This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2022-3986C

Sandia National Laboratories

Exceptional service in the national interest

# A Matrix-Free Approach for Algebraic Multigrid

Graham Harper
Center for Computing Research
Sandia National Laboratories

04/05/2022

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

# Introduction & Motivation

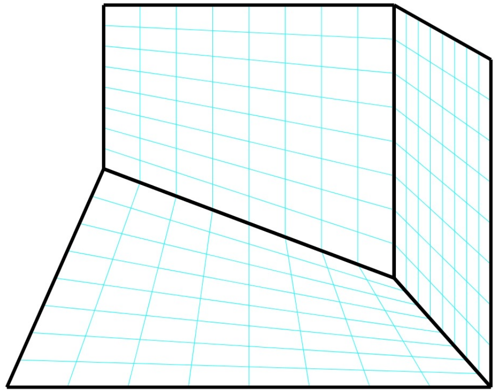
- Collaborators: Ray Tuminaro, David Noble
- Funded by ASCR

- Motivating statements:
  1. Many large-scale high-performance finite element applications are memory-bound
  2. Matrix-free methods trade lower memory usage for higher computational cost
  3. Accelerators such as GPUs may be utilized more effectively in lower-memory scenarios
  4. Without preconditioning, matrix-free performance isn't worth it

- Approaches with GMG have been studied, less with AMG

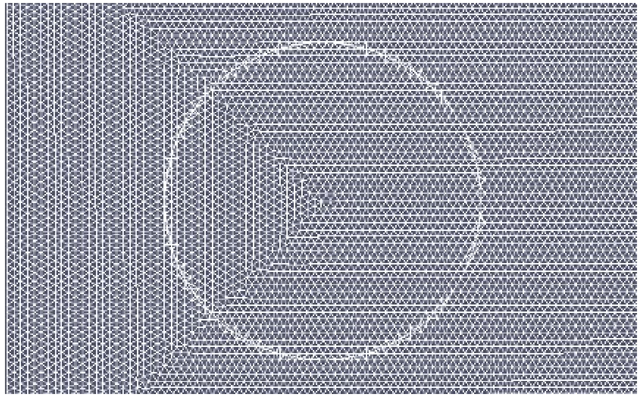
# Potential Approaches for Matrix-Free AMG

- Active subjects include p-coarsening, LOR

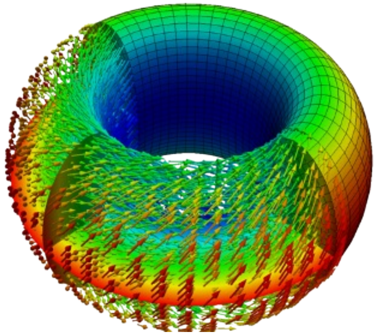


- Auxiliary operators

- AMG needs some sort of graph, explicit RAP
  - Limits benefits on simple scalar problems
  - Mesh graph (CG FEMs) or dual graph (DG FEMs)
  - Alleviate memory constraints via aggressive coarsening



- Applications of interest
  - GMG-AMG combinations
  - Hybrid-structure meshes (HHGs)
  - Collocated DOF multiphysics problems



$$\begin{bmatrix} A & B^T \\ B & -D \end{bmatrix}$$

# Potential Approaches for Matrix-Free AMG

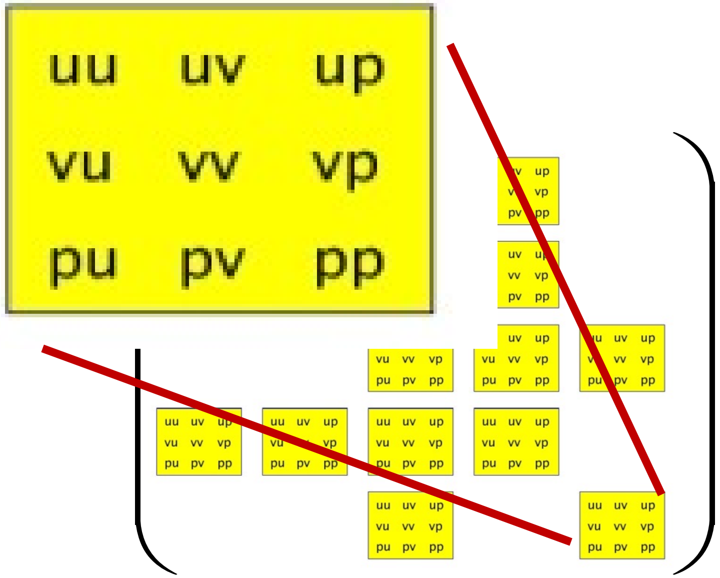
- Specialized “probing” of a matrix-free operator
  - Combine with cheap graph coloring algorithms
  - Compute all matrix entries with few matvecs

