HPSF CONFERENCE 2025, CHICAGO, IL

MARCH 5-8, 2025

Sandia National Laboratories

Exceptional service in the national interest

WHY YOUR SCIENCE APPLICATION SHOULD BE USING TRILINOS LINEAR SOLVERS

Sandia National Laboratories

Jonathan Hu

HPSF Conference 2025

May 5-8, 2025, Chicago, IL

1949–2024

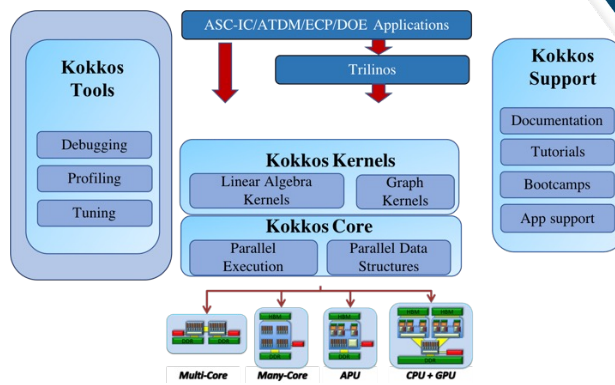
SANDIA NATIONAL LABORATORIES

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

SAND2025-05426C

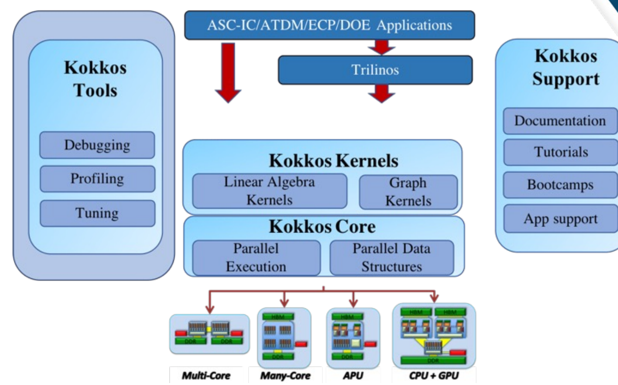
TRILINOS PROJECT

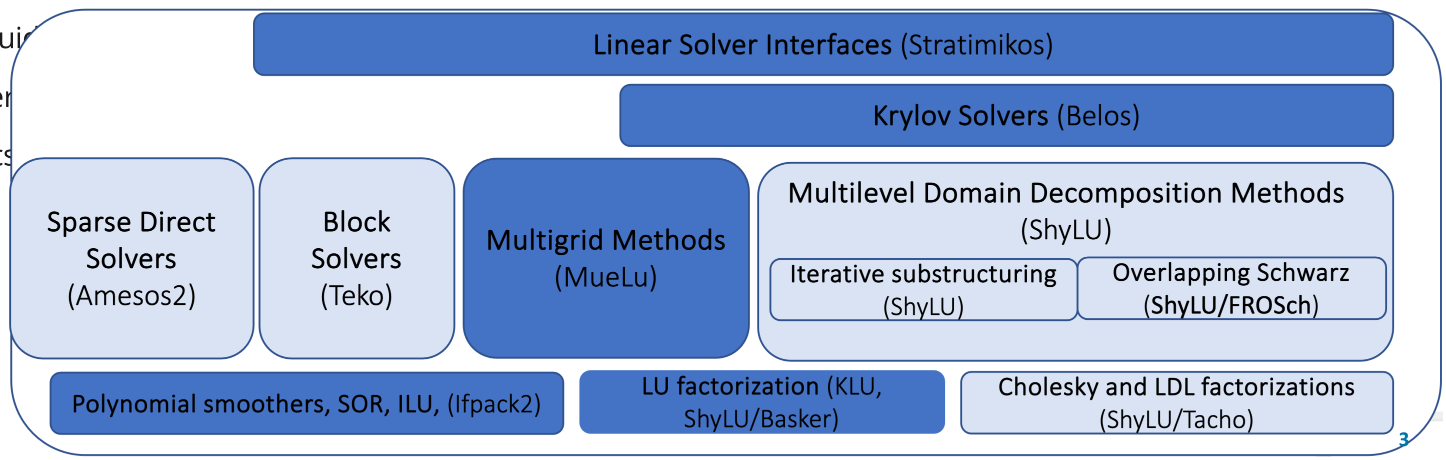
- Numerical libraries project
  - Relies on Kokkos ecosystem for on-node performance portability (github.com/kokkos)
  - Open source (github.com/trilinos/Trilinos)
  - Part of the High Performance Software Foundation (https://hpsf.io)
- Supports many architectures
  - CPUs (AMD, Intel)
  - Accelerators (Nvidia, AMD, Intel)
- Enabling technology for variety of simulations codes
  - Sandia internal (thermal fluid, plasma, multimaterial)
  - DOE Office of Science (energy, complex systems modelling)
  - University-led (multiphysics)



TRILINOS PROJECT

- Numerical libraries project
  - Relies on Kokkos ecosystem for on-node performance portability (github.com/kokkos)
  - Open source (github.com/trilinos/Trilinos)
  - Part of the High Performance Software Foundation (https://hpsf.io)
- Supports many architectures
  - CPUs (AMD, Intel)
  - Accelerators (Nvidia, AMD, Intel)
- Enabling technology for variety of simulations codes
  - Sandia internal (thermal flui
  - DOE Office of Science (ener
  - University-led (multiphysics





