

Parallelization of a Modern CFD Incompressible Turbulent Flow Code

J. M. McDonough^{1*}, T. Yang¹, M. Sheetz²

¹Department of Mechanical Engineering
University of Kentucky, Lexington, KY 40506-0503 USA

²Computing Services
University of Kentucky, Lexington, KY 40506 USA

Key words: Incompressible flow, parallelization, turbulence

Introduction

At this point in the beginning of the 21st Century there are already numerous reliable, inexpensive and widely-used commercial CFD codes. Fluent, STAR-CD and Flow3D are representative examples. In light of this, one might question the need for developing yet another incompressible flow code. But it is important to recognize that in many of the most successful commercial codes the underlying Navier--Stokes (N.-S.) solution procedures and the numerical analysis used to implement them are quite old, and we contend, in some cases outdated. In particular, most (but not all) commercial CFD software is based on one form or another of the SIMPLE algorithm (Patankar [1]). These basic approaches are not efficient for time-dependent simulations that are becoming increasingly more widely performed as computing power continues to improve. Furthermore, although most CFD code vendors now claim the ability of their product to employ large-eddy simulation (LES) for turbulence calculations, one essentially always finds that the manner in which LES has been implemented is far from what would be considered a true, theoretical LES, and in fact is what is sometimes termed “very large eddy simulation” (VLES), or essentially equivalently, unsteady Reynolds-averaged Navier--Stokes (URANS) modeling. Finally, and in many respects the most important point with regard to the present paper, is the fact that parallelization of these commercial codes has in almost all cases been done long after the codes were originally constructed; *i.e.*, they were not designed to be parallelized at the time they were initially coded.

We contend that despite the clear success of many commercial CFD codes, the shortcomings noted above are of such significance to warrant development of a new code--maybe even a new generation of codes--at least for research purposes, but probably even for commercial applications in the longer term. The work to be reported here has been underway for much of the past decade, and various portions of it have previously been reported (several times in *Parallel CFD Proceedings* volumes). But it has only been in the past year that this work has reached the maturity needed for complete implementation of a new code. In the remainder of this abstract (and in the full-length paper) we will briefly recall the basic system of equations that must be solved, the numerical algorithms used to accomplish this, and parallelization results for several different cases.

Governing equations and implementation

The equations being treated are the incompressible N.-S. equations (along with any other appropriate transport equations, not specifically considered herein):

$$\frac{DU}{Dt} = -\nabla P + \nu \Delta U + \mathbf{F}_B \quad \text{on } \Omega \subset \mathbb{R}^d, \quad d = 2, 3, \quad (1a)$$

with the divergence-free constraint

$$\nabla \cdot U = 0. \quad (1b)$$

* E-mail: jmmcd@uky.edu

Here, $\mathbf{U} \approx (u, v, w)^T$ is the velocity vector; P is pressure divided by constant density; \mathbf{F}_B is a similarly-scaled body force vector, and ν is kinematic viscosity. The differential operators, all of which can be assumed to be expressed in generalized coordinates are: the substantial derivative, D/Dt , the gradient and divergence, ∇ and $\nabla \cdot$, respectively, and the Laplacian, Δ .

Equations (1), together with appropriate initial and boundary conditions needed to constitute a well-posed problem, are solved on bounded domains Ω in terms of staggered, structured-grid formulations consisting of Gresho's projection 1 method [2] and standard numerical analytic techniques including trapezoidal integration in time, centered differencing in space, treatment of nonlinearities via the Newton--Kantorovich procedure and use of simple linear filters to treat cell- Re and aliasing problems (see Yang and McDonough [3]), the last two of which are not generally available--nor is use of projection--in current commercial codes.

Finally, the turbulence model is an updated version of the Hylin and McDonough approach [4] summarized by Sagaut [5], providing a high-fidelity form of LES that does not require wall models, and yet maintains efficiency (even with no parallelization) comparable to that of k - ϵ models.

