta/2)\sin(\theta)\}$$

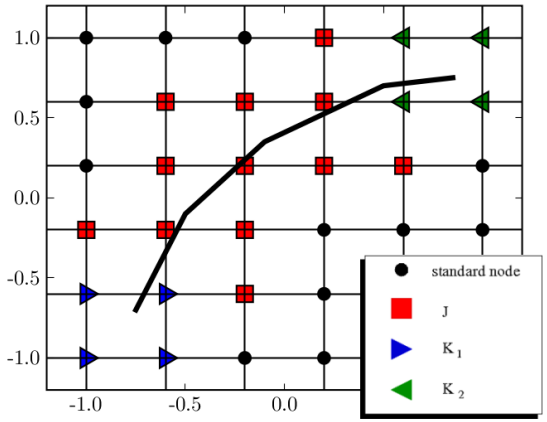


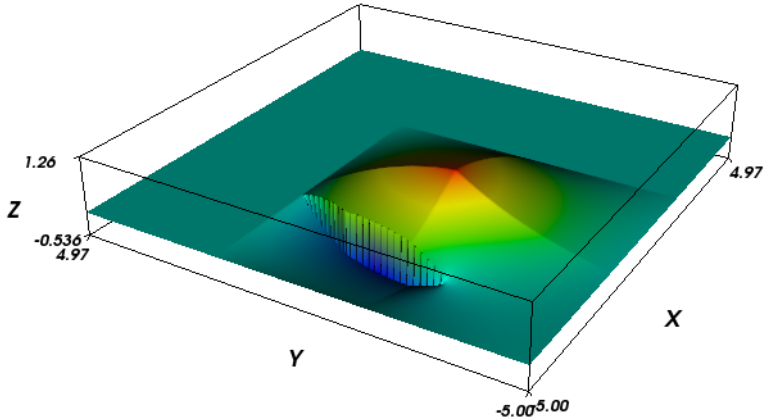
$r \equiv \left| x - x^j_{tip} \right|$

θ is the angle from the tangent of Γ^k at x^k_{tip} to $x - x^k_{tip}$



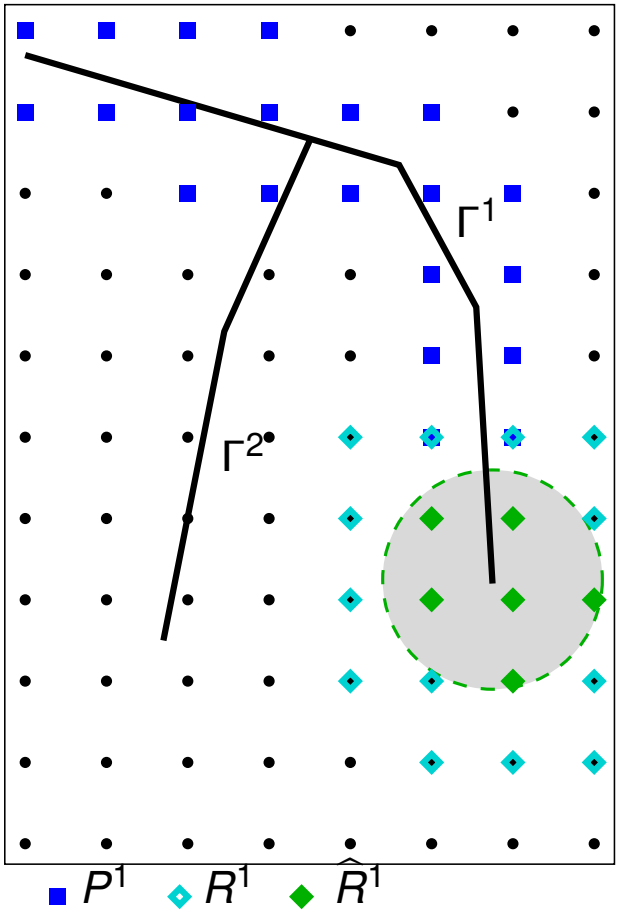
Standard Basis Function



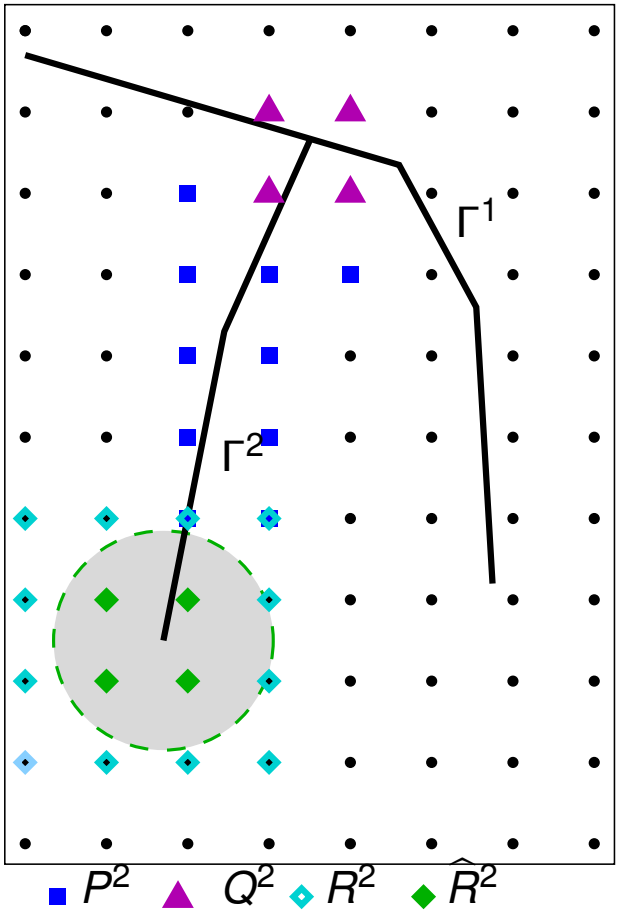


Tip-extended Basis Function

(Belytschko & Black, 1999)
(Dolbow, Moes, Belytschko 2000)



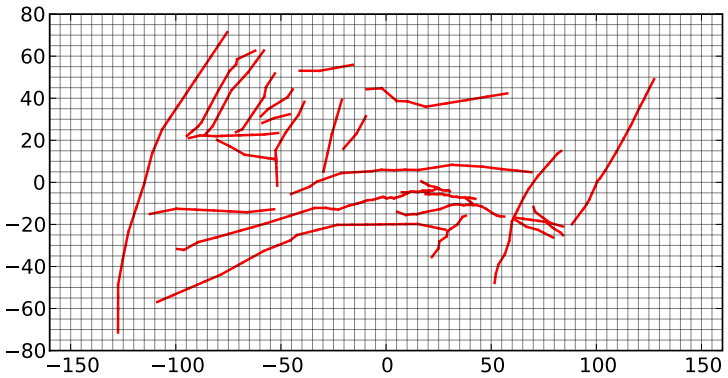
Extensions on crack 1



Extensions on crack 2

Implementation

- hybrid code – Python/Cython/C
- serial code, hard on memory
- leverages PETSc for solvers (all the work)
- leverages petsc4py to use PETSc
- prototype for inclusion in larger, community codes such as PyLith



Earthquakes

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = \nabla \cdot \boldsymbol{\sigma}(\mathbf{u})$$

- no interpenetration:

$$[\![\mathbf{u} \cdot \widehat{\mathbf{n}}]\!]_\Gamma = 0$$

- stick (Γ^D):

$$x \in \Gamma^N \; s.t. \; \left[\!\!\left[\frac{\partial \mathbf{u}}{\partial t}\right]\!\!\right]_{\Gamma^N} = 0 \Longrightarrow x \in \Gamma^D$$

$$\Rightarrow \left[\!\!\left[\frac{\partial \mathbf{u}}{\partial t}\right]\!\!\right]_{\Gamma^D} = 0$$

