Improving the Performance of the GMRES method using Mixed Precision Techniques

Neil Lindquist, Piotr Luszczek, Jack Dongarra

Smoky Mountains Conference

August 27th, 2020

GMRES

- General purpose, sparse linear solver
  - Iterative, Krylov solver
- Memory bound performance
  - Mix single and double precision

GMRES Algorithm

GMRES$_{res}(\boldsymbol{A}, \boldsymbol{x_0}, \boldsymbol{b}, \boldsymbol{M}^{-1})$ — Computing $\boldsymbol{Ax} = \boldsymbol{b}$. $\boldsymbol{A}^{-1} \approx \boldsymbol{M}^{-1}$

for $k = 0, 1, 2, \ldots$ — Restarts

$\boldsymbol{r_k} \leftarrow \boldsymbol{b} - \boldsymbol{Ax_k}$

$\boldsymbol{z_k} \leftarrow \boldsymbol{M}^{-1}\boldsymbol{r_k}$

$\beta \leftarrow \|\boldsymbol{z_k}\|_2$

$\boldsymbol{V}_{:,0} \leftarrow \boldsymbol{z_k}/\beta$

$\boldsymbol{s} \leftarrow [\beta, 0, 0, \ldots, 0]^T$

for j $= 0, 1, 2, \ldots$ — Iteration count

$\boldsymbol{w} \leftarrow \boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{V}_{:,j}$

$\boldsymbol{w}, \boldsymbol{H}_{:,j} \leftarrow orthogonalize(\boldsymbol{w}, \boldsymbol{V}_{:,j})$

$\boldsymbol{H}_{j+1,j} \leftarrow \|\boldsymbol{w}\|_2$

$\boldsymbol{V}_{:,j+1} \leftarrow \boldsymbol{w}/\|\boldsymbol{w}\|_2$

$\boldsymbol{H}_{:,j} \leftarrow \boldsymbol{G_0}\boldsymbol{G_1} \ldots \boldsymbol{G_{j-1}}\boldsymbol{H}_{:,j}$

$\boldsymbol{G_j} \leftarrow rotation\_matrix(\boldsymbol{H}_{:,j})$

$\boldsymbol{H}_{:,j} \leftarrow \boldsymbol{G_j}\boldsymbol{H}_{:,j}$

$\boldsymbol{s} \leftarrow \boldsymbol{G_j}\boldsymbol{s}$

$\boldsymbol{u_k} \leftarrow \boldsymbol{V}\boldsymbol{H}^{-1}\boldsymbol{s}$

$\boldsymbol{x_{k+1}} \leftarrow \boldsymbol{x_k} + \boldsymbol{u_k}$

GMRES Algorithm

$\text{GMRES}_{res}(\boldsymbol{A}, \boldsymbol{x_0}, \boldsymbol{b}, \boldsymbol{M}^{-1})$

Computing $\boldsymbol{Ax} = \boldsymbol{b}$. $\boldsymbol{A}^{-1} \approx \boldsymbol{M}^{-1}$

for $k = 0, 1, 2, \ldots$

Restarts

Double: $\boldsymbol{r_k} \leftarrow \boldsymbol{b} - \boldsymbol{Ax_k}$

$\boldsymbol{z_k} \leftarrow \boldsymbol{M}^{-1}\boldsymbol{r_k}$

$\beta \leftarrow \|\boldsymbol{z_k}\|_2$

$\boldsymbol{V}_{:,0} \leftarrow \boldsymbol{z_k}/\beta$

$\boldsymbol{s} \leftarrow [\beta, 0, 0, \ldots, 0]^T$

for j $= 0, 1, 2, \ldots$

Iteration count

$\boldsymbol{w} \leftarrow \boldsymbol{M}^{-1}\boldsymbol{AV}_{:,j}$

$\boldsymbol{w}, \boldsymbol{H}_{:,j} \leftarrow orthogonalize(\boldsymbol{w}, \boldsymbol{V}_{:,j})$