CURRENT SOLVERS DEVELOPERS

- Luc Berger-Vergiat
- Erik Boman
- Filipe Cumaru
- Vinh Dang
- Max Fernbach
- Jim Foucar
- Christian Glusa
- Graham Harper
- Alexander Heinlein
- Kyrill Ho
- Jonathan Hu
- Brian Kelley
- Sebastian Kinnewig
- Jascha Knepper
- Jennifer Loe
- Matthias Mayr
- Roger Pawlowski
- Malachi Phillips
- Siva Rajamanickam
- Lea Sassmannshausen
- Chris Siefert
- Heidi Thornquist
- Ray Tuminaro
- Ichitaro Yamazaki

- many other contributors!

ITERATIVE KRYLOV METHODS (BELOS)

C++ framework for developing iterative algorithms for solving large-scale, linear problems.

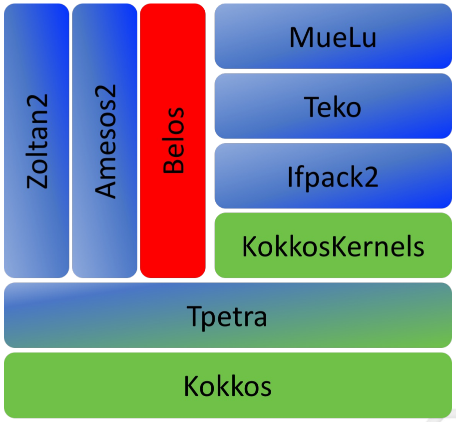
- Standard Krylov methods for single systems — $Ax=b$
  - Simultaneously solved systems w/ multiple RHS: $AX = B$
  - Sequentially solved systems w/ multiple RHS: $AX_i = B_i,\ i=1,\dots,t$
  - Sequences of multiple RHS systems: $A_iX_i = B_i,\ i=1,\dots,t$
- Advanced methods
  - Block methods: block GMRES, block CG/BICG
  - "Seed" solvers: hybrid GMRES
  - Recycling solvers: CGRODR (block recycling GMRES)
  - Restarting techniques, orthogonalization techniques, …
- Algorithms implemented using traits & abstract base classes

ALGEBRAIC SOLVERS (IFPACK2)

- Methods are typically used as Krylov preconditioner or multigrid smoother
- Provides:
  - SOR iterative methods: Jacobi, Gauss-Seidel
  - Chebyshev polynomial
  - Incomplete factorizations (traditional & iterative)
  - Additive Schwarz
  - Block Jacobi
  - Block tri-diagonal
- Interfaces to shared memory algorithms in Kokkos-Kernels
  - ILUT, ILU(k)

MULTIGRID (MUELU)

- Smoothers: Jacobi, GS, $\ell 1$ GS, polynomial, ILU (traditional & iterative), block
- Unstructured algorithms
  - classic smoothed aggregation (SA)
  - non-symmetric AMG
  - AMG for Maxwell's equations
- Structured Algorithms
  - semi-coarsening AMG, geometric MG
  - structured-grid aggregation-based MG
- Matrix load-balancing: multijagged, RCB
- Leverages many other Trilinos libraries
  - Shared memory parallelism from Kokkos → architecture portability
  - Sparse distributed linear algebra: Tpetra
  - Distributed smoothers: Ifpack2
  - Shared memory smoothers, SpGEMM, distance-2 coloring: Kokkos-Kernels
  - Load balancing: Zoltan2
  - Direct Solvers: Amesos2



FROSch (Fast and Robust Overlapping Schwarz) Framework in Trilinos

Software

- Object-oriented C++ domain decomposition solver framework with MPI-based distributed memory parallelization
- Part of TRILINOS with the parallel linear algebra based on TPETRA
- Node-level parallelization and performance portability on CPU and GPU architectures through KOKKOS and KOKKOSKERNELS
- Accessible through unified TRILINOS solver interface STRATIMIKOS

Methodology

- **Parallel scalable multi-level Schwarz domain decomposition preconditioners**
- **Algebraic construction** based on the parallel distributed system matrix
- **Extension-based coarse spaces**

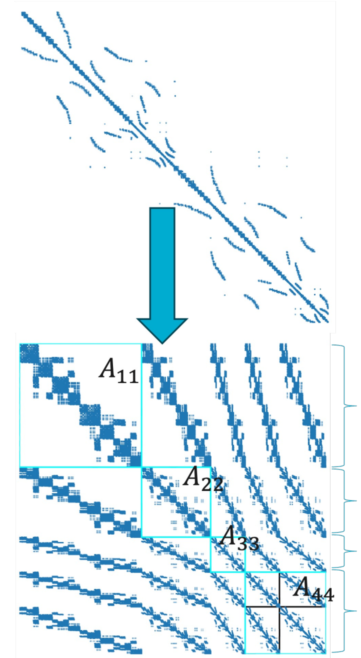
Team (active)

