SAND2008-3745C

Remap of solenoidal fields on unstructured quadrilateral grids

Pavel Bochev
Applied Mathematics and Applications
Sandia National Laboratories
MS-1320 Albuquerque, NM 87185

Mikhail Shashkov
Theoretical Division T-7
Los Alamos National Laboratory
MS-B284, Los Alamos, NM 87545

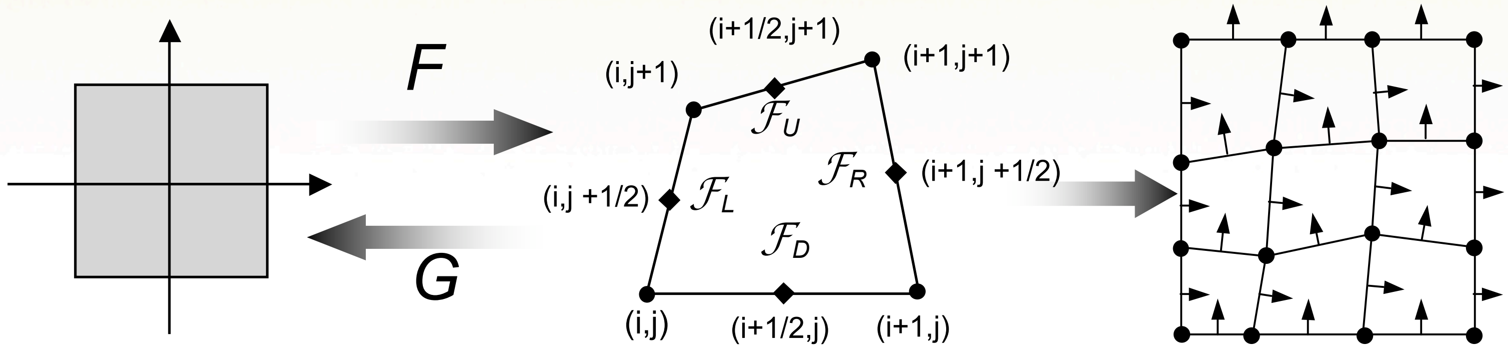
ALE from ART to SCIENCE
June 10th - June 13th
LANL

Supported in part by

Summary

- Edge and face finite element spaces on quads
- Formal statement of the remap problem
- Constrained Interpolation (CI) remapper
  - Recovery
  - Optimization
  - Interpolation
- Numerical results
  - Optimization of potentials
  - Discontinuous B

Edge and Face Elements on Logically Rectangular Grids



Virtual edge functions W¹

$$W_{ij}(\mathbf{x}) = \frac{1}{4}\left(1 - G_1(\mathbf{x})\right)\left(1 - G_2(\mathbf{x})\right)\vec{\mathbf{k}}$$

$$W_{i+1,j}(\mathbf{x}) = \frac{1}{4}\left(1 + G_1(\mathbf{x})\right)\left(1 - G_2(\mathbf{x})\right)\vec{\mathbf{k}}$$

$$W_{i,j+1}(\mathbf{x}) = \frac{1}{4}\left(1 - G_1(\mathbf{x})\right)\left(1 + G_2(\mathbf{x})\right)\vec{\mathbf{k}}$$

$$W_{i+1,j+1}(\mathbf{x}) = \frac{1}{4}\left(1 + G_1(\mathbf{x})\right)\left(1 + G_2(\mathbf{x})\right)\vec{\mathbf{k}}$$

Local shape functions

Face functions W²

$$W_L(\mathbf{x}) = \frac{1}{4}\left(1 - G_1(\mathbf{x})\right)\nabla G_2(\mathbf{x}) \times \vec{\mathbf{k}}$$

$$W_R(\mathbf{x}) = \frac{1}{4}\left(1 + G_1(\mathbf{x})\right)\nabla G_2(\mathbf{x}) \times \vec{\mathbf{k}}$$

$$W_D(\mathbf{x}) = \frac{1}{4}\left(1 - G_2(\mathbf{x})\right)\vec{\mathbf{k}} \times \nabla G_1(\mathbf{x})$$

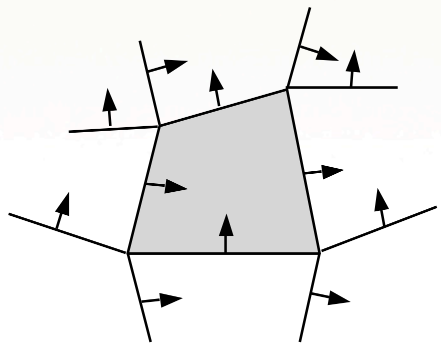
$$W_U(\mathbf{x}) = \frac{1}{4}\left(1 - G_2(\mathbf{x})\right)\vec{\mathbf{k}} \times \nabla G_1(\mathbf{x})$$

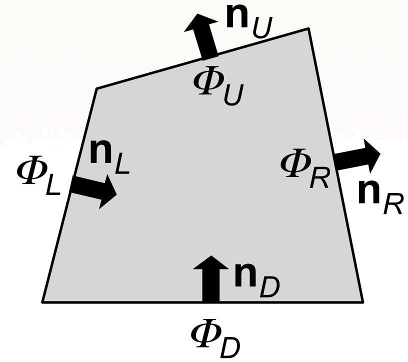
Piola transform is "hidden" in function definition

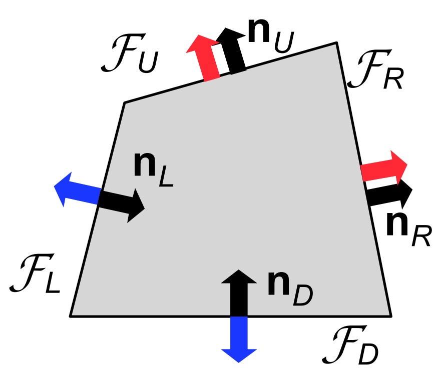
Sandia National Laboratories

Definition of discrete derivative

Duality of ∂ and d







$$\mathbf{B} = \Phi_D W_D + \Phi_R W_R + \Phi_U W_U + \Phi_L W_L$$

$$\Phi_{\mathcal{F}} = \int_{\mathcal{F}} \mathbf{B} \bullet \mathbf{n}\, ds$$

$$\partial \mathcal{K} = -\mathcal{F}_D + \mathcal{F}_R + \mathcal{F}_U - \mathcal{F}_L = \sum \sigma_s \mathcal{F}_s$$

$$\mathbf{dB}V(K) = -\Phi_D + \Phi_R + \Phi_U - \Phi_L = \sum \sigma_s \Phi_s$$

d=div

Coordinate independent definitions of grad, curl, div

Exactness Relation

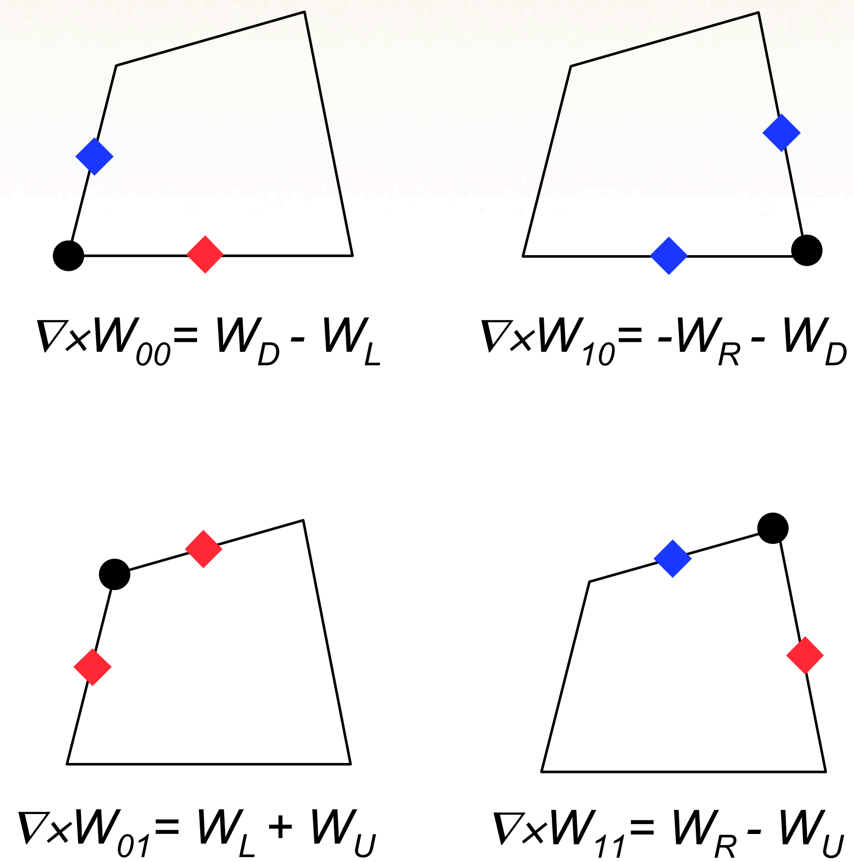
$$\mathbf{A} = \begin{pmatrix} 0 & 0 & \phi \end{pmatrix}^T$$

$$\Downarrow$$

$$\mathbf{A} = \phi_{00} W_{00} + \phi_{10} W_{10} + \phi_{01} W_{01} + \phi_{11} W_{11}$$

$$\nabla \times \mathbf{A} = \phi_{00}\left(W_D - W_L\right) + \phi_{10}\left(-W_L - W_R\right)$$
$$+\phi_{01}\left(W_L + W_U\right) + \phi_{11}\left(W_R - W_U\right)$$

$$\nabla \times \mathbf{A} = \left(\phi_{00} - \phi_{10}\right)W_D + \left(\phi_{01} - \phi_{11}\right)W_U$$
$$+\left(\phi_{11} - \phi_{10}\right)W_R + \left(\phi_{01} - \phi_{00}\right)W_L$$