Single: $\boldsymbol{H}_{j+1,j} \leftarrow \|\boldsymbol{w}\|_2$

$\boldsymbol{V}_{:,j+1} \leftarrow \boldsymbol{w}/\|\boldsymbol{w}\|_2$

$\boldsymbol{H}_{:,j} \leftarrow \boldsymbol{G_0}\boldsymbol{G_1} \ldots \boldsymbol{G_{j-1}}\boldsymbol{H}_{:,j}$

$\boldsymbol{G_j} \leftarrow rotation\_matrix(\boldsymbol{H}_{:,j})$

$\boldsymbol{H}_{:,j} \leftarrow \boldsymbol{G_j}\boldsymbol{H}_{:,j}$

$\boldsymbol{s} \leftarrow \boldsymbol{G_j}\boldsymbol{s}$

$\boldsymbol{u_k} \leftarrow \boldsymbol{VH}^{-1}\boldsymbol{s}$

Double: $\boldsymbol{x_{k+1}} \leftarrow \boldsymbol{x_k} + \boldsymbol{u_k}$

Effect on Memory Allocation

- Double: $8kn + 12n_{nz} + O(n + k^2)$ bytes
- Mixed: $4kn + 16n_{nz} + O(n + k^2)$ bytes
  - Including A, x, b
  - Excluding preconditioner
  - At most k inner iterations per restart
  - CSR matrix format

GMRES Algorithm

Double:

Single:

Double:

$\text{GMRES}_{res}(\boldsymbol{A}, \boldsymbol{x_0}, \boldsymbol{b}, \boldsymbol{M}^{-1})$

for $k = 0, 1, 2, \ldots$

$\boldsymbol{r_k} \leftarrow \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x_k}$

$\boldsymbol{z_k} \leftarrow \boldsymbol{M}^{-1}\boldsymbol{r_k}$

$\beta \leftarrow \|\boldsymbol{z_k}\|_2$

$\boldsymbol{V}_{:,0} \leftarrow \boldsymbol{z_k}/\beta$

$\boldsymbol{s} \leftarrow [\beta, 0, 0, \ldots, 0]^T$

for j $= 0, 1, 2, \ldots$

$\boldsymbol{w} \leftarrow \boldsymbol{M}^{-1}\boldsymbol{A}\boldsymbol{V}_{:,j}$

$\boldsymbol{w}, \boldsymbol{H}_{:,j} \leftarrow orthogonalize(\boldsymbol{w}, \boldsymbol{V}_{:,j})$

$\boldsymbol{H}_{j+1,j} \leftarrow \|\boldsymbol{w}\|_2$

$\boldsymbol{V}_{:,j+1} \leftarrow \boldsymbol{w}/\|\boldsymbol{w}\|_2$

$\boldsymbol{H}_{:,j} \leftarrow \boldsymbol{G_0}\boldsymbol{G_1} \ldots \boldsymbol{G_{j-1}}\boldsymbol{H}_{:,j}$

$\boldsymbol{G_j} \leftarrow rotation\_matrix(\boldsymbol{H}_{:,j})$

$\boldsymbol{H}_{:,j} \leftarrow \boldsymbol{G_j}\boldsymbol{H}_{:,j}$

$\boldsymbol{s} \leftarrow \boldsymbol{G_j}\boldsymbol{s}$

$\boldsymbol{u_k} \leftarrow \boldsymbol{V}\boldsymbol{H}^{-1}\boldsymbol{s}$

$\boldsymbol{x_{k+1}} \leftarrow \boldsymbol{x_k} + \boldsymbol{u_k}$

Computing $\boldsymbol{Ax} = \boldsymbol{b}$. $\boldsymbol{A}^{-1} \approx \boldsymbol{M}^{-1}$

Restarts

Iteration count

GMRES Simplified Algorithm

$$\text{GMRES}_{res}(\boldsymbol{A}, \boldsymbol{x_0}, \boldsymbol{b}, \boldsymbol{M}^{-1})$$

$$\text{for} \;\; k = 0, 1, 2, \ldots$$

Double: $\boldsymbol{r_k} \leftarrow \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x_k}$