- Filipe Cumaru (TU Delft)
- Alexander Heinlein (TU Delft)
- Kyrill Ho (UCologne)
- Sebastian Kinnewig (LUH)
- Axel Klawonn (UCologne)
- Jascha Knepper (UCologne)
- Stephan Köhler (TUBAF)
- Friederike Röver (TUBAF)
- Siva Rajamanickam (SNL)
- Oliver Rheinbach (TUBAF)
- Lea Saßmannshausen (UCologne)
- Ichitaro Yamazaki (SNL)

Slide courtesy A. Heinlein

MONOLITHIC BLOCK MULTI-PHYSICS PRECONDITIONING (TEKO)

- Provides block inverse schemes:
  - <u>Generic</u>: Block Jacobi, (Hierarchical) Block Gauss-Seidel
  - <u>Navier-Stokes</u>: SIMPLEC, PCD, LSC
  - <u>*Extensible:*</u> Easy-to-develop user-defined block splitting
- Sub-blocks inverted via various Trilinos packages:
  - Ifpack2, Belos, MueLu, etc.
  - *Adaptive* sub-block solvers based on target residual reduction
- Interfaces with Tpetra and Epetra types
  - GPU sub-block assembly for Tpetra
- Automated preconditioner selection for generic multi-physics
  - Talk by Malachi Phillips @ 4:00 PM
- Application talk by Matthias Mayr @ 3:40pm

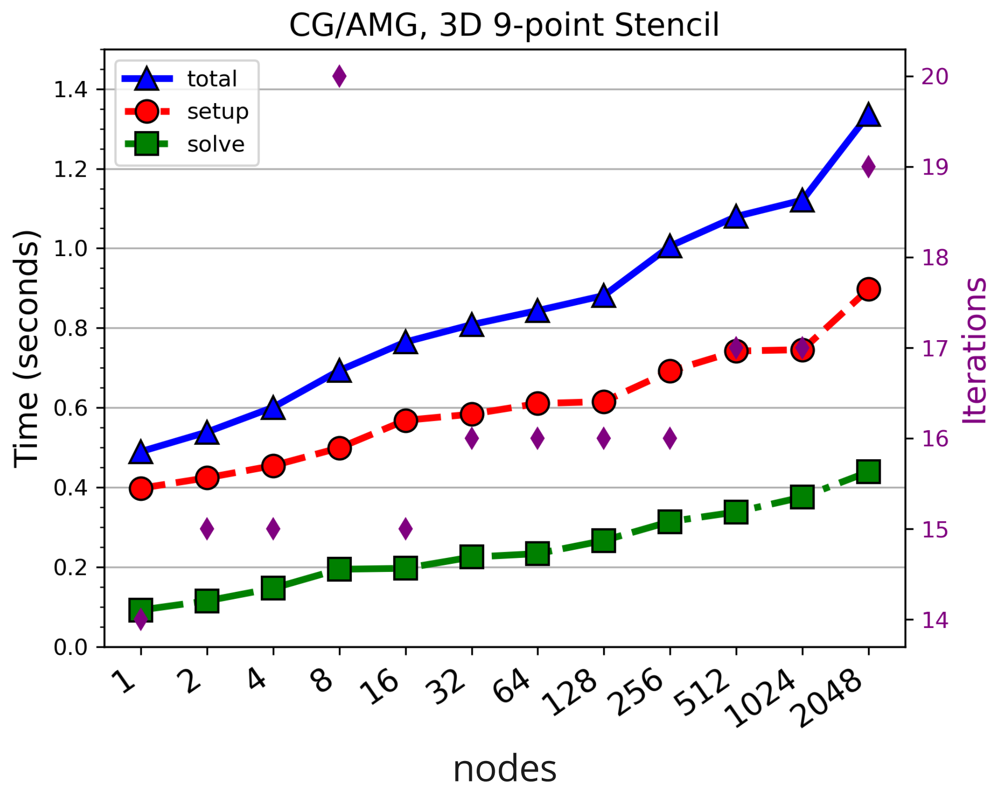


(Teko means “fuse”)

NUMERICAL EXAMPLES

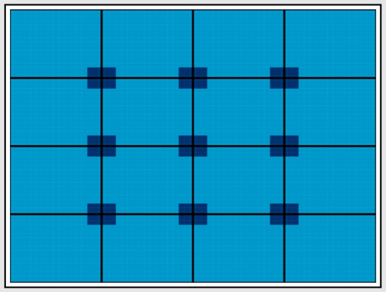
ALGEBRAIC MULTIGRID PERFORMANCE, 3D MODEL PROBLEM

- 3D 9-point stencil
- Weak scaling, 300k rows/rank
- Preconditioned conjugate gradient
  - aggregation-based algebraic multigrid
  - Polynomial smoother on all levels
- Frontier
  - AMD MI-250x
  - 8 to 16384 MPI ranks

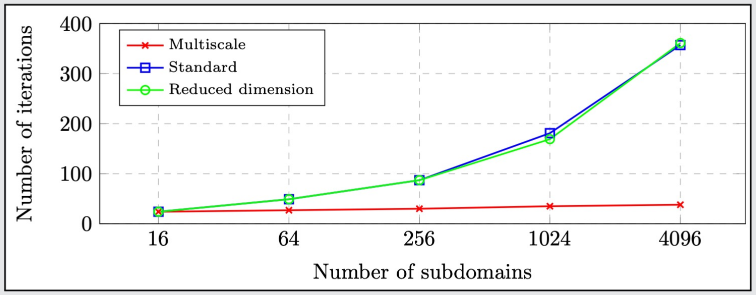


Weak Scalability for the Scalar Poisson Equation (FROSch)

