SAND2013-5320C

On Parallelization of Energy Minimizing Multigrid

Andrey Prokopenko

Jeremie Gaidamour, Jonathan Hu, Ray Tuminaro

SIAM Annual Meeting
8 July 2013

Sandia National Laboratories

Sandia National Laboratories is a multi-program laboratory operated by Sandia Corp, 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.

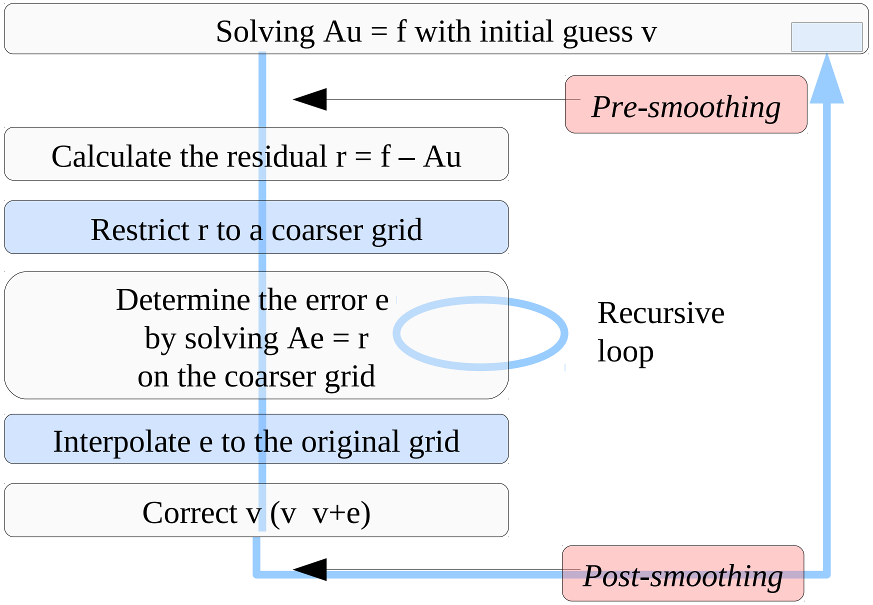
Sandia National Laboratories

Outline

- Introduction
- Energy-minimization based AMG
  - Motivations
  - Algorithm
- Parallel implementation
- Setup amortization
- Conclusion

AMG

- Iterative method for solving linear equations
- Commonly used as a preconditioner
- Idea: capture error at multiple resolutions using grid transfer operator:
  - **Smoothing** damps the oscillatory error (high energy)
  - **Coarse grid correction** reduces the smooth error (low energy)



Solving Au = f with initial guess v

Pre-smoothing

Calculate the residual r = f – Au

Restrict r to a coarser grid

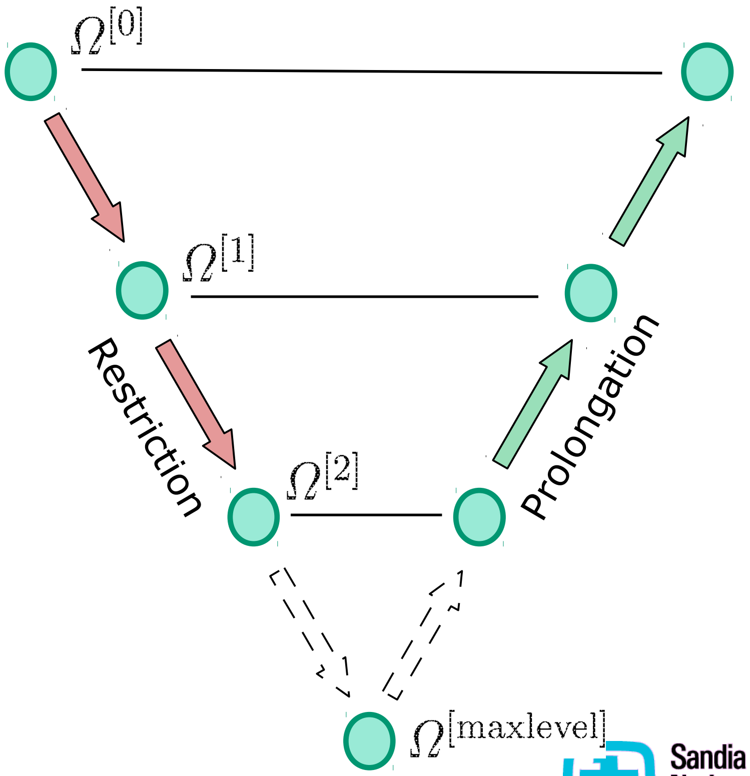
Determine the error e by solving Ae = r on the coarser grid

Recursive loop

Interpolate e to the original grid

Correct v (v v+e)

