SAND2008-2180C

Recent Algorithmic (and Practical) Developments in ML

Chris Siefert

Ray Tuminaro, Jonathan Hu, Pavel Bochev, Erik Boman and Karen Devine

Sandia National Laboratories

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

Outline

- Introduction to ML.
- Solving Maxwell's Equations w/ RefMaxwell.
- Repartitioning w/ Zoltan and Hypergraphs.
- Conclusions & Future Work.

Trilinos Summary

Core

- Teuchos
- Zoltan
- Thyra
- RTOp
- Epetra
- EpetraExt
- ForTrilinos
- Isorropia

Discretizations

- Intrepid
- phdMesh
- Rythmos

Solvers

- AztecOO
- ML
- NOX
- Meros
- LOCA
- Ifpack
- Amesos
- Anasazi
- Belos
- Moocho

Methods

- Moertel
- Sacado

Trilinos Summary

Core

- Teuchos
- Zoltan
- Thyra
- RTOp
- Epetra
- EpetraExt
- ForTrilinos
- Isorropia

Discretizations

- Intrepid
- phdMesh
- Rythmos

Solvers

- AztecOO
- ML
- NOX
- Meros
- LOCA
- Ifpack
- Amesos
- Anasazi
- Belos
- Moocho

Methods

- Moertel
- Sacado

Sandia National Laboratories

ML Features (1)

- ML provides scalable multilevel/multigrid preconditioners.
- Method types
  - Smoothed Aggregation (SA) - symmetric or nearly symmetric problems.
  - Non-symmetric SA - non-symmetric problems.
  - MatrixFree - matrix-free SA.
  - DD / DD-ML - domain decomposition.
  - Maxwell - Maxwell's equations.
  - RefMaxwell - new method for Maxwell's equations.

ML Features (2)

- Simple Trilinos interface.
- Teuchos::ParameterList driven options.
  Has sensible defaults (override what you don't like).
- Parameter validation for accuracy.
- MATLAB interface for some features (MLMEX).

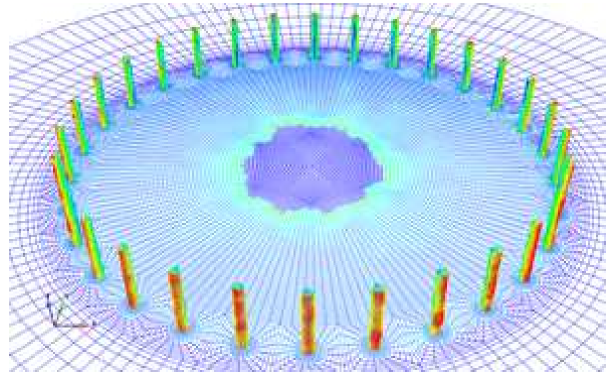
Using ML

```
// Start with a problem & build solver
Epetra_LinearProblem Problem(A, &LHS, &RHS);
AztecOO solver(Problem);

// Override any defaults
Teuchos::ParameterList List;
List.set("smoother: sweeps",2);

// Build the preconditioner
MultiLevelPreconditioner Prec(A,List);
solver.SetPrecOperator(Prec);

solver.Iterate(100,1e-12);// Solve
```

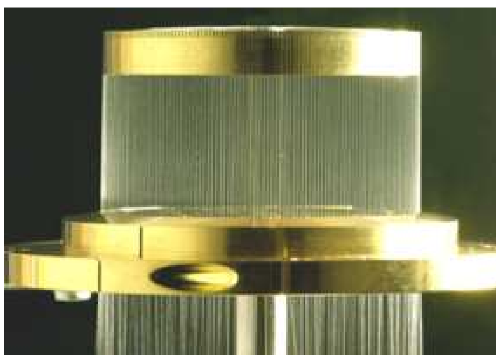
Outline

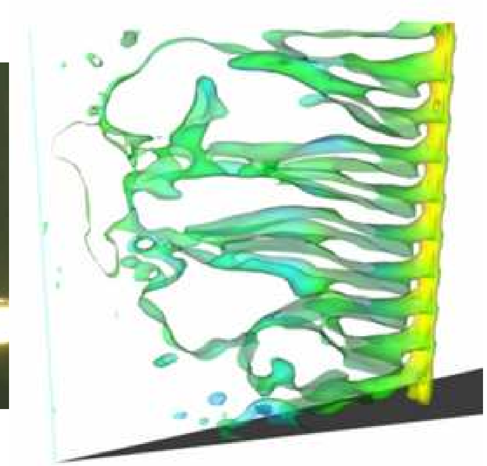
- Introduction to ML.
- Solving Maxwell's Equations w/ RefMaxwell.
- Repartitioning w/ Zoltan and Hypergraphs.
- Conclusions & Future Work.

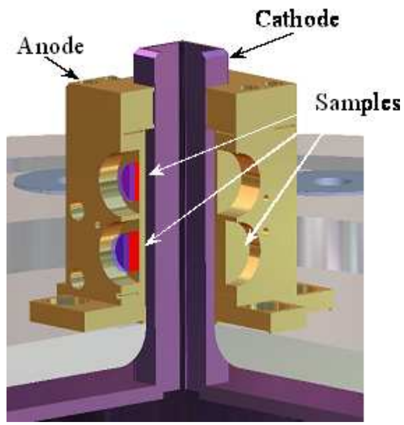
Target Applications



- Electromagnetic phenomena modeled by Maxwell's equations occur in many Sandia applications.
- HEDP: Wire arrays and liners for Z machine simulations.
- Magnetic Launch: Coil & rail guns (ONR).
- Code: ALEGRA & Trilinos/ML (SNL).