2D heterogeneous Poisson problem (Cumaru, Heinlein, and Hajibeygi, 2024)



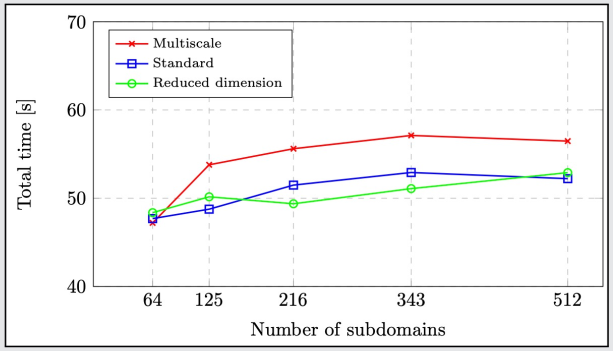
Dark blue: $\alpha = 10^8$; **light blue:** $\alpha = 1$

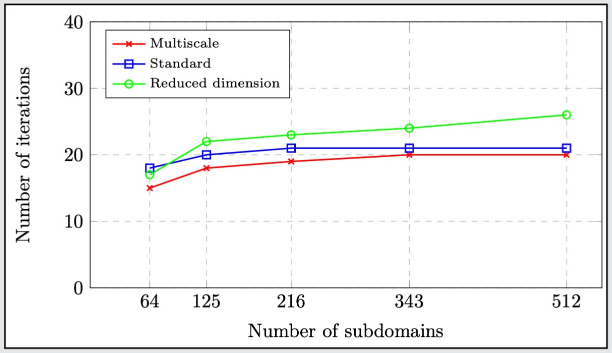


Comparison of coarse spaces with $H/h = 16$, two layers of overlap

The **algebraic multiscale coarse space** is also **robust** for **highly heterogeneous problems**.

3D homogeneous Poisson problem on FROSch running on DelftBlue (TU Delft)





Comparison of coarse spaces with $H/h = 30$, two layers of overlap

Slide courtesy A. Heinlein

APPLICATIONS

TRILINOS: REMOVAL OF 32-BIT STACK

- Trilinos initially based on sparse linear algebra library, Epetra
  - 32-bit, CPU only
  - MPI-centric, with limited threading support
- To use Trilinos, many Sandia apps adopted Epetra linear algebra model

- In Fall 2025, Epetra and associated libraries are slated to be removed

- <u>Question</u>: How performant are Tpetra-based solvers relative to Epetra-based counterparts?

THREE PERFORMANCE TEST PROBLEMS FOR RESISTIVE MHD IN ALEGRA

1. 2D axisymmetric exploding wire system
   - *Up to 23 million unknowns*
   - *Up to 1152 cores*

2. 3D high-velocity impact in magnetic field
   - *Up to 21 million unknowns*
   - *Up to 1152 cores*

3. 3D exploding bridgewire (EBW) detonator
   - *132 million unknowns*
   - *720 cores*

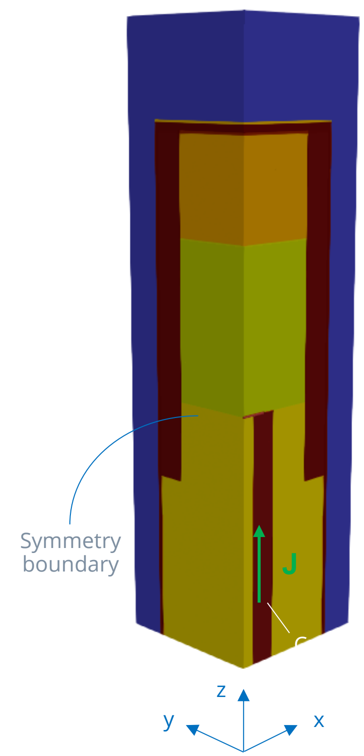


Figure from https://doi.org/10.1016/j.cma.2024.117164

Courtesy: J. Niederhaus

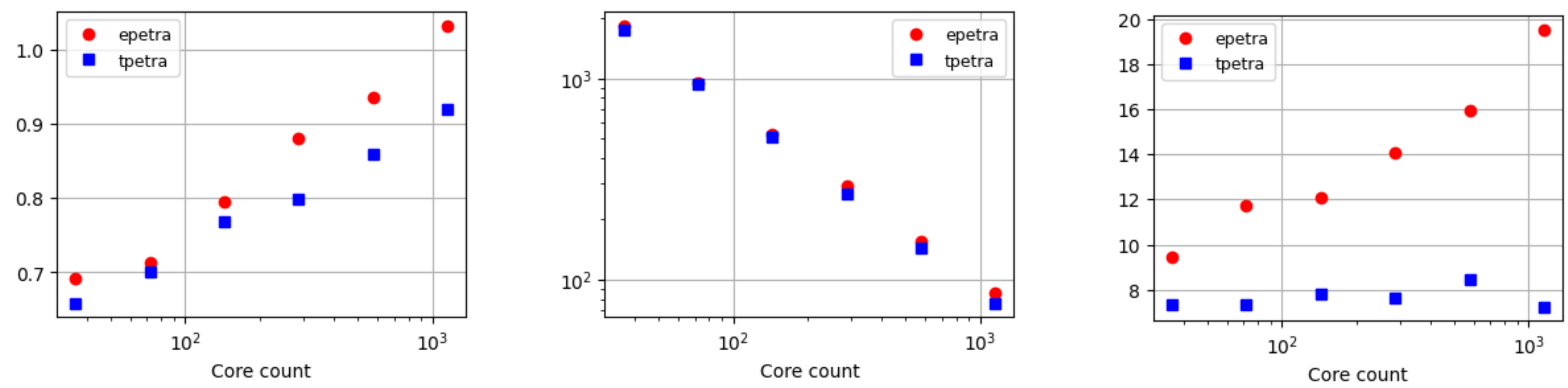
1. EXPLODING WIRE, 2D, WEAK SCALING STUDY (CPU)

