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.)

- Overview & Motivation
  - A brief review of XFEM
    - Why does standard SA-AMG fail & how to fix it
      - Examples
        - Conclusion

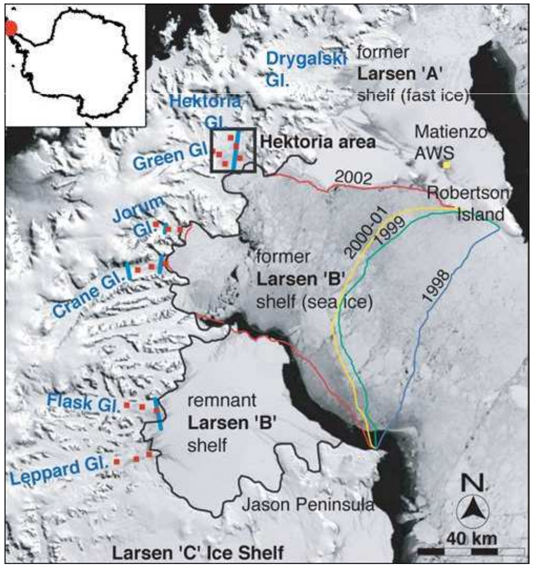
Sandia is a multiprogram laboratory managed and operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Fracture of ice

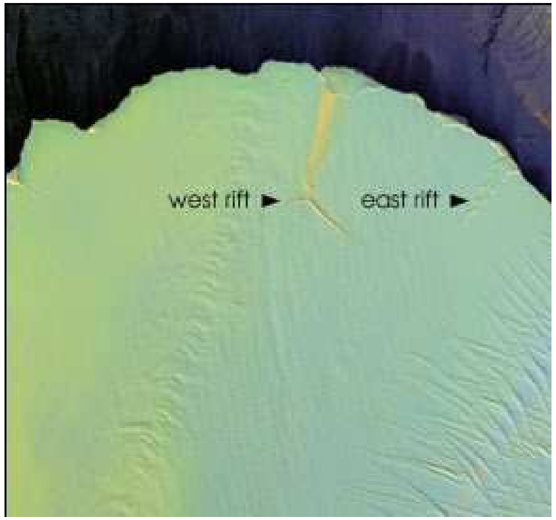
Objective: Employ parallel computers to better understand how fracture of land ice affects the global climate. Fracture happens e.g. during

- 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.

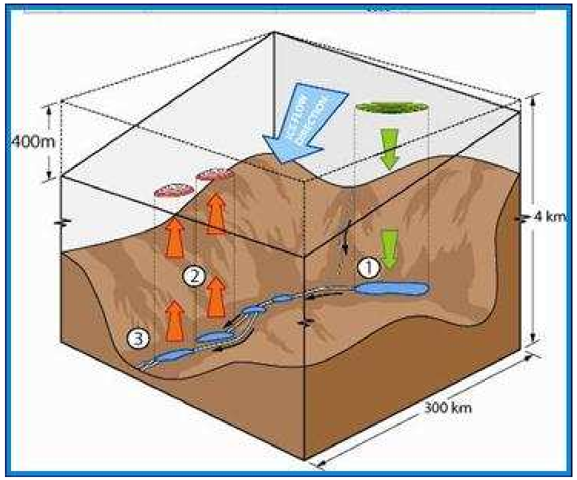
Ice shelves in Antarctica:



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



Amery ice shelf

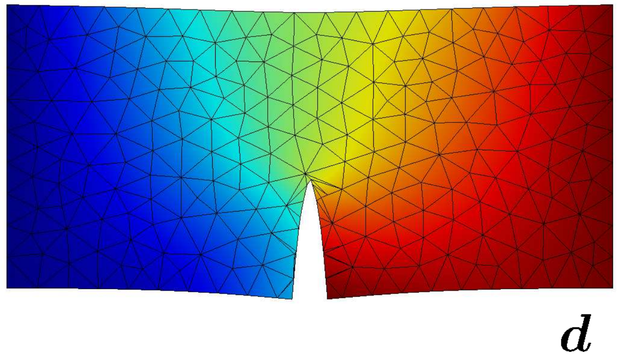


Glacial hydrology
(Source: http://www.sale.scar.org)

