PETASCALE DNS USING THE FAST POISSON SOLVER PSH3D

Darren Adams[1], Michael Dodd[2] & Antonino Ferrante *[2]
[1]*The National Center for Supercomputing Applications (NCSA), University of Illinois, Urbana, IL, USA*
[2]*William E. Boeing Department of Aeronautics & Astronautics, University of Washington, Seattle, WA, USA*

Summary Direct numerical simulation (DNS) of high Reynolds number ($Re = \mathcal{O}(10^5)$) turbulent flows requires computational meshes of $\mathcal{O}(10^{12})$ grid points. Thus, DNS requires the use of petascale supercomputers. DNS often requires the solution of a Helmholtz (or Poisson) equation for pressure, which constitutes the bottleneck of the solver. We have developed and implemented a parallel solver of the Helmholtz equation in 3D called petascale Helmholtz 3D (PSH3D). The numerical method underlying PSH3D combines a parallel 2D Fast Fourier transform (P2DFFT) and a parallel linear solver (PLS). Our numerical results show that PSH3D scales up to at least 262,144 cores. PSH3D has a peak performance 6× faster than 3D FFT-based methods (e.g., P3DFFT) when used with the partial-global optimization. We have verified that the use of PSH3D with the partial-global optimization in our DNS solver does not reduce the accuracy of the numerical solution when tested for the Taylor-Green vortex flow.

INTRODUCTION AND NUMERICAL METHODS

DNS of turbulent flows, e.g. for solving the incompressible Navier-Stokes equations, using the pressure correction method [1], requires a solution to the Poisson equation for pressure at each time integration step. The Poisson solver is the bottleneck of such DNS codes. The current work extends the terascale parallel Poisson solver developed by [3, 4] based on [6], which uses 1D domain decomposition (1DD), to a petascale solver using 2D domain decomposition (2DD).

The key aspect of this method that enabled the advance toward a petascale implementation is the parallel Helmholtz/Poisson solver and is described briefly here. The general 3D Helmholtz equation

$$\nabla^2 p - \lambda^2 p = q. \tag{1}$$

is discretized using second-order finite differences on a uniform Cartesian mesh. $p_{(i,j,k)}$ and $q_{(i,j,k)}$ are unknowns and knowns, respectively, located at grid point (i, j, k). Applying a 2D FFT to the discrete form of Eq. (1) results in a system of tri-diagonal linear equations,

$$p^*_{(m,n,k-1)} - \alpha p^*_{(m,n,k)} + p^*_{(m,n,k+1)} = q^*_{(m,n,k)} \Delta z^2 \tag{2}$$

for each set of Fourier wavenumbers (m, n), where $p^*_{(m,n,k)}$ and $q^*_{(m,n,k)}$ are the Fourier transforms of $p_{(i,j,k)}$ and $q_{(i,j,k)}$, respectively and

$$\alpha = 2 + (\lambda^2 + \lambda_{(m,n)})\Delta z^2 \tag{3}$$

and

$$\lambda_{(m,n)} = \frac{2(1 - \cos(\frac{2\pi m}{N_x}))}{\Delta x^2} + \frac{2(1 - \cos(\frac{2\pi n}{N_y}))}{\Delta y^2}. \tag{4}$$

Solutions to Eq. (2) yield the second-order accurate $p^*_{(m,n,k)}$ to which an inverse 2D FFT is applied to obtain $p_{(i,j,k)}$.

Applying the 2DD, the grid $(N_x \times N_y \times N_z)$ is partitioned into N sub-domains $(N_x \times \frac{N_y}{P_y} \times \frac{N_z}{P_z})$ where the total number of processors is: $P_y \times P_z$ and $P_y \leq N_y$, $P_z \leq N_z$ and $P_y \leq \frac{N_x}{2} + 1$. The complex linear system in Eq. (2) is now sub-divided across P_z domain partitions, where $P_y \leq \frac{N_x}{2} + 1$. A divide-and-conquer strategy, developed by Schumann and Strietzel [7], leads to a PLS algorithm that only involves nearest neighbor communication *except* for a boundary correction step which must satisfy the following constraint equation

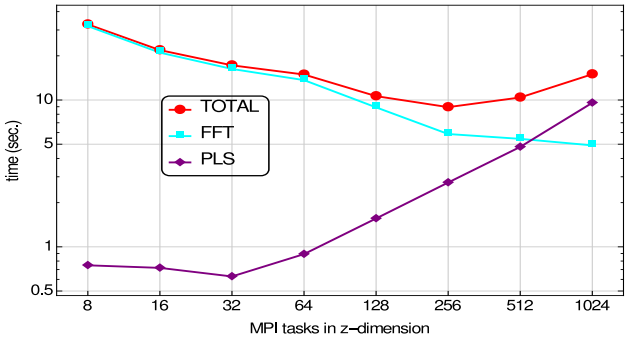
$$r_n R_{\kappa-1} - (\alpha + 2r_2)R_\kappa + r_n R_{\kappa-1} = q_1^\kappa - p_n^\kappa - p_2^\kappa. \tag{5}$$

where $\kappa = 1, 2, .., P_z$.

