# Adaptive multigrid
## via subcycling on complementary grids

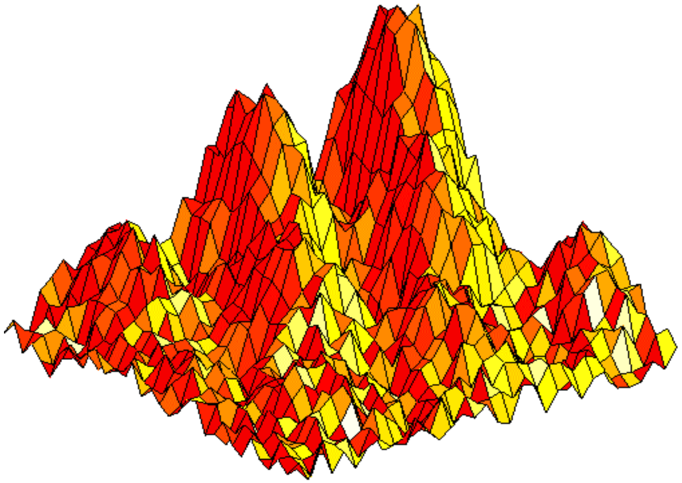
Tim Chartier
Department of Mathematics
University of Washington
chartier@math.washington.edu

# Collaborators

- This work was funded under a Research and Development grant at Lawrence Livermore National Lab.
- Primarily collaboration with Edmond Chow
- Thanks also extended to the following:
  - Steve McCormick
  - John Ruge
  - Tom Manteuffel
  - Marian Brezina
  - Rob Falgout