Single: $\boxed{\boldsymbol{u_k} \leftarrow \text{GMRES}_{no\;res}(\boldsymbol{A}, \vec{\boldsymbol{0}}, \boldsymbol{r_k}, \boldsymbol{M}^{-1})}$

Double: $\boldsymbol{x_{k+1}} \leftarrow \boldsymbol{x_k} + \boldsymbol{u_k}$

GMRES Simplified Algorithm

$$\text{GMRES}_{res}(\boldsymbol{A}, \boldsymbol{x_0}, \boldsymbol{b}, \boldsymbol{M}^{-1})$$

for $k = 0, 1, 2, \ldots$

Double: $\boldsymbol{r_k} \leftarrow \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x_k}$

Single: $\boxed{\boldsymbol{u_k} \leftarrow \boldsymbol{A}^{-1}\, \boldsymbol{r_k}}$

Double: $\boldsymbol{x_{k+1}} \leftarrow \boldsymbol{x_k} + \boldsymbol{u_k}$

Restart Strategies

1. Fixed # iterations

Restart Strategies

1. Fixed # iterations
2. Approximation of residual norm
    1. Threshold
    2. Improvement stops

Restart Strategies

1. Fixed # iterations
2. Approximation of residual norm
    1. Threshold
    2. Improvement stops
3. Detect basis losing independence

Restart Strategies

1. Fixed # iterations
2. Approximation of residual norm
    1. Threshold
    2. Improvement stops
3. Detect basis losing independence
4. # iterations for first restart

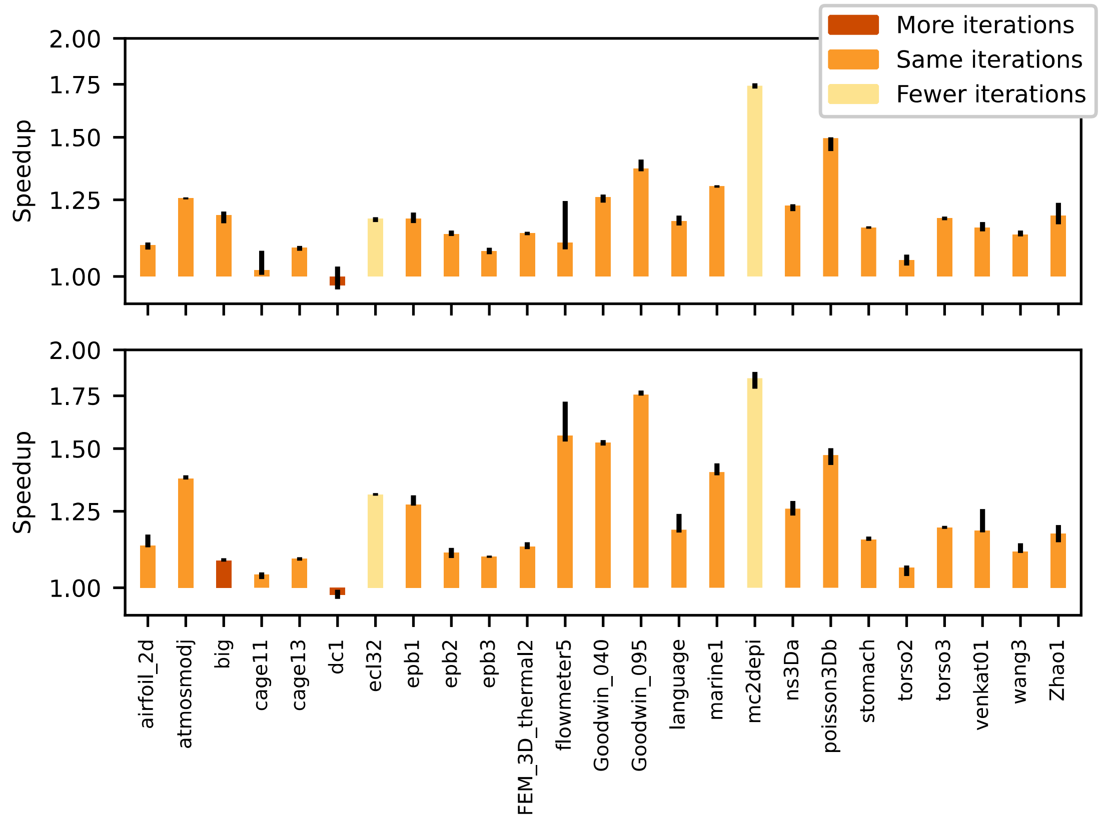
Effect on Performance: Configuration

- ILU(0) preconditioner (M^{-1})
- CSR matrix format
- KokkosKernels with Intel MKL
- 20-core Haswell node
  - 2x Intel® Xeon® E5-2650 v3 processors