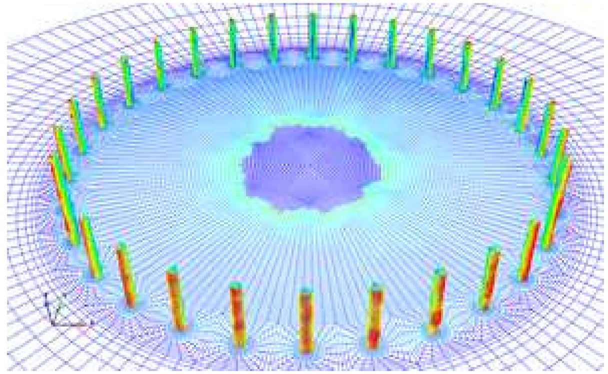




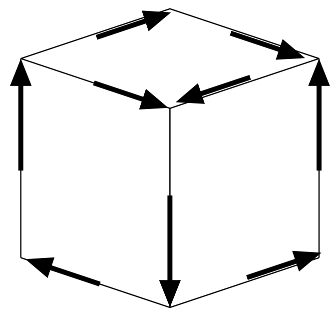
Sandia National Laboratories

Maxwell's Equations

$$\nabla \times \frac{1}{\mu}\nabla \times \mathbf{E} + \sigma\mathbf{E} = 0 \quad \text{in } \Omega$$
$$\mathbf{n} \times \mathbf{E} = 0 \quad \text{on } \Gamma$$
$$\mathbf{n} \times \frac{1}{\mu}\nabla \times \mathbf{E} = 0 \quad \text{on } \Gamma^*$$

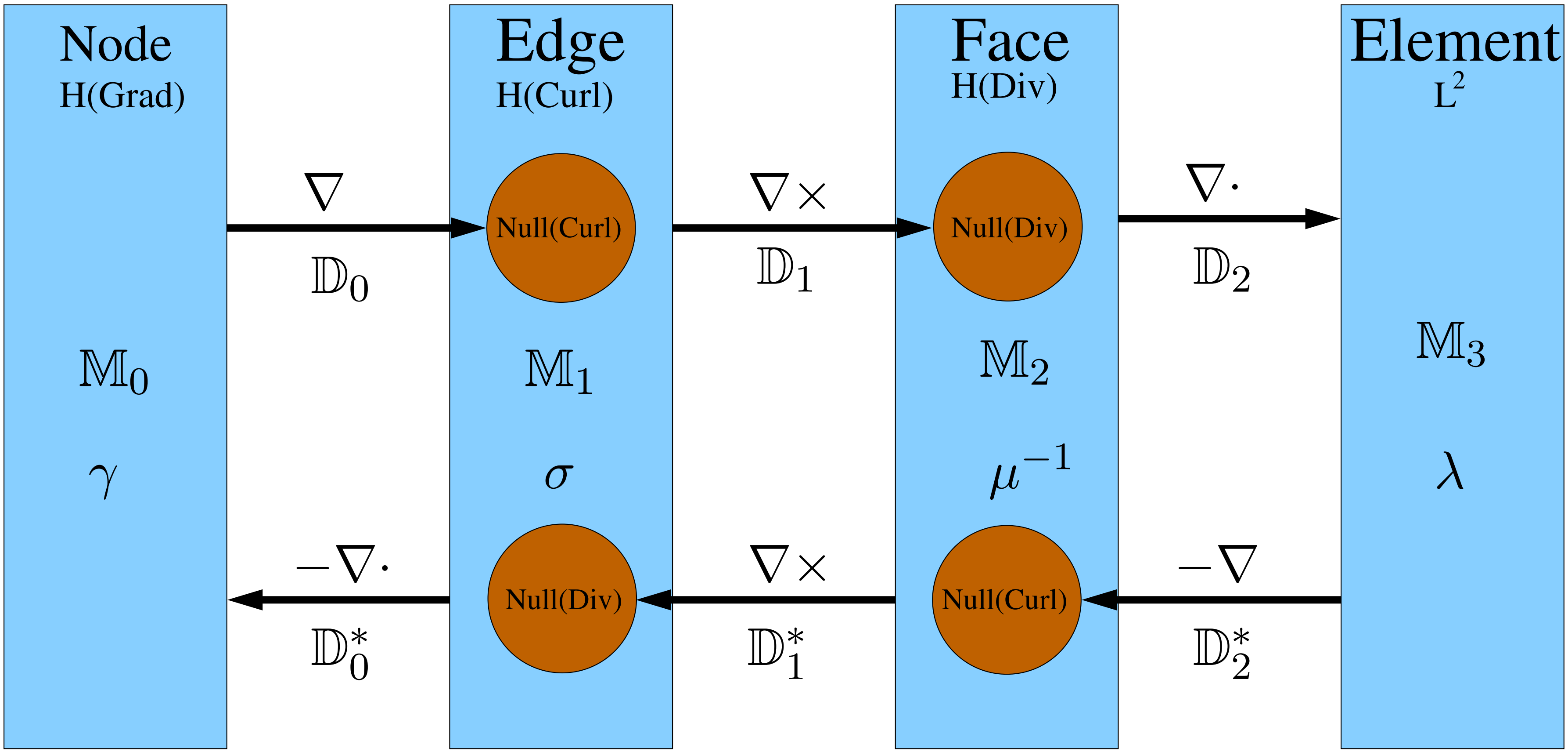


- $\nabla \times \nabla\phi = 0 \Rightarrow$ large null space complicates discretization + solver.
- Large jumps in σ.
- Significant mesh stretching.
- Large problems & repeated solves $\rightarrow$ Scalable linear solvers are critical.