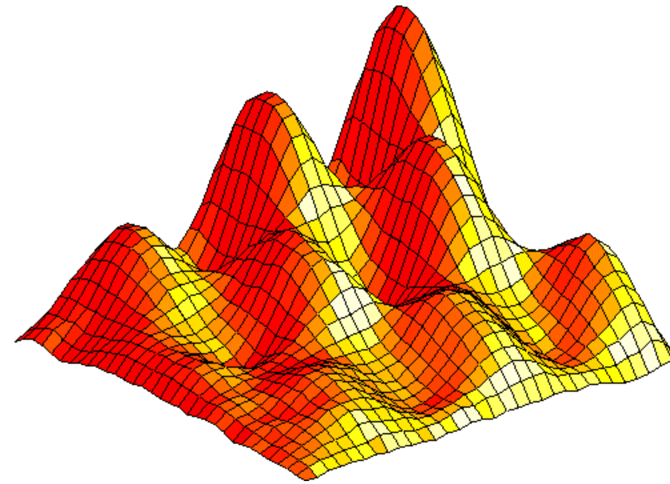
# Multigrid cycle

## for elliptic problems



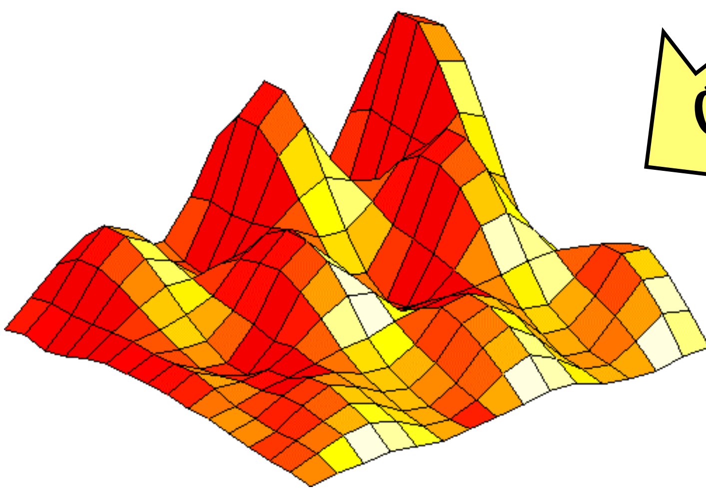
Initial Error on fine grid

Relax



Smoothed Error on fine grid

CG Transfer



Smoothed Error on coarse grid

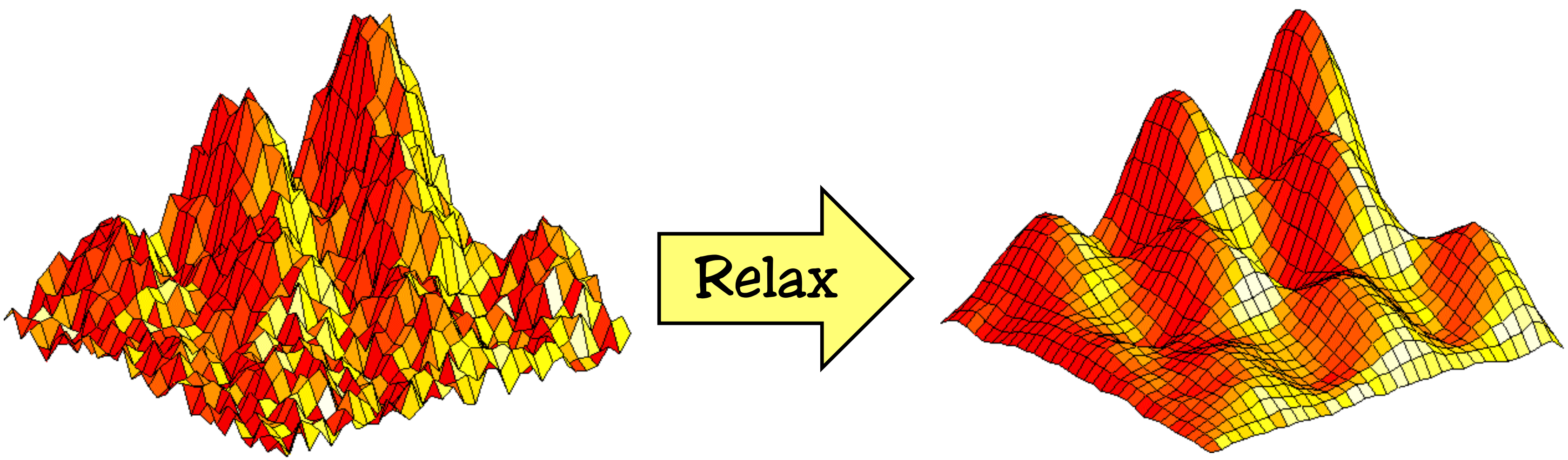
Error must be reduced by

- relaxation
- coarse–grid correction

# Smooth error

So, relaxation (Jacobi or Gauss-Siedel) produced geometrically smooth error for the PDE:

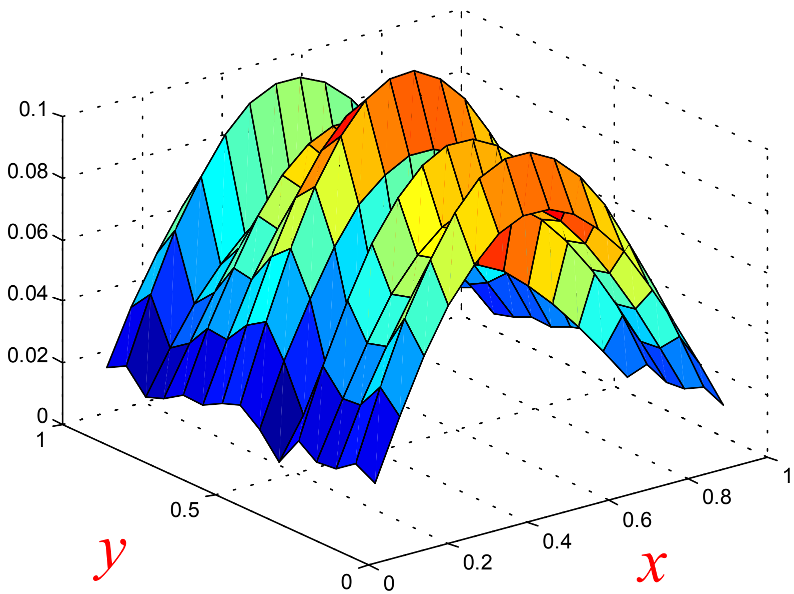
$$u_{xx} + u_{yy} = f$$




**Initial Error** **Smoothed Error**

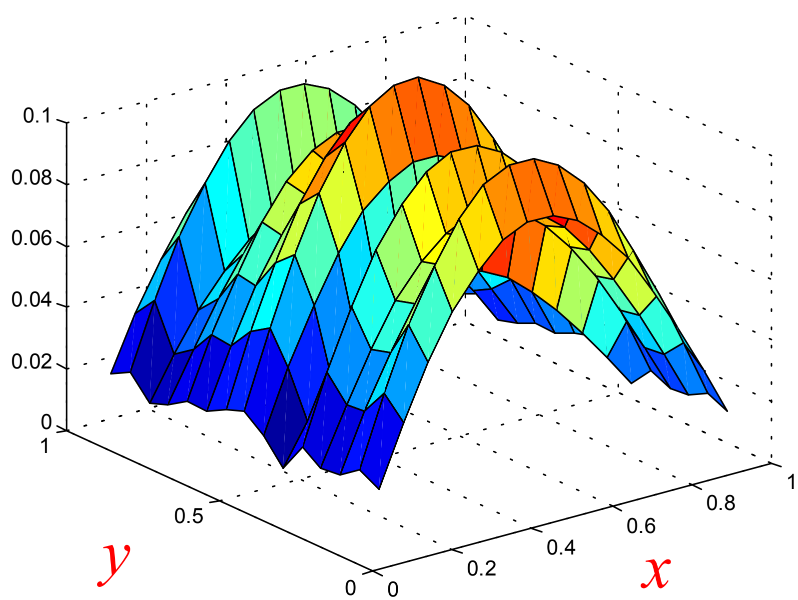
# Algebraically smooth error

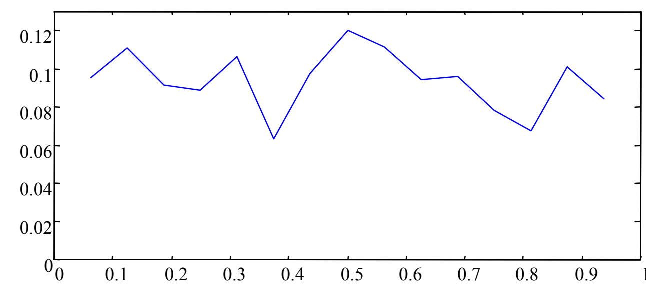
Consider $u_{xx} + (0.001)\, u_{yy} = f$. Then 50 sweeps of Gauss-Seidel produce the following error:



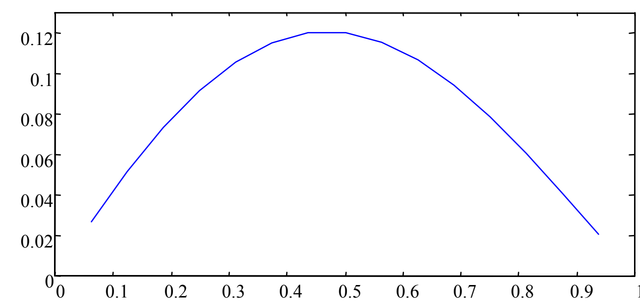
# Algebraically smooth error

Another look at what it means to be smooth can be seen in the following observation:




Error along line of constant *x*




Error along line of constant *y*

*Algebraically smooth error* is not always geometrically smooth. Automatically choosing appropriate interpolation weights is a goal of algebraic methods.

# Algebraically smooth error

Assume standard relaxation methods, such as Richardson.

$$\mathbf{u}_1 = \mathbf{u}_0 + \frac{\omega}{\|A\|_2}\mathbf{r}_0 \Rightarrow \mathbf{e}_1 = \left( I - \frac{\omega}{\lambda_n} A \right)\mathbf{e}_0$$