Performance - Restarted

- Target accuracy $10^{-10} = \frac{\|b - Ax\|_2}{\|A\|_F \|x\|_2 + \|b\|_2}$
- Restart: half the iterations needed for double-precision implementation

Performance - Restarted

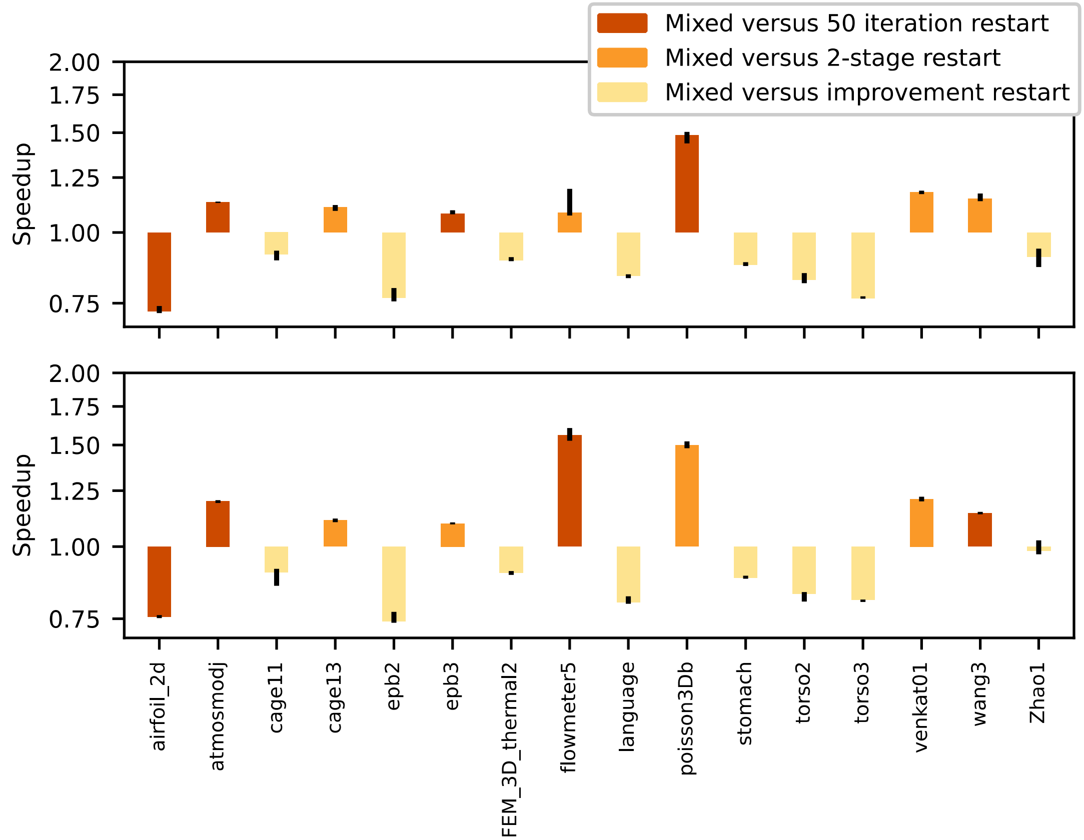


- Speedups of median times
  - 5 runs
  - Error bars: mins and maxes
- Geometric mean of speedup
  - MGS: 19%
  - CGSR: 24%

