# Improving the Performance of GMRES with Mixed Precision

**Neil Lindquist**, Piotr Luszczek, Jack Dongarra

SIAM PP20

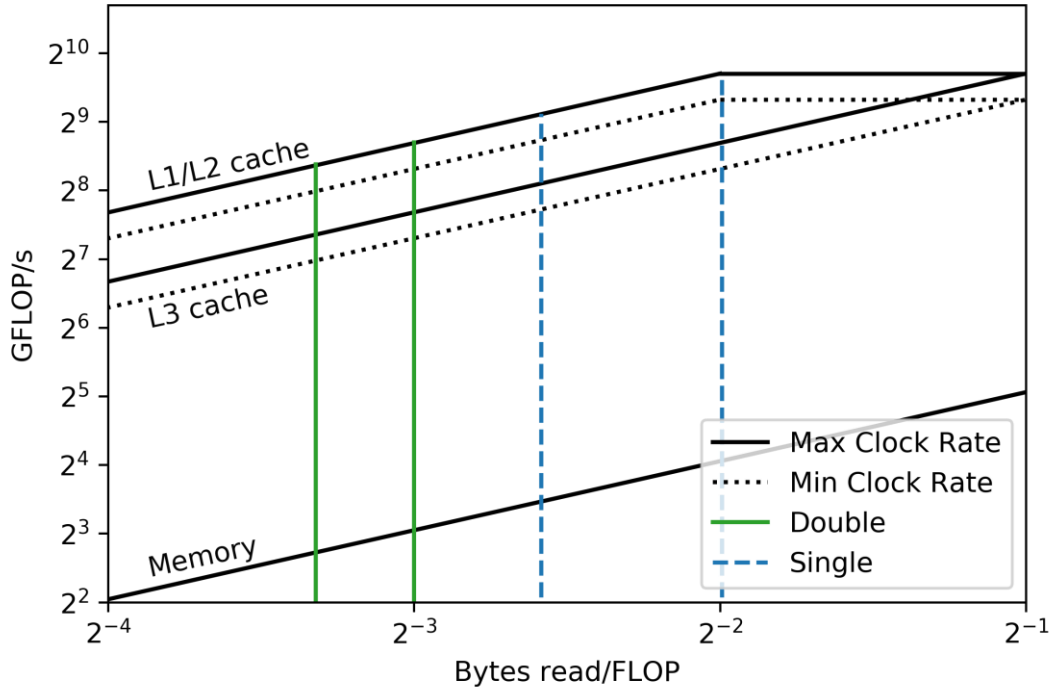
February 13th, 2020

ICL INNOVATIVE COMPUTING LABORATORY

THE UNIVERSITY OF TENNESSEE KNOXVILLE

# GMRES

- General purpose, sparse linear solver
- Memory bound performance






software.intel.com/sites/default/files/managed/9e/bc/64-ia-32-architectures-optimization-manual.pdf
www.intel.com/content/www/us/en/ark/products/81705/intel-xeon-processor-e5-2650-v3-25m-cache-2-30-ghz.html
www.intel.com/content/dam/www/public/us/en/documents/specification-updates/xeon-e5-v3-spec-update.pdf

# GMRES Algorithm

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

for $i = 0, 1, 2, \ldots$ — Restarts

$r_i \leftarrow b - Ax_i$

$w \leftarrow M^{-1} r_i$

$\beta \leftarrow \|w\|_2$

$V_{:,0} \leftarrow w/\beta$

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

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

$w \leftarrow M^{-1} A V_{:,j}$

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

$H_{l,j} \leftarrow w \cdot V_{:,l}$

$w \leftarrow w - H_{l,j} V_{:,l}$

$H_{j+1,j} \leftarrow \|w\|_2$

Let $G_j$ s.t. $0 = \left(G_j H\right)_{j+1,j}$

$H_{:,j} \leftarrow G_0 G_1 \ldots G_{j-1} H_{:,j}$

$H \leftarrow G_j H$

$s \leftarrow G_j s$

If $s_j$ small enough, then break.

