

# AMGe: Element Based Algebraic Coarse Spaces with Applications

Panayot S. Vassilevski

Center for Applied Scientific Computing (CASC)

Lawrence Livermore National Laboratory (LLNL)  
Livermore, CA



23<sup>th</sup> Conference on Domain Decomposition Methods, Jeju, Korea

Work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under

Contract DE-AC52-07NA27344



# Collaborators

Andrew Barker (LLNL)

Yalchin Efendiev (Texas A & M)

Pasqua D'Ambra (CNR, Naples)

Max La Cour Christensen (Technical University, Denmark)

Xiaozhe Hu (Tufts)

Delyan Kalchev (CU Boulder)

Christian Ketelsen (CU Boulder)

Chak Shing Lee (Texas A & M University)

Ilya Lashuk (Tufts)

Umberto Villa (ICES - UT Austin)

Jinchao Xu (Penn State)

Ludmil Zikatanov (Penn State)

- 1 MG and AMG: background and recent developments

# Overview

- 1 MG and AMG: background and recent developments
- 2 From TG convergence to coarse spaces approximation

# Overview

- 1 MG and AMG: background and recent developments
- 2 From TG convergence to coarse spaces approximation
- 3 AMGe and numerical upscaling

# Overview

- 1 MG and AMG: background and recent developments
- 2 From TG convergence to coarse spaces approximation
- 3 AMGe and numerical upscaling
- 4 de Rham complexes on agglomerated elements

# Overview

- 1 MG and AMG: background and recent developments
- 2 From TG convergence to coarse spaces approximation
- 3 AMGe and numerical upscaling
- 4 de Rham complexes on agglomerated elements
- 5 Other AMGe applications

# Overview

- 1 MG and AMG: background and recent developments
- 2 From TG convergence to coarse spaces approximation
- 3 AMGe and numerical upscaling
- 4 de Rham complexes on agglomerated elements
- 5 Other AMGe applications
- 6 Conclusions

## The MG

- **Started** with R. P. Fedorenko (early 60s);
- **Made real impact** due to Achi Brandt, W. Hackbusch (late 70s); Randy Bank, Steve McCormick, Stüben and Trottenberg, Harry Yserentant (1st and 2nd European MG conferences, early 80s), and many others after that.
- **The BPX and regularity-free theory** by Jinchao Xu, and Bramble and Pasciak, and Junping Wang (mid-to-late 80s and early 90s), and Griebel and Oswald (1995).
- **The TL HB** by Bank and Dupont 1980, Axelsson and Gustafsson (1983); the **ML HB** method by Yserentant (**additive**) and Bank, Dupont and Yserentant (**multiplicative**)- (mid-to-late 80s);
- **The algebraic stabilization of HB**: the **AMLI** method by Axelsson and PSV (late 80s-early 90s); and the **wavelet-like HB** stabilization by PSV and Junping Wang (mid-to-late 90s).

## The MG (cont.)

- The **XZ-identity** (Ludmil Zikatanov and Jinchao Xu (2002)).
- The **AMLI-MG** and its nonlinear version (PSV (2008), with analysis in Y. Notay and PSV (2008), and, in Xiaozhe Hu, PSV, and Jinchao Xu (2013)).
- Algebraic convergence analysis showing that the **TG convergence improves with increasing the smoothing steps** (Xiaozhe Hu, PSV and Jinchao Xu, 2015).

## The AMG

- The *classical* **AMG**: originally proposed by Achi Brandt, Steve McCormick and John Ruge (early 80s), and the most popular paper by J. Ruge and K. Stüben (87).
- The **SA-AMG**: P. Vaněk (1992), and P. Vaněk, M. Brezina and J. Mandel (90s).
- To address **AMG scalability**, there was a large effort (started in late 90s) at LLNL in collaboration with CU Boulder; in particular the **scalable solvers library hypre** was designed and developed; also **new** AMG methods were proposed: **AMGe** (2001), the **spectral AMGe** (2003 and 2007), the **adaptive SA-AMG** (2004), **adaptive AMG** (2006), and **adaptive AMGe** (2008).

## The AMG (cont.)

- [ML convergence analysis](#) of SA-AMG: P. Vaněk, M. Brezina and J. Mandel (2001) and in M. Brezina, P. Vaněk, and PSV (2012).
- The [TL spectral SA-AMG](#): proposed in a CU Denver report: M. Brezina, C. Heberton, J. Mandel, and P. Vaněk (99), and its [spectral SA-AMGe](#) version in M. Brezina and PSV (2011).

# MG and AMG: background and recent developments

- The **auxiliary space** preconditioning:
  - The **fictitious space lemma**: Sergei Nepomnyaschikh (80s and 90s),
  - It is related to the **distributive relaxation** of Achi Brandt (70s) and **transformative smoothers** by G. Wittum (mid-to-late 80s)
  - The **general setting** is due to Jinchao Xu (1996).

A main application: the **HX-decomposition** by Ralf Hiptmair and Jinchao Xu (2006) which led to the **scalable software** by Tzanio Kolev and PSV: **AMS-  $H(\text{curl})$**  (2009) and **ADS:  $H(\text{div})$**  (2012), available in *MFEM* and *hypre* ).

- **Additive representation of (A)MG**: PSV (2008) and its impact on **parallel AMG coarsening**: PSV and U. Yang (2014).  
(Additive convergence analysis of V-cycle MG can be found earlier in S. Brenner (2002).)

## The AMG (cont.)

In recent years: [Explosion of AMG implementations](#): with various applications, especially in oil reservoir simulations.

# AMG: a general philosophy for designing fast algorithms

In **summary**:

“(A)MG can be viewed as a **recursive divide-and-conquer** methodology for designing fast algorithms that have the potential for optimal order complexity.

The AMG algorithm (to be designed) aims to partition the solution space in two complementary components:

- (i) The 1st component can be handled by local (order  $\mathcal{O}(n)$ ) operations;
- (ii) The second component, giving rise to a problem with reduced dimension, should maintain the main properties of the original problem so that recursion can be applied.

The decomposition is done **implicitly** by the algorithm we design. ”

The above items (i)-(ii) are a version of Achi Brandt’s definition (2000) of “**compatible relaxation**” coarsening.

I.e., **if we knew** the solution at the coarse level, we should be able to recover the remaining part of the solution **fast in  $\mathcal{O}(n)$**  operations.

# The two-grid method: tools, TG operator and some basic theory

MG = AMG as algorithms.

They differ in **terms of the setup**: in MG the tools are given, whereas in AMG, the method builds the missing **tools**.

- $A$  - the given  $n \times n$  s.p.d. matrix.
- $M$  - the smoother (weighted Jacobi, Gauss-Seidel, incomplete factorization matrices, etc.). In theory, we need  $\|I - M^{-1}A\|_A < 1$  or equivalently  $M + M^T - A$  be s.p.d.
- $P : \mathbb{R}^{n_c} \mapsto \mathbb{R}^n$ ,  $n_c < n$  - the interpolation matrix;  $P^T$  is the “restriction” matrix.
- $A_c = P^T A P$  - the coarse  $n_c \times n_c$  matrix.
- Set  $A := A_c$  and repeat.

# The two-grid (TG) algorithm for $A\mathbf{x} = \mathbf{b}$

Given a current iterate  $\mathbf{x}$  (initially  $\mathbf{x} = 0$ ), perform:

- “Pre-smoothing”: solve  $M\mathbf{y} = \mathbf{b} - A\mathbf{x}$  and compute the intermediate iterate

$$\mathbf{x} := \mathbf{x} + \mathbf{y} = \mathbf{x} + M^{-1}(\mathbf{b} - A\mathbf{x}).$$

- Restrict the residual, i.e., compute

$$\mathbf{r}_c = P^T(\mathbf{b} - A\mathbf{x}).$$

- Solve for a coarse-grid correction,

$$A_c\mathbf{x}_c = \mathbf{r}_c.$$

- Interpolate and compute next intermediate iterate  $\mathbf{x} := \mathbf{x} + P\mathbf{x}_c$ .
- “Post-smoothing”: solve  $M^T\mathbf{z} = \mathbf{b} - A\mathbf{x}$ , and compute the next two-grid iterate,

$$\mathbf{x}_{TG} = \mathbf{x} + \mathbf{z} = \mathbf{x} + M^{-T}(\mathbf{b} - A\mathbf{x}).$$

# The TG operator

The TG algorithm with zero initial iterate provides a mapping  $B_{TG}$

$$\text{input } \mathbf{b} \mapsto \text{output } B_{TG}^{-1}\mathbf{b} = \mathbf{x}_{TG}.$$

# Expressions for the TG and MG operators

We can define  $B_{TG}^{-1}$  using the TG iteration matrix,

$$\begin{aligned} E_{TG} &= I - B_{TG}^{-1}A \\ &= (I - M^{-T}A)(I - PA_c^{-1}P^T A)(I - M^{-1}A). \end{aligned}$$

Solving for  $B_{TG}^{-1}$ , letting  $\bar{M} = M(M + M^T - A)^{-1}M^T$ , gives

$$B_{TG}^{-1} = \bar{M}^{-1} + (I - M^{-T}A)PA_c^{-1}P^T(I - AM^{-1}).$$

In the multilevel case, we have

$$B_k^{-1} = \bar{M}_k^{-1} + (I - M_k^{-T}A_k)P_k B_{k+1}^{-1} P_k^T (I - A_k M_k^{-1}).$$

# Additive form of the MG operator

Introducing the smoothed interpolant (as in SA AMG)

$$\bar{P}_k = (I - M_k^{-T} A_k) P_k,$$

we end up with the additive representation

$$B_k^{-1} = \bar{M}_k^{-1} + \bar{P}_k B_{k+1}^{-1} \bar{P}_k^T.$$

Using recursion, we have the additive form of the MG operator

$$B_{V\text{-cycle}} = B_0,$$

$$B_{V\text{-cycle}}^{-1} = \sum_k \bar{P}_0 \dots \bar{P}_{k-1} \bar{M}_k^{-1} (\bar{P}_0 \dots \bar{P}_{k-1})^T.$$

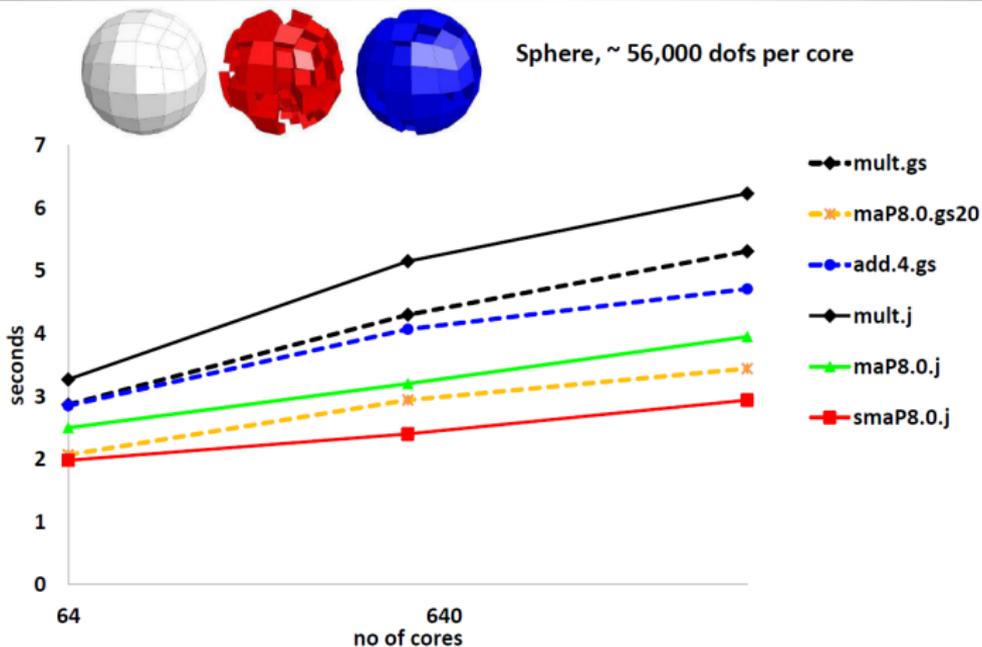
The additive MG, BPX, has the same form

$$B_{BPX}^{-1} = \sum_k P_0 \dots P_{k-1} \bar{M}_k^{-1} (P_0 \dots P_{k-1})^T.$$

The difference is in the interpolation matrices; in MG we use the smoothed ones,  $\bar{P}_k$ , whereas in BPX, we use the original ones,  $P_k$ .