Performance – Non-restarted Option

- Target accuracy: $10^{-10} = \frac{\|b - Ax\|_2}{\|A\|_F \|x\|_2 + \|b\|_2}$
- Matrices restarted ≤ 50 iterations in last test
- Restart: 50 iterations plus one of:
  1. Residual improved by 10^{-6} for 1^{st} restart, then same # iterations
  2. Residual improved by 10^{-8}
     - Non-restarted
     - only baseline
  3. Nothing
     - only baseline

Performance – Non-restarted Option



- Speedups of median times
  - 5 runs
  - Error bars: mins and maxs
- Geometric mean of speedup
  - MGS: -4%
  - CGSR: 0%
- Speedup ⇔ baseline restarted

Future Directions

- Choice of low-precision
  - Half, Bfloat16, integers
  - Compression
- Hardware
  - GPUs
  - Distributed
- Algorithms
  - Communication hiding/avoiding

Conclusions

- With restarts, mixed-precision GMRES
  - Converges to double-precision accuracy
  - Outperforms restarted, double-precision GMRES

Extra Slides

Effect on Convergence: Configuration

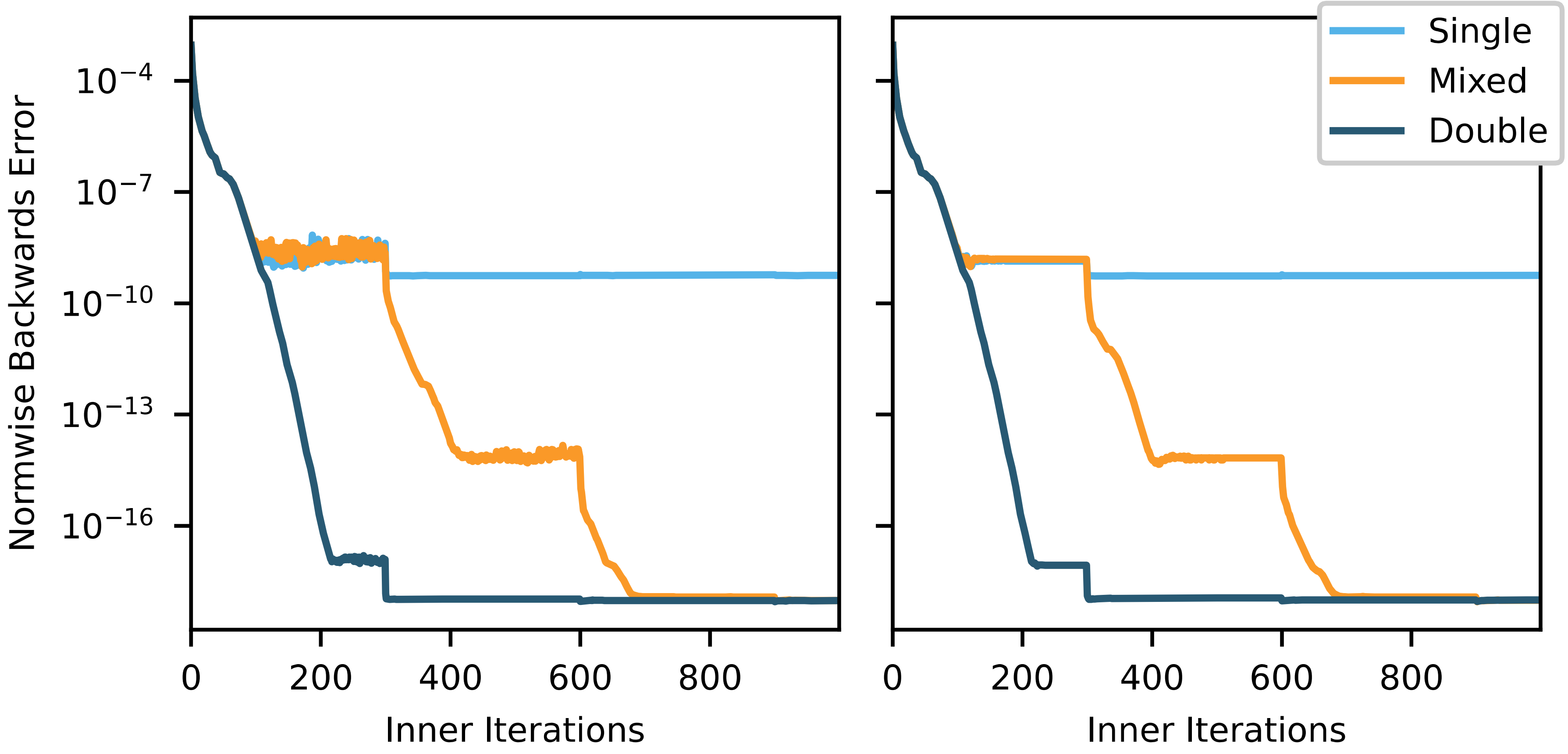
- ILU(0) preconditioner (M^{-1})
- CSR matrix format
- Custom, mixed precision kernels w/ Kokkos
- 20-core Haswell node
  - 2x Intel® Xeon® E5-2650 v3 processors
- 2 orthogonalization schemes
  - Modified Gram-Schmidt (MGS)
  - Classical Gram-Schmidt with Reorthogonalization (CGSR)

