High-fidelity simulation of wind-turbine incompressible flows with classical and aggregation AMG preconditioners

SAND2018-3056C

15th Copper Mountain Conference On Iterative Methods

Stephen Thomas[1] Shreyas Ananthan[1] Shashank Yellapantula[1] Jonathan J. Hu[3] Michael A. Sprague[1,2]

[1] High Performance Algorithms and Complex Fluids Group
National Renewable Energy Laboratory, Golden CO

[2] National Wind Technology Center
National Renewable Energy Laboratory, Golden CO

[3] Center for Computing Research
Sandia National Laboratories, Livermore, CA

Copper Mountain, Colorado
March 27, 2018

Outline

- Solve time-dependent incompressible Navier-Stokes equations
- Resolve near blade surface boundary layer and wake vortices
- Compare the two fundamental algebraic-multigrid (AMG) approaches :
  - Smoothed Aggregation AMG (SA-AMG), as implemented in MueLu (with linear solve in Tpetra Belos ICGS-GMRES);
  - Classical Ruge-Stüben AMG (C-AMG), as implemented in Hypre (with linear solve in Hypre's MGS-GMRES);
- Examine ExaWind baseline problem:
  - Vestas V27 wind turbine with sliding mesh interface
- Determine V-cycle complexity C and stencil widths S_{avg}
- Compare GMRES iterations, solver time, set-up time.

EXASCALE COMPUTING PROJECT

Exascale Computational Challenge

Simulation of wake turbulence difficult problem:

- Vestas V27 with 27m turbine diameter. 225 KW
- Matrices are re-initialized every time-step
- Large dynamic scale range and high aspect-ratios
- Sliding mesh adds skew-symmetric terms to non-symmetric matrices
- Ill-conditioned pressure-Poisson problem
- $C = 50$, $\Delta t = 1e-4$s, $\mathcal{O}(10000)$ time steps per revolution. 400 hours on Cori 24K cores
- Peta-scale single to exascale multiple turbines

EXASCALE COMPUTING PROJECT

Incompressible Navier-Stokes Equations

Control Volume Finite Element Method results in block-matrix: saddle-point, indefinite

$$\begin{bmatrix} A & G \\ D & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ b \end{bmatrix}$$

$A = I/\Delta t + \mu L + N$, Laplacian L, viscosity μ and nonlinear convection N.

Consider the block LU factorization of the matrix

$$\begin{bmatrix} A & G \\ D & 0 \end{bmatrix} = \begin{bmatrix} A & 0 \\ D & -DA^{-1}G \end{bmatrix} \begin{bmatrix} I & A^{-1}G \\ 0 & I \end{bmatrix}$$

Inversion of A^{-1} to form the Schur complement matrix $S = -DA^{-1}G$ would be costly.

Matrix splittings (Chorin-Teman, Yosida) employ $\tau = \mathrm{diag}(A)^{-1}$,

$$\int L_1 \phi \, dV = \int \tau_1 \nabla \phi \cdot \mathrm{d}A, \quad \int L_2 \phi \, dV = \int \tau_2 \nabla \phi \cdot \mathrm{d}A$$

$B_1 = -L_1$ is the finite-volume Laplacian for the pressure-Poisson linear system

EXASCALE COMPUTING PROJECT

Incompressible Navier-Stokes Equations

These matrices are introduced into an approximate *LU* block factorization as follows

$$\begin{bmatrix} A & 0 \\ D & B_1 \end{bmatrix} \begin{bmatrix} I & B_2 G \\ 0 & I \end{bmatrix} \begin{bmatrix} u^{n+1} \\ \Delta p^{n+1} \end{bmatrix} + \begin{bmatrix} I & 0 \\ DB_4 & B_3 \end{bmatrix} \begin{bmatrix} I & G \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 \\ p^n \end{bmatrix} = \begin{bmatrix} f \\ b \end{bmatrix}$$

where $B_2 = \tilde{\tau}_2 I$, $B_3 = -L_2$ and $B_4 = \tilde{\tau}_4 I$, $\tilde{\tau}_i = \tau_i/\rho$.