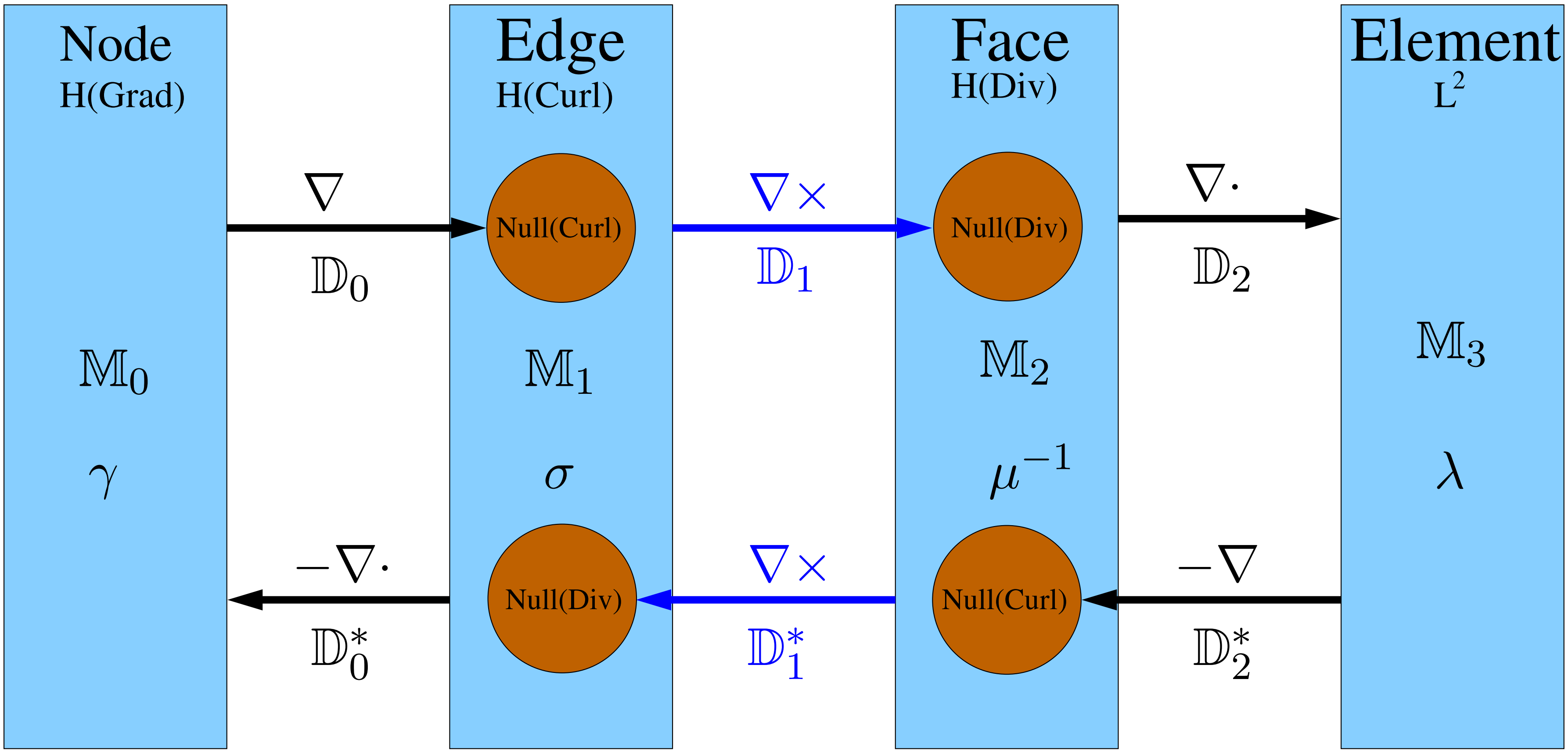


Sandia National Laboratories

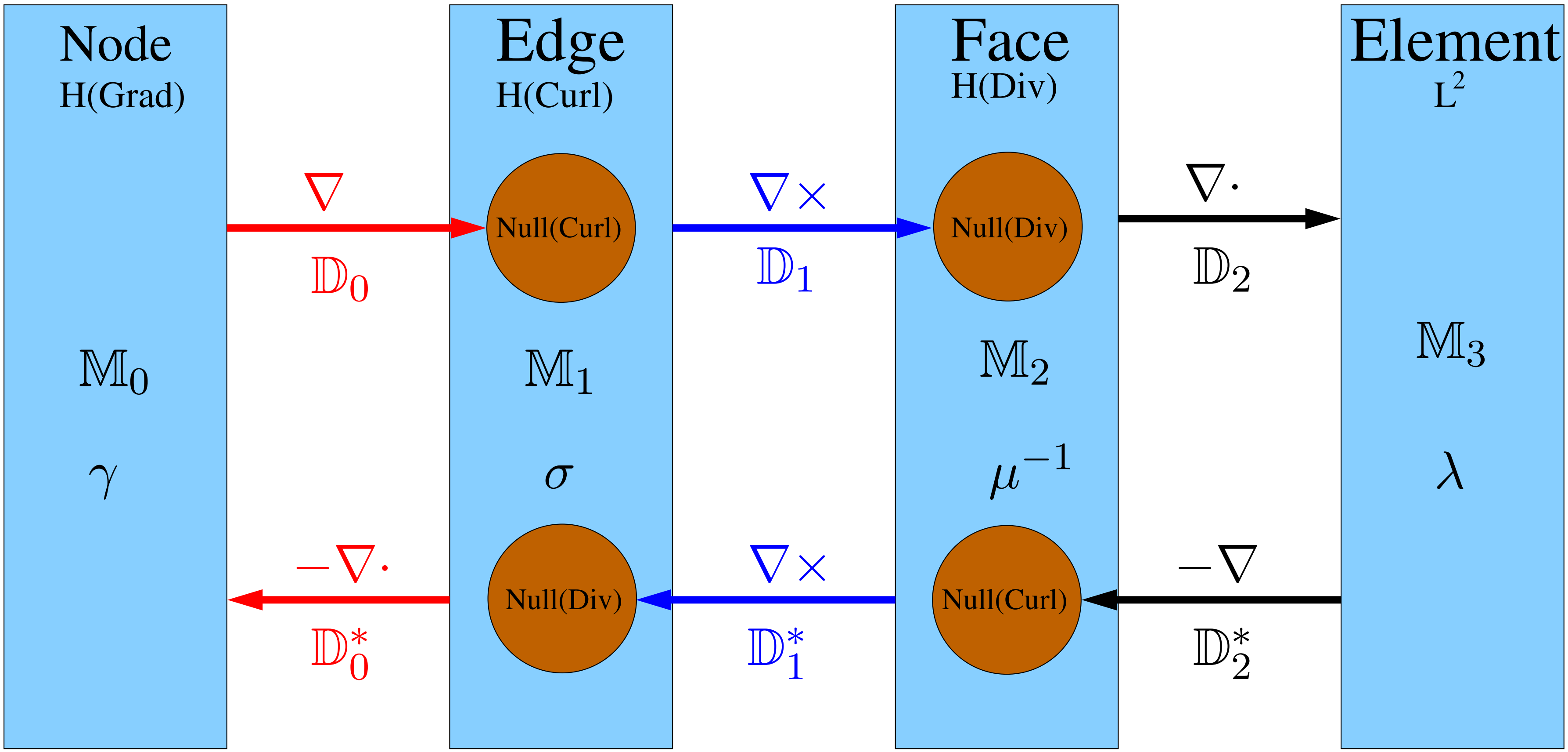
Continuous/Discrete Relationship



Continuous/Discrete Relationship



Hodge Laplacian



- $\Delta = \nabla \times \nabla\times + \nabla\nabla\cdot$.
- $\mathbb{L}_1 = \mathbb{D}_1^*\mathbb{D}_1 + \mathbb{D}_0\mathbb{D}_0^*$

Discrete Hodge Decomposition

$$(\mathbb{M}_1\mathbb{D}_1^*\mathbb{D}_1 + \mathbb{M}_1)e = b$$

- Consider

$$e = a + \mathbb{D}_0 p,$$

where $\mathbb{D}_0^* a = 0$.

- This gives us the block 2×2 system

$$\begin{bmatrix} \mathbb{M}_1\mathbb{D}_1^*\mathbb{D}_1 + \mathbb{M}_1 & \mathbb{M}_1\mathbb{D}_0 \\ \mathbb{M}_0\mathbb{D}_0^* & \mathbb{M}_0\mathbb{D}_0^*\mathbb{D}_0 \end{bmatrix} \begin{bmatrix} a \\ p \end{bmatrix} = \begin{bmatrix} b \\ \mathbb{D}_0^T b \end{bmatrix}.$$

Sandia National Laboratories

Discrete Hodge Decomposition

$$(\mathbb{M}_1 \mathbb{D}_1^* \mathbb{D}_1 + \mathbb{M}_1)e = b$$

- Consider

$$e = a + \mathbb{D}_0 p,$$

where $\mathbb{D}_0^* a = 0$.

- This gives us the block 2×2 system

$$\begin{bmatrix} \mathbb{M}_1 \mathbb{D}_1^* \mathbb{D}_1 + \mathbb{M}_1 + \mathbb{M}_1 \mathbb{D}_0 \mathbb{D}_0^* & \mathbb{M}_1 \mathbb{D}_0 \\ \mathbb{M}_0 \mathbb{D}_0^* & \mathbb{M}_0 \mathbb{D}_0^* \mathbb{D}_0 \end{bmatrix} \begin{bmatrix} a \\ p \end{bmatrix} = \begin{bmatrix} b \\ \mathbb{D}_0^T b \end{bmatrix}.$$

- We can add $\mathbb{D}_0 \mathbb{D}_0^*$ w/o changing answer!