Effect on Convergence: Configuration

- airfoil_2d from SuiteSparse collection
  - $n = 14{,}214$
  - $nnz = 259{,}688$
  - $\kappa_2 = 1.8 \cdot 10^6$
- Error if GMRES stopped

$$\frac{\|b - Ax\|_2}{\|A\|_F \|x\|_2 + \|b\|_2}$$

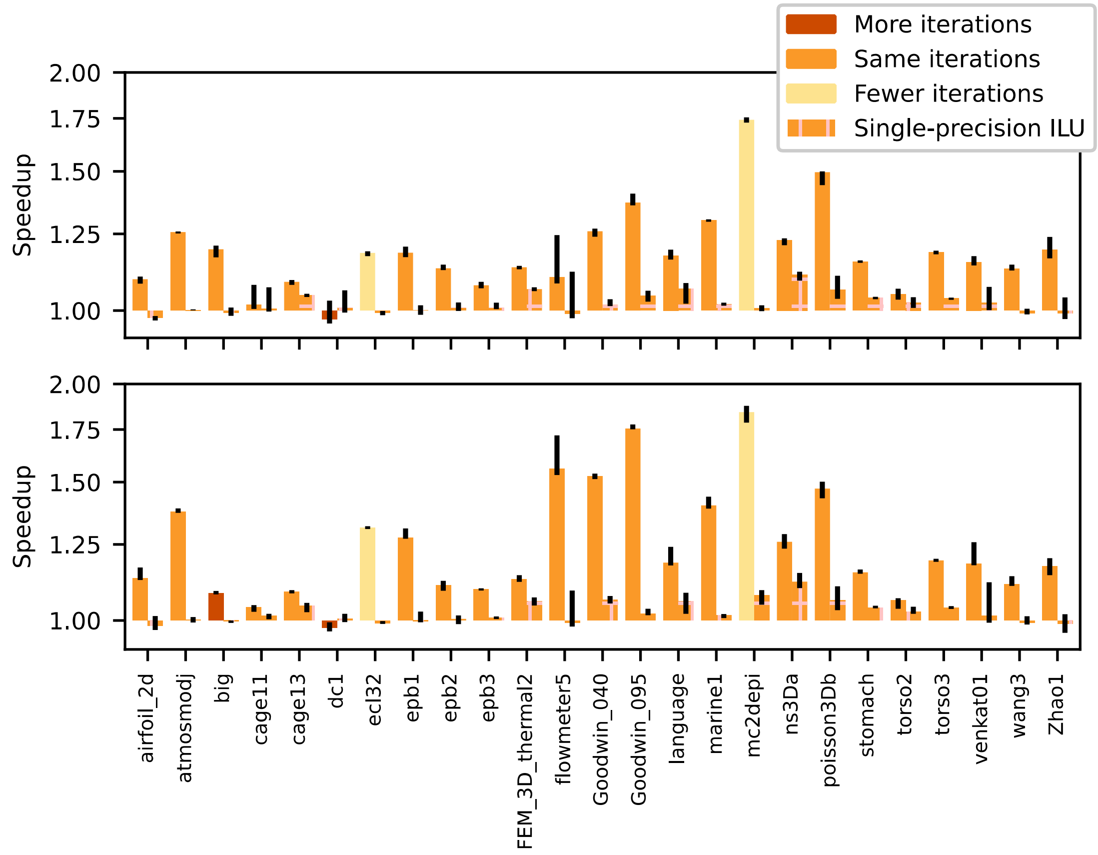
Accuracy results



Modified Gram-Schmidt Orthogonalization (MGS)

