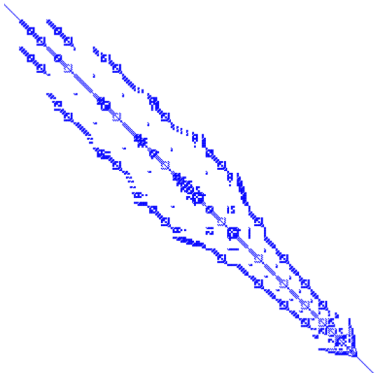
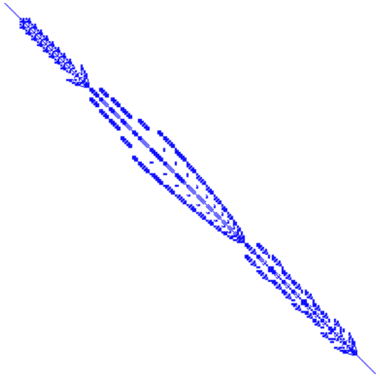
An algebraic multi-grid implementation in FEniCS for solving 3d fracture problems using XFEM

Axel Gerstenberger, Garth N. Wells, Chris Richardson,
David Bernstein, and Ray Tuminaro





FEniCS’14, Paris, France

Overview

This talk is about efficiently solving XFEM based fracture simulations using Algebraic Multi Grid methods.

Questions answered by this talk:

- What problems can be solved by our approach?
- Why does it work?
- How do I need to change my XFEM software to use it?

Structure of the talk:

Part I : Theory & 2d implementation

Part II: Extensions to 3D and FEniCS

Part I

Theory

&

2d Implementation

Algebraic Multigrid Techniques for the eXtended Finite Element Method

Axel Gerstenberger, Ray Tuminaro

Thanks to: E. Boman, J. Gaidamour (Sandia),
B. Hiriyur, H. Waisman (Columbia U.)

- Motivation
  - A brief review of XFEM & Smoothed Aggregation - Algebraic Multigrid (SA-AMG)
    - Why does standard SA-AMG fail & how to fix it
      - Examples
        - Conclusion

SAND 2011-7629C

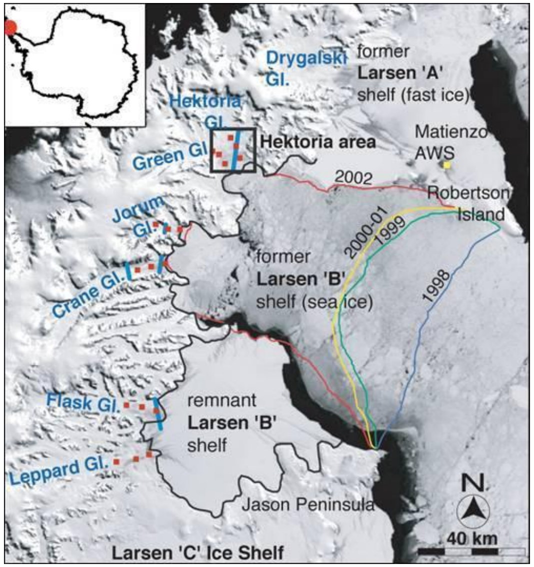
Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Fracture of ice

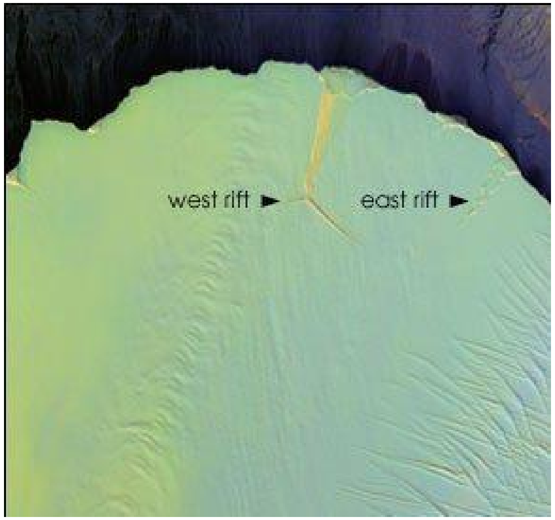
Objective: Employ parallel computers to better understand how fracture of land ice affects the global climate. Fracture happens e.g. during

- the collapse of ice shelves,
- the calving of large icebergs, and
- the role of fracture in the delivery of water to the bed of ice sheets.

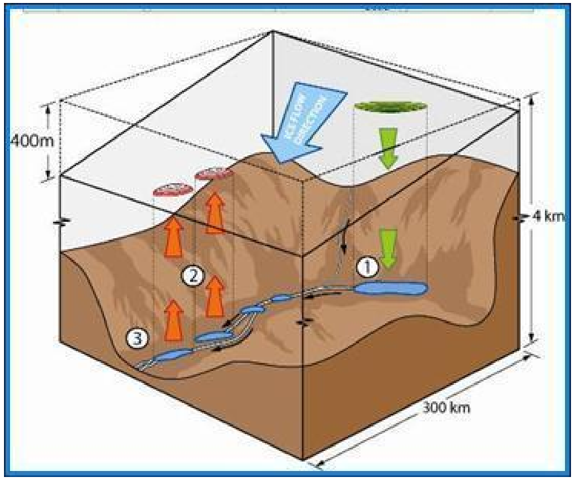
Ice shelves in Antarctica:



Larsen 'B' diminishing shelf 1998-2002
Other example: Wilkins ice shelf 2008



Amery ice shelf



Glacial hydrology
(Source: http://www.sale.scar.org)

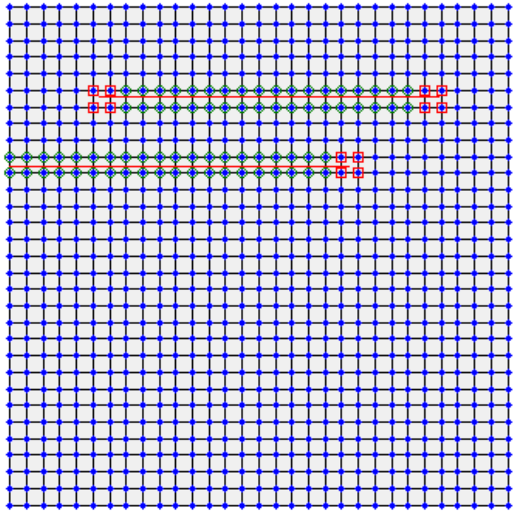
Linear elastic XFEM Formulation for Cracks

