SAND2025-05458C

Sandia National Laboratories

Exceptional service in the national interest

Automated Preconditioner Design in the Trilinos/Teko Package (UUR)

Malachi Phillips and Aidan Gould

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.

- Interested in solving complex multiphysics problems (e.g., thermal batteries)
  - Often require monolithic linear system approaches (i.e., no operator splitting)
- Thermal batteries include several multi-physics couplings[0]:
  - Butler-Volmer
  - Stefan-Maxwell
  - Darcy's Law
  - Continuity
- *Weak Scaling:* 57,640 to 461,120 DOFs

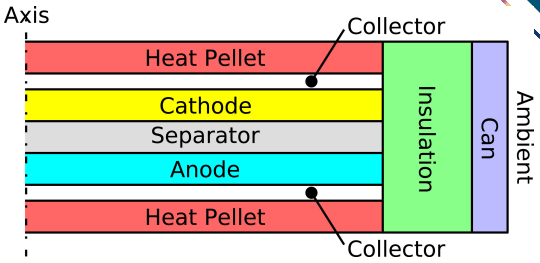
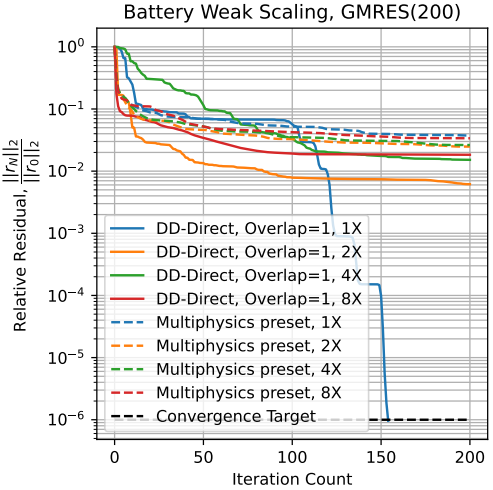


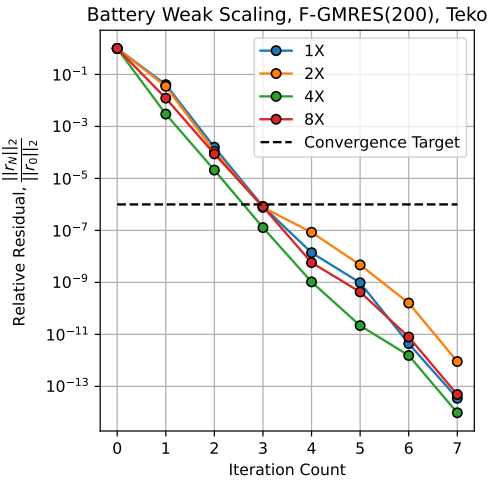
Figure: 2D axisymmetric multi-physics simulation domain[0]

- Compare (flexible) GMRES with two preconditioners:
  - 'Black-box' domain-decomposition
  - 'Physics-aware' block preconditioning

[0] Voskuilen, Moffat, Schroeder, and Roberts, "Multi-fidelity electrochemical modeling of thermally activated battery cells".



(a) 'Black-box'



(b) 'Physics-aware'

Figure: 'Black-box' versus 'physics-aware' preconditioner performance.