$u_i \leftarrow V H^{-1} s$

$x_{i+1} \leftarrow x_i + u_i$

# Computation Requirement

- Each iteration, $j$, requires:
  - $4jn + 4nnz + \Theta(n)$ FLOP
- Restart with $k$ iterations requires:
  - $2k^2n + 4k\ nnz + \Theta(kn + nnz)$ FLOP
  - $8kn + 32n + 8k^2 + \Theta(k)$ bytes

# GMRES Algorithm

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

for $i = 0, 1, 2, \ldots$ — Restarts

$r_i \leftarrow b - Ax_i$

$w \leftarrow M^{-1} r_i$

$\beta \leftarrow \|w\|_2$

$V_{:,0} \leftarrow w/\beta$

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

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

$w \leftarrow M^{-1} A V_{:,j}$

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

$H_{l,j} \leftarrow w \cdot V_{:,l}$

$w \leftarrow w - H_{l,j} V_{:,l}$

$H_{j+1,j} \leftarrow \|w\|_2$

Let $G_j$ s.t. $0 = \left(G_j H\right)_{j+1,j}$

$H_{:,j} \leftarrow G_0 G_1 \ldots G_{j-1} H_{:,j}$

$H \leftarrow G_j H$

$s \leftarrow G_j s$

If $s_j$ small enough, then break.

$u_i \leftarrow V H^{-1} s$

$x_{i+1} \leftarrow x_i + u_i$

# GMRES Algorithm

Double:

Single:

Double:

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

for $i = 0, 1, 2, \ldots$ — Restarts

$r_i \leftarrow b - Ax_i$

$w \leftarrow M^{-1} r_i$

$\beta \leftarrow \|w\|_2$

$V_{:,0} \leftarrow w/\beta$

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

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

$w \leftarrow M^{-1} A V_{:,j}$

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

$H_{l,j} \leftarrow w \cdot V_{:,l}$

$w \leftarrow w - H_{l,j} V_{:,l}$

$H_{j+1,j} \leftarrow \|w\|_2$

Let $G_j$ s.t. $0 = (G_j H)_{j+1,j}$

$H_{:,j} \leftarrow G_0 G_1 \ldots G_{j-1} H_{:,j}$

$H \leftarrow G_j H$

$s \leftarrow G_j s$

If $s_j$ small enough, then break.

$u_i \leftarrow V H^{-1} s$

$x_{i+1} \leftarrow x_i + u_i$

# GMRES Simplified Algorithm

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

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

Double: $r_i \leftarrow b - Ax_i$

Single: $u_i \leftarrow \text{GMRES}_{no\ res}(A, \vec{0}, r_i, M^{-1})$

Double: $x_{i+1} \leftarrow x_i + u_i$

# Effect on Memory Allocation

- Double: $8kn + 12nnz + 32n + 8k^2$ bytes
- Mixed: $4kn + 12nnz + 28n + 4k^2$ bytes
  - Including GMRES internals, $M^{-1}$ =ILU(0)
  - Excluding $A, x, b$
  - At most $k$ inner iterations before restarting