Sandia National Laboratories

Preconditioning

$$\begin{bmatrix} \mathbb{M}_1\mathbb{D}_1^*\mathbb{D}_1 + \mathbb{M}_1\mathbb{D}_0\mathbb{D}_0^* + \mathbb{M}_1 & \mathbb{M}_1\mathbb{D}_0 \\ \mathbb{M}_0\mathbb{D}_0^* & \mathbb{M}_0\mathbb{D}_0^*\mathbb{D}_0 \end{bmatrix} \begin{bmatrix} a \\ p \end{bmatrix} = \begin{bmatrix} b \\ \mathbb{D}_0^T b \end{bmatrix}.$$

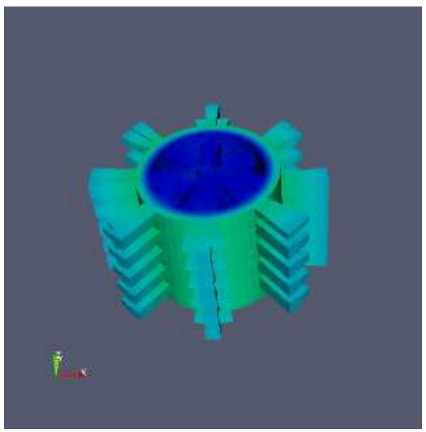
- Use preconditioner:

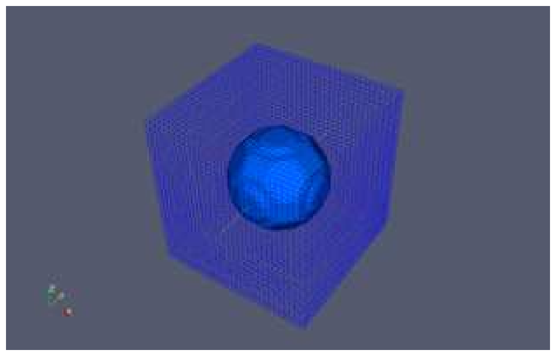
$$P^{-1} = \begin{bmatrix} I & \mathbb{D}_0 \end{bmatrix} \begin{bmatrix} \text{AMG}_{11} & \\ & \text{AMG}_{22} \end{bmatrix} \begin{bmatrix} I \\ \mathbb{D}_0^T \end{bmatrix}$$

- This preconditioner is implemented in ml/src/RefMaxwell in Trilinos 8.0.