The equations solved at each outer nonlinear iteration of the time-step are given by

$$\begin{aligned} A\Delta\hat{u} &= f - Gp^n - Au^n && (1) \\ -L_1 \Delta p^{n+1} &= D(\hat{u} + \tilde{\tau}_2 G p^n) + L_2 p^n + b && (2) \\ u^{n+1} &= \hat{u} - \tilde{\tau}_3 G \Delta p^{n+1} && (3) \end{aligned}$$

where $\Delta\hat{u} = \hat{u} - u^n$, $\Delta p^{n+1} = p^{n+1} - p^n$.

Solve for pressure increment (2) with restarted GMRES(m) and AMG preconditioner.

Sliding Mesh: Nonsymmetric Matrices

- Sliding mesh based on discontinuous-Galerkin (DG) interior penalty (IP) term
- Adjacent Ω_A and Ω_B contribute non-symmetric terms in pressure-Poisson matrix
- Skew-Symmetric flux terms $\hat{C}(A,B) = -\hat{C}(B,A)$ at interface Γ^{BA}

$$
\begin{aligned}
\int_{\Gamma\setminus\Gamma^{AB}} \rho\, \hat{u}\, n_j \, d\Gamma + \int_{\Gamma^{AB}} \hat{C}(A,B)\, d\Gamma &= 0 \\
\int_{\Gamma\setminus\Gamma^{BA}} \rho\, \hat{u}\, n_j \, d\Gamma + \int_{\Gamma^{BA}} \hat{C}(B,A)\, d\Gamma &= 0
\end{aligned}
$$

$$
\hat{C}(\alpha,\beta) = \frac{1}{2}(\, m_j^{\alpha}\, n_j^{\alpha} - m_j^{\beta}\, n_j^{\beta}\,) + \lambda_C^{\alpha\beta}\,(\, p^{\alpha} - p^{\beta}\,), \quad \lambda_C^{\alpha\beta} = \frac{\tau}{2}\left(\frac{1}{L^{\alpha}} + \frac{1}{L^{\beta}}\right)
$$

Algebraic Multigrid

Smoothed aggregation (SA-AMG) is based on strongly connected neighborhoods:

$$N_i(\theta) = \left\{ j : |a_{ij}| \geq \theta \sqrt{a_{ii}\, a_{jj}} \right\}$$

Define C_i as the set of nodes in aggregate i and $\tilde{P}_{ij} = 1$, $i \in C_j$, and 0 otherwise.
Prolongator, $P = \left(I - \omega D^{-1} A^F \right) \tilde{P}$ where $D = \mathrm{diag}(A)$, A^F is the filtered matrix,

The classical Ruge-Stüben AMG (C-AMG) algorithm works in two passes:

- The first pass will make a selection of coarse nodes based on the number of strong connections that each node has
- The second pass is a refinement pass. It checks to make sure there are enough coarse nodes that information is not lost

Given a threshold value $0 < \theta \leq 1$, the node u_i is strongly connected to u_j if

$$|a_{ij}| \geq \theta \max_{k \neq i} |a_{ik}|$$

The prolongation matrix P is constructed row wise.

Set-up for C-AMG

Hypre–BoomerAMG : V-cycle

- parallel modified independent set (PMIS) coarsening of de Sterck (2004)
- strength of connection threshold $\theta = 0.25$
- extended$+i$ coarse–fine interpolation. $pmax = 2$ elements interpolation stencil
- aggressive coarsening on first 3 levels with multi-pass interpolation $pmax = 2$
- hybrid Gauss-Seidel smoother with 2 sweeps per level
- interpolation truncation $\tau = 0.25$
- retain transpose in Galerkin triple-product
- Sparsification on first 3 levels: drop tolerances $\gamma = [\,0.0, 0.01, 0.01\,]$.

Set-up for SA-AMG

Trilinos-MueLu : V-cycle

- symmetric SA coarsening
- unsmoothed prolongation $\omega = 0$
- aggregate size min 3, max 8
- strength of connection threshold $\theta = 0.03$ and distance Laplacian dropping
- implicit operator-stencil
- L_1 Gauss-Seidel smoothers. 2 sweeps
- re-balancing coarse matrices to improve parallel performance
- SuperLU coarse solver

Complexity and Stencil Width

Operator complexity C is defined as

$$C = \sum_{l=0}^{m} nnz(A_l)/nnz(A).$$

Directly correlated to the solver execution time

Average stencil size $s(A_l)$ is the average number of non-zero elements per row of A_l