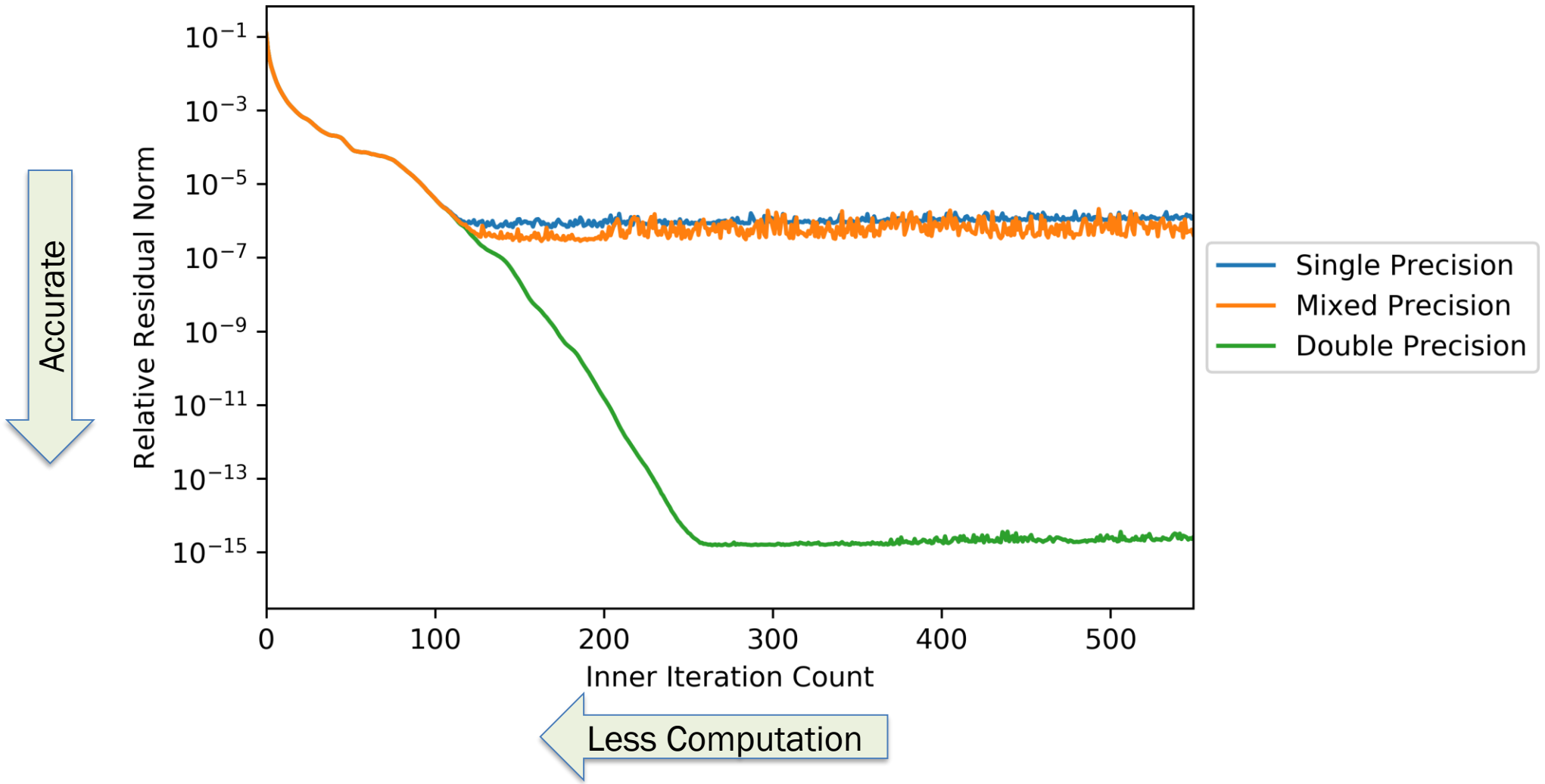
# Effect on Convergence: Setup

- ILU(0) preconditioner ($M^{-1}$)
- CSR matrix format
- Custom, mixed precision kernels
  - Kokkos for storage and parallelism
- A 20-core Haswell node
  - 2x Intel® Xeon® E5-2650 v3 processors