# Performance on finite element test problems

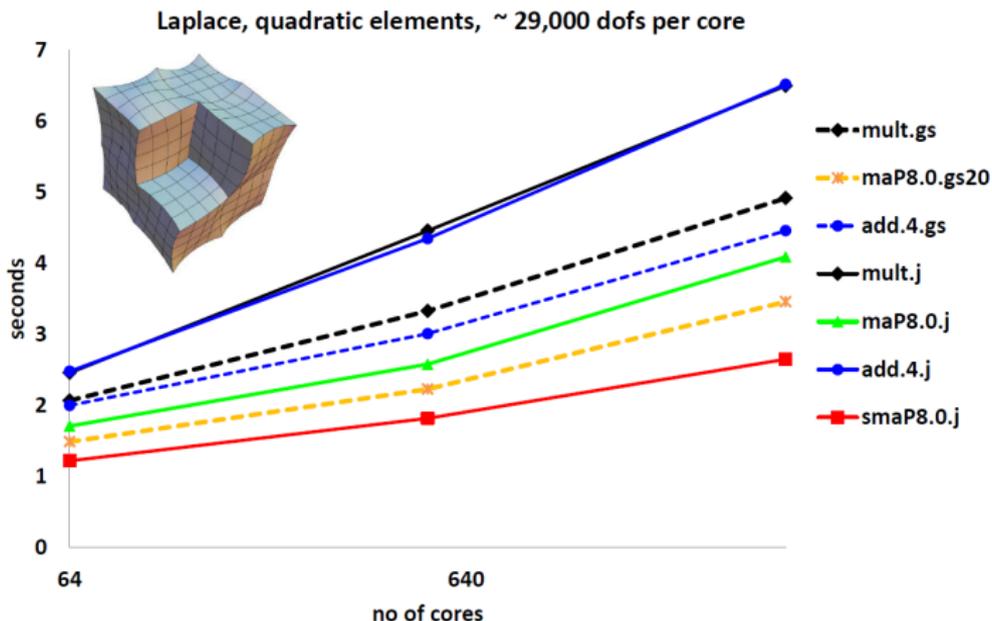
## Solve times on Hera, AMG-PCG, 3D problem with jumps,



Naming convention: Variant.Level.Smoother.

# Performance on finite element test problems

## Solve times on Hera, LLNL Linux cluster, AMG-PCG,



Naming convention: [Variant.Level.Smoother](#).

# Characterization of $K_{TG}$

We are interested in the spectral equivalence relations

$$\mathbf{v}^T A \mathbf{v} \leq \mathbf{v}^T B_{TG} \mathbf{v} \leq K_{TG} \mathbf{v}^T A \mathbf{v}.$$

The method is optimal if  $K_{TG}$  is a mesh-independent constant.

For any  $A$ -convergent smoother  $M$ , the TG method is  $A$ -convergent. I.e., we have that  $B_{TG}^{-1}$  is s.p.d. and

$$\mathbf{v}^T A \mathbf{v} \leq \mathbf{v}^T B_{TG} \mathbf{v}.$$

The following main [characterization](#) holds:

## Theorem (Falgout, PSV and Zikatanov (2005))

$$K_{TG} = \max_{\mathbf{v}} \frac{\min_{\mathbf{v}_c} \|\mathbf{v} - P\mathbf{v}_c\|_{\tilde{M}}^2}{\mathbf{v}^T A \mathbf{v}}.$$

$\tilde{M} = M^T (M + M^T - A)^{-1} M$  is the symmetrized smoother

(such as symmetric Gauss–Seidel).

The above result can also be derived from the TLXZ-identity.

# The weak approximation property

If  $\tilde{M} \simeq D_A$ , the diagonal of  $A$  (typical case):

- convergent TG implies the “weak” *approximation property*:

$$\|\mathbf{v} - P\mathbf{v}_c\|_{D_A}^2 \simeq \|\mathbf{v} - P\mathbf{v}_c\|_{\tilde{M}}^2 \leq K_{TG} \mathbf{v}^T A \mathbf{v}.$$

- In the simplest case,  $D_A \simeq \|A\| I$ , it takes the form

$$\|A\|^{\frac{1}{2}} \|\mathbf{v} - P\mathbf{v}_c\| \leq \eta_w \|\mathbf{v}\|_A.$$

# The finite element case: necessary condition for TG convergence

- Let  $A$  come from a bilinear form  $a(., .)$  and fine-grid f.e. space  $S_h$ ,
- Use a coarse space  $S_H$  on a mesh  $H \simeq h$ .
- Then, the matrix-vector “**weak approximation property**” translates to

$$\inf_{v_H \in S_H} \|v_h - v_H\|_{0, \rho} \leq CH \sqrt{a(v_h, v_h)}.$$

The left-hand side is a  $\rho$ -weighted  $L_2$ -norm (the weight  $\rho$  comes from the diagonal of  $A$ ).

It is a **necessary condition** for uniform (in  $h \mapsto 0$ ) TG convergence.

That is, the **coarse space**  $S_H$  cannot be arbitrary.

# AMG is an inverse problem

In AMG, we need to generate the coarse hierarchy

- interpolation matrices  $P$ ;
- coarse-grid matrices  $A_c = P^T A P$ .

This is an “ill-posed” inverse problem: i.e., there are many coarse spaces that can do similar (good or bad) job.

# The necessary condition gives guidelines for AMG

The **necessary condition** for TG when  $M \simeq \|A\|I$ , reads

$$\|A\|^{\frac{1}{2}} \|\mathbf{v} - P\mathbf{v}_c\| \leq \eta_w \|\mathbf{v}\|_A.$$

It gives the **major guidelines**:

- If  $\mathbf{v}$  is such that  $A\mathbf{v} \approx 0$ , then  $P\mathbf{v}_c \approx \mathbf{v}$ .

That is, **the coarse space** should essentially **contain** (or approximate very well) **vectors** in the **near nullspace of  $A$** .

- Another corollary is that the coarse interpolant should be bounded in energy:

$$\|P\mathbf{v}_c\|_A \leq \|\mathbf{v}\|_A + \|A\|^{\frac{1}{2}} \|\mathbf{v} - P\mathbf{v}_c\| \leq (1 + \eta_w) \|\mathbf{v}\|_A.$$

That is, we want to have an **interpolation mapping  $P$**  that exhibits some **“energy”** minimization property.

# Element agglomeration algebraic MG (AMGe)

If we use fine-grid **finite element** information, a subclass of AMG is the so-called

**element based AMG**, or **AMGe**, proposed in 2000 (by a team from CU Boulder and CASC, LLNL).

If we generate coarse counterparts of elements and element matrices by recursively agglomerating fine-grid elements, we end up with the

**element agglomeration AMGe** proposed in 2001 (J. Jones and P.S.V.).

# AMGe: an element agglomeration algebraic MG

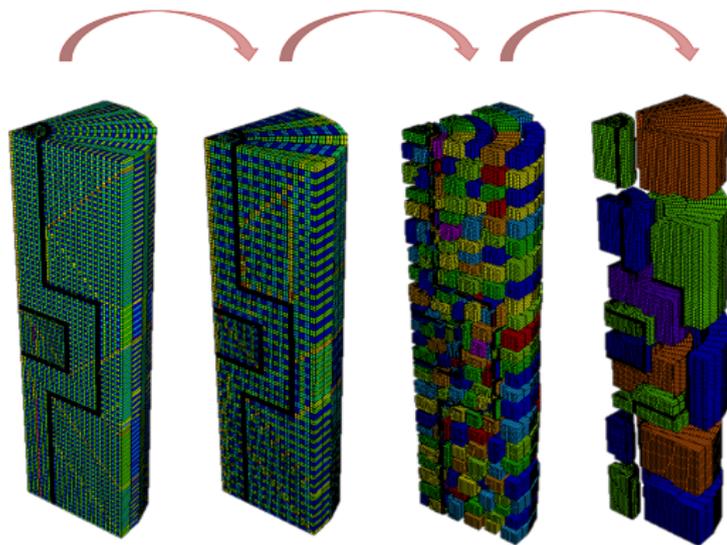
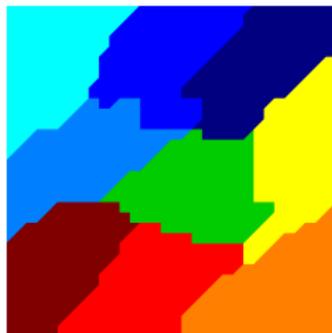


Figure: *Illustration of unstructured agglomerates in 3D*

# The spectral AMGe: a scale of coarse spaces with approximation that improves

In Chartier et al. (2003, and in DD16 Proceedings, 2007), the following spectral choice of coarse dofs was proposed:

$$-\nabla \cdot [k(x) \nabla p] = f \Rightarrow \text{Finite Elements} \Rightarrow \mathbf{A}\mathbf{u} = \mathbf{f}$$



Given an unstructured FE mesh. Would like a **coarse space** that we can use for accurate coarse discretization and multilevel solvers.

Subdivide the mesh into nonoverlapping groups of **agglomerated elements**  $\{T\}$ .

Coarse basis functions come from low-energy eigenmodes of subproblems on agglomerates:

$$\{\mathbf{q}_k\} \text{ s.t. } A_T \mathbf{q}_k = \lambda_k D_T \mathbf{q}_k \text{ for all } \lambda_k \leq \theta \|D_T^{-\frac{1}{2}} A_T D_T^{-\frac{1}{2}}\|$$

We select the 1st  $m_T \geq 1$  **near-null space** components  $\mathbf{q}_k$ .

# Spectral Agglomerated Element AMGe and SA

Form local interpolant  $\bar{P}_i$  for  $i$ th aggregate (subset of  $T$ ) using  $\mathbf{q}_k|_T$ .



Then form (global) tentative **interpolation operator**.

$$\bar{P} = \begin{bmatrix} \bar{P}_1 & 0 & & 0 \\ 0 & \bar{P}_2 & & 0 \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \dots & \bar{P}_{n_c} \end{bmatrix}$$

Smooth coarse basis for better **energy stability**:

$$P = (I - p_\nu (D^{-1}A)) \bar{P} \text{ for some polynomial } p_\nu.$$

- This version is from Marian Brezina and P.S.V. (2012). An earlier version: M. Brezina, C. Heberton, J. Mandel, and P. Vaněk (1999).
- The convergence of the TG SA- $\rho$ AMGe improves with increasing the polynomial degree (Xiaozhe Hu, PSV and Jinchao Xu (2015)).

# SA- $\rho$ AMGe: upscaling error estimates

By construction, we have the error estimate (Brezina and P.S.V. 2012):

$$H^{-1} \|\mathbf{v} - PQ\mathbf{v}\|_G \leq \beta_1 \left( 1 + \max_T \left( \frac{h^2}{H^2 \lambda_{m_T+1}} \right)^{\frac{1}{2}} \right) \|\mathbf{v}\|_A,$$

and energy stability

$$\|PQ\mathbf{v}\|_A \leq \left( 1 + \beta_2 \max_T \left( \frac{h^2}{H^2 \lambda_{m_T+1}} \right)^{\frac{1}{2}} \right) \|\mathbf{v}\|_A.$$

The norm  $\|\cdot\|_G$  corresponds to the weighted  $L_2$ -norm:  $(\int k(\mathbf{x})v^2(\mathbf{x}) d\mathbf{x})^{\frac{1}{2}}$ .

- The constants  $\beta_s$  are independent of the coefficient  $k = k(\mathbf{x})$ .
- Also,  $\lambda_{m_T+1}$  behaves like  $(\frac{h}{H})^2$ .

# Coarse spaces for numerical upscaling

- In many applications, repeated simulations are needed, which can easily become **computationally infeasible** unless **dimension (model) reduction** is applied.
- It can be achieved by **accurate coarse** models referred to as **numerical upscaling**.
- Our approach is based on discretizations using the **AMGe coarse spaces** with guaranteed approximation properties.

# Achieving practical numerical upscaling

To achieve practical numerical upscaling we need:

- to reduce the problem size while maintaining reasonable accuracy;
- to reduce the memory to store the coarse (upscaled) problem.

The first item is typically the main goal that many researchers try to accomplish. It is related to the *arithmetic complexity* (AC)

$$AC \equiv \frac{\# \text{dofs}_{\text{fine}} + \# \text{dofs}_{\text{coarse}}}{\# \text{dofs}_{\text{fine}}} < 2 \text{ or even better } AC \approx 1.$$

# Achieving practical numerical upscaling

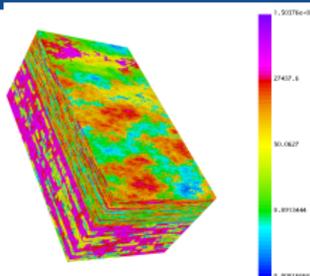
However, the second item is even more crucial for achieving upscaling:

If the memory to store the coarse problem is larger than the one of the fine-grid problem, even if we have substantially reduced the number of degrees of freedom, this is NOT a meaningful “reduced dimension model”.