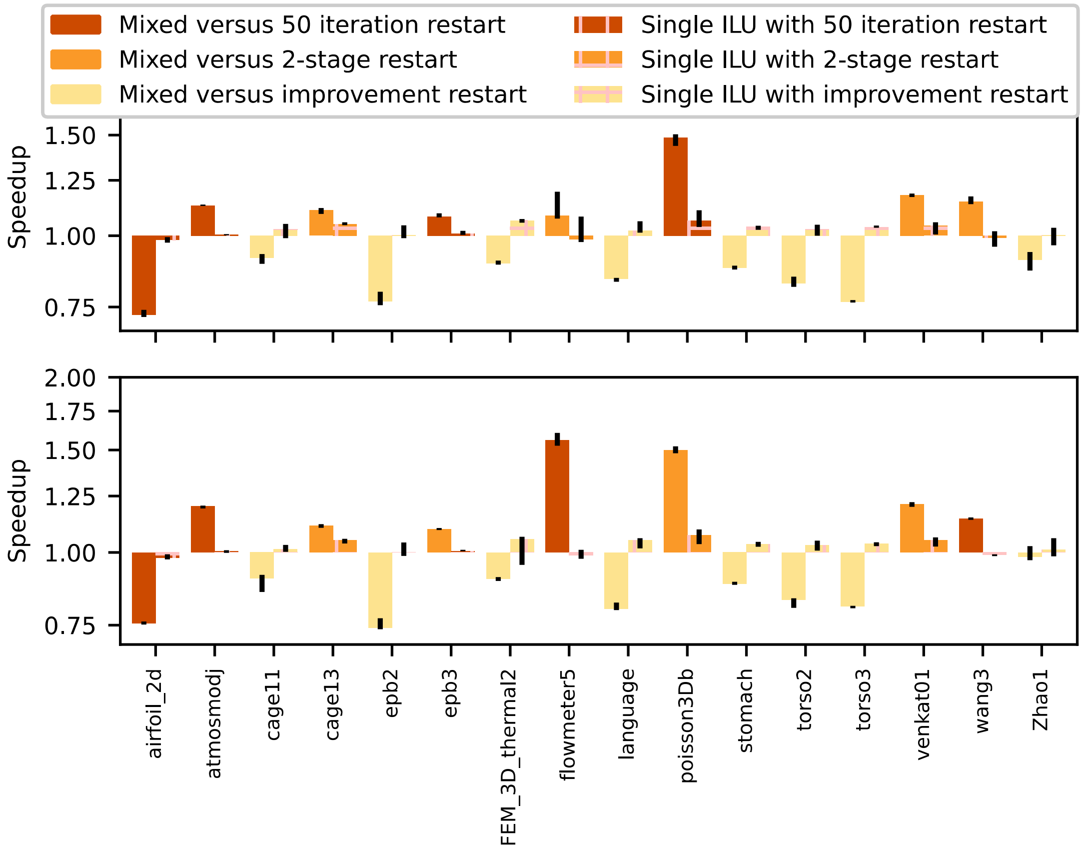
Classical Gram-Schmidt with Reorthogonalization (CGSR)

Performance - Restarted



- Speedups of median times for 5 runs
  - Error bars for minimums and maximums
- Geometric mean of speedup
  - MGS: 19%
  - CGSR: 24%
  - Single-precision preconditioner: 2%

Performance - Restarted



- Speedups of median times for 5 runs
  - Error bars for minimums and maximums
- Geometric mean of speedup
  - MGS: -4%
  - CGSR: 0%
  - Single-precision preconditioner:
    - MGS: 2%
    - CGSR: 1%