Performance comparison of ALEGRA application using Trilinos/Epetra and Trilinos/Tpetra stacks:

- Conjugate gradient Krylov solver, SA-AMG preconditioner

Sandia “Eclipse” CTS-1 cluster

- dual socket Intel Cascade Lake (2.1 – 2.9 GHz), 128GB or 192 GB per node
- Cornelis Omni-path interconnect



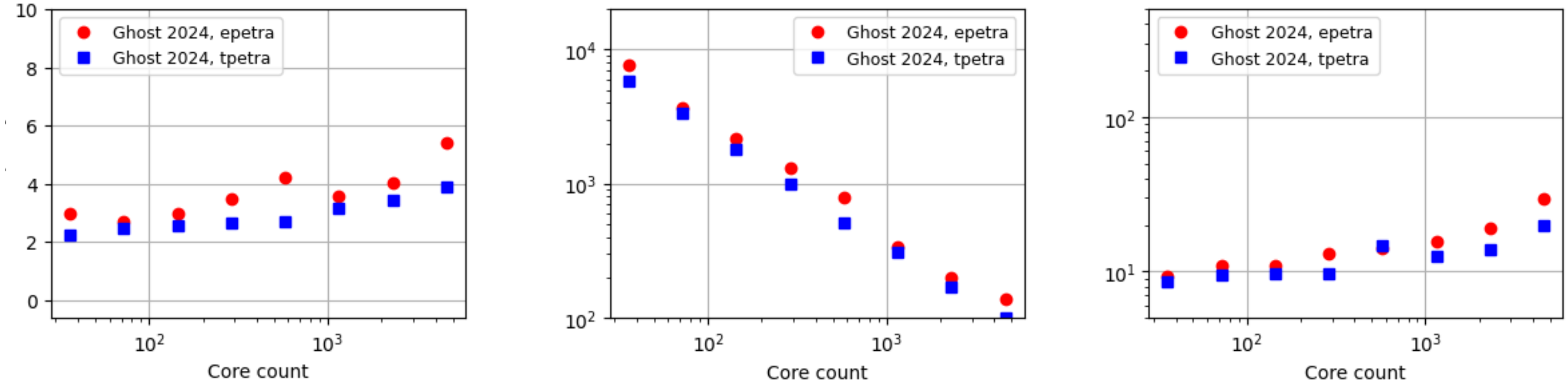
Courtesy: J. Niederhaus

2. IMPACT IN MAGNETIC FIELD, 3D, WEAK SCALING (CPU)

Alegra simulation, **magnetized oblique impact** problem, comparing performance of Trilinos/Epetra and Trilinos/Tpetra stacks:

- Epetra: (AztecOO/ML) Conjugate gradient with Maxwell2 formulation
- Tpetra: (Belos/MueLu) Conjugate gradient with Maxwell2 formulation

- Sandia "Ghost" CTS-1 cluster
  - Dual socket 2.1 Ghz Broadwell, 128 GB per node, Cornelis Omni-Path interconnect



Courtesy: J. Niederhaus

3. EXPLODING BRIDGEWIRE (EBW) DETONATOR, 3D

Single large **EBW** simulation run using:

- Epetra: (AztecOO/ML) Conjugate gradient, Maxwell2 formulation
- Tpetra: (Belos/MueLu) Conjugate gradient, Maxwell2 formulation