Computational Modeling of Fracture

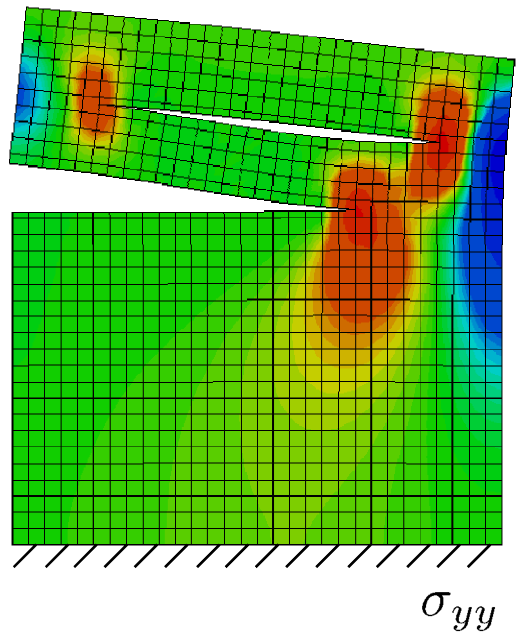
Classical FEM approach to fracture mechanics

- Mesh conforms to crack boundaries
- Crack propagation → remeshing at each step
  - Requires fine mesh for tip singularities
  - Mesh smoothing for ‘ugly’ elements



eXtended Finite Element Method (XFEM)*

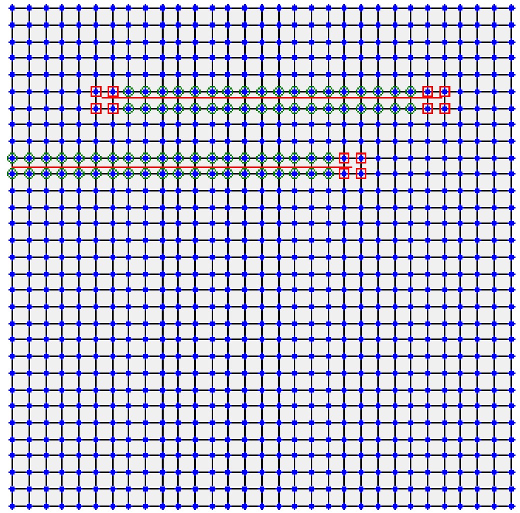
- Base mesh independent of crack geometry
- Crack propagation → adding “enriched” DOF with special basis functions to existing nodes
  - Number of DOFs change, mesh does not



** Belytschko & Black (1999), Moës et al. (1999)*

XFEM Formulation for Cracks

Displacement approximation (shifted basis form.)



$$
\begin{aligned}
u^h(\mathrm{x}) &= \sum_{I=1}^{n} N_I(\mathrm{x}) u_I \\
&\quad + \sum_{i=1}^{n_h} N_{I_i}(\mathrm{x}) \left(H(\mathrm{x}) - H\left(\mathrm{x}_{I_i}\right)\right) a_{I_i} \\
&\quad + \sum_{i=1}^{n_f} N_{\hat{I}_i}(\mathrm{x}) \sum_{J=1}^{n_J} \left(F_J(\mathrm{x}) - F_J\left(\mathrm{x}_{\hat{I}_i}\right)\right) b_{\hat{I}_i J}
\end{aligned}
$$

- 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: use identical approximation for test function $\delta\boldsymbol{d}(\boldsymbol{x})$

→ Symmetric global system

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

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

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

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

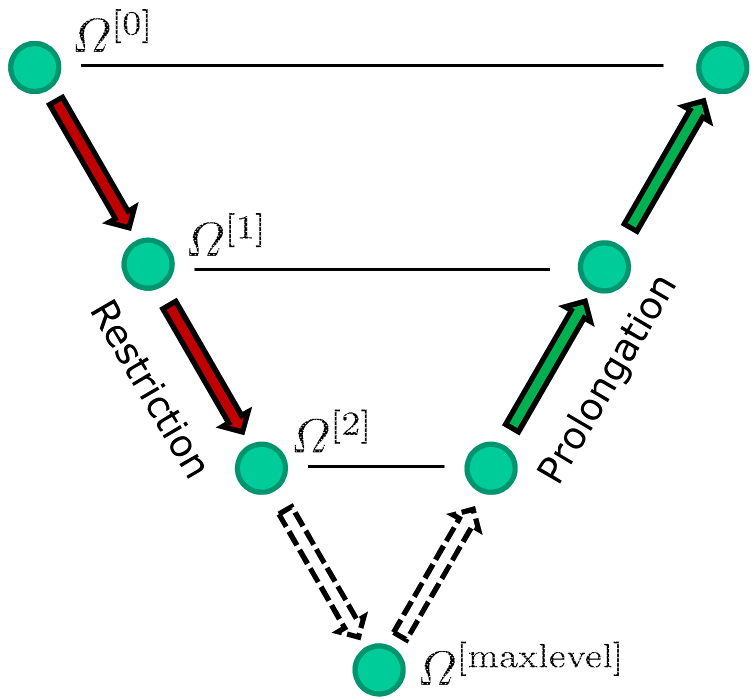
Sandia National Laboratories

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 $R^{[k]}$ and $P^{[k]}$
- **In AMG**, transfer operators are obtained from **graph information of *A***
- Interpolation complements relaxation