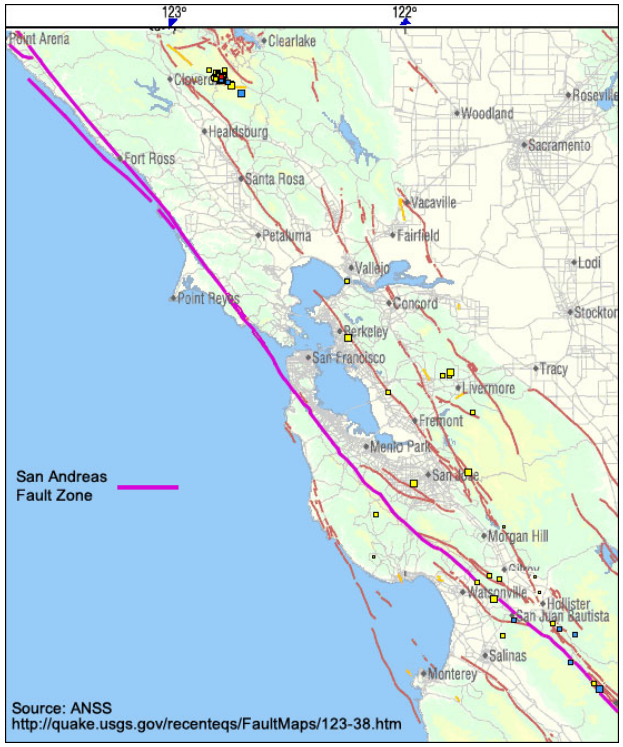
- slip (Γ^N):

$$x \in \Gamma^D \; s.t. \; \left|\widehat{\mathbf{t}}^T \cdot \boldsymbol{\sigma} \cdot \widehat{\mathbf{n}}|_{\Gamma^N}\right| > \left(\widehat{\mathbf{n}}^T \cdot \boldsymbol{\sigma} \cdot \widehat{\mathbf{n}}|_{\Gamma^D}\right) \mu_s$$

$$\Longrightarrow x \in \Gamma^N$$

$$\Rightarrow \widehat{\mathbf{t}}^T \cdot \boldsymbol{\sigma} \cdot \widehat{\mathbf{n}}|_{\Gamma^N}$$

$$= -\widehat{\left[\!\!\left[\frac{\partial \mathbf{u}}{\partial t}\right]\!\!\right]}_{\Gamma^N} \left(\widehat{\mathbf{n}}^T \cdot \boldsymbol{\sigma} \cdot \widehat{\mathbf{n}}|_{\Gamma^N}\right) \mu$$



Earthquakes

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = \nabla \cdot \boldsymbol{\sigma}(\mathbf{u})$$

- no interpenetration:

$$[\![\mathbf{u} \cdot \widehat{\mathbf{n}}]\!]_\Gamma = 0$$

- stick (Γ^D):

$$x \in \Gamma^N \; s.t. \; \left[\!\!\left[\frac{\partial \mathbf{u}}{\partial t} \right]\!\!\right]_{\Gamma^N} = 0 \Longrightarrow x \in \Gamma^D$$

$$\Rightarrow \left[\!\!\left[\frac{\partial \mathbf{u}}{\partial t} \right]\!\!\right]_{\Gamma^D} = 0$$

- slip (Γ^N):

$$x \in \Gamma^D \; s.t. \left| \widehat{\mathbf{t}}^T \cdot \boldsymbol{\sigma} \cdot \widehat{\mathbf{n}}|_{\Gamma^N} \right| > \left(\widehat{\mathbf{n}}^T \cdot \boldsymbol{\sigma} \cdot \widehat{\mathbf{n}}|_{\Gamma^D} \right) \mu_s$$

$$\Longrightarrow x \in \Gamma^N$$

$$\Rightarrow \widehat{\mathbf{t}}^T \cdot \boldsymbol{\sigma} \cdot \widehat{\mathbf{n}}|_{\Gamma^N}$$

$$= -\widehat{\left[\!\!\left[\frac{\partial \mathbf{u}}{\partial t} \right]\!\!\right]}_{\Gamma^N} \left(\widehat{\mathbf{n}}^T \cdot \boldsymbol{\sigma} \cdot \widehat{\mathbf{n}}|_{\Gamma^N} \right) \mu$$

Earthquakes

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = \nabla \cdot \boldsymbol{\sigma}(\mathbf{u})$$

- no interpenetration:

$$[\![\mathbf{u} \cdot \widehat{\mathbf{n}}]\!]_\Gamma = 0$$

- stick (Γ^D):

$$x \in \Gamma^N \; s.t. \; \left[\!\!\left[\frac{\partial \mathbf{u}}{\partial t} \right]\!\!\right]_{\Gamma^N} = 0 \Longrightarrow x \in \Gamma^D$$

$$\Rightarrow \left[\!\!\left[\frac{\partial \mathbf{u}}{\partial t} \right]\!\!\right]_{\Gamma^D} = 0$$

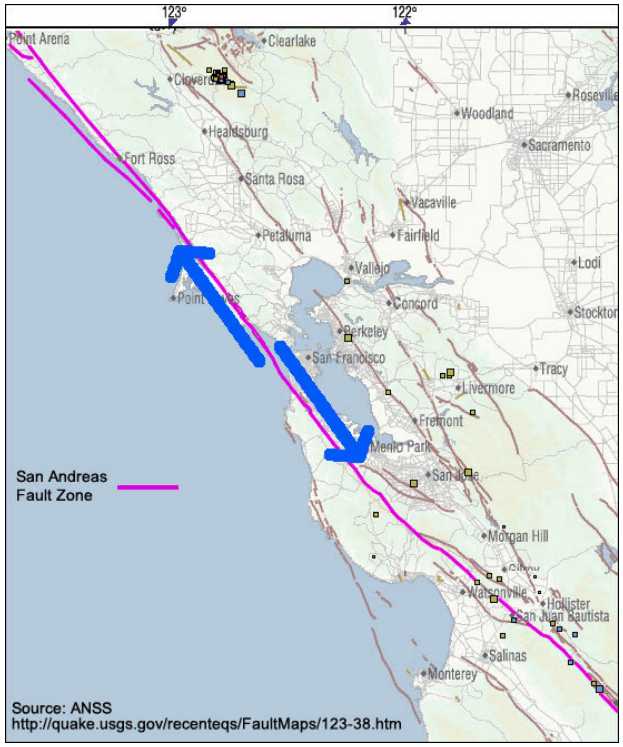
- slip (Γ^N):

$$x \in \Gamma^D \; s.t. \left| \widehat{\mathbf{t}}^T \cdot \boldsymbol{\sigma} \cdot \widehat{\mathbf{n}}|_{\Gamma^N} \right| > \left(\widehat{\mathbf{n}}^T \cdot \boldsymbol{\sigma} \cdot \widehat{\mathbf{n}}|_{\Gamma^D} \right) \mu_s$$

$$\Longrightarrow x \in \Gamma^N$$

$$\Rightarrow \widehat{\mathbf{t}}^T \cdot \boldsymbol{\sigma} \cdot \widehat{\mathbf{n}}|_{\Gamma^N}$$

$$= -\widehat{\left[\!\!\left[\frac{\partial \mathbf{u}}{\partial t} \right]\!\!\right]}_{\Gamma^N} \left(\widehat{\mathbf{n}}^T \cdot \boldsymbol{\sigma} \cdot \widehat{\mathbf{n}}|_{\Gamma^N} \right) \mu$$



Earthquakes

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} = \nabla \cdot \boldsymbol{\sigma}(\mathbf{u})$$