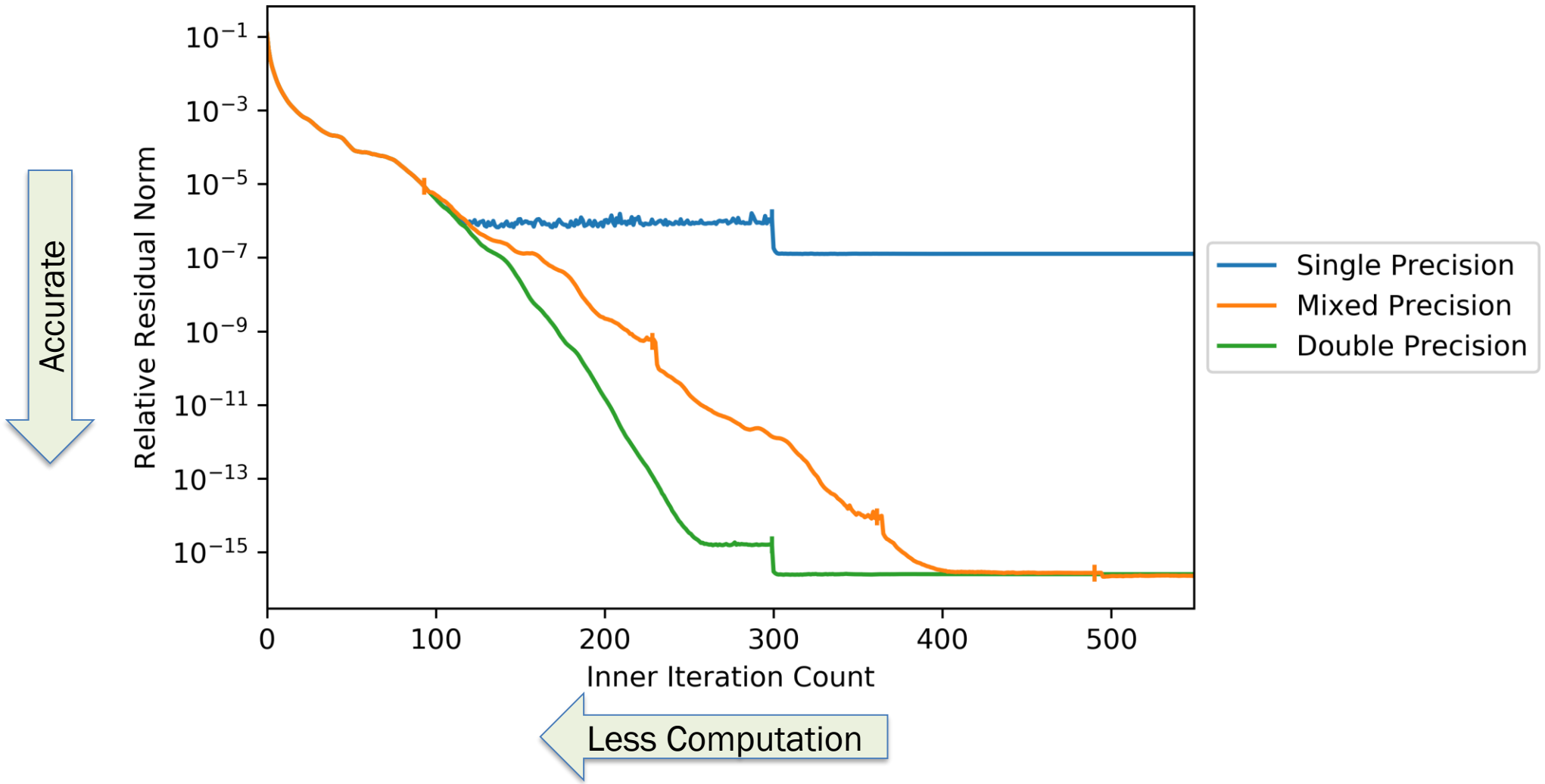
# Effect on Convergence: Setup

- airfoil_2d from SuiteSparse collection
  - $n = 14{,}214$
  - $nnz = 259{,}688$
  - $\kappa_2 = 1.8 \cdot 10^6$
- Plots show residual if GMRES terminated after that iteration

# Effect on Convergence: Without Restarts



# Effect on Convergence: With Restarts



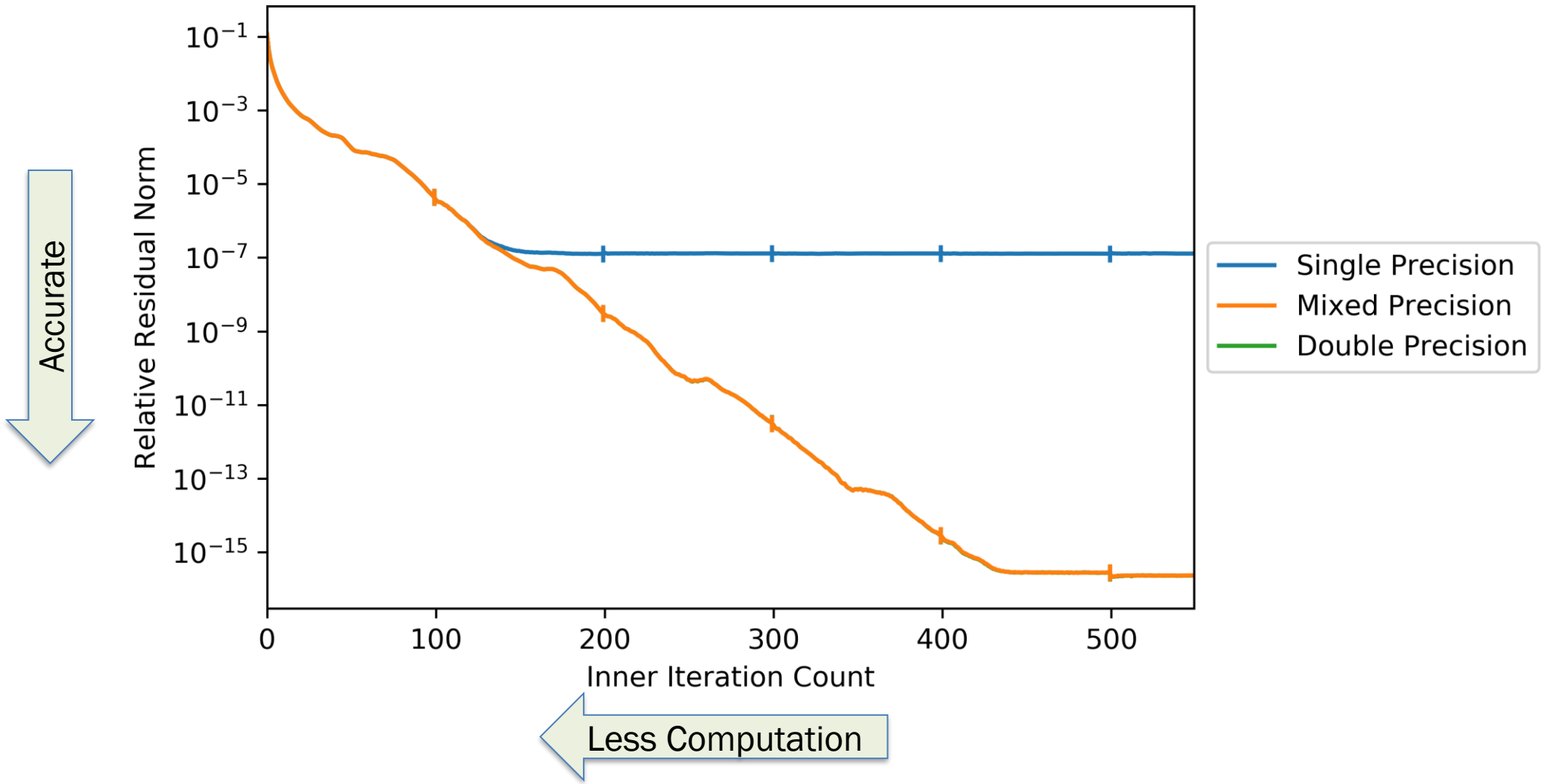
# When to Restart?