Discrete Poincare Lemma: $\nabla \cdot \mathbf{B} = 0 \Leftrightarrow \mathbf{B} = \nabla \times \mathbf{A}$

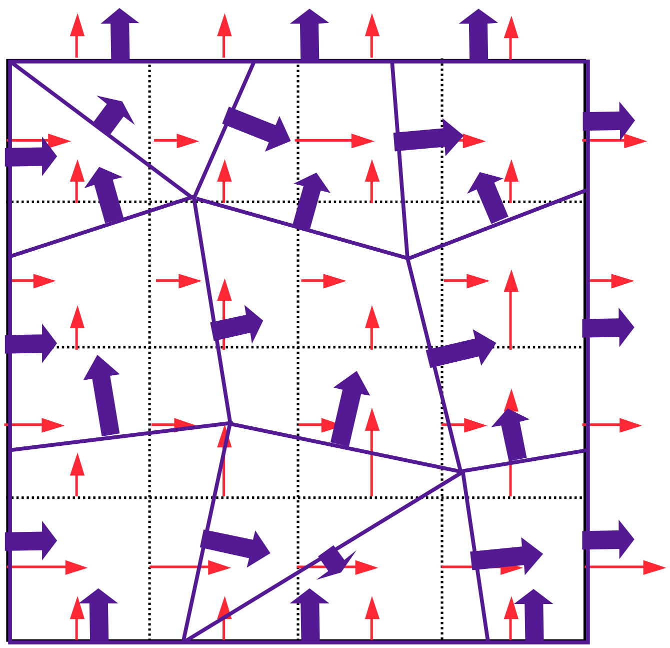
(Formal) Statement of the remap problem:

Given:

- **NEW** and **OLD** meshes with possibly **different** types and numbers of cells
- Discrete solenoidal field **B** on the **OLD** mesh, represented by its fluxes

Find:

- Discrete solenoidal field **B** on the **NEW** mesh such that $\mathbf{B} \approx \mathbf{B}$



A good remapper

- ✓ Is accurate
- ✓ Preserves energy
- ✓ Has good feature retention
- ✓ Does not “ring” at jumps
- ✓ Is efficient (EXPLICIT)!

Taxonomy of Remap Algorithms

Remap via Transport Algorithms: (*Robinson, Bochev, Rambo, SAND2001-2146P*)

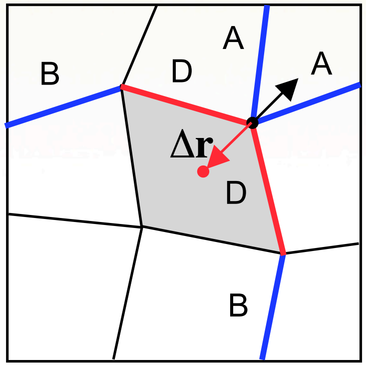
- ❑ Same connectivity required for grids
- ❑ Explicit
- ❑ Upwinding required.
- ❑ Equivalent to Taylor expansion:

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \left(\mathbf{B} \times \mathbf{u}_{REL}\right)$$

$$\mathbf{B}^{n+1} = \mathbf{B}^{n} - \Delta t \nabla \times \left(\mathbf{B}^{n} \times \mathbf{u}_{REL}\right)$$

details

$$\mathbf{B}(\mathbf{r} + \Delta\mathbf{r}) = \mathbf{B}(\mathbf{r}) - \nabla \times \left(\Delta\mathbf{r} \times \mathbf{B}\right) + O\left(\left|\Delta\mathbf{r}\right|^{2}\right)$$



Remap via Constrained Optimization: (*Carey et al. IJNME 50, **2001**, Girault **2003***)

- ❑ Arbitrary grids
- ❑ Implicit: solution of a saddle-point “Darcy” problem **required**:

$$\min_{\mathbf{B}^{new} \in W^{new}(\Omega)} \int_{\Omega} \left|\mathbf{B}^{old} - \mathbf{B}^{new}\right|^{2} dx \quad \text{subject to } \nabla \cdot \mathbf{B}^{new} = 0 \quad \Rightarrow \quad \begin{pmatrix} A & C \\ C^{T} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{B}^{new} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{B}^{old} \\ 0 \end{pmatrix}$$

Remap via Grid Structure:(*Balsara JCP 174, **2001**, Roe & Toth JCP 180, **2002***)

- ❑ Limited to Cartesian grids
- ❑ Additional restrictions required: factor-of-2 resolution change, etc.

Sandia National Laboratories

Constrained Interpolation (CI) Algorithm

Principal idea: exploit **existence** of discrete potentials in exact sequences
⇒ **divergence-free constraint automatically satisfied**

Key component: an **explicit** potential recovery algorithm

RECOVERY

$$W^1 \overset{C}{\mapsto} W^2 \overset{D}{\mapsto} W^3$$

$$D\mathbf{B} = \mathbf{0} \Leftrightarrow \mathbf{B} = C\mathbf{A}$$

$$\begin{array}{ccc}
\nabla \cdot \mathbf{B}^{old} = 0 & & \nabla \cdot \mathbf{B}^{new} = 0 \\
\downarrow & & \uparrow \\
\mathbf{B}^{old} = \nabla \times \mathbf{A}^{old} & & \mathbf{B}^{new} = \nabla \times \mathbf{A}^{new} \\
\downarrow & & \uparrow \\
\mathbf{A}^{old} & \longrightarrow & \mathbf{A}^{new}
\end{array}$$

REMAP

POSTPROCESSING

$$\mathcal{R}\left(\mathbf{A}^{old}\right) \mapsto$$

Patch recovery
PP operators
FV reconstruction

"reconstruction"

OPTIMIZATION

$$\mathbf{A}\ (\lambda) = \lambda \mathbf{A}^{old} + (1-\lambda)\mathcal{R}\left(\mathbf{A}^{old}\right)$$

$$\lambda_{opt} = \arg\min J\left(\mathbf{A}^{old}, \mathbf{A}(\lambda)\right)$$

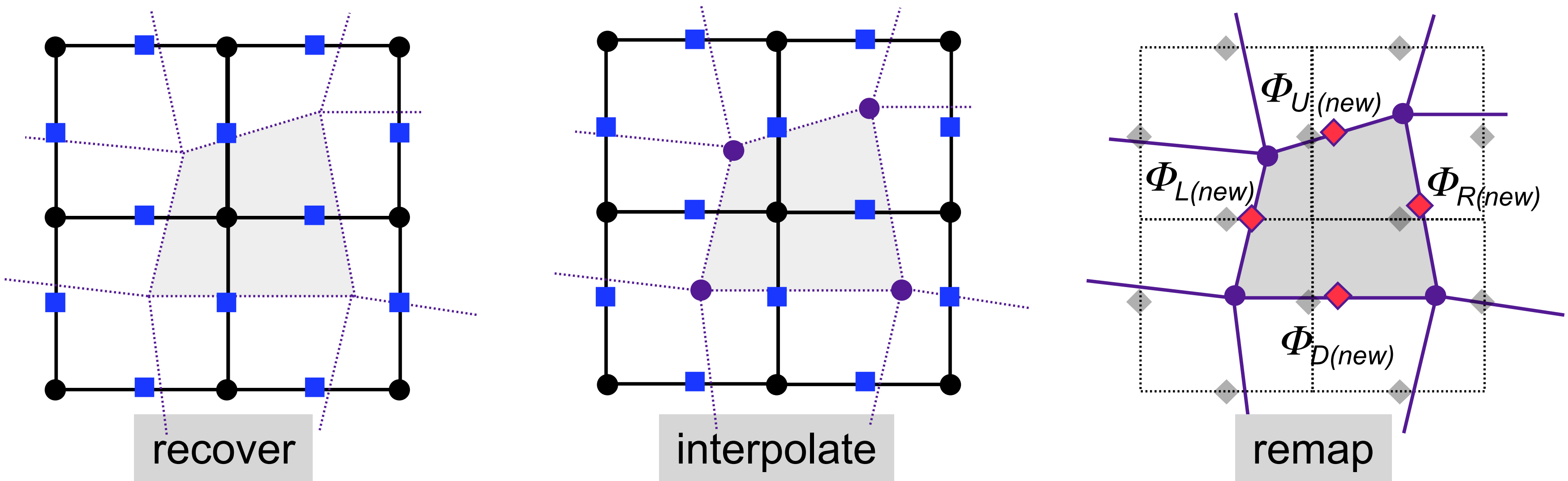
"limiting"