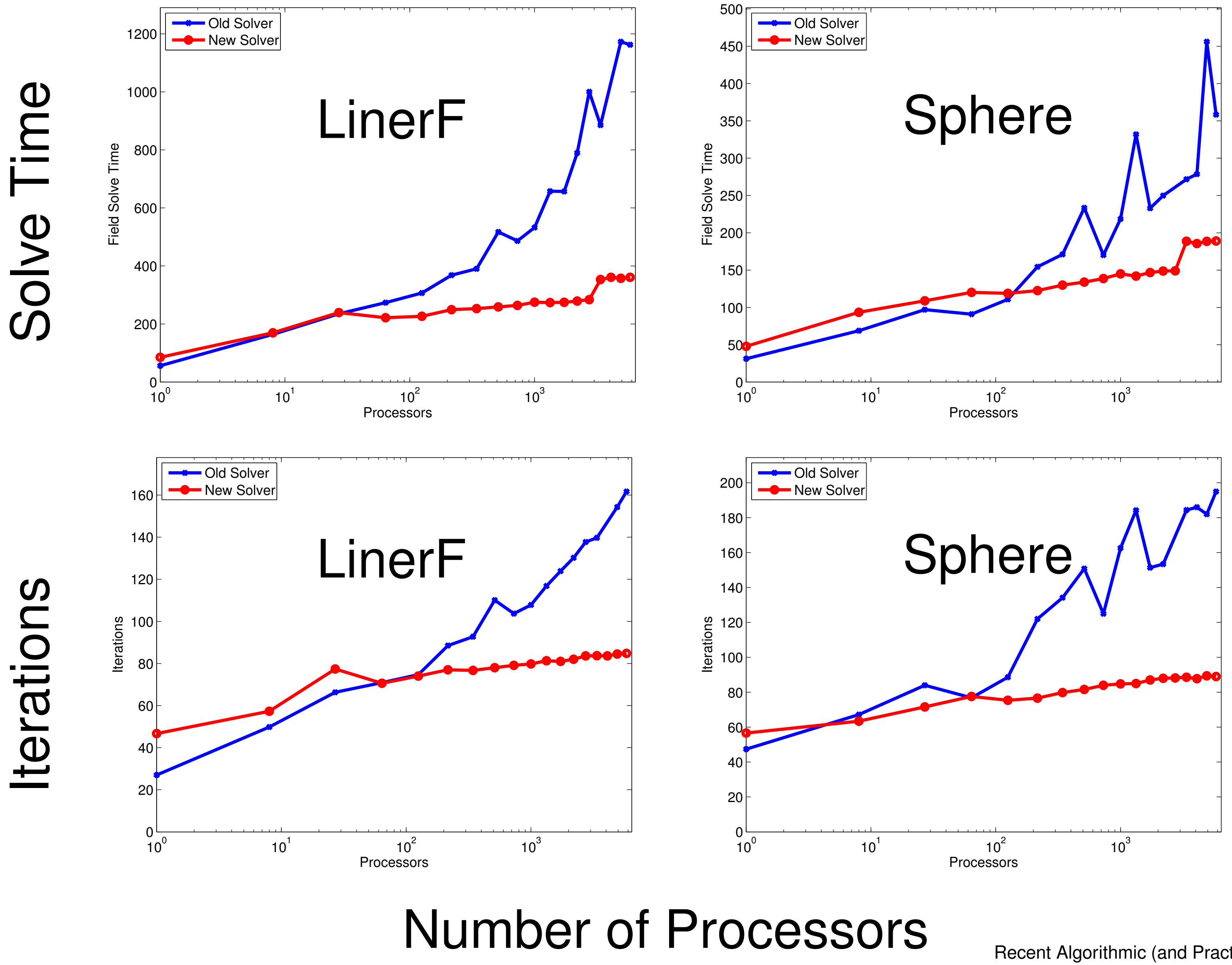
3D Weak Scaling

- Problem Code: ALEGRA (SNL).
- Problems: LinerF, Sphere.
- Material Parameters: $1e6$ jump in conductivity.
- Geometry: Regular meshes.
- Compare Maxwell vs. RefMaxwell