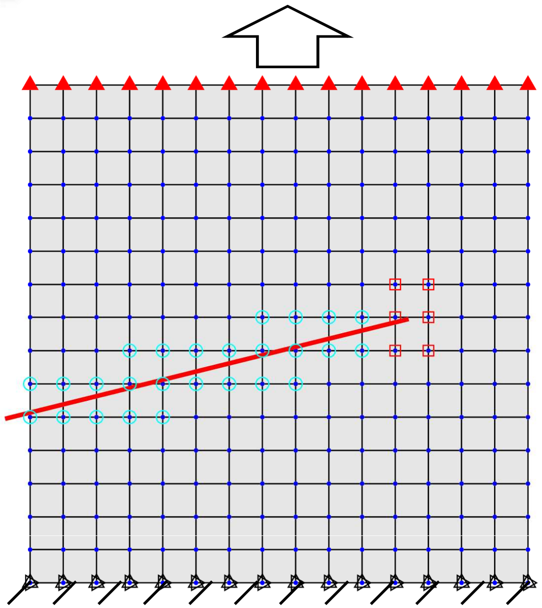
solve $Au=b$ using recursive multilevel V Cycle:

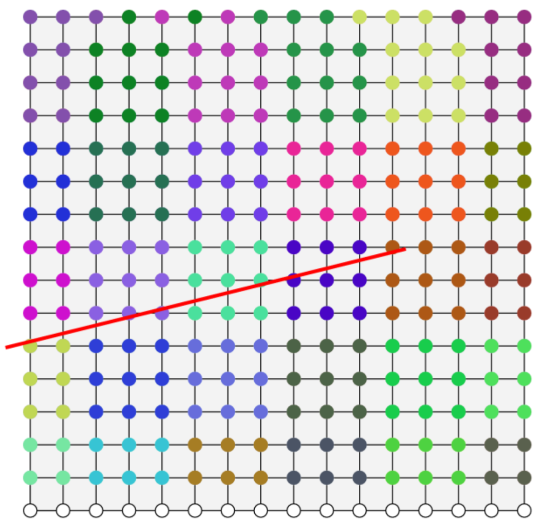
$$
\begin{array}{l}
\textbf{function} \quad \boldsymbol{u} \leftarrow \mathrm{multilevel}(\boldsymbol{b},\ \boldsymbol{u},\ k) \\
\boldsymbol{u} = \mathrm{S}^{[k]}_{\mathrm{PreSmoother}}(\boldsymbol{A}^{[k]},\ \boldsymbol{b},\ \boldsymbol{u}) \\
\textbf{if} \ k < \mathrm{maxlevel} \ \textbf{then} \\
\quad \boldsymbol{b}_{\mathrm{c}} = \boldsymbol{R}^{[k]} \cdot (\boldsymbol{b} - \boldsymbol{A}^{[k]}\boldsymbol{u}) \\
\quad \boldsymbol{u}_{\mathrm{c}} = \mathrm{multilevel}(\boldsymbol{b}_{\mathrm{c}},\ \mathbf{0},\ k+1) \\
\quad \boldsymbol{u} = \boldsymbol{u} + \boldsymbol{P}^{[k]} \cdot \boldsymbol{u}_{\mathrm{c}} \\
\quad \boldsymbol{u} = \mathrm{S}^{[k]}_{\mathrm{PostSmoother}}(\boldsymbol{A}^{[k]},\ \boldsymbol{b},\ \boldsymbol{u}) \\
\textbf{return} \quad \boldsymbol{u}
\end{array}
$$