Post-smoothing



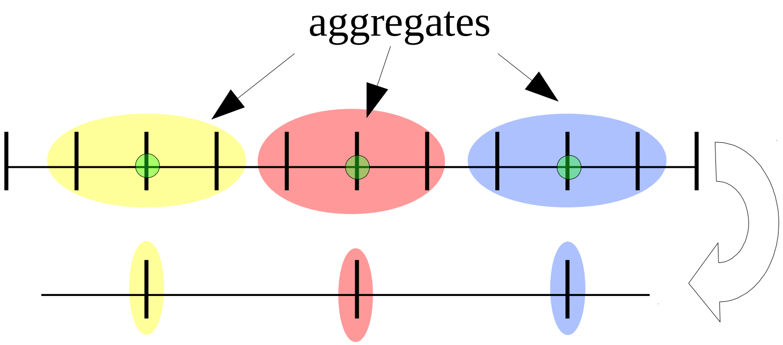
Prolongator requirements

Few desired properties

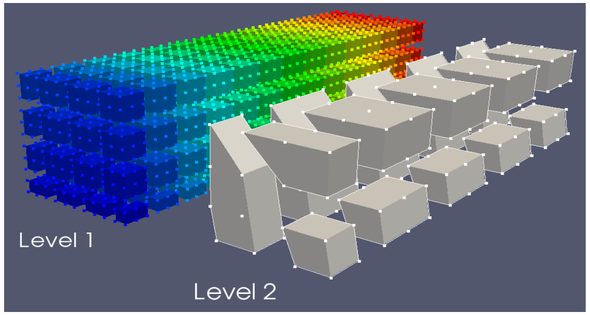
- **preservation of null space**: the span of basis functions on each coarse level should contain zero energy modes
- **minimization of energy**: basis functions on the coarse levels should have as small energy as possible
- **bounded intersection:** the supports of the basis functions on the coarse levels should overlap as little as possible.

Smoothed Aggregation

SA prolongator is constructed in a few steps



- Construct aggregates
  - Select a set of root nodes ●
  - Group unknowns into aggregates



- Construct tentative prolongator and coarse nullspace
  - Restrict fine nullspace onto aggregates
  - Do QR decomposition

    We satisfy $\mathrm{P}_{tent} B_c = B$

$$B=\begin{bmatrix}1\\1\\1\\1\\1\\1\\\vdots\end{bmatrix} \Rightarrow \begin{bmatrix}1 & & \\ 1 & & \\ 1 & & \\ & 1 & \\ & 1 & \cdots \\ & 1 & \\ & & \ddots\end{bmatrix}$$

- Decrease energy of P_{tent} by smoothing

  $\mathrm{P} = (\mathrm{I} - \omega D^{-1} A) P_{tent}$

  May not satisfy $\mathrm{P}_{SA} B_c = B$

Energy minimization

Energy minimization

Energy minimization is a general framework.

Idea: construct the prolongator P by minimizing the energy of each column P_k while enforcing constraints.

Find P:

$$\mathrm{P} = \operatorname{argmin} \sum \|P_k\|_{\mathcal{X}}$$

subject to

- specified sparsity pattern;
- nullspace preservation.

Advantages:

- Flexibility (input):
  - accept any sparsity pattern (arbitrary basis function support)
  - enforce constraints: important modes requiring accurate interpolation
  - choice of norm for minimization and search space
- Robustness

Constraint matrix

- Sparsity pattern
- B, B_c fine and coarse mode(s) requiring accurate interpolation

inputs

Preservation of the nullspace: for instance $\mathrm{P}\mathbf{1} = \mathbf{1}$

$$N = \begin{bmatrix} * & * \\ * & 0 \\ * & * \\ 0 & * \end{bmatrix} \qquad \mathrm{PB}_c = B \Leftrightarrow \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \\ p_{31} & p_{32} \\ p_{41} & p_{42} \end{bmatrix} \begin{bmatrix} b^c_{11} \\ b^c_{21} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{41} \end{bmatrix}$$

- Representation of the constraints in the algorithm:

$$\mathrm{XP} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ b^c_{11} & 0 & 0 & 0 & b^c_{21} & 0 & 0 & 0 \\ 0 & b^c_{11} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b^c_{11} & 0 & 0 & 0 & b^c_{21} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & b^c_{21} \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \\ p_{41} \\ p_{12} \\ p_{22} \\ p_{32} \\ p_{42} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ b_{11} \\ b_{21} \\ b_{31} \\ b_{41} \end{bmatrix}$$

Constraint matrix

Two nullspace vectors:

$$P \begin{bmatrix} b^c_{11} & b^c_{12} \\ b^c_{21} & b^c_{22} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \end{bmatrix}$$

$$\begin{bmatrix} b^c_{11} & 0 & 0 & b^c_{21} & 0 & 0 \\ 0 & b^c_{11} & 0 & 0 & 0 & 0 \\ 0 & 0 & b^c_{11} & 0 & b^c_{21} & 0 \\ 0 & 0 & 0 & 0 & 0 & b^c_{21} \\ b^c_{12} & 0 & 0 & b^c_{22} & 0 & 0 \\ 0 & b^c_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & b^c_{12} & 0 & b^c_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & b^c_{22} \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \\ p_{12} \\ p_{32} \\ p_{42} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{41} \\ b_{12} \\ b_{22} \\ b_{32} \\ b_{42} \end{bmatrix}$$

Energy-minimization algorithm

Find P:

$$\mathrm{P} = \operatorname{argmin} \sum \|P_k\|_{\mathcal{X}}$$

subject to

- specified sparsity pattern;
- nullspace preservation.

Solve AP = 0
in a constrained Krylov space

- Definition of energy $\|\cdot\|_{\mathcal{X}}$ depends on Krylov method
  - A for CG
  - $A^T A$ for GMRES