In AMG terminology, we need to ensure that the *operator complexity* (OC):

$$OC \equiv \frac{\text{nnz}_{\text{fine}} + \text{nnz}_{\text{coarse}}}{\text{nnz}_{\text{fine}}} < 2 \text{ or even better } OC \approx 1.$$

# SA- $\rho$ AMGe: examples for upscaling in 2D



- a 3D domain with dimensions  $1200 \times 2200 \times 170$  units, divided into cells of size  $20 \times 10 \times 2$  units.
- The fine-scale model in 3D has  $60 \times 220 \times 85$  cells.
- The 3D domain is cut into 85 horizontal slices and we solve a 2D problem for each slice.
- Each 2D domain has dimension  $1200 \times 2200$  units, divided into cells of size  $20 \times 10$ .
- Each 2D mesh has  $60 \times 220$  elements (13200 fine-grid elements).
- The coefficient  $k(\mathbf{x})$  is a piecewise constant scalar function.

# SA- $\rho$ AMGe: coarse space $V_H$ properties

The characteristics of the constructed coarse space  $V_H$  are:

- 1 the number of coarse degrees of freedom (**coarse space dimension**);
- 2 the **sparsity** pattern of the coarse stiffness matrix  $A_c = P^T A P$

$$O.C. = \frac{nnz(A) + nnz(A_c)}{nnz(A)};$$

- 3 the **energy error** reduction:

$$\sqrt{\frac{a(u_h - u_H, u_h - u_H)}{a(u_h, u_h)}} = \frac{\|\mathbf{u} - P A_c^{-1} P^T A \mathbf{u}\|_A}{\|\mathbf{u}\|_A},$$

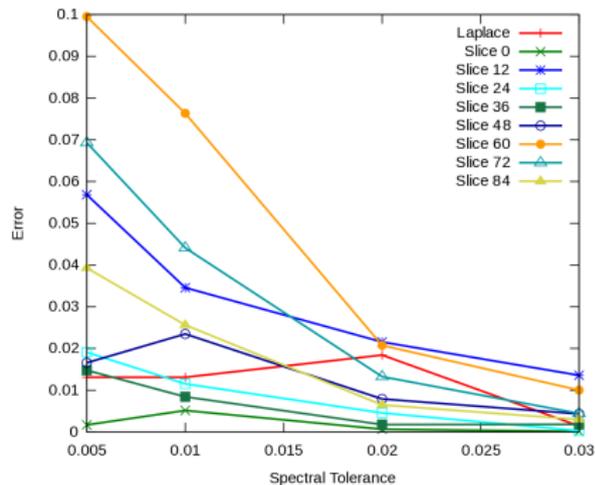
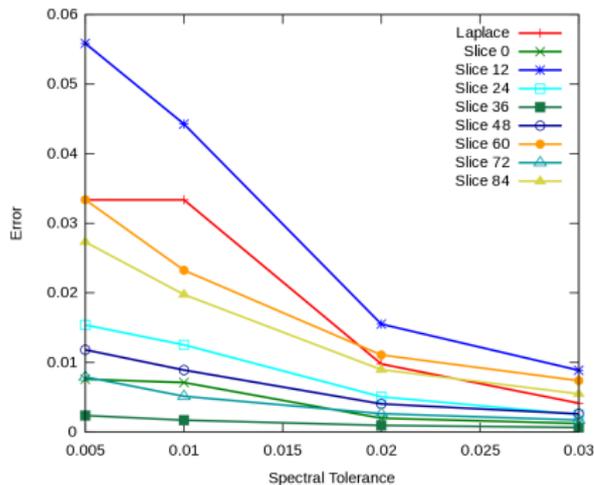
where  $\mathbf{u}$  solves  $A \mathbf{u} = \mathbf{f}$ .

- 4 the **weighted- $L_2$  error** reduction: Let  $G$  be the  $k$ -weighted mass matrix.

$$\frac{\|u_h - u_H\|_{0,k}}{\|u_h\|_{0,k}} = \frac{\|\mathbf{u} - P A_c^{-1} P^T A \mathbf{u}\|_G}{\|\mathbf{u}\|_G},$$

where  $\mathbf{u}$  solves  $A \mathbf{u} = \mathbf{f}$ .

# SA- $\rho$ AMGe: Approximation property versus spectral tolerance

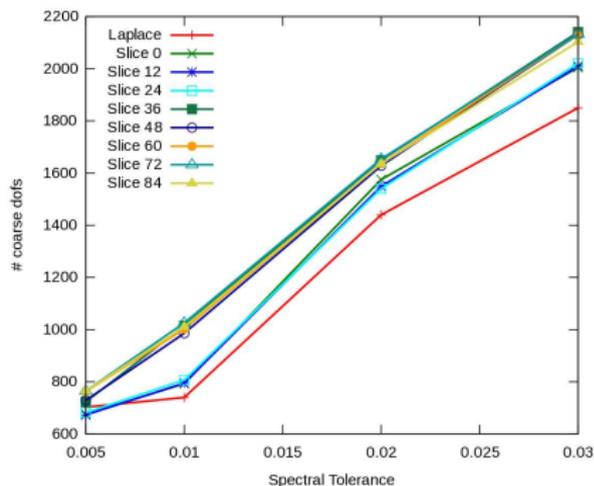
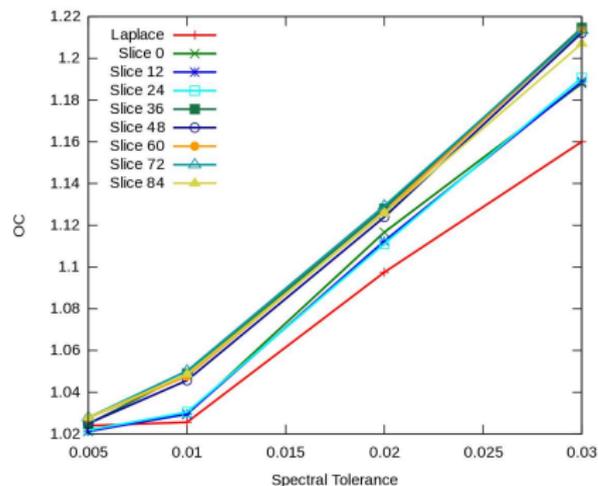


left:  $\|\mathbf{u} - P\mathbf{u}_c\|_{A(k)} / \|\mathbf{u}\|_{A(k)}$

right:  $\|\mathbf{u} - P\mathbf{u}_c\|_{G(k)} / \|\mathbf{u}\|_{G(k)}$

1 mesh refinement and varying  $\theta$  (53361 fine dofs,  $nnz(A) = 476881$ , 370 agglomerates).

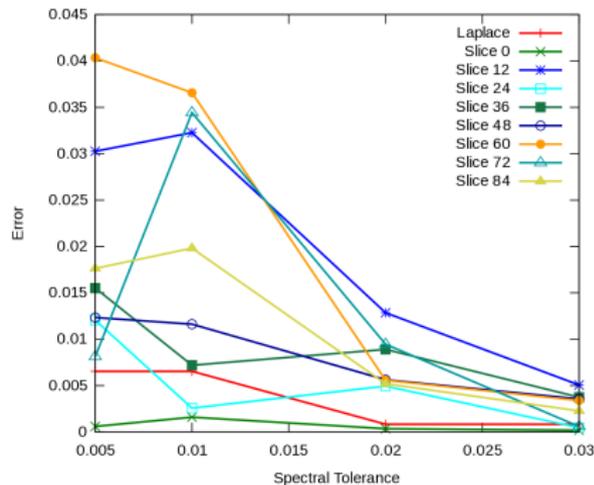
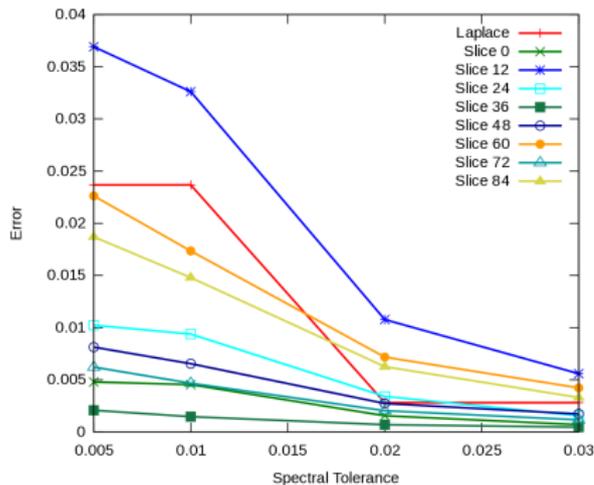
# SA- $\rho$ AMGe: Operator complexity and # coarse dofs versus spectral tolerance



left: Operator complexity  
right: Number of coarse dofs

1 mesh refinement and varying  $\theta$  (53361 fine dofs,  $nnz(A) = 476881$ , 370 agglomerates).

# SA- $\rho$ AMGe: Approximation property versus spectral tolerance

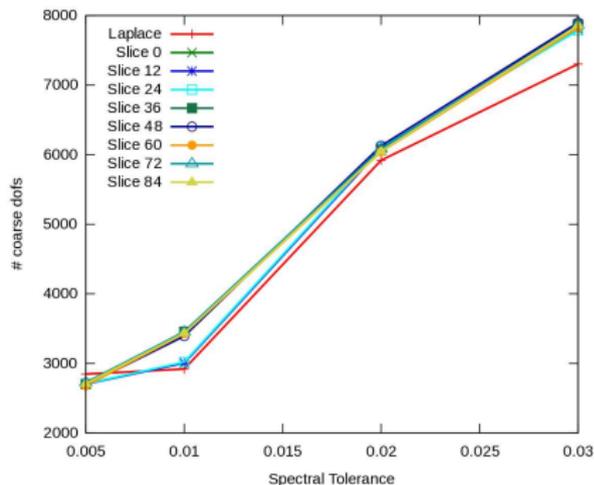
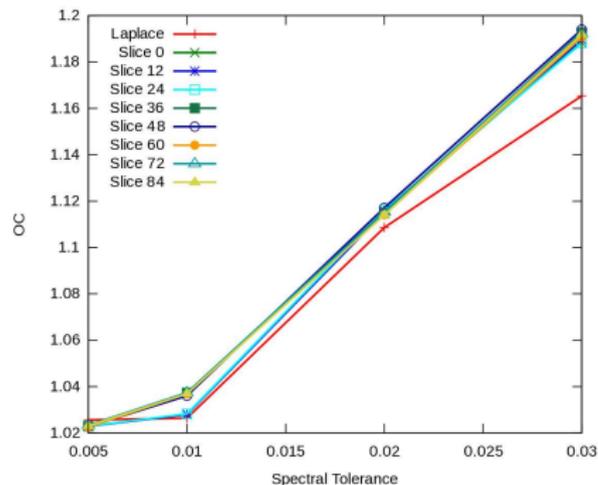


left:  $\|\mathbf{u} - P\mathbf{u}_c\|_{A(k)} / \|\mathbf{u}\|_{A(k)}$

right:  $\|\mathbf{u} - P\mathbf{u}_c\|_{G(k)} / \|\mathbf{u}\|_{G(k)}$

2 steps of refinement and varying  $\theta$  (212321 fine dofs,  $nnz(A) = 1904161$ , 1460 agglomerates)

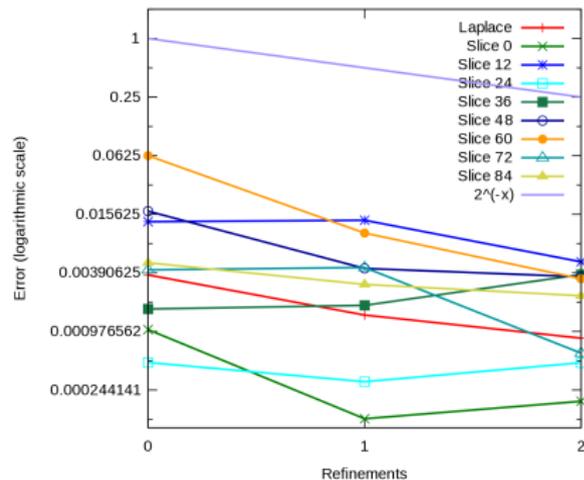
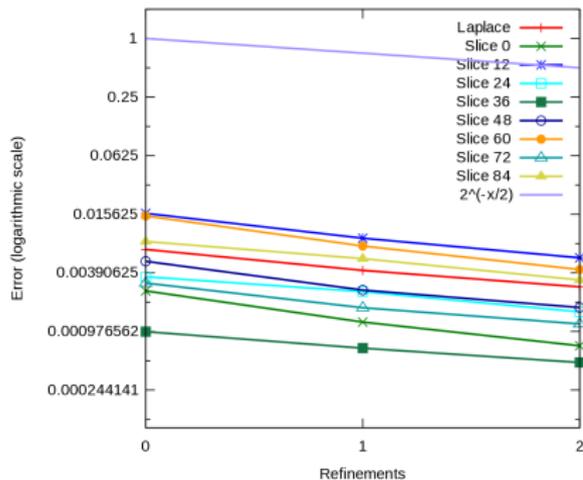
# SA- $\rho$ AMGe: Operator complexity and # coarse dofs versus spectral tolerance



left: Operator complexity  
right: Number of coarse dofs

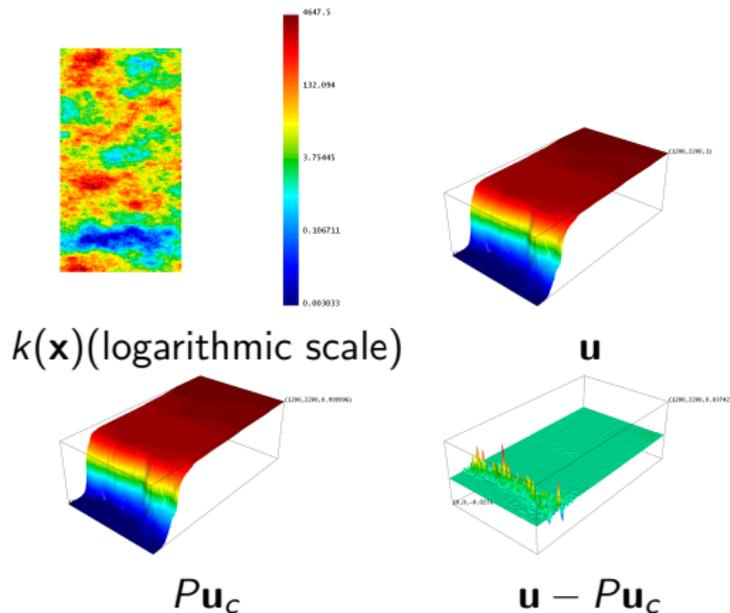
2 steps of refinement and varying  $\theta$  (212321 fine dofs,  $nnz(A) = 1904161$ , 1460 agglomerates)