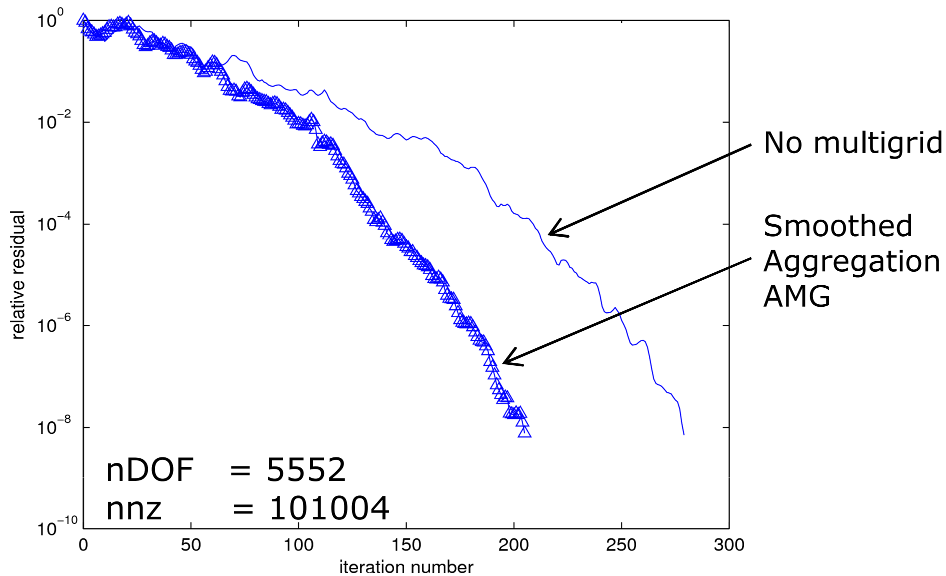


- iterative smoother on finest and intermediate levels
- direct solve at the coarsest level

‘Standard’ SA-AMG for fracture problems



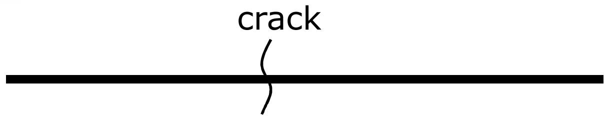




Possible issues:

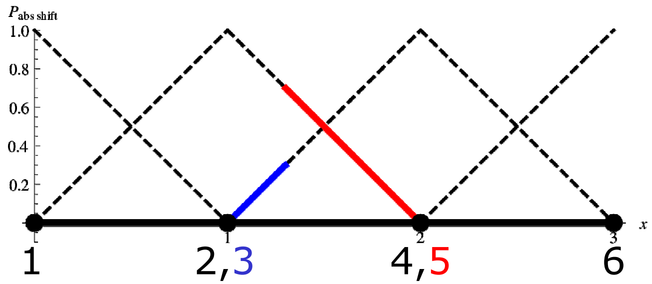
- Aggregation
  - Aggregates should not cross crack
- Nullspace
  - Elasticity: 3 ZEMs
  - Uncoupled domains: 6 ZEMs? or more?
- Assumption of 2 unknowns per node fails
  - 2, 4, or 10 DOFs per node

Distinct region representation



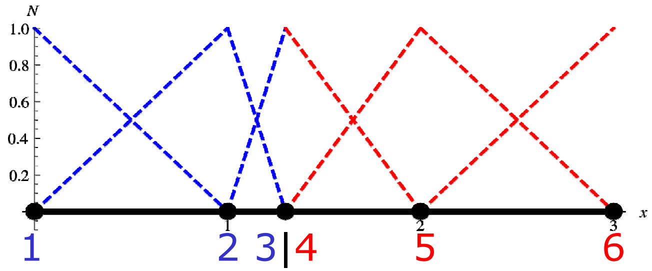
K M

XFEM: modified shifted enrichment $\sum_I N_I(x)\,|H(x) - H(x_I)|\,a_I$



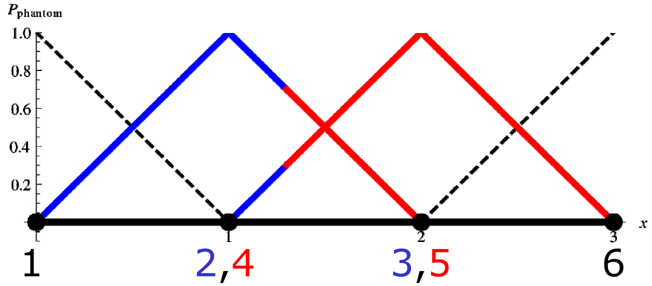
$$\frac{EA}{2h_1}\begin{bmatrix} 2 & -2 & 0 & 0 & 0 & 0 \\ -2 & 4 & 1 & -2 & -1 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 \\ 0 & -2 & -1 & 4 & 1 & -2 \\ 0 & -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & -2 & 0 & 2 \end{bmatrix} \qquad \frac{\rho A h_1}{24}\begin{bmatrix} 8 & 4 & 0 & 0 & 0 & 0 \\ 4 & 16 & 1 & 4 & 2 & 0 \\ 0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