Displacement approximation (shifted basis form.)

$$u^h(\mathbf{x}) = \sum_{I=1}^{n} N_I(\mathbf{x}) u_I$$

$$🟩\ + \sum_{i=1}^{n_h} N_{I_i}(\mathbf{x}) \left(H(\mathbf{x}) - H(\mathbf{x}_{I_i}) \right) a_{I_i}$$

$$🟥\ + \sum_{i=1}^{n_f} N_{\hat{I}_i}(\mathbf{x}) \sum_{J=1}^{n_J} \left(F_J(\mathbf{x}) - F_J\left(\mathbf{x}_{\hat{I}_i}\right) \right) b_{\hat{I}_i J}$$



- 🟩 Jump Enrichment
- 🟥 Tip Enrichment (brittle crack)

$$H(\boldsymbol{x}) = \begin{cases} 0.5 & \text{in } \Omega^+ \\ -0.5 & \text{in } \Omega^- \end{cases}$$

$$F_J(r,\theta) = \left\{ \overbrace{\sqrt{r}\sin\left(\frac{\theta}{2}\right)}^{J=1}, \overbrace{\sqrt{r}\cos\left(\frac{\theta}{2}\right)}^{J=2}, \overbrace{\sqrt{r}\sin\left(\frac{\theta}{2}\right)\sin(\theta)}^{J=3}, \overbrace{\sqrt{r}\cos\left(\frac{\theta}{2}\right)\sin(\theta)}^{J=4} \right\}$$

Bubnov-Galerkin method → Symmetric global system

$$\boldsymbol{A} = \textstyle\sum_e \int_{\Omega_e} \boldsymbol{B}_e^{\mathrm{T}} \boldsymbol{C} \boldsymbol{B}_e \, \mathrm{d}\boldsymbol{x}$$

$$\boldsymbol{f} = \textstyle\sum_e \int_{\Gamma_e} \boldsymbol{N}_e^{\mathrm{T}} h \, \mathrm{d}\boldsymbol{x} + \sum_e \int_{\Omega_e} \boldsymbol{N}_e^{\mathrm{T}} \rho \, \mathrm{d}\boldsymbol{x}$$

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}$$

Current implementation: bi-linear, Lagrange polynomials, quad4 elements

Sandia National Laboratories

Multigrid principles

- Oscillatory components of error are reduced effectively by smoothing, but smooth components attenuate slower

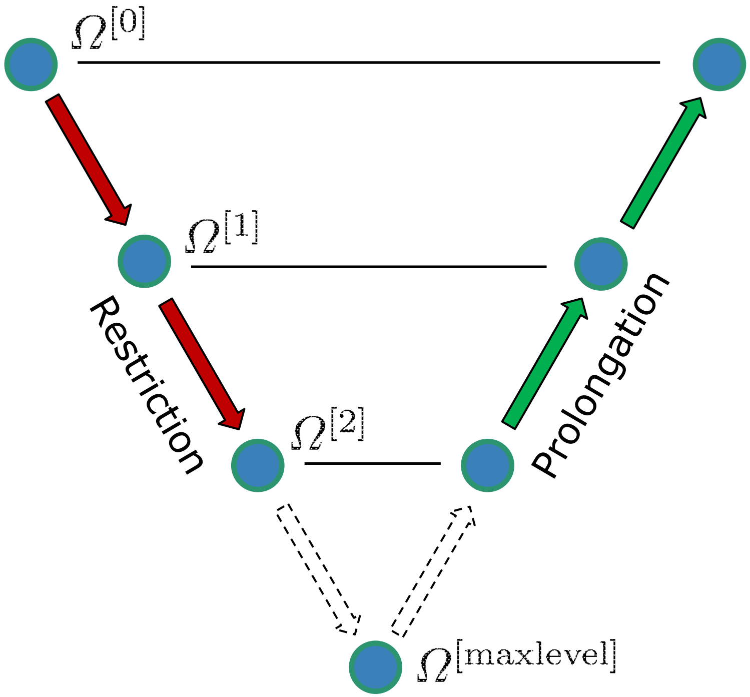
  → capture error at multiple resolutions using grid transfer operators $\boldsymbol{R}^{[k]}$ and $\boldsymbol{P}^{[k]}$

  → optimal number of linear solver iterations
- In **AMG**, transfer operators are obtained from **graph information of *A***

  → ideal for general, unstructured meshes

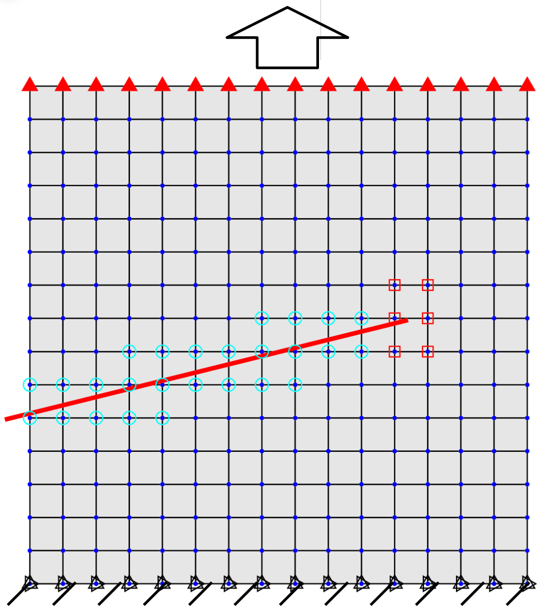
solve Au=b using recursive multilevel V Cycle:

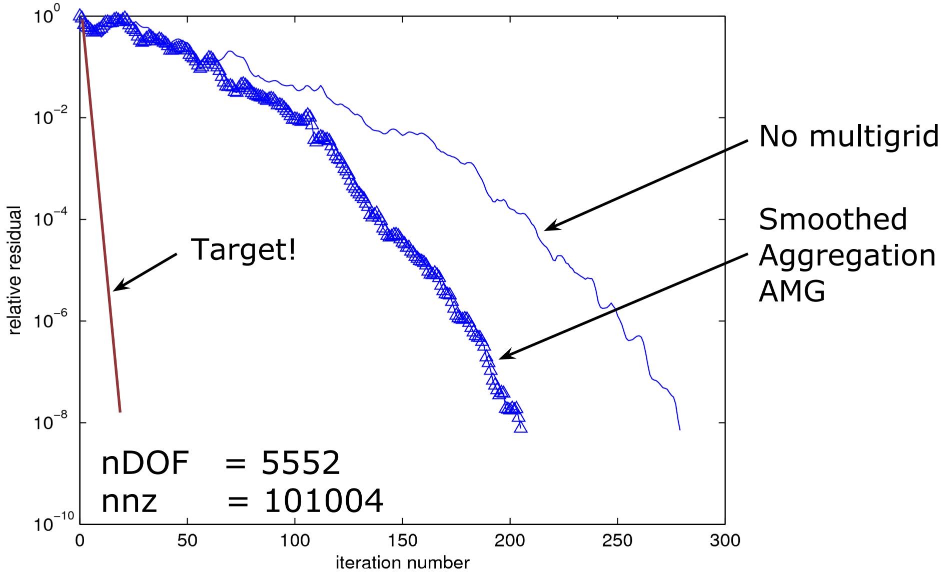
```
function  $\boldsymbol{u} \leftarrow \mathrm{multilevel}(\boldsymbol{b},\ \boldsymbol{u},\ k)$
 smooth (pre-smoothing)
 If $k < maxlevel$:
     restrict u to coarser level
     compute u on coarser level
     interpolate u to finer level
     smooth (post-smoothing)
 return u
```



- iterative smoothers on finest and intermediate levels
- direct solve at the coarsest level

‘Standard’ SA-AMG for fracture problems

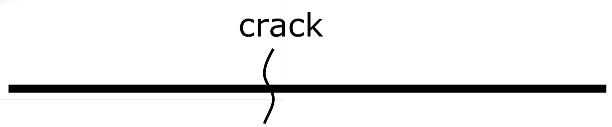




Possible issues:

- XFEM matrix graph messes with aggregation
  - Assumption of 2 unknowns per node not true
  - Aggregates should not cross crack
- How to define rigid body modes?
  - Modes are used to define nullspace
- How to deal with large condition numbers?
  - Define smoothers for each level

Distinct region representation

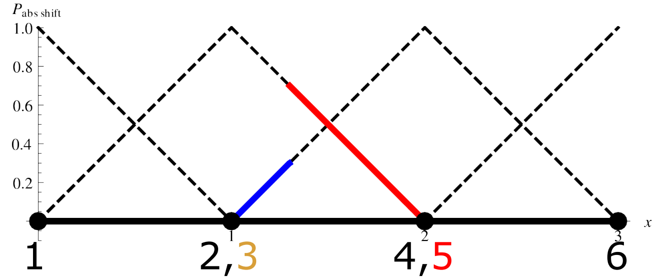


K

M

XFEM: modified shifted enrichment

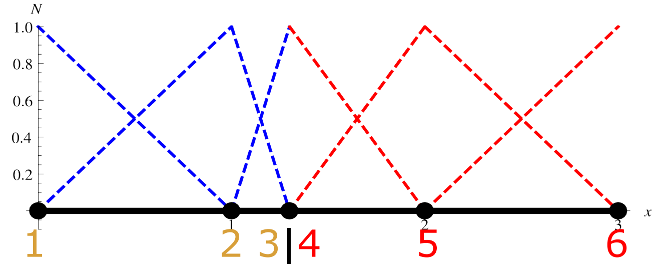
$\sum_I N_I(x)\,|H(x) - H(x_I)|\,a_I$



$$\frac{EA}{2h_1}\begin{bmatrix} 2 & -2 & 0 & 0 & 0 & 0 \\ -2 & 4 & 1 & -2 & -1 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 \\ 0 & -2 & -1 & 4 & 1 & -2 \\ 0 & -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & -2 & 0 & 2 \end{bmatrix}$$

$$\frac{\rho A h_1}{24}\begin{bmatrix} 8 & 4 & 0 & 0 & 0 & 0 \\ 4 & 16 & 1 & 4 & 2 & 0 \\ 0 & 1 & 1 & 2 & 0 & 0 \\ 0 & 4 & 2 & 16 & 1 & 4 \\ 0 & 2 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 4 & 0 & 8 \end{bmatrix}$$

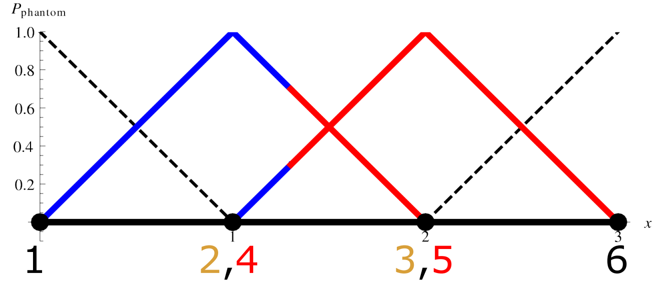
FEM



$$\frac{EA}{2h_1}\left[\begin{array}{ccc|ccc} 2 & -2 & 0 & 0 & 0 & 0 \\ -2 & 6 & -4 & 0 & 0 & 0 \\ 0 & -4 & 4 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 4 & -4 & 0 \\ 0 & 0 & 0 & -4 & 6 & -2 \\ 0 & 0 & 0 & 0 & -2 & 2 \end{array}\right]$$

$$\frac{\rho A h_1}{24}\left[\begin{array}{ccc|ccc} 8 & 4 & 0 & 0 & 0 & 0 \\ 4 & 12 & 2 & 0 & 0 & 0 \\ 0 & 2 & 4 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 4 & 2 & 0 \\ 0 & 0 & 0 & 2 & 12 & 4 \\ 0 & 0 & 0 & 0 & 4 & 8 \end{array}\right]$$