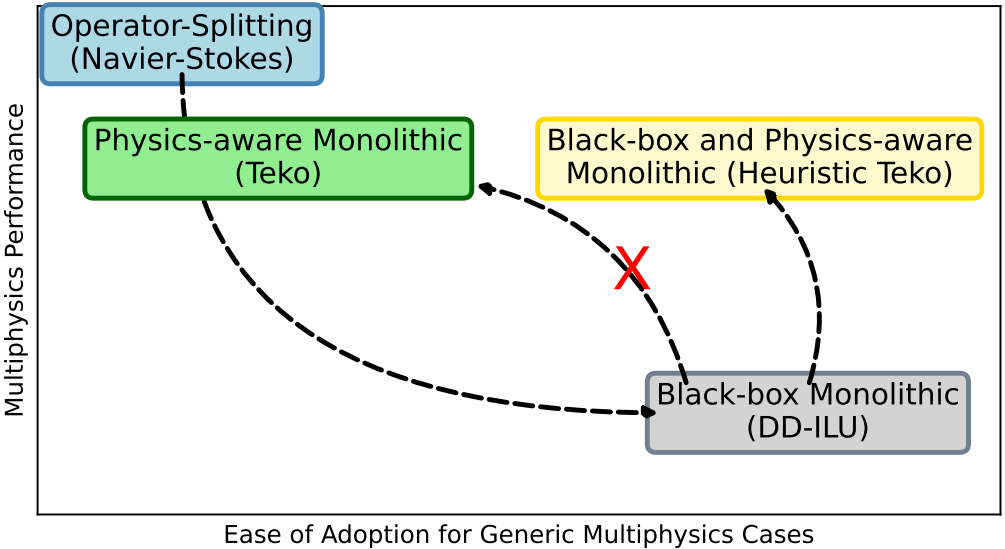
- 'Physics-aware' block approach (Trilinos/Teko) works well for multi-physics problems
- Why not always use a 'physics-aware' preconditioner?
  - Even limiting to *block Gauss-Seidel*, requires:

Ingredients for 'Physics-aware' Solver Setup

1. Physics-to-sub-block mapping
2. Ordering sub-blocks
3. Solvers/preconditioners for each sub-block

- *Goal:* provide 'physics-aware' performance with 'black-box' interface:

```
begin tpetra equation solver teko_linear_solver
  begin preset solver
     solver type = teko_multiphysics
  end preset solver
end tpetra equation solver
```


Operator-Splitting
(Navier-Stokes)
Physics-aware Monolithic
(Teko)
Black-box and Physics-aware
Monolithic (Heuristic Teko)
Black-box Monolithic
(DD-ILU)
Multiphysics Performance
Ease of Adoption for Generic Multiphysics Cases

- Example user-app interaction with 'black-box'-like interface:

```
// based on driver-heuristic-permutation.cpp
using Teko::TpetraHelpers::BlockedTpetraOperator;

// vector<vector<GO>> with (rank-local) GOs for each physics
auto A_b = make_rcp<BlockedTpetraOperator>(dof_gids, crsMat);

// Run alg, generate gids vector<vector<GO>> for (merged) physics
auto [permutation, score] = Teko::generate_heuristic_permutation(A_b);
auto gids = Teko::construct_block_gids_from_permutation(permutation,
    dof_gids);
A_b = make_rcp<BlockedTpetraOperator>(gids, crsMat);

// Generate new parameters from permutation
RCP<ParameterList> xmlList = Teko::
    generate_parameters_from_permutation(permutation, "TekoPrec");
```

- *How do we pick the right grouping/ordering for block Gauss-Seidel?*

Ordering sub-blocks:

$$\kappa\left(\underbrace{\begin{bmatrix} A & B \\ & D \end{bmatrix}^{-1}}_{\tilde{\mathcal{M}}[\mathcal{A}]^{-1}}\underbrace{\begin{bmatrix} A & B \\ C & D \end{bmatrix}}_{\mathcal{A}}\right) \neq \kappa\left(\underbrace{\begin{bmatrix} D & C \\ & A \end{bmatrix}^{-1}}_{\tilde{\mathcal{M}}[\mathcal{R}\mathcal{A}\mathcal{R}^T]^{-1}}\underbrace{\begin{bmatrix} D & C \\ B & A \end{bmatrix}}_{\mathcal{R}\mathcal{A}\mathcal{R}^T}\right)$$

Combinatoric Minimization Problem

Find block numbering permutation $\mathcal{R}^*$ such that

$$\mathcal{R}^*(\mathcal{A}) = \arg\min_{\forall \mathcal{R}} \left\| \tilde{\mathcal{M}}\left[\mathcal{R}\mathcal{A}\mathcal{R}^T\right] - \mathcal{R}\mathcal{A}\mathcal{R}^T \right\|_F^2 .$$

- Isomorphic to NP-hard linear ordering problem (`LOP`) in operations research
- Use available branch-and-bound solver with Lagrangian relaxation[1]

[1] Charon and Hudry, "A branch-and-bound algorithm to solve the linear ordering problem for weighted tournaments".