FEM



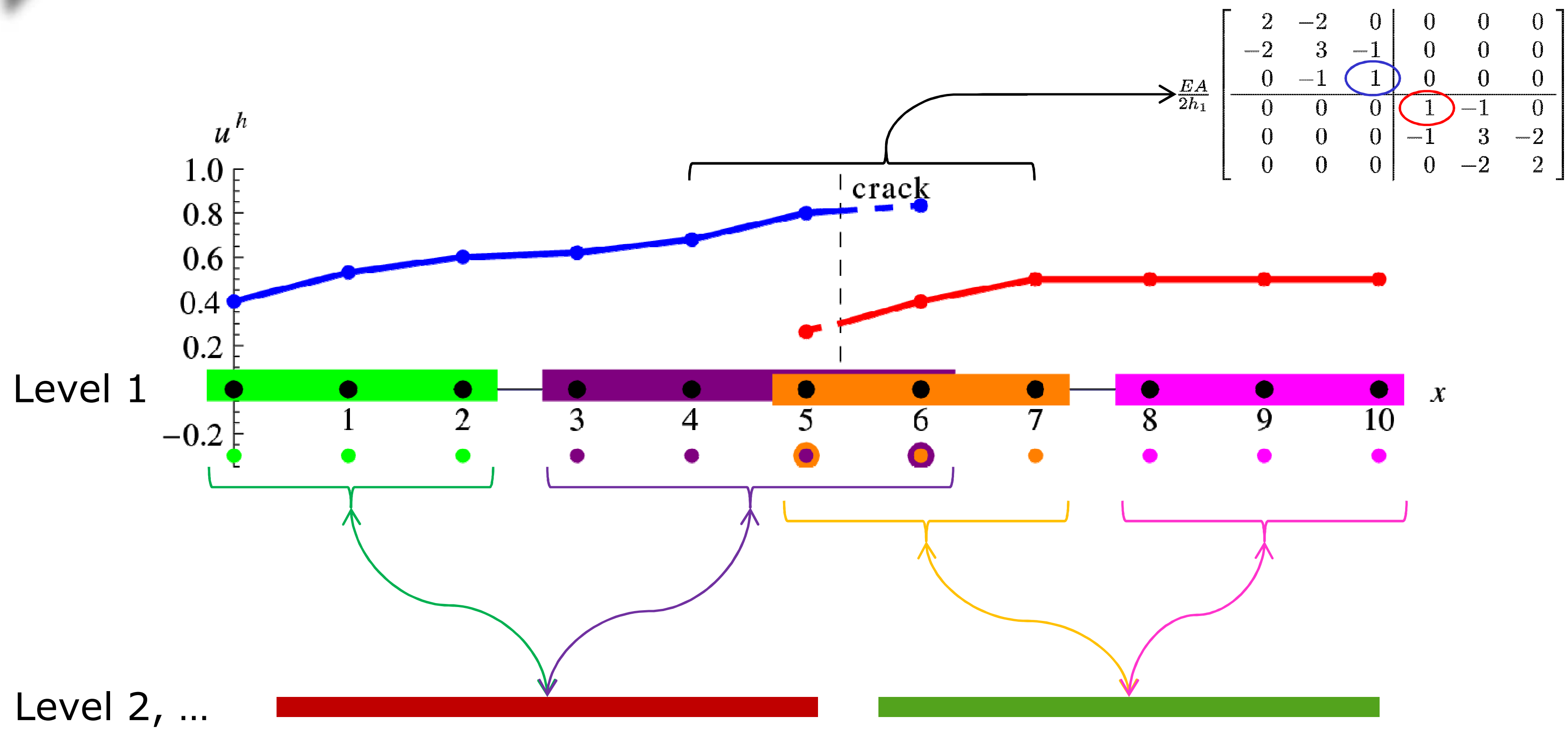
$$\frac{EA}{2h_1}\left[\begin{array}{ccc|ccc} 2 & -2 & 0 & 0 & 0 & 0 \\ -2 & 6 & -4 & 0 & 0 & 0 \\ 0 & -4 & 4 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 4 & -4 & 0 \\ 0 & 0 & 0 & -4 & 6 & -2 \\ 0 & 0 & 0 & 0 & -2 & 2 \end{array}\right] \qquad \frac{\rho A h_1}{24}\left[\begin{array}{ccc|ccc} 8 & 4 & 0 & 0 & 0 & 0 \\ 4 & 12 & 2 & 0 & 0 & 0 \\ 0 & 2 & 4 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 4 & 2 & 0 \\ 0 & 0 & 0 & 2 & 12 & 4 \\ 0 & 0 & 0 & 0 & 4 & 8 \end{array}\right]$$