- Too few restarts: improvement stalls
- Too many restarts: rate of convergence slows

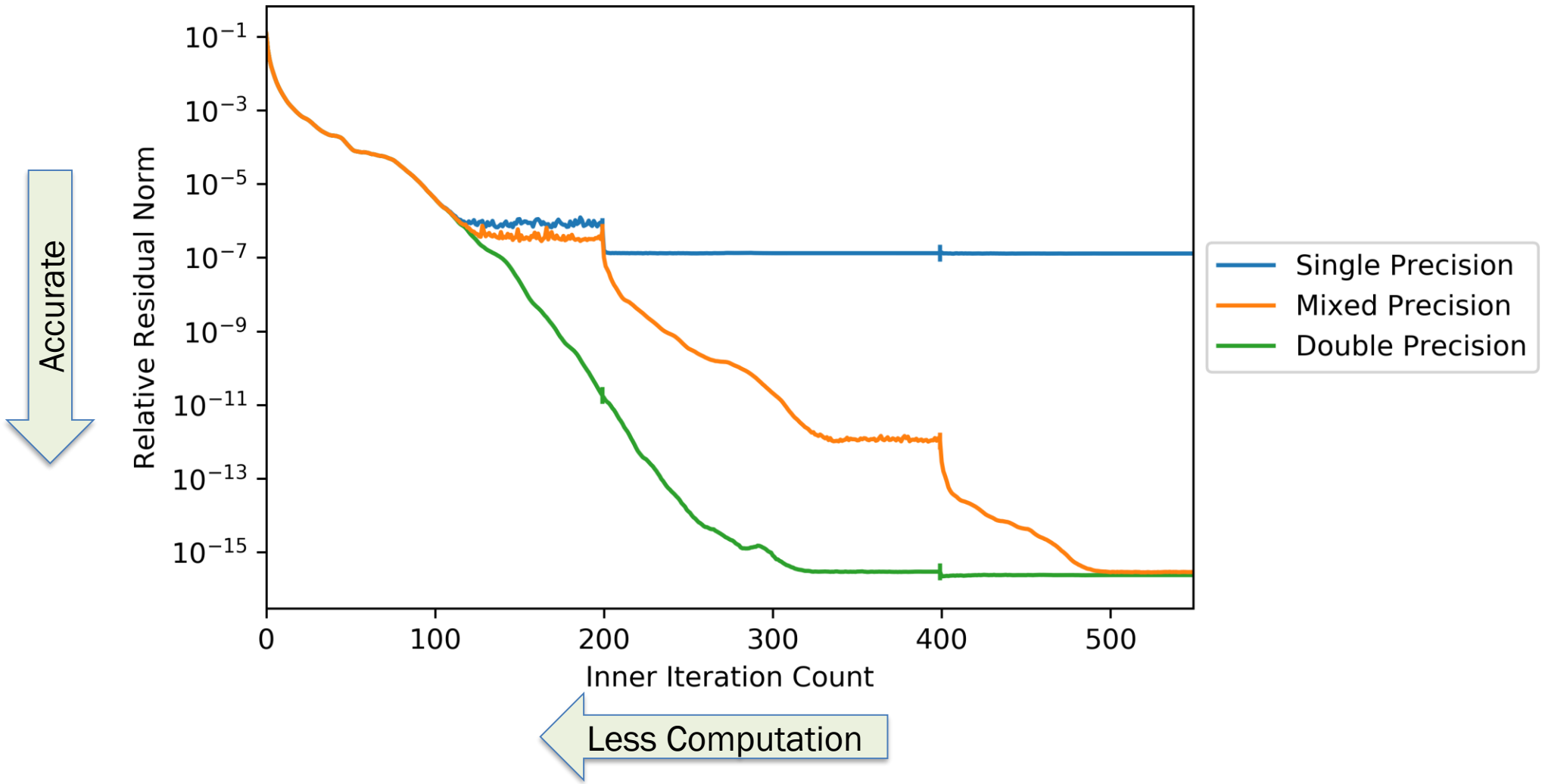
# Possible Restart Strategies

- Fixed iteration count
- Fixed improvement tolerance
- Detecting stalled improvement
  - Change in improvement
  - Versus double precision iteration