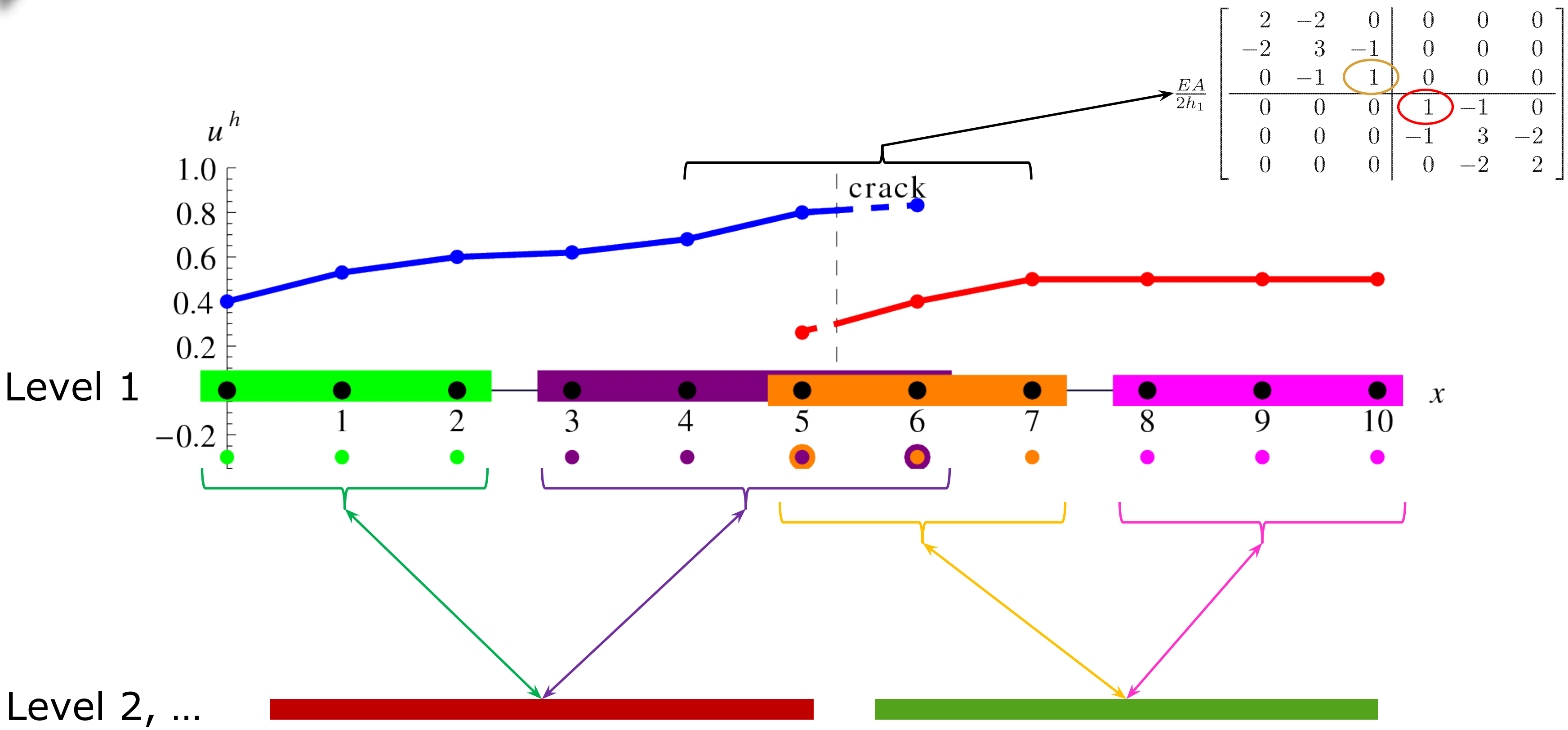
Phantom node approach



$$\frac{EA}{2h_1}\left[\begin{array}{ccc|ccc} 2 & -2 & 0 & 0 & 0 & 0 \\ -2 & 3 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 3 & -2 \\ 0 & 0 & 0 & 0 & -2 & 2 \end{array}\right]$$

$$\frac{\rho A h_1}{24}\left[\begin{array}{ccc|ccc} 8 & 4 & 0 & 0 & 0 & 0 \\ 4 & 15 & 2 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 2 & 15 & 4 \\ 0 & 0 & 0 & 0 & 4 & 8 \end{array}\right]$$

Aggregation for phantom nodes: 1D

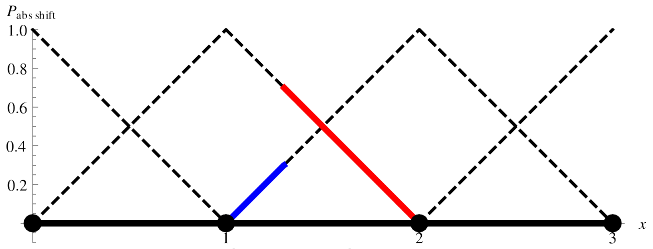


$$\frac{EA}{2h_1}\left[\begin{array}{ccc|ccc} 2 & -2 & 0 & 0 & 0 & 0 \\ -2 & 3 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 3 & -2 \\ 0 & 0 & 0 & 0 & -2 & 2 \end{array}\right]$$

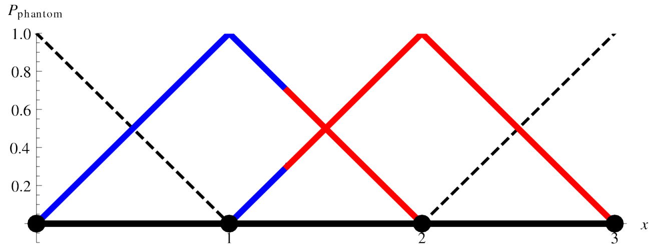
Aggregates are **not** connected on any level!

Change of basis: 1d

Do XFEM developers have to use the phantom node approach? No!