INTERPOLATION

$$\mathbf{A}^{new} = \mathcal{I}\mathbf{A}\left(\lambda_{opt}\right)$$

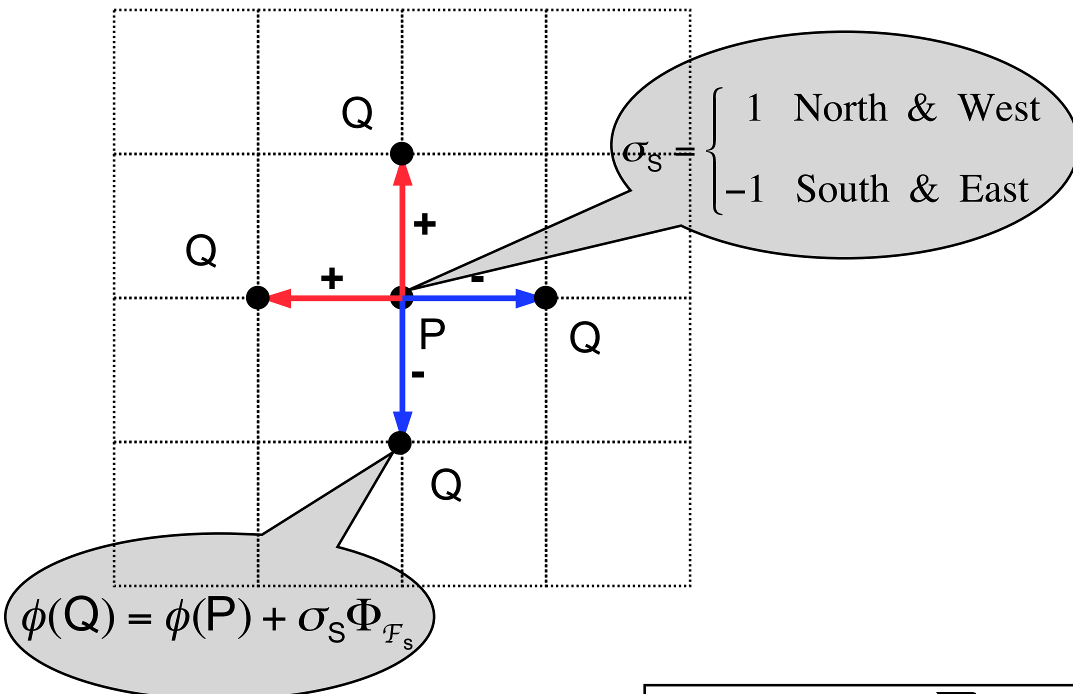
"upwinding"

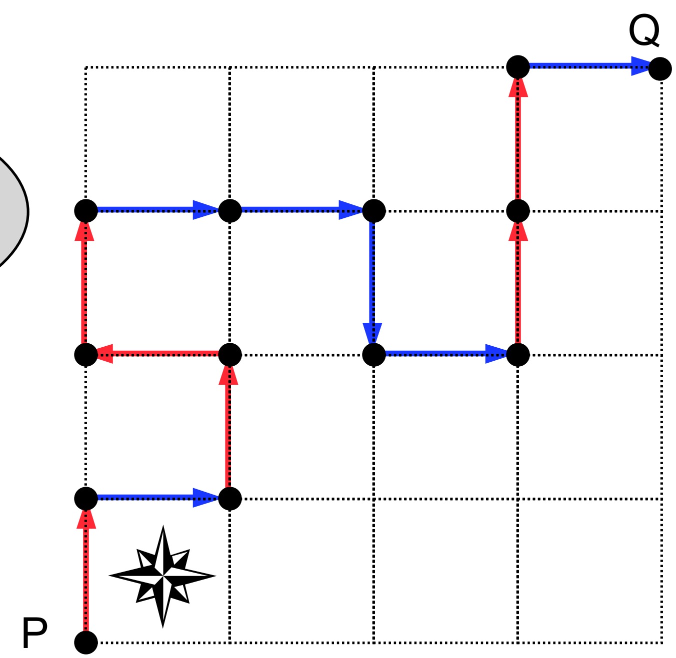
CI Remap in a Nutshell



Explicit Potential Recovery

$$\left.\begin{aligned}\nabla\times\mathbf{A}_{\mathbf{B}} &= (\phi_{00}-\phi_{10})W_D + (\phi_{01}-\phi_{11})W_U \\ &\quad + (\phi_{11}-\phi_{10})W_R + (\phi_{01}-\phi_{00})W_L \\ \mathbf{B} &= \Phi_D W_D + \Phi_U W_U + \Phi_R W_R + \Phi_L W_L\end{aligned}\right\} \Rightarrow \nabla\times\mathbf{A}_{\mathbf{B}} = \mathbf{B} \Leftrightarrow \begin{cases}\Phi_D = (\phi_{00}-\phi_{10}) & \Phi_L = (\phi_{01}-\phi_{00}) \\ \Phi_U = (\phi_{01}-\phi_{11}) & \Phi_R = (\phi_{11}-\phi_{10})\end{cases}$$

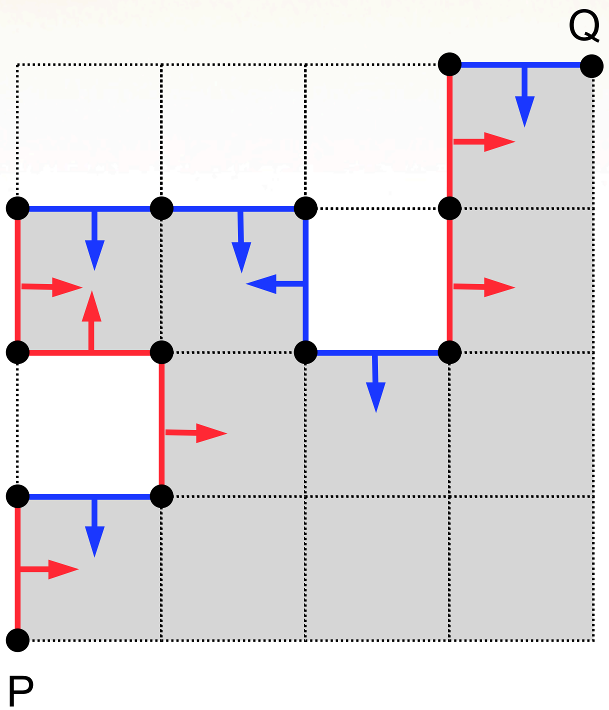




$$\boxed{\phi(Q) = \phi(P) + \sum_{\mathcal{F}_S} \sigma_S \Phi_{\mathcal{F}_S}}$$

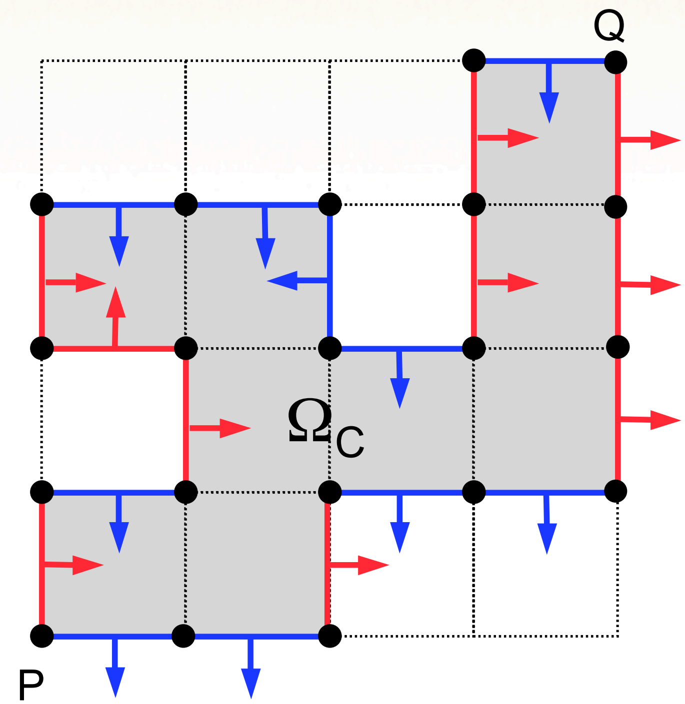
Sandia National Laboratories

Path Independence



$$C_1 = \sum_{\mathcal{F}_{S_1}} \sigma_{S_1} \mathcal{F}_{S_1}$$

$$C_2 = \sum_{\mathcal{F}_{S_2}} \sigma_{S_2} \mathcal{F}_{S_2}$$



$$\left.\begin{array}{l} \partial\Omega_C = C_2 - C_1 \\ \nabla \cdot \mathbf{B} = 0 \end{array}\right\} \Rightarrow 0 = \int_{\Omega_C} \nabla \cdot \mathbf{B}\, dx = \int_{\partial\Omega_C} \mathbf{B} \cdot \mathbf{n}\, dx$$

$$\Rightarrow 0 = \int_{C_2} \mathbf{B} \cdot \mathbf{n}\, dx - \int_{C_1} \mathbf{B} \cdot \mathbf{n}\, dx \quad \Rightarrow \quad \boxed{\int_{C_2} \mathbf{B} \cdot \mathbf{n}\, dx = \int_{C_1} \mathbf{B} \cdot \mathbf{n}\, dx = \varphi(Q)}$$