Scaling: Old vs. New

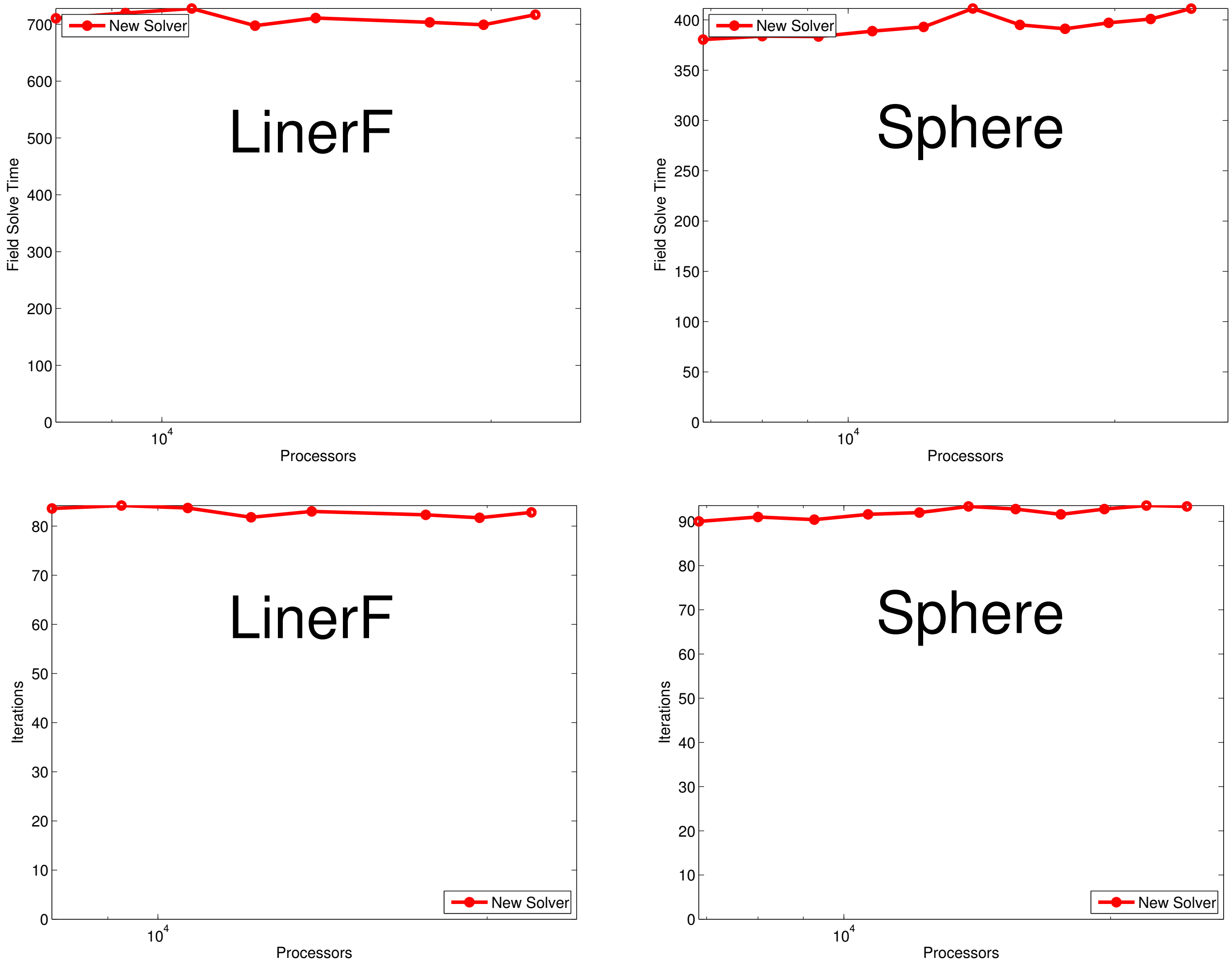


Sandia National Laboratories

Jumbo Scaling: New

Solve Time

Iterations



Number of Processors

Outline

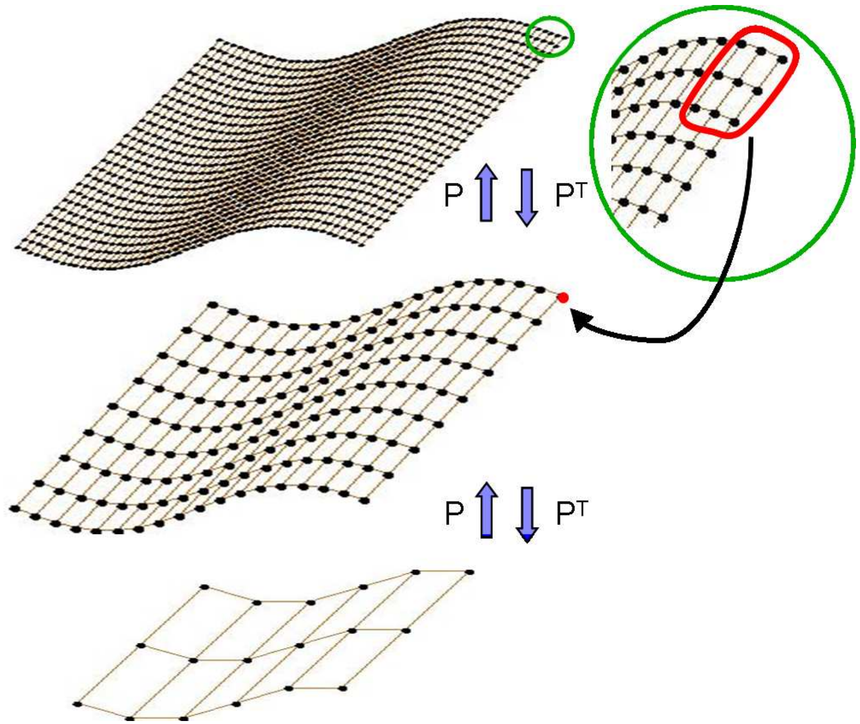
- Introduction to ML.
- Solving Maxwell's Equations w/ RefMaxwell.
- Repartitioning w/ Zoltan and Hypergraphs.
- Conclusions & Future Work.

Why Repartitioning?

- Coarse grids ⇒ less work per proc ⇒ poor performance.
- Solution: Move data to leave some procs idle.