- no interpenetration:

$$[\![\mathbf{u} \cdot \widehat{\mathbf{n}}]\!]_\Gamma = 0$$

- stick (Γ^D):

$$x \in \Gamma^N \; s.t. \; \left[\!\!\left[\frac{\partial \mathbf{u}}{\partial t} \right]\!\!\right]_{\Gamma^N} = 0 \Longrightarrow x \in \Gamma^D$$

$$\Rightarrow \left[\!\!\left[\frac{\partial \mathbf{u}}{\partial t} \right]\!\!\right]_{\Gamma^D} = 0$$

- slip (Γ^N):

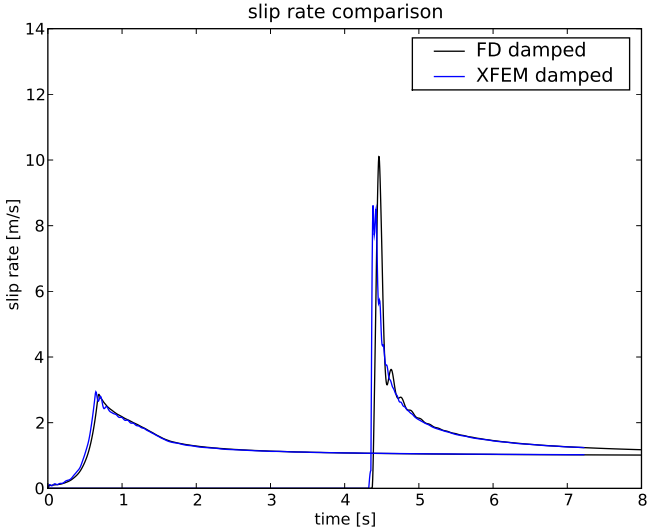
$$x \in \Gamma^D \; s.t. \left| \widehat{\mathbf{t}}^T \cdot \boldsymbol{\sigma} \cdot \widehat{\mathbf{n}}|_{\Gamma^N} \right| > \left(\widehat{\mathbf{n}}^T \cdot \boldsymbol{\sigma} \cdot \widehat{\mathbf{n}}|_{\Gamma^D} \right) \mu_s$$

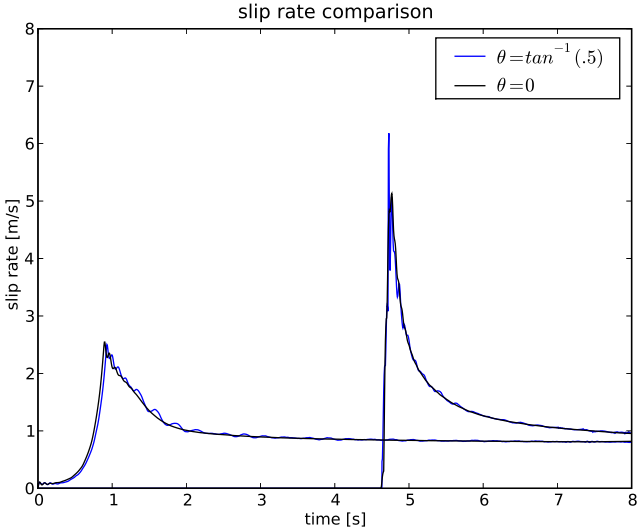
$$\Longrightarrow x \in \Gamma^N$$

$$\Rightarrow \widehat{\mathbf{t}}^T \cdot \boldsymbol{\sigma} \cdot \widehat{\mathbf{n}}|_{\Gamma^N}$$

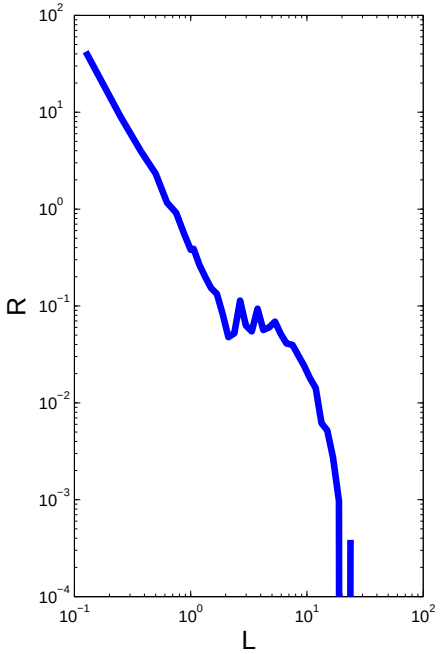
$$= -\widehat{\left[\!\!\left[\frac{\partial \mathbf{u}}{\partial t} \right]\!\!\right]}_{\Gamma^N} \left(\widehat{\mathbf{n}}^T \cdot \boldsymbol{\sigma} \cdot \widehat{\mathbf{n}}|_{\Gamma^N} \right) \mu$$