Recovery on Logically Rectangular Grids

1. Choose a spanning tree

$$C_{sp} = \sum_{\mathcal{F}_{S_{sp}}} \sigma_{S_{sp}} \mathcal{F}_{S_{sp}}$$

2. Initialize the root

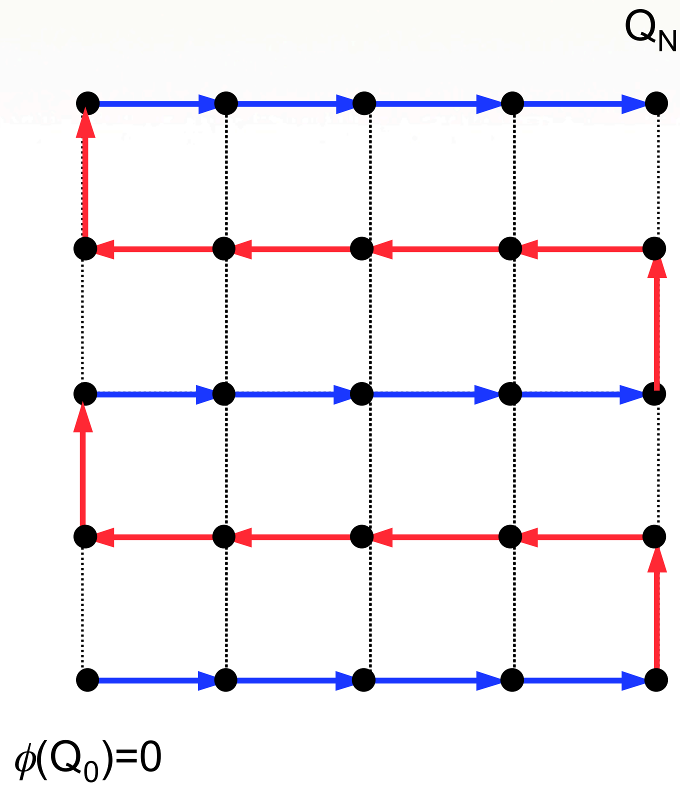
$$\phi(Q_0) = 0$$

3. Traverse the branches

for k=1:N

$$\phi(Q_k) = \phi(Q_{k-1}) + \sigma_k \Phi_{\mathcal{F}_k}$$

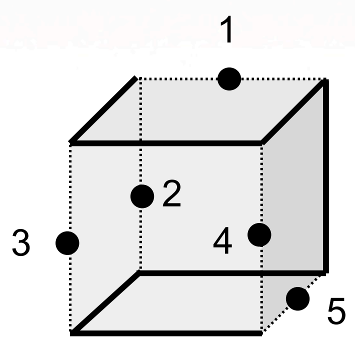
end



3D recovery on logically Cartesian grids similar

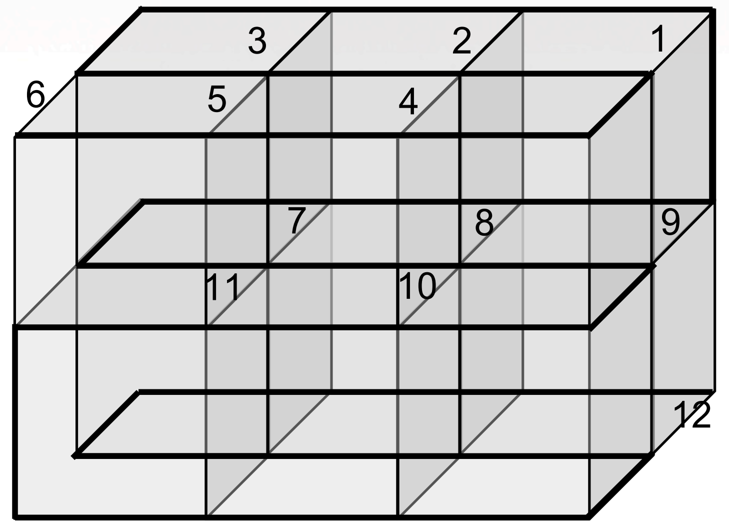
Base potential recovery in 3D

On one cell:



6 flux DOF - 1 constraint = 5
12 edges - 7SP edges = 5

On a logically rectangular mesh

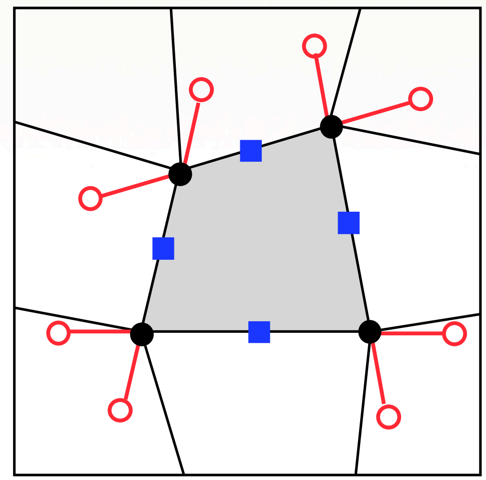


1. Choose a spanning tree and mark edges on co-spanning tree as **free**
2. Order faces with respect to the number of free edges
3. Recover $\mathbf{A}_B$ on all faces with 1 free edge; update number of free edges
4. If no faces with free edges left **then** stop, **else** proceed to step 3

Post-processing and Optimization

Post-processing

- A_B → **base** Q1 (bilinear interpolant)
- A_E → **extended** 8 node serendipity



Optimization

$$\mathbf{A}(\lambda) = \lambda \mathbf{A}^{old} + (1-\lambda)\mathcal{R}(\mathbf{A}^{old})$$

$$\lambda_{opt} = \arg\min J(\mathbf{A}^{old}, \mathbf{A}(\lambda))$$

A. Feedback

$$\lambda_{new} = \begin{cases} \max(0, \lambda_{old} + \varepsilon) & \varepsilon < 0 \\ \min(1, \lambda_{old} + \varepsilon) & \varepsilon > 0 \end{cases}; \qquad \varepsilon = \theta\left(1 - \frac{\|\mathbf{B}_{old}\|_\Omega}{\|\mathbf{B}_{new}\|_\Omega}\right)$$

B. l-parameter control

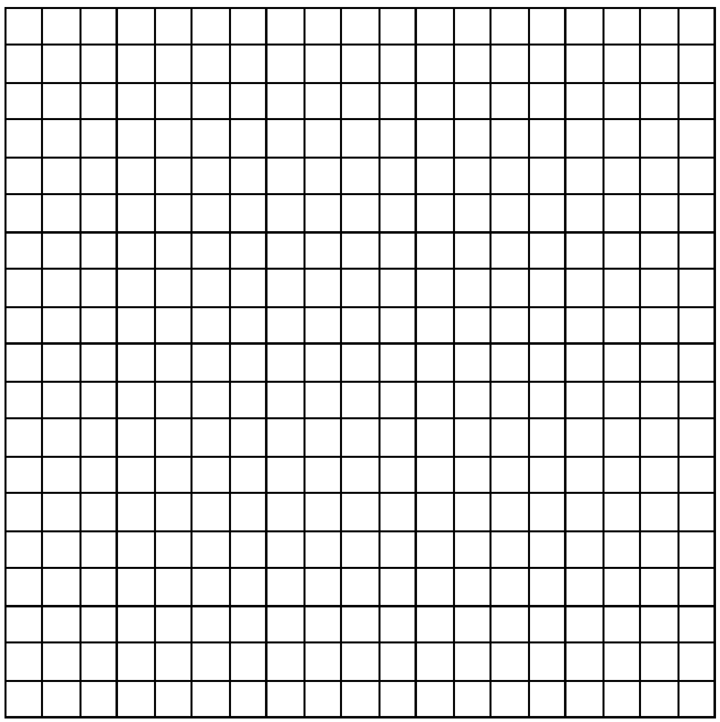
$$\lambda(\Omega) \rightarrow \arg\min\left(\|\mathbf{B}_{old}\|_\Omega^2 - \|\mathbf{B}_{new}(\lambda(\Omega))\|_\Omega^2\right)^2$$

C. Cell-by-cell control

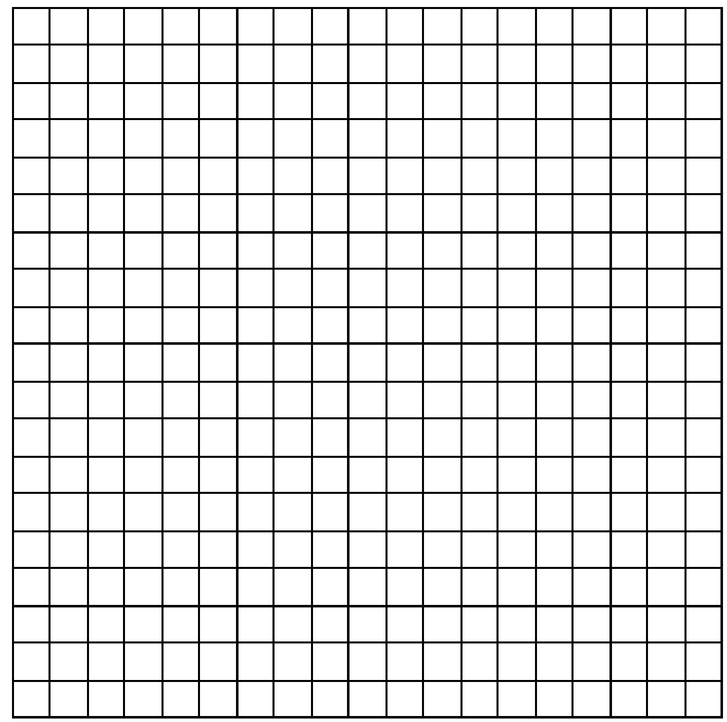
Requires conservative remap of magnetic energy (see *Shashkov and Margolin, LANL LAUR-2002*)