Physics-to-sub-block mapping:

- Strongly coupled physics may require monolithic approach:

$$\kappa\left(\underbrace{\begin{bmatrix} A & B & C \\ & E & F \\ & & J \end{bmatrix}^{-1}}_{\tilde{\mathcal{M}}[\mathcal{A}]^{-1}}\underbrace{\begin{bmatrix} A & B & C \\ D & E & F \\ G & H & J \end{bmatrix}}_{\mathcal{A}}\right) \neq \kappa\left(\underbrace{\begin{bmatrix} \begin{bmatrix} A & B \\ D & E \end{bmatrix} & \begin{bmatrix} C \\ F \end{bmatrix} \\ & J \end{bmatrix}^{-1}}_{\tilde{\mathcal{M}}[\mathcal{A}']^{-1}}\underbrace{\begin{bmatrix} \begin{bmatrix} A & B \\ D & E \end{bmatrix} & \begin{bmatrix} C \\ F \end{bmatrix} \\ \begin{bmatrix} G & H \end{bmatrix} & J \end{bmatrix}}_{\mathcal{A}'}\right)$$

Combinatoric Minimin Problem

Find the minimum number of block merging operations such that

$$\left\| \tilde{\mathcal{M}}\left[\mathcal{R}^*\mathcal{A}'\mathcal{R}^*\right] - \mathcal{R}^*\mathcal{A}'\mathcal{R}^* \right\|_F \leq \tau,$$

where $\mathcal{R}^* = \mathcal{R}^*\left[A'\right]$ is the solution to the linear ordering problem and τ represents a user-defined target error reduction.

Physics-to-sub-block mapping:

- Use greedy heuristic algorithm, combined with `LOP` solver:

Algorithm Greedy Block Merging Algorithm

Require: Matrix $\mathcal{A}$ with n_b blocks, user-provided threshold τ.

Ensure: Re-grouped matrix $\mathcal{A}'$ with $n_b' < n_b$ blocks.

1: $\mathcal{A}^{(0)} \leftarrow \texttt{LOP}(\mathcal{A})$

2: $k \leftarrow 0$

3: **while** $\left\| \tilde{\mathcal{M}}\left[\mathcal{A}^{(k)}\right] - \mathcal{A}^{(k)} \right\|_F \geq \tau$ **do**

4: $\quad k \leftarrow k + 1$

5: $\quad (i^*, j^*) \leftarrow \underset{i,j,i>j}{\arg\max} \left\| \mathcal{A}^{(k-1)} \right\|$

6: $\quad \mathcal{A}^{(k)} \leftarrow \texttt{LOP}(\texttt{combineBlocks}(\mathcal{A}^{(k-1)}, i^*, j^*))$

7: **end while**

8: **return** $\mathcal{A}^{(k)}$

- *Future work:* Develop branch-and-bound bootstrapped by `LOP`

- Ordering/grouping solved – let's revisit the user app:

```
// based on driver-heuristic-permutation.cpp
using Teko::TpetraHelpers::BlockedTpetraOperator;

// vector<vector<GO>> with (rank-local) GOs for each physics
auto A_b = make_rcp<BlockedTpetraOperator>(dof_gids, crsMat);

// Run alg, generate gids vector<vector<GO>> for (merged) physics
auto [permutation, score] = Teko::generate_heuristic_permutation(A_b);
auto gids = Teko::construct_block_gids_from_permutation(permutation,
    dof_gids);
A_b = make_rcp<BlockedTpetraOperator>(gids, crsMat);

// Generate new parameters from permutation
RCP<ParameterList> xmlList =
    Teko::generate_parameters_from_permutation(permutation, "TekoPrec");
```

- *How do we pick the solvers for each of the sub-blocks?*

Sub-block solver choice:

$$\kappa\left(\underbrace{\begin{bmatrix} A & B \\ & D \end{bmatrix}^{-1}}_{\tilde{\mathcal{M}}[\mathcal{A}]^{-1}}\underbrace{\begin{bmatrix} A & B \\ C & D \end{bmatrix}}_{\mathcal{A}}\right) \neq \kappa\left(\underbrace{\begin{bmatrix} M_A & B \\ & M_D \end{bmatrix}^{-1}}_{\mathcal{M}[\mathcal{A}^{-1}]}\underbrace{\begin{bmatrix} A & B \\ C & D \end{bmatrix}}_{\mathcal{A}}\right)$$