Computation Dominated

Communication Dominated

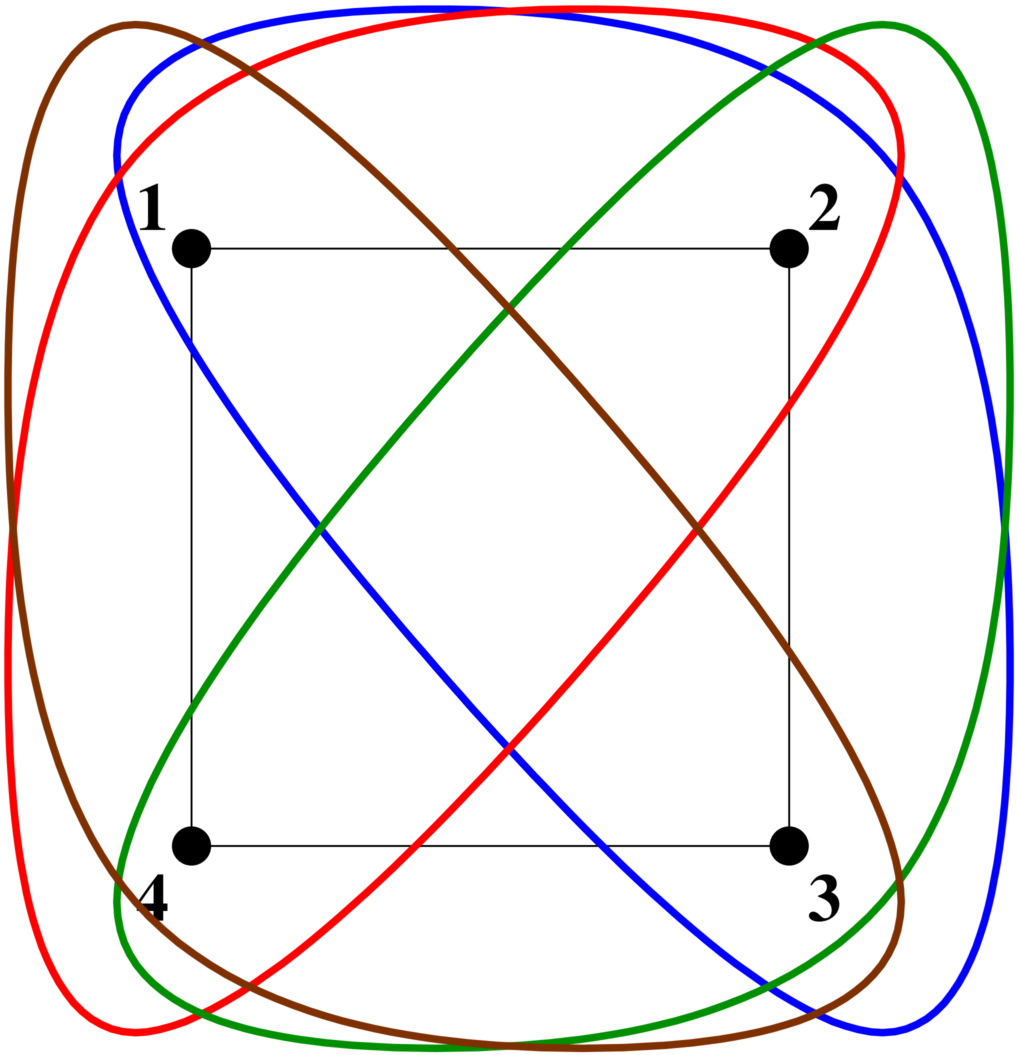


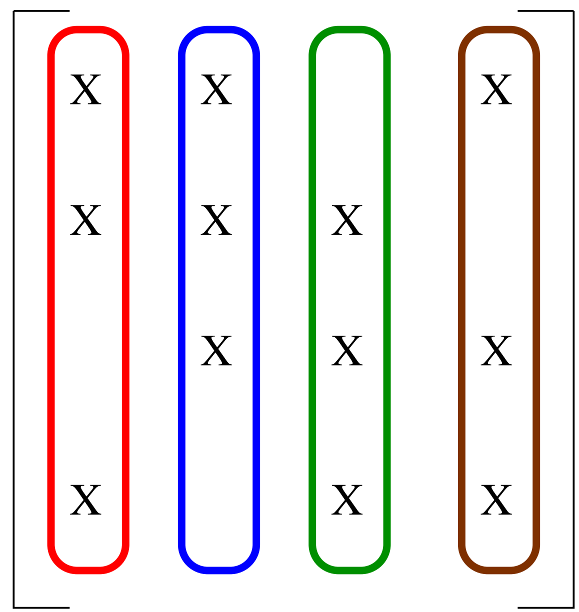
Sandia National Laboratories

Why Repartitioning?

- Goals
  - Move to a subset of processors
    $\Rightarrow$ Increase computation to communication ratio.
  - Rebalance load.
- Method
  - ML current supports RCB via Zoltan.
  - What about irregular meshes?
  - What about consistency between partitions at different levels?
  - New Feature: Hypergraph Repartitioning.
    $\Rightarrow$ To be released in Trilinos 9.0.

What is a Hypergraph?



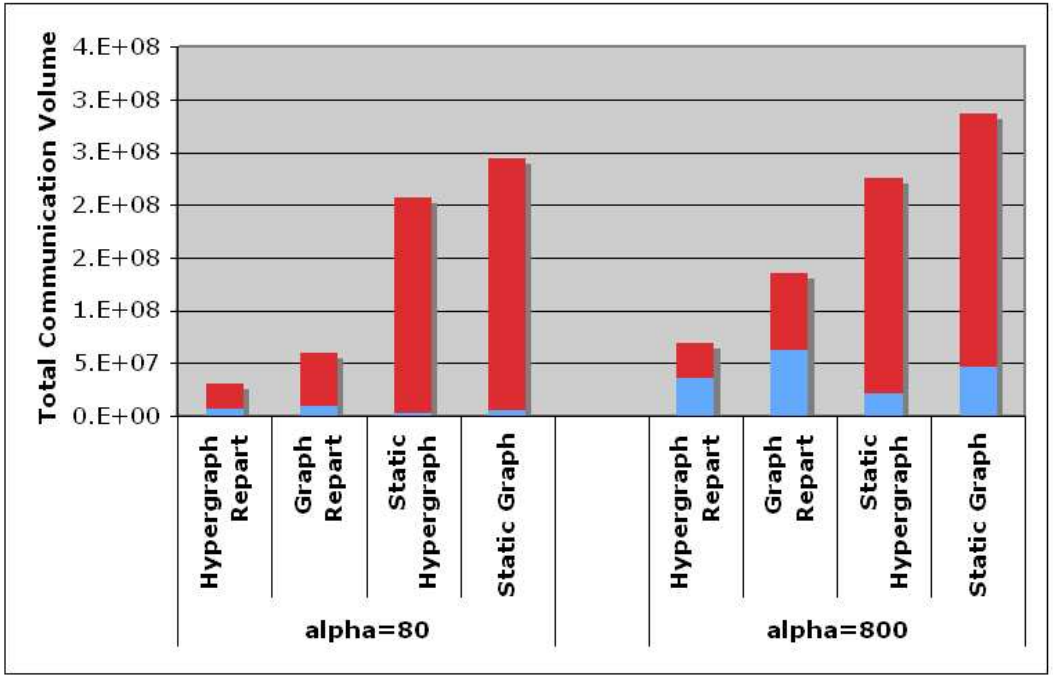


Sandia National Laboratories

Why Hypergraphs?

- For (compressed row) matrix:
  - Vertices == rows.
  - Hyperedges == columns.
  - Weights == Communication volume for that edge $\Rightarrow$ $w_e \cdot$(# processors in edge - 1).
- Why Hypergraphs?
  - Models structurally non-symmetric systems.
  - Models communication costs — esp. important for non-homogeneous meshes.
  - Minimizes combined cost of application communication and data migration.

Zoltan at Work



Data Migration Communication

Application Execution Communication

- 680k rows, 2.3M nonzeros

Outline

- Introduction to ML.
- Solving Maxwell's Equations w/ RefMaxwell.
- Repartitioning w/ Zoltan and Hypergraphs.
- Conclusions & Future Work.

Conclusions

- RefMaxwell
  - Handles jumpy coefficients well.
  - Utilizes existing technology.
  - Scalability up to 24.5k procs!
- Zoltan & Hypergraph Repartitioning
  - Accurately models application and data migration costs.
  - Good results on test problems.
  - Future: ML's unstructured mesh applications.

Future Directions

- TOPS-II: Interface w/ PETSc.
- Extreme Scalability.
- Adaptive methods for circuit problems.
- Improved multimaterial algorithms.