Numerical Results: Cyclic Remap

Wave Mesh



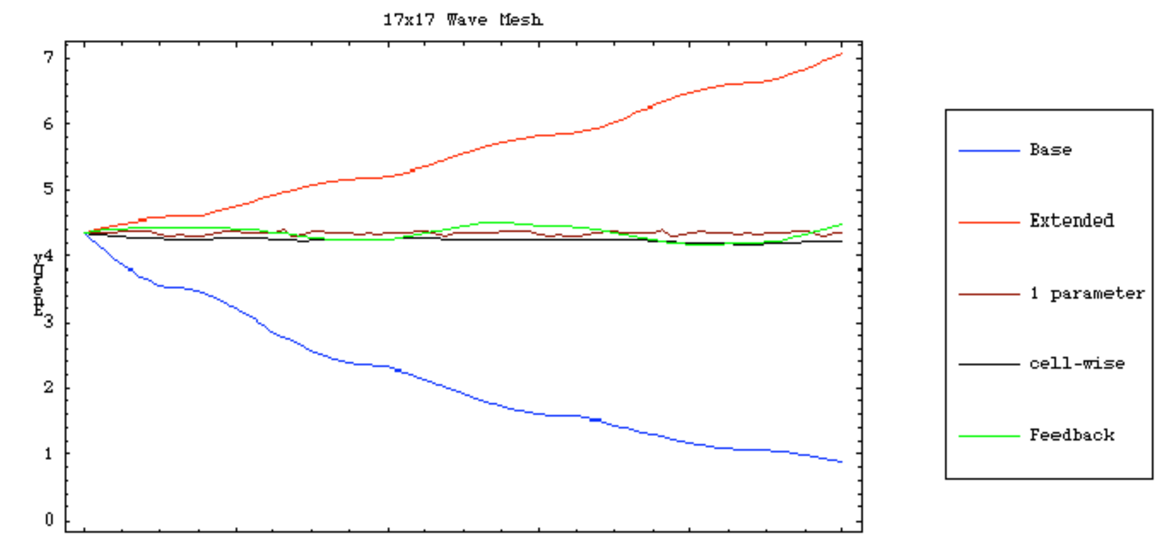
Random Mesh

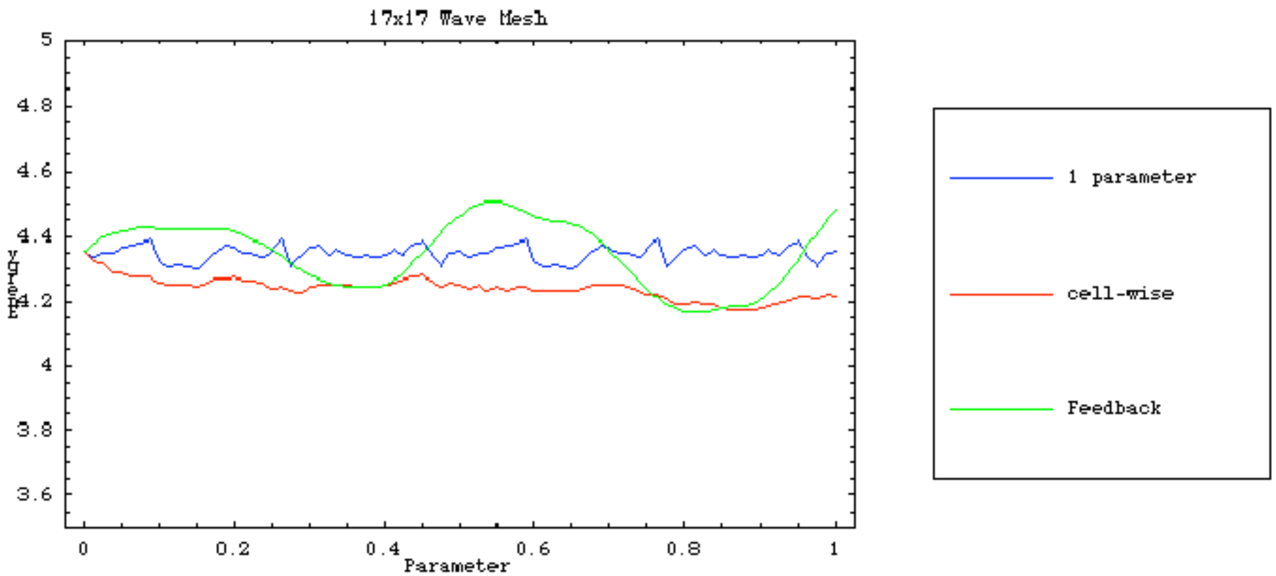


100 cycles

Shashkov and Margolin, LANL LAUR-2002

Potential Optimization: Wave Mesh



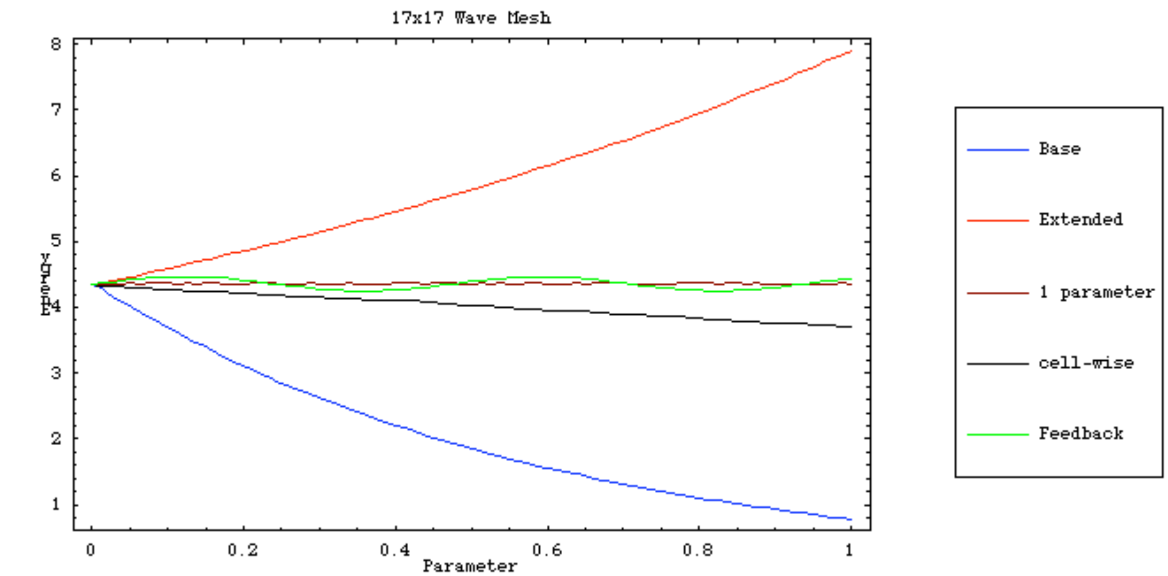


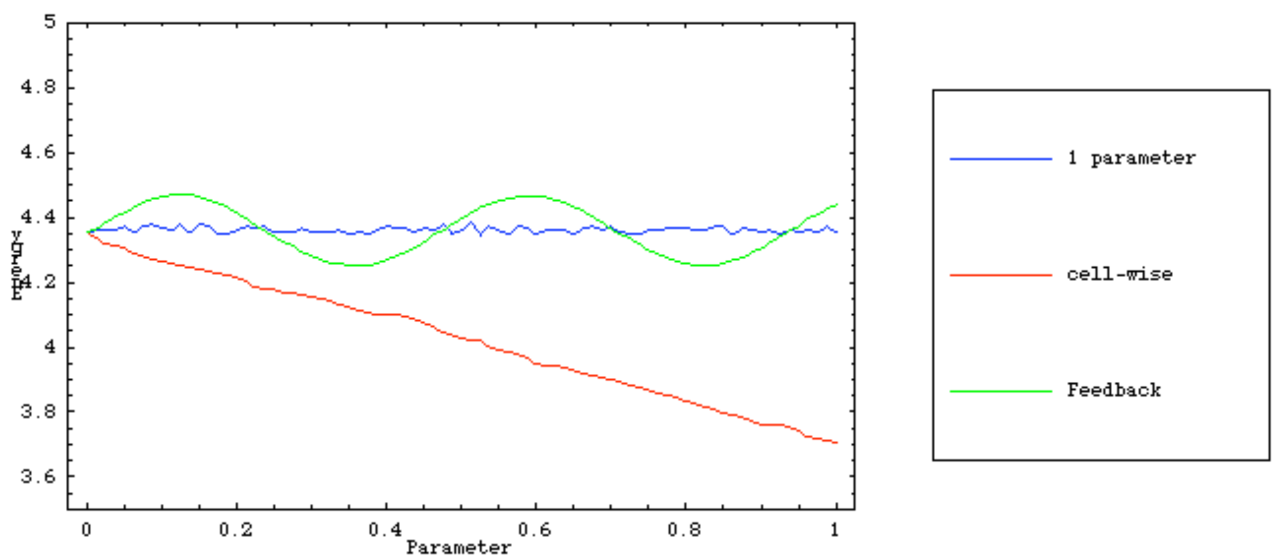
17x17 Wave Mesh:
80 cycles

$$\mathbf{B} = \nabla \times \mathbf{A};$$

$$\mathbf{A} = \sin(2\pi x)\sin(2\pi y)$$

Potential Optimization: Random Mesh





17x17 Random Mesh: 80 cycles

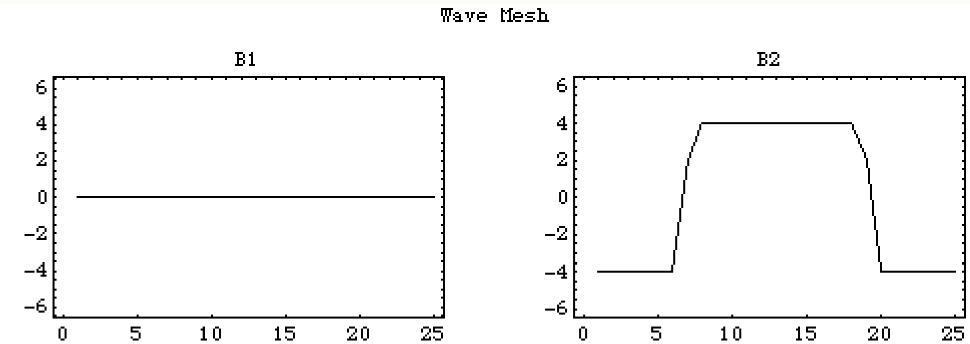
$$\mathbf{B} = \nabla \times \mathbf{A};$$

$$\mathbf{A} = \sin(2\pi x)\sin(2\pi y)$$

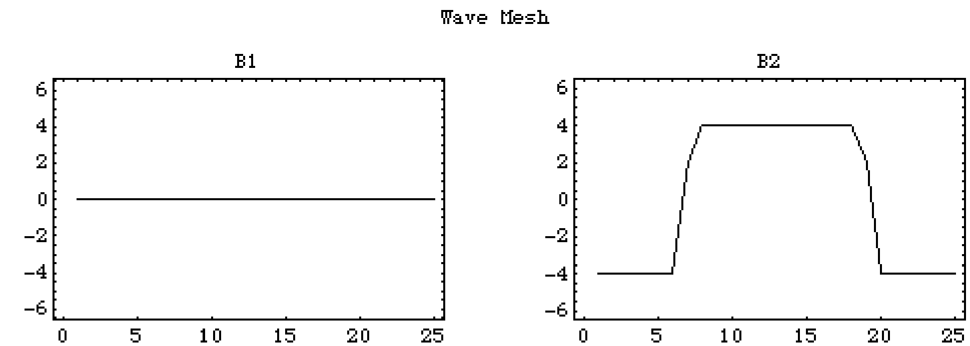
Potential Optimization:
Discontinuous Field

30x30 Wave Mesh:
100 cycles

1-parameter control

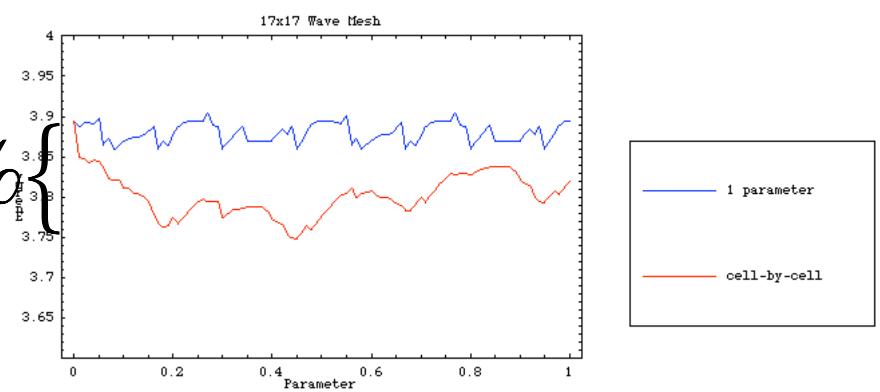


Cell-by-cell control



Energy

$< 4\%$

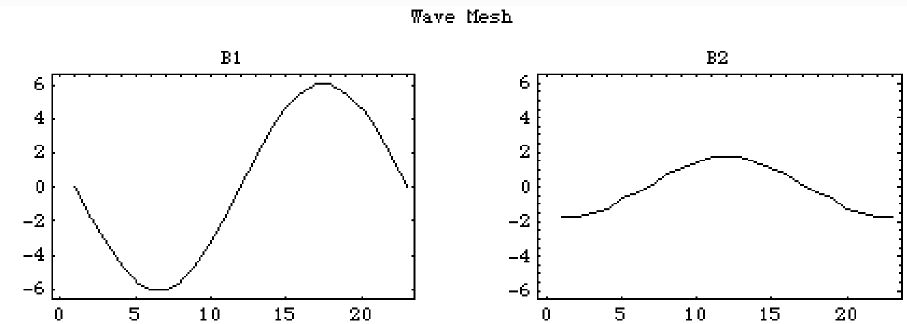


CI Base vs. CT Donor:
Smooth Field

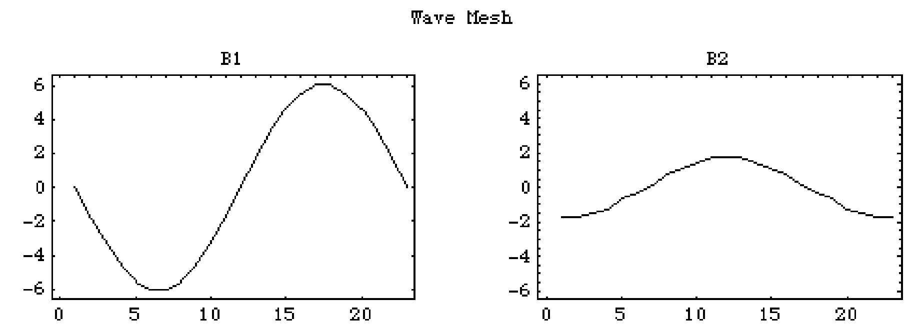
**30x30 Wave Mesh:
100 cycles**

$$\mathbf{B} = \nabla \times \mathbf{A}; \quad \mathbf{A} = \sin(2\pi x)\sin(2\pi y)$$

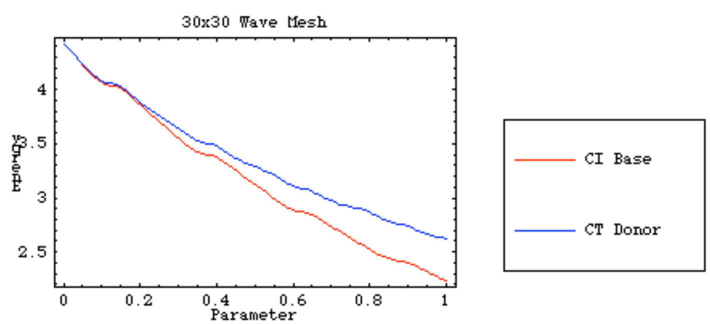
CI Base:



CT Donor:



Energy:

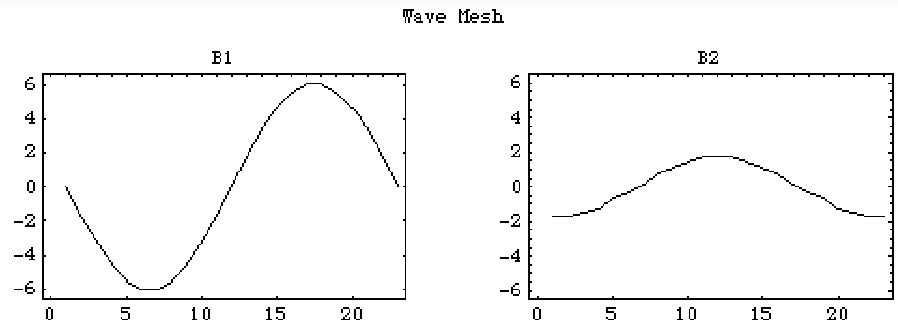


CI Base vs. CT Donor:
Smooth Field

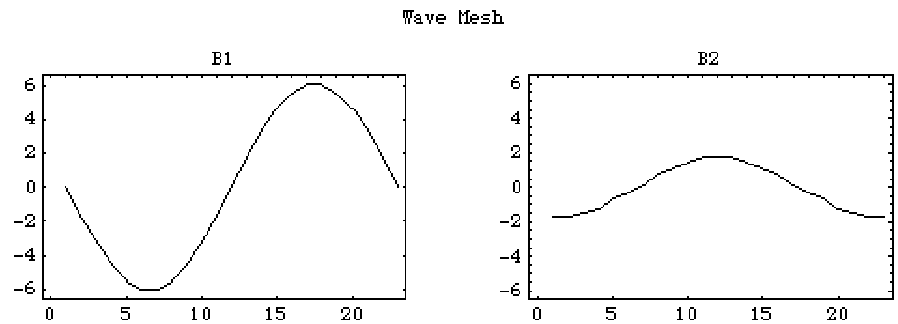
30x30 Random Mesh:
100 cycles

$$\mathbf{B} = \nabla \times \mathbf{A}; \quad \mathbf{A} = \sin(2\pi x)\sin(2\pi y)$$