Phantom node approach



$$\frac{EA}{2h_1}\left[\begin{array}{ccc|ccc} 2 & -2 & 0 & 0 & 0 & 0 \\ -2 & 3 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 3 & -2 \\ 0 & 0 & 0 & 0 & -2 & 2 \end{array}\right] \qquad \frac{\rho A h_1}{24}\left[\begin{array}{ccc|ccc} 8 & 4 & 0 & 0 & 0 & 0 \\ 4 & 15 & 2 & 0 & 0 & 0 \\ 0 & 2 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 2 & 15 & 4 \\ 0 & 0 & 0 & 0 & 4 & 8 \end{array}\right]$$

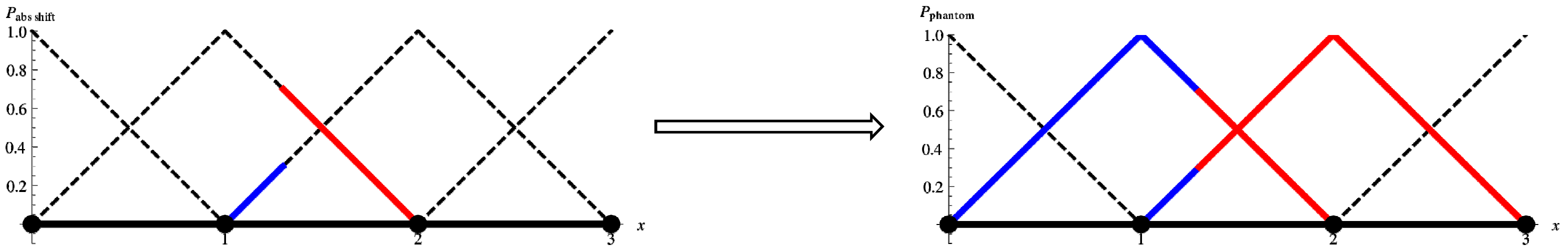
Aggregation for phantom nodes: 1D



Aggregates seemingly overlap, but are **not** connected on any level!

Transformation

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: $\phi_I - \bar{\phi}_I = \phi_\alpha$

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

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

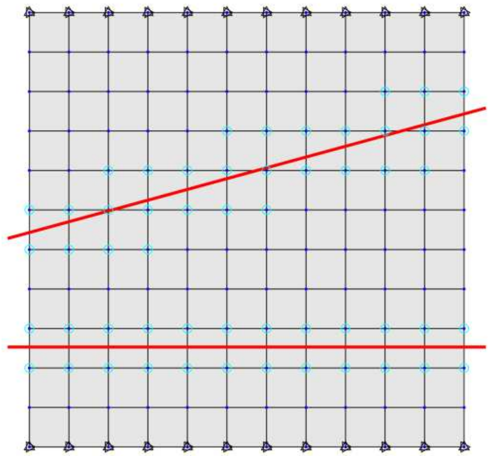
$$\boldsymbol{G}^{\mathrm{T}} = \begin{array}{cc} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} & \begin{array}{c} u_1 \\ u_2 \\ a_3 \\ a_2 \\ u_3 \\ u_4 \end{array} \end{array}$$

(similar: Menouillard 2008, …)

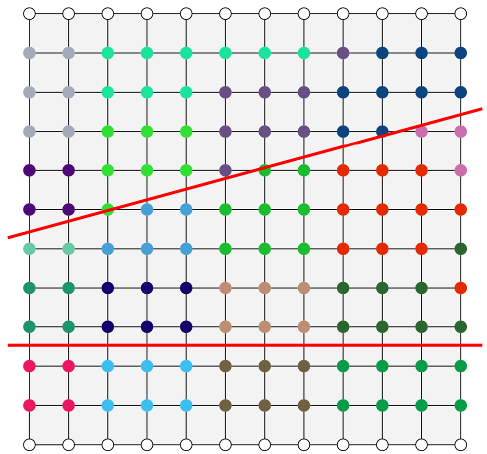
G

- is extremely sparse,
- is simple to produce,
- transformations can be processor-local, and
- exists for higher order Lagrange Polynomials and multiple dimensions.