XFEM: modified shifted enrichment



Phantom node approach

For each node I with jump DOFs: $\phi_I - \bar{\phi}_I = \phi_\alpha$

$$\bar{\phi}_I = \bar{\phi}$$

$$\boldsymbol{G}^{\mathrm{T}} \cdot \boldsymbol{A} \cdot \boldsymbol{G} \cdot \boldsymbol{G}^{-1} \cdot \boldsymbol{u} = \boldsymbol{G}^{\mathrm{T}} \cdot \boldsymbol{f}$$

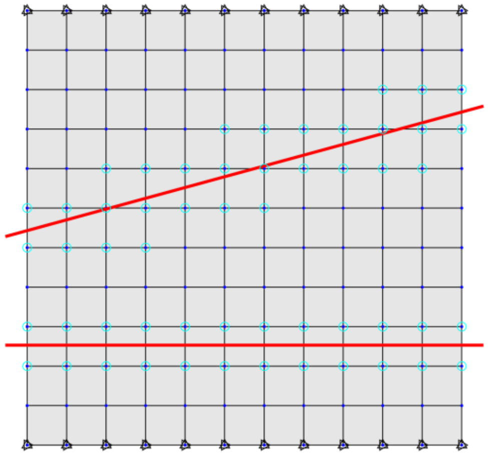
$$\boldsymbol{G}^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(similar: Menouillard 2008, …)

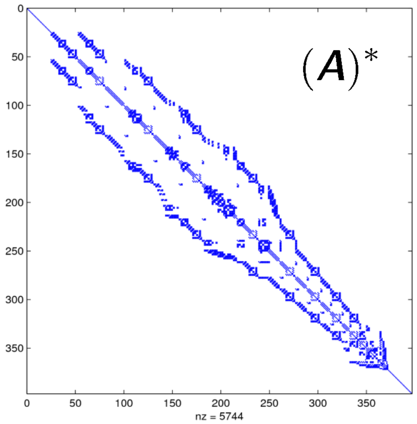
G

- is extremely sparse,
- is simple to produce,
- exists for higher order Lagrange Polynomials and multiple dimensions.

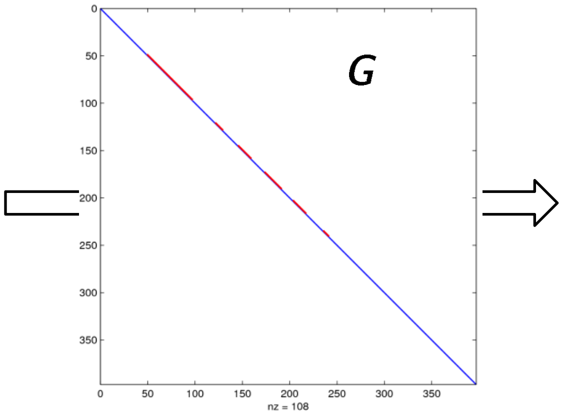
Change of basis: 2d



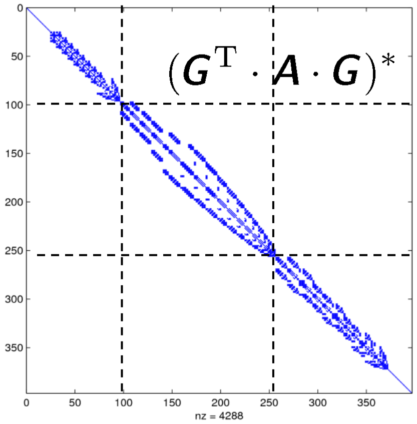
Mesh + BC + Enrichment



Modified shifted enrichment



$G^{\mathrm{T}} \cdot A \cdot G \cdot G^{-1} \cdot u = G^{\mathrm{T}} \cdot f$



Phantom node approach

( )* → sym. rev. Cuthill-McKee permutation for visualization

Null Space

Prolongation/Restriction should preserve zero-energy modes!

2D elasticity problem has 3 Zero Energy Modes (ZEMs):

$$\begin{array}{c|c|c|c|}
 & 1 & 2 & 3 \\
 & & & \dots \\
d_{xI} & 1 & 0 & \text{-y\_I} \\
d_{yI} & 0 & 1 & \text{x\_I} \\
 & & & \dots
\end{array}$$

Null space for phantom node approach

- Standard DOFs are treated as usual
- Phantom DOFs are treated like Standard DOFs

Null space for shifted enrichment approach

- Enriched DOFs don’t contribute to rigid body motion
  - Put 0 into their respective rows
- Change of basis transformation also for nullspace

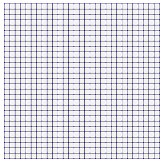
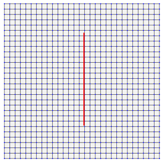
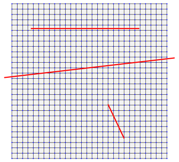
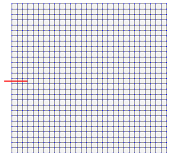
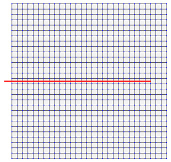
Results for shifted jump enrichments

Conj. Gradient preconditioned with AMG,

- Direct solve on coarse level,
- Block-GS on all finer levels.

A	Shifted enrichment
$G^{\mathrm{T}} \cdot A \cdot G$	Phantom node

Using transformation to phantom node setup is crucial to allow standard graph-based aggregation!