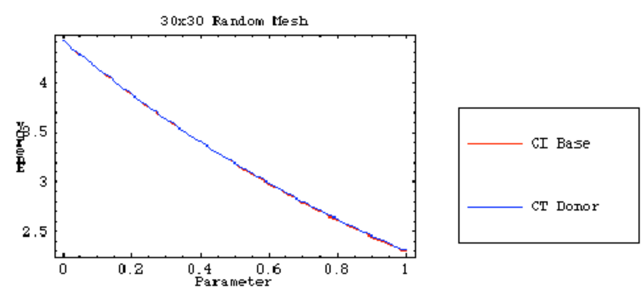
CI Base:



CT Donor:



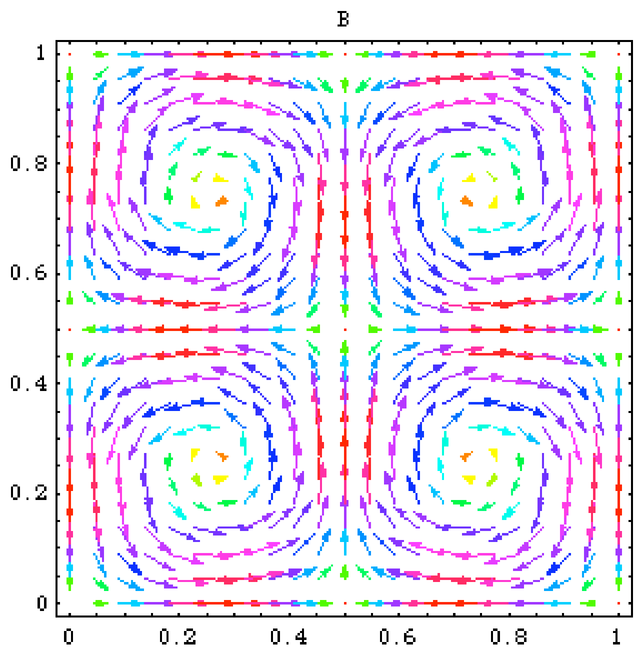
Energy:



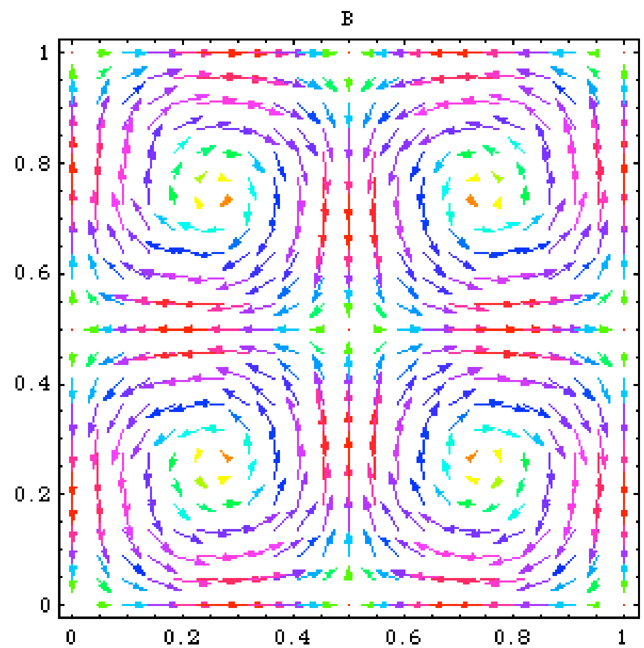
CI Base vs. CT Donor:
Smooth Field

Feature retention $\mathbf{B} = \nabla \times \mathbf{A}; \quad \mathbf{A} = \sin(2\pi x)\sin(2\pi y)$

CI Base



Constrained Transport Donor



30x30 Wave mesh:
100 cycles

Conclusions

Constrained Interpolation Remap:

- Algorithm applicable to arbitrary pairs of grids:

  1. Old and new grid can have different topologies (refinement)
  2. New grid is not necessarily a small perturbation of the old one (no CFL condition required for stability)
  3. High accuracy retained for unstructured grids
  4. Explicit!

- Modular: can be extended to different discretizations (FE, FV, FD)

Constrained Transport (CT) Remapper

Based on CT scheme of Evans, Hawley, The Astrophysical Journal 332, 1988

Advection equation

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \left(\mathbf{B} \times \mathbf{u}_{REL}\right)$$

$$\mathbf{E} = \mathbf{B} \times \mathbf{u}_{REL}$$

Discretization

$$\mathbf{B}^{old} = \sum_{\mathcal{F}} \Phi_{\mathcal{F}}^{old} W_{\mathcal{F}}^{old}$$

$$\mathbf{E}^{old} = \sum_{\mathcal{N}} C_{\mathcal{N}}^{old} W_{\mathcal{N}}^{old}$$

CT Update

$$\mathbf{B}^{new} = \mathbf{B}^{old} - \Delta t \nabla \times \mathbf{E}^{old}$$

Flux update

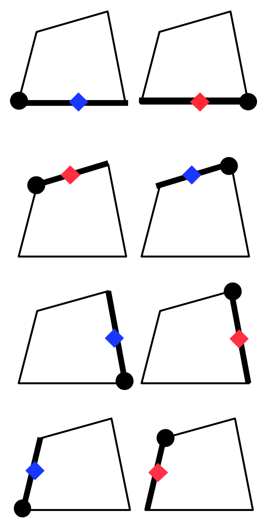
Exactness

$$\Phi_D^{new} = \Phi_D^{old} + \Delta t\left(C_{00}^{old} - C_{10}^{old}\right)$$

$$\Phi_U^{new} = \Phi_U^{old} + \Delta t\left(C_{01}^{old} - C_{11}^{old}\right)$$

$$\Phi_R^{new} = \Phi_R^{old} + \Delta t\left(C_{11}^{old} - C_{10}^{old}\right)$$

$$\Phi_L^{new} = \Phi_L^{old} + \Delta t\left(C_{01}^{old} - C_{00}^{old}\right)$$



$$\nabla \times \mathbf{E}^{old} = \left(C_{00}^{old} - C_{10}^{old}\right)W_D^{old} + \left(C_{01}^{old} - C_{11}^{old}\right)W_U^{old} + \left(C_{11}^{old} - C_{10}^{old}\right)W_R^{old} + \left(C_{01}^{old} - C_{00}^{old}\right)W_L^{old}$$

$$\nabla \cdot \mathbf{B}^{old} = 0 \quad \Rightarrow \quad \nabla \cdot \mathbf{B}^{new} = 0$$

Computation of EMF in CT

E is required at virtual edges **P** → $C^{old}_{\mathcal{N}} = \left(\mathbf{u}_2\mathbf{B}^{old}_1 - \mathbf{u}_1\mathbf{B}^{old}_2\right)(\mathrm{P})$

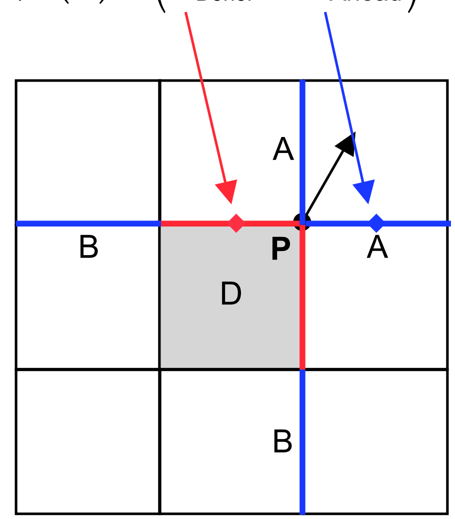
B is discontinuous at **P**

⇒ **B** must be (upwind) averaged at **P** → $\mathbf{B}^{old}_i(\mathrm{P}) = \left(\mathbf{B}^{old}_{\text{Donor}} + \hat{\mathbf{B}}^{old}_{\text{Ahead}}\right)/2$