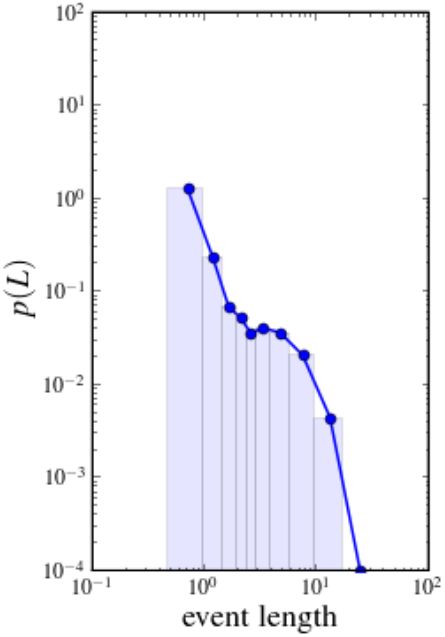
Verification: Mode III





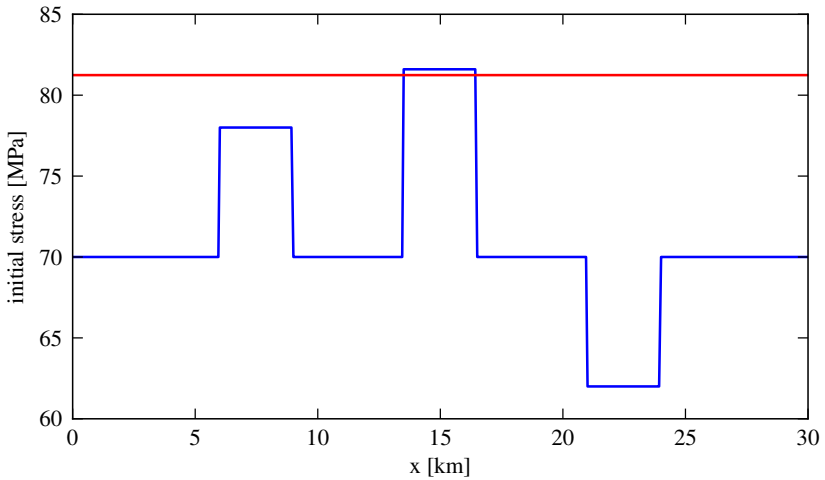
FD

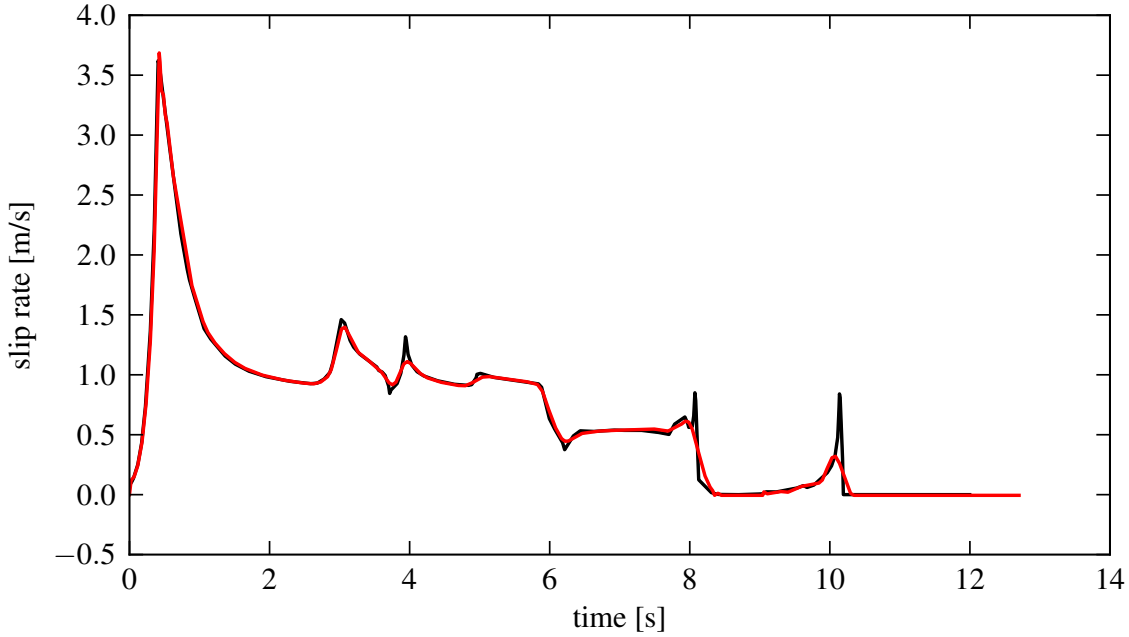




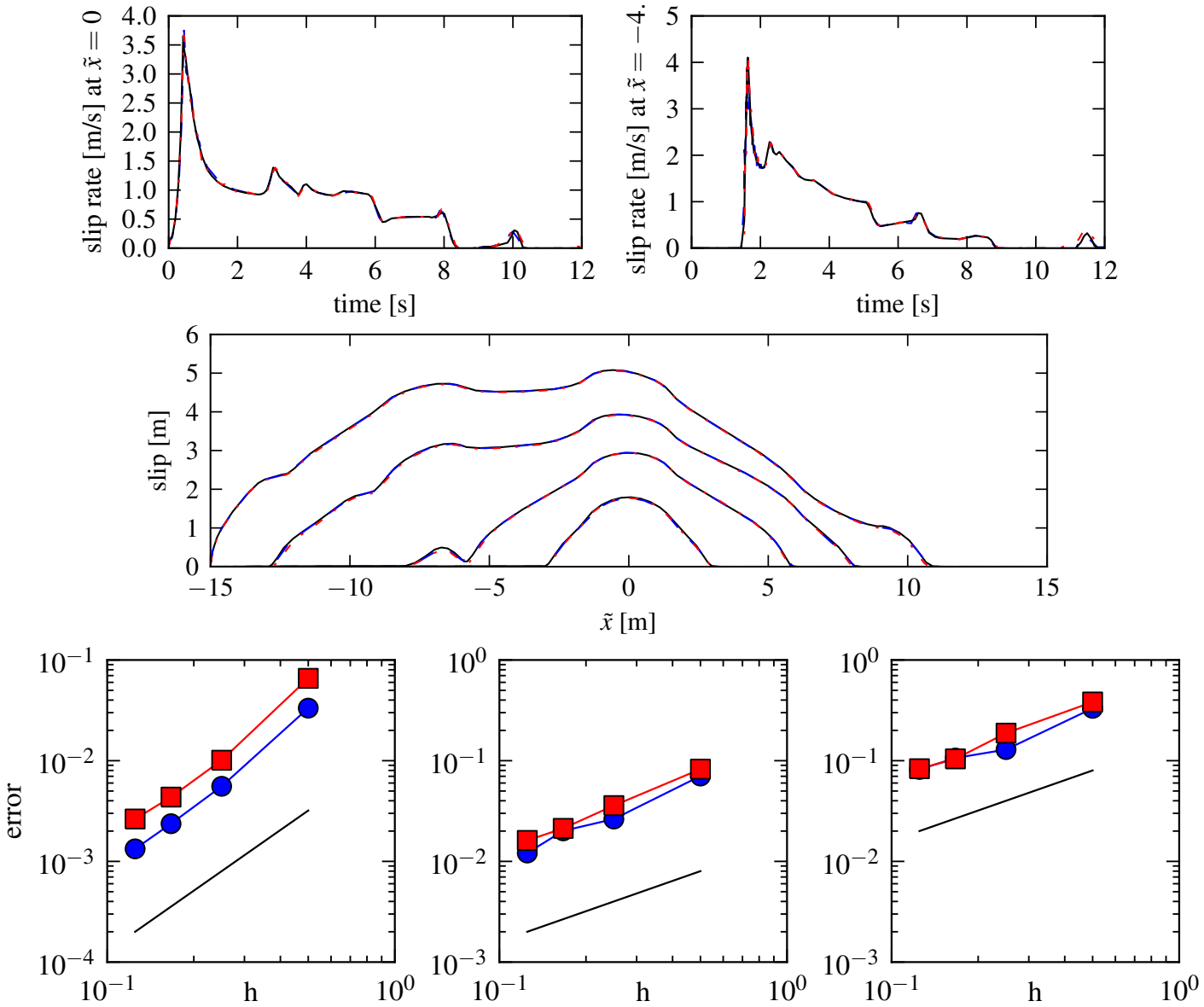
XFEM

Verification: Mode II, Source Dynamics Benchmark, TPV-205

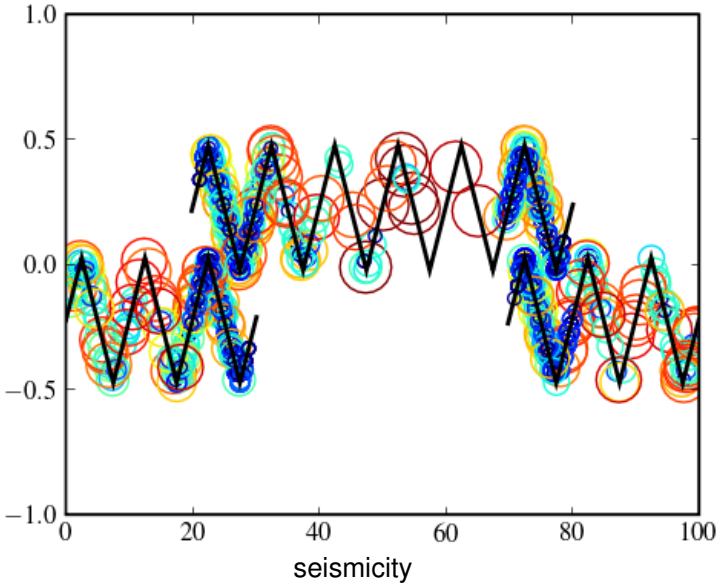


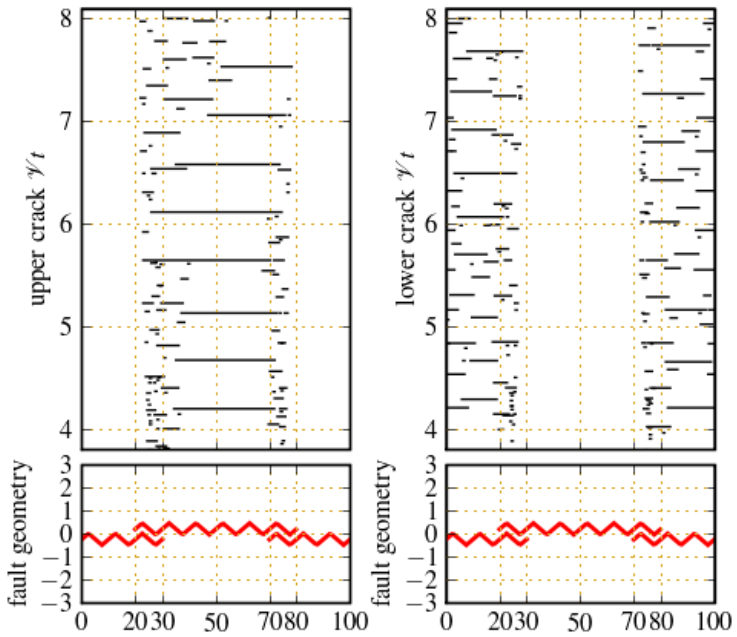


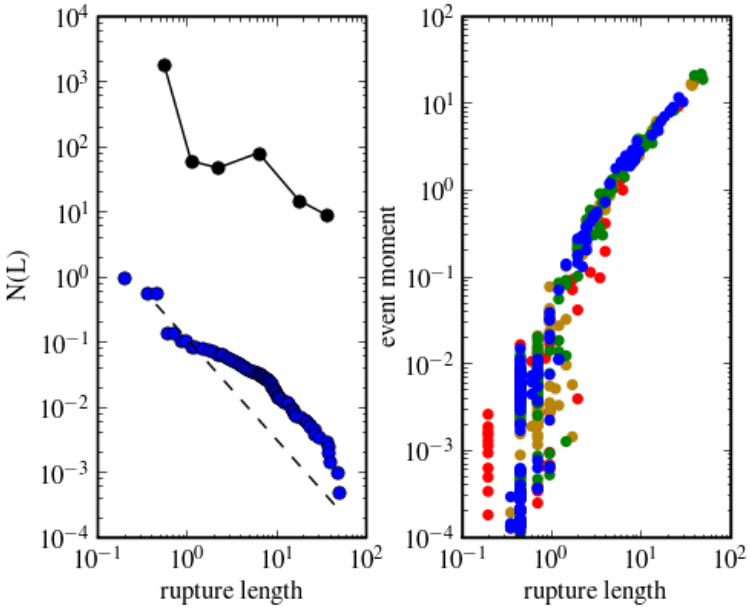
Verification: Mode II, Source Dynamics Benchmark, TPV-205