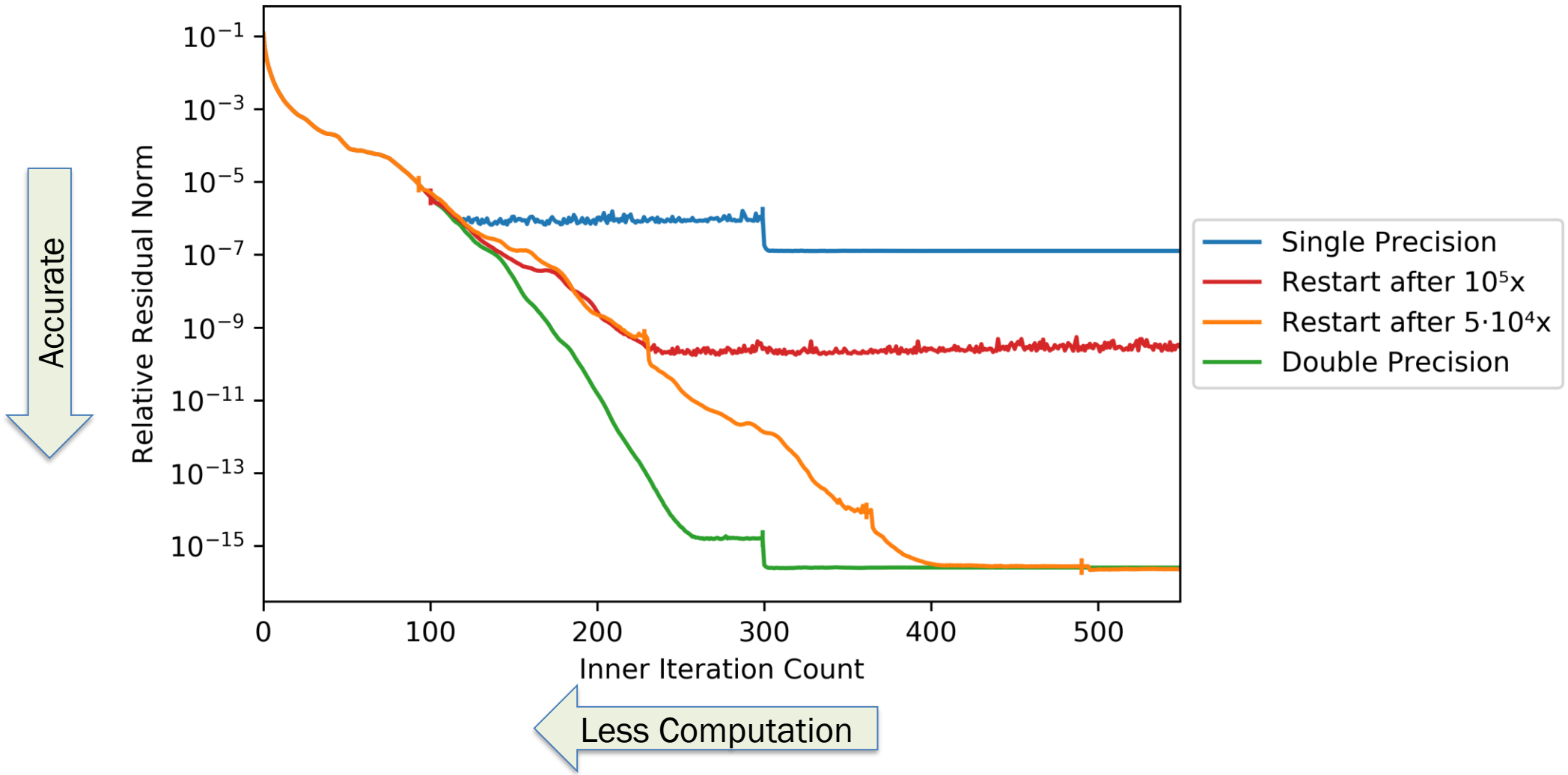
# Restart Strategy: Fixed Iteration (100)