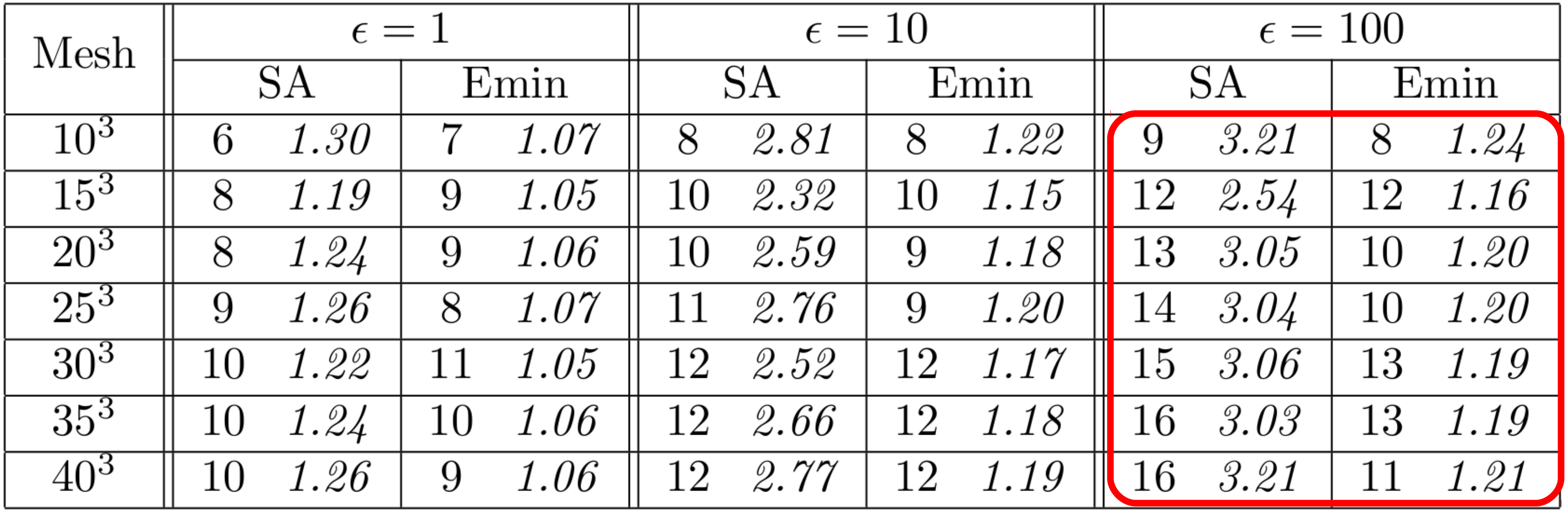
Energy minimization algorithm

```
Construct aggregates
$\mathcal{N} = |A||P^{(0)}|$                          ▷ Select sparsity pattern

$D = diag(A)$                                         ▷ Diagonal preconditioner
$R = -AP^{(0)}$                                       ▷ Initial residual
$R = \textbf{enforce}(R, \mathcal{N})$                ▷ Enforce sparsity on $R$
$R = \textbf{project}(R, X)$                          ▷ Enforce $RB_c = \mathbf{0}$

for $i$ to iter do
    $Z = D^{-1}R$
    $\gamma =< R, Z >_F$
    if $i$ is 1 then
        $Y = Z$
    else
        $\beta = \gamma / \gamma_{old};$
        $Y = Z + \beta Y$                             ▷ New search direction
    end if
    $\gamma_{old} = \gamma$
    $Y_A = AY$
    $Y_A = \textbf{enforce}(Y_A, \mathcal{N})$        ▷ Enforce sparsity on $Y_A$
    $Y_A = \textbf{project}(Y_A, B_c)$                ▷ Enforce $Y_A B_c = \mathbf{0}$
    $\alpha = \gamma / < Y, Y_A >_F$
    $P^{(i)} = P^{(i-1)} + \alpha Y$                  ▷ Update prolongator
    $R = R - \alpha Y_A$                              ▷ Update residual
```

Comparison with Smoothed Aggregation

- SA: 6 DOFs/node
- Energy Minimization: 3 DOFs/node, 6 nullspace vectors

Tab. : Iteration count and *complexity* (lower complexity = faster run time) for increasing mesh sizes and stretch factors.