- An example multiphysics application
  - Start with mesh nodal graph
  - Perform coarsening on the graph
  - Generate graph transfers using distances
  - Create coarse matrix using a specialized multiplication
  - Coarser levels are handled more traditionally
- Reduces overall memory constraints

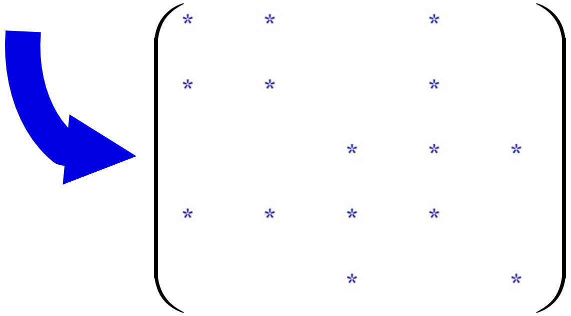
# AMG for Multiphysics

Consecutive DOFs within nodes

$$[u_1\ v_1\ p_1 \quad \cdots \quad u_n\ v_n\ p_n]$$

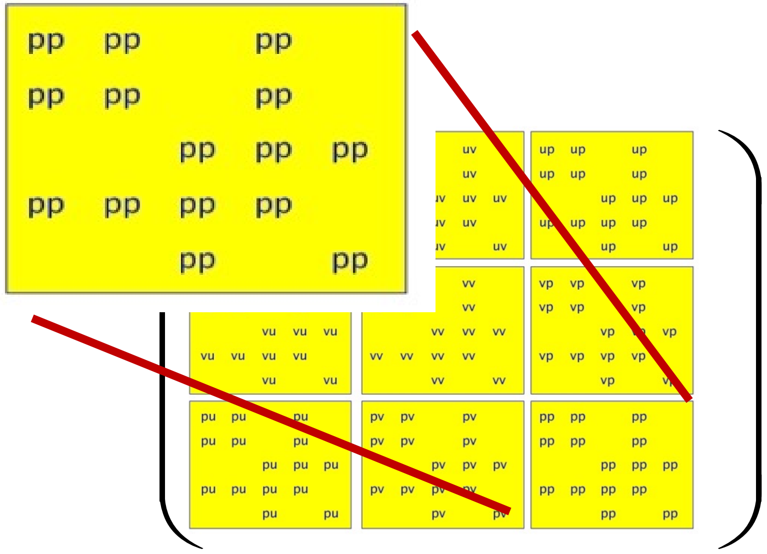



- Graph algorithms on nodes



Consecutive DOFs within fields

$$[u_1 \quad \cdots \quad u_n \quad v_1 \quad \cdots \quad v_n \quad p_1 \quad \cdots \quad p_n]$$



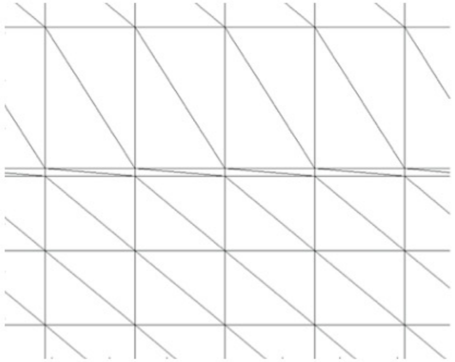
- Leads to blocked prolongator structure