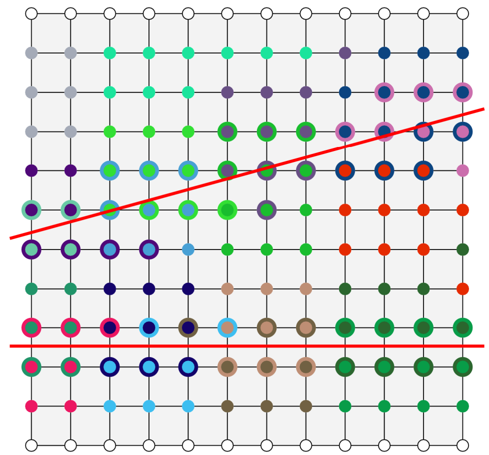
Aggregation for phantom nodes: 2D



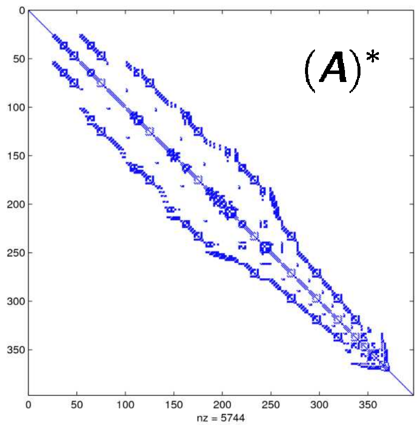
Mesh + BC + Enrichment



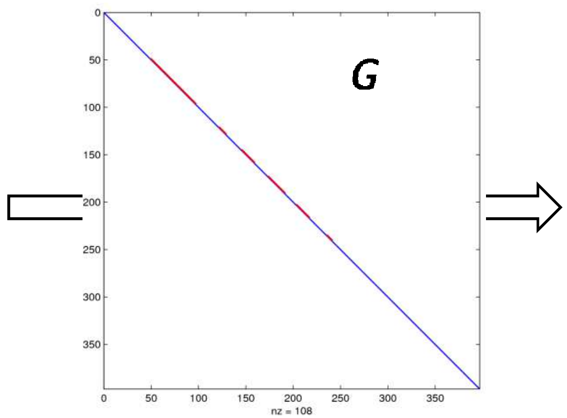
Standard DOFs only



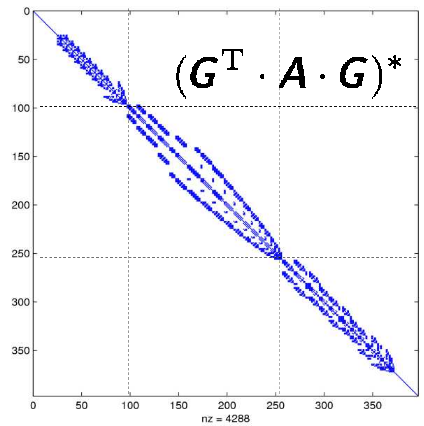
Standard + Phantom DOFs



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

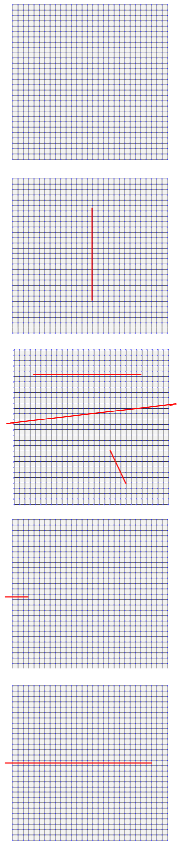
Sandia National Laboratories

Prelim. results for jump enrichments only

CG preconditioned with AMG

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

If one wants to use the standard graph-based aggregation, then using Phantom node setup is crucial!



Case	$n_e \times n_e$	$\alpha_{\text{cond.}}$	n_{iter}			
			$\boldsymbol{A}$		$\boldsymbol{G}^{\mathrm{T}} \cdot \boldsymbol{A} \cdot \boldsymbol{G}$	
			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

NullSpace for Jump & Tip Enrichments

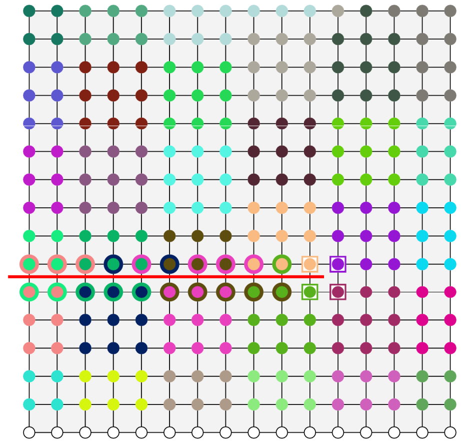
2D Elasticity problem has 3 Zero Energy Modes (ZEMs):

	1	2	3
		...	
d_{xI}	1	0	-y_I
d_{yI}	0	1	x_I
		...	