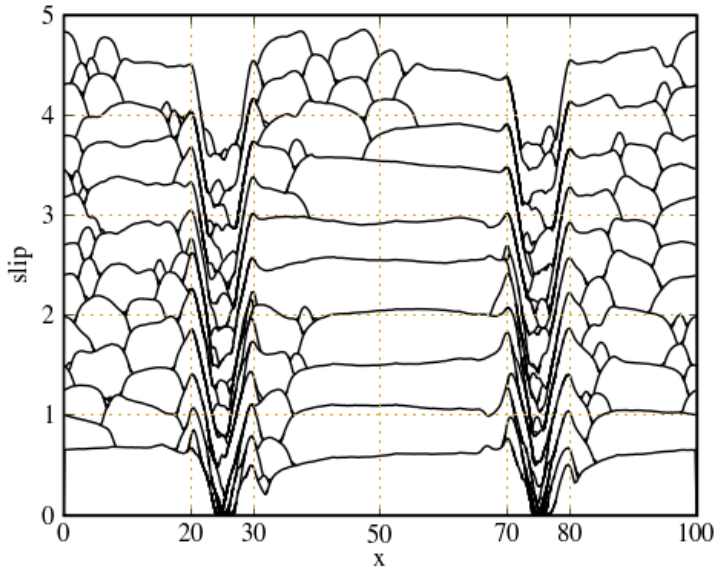


Mode III, Repeated Rupture

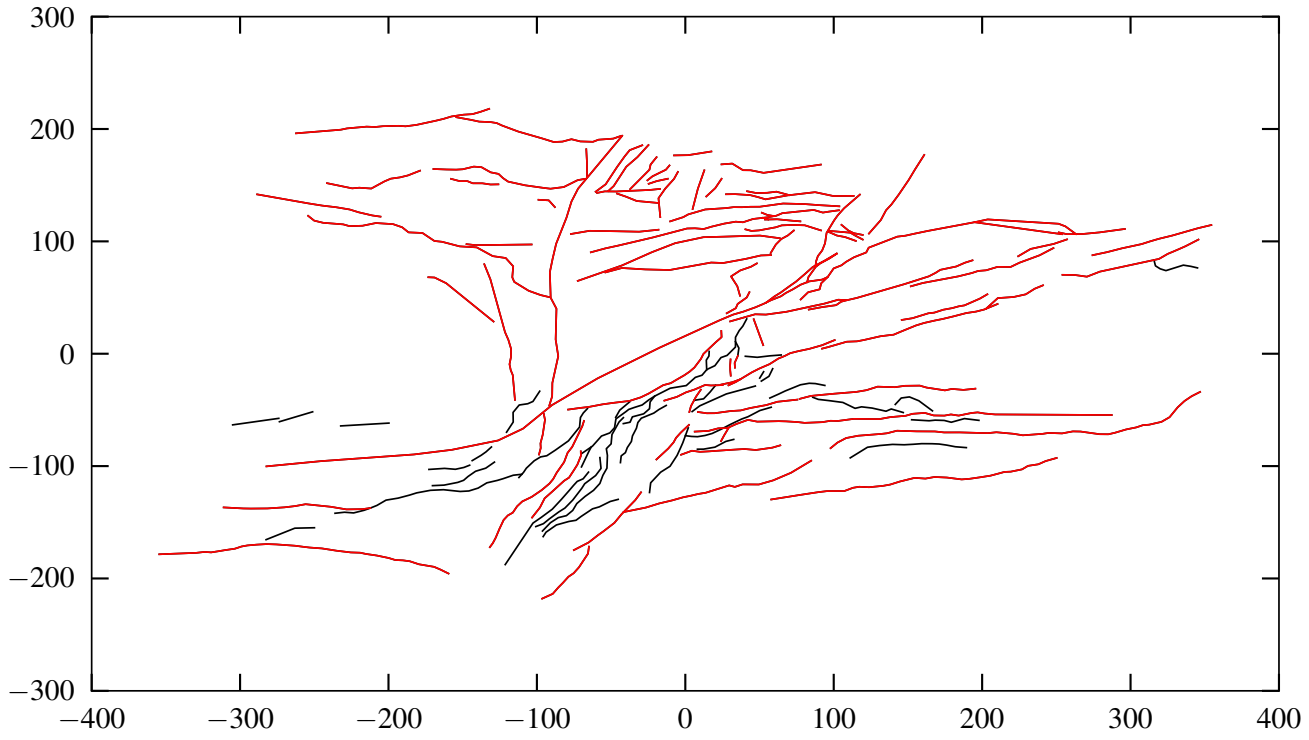




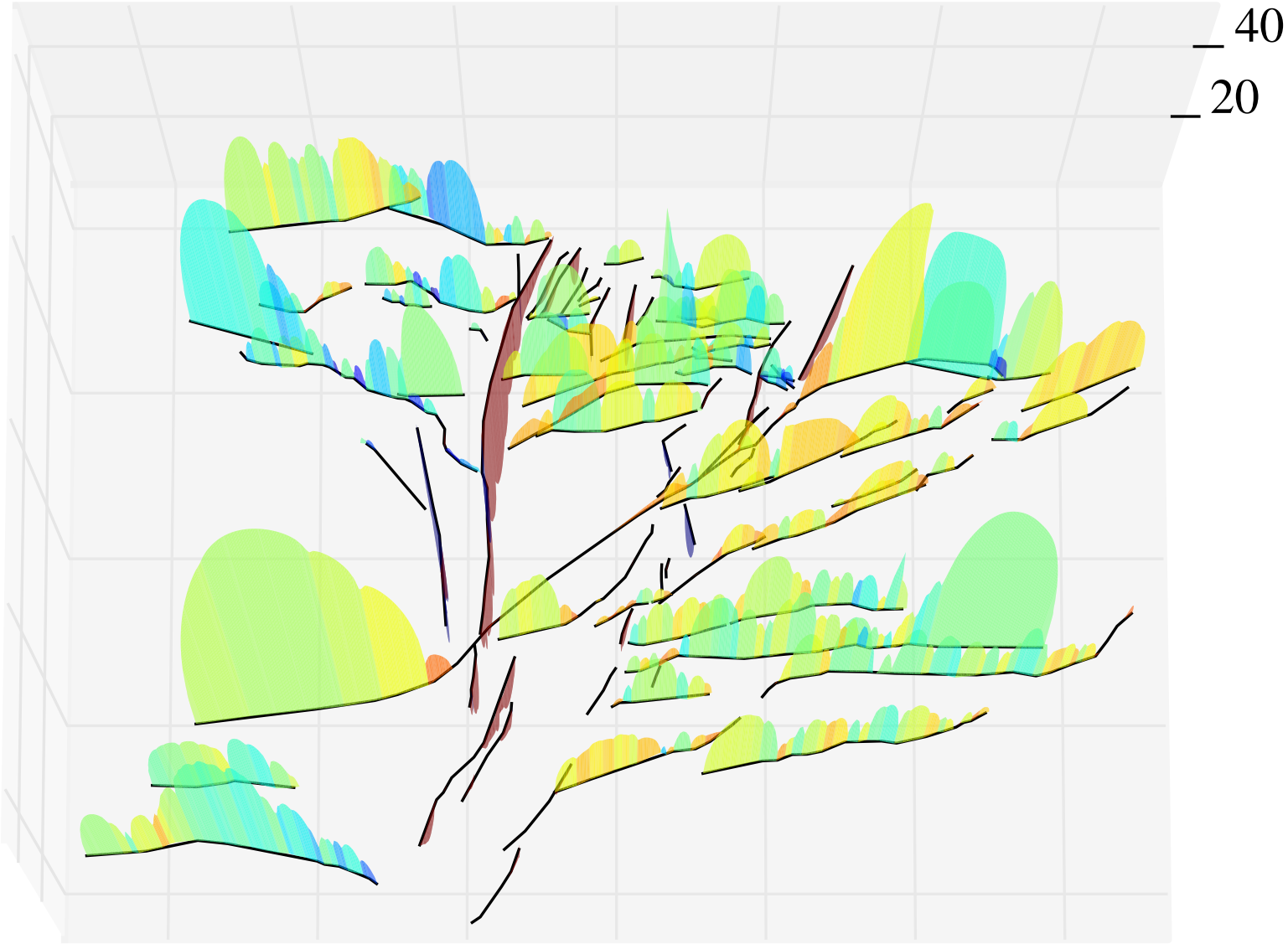




Mode II, CFM Surface Traces