# SA- $\rho$ AMGe: fixed $H/h$ and $\theta = 0.03$



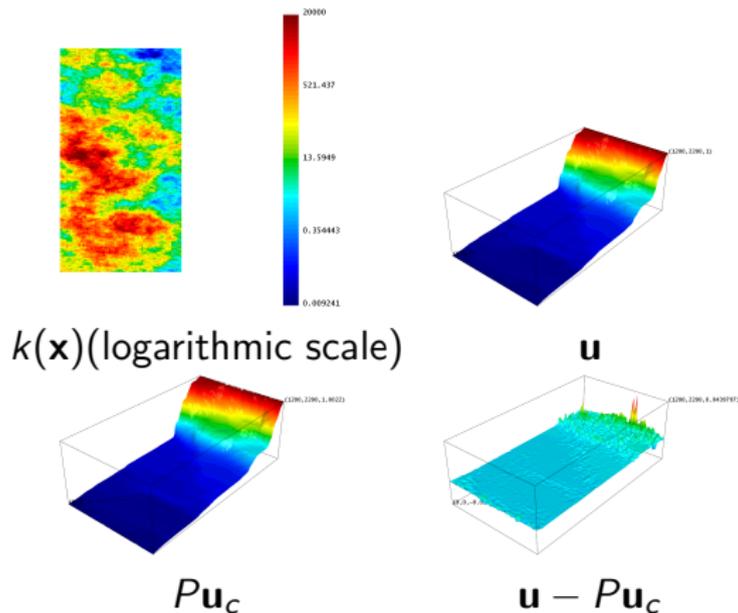
left:  $\|\mathbf{u} - P\mathbf{u}_c\|_{A(k)} / \|\mathbf{u}\|_{A(k)}$   
right:  $\|\mathbf{u} - P\mathbf{u}_c\|_{G(k)} / \|\mathbf{u}\|_{G(k)}$

# SA- $\rho$ AMGe: $k(\mathbf{x})$ , $\mathbf{u}$ , $P\mathbf{u}_c$ and $\mathbf{u} - P\mathbf{u}_c$



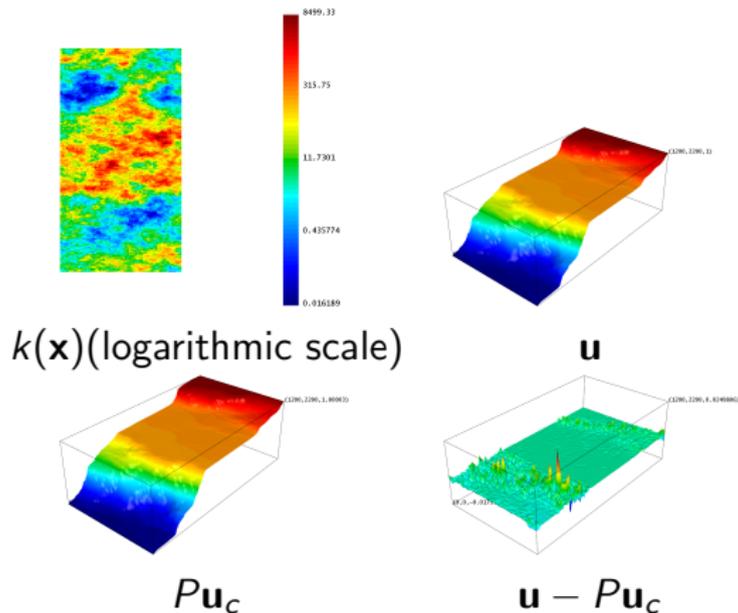
Slice 0 ( $\theta = 0.03$  and the mesh is refined once).

# SA- $\rho$ AMGe: $k(\mathbf{x})$ , $\mathbf{u}$ , $P\mathbf{u}_c$ and $\mathbf{u} - P\mathbf{u}_c$



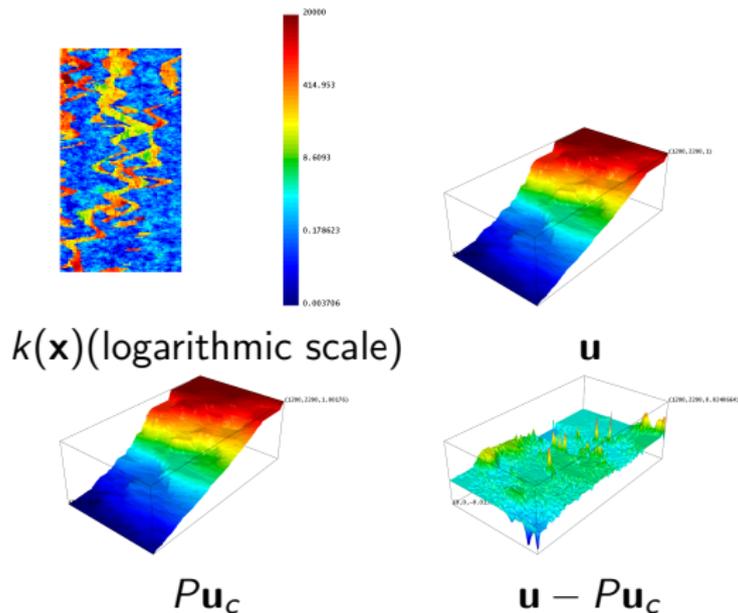
Slice 12 ( $\theta = 0.03$  and the mesh is refined once).

# SA- $\rho$ AMGe: $k(\mathbf{x})$ , $\mathbf{u}$ , $P\mathbf{u}_c$ and $\mathbf{u} - P\mathbf{u}_c$



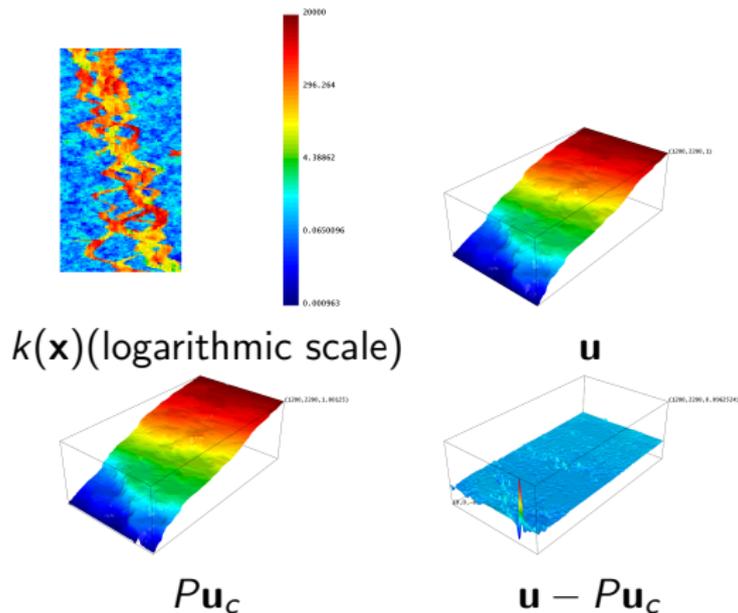
Slice 24 ( $\theta = 0.03$  and the mesh is refined once).

# SA- $\rho$ AMGe: $k(\mathbf{x})$ , $\mathbf{u}$ , $P\mathbf{u}_c$ and $\mathbf{u} - P\mathbf{u}_c$



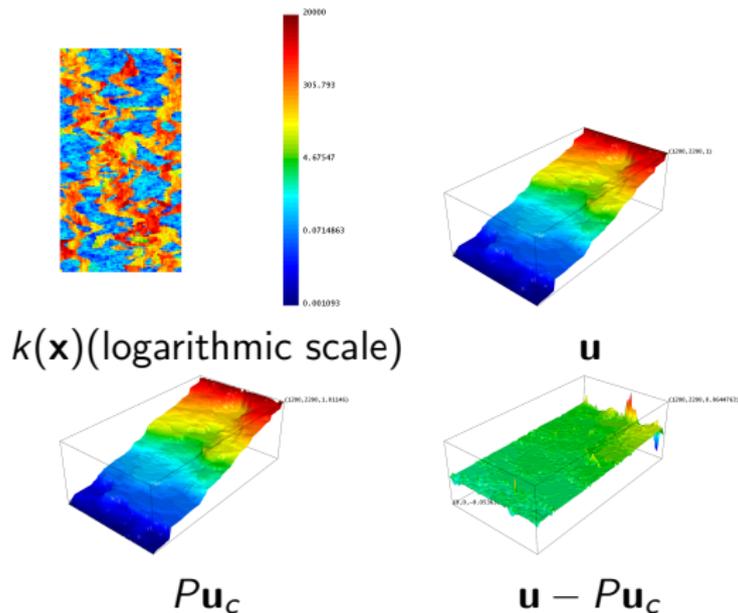
Slice 36 ( $\theta = 0.03$  and the mesh is refined once).

# SA- $\rho$ AMGe: $k(\mathbf{x})$ , $\mathbf{u}$ , $P\mathbf{u}_c$ and $\mathbf{u} - P\mathbf{u}_c$



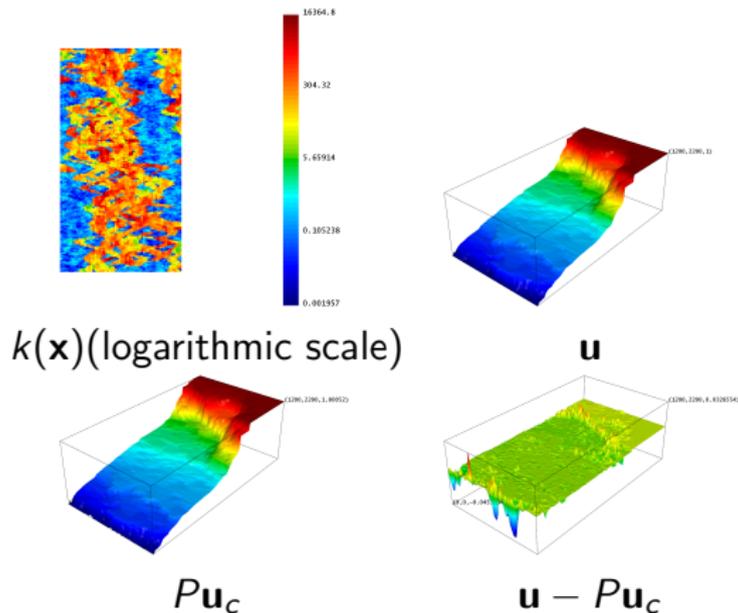
Slice 48 ( $\theta = 0.03$  and the mesh is refined once).