Nullspace for phantom node setup

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

- *Extra tip DOFs don’t contribute to rigid body motion*
  - Put 0 into their respective rows
  - → no coarse level contribution in prolongation & restriction
  - → smoothing only on finest level (fine scale feature)

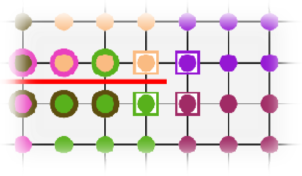


$$+\sum_{i=1}^{n_f} N_{\hat{I}_i}(\mathrm{x}) \sum_{J=1}^{n_J} \left(F_J(\mathrm{x}) - F_J\left(\mathrm{x}_{\hat{I}_i}\right)\right) b_{\hat{I}_i J} \qquad F_J(r,\theta) = \left\{ \overbrace{\sqrt{r}\sin\left(\frac{\theta}{2}\right)}^{J=1}, \overbrace{\sqrt{r}\cos\left(\frac{\theta}{2}\right)}^{J=2}, \overbrace{\sqrt{r}\sin\left(\frac{\theta}{2}\right)\sin(\theta)}^{J=3}, \overbrace{\sqrt{r}\cos\left(\frac{\theta}{2}\right)\sin(\theta)}^{J=4} \right\}$$

Sandia National Laboratories

Smoothing

- Finest Level: Use special tip smoother D^{tip} in addition to standard (Block-) Gauss-Seidel smoothing



Reason for special smoothing:
- dense blocks (40x40 for quad4)
- high condition number

  – Tip smoother: direct solve for each small tip block
  – Pre-smoother

$$\boldsymbol{u} \leftarrow \mathrm{GaussSeidel}(\boldsymbol{u}, \tilde{\boldsymbol{A}}, \boldsymbol{b})$$

$$\boldsymbol{u} \leftarrow \boldsymbol{u} + \boldsymbol{D}^{\text{tip}} \cdot (\boldsymbol{b} - \tilde{\boldsymbol{A}} \cdot \boldsymbol{u})$$

  - Post-smoother

$$\boldsymbol{u} \leftarrow \boldsymbol{u} + \boldsymbol{D}^{\text{tip}} \cdot (\boldsymbol{b} - \tilde{\boldsymbol{A}} \cdot \boldsymbol{u})$$

$$\boldsymbol{u} \leftarrow \mathrm{GaussSeidel}(\boldsymbol{u}, \tilde{\boldsymbol{A}}, \boldsymbol{b})$$

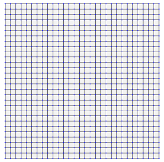
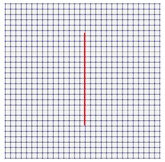
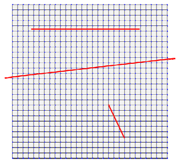
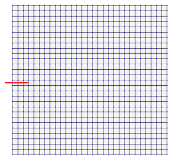
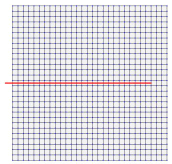
Pre-Post-smoother symmetry is important

- All coarser levels: standard (Block-) Gauss-Seidel
- Coarsest Level: standard direct solve

Numerical Results for full XFEM system

CG preconditioned with AMG

Special tip smoother is essential to deal with tip enrichments!

Case	$n_e \times n_e$	$\alpha_{\text{cond.}}$	n_{iter}				
			1L	ML	ML, NS	ML, MS	ML, MS, NS
I	30×30	3e+03	32	9	9	9	9
	60×60	1e+04	63	10	10	10	10
	90×90	3e+04	93	11	11	11	11
	120×120	5e+04	123	11	11	11	11
II	30×30	2e+07	115	84	75	21	18
	60×60	8e+08	-	115	97	24	20
	90×90	8e+09	-	141	114	27	23
	120×120	3e+10	-	-	143	28	23
III	30×30	5e+07	143	122	94	24	18
	60×60	1e+09	-	180	158	27	20
	90×90	2e+10	-	-	-	29	20
	120×120	3e+10	-	-	-	37	26
1a	30×30	6e+05	66	31	31	16	16
	60×60	3e+06	117	31	31	18	18
	90×90	1e+07	165	33	33	20	20
	120×120	2e+07	-	32	32	19	19
1c	30×30	1e+08	86	34	34	21	20
	60×60	7e+08	157	35	35	23	23
	90×90	2e+09	-	35	35	24	24
	120×120	3e+09	-	37	37	26	26

Operator complexity: 1.28-1.40

Concluding Remarks

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

Key components:

- System matrix must be in phantom-node form
  - Either you already have it, (voids, fluid-structure interaction, …) , or
  - do a simple transformation $\boldsymbol{G}^{\mathrm{T}} \cdot \boldsymbol{A} \cdot \boldsymbol{G} \cdot \boldsymbol{G}^{-1} \cdot \boldsymbol{u} = \boldsymbol{G}^{\mathrm{T}} \cdot \boldsymbol{f}$
- Adapt nullspace with zero entries for extra tip DOFs
- Two-step smoothing on finest level

Future Directions

- What happens to tiny element fractions (conditioning)?
- 3d implementation