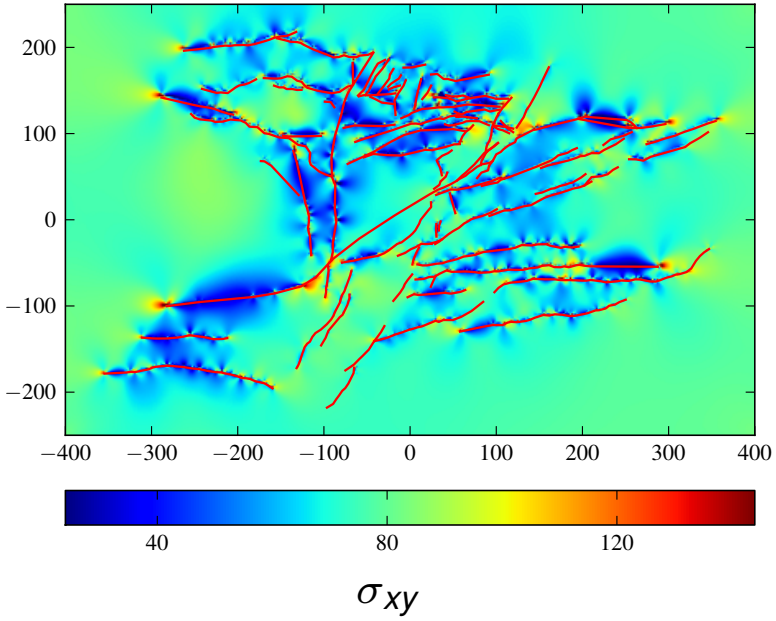


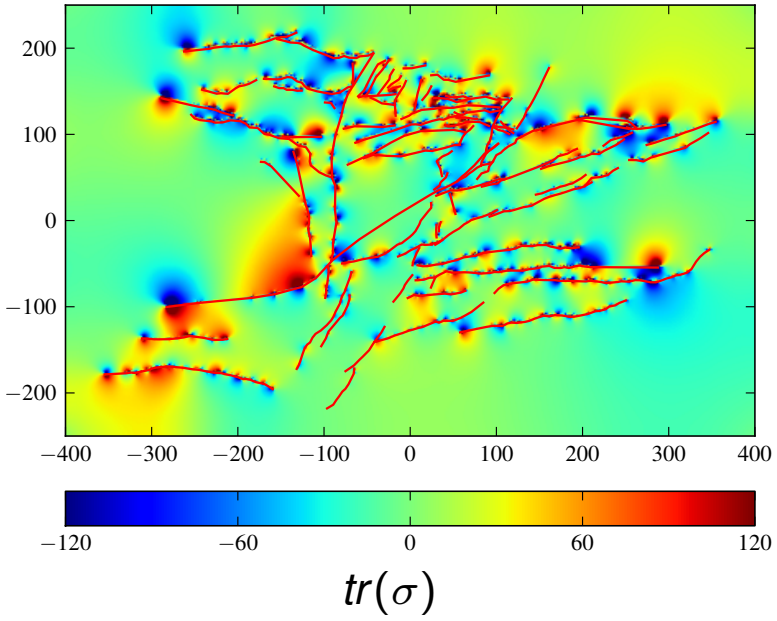
Mode II, CFM Surface Traces



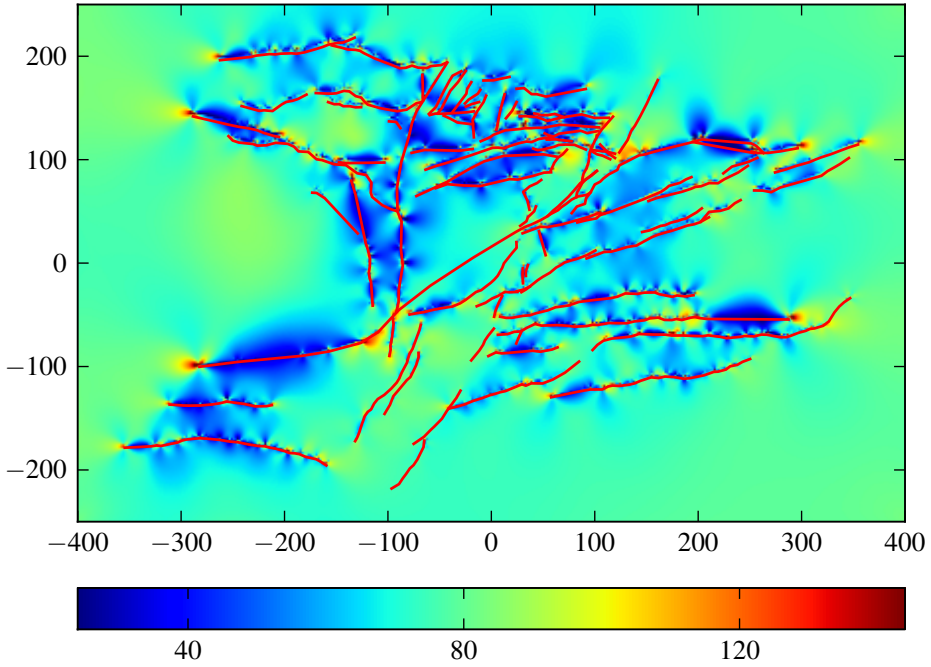
slip [m] after $\sigma_{xy} = .75\sigma_n$ with $\mu_s = 0.2$