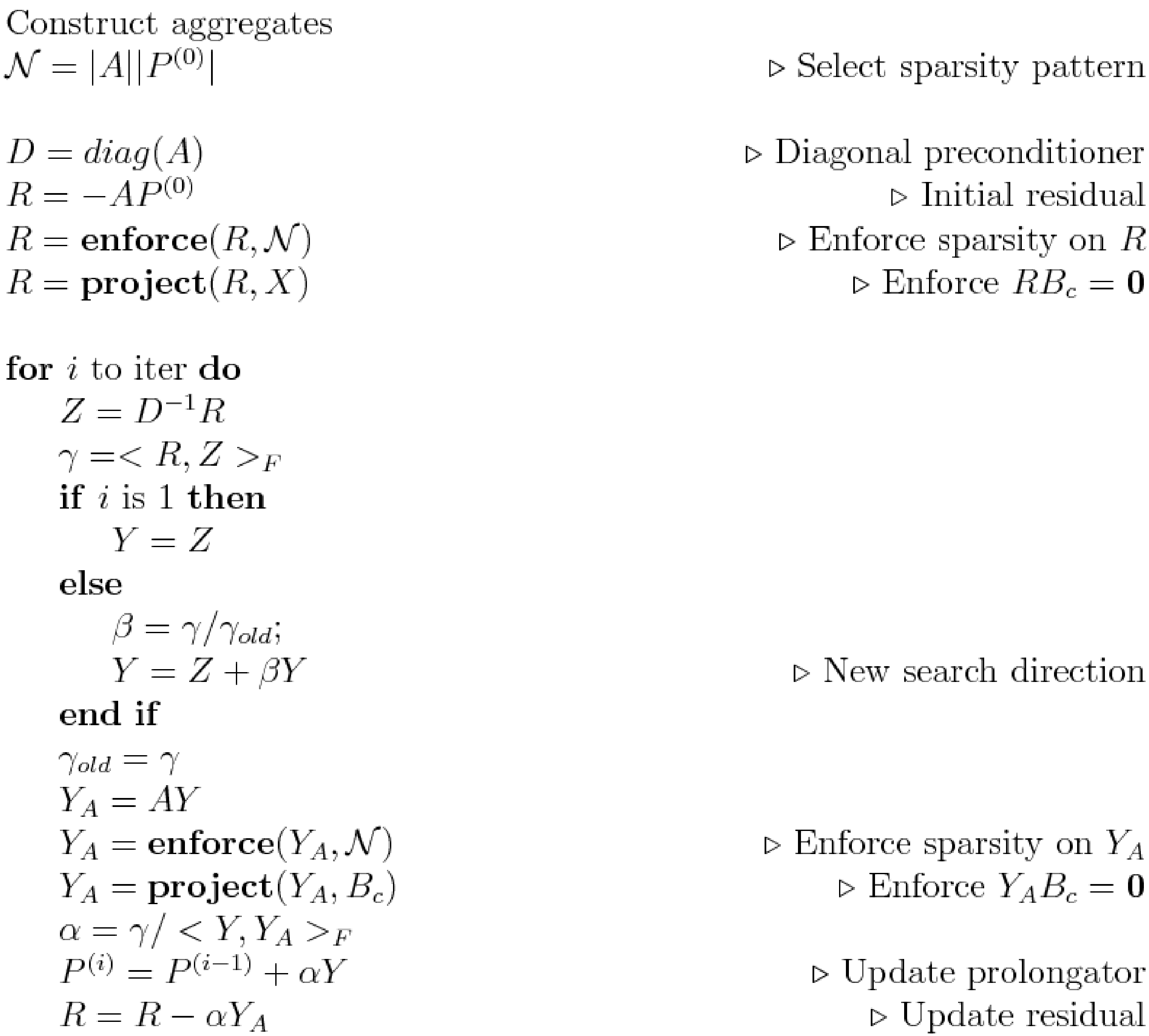
Mesh	$\epsilon = 1$				$\epsilon = 10$				$\epsilon = 100$			
	SA		Emin		SA		Emin		SA		Emin	
10^3	6	*1.30*	7	*1.07*	8	*2.81*	8	*1.22*	9	*3.21*	8	*1.24*
15^3	8	*1.19*	9	*1.05*	10	*2.32*	10	*1.15*	12	*2.54*	12	*1.16*
20^3	8	*1.24*	9	*1.06*	10	*2.59*	9	*1.18*	13	*3.05*	10	*1.20*
25^3	9	*1.26*	8	*1.07*	11	*2.76*	9	*1.20*	14	*3.04*	10	*1.20*
30^3	10	*1.22*	11	*1.05*	12	*2.52*	12	*1.17*	15	*3.06*	13	*1.19*
35^3	10	*1.24*	10	*1.06*	12	*2.66*	12	*1.18*	16	*3.03*	13	*1.19*
40^3	10	*1.26*	9	*1.06*	12	*2.77*	12	*1.19*	16	*3.21*	11	*1.21*

3.85x

complexity: $\dfrac{\sum_i nnz(A_i)}{nnz(A)}$

Parallel implementation

Energy minimization algorithm



```
Construct aggregates
$\mathcal{N} = |A||P^{(0)}|$                          ▷ Select sparsity pattern

$D = diag(A)$                                         ▷ Diagonal preconditioner
$R = -AP^{(0)}$                                       ▷ Initial residual
$R = \textbf{enforce}(R, \mathcal{N})$                ▷ Enforce sparsity on $R$
$R = \textbf{project}(R, X)$                          ▷ Enforce $RB_c = \mathbf{0}$

for $i$ to iter do
    $Z = D^{-1}R$
    $\gamma =< R, Z >_F$
    if $i$ is 1 then
        $Y = Z$
    else
        $\beta = \gamma/\gamma_{old};$
        $Y = Z + \beta Y$                             ▷ New search direction
    end if
    $\gamma_{old} = \gamma$
    $Y_A = AY$
    $Y_A = \textbf{enforce}(Y_A, \mathcal{N})$        ▷ Enforce sparsity on $Y_A$
    $Y_A = \textbf{project}(Y_A, B_c)$                ▷ Enforce $Y_A B_c = \mathbf{0}$
    $\alpha = \gamma / < Y, Y_A >_F$
    $P^{(i)} = P^{(i-1)} + \alpha Y$                  ▷ Update prolongator
    $R = R - \alpha Y_A$                              ▷ Update residual
```