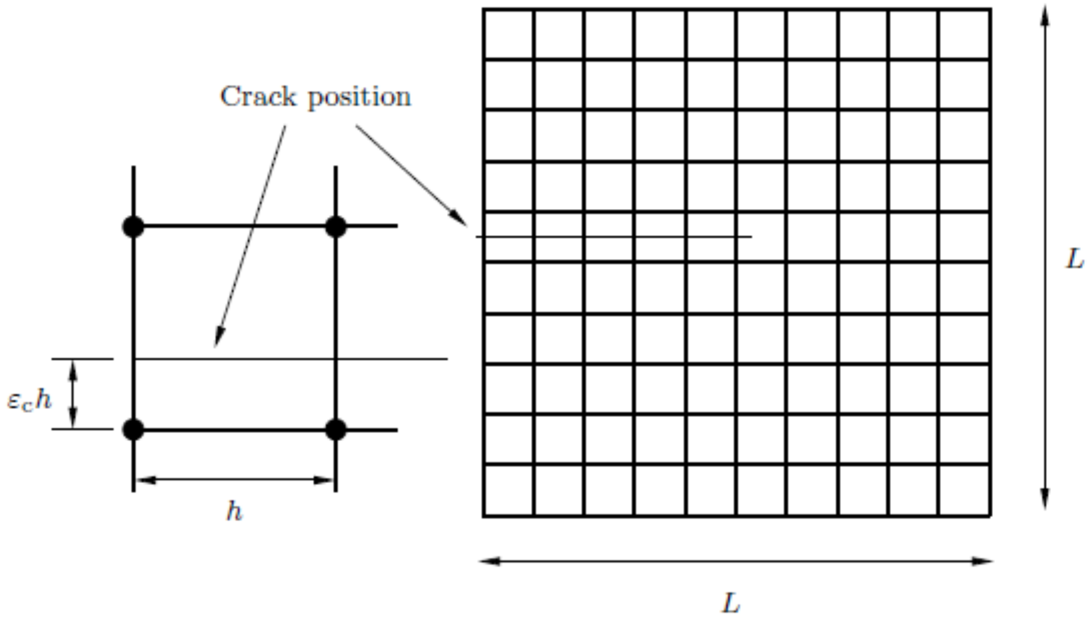
Case	$n_e \times n_e$	$\alpha_{\text{cond.}}$	n_{iter}			
			A		$G^{\mathrm{T}} \cdot A \cdot G$	
			1L	ML	1L	ML
I	30×30	3e+03	32	9	32	9
	60×60	1e+04	63	10	63	10
	90×90	3e+04	93	11	93	11
	120×120	5e+04	123	11	123	11
II	30×30	2e+06	59	40	53	12
	60×60	1e+06	109	58	104	13
	90×90	2e+06	159	65	156	14
	120×120	1e+07	-	81	-	15
III	30×30	1e+04	46	25	42	11
	60×60	5e+04	86	33	83	13
	90×90	1e+05	127	40	127	15
	120×120	2e+05	170	44	167	15
1a	30×30	1e+05	54	16	54	11
	60×60	4e+05	106	21	105	14
	90×90	1e+06	157	24	157	16
	120×120	2e+06	-	26	-	16
1c	30×30	2e+07	78	38	76	16
	60×60	7e+07	150	53	146	17
	90×90	1e+08	-	63	-	18
	120×120	2e+08	-	73	-	21

OC: 1.28-1.40

Conditioning: “Difficult” problems 1

What happens, if we move the crack near element edges?

Example 1:



epsilon	cond. number	# iter
5.0 x 10^{-1}	2.0 x 10^{05}	14
1.0 x 10^{-2}	2.0 x 10^{05}	14
1.0 x 10^{-2}	1.0 x 10^{06}	14
1.0 x 10^{-6}	1.0 x 10^{08}	14
1.0 x 10^{-8}	1.0 x 10^{10}	14

mesh size: 40x40 elements

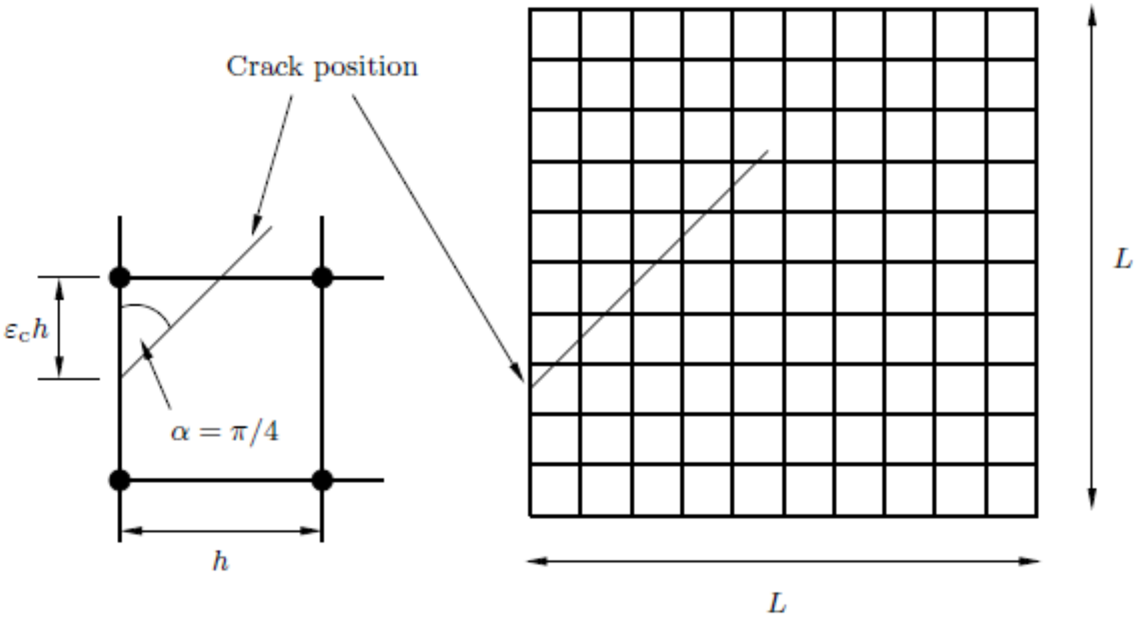
“Block Gauss-Seidel solver generally insensitive to certain conditioning problems that can be solved by diagonal scaling.”

A. Gerstenberger and R. Tuminaro, An algebraic multigrid approach to solve extended finite element method based fracture problems, *Int. J. Numer. Meth. Engng.*, Vol. **94(3)**, 248--272, 2013, DOI: 10.1002/nme.4442

Conditioning: “Difficult” problems 2

What happens, if we move the crack near nodes?

Example 2:



epsilon	cond. number	# iter
5.0 x 10^{-1}	3.0 x 10^{05}	14
1.0 x 10^{-1}	2.0 x 10^{06}	14
1.0 x 10^{-2}	2.0 x 10^{10}	14
1.0 x 10^{-3}	2.0 x 10^{14}	14

mesh size: 40x40 elements

“Block Gauss-Seidel solver generally insensitive to certain conditioning problems that can be solved by diagonal scaling.”

A. Gerstenberger and R. Tuminaro, An algebraic multigrid approach to solve extended finite element method based fracture problems, *Int. J. Numer. Meth. Engng.*, Vol. **94(3)**, 248--272, 2013, DOI: 10.1002/nme.4442

Concluding Remarks

Standard SA-AMG methods can be used, if proper input is provided!