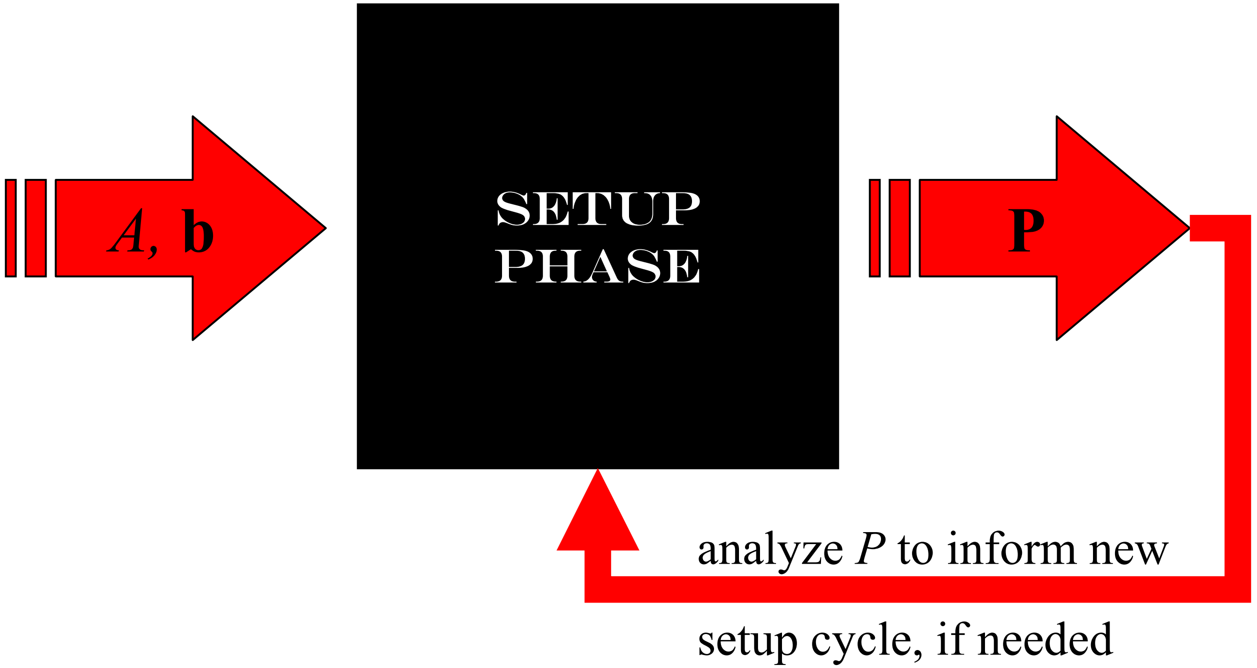
$$\text{If } A\mathbf{e}_0 = \lambda\mathbf{e}_0, \text{ then } \mathbf{e}_1 = \left( 1 - \omega\frac{\lambda}{\lambda_n} \right)\mathbf{e}_0$$

$$\text{If } \lambda << \lambda_n \Rightarrow \mathbf{e}_1 \approx \mathbf{e}_0$$

$$\lambda \approx \lambda_n \quad \Rightarrow \text{factor} \approx 1 - \omega$$

Eigenvectors associated with small eigenvalues of $A$ must be approx. by coarse–grid correction

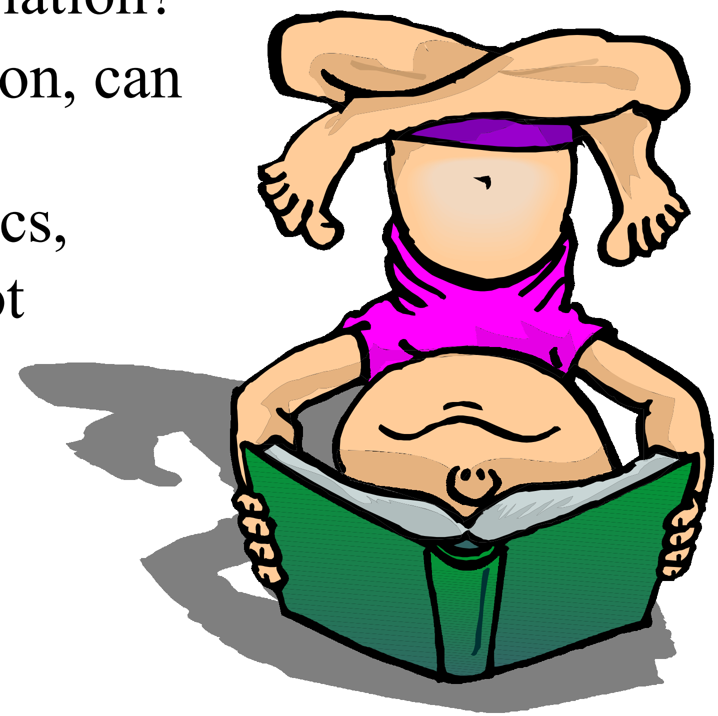
# Adaptive MG




- Since relaxation is fixed, the goal of adaptive MG schemes is to choose an effective interpolation matrix $P$ ($R = P^T$).
- If $P$ appears ineffective, design an algebraic algorithm to improve prolongation.