Energy minimization algorithm

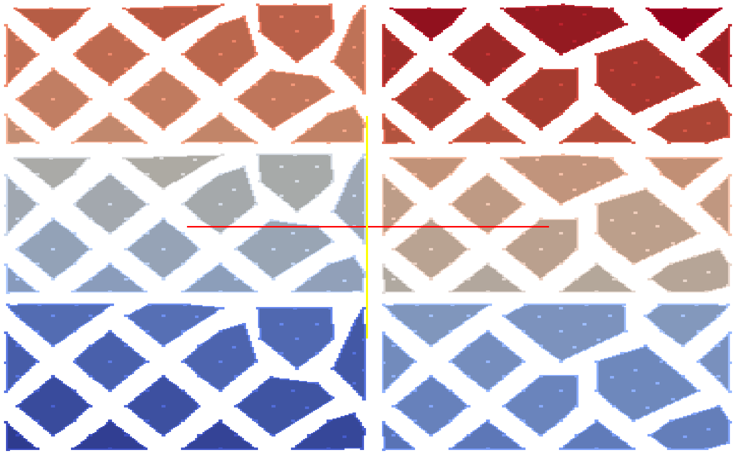
```
Construct aggregates
$\mathcal{N} = |A||P^{(0)}|$                          ▷ Select sparsity pattern

$D = diag(A)$                                         ▷ Diagonal preconditioner
$R = -AP^{(0)}$                                       ▷ Initial residual
$R =$ enforce$(R, \mathcal{N})$                       ▷ Enforce sparsity on $R$
$R =$ project$(R, X)$                                 ▷ Enforce $RB_c = \mathbf{0}$

for $i$ to iter do
    $Z = D^{-1}R$
    $\gamma =< R, Z >_F$
    if $i$ is 1 then
        $Y = Z$
    else
        $\beta = \gamma/\gamma_{old};$
        $Y = Z + \beta Y$                             ▷ New search direction
    end if
    $\gamma_{old} = \gamma$
    $Y_A = AY$
    $Y_A =$ enforce$(Y_A, \mathcal{N})$               ▷ Enforce sparsity on $Y_A$
    $Y_A =$ project$(Y_A, B_c)$                       ▷ Enforce $Y_A B_c = \mathbf{0}$
    $\alpha = \gamma / < Y, Y_A >_F$
    $P^{(i)} = P^{(i-1)} + \alpha Y$                  ▷ Update prolongator
    $R = R - \alpha Y_A$                              ▷ Update residual
```

Parallel aggregation

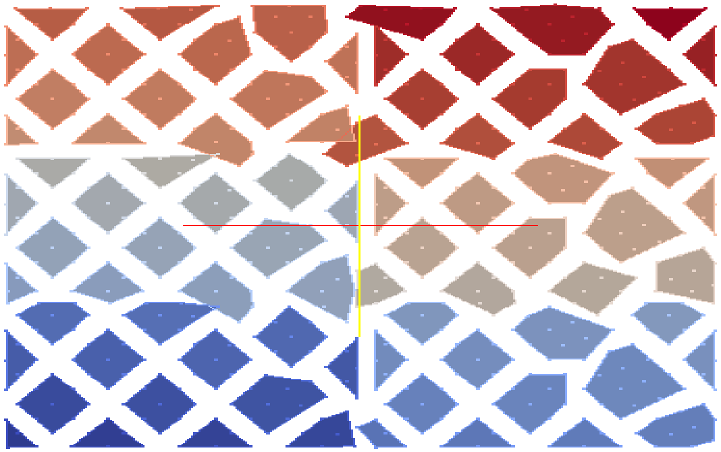
Two choices: coupled and uncoupled aggregation

- Uncoupled aggregation aggregates only inside a subdomain
- Coupled aggregation allows aggregates to cross subdomain boundary
- Coupled aggregation is more expensive, but has convergence similar to the serial case

Uncoupled



Coupled



Constraints in parallel

Let P have the following pattern and nullspace consist of two vectors

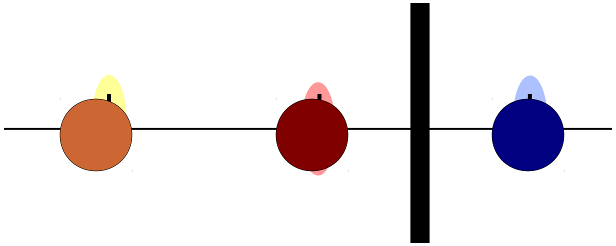
$$P \begin{bmatrix} b^c_{11} & b^c_{12} \\ b^c_{21} & b^c_{22} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \end{bmatrix} \qquad P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & 0 \\ p_{31} & p_{32} \\ 0 & p_{41} \end{bmatrix}$$

$$\begin{bmatrix} b^c_{11} & 0 & 0 & b^c_{21} & 0 & 0 \\ 0 & b^c_{11} & 0 & 0 & 0 & 0 \\ 0 & 0 & b^c_{11} & 0 & b^c_{21} & 0 \\ 0 & 0 & 0 & 0 & 0 & b^c_{21} \\ b^c_{12} & 0 & 0 & b^c_{22} & 0 & 0 \\ 0 & b^c_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & b^c_{12} & 0 & b^c_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & b^c_{22} \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{21} \\ p_{31} \\ p_{12} \\ p_{32} \\ p_{42} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{41} \\ b_{12} \\ b_{22} \\ b_{32} \\ b_{42} \end{bmatrix}$$

$$\begin{bmatrix} b^c_{11} & b^c_{21} & 0 & 0 & 0 & 0 \\ b^c_{12} & b^c_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & b^c_{11} & 0 & 0 & 0 \\ 0 & 0 & b^c_{12} & 0 & 0 & 0 \\ 0 & 0 & 0 & b^c_{11} & b^c_{21} & 0 \\ 0 & 0 & 0 & b^c_{12} & b^c_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & b^c_{21} \\ 0 & 0 & 0 & 0 & 0 & b^c_{22} \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{21} \\ p_{32} \\ p_{41} \\ p_{42} \end{bmatrix} = \begin{bmatrix} b_{11} \\ b_{12} \\ b_{21} \\ b_{22} \\ b_{31} \\ b_{32} \\ b_{41} \\ b_{42} \end{bmatrix}$$