# SA- $\rho$ AMGe: $k(\mathbf{x})$ , $\mathbf{u}$ , $P\mathbf{u}_c$ and $\mathbf{u} - P\mathbf{u}_c$



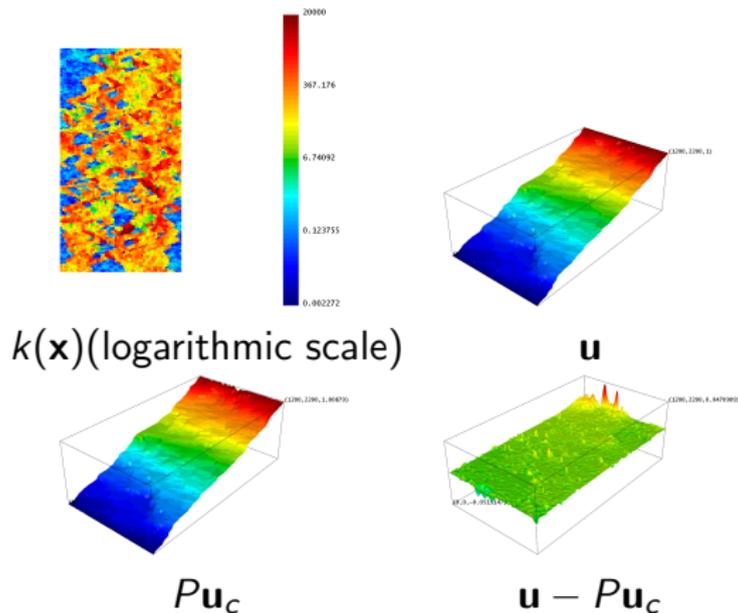
Slice 60 ( $\theta = 0.03$  and the mesh is refined once).

# SA- $\rho$ AMGe: $k(\mathbf{x})$ , $\mathbf{u}$ , $P\mathbf{u}_c$ and $\mathbf{u} - P\mathbf{u}_c$



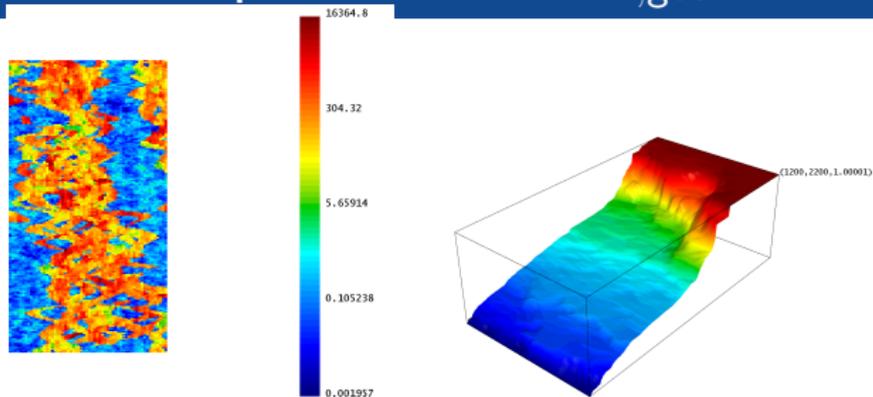
Slice 72 ( $\theta = 0.03$  and the mesh is refined once).

# SA- $\rho$ AMGe: $k(\mathbf{x})$ , $\mathbf{u}$ , $P\mathbf{u}_c$ and $\mathbf{u} - P\mathbf{u}_c$

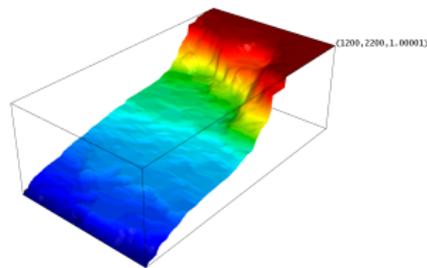


Slice 84 ( $\theta = 0.03$  and the mesh is refined once).

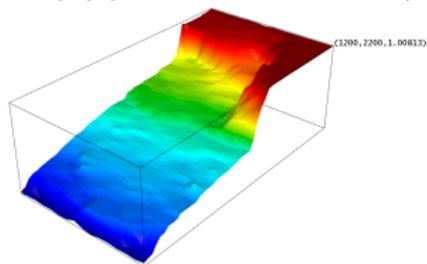
# SA- $\rho$ AMGe: Comparison with $\mathbf{u}_{H,\text{geom}}$



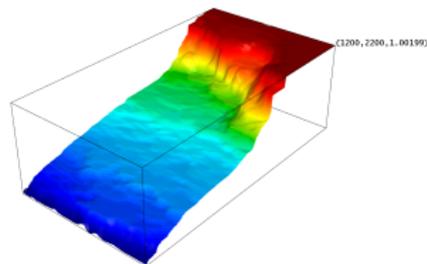
$k(\mathbf{x})$  (logarithmic scale)



$\mathbf{u}$



$\mathbf{u}_{H,\text{geom}}$

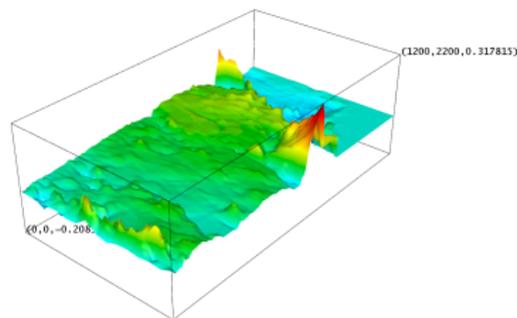


$P\mathbf{u}_c$

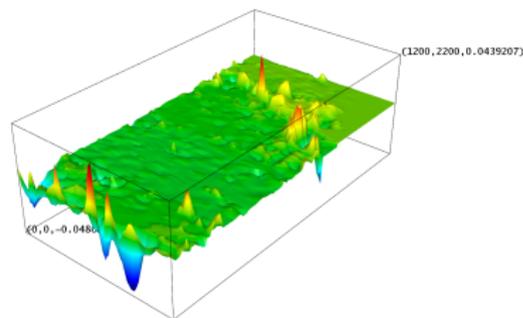
Slice 72: Fine-grid solution  $\mathbf{u}$ , geometric coarse-grid solution  $\mathbf{u}_{H,\text{geom}}$  ( $H/h = 4$ ), and spectral SA-AMGe coarse solution  $P\mathbf{u}_c$

( $H/h = 12$  and  $\theta = 0.03$ ).

# SA- $\rho$ AMGe: Comparison with $\mathbf{u}_{H,\text{geom}}$



$\mathbf{u} - \mathbf{u}_{H,\text{geom}}$



$\mathbf{u} - P\mathbf{u}_c$

Slice 72: Errors between fine-grid solution and geometric coarse-grid solution ( $H/h = 4$ ), and between fine-grid solution and spectral SA-AMGe coarse solution ( $H/h = 12$  and  $\theta = 0.03$ ).

# Main project: Coarsening de Rham complexes on agglomerated elements

$$\begin{array}{ccccccc} S_h & \xrightarrow{\nabla} & \mathbf{Q}_h & \xrightarrow{\text{curl}} & \mathbf{R}_h & \xrightarrow{\text{div}} & M_h \\ \Pi_H^S \downarrow & & \Pi_H^Q \downarrow & & \Pi_H^R \downarrow & & \Pi_H^M \downarrow \\ S_H & \xrightarrow{\nabla} & \mathbf{Q}_H & \xrightarrow{\text{curl}} & \mathbf{R}_H & \xrightarrow{\text{div}} & M_H \end{array}$$

- Initial approach (handles solvers only):

J. E. Pasciak and P. S. V., "Exact de Rham Sequences of Spaces Defined on Macro-elements in Two and Three Spatial Dimensions," SIAM Journal on Scientific Computing **30**(2008), pp. 2427-2446.

- Improved approach (handles solvers and upscaling) for  $H(\text{div})$ :

I. Lashuk and P. S. V., "Element Agglomeration Coarse Raviart-Thomas Spaces With Improved Approximation Properties," Numerical Linear Algebra with Applications **19**(2012) pp. 412-426.

- The general case:

I. Lashuk and P. S. V., "The Construction of Coarse de Rham Complexes with Improved Approximation Properties," Computational Methods in Applied Mathematics **14**(2)(2014), pp. 257-303.

- Use the coarse hierarchies for:
  - ① multilevel Monte Carlo and multilevel Markov Chain Monte Carlo (MCMC) methods.
  - ② adaptively changing the coarse hierarchies when the PDE coefficients change (as in MCMC).
  - ③ multilevel solvers for the upscaled problem (which can still be fairly large) for  $H(\text{div})$  and  $H(\text{curl})$  problems.
- Since the approach is algebraic, we can target non-PDE applications:
  - ① Coarsening of graphs (arising in various network simulations);
  - ② Coarse de Rham sequences for graph Laplacian.

## Standard Monte Carlo

The standard MC estimator for a quantity of interest  $E(Q)$  is

$$\frac{1}{N} \sum_{i=1}^N Q_h(\omega_i).$$

Its mean square error  $MSE$  is given by

$$MSE = \frac{1}{N} \text{Var} [Q_h] + (E [Q - Q_h])^2.$$

- The second term (which we do not have explicitly) is discretization error. It gets smaller when  $h \mapsto 0$ .
- The first term can be reduced by choosing large  $N$ .

# Application to MC simulations

- MC method requires generating a lot of samples which involves solving PDE on fine mesh, so even a single solve may pose a significant challenge.
- Our goal is to use the coarse spaces from the respective column(s) of the hierarchy of the coarse de Rham complexes to speed-up the MC process, using the MLMC.

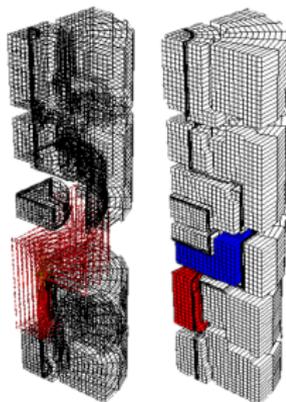


Figure: Coarse  $H(\text{div})$ -conforming shape function and its divergence.

## Multilevel Monte Carlo

The MLMC method from

M.B. Giles, "Multilevel Monte Carlo path simulation," *Operations Research*, **56**(3):607617, 2008.

relies on the multilevel decomposition

$$E[Q_h] = E[Q_L] + \sum_{l=1}^L E[Q_{l-1} - Q_l], \text{ where } Q_0 = Q_h.$$

The decomposition is useful since the mean square error is estimated as

$$MSE = \frac{1}{N_L} \text{Var}[Q_L] + \sum_{l=1}^L \text{Var}[Q_{l-1} - Q_l] + (E[Q - Q_0])^2.$$

The first term is on the coarsest mesh  $h_L$ , hence fixed, the intermediate terms have the property

$$\text{Var}[Q_{l-1} - Q_l] \ll \text{Var}[Q_l],$$

hence require much less samples, and the last one is the fine-grid discretization error.

In what follows, we apply the algorithm as described and analyzed in

K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup, "Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients," *Computing and Visualization in Science*, **14**(1):315, 2011.

# Estimating effective permeability in subsurface flow

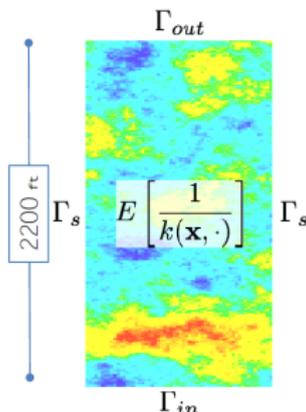
We solve mixed Darcy system:

$$\begin{aligned} k^{-1}(\mathbf{x}, \omega) \mathbf{u} + \nabla p &= 0, \\ \operatorname{div} \mathbf{u} &= 0, \end{aligned}$$

with boundary conditions  $p = 1$  on  $\Gamma_{in}$ ,  $p = 0$  on  $\Gamma_{out}$ ,  $\mathbf{u} \cdot \mathbf{n} = 0$  on  $\Gamma_s$ .

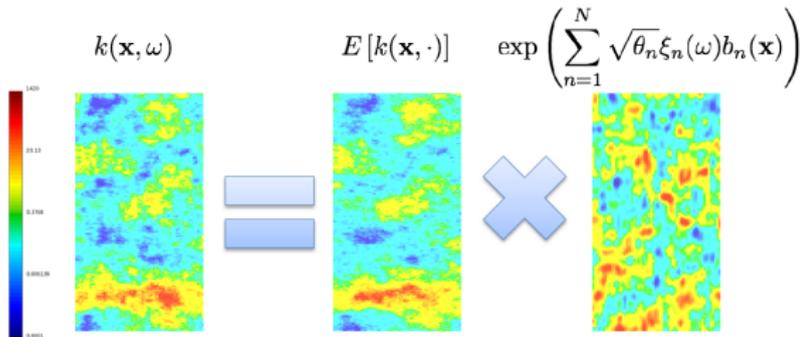
The quantity of interest  $Q = E[k_{eff}(\cdot)]$  is the expected value of

$$k_{eff} = \frac{1}{\Delta P} \int_{\Gamma_{out}} \mathbf{u}(\cdot, \omega) \cdot \mathbf{n} dS.$$



## Generating Samples

We use the truncated Karhunen-Loève expansion to generate spatially correlated random permeability (correlation length  $\lambda$ ).