Convergence Result

A single resistant sub-block solver can derail the entire solver:

$$\begin{aligned}\kappa\Big(\mathcal{M}\left[\mathcal{A}\right]^{-1}\mathcal{A}\Big) &\leq \kappa\Big(\mathcal{M}\left[\mathcal{A}\right]^{-1}\tilde{\mathcal{M}}\left[\mathcal{A}\right]\Big)\cdot\kappa\Big(\tilde{\mathcal{M}}\left[\mathcal{A}\right]^{-1}\mathcal{A}\Big)\\ &\geq \underbrace{\max_{\forall M_{ii}^{-1}A_{ii}}\left(\kappa\big(M_{ii}^{-1}A_{ii}\big)\right)}_{\text{Sub-block Solver Conditioning}}\cdot\underbrace{\kappa\Big(\tilde{\mathcal{M}}\left[\mathcal{A}\right]^{-1}\mathcal{A}\Big)}_{\text{Multi-physics Coupling}}\end{aligned}$$

Sub-block solver choice:

- Use adaptive sub-block solver with schedule, e.g.:
  1. GMRES(30), Jacobi
  2. GMRES(30), DD(0)-ILU(0)
  3. GMRES(30), DD(1)-ILU(1)
  4. GMRES(30), DD(2)-ILU(2)
  5. Sparse direct solver
- Estimate preconditioner quality through residual reduction

$$\frac{\left\|\underline{b} - AM_A^{-1}\underline{b}\right\|_2}{\|\underline{b}\|_2} > \epsilon$$

- Move up the schedule to progressively more robust/expensive solvers
- User able to provide custom settings for merged and unmerged blocks
  - e.g., AMG continuity/pressure solve
- *Future work:* employ block AMG for merged blocks

Performance Results:

Table: Ablation performance test, one node on Amber.

Preconditioner	**Linear Solver Time (s)**	**Overall Simulation Time (s)**
DD(1)-ILU(1)	121.747	224
Teko, Hand Chosen	61.359	154
Teko Preset	32.584 **(3.74x Speedup)**	138 **(1.62x Speedup)**

Table: Coupled chemistry performance test, two nodes on Amber.

Preconditioner	**Linear Solver Time (s)**	**Overall Simulation Time (s)**
DD(0)-ILU(0)	515.323	714
Teko Preset	226.093 **(2.28x Speedup)**	353 **(2.02x Speedup)**

Table: End-to-end battery simulation, 20 ranks on local workstation

Preconditioner	**Linear Solver Time (s)**	**Overall Simulation Time (s)**
Amesos2/KLU2	5934.024	7305
Teko, Hand Chosen	957.081	4391
Teko Preset	693.531 **(8.56x Speedup)**	3872 **(1.89x Speedup)**

Table: Porous flow case, 'in-the-wild' user-case, two nodes on Amber.

Preconditioner	**Linear Solver Time (s)**	**Overall Simulation Time (s)**
DD(0)-ILU(0)	1908.776	2532
Teko Preset	223.608 **(8.54x Speedup)**	785 **(3.23x Speedup)**

Concluding remarks:

- 'Physics-aware' Teko preconditioners can outperform 'black-box' preconditioners (DD-ILU)
  - 'Physics-aware' preconditioners require user expertise, difficult to set-up
- Introduced heuristic algorithms based on minimizing $\|\mathcal{M}[\mathcal{A}] - \mathcal{A}\|_F$
  - Leverage pre-existing algorithms for discrete optimization problems
  - Enable application user/developer to bootstrap settings
  - *Requires some modification in user application*
    - *Future work:* drive everything through XML
- Performance results indicate heuristic approach is *reasonable*
  - Comparable or better performance to hand-selected settings
  - Additional input from user (e.g., MueLu setting for known pressure blocks)
- *Almost* available in Trilinos/develop with
  `https://github.com/trilinos/Trilinos/pull/14036`
- Pre-print underway, not yet available