Constraints in parallel

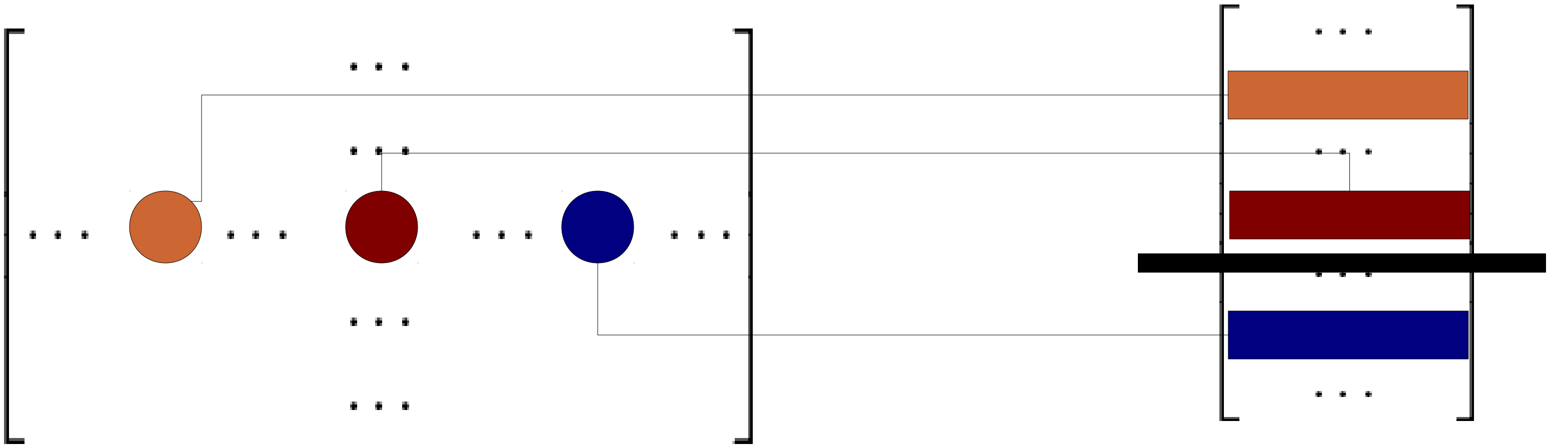
What does each block correspond to?



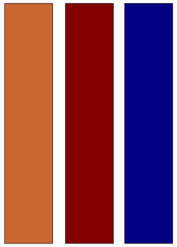
Consider a row of P with three nonzeros

Prolongator row

Coarse nullspace



Block of the constraint corresponding to the row



Energy minimization algorithm (updated)

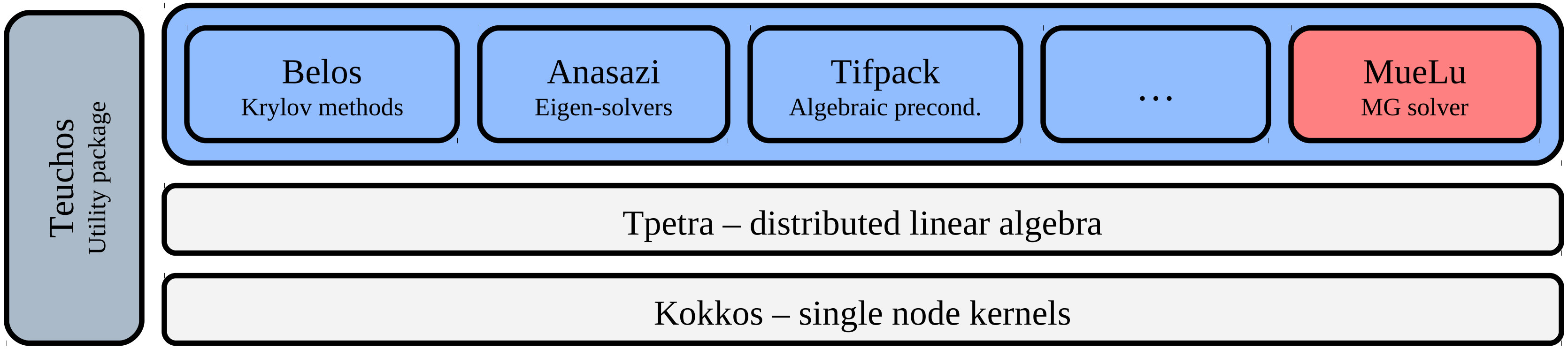
```
Construct aggregates
$\mathcal{N} = |A||P^{(0)}|$                                   ▷ Select sparsity pattern
Import ghost components of nullspace vectors

$D = diag(A)$                                                 ▷ Diagonal preconditioner
$R = -AP^{(0)}$                                               ▷ Initial residual
$R = \mathbf{enforce}(R, \mathcal{N})$                        ▷ Enforce sparsity on $R$
$R = \mathbf{project}(R, X)$                                  ▷ Enforce $RB_c = \mathbf{0}$

for $i$ to iter do
    $Z = D^{-1}R$
    $\gamma =< R, Z >_F$
    if $i$ is 1 then
        $Y = Z$
    else
        $\beta = \gamma / \gamma_{old}$;
        $Y = Z + \beta Y$                                     ▷ New search direction
    end if
    $\gamma_{old} = \gamma$
    $Y_A = AY$
    $Y_A = \mathbf{enforce}(Y_A, \mathcal{N})$                ▷ Enforce sparsity on $Y_A$
    $Y_A = \mathbf{project}(Y_A, B_c)$                        ▷ Enforce $Y_A B_c = \mathbf{0}$
    $\alpha = \gamma / < Y, Y_A >_F$
    $P^{(i)} = P^{(i-1)} + \alpha Y$                          ▷ Update prolongator
    $R = R - \alpha Y_A$                                      ▷ Update residual
```

MueLu

- Future package of the Trilinos project (to replace ML)
  - Massively parallel
  - Multicore and GPU aware
  - Templated types for mixed precision calculation (32-bit – 64-bit) and type complex