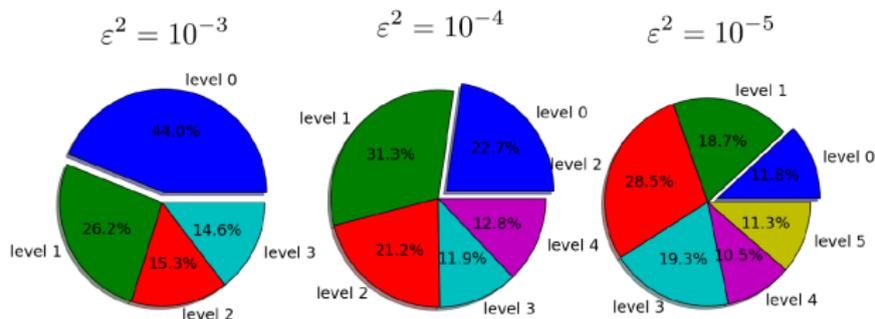


$E[k(\mathbf{x}, \cdot)]$  XY slice from SPE10 dataset

$\xi_n(\omega) \sim N(0, \sigma^2)$  i.i.d.

$(\theta_n, b_n(\mathbf{x}))$  eigenpairs of the covariance function  $C(\mathbf{x}, \mathbf{y}) = \int_{\Omega} \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|_p}{\lambda}\right) d\Omega$

## Multilevel acceleration of MC



The percentage of time spent on the fine grid decreases as we require for more accuracy and we allow for more levels.

# Parallel Multilevel Monte Carlo - Example

- Estimate water cut in top right well using the top layer of SPE10
- Parallel sampling with 60 cores.
- 38 eigenmodes in both directions & variance target:  $10^{-5}$

Level	#DoFs	#samples	$E[Q_l]$	$\text{Var}[Q_l - Q_{l+1}]$	Cost (seconds) per realization
Level 0 (fine)	66280	180	0.1112	0.00042182	100.8
Level 1	7409	840	0.1132	0.00149328	12.5
Level 2	639	2820	0.1211	0.00109587	0.75

MLMC:  $180 \times 100.8 + 840 \times 12.5 + 2820 \times 0.75 \approx$  **8.5** hours

MC needed 1020 realizations to converge:  $1020 \times 100.8 \approx$  **28.6** hours

Note: KLE not a scalable approach.

# Numerical upscaling of incompressible flow in reservoir simulation: governing equations

$$\mathbf{K}^{-1} \lambda^{-1}(S) \mathbf{u} + \nabla p = \left( \sum_{\alpha} \rho_{\alpha} f_{\alpha}(S) \right) g \nabla z \quad (1)$$

$$\nabla \cdot \mathbf{u} = q \quad (2)$$

$$\phi \frac{\partial S_{\alpha}}{\partial t} + \nabla \cdot \mathbf{u}_{\alpha}(S_{\alpha}) = \frac{q_{\alpha}}{\rho_{\alpha}}, \quad (3)$$

where

- $S_{\alpha}$  is the phase saturation
- $p$  is the pressure
- $\rho^{\alpha}$  is the mass density
- $q_{\alpha}$  is the source term
- $\phi$  is the porosity.
- $\alpha = o, w, g$ , and  $o$  stands for oil,  $w$  stands for water, and  $g$  stands for gas.
- $\mathbf{K}$  is the absolute permeability tensor
- $\mu_{\alpha}$  is viscosity
- $k_{r,\alpha}$  is relative permeability,
- and  $g$  is the gravitational acceleration.

Details in: ZHANGXIN CHEN, GUANREN HUAN, AND YUANLE MA, *Computational Methods for Multiphase Flows in Porous*

*Media*, Society for Industrial and Applied Mathematics, 2006.

# Model and solution scheme

- Two-phase incompressible rock and fluids
- Total velocity formulation (primary variables:  $\mathbf{u}$ ,  $p$ ,  $S$ )
- Improved IMPES (implicit for velocity and pressure, explicit for saturations)

This formulation leads to the solution of a large sparse indefinite linear system (saddle point problem) for velocity and pressure

$$\begin{bmatrix} M & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ P \end{bmatrix} = \begin{bmatrix} \mathbf{F}_u \\ F_p \end{bmatrix}, \quad (4)$$

followed by several explicit (Forward Euler) updates of the saturations

$$S_\alpha^{n+1} = S_\alpha^n + \Delta t \frac{1}{\phi} W^{-1} (F(S^n, \mathbf{u}) + \frac{1}{\rho_\alpha} W q_\alpha(S^n, p)) \quad (5)$$

Operator-dependent coarse spaces with **improved approximation properties** for velocity. 2-step method to find coarse velocity space:

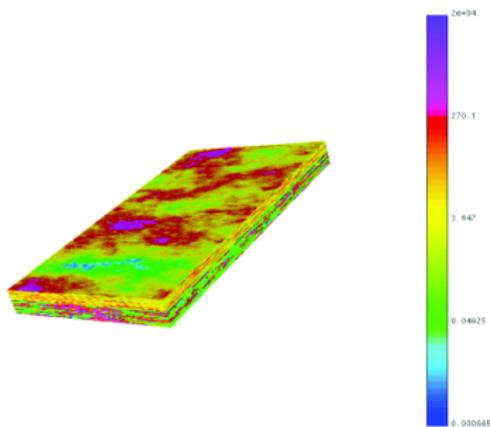
- 1 Singular Value Decomposition to find coarse basis functions on coarse faces.
  - 2 Solution of local saddle point problem to extend basis functions into the interior of the neighboring agglomerated elements.
- Possesses same stability and approximation properties as the original discretization.
  - Well-suited for **parallelization** with e.g. MPI(+OpenMP).

I.V. Lashuk and P.S. Vassilevski. Element agglomeration coarse Raviart-Thomas spaces with improved approximation properties.

Numerical Linear Algebra with Applications, 19(2):414-426, 2012

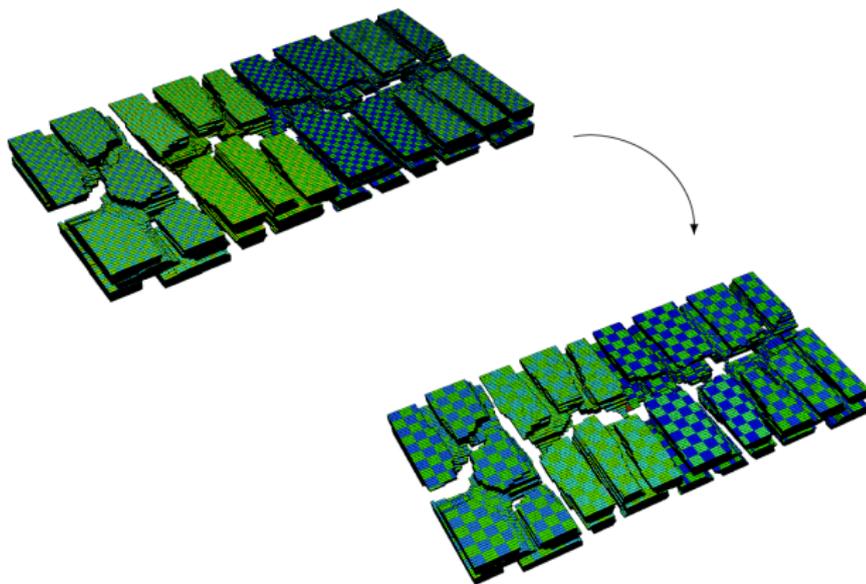
# Parallel upscaling

- SPE10: 60x220x85 elements
- Parallel subdomain split with METIS (48 cores)
- Cartesian agglomeration on parallel subdomains
- 3 levels (including fine grid) with coarsening factors: Level 0 $\Rightarrow$ 1: (2,4,1) and Level 1 $\Rightarrow$ 2: (2,2,2)
- 15 years of water injection



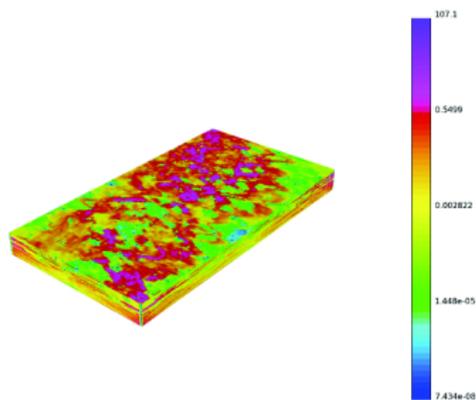
# Agglomeration in parallel

## Agglomeration in parallel

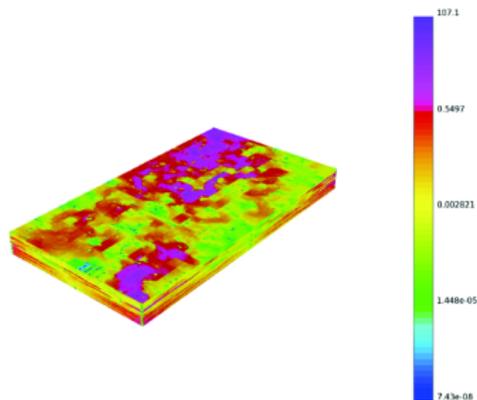


# Parallel results - total velocity after 15 years

## Parallel results - total velocity



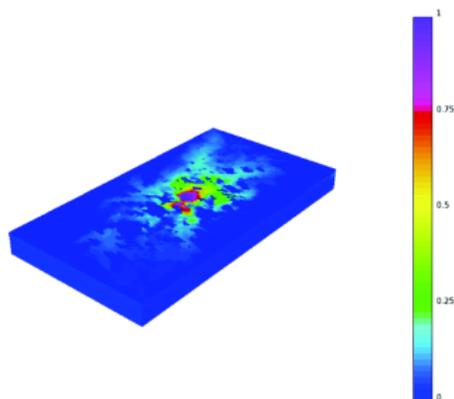
(a) Level 0 (fine grid)



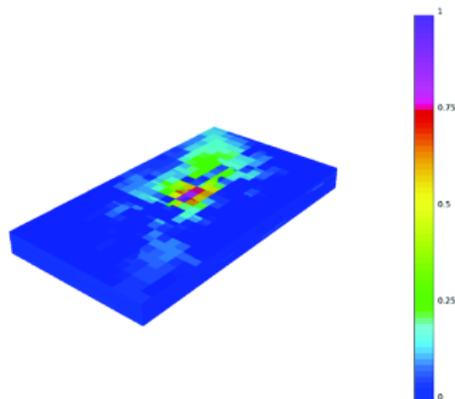
(b) Level 2 (x64)

# Parallel results - saturation of water after 15 years

## Parallel results - saturation of water



(c) Level 0 (fine grid)



(d) Level 2 (x64)