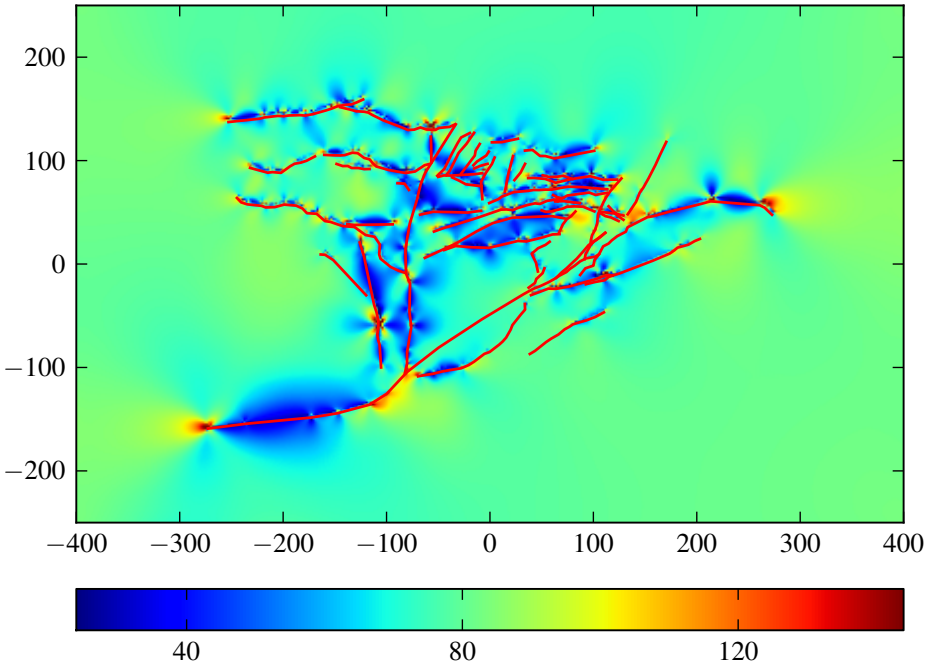
Mode II, CFM Surface Traces

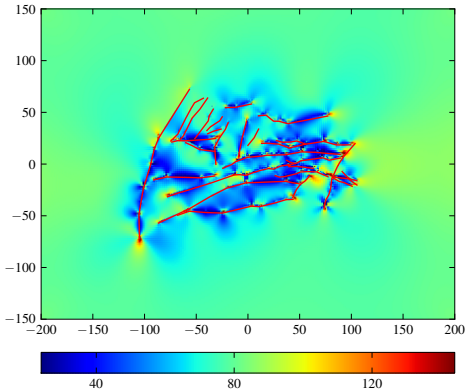




Mode II, CFM Surface Traces







Conclusions

- Mesh generation is a hard problem, and is the primary impediment to system-scale models and simulation.
- FEM methods, even with a robust mesh generator, won't deal with all problems well (robustness to fault geometry, addition/subtraction/formation of faults)
- XFEM allows relaxation of mesh constraints on the FEM.
- In 2D, this enables solution of deformation in fault system simulations (with friction)

Discussion

Incorporation of XFEM in PyLith.

- Requires methods for dealing with fault mesh and background mesh interactions (discrete differential geometry, distance functions, etc) *Mesh generators do this now*
- Requires a fully weak formulation with stabilized, weak boundary conditions to deal with BB conditions while calculating tractions.

BB-condition

Incorporation of tractions and frictional failure requires a mixed formulation:

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{pmatrix} u \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$$

where λ are tractions. Where the fault is stuck, this is the tractions required to keep the fault stuck. Where the fault is slipping, this is friction.

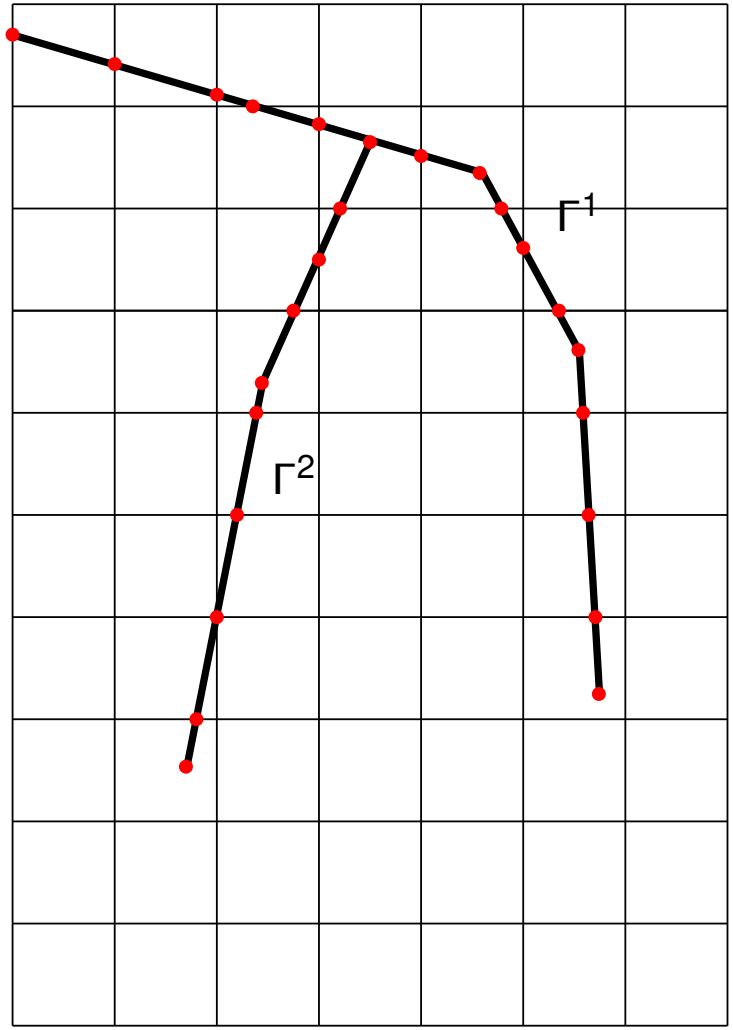
$u \in \mathcal{V}$, but we must choose a function space for λ, i.e. $\lambda \in \mathcal{W}$.

A compatibility condition, or Babuska-Brezzi (BB) condition limits the choices of this space. This condition requires that the bulk solution u not under-constrain the traction solution λ.