# The Distance Laplacian

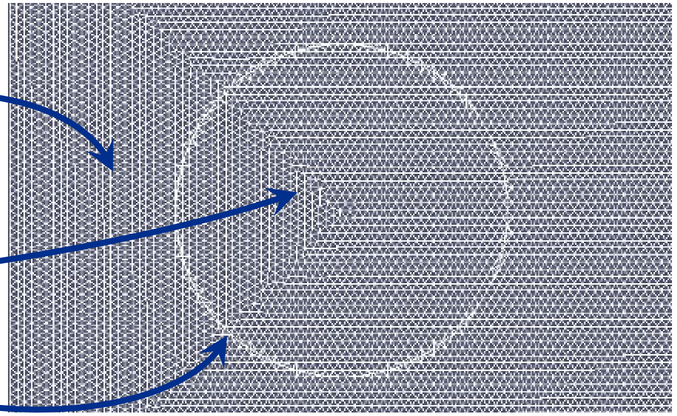
- Focus on multiphysics applications with collocated DOFs from here forward

- The distance Laplacian is defined by

$$L_{ij} = \begin{cases} -1/d(i,j), & i \neq j, A_{ij} \neq 0 \\ -\sum_{k \neq i} L_{ik}, & i = j \\ 0, & \text{otherwise} \end{cases}$$



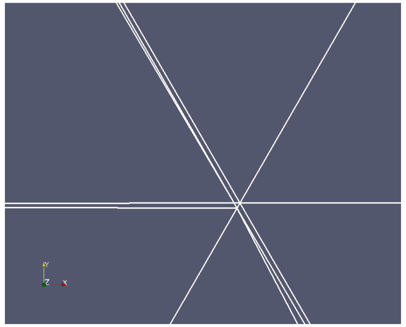
- Implemented in MueLu/ML with more features in progress
  - 3 dofs/node (velocities, water pressure)
  - 3 dofs/node (velocities, air pressure)
  - 4 dofs/node (velocities, air & water pressure)



# Matrix-Free Aspects of the Distance Laplacian

- Computing $L_{ij}$ only requires mesh, connectivity
- Reduces overall problem size

- Generally applicable (using DOF information)
  - Interface problems
  - Multiple species
  - Stabilized-equal order
  - Poor mesh quality



# Smoothed Aggregation AMG

- Smoothed aggregation = prolongator smoother * tentative prolongator

$$P_\ell = \left(I - \omega D_\ell^{-1} A_\ell\right) P_\ell^{(t)}$$

where

$$\omega = \frac{4}{3\lambda_{\ell,m}} \qquad \lambda_{\ell,m} = \rho(D_\ell^{-1} A_\ell)$$

- Distance Laplacian SA-AMG:
  - Replace $A_1$ with L
  - “Unsmoosh” resulting P
  - May resemble linear interpolation

$$\begin{array}{ccc} A & & P_A \\ \downarrow & & \uparrow \\ L & \longrightarrow & P_L \end{array}$$

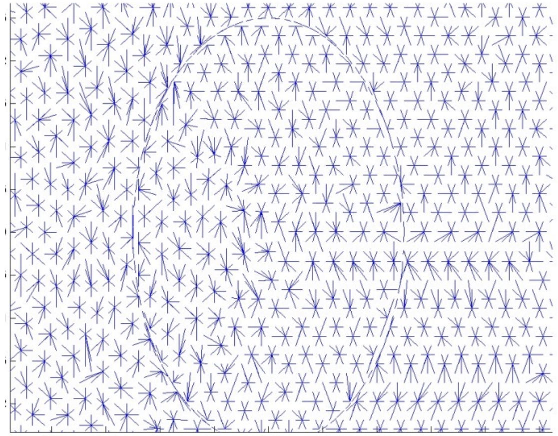
# Strategies for SA-AMG

- Matrix-free case:
  - Eigenvalue estimate and diagonal are friendly
  - Smoothers need to be carefully chosen
  - Aggregates from mesh graph