# Numerical upscaling of incompressible flow in reservoir simulation: SAIGUP mesh

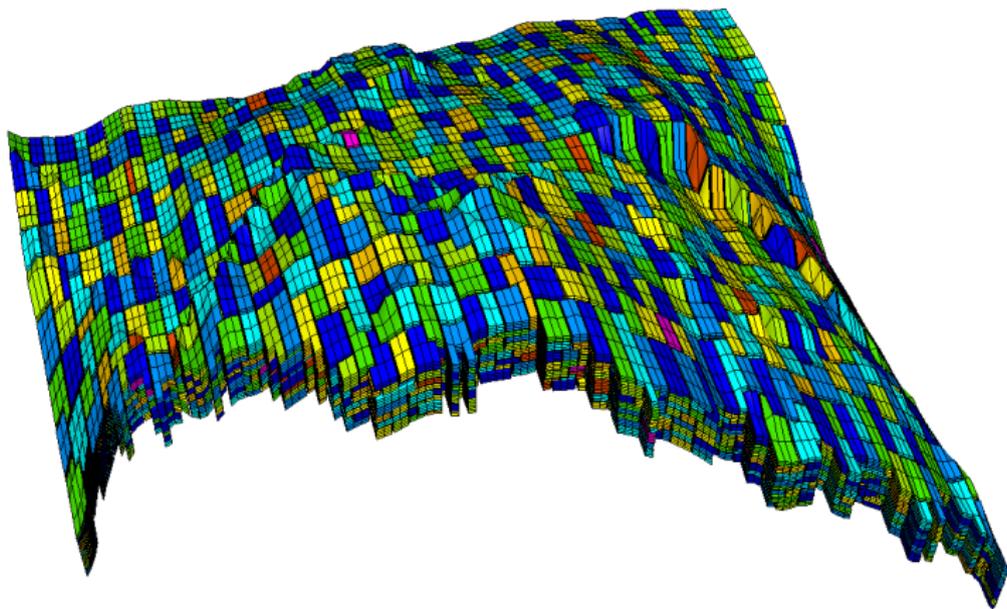


Figure: Full coarsening: 16 fine elements per agglomerate

# Numerical upscaling of incompressible flow in reservoir simulation

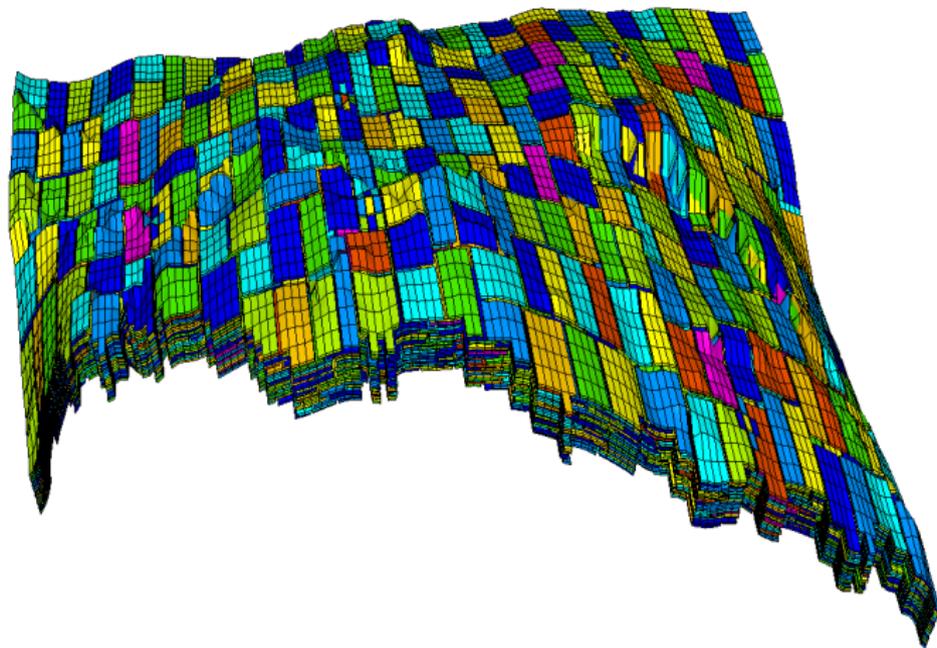


Figure:  $(x, y)$ -semi-coarsening: 16 fine elements per agglomerate

# Numerical upscaling of incompressible flow in reservoir simulation

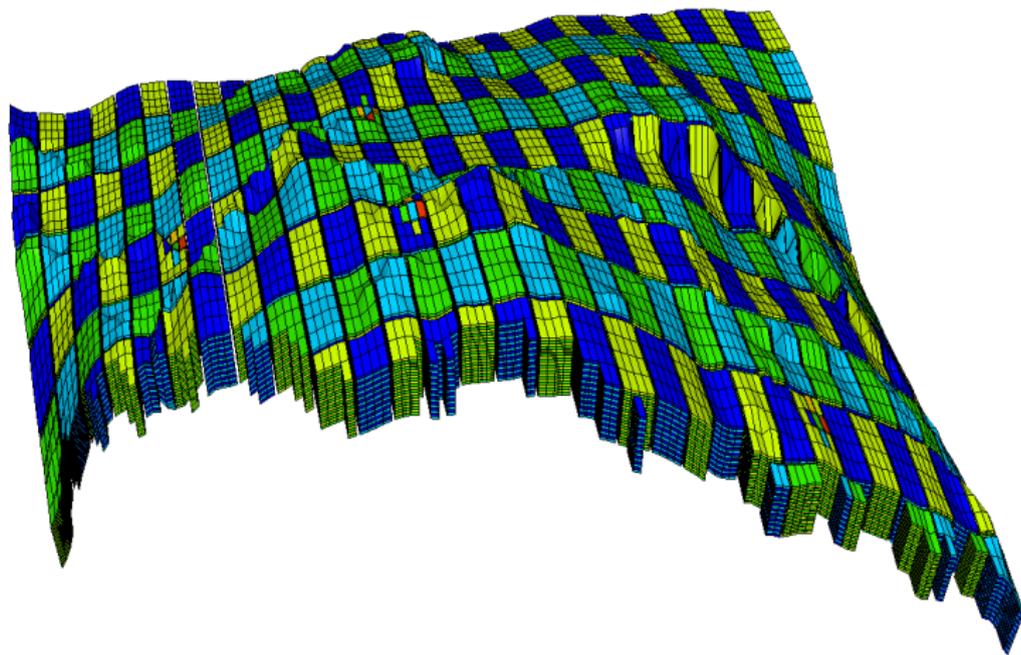


Figure: Structured  $(x, y)$ -semi-coarsening (Cartesian semi): 16 fine elements per agglomerate

# Numerical upscaling of incompressible flow in reservoir simulation

We used the coarse Raviart-Thomas  $H(\text{div})$  spaces from the AMGe multilevel hierarchy of the de Rham sequence to discretize the mixed f.e. problem for the total velocity.

Problem	#elements	#faces	#DoFs	nnz	arithmetic complexity	operator complexity
Fine grid	78720	243576	479736	3549946	-	-
Full coarsening (4)	17210	82081	168060	3039542	1.41412	1.85622
Full coarsening (16)	4920	25633	65422	1620930	1.17211	1.45661
Full coarsening (4)*	17960	84401	171960	3054520	1.4221	1.86044
Full coarsening (16)*	5616	28108	69924	1697050	1.18211	1.47805
Semi coarsening (4)*	14761	73982	173180	4080239	1.44573	2.14938
Semi coarsening (16)*	5629	30408	81513	2564226	1.21798	1.72233
Cartesian semi (4)*	20660	64372	175336	2194200	1.41582	1.61809
Cartesian semi (16)*	5970	19001	61025	901477	1.1523	1.25394
Cartesian semi (64)*	2100	6653	20683	318655	1.05114	1.08976

**Table:** Degrees of freedom, number of non-zeros and complexities. \* means elements with wells and immediate neighbor elements of wells are left unagglomerated.

# Numerical upscaling of incompressible flow: water saturation results

Next figures show water saturation after 30 years of injection for both the fine grid and upscaled solutions. Elements with production wells are marked with a pink color.

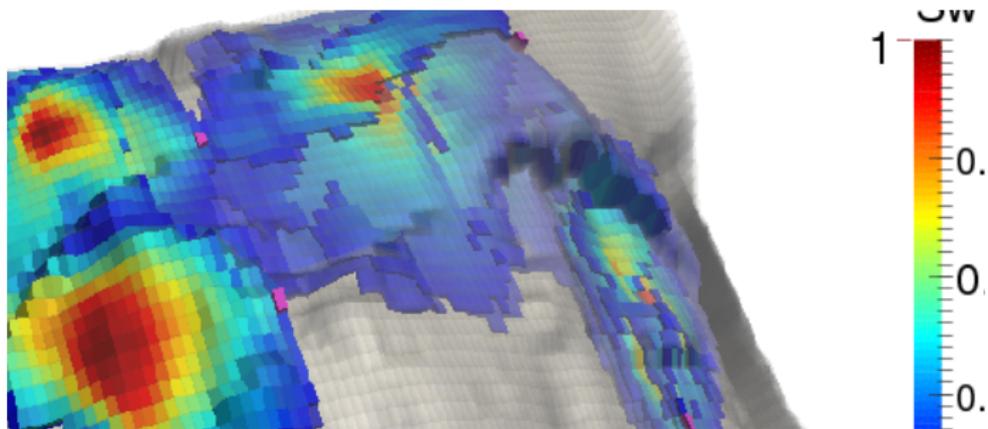


Figure: Fine grid reference solution.

# Numerical upscaling of incompressible flow: water saturation results

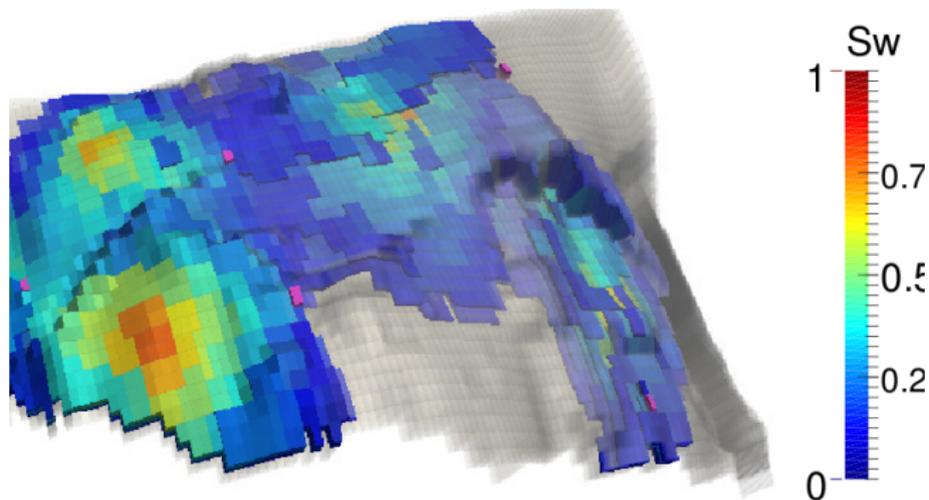


Figure:  $(x, y)$ -semi-coarsening (4 fine elements per agglomerate).

# Numerical upscaling of incompressible flow: water saturation results

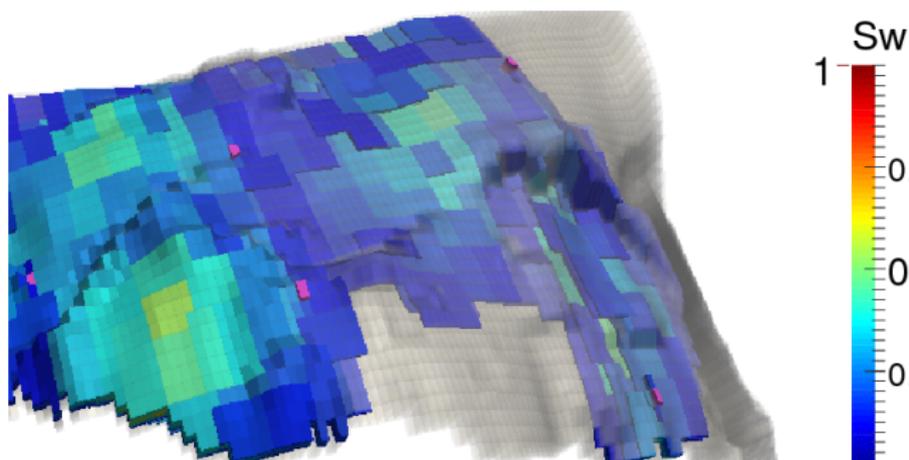


Figure:  $(x, y)$ -semi-coarsening (16 fine elements per agglomerate).

# Numerical upscaling of incompressible flow in reservoir simulation

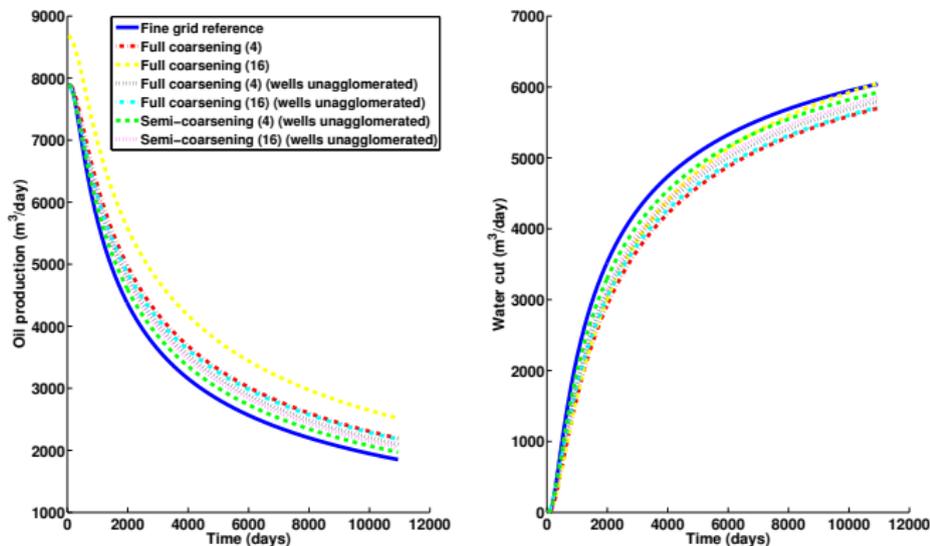


Figure: Daily oil production and water cut for fine grid  $60 \times 220$  and upscaled models.