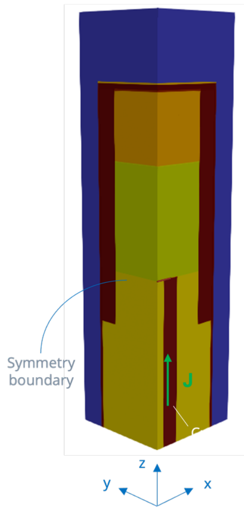
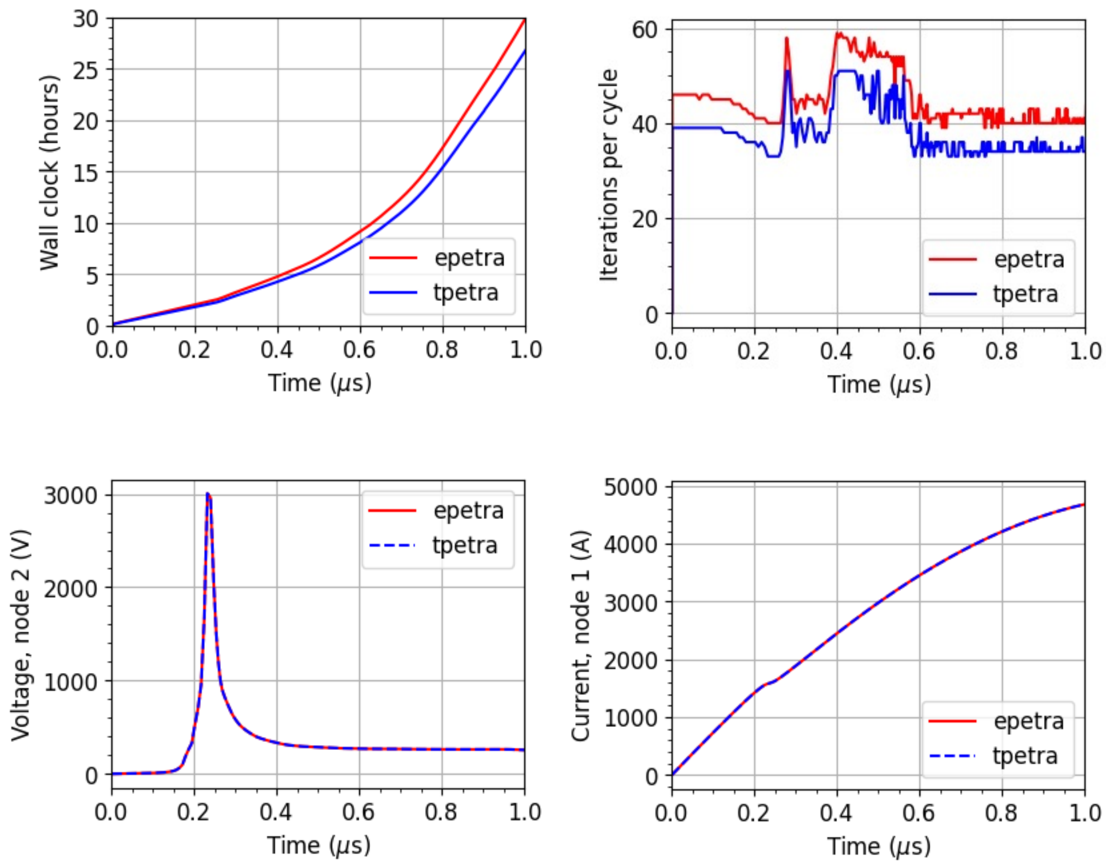
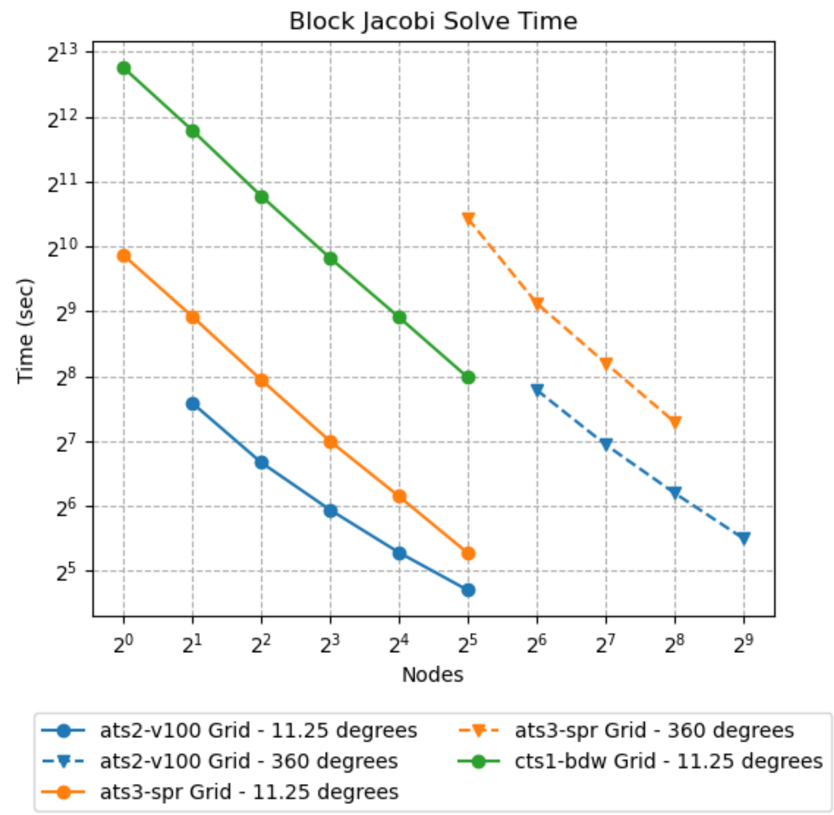


Figure from https://doi.org/10.1016/j.cma.2024.117164



BLOCK JACOBI SOLVER IN CFD SIMULATIONS

- Sandia SPARC application
  - CFD simulations for high-speed flows
- Test case: wall-modeled LES, perfect gas
  - 5 DOFs/cell
  - Two grid resolutions
    - 11.25 degree – 17.3e6 cells
    - 360 degree – 5.53e7 cells
- Solver
  - Ifpack2 Block Jacobi
- Architectures
  - ats2: Nvidia V100
  - ats3: Intel Sapphire Rapids
  - cts1: Intel Broadwell