- Objective is to solve problem with billions of DOF on 100Ks of cores…

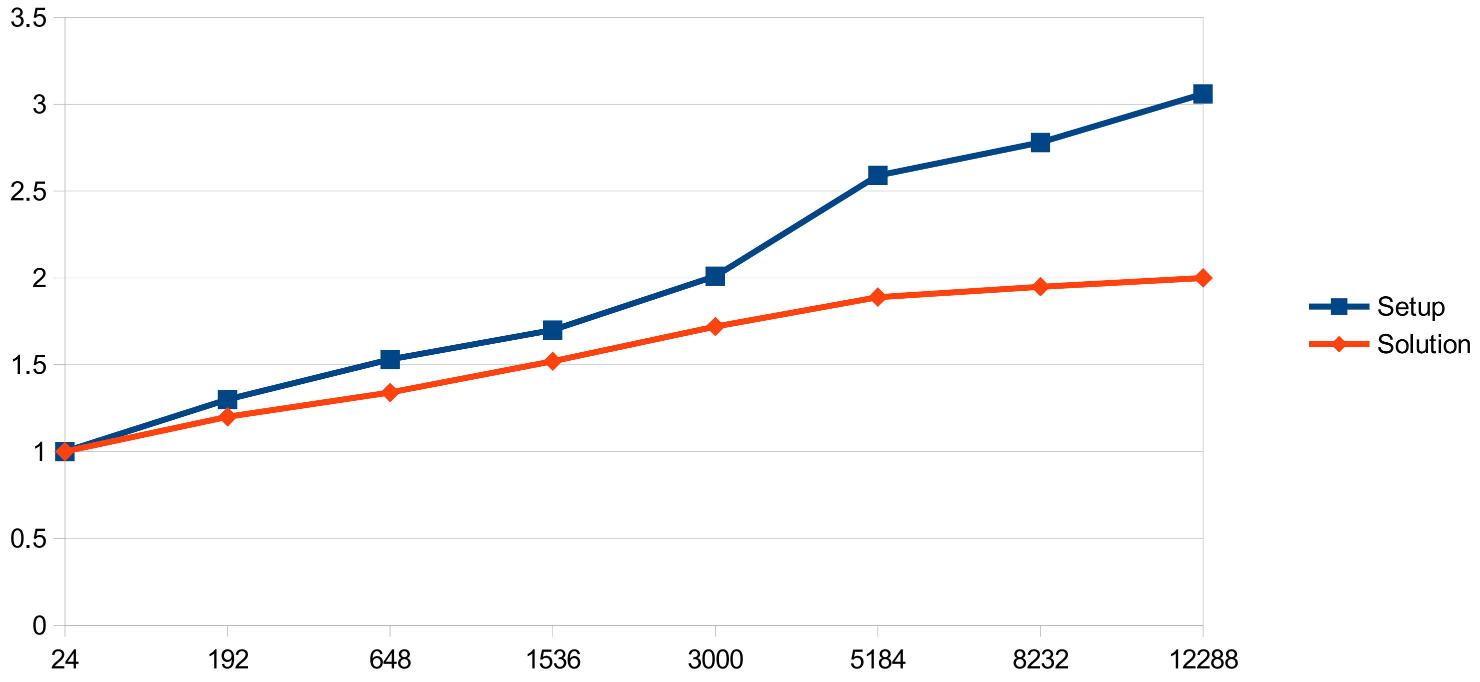
- Leverage the Trilinos software stack:



- Currently in development…

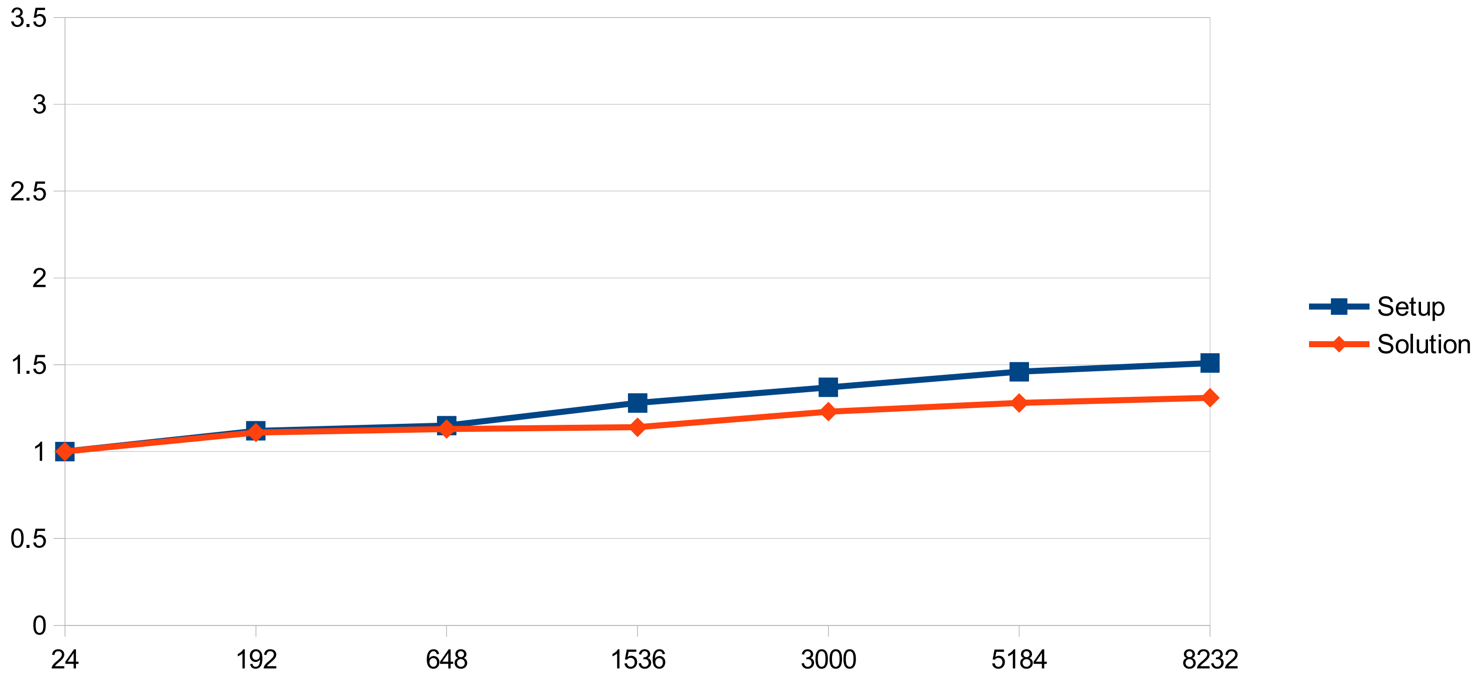
Numerical results – Laplace 3D

- Laplace 3D, 7 point stencil
- Energy minimization
  - 2 CG iterations
  - Initial guess: tentative prolongator
  - Sparsity pattern: same as SA



Numerical results – Elasticity 3D

- Elasticity 3D, Poisson ratio 0.25
- Energy minimization
  - 2 CG iterations
  - Initial guess: tentative prolongator
  - Sparsity pattern: same as SA



Setup amortization

Setup amortization: reuse

- Emin setup may be expensive (several times that of SA)
- Typically, we need multigrid for each linear iteration of Newton, therefore it is reasonable to assume that the system does not change too much
- Many components of the setup phase can be reused
  - Initial prolongator
  - Sparsity patterns (assuming no filtering)
  - Matrix graphs (assuming fixed mesh)

Strategies for a sequence of systems

- **No reuse**: construct multigrid anew for each iteration
- **Simple reuse**: construct multigrid only for the first iteration, and then use the same preconditioner for all iteration
- **Fast reuse**: construct multigrid with multiple iterations on the first step, and use fewer iterations for consecutive step, reusing constructed prolongators and graphs

Numerical example: icesheet model

System of two coupled non-linear PDEs

$$\begin{cases} -\nabla \cdot (2\mu\dot{\boldsymbol{\epsilon}}_1) & = & -\rho g \frac{\partial s}{\partial x}, \\ -\nabla \cdot (2\mu\dot{\boldsymbol{\epsilon}}_2) & = & -\rho g \frac{\partial s}{\partial y} \end{cases}$$

with Glen's law viscosity

$$\mu = \frac{1}{2} A^{-\frac{1}{n}} \left(\dot{\epsilon}_{xx}^2 + \dot{\epsilon}_{yy}^2 + \dot{\epsilon}_{xx}\dot{\epsilon}_{yy} + \dot{\epsilon}_{xy}^2 + \dot{\epsilon}_{xz}^2 + \dot{\epsilon}_{yz}^2 + \gamma \right)^{\left(\frac{1}{2n} - \frac{1}{2}\right)}$$

$$\dot{\boldsymbol{\epsilon}}_1^T = \left(\; 2\dot{\epsilon}_{xx} + \dot{\epsilon}_{yy}, \quad \dot{\epsilon}_{xy}, \quad \dot{\epsilon}_{xz} \; \right)$$

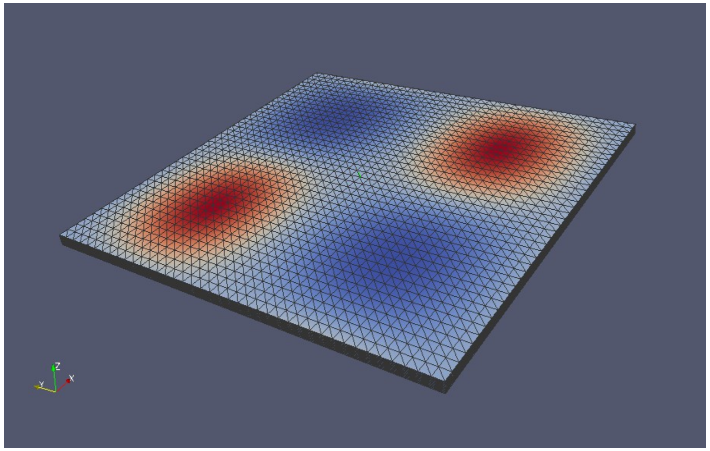
$$\dot{\boldsymbol{\epsilon}}_2^T = \left(\; \dot{\epsilon}_{xy}, \quad \dot{\epsilon}_{xx} + 2\dot{\epsilon}_{yy}, \quad \dot{\epsilon}_{yz} \; \right)$$

$$\dot{\epsilon}_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$$

$$\begin{array}{lcl} A & = & \text{flow rate factor} \\ n & = & \text{Glen's law exponent} = 3 \\ \gamma & = & \text{regularization parameter} \\ \beta & = & \text{sliding coefficient} \geq 0 \end{array}$$

Discretization: classical Galerkin FEM with structured or unstructured mesh.

Nonlinear solver: Newton's method



Numerical results: icesheet model

Step	Emin(6)	Emin(1)	Emin(6,1)
2	17	30	17
8	16	32	17
12	17	33	18
18	17	36	18
23	17	36	18
28	17	34	18

Summary

- Energy minimization AMG is flexible

- Energy minimization AMG is suitable for parallelization
  - Standard parallel operations (MxM, BLAS1) are well known
  - Constraint application could be done locally storing ghost info

- Preliminary results show promise

- There are ways to reduce setup cost