# Numerical upscaling of incompressible flow in reservoir simulation

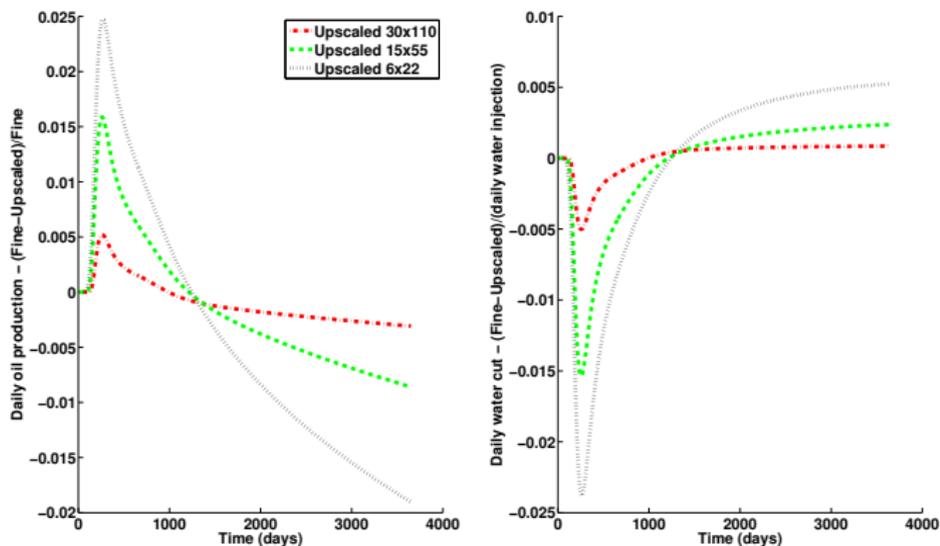


Figure: Difference between fine grid  $60 \times 220$  and upscaled models in terms of daily production.

# Numerical upscaling of incompressible flow in reservoir simulation

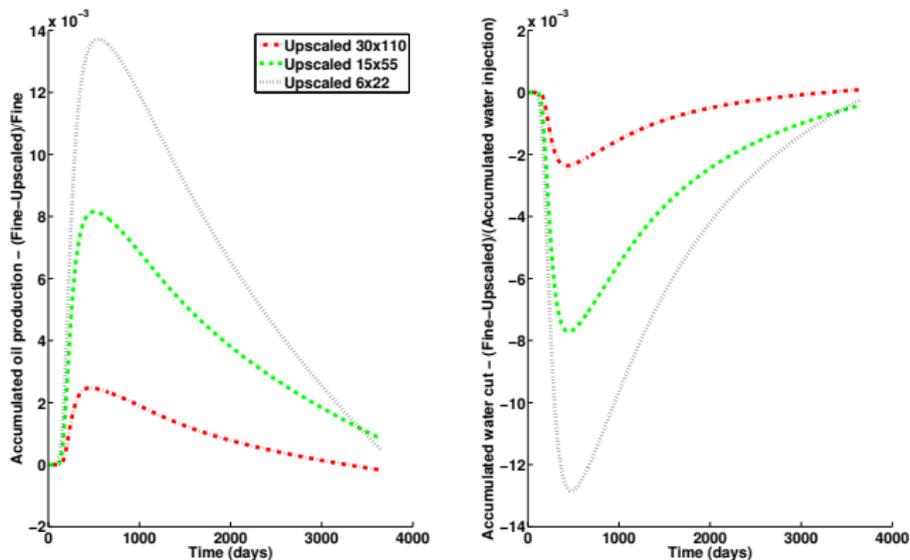


Figure: Difference between fine grid  $60 \times 220$  and upscaled models in terms of accumulated production.

# The coarse Raviart–Thomas portion of the de Rham complex: some details

We are given  $\mathbf{R}_h \subset H(\text{div})$  and  $W_h \subset L_2$  on a simplicial triangulation  $\mathcal{T}_h$  of  $\Omega \subset \mathbb{R}^3$ .

We have generate agglomerated elements  $T$  that cover all fine-grid elements with some regular topology.

Assumption:

(A) On each  $T$  we are given a set of function  $\{\bar{\mathbf{r}}_i^{(T)}\}$  and  $\{\bar{\rho}_j^{(T)}\}$  that we want to include (locally) in the coarse pair of spaces  $\mathbf{R}_H$  and  $W_H$ .

We assume

$$\text{div } \bar{\mathbf{r}}_i^{(T)} \in \{\bar{\rho}_j^{(T)}\}.$$

The constant is contained in the set  $\{\bar{\rho}_j^{(T)}\}$ , which is  $L_2(T)$ -orthogonal.

# The coarse Raviart–Thomas portion of the de Rham complex: some details

On each coarse face  $F = \partial T_- \cap \partial T_+$ , we choose a linearly independent set of traces  $\{\mathbf{r}_H^F \cdot \mathbf{n}_F\}$  that span the traces of the given sets  $\{\bar{\mathbf{r}}_i^{(T-)} \cdot \mathbf{n}_F\}$  and  $\{\bar{\mathbf{r}}_i^{(T+)} \cdot \mathbf{n}_F\}$  plus the function with  $\mathbf{1}$  constant normal trace on  $F$ .

- Each trace  $\mathbf{r}_H^F \cdot \mathbf{n}_F$  extended by zero on all other faces of  $T$ , is extended in the interior of  $T$  by solving the local mixed problem

$$\begin{aligned} a_T(\mathbf{r}_H^F, \mathbf{v}) + (\operatorname{div} \mathbf{v}, p)_T &= 0, \text{ for all } \mathbf{v} \in \mathbf{R}_h(T), \mathbf{v} \cdot \mathbf{n} = 0 \text{ on } \partial T \\ (\operatorname{div} \mathbf{r}_H^F, q)_T &= \operatorname{const}(1, q)_T, \text{ for all } q \in W_h(T). \end{aligned}$$

- Complete the spaces by adding element bubbles:
  - $\mathbf{r}_H^0$  such that  $\operatorname{div} \mathbf{r}_H^0 = \bar{p}_i^{(T)}$  for all  $\bar{p}_i^{(T)}$  with zero meanvalue on  $T$ ;
  - For any given  $\bar{\mathbf{r}}_i$  subtract its  $F$ -interpolant spanned by the  $F$ -based basis functions. The difference is an element bubble. We add its orthogonal complement to the previous set of bubbles.

Define  $\mathbf{R}_H = \operatorname{span}\{\mathbf{r}_H^F, \mathbf{r}_H^0\}$ . By construction (and assumption) we have

$$\operatorname{div} \mathbf{R}_H = W_H.$$

# The coarse Raviart–Thomas portion of the de Rham complex: some examples

Examples of the sets  $\{\bar{\mathbf{r}}_i^{(T)}\}$  and  $\{\bar{p}_j^{(T)}\}$ .

- The same piecewise polynomials that are contained in the fine-grid spaces  $\mathbf{R}_h$  and  $W_h$ .
- For  $\{\bar{p}_j^{(T)}\}$ , and the traces  $\bar{\mathbf{r}}_i^F$ , we can use the following extension of the spectral AMGe method:

On each  $T$  solve the eigenvalue problem for  $\mu$ ,  $p$  and  $\mathbf{u}$ :

$$\begin{aligned} a_T(\mathbf{u}, \mathbf{v}) + (\operatorname{div} \mathbf{v}, p) + (\mu, \mathbf{v} \cdot \mathbf{n})_{\partial T} &= 0, \\ (\operatorname{div} \mathbf{u}, q) &= -\lambda(p, q), \\ (\mathbf{u} \cdot \mathbf{n}, \theta)_{\partial T} &= -\lambda(\mu, \theta)_F. \end{aligned}$$

Then from the lower portion of the spectrum, we select the respective  $p$  and  $\mu$  and set  $\bar{p}_j^{(T)} = p$  and  $\bar{\mathbf{r}}_i^F = \mu|_F$  for each  $F \subset \partial T$ .

# The coarse de Rham complex: Nédélec and $H^1$ -conforming spaces

Given a set of functions  $\{\bar{\mathbf{q}}_k^{(T)}\}$ :  $\text{curl } \bar{\mathbf{q}}_k^{(T)}$  is spanned by the RT set  $\{\bar{\mathbf{r}}_i^{(T)}\}$ .

- Use the traces on coarse edges, extend them into each coarse face  $F$  by solving a local mixed problem of each  $F$ , and add face bubbles.
- Extend the constructed face data in the interior of the agglomerated elements and add element bubbles.

The construction ensures:

- A coarse divergence-free function is a curl of a coarse Nédélec function;
- There are two computable projections such that

$$\text{curl } \pi_H^{\mathbf{Q}} \mathbf{q} = \pi_H^{\mathbf{R}} \text{curl } \mathbf{q} \text{ for any } \mathbf{q} \in \mathbf{Q}_h.$$

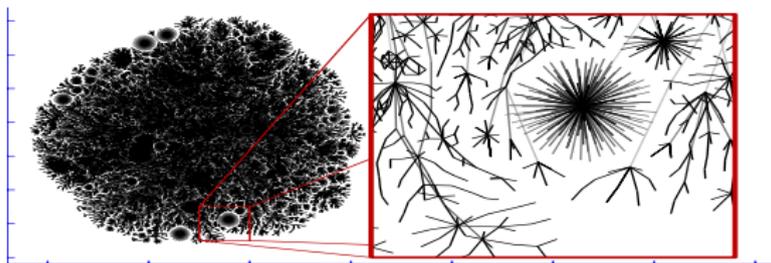
The final pair of spaces  $S_h \subset H^1$  and  $\mathbf{Q}_h \subset H(\text{curl})$  is handled similarly.

I. Lashuk and P. S. V., "The Construction of Coarse de Rham Complexes with Improved Approximation Properties,"

Computational Methods in Applied Mathematics **14**(2)(2014), pp. 257-303.

# Coarse de Rham complexes on graphs

A graph  $G$  consist of a set of vertices  $V$  and a set of edges  $e = (i, j) \in E$ ,  $i, j \in V$ .



An internet graph from <http://opte.org>

- a vertex-based space  $S$ ;
- an edge-based space  $\mathbf{U}$ .

$$(\text{Grad } v)(e) = \epsilon_e(v_i - v_j), \quad e = (i, j).$$

$$\text{Div} = (-\text{Grad})^T.$$

Graph Laplacian  $L$ :

$$(Lu, v) = \sum_{e=(i,j) \in E} (\text{Grad } v)(e)(\text{Grad } u)(e).$$

# Coarse de Rham complexes on graphs

Graph Laplacian system  $Lp = f$  can be written in a mixed form:

$$\begin{aligned} \mathbf{u} + \text{Grad } p &= 0, \\ \text{Div } \mathbf{u} &= f. \end{aligned}$$

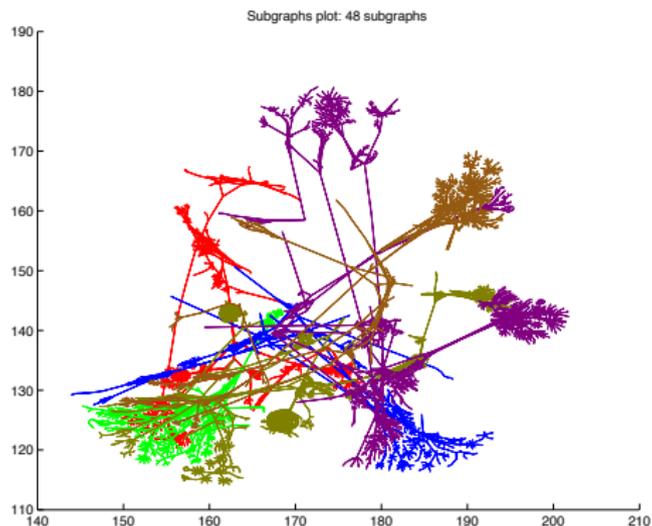
- edges as “elements”
- the local quadratic forms  $L_e(u, v) = (\text{Grad } v)(e)(\text{Grad } u)(e)$  as “element matrices”.

Using [edge-agglomeration](#), the AMGe constructs coarse edge space  $\mathbf{U}_H$  and a coarse vertex space  $S_H$  (of piecewise constants). Also there is a computable projection  $\pi_H$ , such that (P.S.V. and Ludmil Zikatanov (2014))

$$\begin{array}{ccc} \mathbf{U} & \xrightarrow{\text{Div}} & S \\ \pi_H \downarrow & & \downarrow Q_H \\ \mathbf{U}_H & \xrightarrow{\text{Div}} & S_H \end{array}$$

is commuting where  $Q_H$  is the  $\ell_2$ -based projection on the space  $S_H$ .

# Coarse de Rham complex for graphs



Plot of the neighborhood of a subgraph. Different colors indicate the different values of the piece-wise constant divergence.

- We have developed an element agglomeration AMGe methodology for coarsening entire de Rham complex of fine-grid spaces, so that
  - the coarse spaces in the complex contain any pre-selected sets of fine-grid functions locally.
  - the coarse complex is also exact;
  - we have access to computable projections operators that ensure commutativity with the respective differential operators, and gives computable coarse operators

$$\nabla_H, \text{curl}_H, \text{div}_H,$$

represented as sparse matrices. Useful for discretization and solver purposes.

- We demonstrated that the coarse hierarchy is useful for numerical upscaling and for running MLMC simulations more effectively on general unstructured agglomerates.