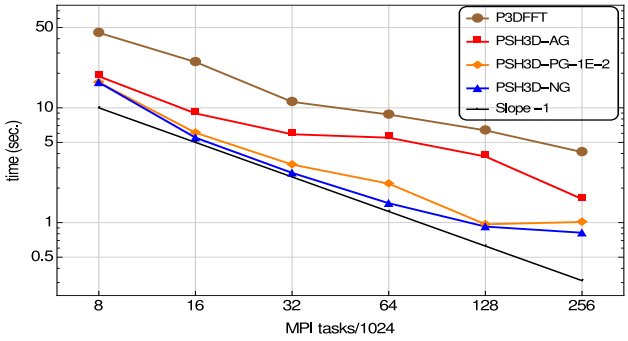
*Corresponding author. Email: ferrante@aa.washington.edu

RESULTS AND CONCLUSIONS

The total PSH3D execution time on a 8192^3 mesh was studied for various 2DD layouts on up to $262{,}144$ processor cores of Blue Waters. Dependence on 2DD layout and full strong scaling results are presented in Figure 1. We first tested a version in which we perform global reductions for all vectors; we refer to this as the "all-global" variant (PSH3D-AG). Next, we implemented a "local-only" version of the PLS, dubbed "no-global" (PSH3D-NG) . The NG simplification avoids the costly global reduction step demanded by the full algorithm and demonstrates the maximum possible level of scaling and performance; only limited by the P2DFFT operation. Finally, we investigate our novel hybrid approach, dubbed "partial-global" (PSH3D-PG), in which we assess the diagonal dominance in Eq. (5) and construct a parallel reduction scheme that applies only to vectors at locations where the ratio of diagonal to off-diagonal terms are above a prescribed tolerance. When optimized through the use of MPI-3 non-blocking collectives, this method shows much better performance relative to all-global while maintaining the second order accuracy expected by the full DNS algorithm.



(a) Total Poisson-time and constituent FFT and PLS times vs. P_z for PSH3D-AG using a series of 2DD layouts on $16{,}384$ processors.



(b) Strong scaling of total wall-clock time to solve the Poisson equation in 3D for variations of PSH3D and P3DFFT solvers [5].

Figure 1: Execution times for parallel Poisson solutions on an 8192^3 mesh using Cray XE6 compute nodes on Blue Waters. The PSH3D-AG method computes global reduction operations for *all* Fourier wave-numbers: (m, n). The PSH3D-PG-"tol" results solve the global reduction for fewer and fewer (m, n) as tol increases. The PSH3D-NG method avoids global reduction operations *completely*.

Performance of PSH3D-PG is shown in Figure 1b for a tolerance value of 10^{-2}. Larger values of the PG-tolerance reduce the number of global solutions that now require a solution, thus creating a more lightweight global-reduction step. The most dramatic improvement using this technique, was observed on 128k processors where the original AG wall-clock time was reduced from about 4 to 1 sec. Our numerical tests show that the method retains second-order accuracy for PG-tolerance values of up to 10^{-1}.

We have developed a new petascale solver for the Helmholtz 3D equation (PSH3D) and tested it on Blue Waters up to $262{,}144$ MPI ranks. The novelty of this method stands in performing a parallel 2DFFT in two spatial directions and a parallel linear solver in the third spatial direction rather than a fully parallel 3D FFT, and in the special "partial-global" treatment in implementing the global correction step. The results show that the PSH3D, when used with the AG PLS, is 30% to 40% faster than P3DFFT [5], and with the optimized PG PLS is more than 6 times faster and 4 times faster than P3DFFT on 128k and 262k MPI ranks of Blue Waters, respectively. We have then extended our DNS solver [2] from 1DD to 2DD and used PSH3D as its Poisson solver. The DNS code has been verified using this highly optimized solver (PG-PLS) by comparing the DNS results with the analytical solution of the Taylor-Green vortex flow [8].

References

[1] A. J. Chorin, *A numerical method for solving incompressible viscous flow problems*, J. Comput. Phys. **2** (1967), no. 1, 12–26.

[2] M. S. Dodd and A. Ferrante, *A fast pressure-correction method for incompressible two-fluid flows*, J. Comput. Phys. **273** (2014), 416–434.

[3] A. Ferrante, *Reduction of skin-friction in a microbubble-laden spatially developing turbulent boundary layer over a flat plate*, Ph.D. thesis, University of California, Irvine, 2004.

[4] A. Ferrante and S. Elghobashi, *On the physical mechanism of two-way coupling in particle-laden isotropic turbulence*, Phys. Fluids **15** (2003), no. 2, 315–329.

[5] D. Pekurovsky, *P3DFFT: A framework for parallel computations of Fourier transforms in three dimensions*, SIAM J. Sci. Comput. **34** (2012), no. 4, C192–C209.

[6] H. Schmidt, U. Schumann, and H. Volkert, *Three dimensional, direct and vectorized elliptic solvers for various boundary conditions*, Rep. 84-15, DFVLR-Mitt, 1984.

[7] U. Schumann and M. Strietzel, *Parallel solution of tridiagonal systems for the Poisson equation*, J. Sci. Comput. **10** (1995), no. 2, 181–190.

[8] G. I. Taylor, *On the decay of vortices in a viscous fluid*, Philos. Mag. **XLVI** (1923), 671–674.