But the most important aspect of this new code is that it was designed (long before coding began) to be parallelized, and even the earliest version of the code (McDonough and Dong [6]) was implemented and tested in parallel mode. In the present paper we present the parallelization strategies employed and provide data on speedups attained to date.

As discussed in [6], the basic strategy for parallelizing this particular code has been to employ a re-interpretation of Douglas and Gunn [7] time splitting to permit 2-D planar solves within a 3-D domain. These are executed in parallel, but in addition, within each such planar solve further parallelization is possible. This immediately provides an automatic two levels of parallelization. Beyond this is the fact that the large-eddy simulation-like turbulence model is implemented in such a way that the large and small scales of the solution can be computed simultaneously during each time step, thus providing yet a third level of parallelization. This last level has not yet been implemented (although the turbulence model, itself, has been, and we expect to have completed this in time to report results at the conference.

Figure 1 displays a rather complicated schematic of the parallelization strategy associated with the first two levels of parallelization described above, and in particular indicates the quite complex array operations that must be performed in order to efficiently implement MPI for the present code. Details of this figure will be presented at the conference.

Preliminary results

Figure 2 presents the speedups obtained up to this point on a 224-processor HP SuperDome SMP. Additional parallel data from a 64-processor PC cluster will be provided at the conference.

The data displayed in this figure correspond to four different tests of parallelization. On each of the 32^3 and 64^3 grids employed, we submitted jobs in both dedicated and non-dedicated modes. Overall, one observes very little difference in performance from one case to the next except in the comparison of dedicated and non-dedicated processing for the 64^3 grid using 32 processors; in this case, dedicated processing is far superior. Furthermore, we should note that although performance is far from the perfect linear speedup, it nevertheless continues to improve significantly through 32 processors in the dedicated mode of operation.

Further results and discussions will be provided at the conference and in the complete paper.