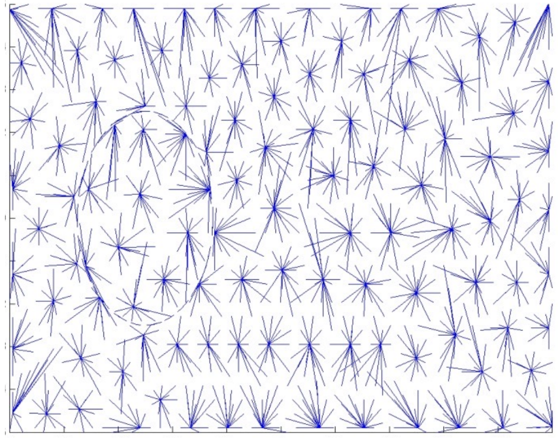
- Aggressive coarsening:
  - Multiple prolongator smooths increases stencil
  - Dropping to fix any poor qualities
    - e.g. locate material jumps with $A_{ii}/L_{ii}$

# Results – Distance Laplacian Only

- Rising bubble problem in Aria:



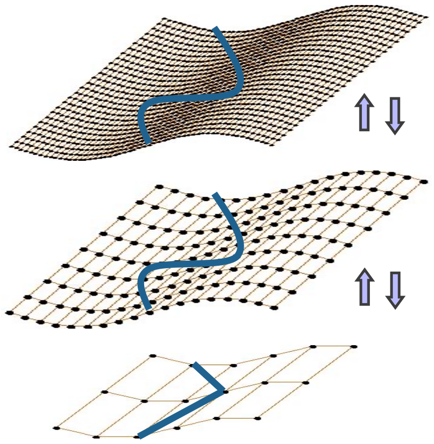
1st level aggregates

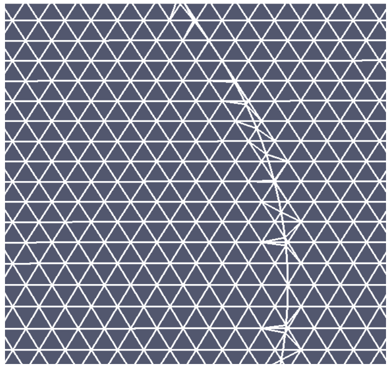


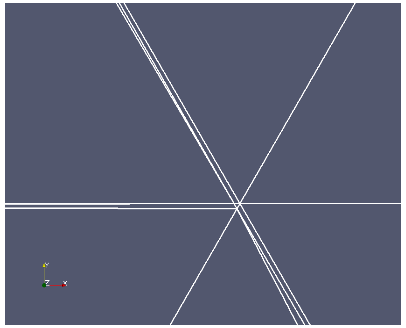
2nd level aggregates

- ILU relaxation to address smoothing concerns
  - Associated with incompressibility constraint
  - Tiny mesh spacing @ interface

| method | iterations |
|---|---|
| ILU only | 180 |
| Unsmoothed/plain aggregation | 25 |
| Smoothed aggregation | 19 |







# Performance Portability for Matrix-Free Solvers

- Performance portability is important as new architectures arise
- Trilinos packages leverage Kokkos for performance portability
- We utilize
  - Intrepid2
  - Panzer
  - Tpetra
  - Belos
  - MueLu

```
apply(const MV& X,
      MV& Y,
      Teuchos::ETransp mode = Teuchos::NO_TRANS,
      scalar_type alpha = Teuchos::ScalarTraits<scalar_type>::one(),
      scalar_type beta = Teuchos::ScalarTraits<scalar_type>::zero()) const
```

```
Y.scale(beta);
```

```
Kokkos::DynRankView<double,PHX::Device>
    physbasisvals("phys basis vals", num_elems, num_basis, num_ip);
Intrepid2::FunctionSpaceTools<PHX::Device::execution_space>
    ::HGRADtransformVALUE(phys_basis_vals, basis_vals);
```

```
kokkos_view_Y(LIDs(e,i),c) += alpha*local_mass(i,j)*kokkos_view_X(LIDs(e,j),c);
```

- Best value for programmatic effort

# Conclusion

- Distance Laplacian
  - Compressed representation of problem
  - Handles poor mesh quality well
- Smoothed aggregation
  - Versatile
- Matrix-Free
  - Plays well with each
  - Hybrid hierarchy approaches
- Plenty of combinations to explore
- Future work
  - Performance tests/improvement
  - Integration with MueLu

# Thank you!

- Questions?