Maximum average stencil size,

$$S_{avg} = \max_{1 \leq l \leq m} s(A_l)$$

Directly correlated to the solver set-up time

Vestas V27 Flow Field

- Turbulent air-flow modeling with wake vortex formation
- Rotational speed 43 RPM with a cross wind of 7.6 m/s

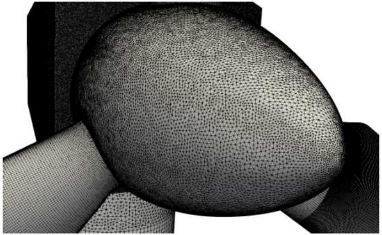


Figure: V27 41a 166M element mesh (45M pressure DOF).

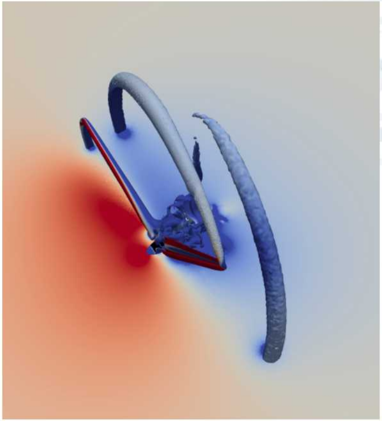


Figure: Pressure at 0.75sec. Isosurface of $v = 10$ m/s.

Pressure Solve and Set-up Times

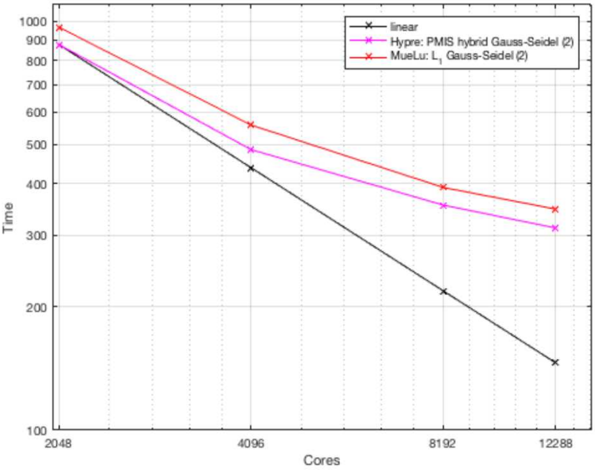
Model	sw	C	S_{avg}	θ	Iterations	Solver (s)	Set-up (s)	Wall-clock (s)
Nalu+Hypre	2	1.149	21.4	0.25	17.7	58.0	24.5	355
Nalu+MueLu	2	1.150	19.6	0.03	25.8	54.9	24	392

Table: V27 41 R1 500M elem, 229M Pressure DOF, 27K DOF/core: 8192 cores Cori, 10 steps.

- SA-AMG with Chebyshev smoother led to non-convergence
- 2 sweeps reduce flops and communication
- Momentum solve now dominant (non set-up) computational cost
- High wall-clock due to sparse matrix initialization (local/global graphs, memory)

Strong Scaling on NERSC-Cori

Total simulation times for 10 time steps of the V27 41 R1 500M elements



Pressure-Poisson Solver

Observations:

- GMRES(50) for Belos/MueLu, and GMRES(10) for Hypre–BoomerAMG. $tol = 1e-5$
- Hypre: aggressive coarsening significantly lowers C and S_{avg}; iterations increase
- Non-Galerkin sparsification further reduces solve and set-up times
- MueLu: unsmoothed prolongation reduce C and S_{avg}

Conclusions:

- Hypre and MueLu coarsening result in similar complexities C and solve times
- Hypre: higher computational cost per GMRES iteration
- Hypre: set-up costs increasing at higher core counts
- MueLu: higher GMRES iteration count that increases with core count
- MueLu: lower solve time with 2 smoother sweeps reduces communication

Thank you! Questions?

Acknowledgements:

- This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering, and early testbed platforms, in support of the nation's exascale computing imperative.
- NREL is a national laboratory of the U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, operated by the Alliance for Sustainable Energy, LLC under Contract No. DE-AC36-08GO28308.
- 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. This report followed the Sandia National Laboratories formal review and approval process (SAND2018-XXXXX). As such, the technical report is suitable for unlimited release.

ECP EXASCALE COMPUTING PROJECT