# Restart Strategy: Fixed Iteration (200)



# Restart Strategy: Fixed Improvement



# Performance Results: Setup

- ILU(0) preconditioner ($M^{-1}$)
- CSR matrix format
- KokkosKernels
  - Backed by Intel's MKL
  - Except for preconditioner
- A 20-core Haswell node
  - 2x Intel® Xeon® E5-2650 v3 processors

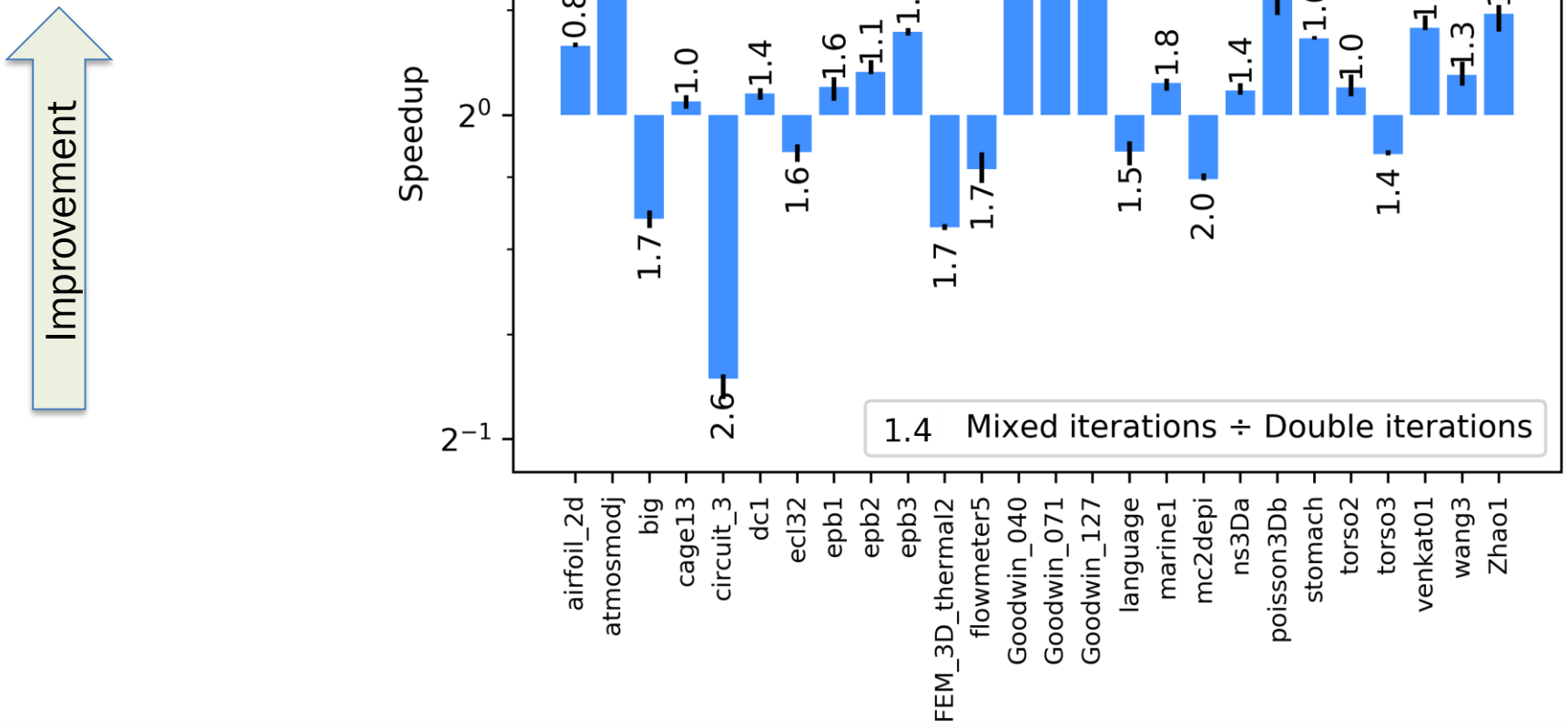
# Performance Results: Setup

- Matrices from SuiteSparse collection
- Solved for preconditioned residual accuracy of $10^{-10}$
- Each trial run 5 times
  - Speedup of medians
  - Error bars for max and min speedup

# Performance Results: Optimal Configuration

- Optimal restart length found for double precision
- Optimal restart length and improvement tolerance found for mixed precision
- Speedup: $\frac{\mathrm{GMRES}_{\mathrm{Double}} + \mathrm{ILU}_{\mathrm{Double}}}{\mathrm{GMRES}_{\mathrm{Mixed}} + \mathrm{ILU}_{\mathrm{Mixed}}}$