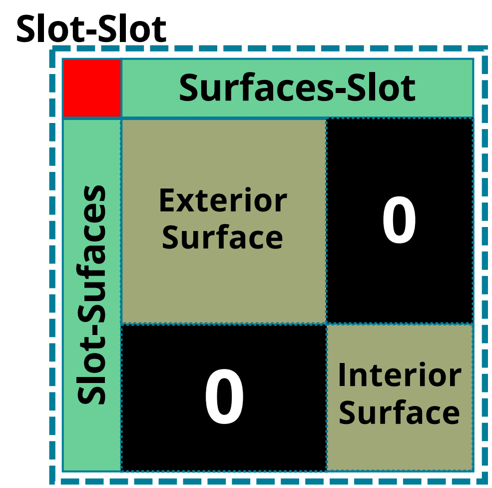


plot courtesy V. Brunini

MULTI-REGION SOLVER AND PRECONDITIONERS FOR ITERATIVE SOLUTION OF HIGH-Q CAVITY PROBLEMS

- Multi-region distributed-memory solver framework:
  - Selection of appropriate solver for each region interaction
  - Use of fast numerical techniques, e.g. adaptive cross approximation (**ACA***)
  - Use of Kokkos, Kokkos Kernels and Belos iterative solvers
  - Reduction of matrix memory footprint (**~5.5x** vs. direct solver on our test problem)
- Current preconditioner: additive Schwarz using local block-diagonal preconditioners from self-interaction blocks
- New local prec. approach: multi-iteration Simple Algebraic Approximate Inverse (**SAAI**)
  - Zero two blocks closest to diagonal blocks and transform diagonal blocks
  - Approximate $X = M^{-1}Y \approx Z^{-1}Y$ via multiple iterations in each preconditioner apply
- Test problem: high-Q cylindrical cavity at f=1. 2GHz with 126,976 unknowns
  - SAAI preconditioner is **~3x** better than local block-diagonal preconditioner in terms of memory footprint, convergence rate, and iterative solution time
- Current work
  - Integration of SAAI preconditioner into Gemma is ongoing
  - Optimizing multi-region solver framework (e.g., matrix fill)

Slide courtesy of Vinh Dang



GMRES results	Block-diagonal	1-iteration SAAI	2-iteration SAAI
Max. number of blocks in Krylov subspace	900	300	300
Krylov subspace memory per rank (MB)	1769.75	596.21	596.21
Number of iterations	1799	1642	548
Solve time (sec.)	213.87	107.80	57.11

SOME ONGOING RESEARCH

ACTIVITIES IN FROSCH

- Land ice simulations & GPUs [I. Yamazaki]
- Optimized Schwarz (https://arxiv.org/abs/2410.22871)
- Optimizing many to few communication [C. Pearson, C. Glusa, I. Yamazaki, A. Heinlein]
- Spectral coarse spaces in FROSch [Jascha Knepper (U. Cologne), I. Yamazaki, A. Heinlein]
- Monolithic coarse spaces in FROSch [L. Sassmannshausen, A. Klawonn, J. Knepper (all U. Cologne), A. Heinlein]
- Algebraic multiscale coarse spaces (ongoing work; see https://arxiv.org/abs/2408.08187)

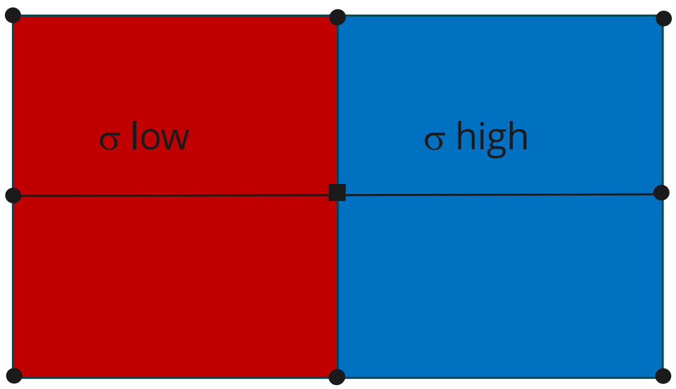
NEW ALGEBRAIC MULTIGRID STRENGTH-OF-CONNECTION MEASURE

- Solving linear systems arising from multimaterial problems is challenging for AMG using standard SoC measure

$$|a_{i,j}| > \epsilon\sqrt{|a_{ii}||a_{jj}|}$$

- New SoC algorithm developed for multi-material problems.