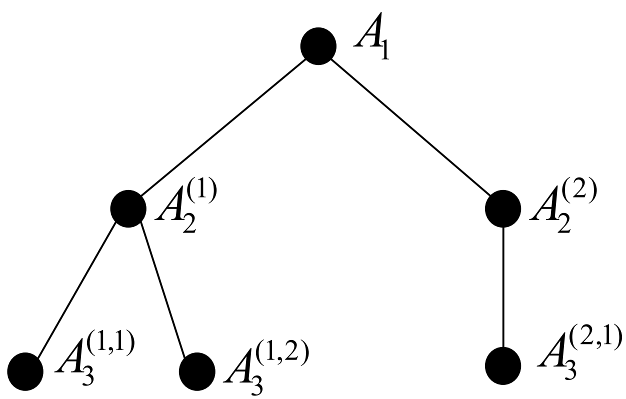
# Improving ineffective interpolation

- If $P$ is not effective, then interpolation is not approximating low mode(s).
- Can the method self-improve interpolation?
- Without doing a spectral decomposition, can we determine the “missed” modes?
- As indicated by the earlier mathematics, the method produces a error that is not captured by the method—a linear combination of the “missed” modes.



# Subcycling on complementary grids

The subcycling takes the following form:




1. Relax $A_1 v_1 = 0$ with random initial guess. Use relaxed vector to form $P_1^{(1)}$ (and $A_2^{(1)}$).
2. Relax $A_2^{(1)} v_2 = 0$ with all-ones initial guess. Use relaxed vector to form $P_2^{(1,1)}$ (and $A_3^{(1,1)}$).

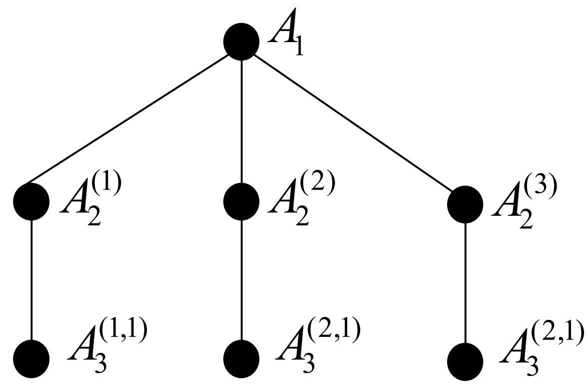
**----- Base cycle complete -----**

3. Test $A_2^{(1)} v_2 = 0$ (using random guess). If conv. slow (as in this e.g.), use relaxed vector to form $P_2^{(1,2)}$ (and $A_3^{(1,2)}$).

⋮

# Subcycling on complementary grids

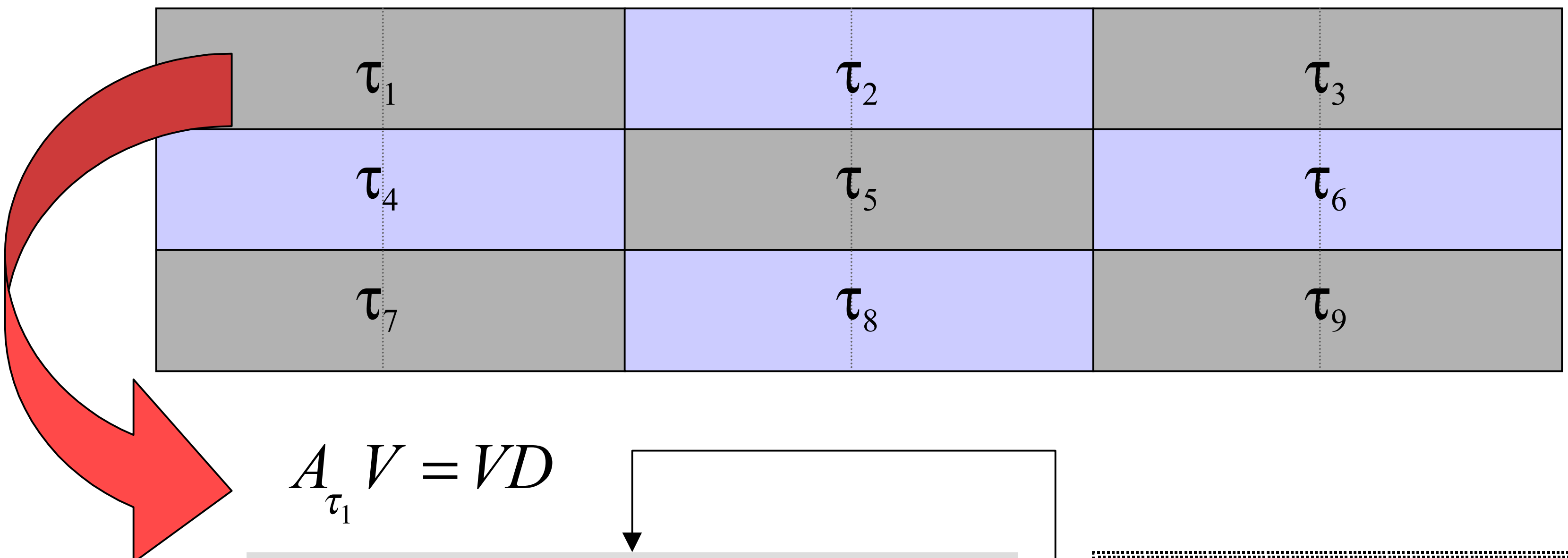
A cheaper variant of this scheme would take the form:




- This defines a general cycling (specifically subcycling).
- How is prolongation formed?

# Spectral AMGe

Let $\mathcal{A}$ be a partition of the space into agglomerates $\tau$.



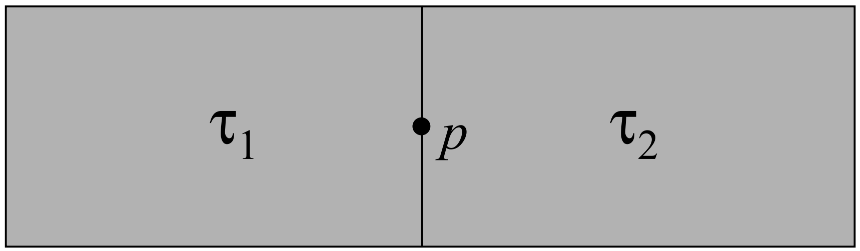

$$A_{\tau_1} V = VD$$

$$P_{\tau_1} = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_{m_\tau} \\ | & | & & | \end{bmatrix}$$

**Idea:** Use representative vectors from adaptive scheme.

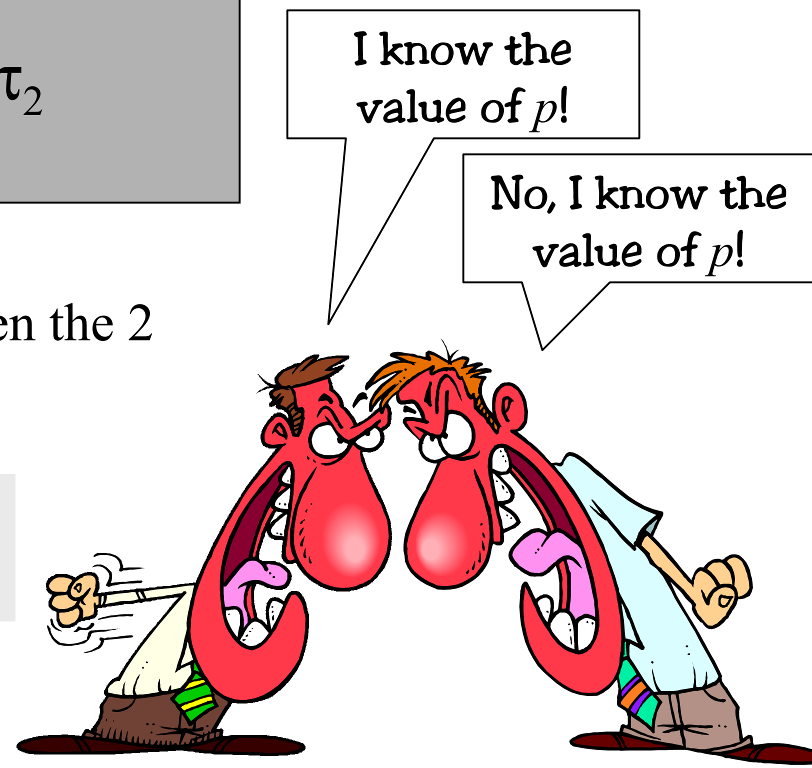
# Forming the global P

What about $p \in \partial\tau$ ?