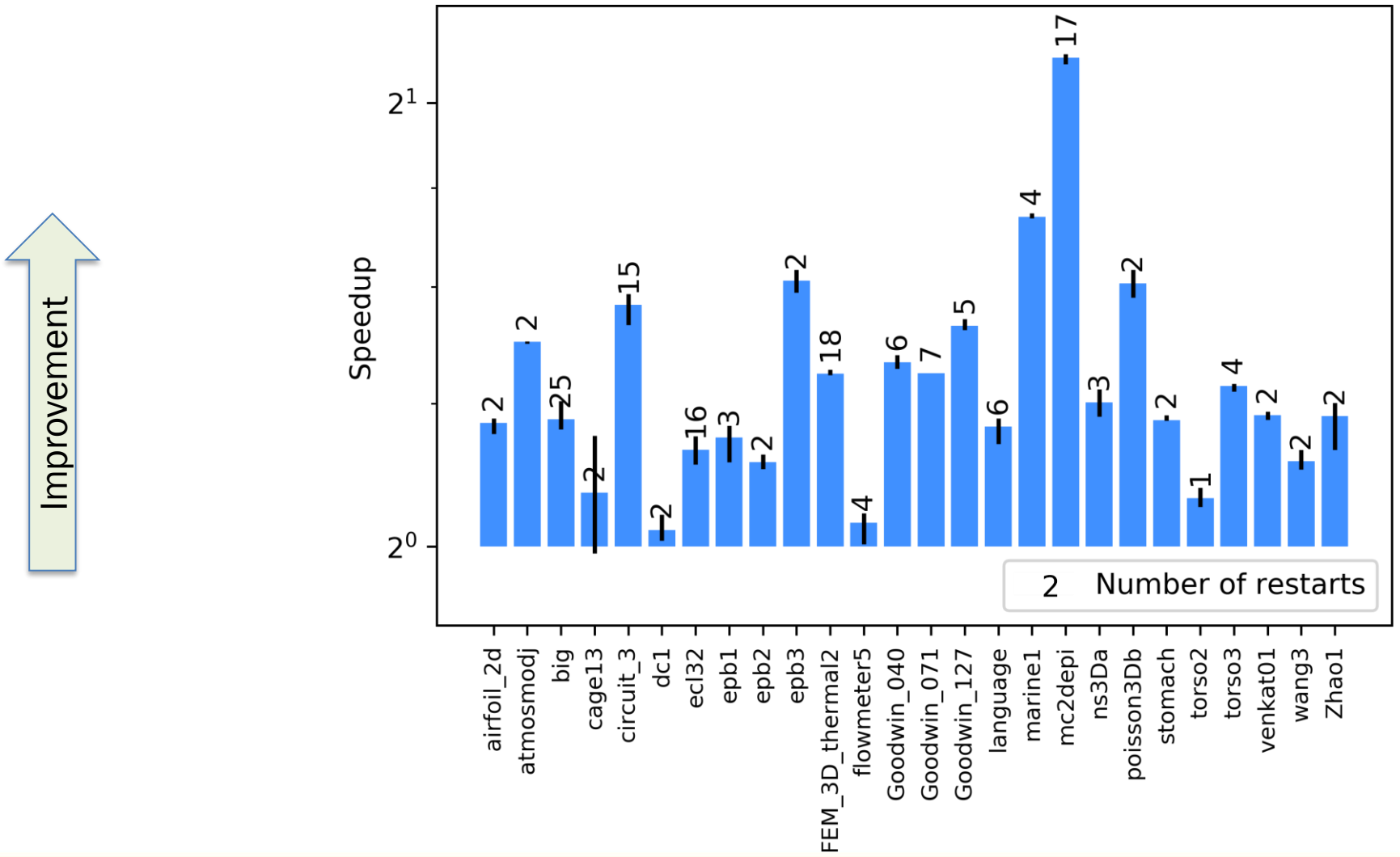
# Timing Results: Optimal Configuration



# Performance Results: Forced Restarts

- $k$ iterations for non-restarting, double precision GMRES to reach $10^{-10}$
- Double and mixed precision were run, restarting every $k/2$ iterations
- Speedup: $\frac{\mathrm{GMRES}_{\mathrm{Double}} + \mathrm{ILU}_{\mathrm{Double}}}{\mathrm{GMRES}_{\mathrm{Mixed}} + \mathrm{ILU}_{\mathrm{Mixed}}}$

# Performance Results: Forced Restarts



# Conclusions

- With appropriate restarts, mixed precision GMRES has
  - double precision accuracy
  - better performance

# Jobs @ ICL

- http://www.icl.utk.edu/jobs
- Research Positions in
  - Numerical Linear Algebra
  - Distributed Computing
  - Performance Measurement and Modeling