Slide courtesy of Chris Siefert

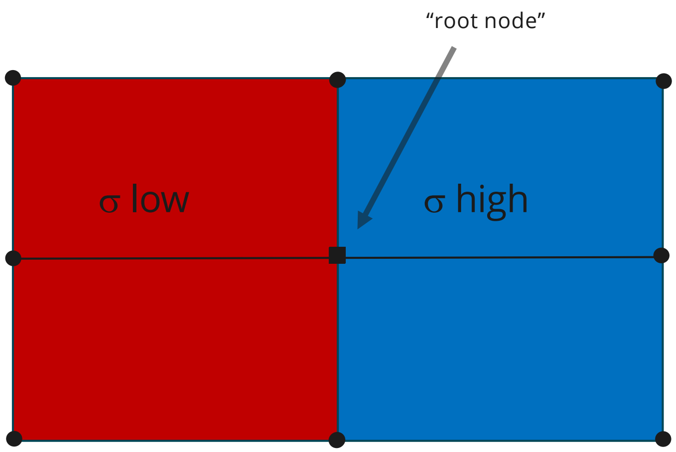


NEW ALGEBRAIC MULTIGRID STRENGTH-OF-CONNECTION MEASURE

- Solving linear systems arising from multimaterial problems is challenging for AMG using standard SoC measure

$$|a_{i,j}| > \epsilon\sqrt{|a_{ii}||a_{jj}|}$$

- New SoC algorithm developed for multi-material problems.
- <u>Example</u>: Consider two-material symmetric operator
  - During AMG coarsening, fine DOFs grouped to create one coarse DOF. Want to preserve material interface.



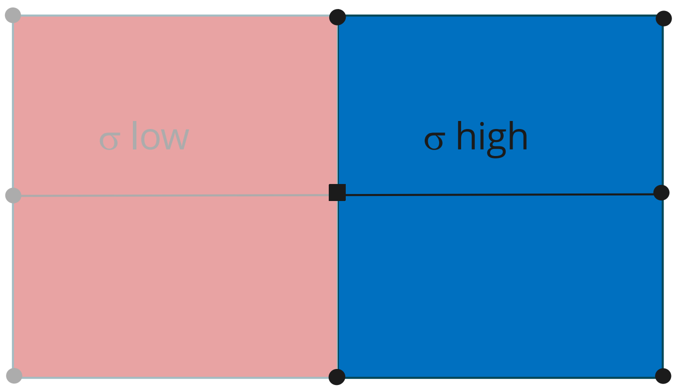
Slide courtesy of Chris Siefert

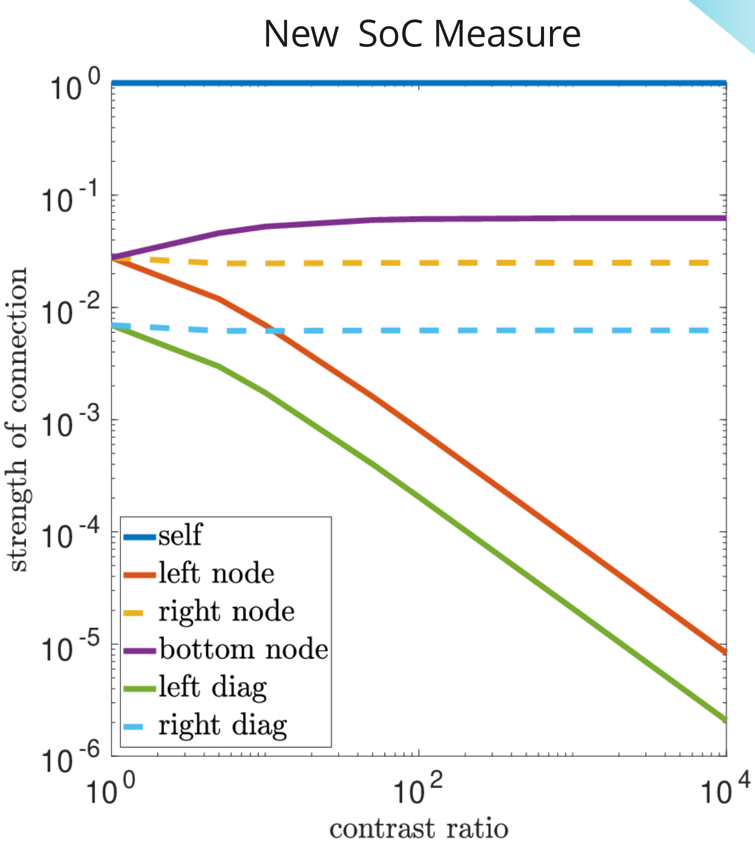
NEW ALGEBRAIC MULTIGRID STRENGTH-OF-CONNECTION MEASURE

- Solving linear systems arising from multimaterial problems is challenging for AMG using standard SoC measure

$$|a_{i,j}| > \epsilon\sqrt{|a_{ii}||a_{jj}|}$$

- New SoC algorithm developed for multi-material problems.
- <u>Example</u>: Consider two-material symmetric operator
  - During AMG coarsening, fine DOFs grouped to create one coarse DOF. Want to preserve material interface.
    - Keep connection to *high* diffusion on right
    - Drop connection connections *low* diffusion on left.





Slide courtesy of Chris Siefert

SOLVING MULTIPHYSICS SYSTEMS USING TEKO

- See Malachi Phillip’s talk later today!

https://commons.wikimedia.org/wiki/File:Furoshiki_designed_by_Friedensreich_Hundertwasser_for_%27Fernw%C3%A4rme_Wien%27_AG.jpg

CONCLUDING REMARKS

- How to get started with Trilinos solvers
  - ATPESC solver tutorials
    - https://tinyurl.com/atpesc-2024-trilinos-solvers or
  - MueLu tutorial
    - coming soon!

- Related talks:
  - Matthias Mayr @ 3:40pm: Application experience
  - Malachi Phillips @ 4pm: Teko preconditioners for multiphysics
  - Christian Glusa @ 5pm: Python interfaces