A **conflict** arises for the value of $p$ between the 2 agglomerates

Let $p$ be the average of the formulas determined in each of its agglomerates.



# Forming the global P

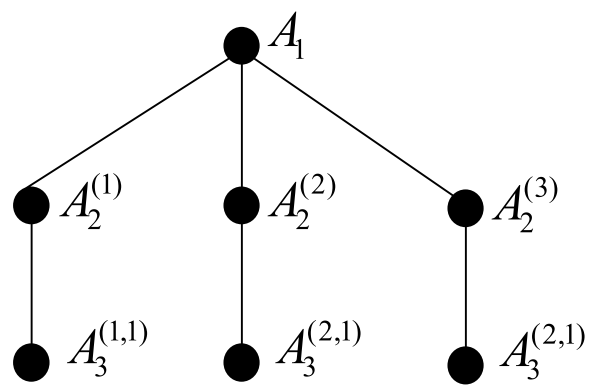
Assume $x$ is a global smooth vector.
For each $\tau \in \mathcal{A}$, form an intra-agglomerate interpolation vector $p_\tau$ as follows:

- Let $x^\tau$ be the restriction of $x$ to agglomerate $\tau$.
- For each dof $i$ in $\tau$ that lies on the boundary of $n_i > 1$ agglomerates, let the i^th component of $x^\tau$ denoted $x_i^\tau$ equal $x_i^\tau / n_i$
- Form $p_\tau$ by extending $x^\tau$ with zeros outside of $\tau$.

The columns of the global interpolation operator P are the vectors $p_\tau$ for all $\tau \in \mathcal{A}$.

# Numerical results

The following numerical experiments utilize the less costly subcycling scheme.




All problems will have Dirichlet boundary and utilize 4 x 4 agglomerates

# Numerical results

34 x 34 Poisson problem square elements

| *k* | Conv. | Subcycles | Grid Complexity | Operator Complexity |
|---|---|---|---|---|
| 2 | 0.20 | 2 | 1.12 | 1.10 |
| 3 | 0.20 | 2 | 1.15 | 1.12 |
| 4 | 0.17 | 2 | 1.15 | 1.13 |

34 x 34 Poisson problem rectangle elements (5:1)

| *k* | Conv. | Subcycles | Grid Complexity | Operator Complexity |
|---|---|---|---|---|
| 2 | 0.22 | 6 | 1.35 | 1.31 |
| 3 | 0.27 | 6 | 1.44 | 1.37 |
| 4 | 0.25 | 7 | 1.54 | 1.45 |

**Note:** Same result if fine level linear equation is diagonally scaled.

# Numerical results

34 x 34 rotated anisotropic diffusion; square elements, $\theta = 0^o$

| *k* | Conv. | Subcycles | Grid Complexity | Operator Complexity |
|---|---|---|---|---|
| 2 | 0.24 | 10 | 1.59 | 1.51 |
| 3 | 0.24 | 11 | 1.81 | 1.68 |
| 4 | * | * | * | * |

34 x 34 rotated anisotropic diffusion; square elements, $\theta = 30^o$

| *k* | Conv. | Subcycles | Grid Complexity | Operator Complexity |
|---|---|---|---|---|
| 2 | 0.22 | 7 | 1.41 | 1.36 |
| 3 | 0.24 | 7 | 1.51 | 1.43 |
| 4 | 0.21 | 8 | 1.62 | 1.51 |

34 x 34 rotated anisotropic diffusion; square elements, $\theta = 45^o$

| *k* | Conv. | Subcycles | Grid Complexity | Operator Complexity |
|---|---|---|---|---|
| 2 | 0.21 | 8 | 1.47 | 1.41 |
| 3 | 0.21 | 8 | 1.59 | 1.50 |
| 4 | 0.26 | 7 | 1.51 | 1.43 |

# Current work

Current and future work includes:

- Continue multilevel testing on problems such as linear elasticity
- Compare efficiency with other adaptive methods
- Consider more costly subcycling scheme
- Combining information from each level