Key components:

- System matrix must be in phantom-node form for jump DOF
  - Either you already have it, (voids, fluid-structure interaction, …) , or
  - do a simple transformation $\boldsymbol{G}^{\mathrm{T}} \cdot \boldsymbol{A} \cdot \boldsymbol{G} \cdot \boldsymbol{G}^{-1} \cdot \boldsymbol{u} = \boldsymbol{G}^{\mathrm{T}} \cdot \boldsymbol{f}$
- Simple null space construction: zero entries for shifted enriched DOF
- Two-step smoothing on finest level (or add your own smoother)

→ Very good convergence behavior.

Current & Future Work

- What happens to tiny element fractions (conditioning)?
- 3d implementation (based on MueLu, the new Multigrid package in Trilinos)

Part II

3D implementation in FEniCS

Introduction - 3d implementation

What problem do we want to solve:

- fracture simulations involving multiple, intersecting cracks
- very large problems

Discretization:

- linear and higher order tets
- absolute shifted enrichment for crack surface
- crack tip treated by adaptivity → no tip enrichment

Required implementation steps:

- compute G
- transform lin. system: $A* = G^T A\ G$, $f* = G^T\ f$,
- transform nullspace $N* = G^{-1} N$
- solve $A^* u^* = f^*$ for u* using standard AMG
- transform back Gu* → u

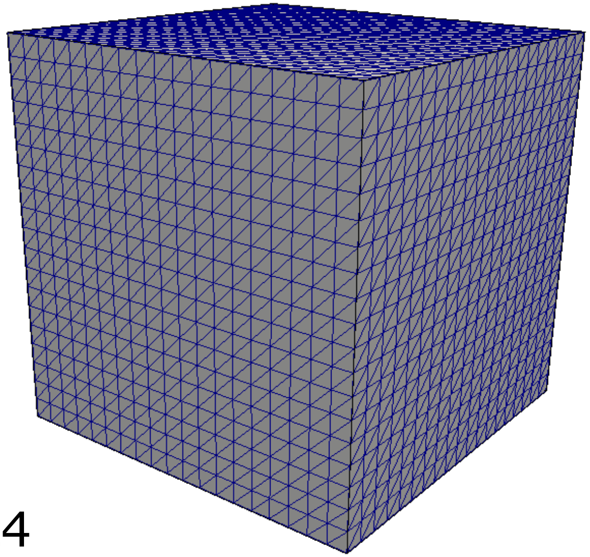
XAMGSolver class - 3d implementation

- Transformation & solver algorithm implemented using Generic interface classes for Vector, Matrix, LinearSolver, and Preconditioner
- Implemented for Epetra/Trilinos and PETSc backend
  - Added Generic PtAP function
  - Added Generic MatrixMarket output for debugging
- Support for PETSc AMG and ML (Trilinos) precond.

LinAlg backend \ AMG lib	PETSc AMG	ML (Trilinos)
PETSc	X	X
Epetra		X

- Sets minimal default parameters for each AMG precond.
- Use new XFEM Dofmap to construct G
- Compute nullspace (currently elasticity and Poisson)