Upwind interpolation at **P**:

$$\hat{\mathbf{B}}^{old}_{\text{Ahead}} = \mathbf{B}^{old}_{\text{Donor}} + S\left(\frac{|\mathcal{F}_{\text{Donor}}| + |\mathcal{F}_{\text{Ahead}}|}{2} - \Delta t\mathbf{u}_i\right)$$

$$\mathbf{B}^{old}_i(\mathrm{P}) = \mathbf{B}^{old}_{i,\text{Donor}}(\mathrm{P}) + \frac{S}{2}\left(\pm\frac{|\mathcal{F}_{\text{Donor}}| + |\mathcal{F}_{\text{Ahead}}|}{2} - \Delta t\mathbf{u}_i\right)$$



$$S = \begin{cases} \min\left(|S_1|,|S_2|\right)\operatorname{sgn}(S_i) & \text{monotone} \\ 2S_1S_2/\left(S_1 + S_2\right) & \text{harmonic} \\ \min\left(|S_3|,|S_4|,\left(|S_1| + |S_2|\right)/2\right)\operatorname{sgn}(S_i) & \text{Van Leer} \\ 0 \quad \text{if } S_1S_2 < 0 \quad \text{or} & \text{donor} \end{cases}$$

$S_1 = (\mathbf{B}_{\mathrm{A}} - \mathbf{B}_{\mathrm{D}})/\left(\left(|\mathcal{F}_{\mathrm{D}}| + |\mathcal{F}_{\mathrm{A}}|\right)/2\right)$; $S_3 = (\mathbf{B}_{\mathrm{A}} - \mathbf{B}_{\mathrm{D}})/\left(|\mathcal{F}_{\mathrm{D}}|/2\right)$

$S_1 = (\mathbf{B}_{\mathrm{D}} - \mathbf{B}_{\mathrm{B}})/\left(\left(|\mathcal{F}_{\mathrm{D}}| + |\mathcal{F}_{\mathrm{B}}|\right)/2\right)$; $S_4 = (\mathbf{B}_{\mathrm{D}} - \mathbf{B}_{\mathrm{B}})/\left(|\mathcal{F}_{\mathrm{D}}|/2\right)$

Sandia National Laboratories

Extension to unstructured meshes

A. Robinson, P. Bochev, P. Rambo. `http://infoserve.sandia.gov/sand_doc/2001/012146p.pdf.`

CT Remap and Taylor expansion

$$\mathbf{B}_1(\mathbf{r}+\Delta\mathbf{r}) = \mathbf{B}_1(\mathbf{r}) + \frac{\partial \mathbf{B}_1(\mathbf{r})}{\partial x}\Delta x + \frac{\partial \mathbf{B}_1(\mathbf{r})}{\partial y}\Delta y + O\left(|\Delta\mathbf{r}|^2\right)$$

$$\mathbf{B}_2(\mathbf{r}+\Delta\mathbf{r}) = \mathbf{B}_2(\mathbf{r}) + \frac{\partial \mathbf{B}_2(\mathbf{r})}{\partial x}\Delta x + \frac{\partial \mathbf{B}_2(\mathbf{r})}{\partial y}\Delta y + O\left(|\Delta\mathbf{r}|^2\right)$$

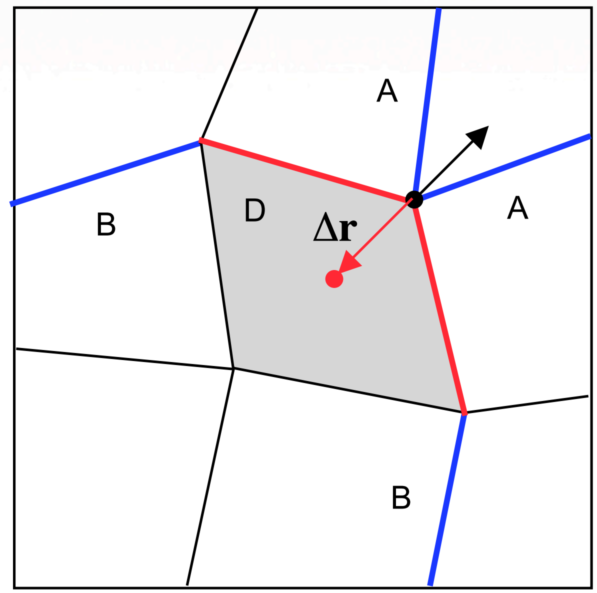
$$\nabla\cdot\mathbf{B} = \frac{\partial \mathbf{B}_1}{\partial x} + \frac{\partial \mathbf{B}_2}{\partial y} = 0$$

$$\mathbf{B}_1(\mathbf{r}+\Delta\mathbf{r}) = \mathbf{B}_1(\mathbf{r}) + \left(\frac{\partial \mathbf{B}_1(\mathbf{r})}{\partial x}\Delta x + \frac{\partial \mathbf{B}_2(\mathbf{r})}{\partial y}\Delta x\right) + \left(\frac{\partial \mathbf{B}_1(\mathbf{r})}{\partial y}\Delta y - \frac{\partial \mathbf{B}_2(\mathbf{r})}{\partial y}\Delta x\right) + O\left(|\Delta\mathbf{r}|^2\right)$$

$$\mathbf{B}_2(\mathbf{r}+\Delta\mathbf{r}) = \mathbf{B}_2(\mathbf{r}) + \left(\frac{\partial \mathbf{B}_2(\mathbf{r})}{\partial x}\Delta x - \frac{\partial \mathbf{B}_1(\mathbf{r})}{\partial x}\Delta y\right) + \left(\frac{\partial \mathbf{B}_2(\mathbf{r})}{\partial y}\Delta y + \frac{\partial \mathbf{B}_1(\mathbf{r})}{\partial x}\Delta y\right) + O\left(|\Delta\mathbf{r}|^2\right)$$

$$\mathbf{B}(\mathbf{r}+\Delta\mathbf{r}) = \mathbf{B}(\mathbf{r}) + \begin{pmatrix} \frac{\partial}{\partial y}(\mathbf{B}_1(\mathbf{r})\Delta y - \mathbf{B}_2(\mathbf{r})\Delta x) \\ -\frac{\partial}{\partial x}(\mathbf{B}_1(\mathbf{r})\Delta y - \mathbf{B}_2(\mathbf{r})\Delta x) \end{pmatrix} + O\left(|\Delta\mathbf{r}|^2\right) = \mathbf{B}(\mathbf{r}) - \nabla\times(\Delta\mathbf{r}\times\mathbf{B})$$

$$\mathbf{B}(\mathbf{r}+\Delta\mathbf{r}) = \mathbf{B}(\mathbf{r}) - \nabla\times\left(\Delta\mathbf{r}\times\mathbf{B}\right) + O\left(|\Delta\mathbf{r}|^2\right)$$

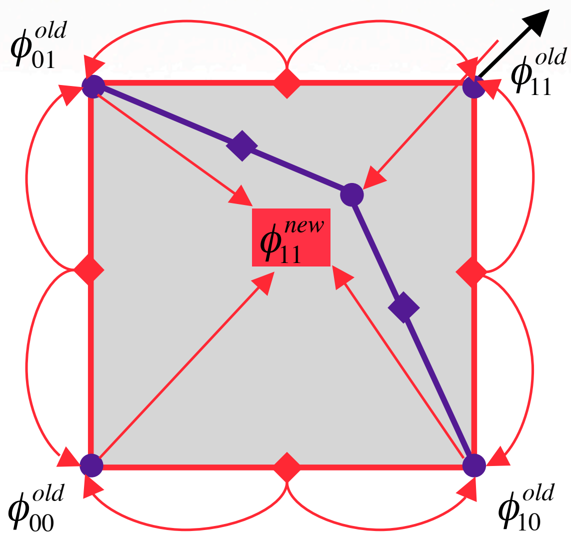


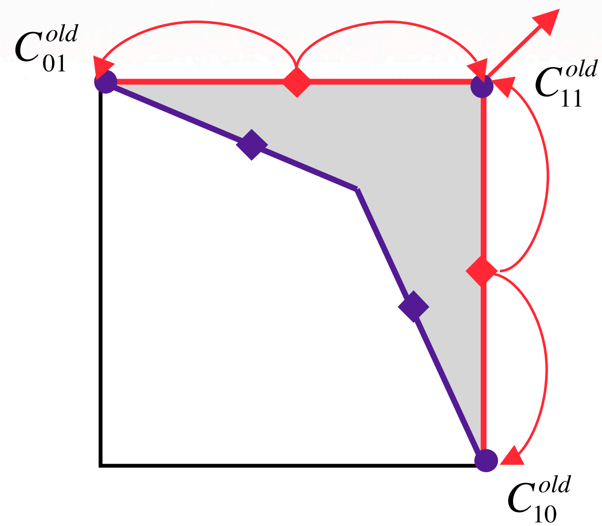
Reconstruction must be exact for the 1st derivatives to get 2nd order accuracy. Not easy on unstructured meshes!

return

Sandia National Laboratories

Base CI Uses Information Better Than Donor Cell CT





$$\Phi_U^{new} = \left(\phi_{01}^{new} - \phi_{11}^{new}\right)$$
$$\Phi_R^{new} = \left(\phi_{11}^{new} - \phi_{10}^{new}\right)$$

$$\Phi_U^{new} = \Phi_U^{old} + \Delta t\left(C_{01}^{old} - C_{11}^{old}\right)$$
$$\Phi_R^{new} = \Phi_R^{old} + \Delta t\left(C_{11}^{old} - C_{10}^{old}\right)$$

ϕ_{ij}^{new} depend on **all 4 old** flux values

C_{ij}^{old} depend **only on 2 old** flux values