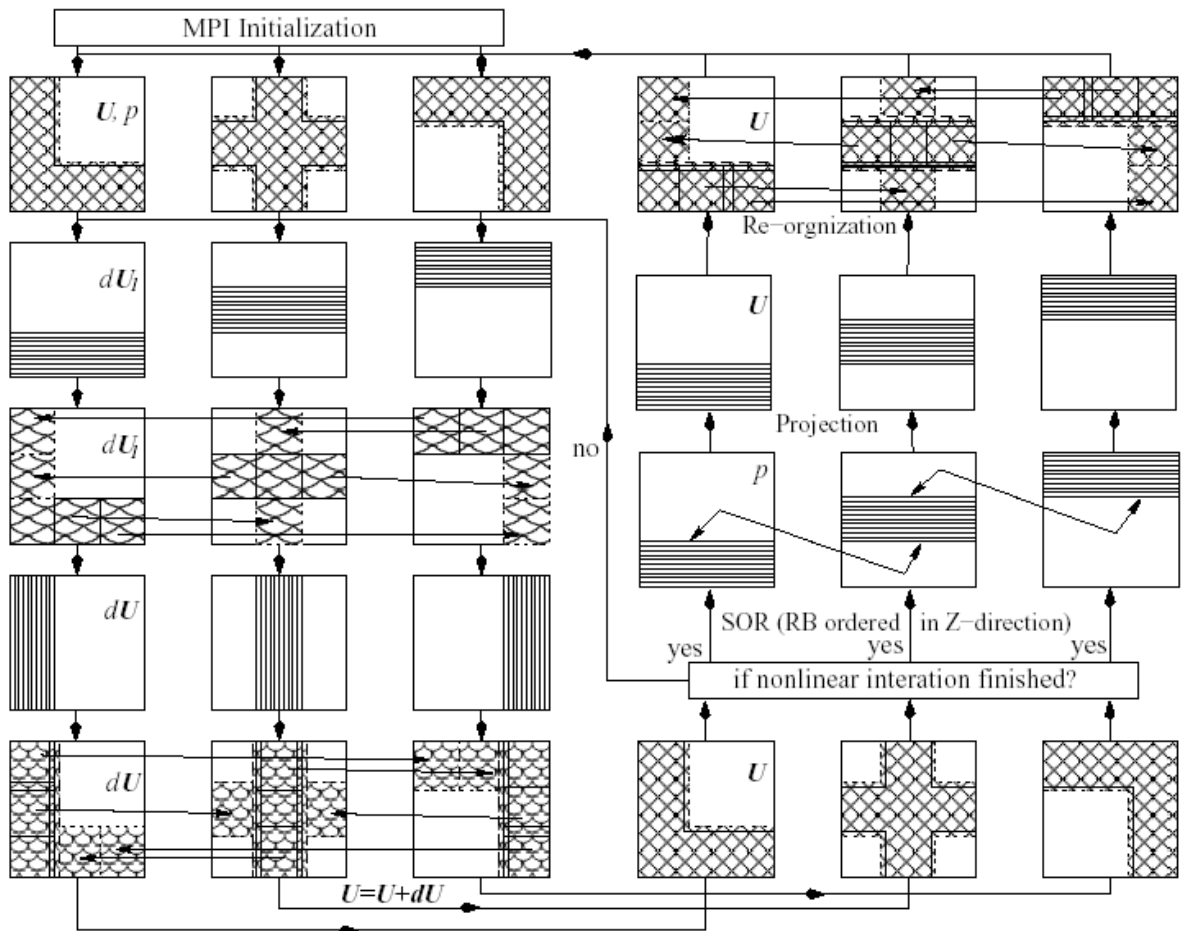


Fig. 1.

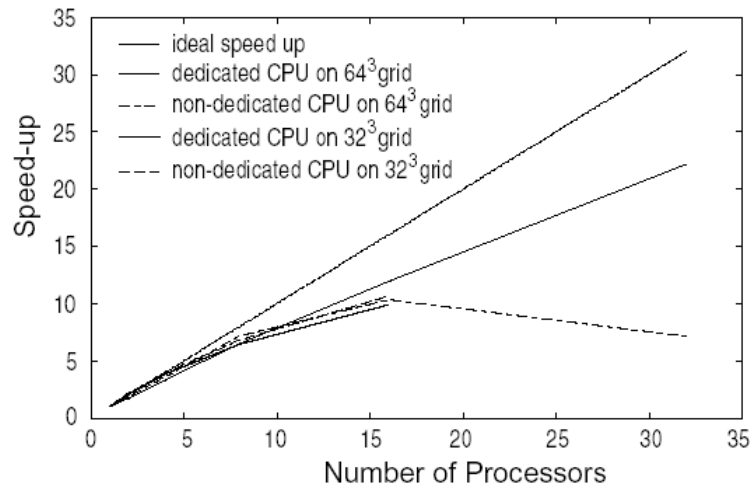


Fig. 2.

References

1. S.V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Corporation, New York, 1980.
2. P.M. Gresho, On the Theory of Semi-Implicit Projection Methods for Viscous Incompressible Flow and Its Implementation Via a Finite Element Method That Also Introduces a Nearly Consistent Mass Matrix. Part 1: Theory, *Int. J. Numer. Meth. Fluids*, **11**, 587--620, 1990.
3. T. Yang and J.M. McDonough, Solution filtering technique for solving Burgers' equation, under review for special issue of *Discrete and Continuous Dynamical Systems*, 2002.

4. E.C. Hylin and J.M. McDonough, Chaotic Small-Scale Velocity Fields as Prospective Models for Unresolved Turbulence in an Additive Decomposition of the Navier-Stokes Equations, *Int. J. Fluid Mech. Res.* **26**, 164-192, 1999.
5. P. Sagaut, *Large Eddy Simulation for Incompressible Flows*, Springer, Berlin, 2001.
6. J.M. McDonough and Shao-Jing Dong, 2-D to 3-D Conversion for Navier—Stokes Codes: Parallelization Issues, presented at *Parallel CFD 2000*, Trondheim, Norway, May 22--25, 2000.
7. J. Douglas and J.E. Gunn, A General Formulation of Alternating Direction Methods Part I. Parabolic and Hyperbolic Problems, *Numer. Math.* **6**, 428--453, 1964.