Example - 3d implementation

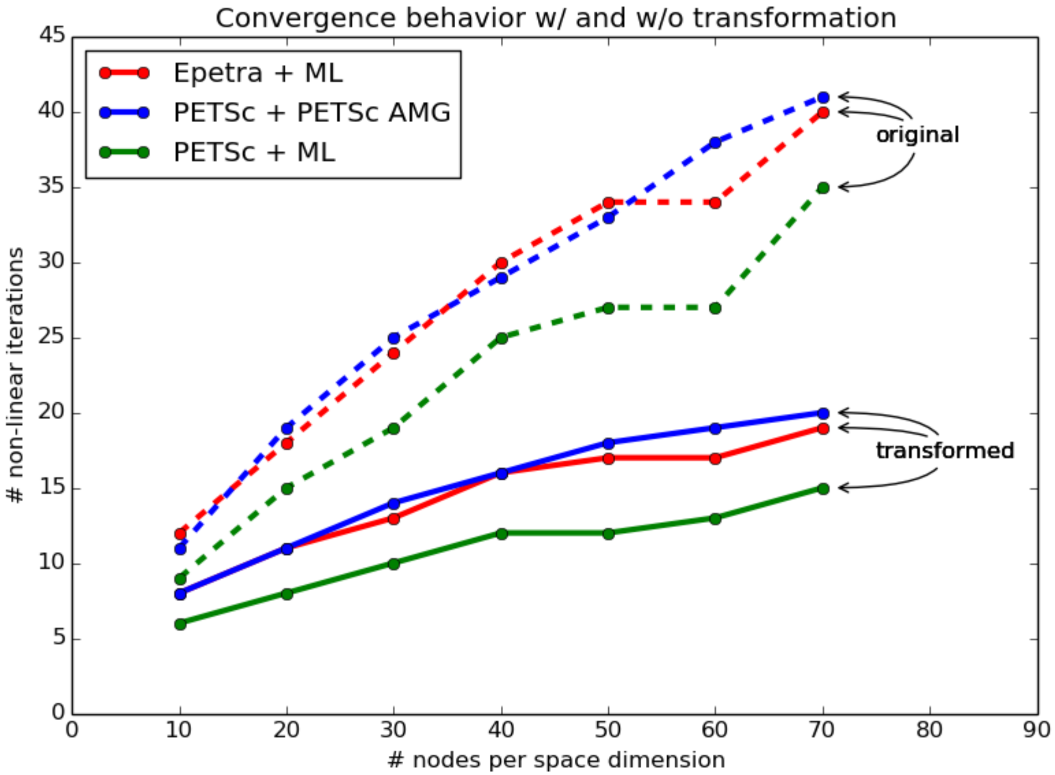


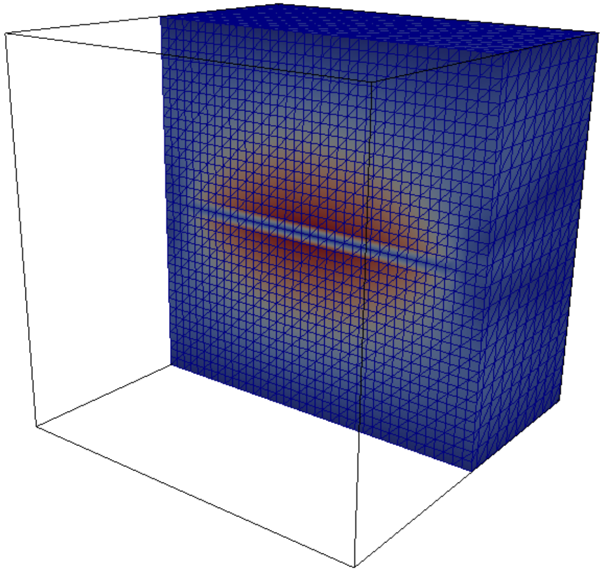
- Setup
  - 100x100x100 unit domain
  - u fixed at top and bottom
  - Circular crack at center, r=90
  - pressure on crack surface

- Discretization
  - Linear Lagrange shape functions → tet4
  - n^3 nodes

- Solver
  - Default GMRES Krylov solver and AMG smoothers,
  - Convergence criteria: rel. error < 1.0e-8
  - Minimum configuration:
    1. mlPC->set("aggregation: threshold", 1.0e-7);
    2. petscPC->parameters("gamg")["threshold"] = 1.0e-7;
    3. petscPC->parameters("ml")["threshold"] = 1.0e-7;

Preliminary results - 3d implementation

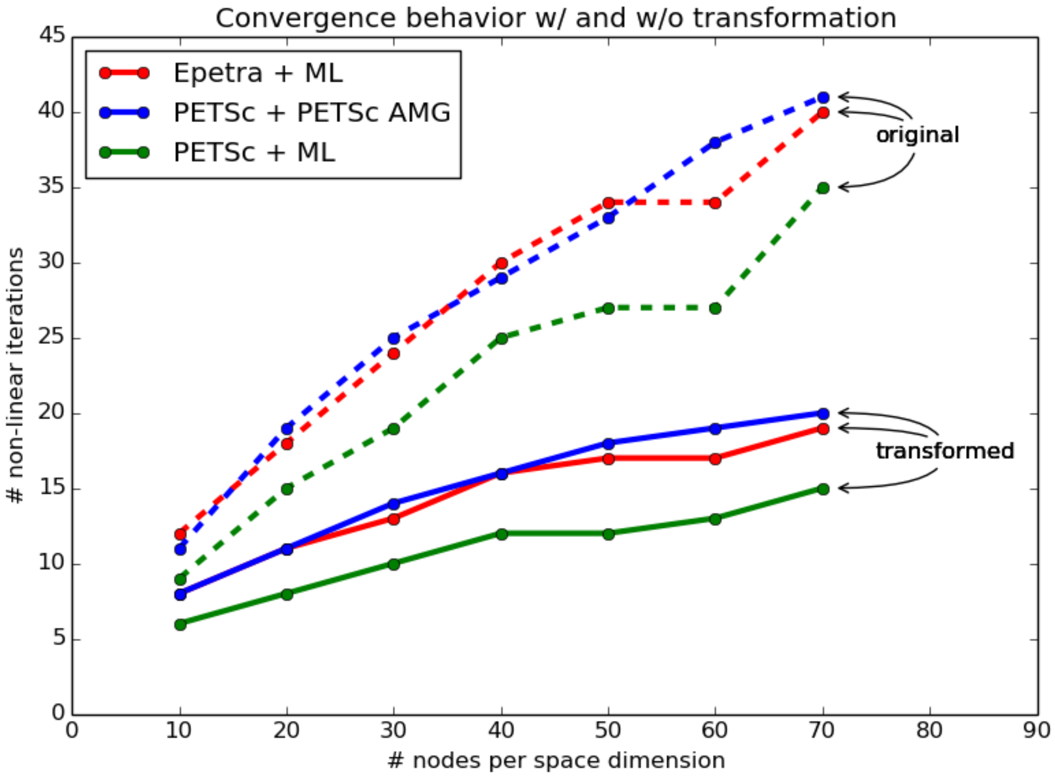


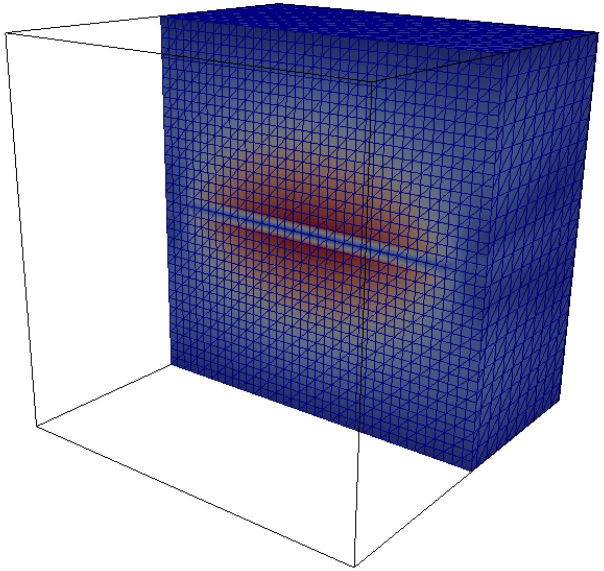


Comments:

- performance gain ~ problem size
- performance gain ~ crack size
  - no cracks → standard FE (not shown here)

Preliminary results - 3d implementation





Comments:

- transformation related computation times relatively small
- preconditioners must be tweaked individually
  - even better performance with block-smoother → ToDo

Summary - 3d implementation

Summary:

3D FEniCS implementation works as expected based on theory and 2d results!

ToDos:

- adapt AMG configurations to problem (Block-GS, aggr. …)
- parallelization
- branching cracks
- replace matrix multiplication with “implicit G”

More details on the method and its properties:

A. Gerstenberger and R. Tuminaro, An algebraic multigrid approach to solve extended finite element method based fracture problems, *Int. J. Numer. Meth. Engng.*, Vol. **94(3)**, 248--272, 2013, DOI: 10.1002/nme.4442

More about the 3d FEniCS implementation:

→ source code will be in FEniCS dev at some point