BB condition: if $B^T_{ij} = b(\psi_i, \lambda_j)$, require: $\inf_{w \in \mathcal{W}} \sup_{v \in \mathcal{V}} \frac{b(v,w)}{|v|\,|w|} > \epsilon > 0$

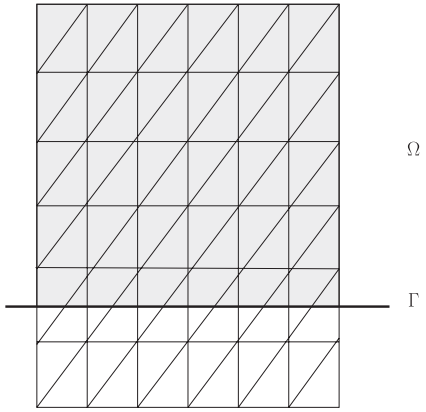
BB-condition

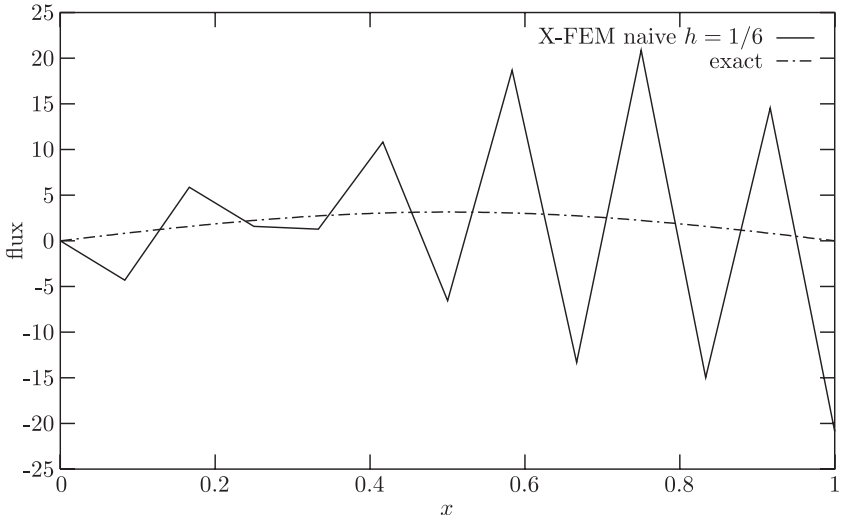
The obvious choice: choose piecewise linear functions on a fault mesh which consists of intersections of the fault with elements of the background mesh.

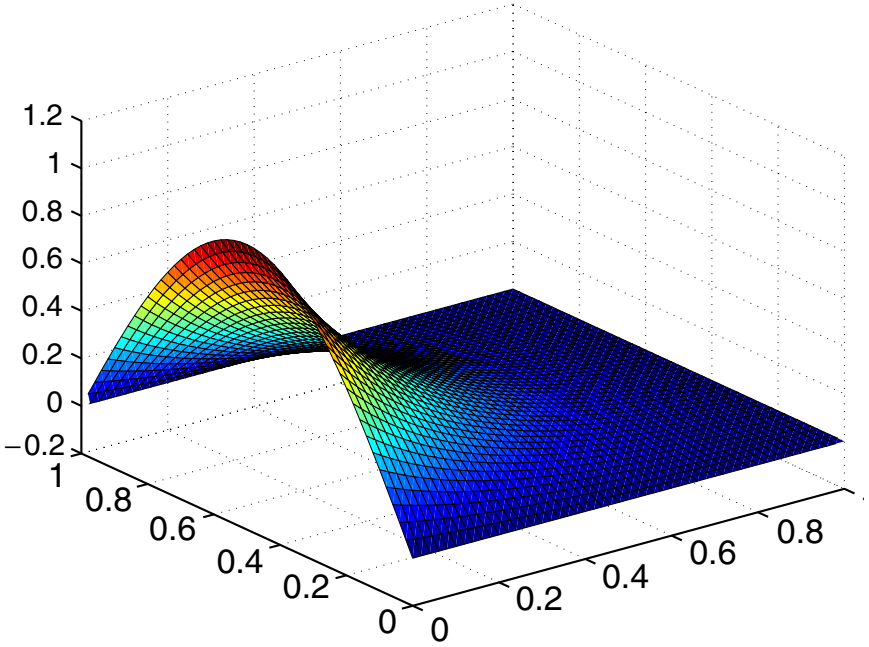


BB-condition

The obvious choice, in XFEM, is bad (Moes et al '06):

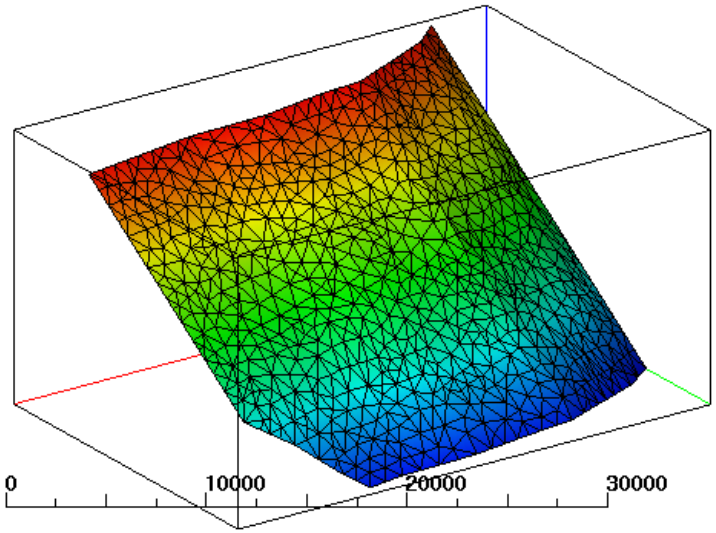


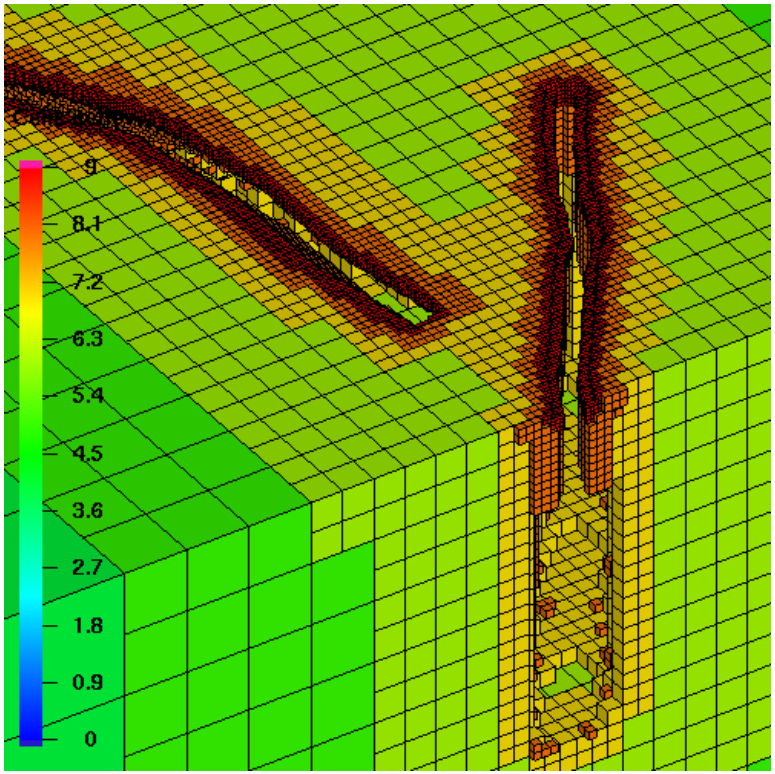




BB-condition

So, require $h^{\Gamma} > h^{\Omega}$





Conclusions

- Mesh generation is a hard problem, and is the primary impediment to system-scale models and simulation.
- FEM methods, even with a robust mesh generator, won't deal with all problems well (robustness to fault geometry, addition/subtraction/formation of faults)
- XFEM allows relaxation of mesh constraints on the FEM.
- In 2D, this enables solution of deformation in fault system simulations (with friction)
- Future work extends this to 3D, within PyLith. Much of the work needed is already done by PyLith and/or mesh libraries such as LaGrit. Compatibility conditions must be considered, but if $h^{\Gamma} > h^{\Omega}$ (a criteria easily enforced via mesh generators), simple function spaces may be BB-compatible.