Parallel Computations of Unsteady Incompressible Viscous Flows with a Fully-Implicit Multigrid Driven Algorithm

by

A. Belov, L. Martinelli, and A. Jameson
Princeton University, Princeton, New Jersey, USA

Proceedings
6th International Symposium on Computational Fluid Dynamics

Lake Tahoe
September 1995

Parallel Computations of Unsteady Incompressible Viscous Flows with a Fully–Implicit Multigrid Driven Algorithm

Andrey Belov Luigi Martinelli Antony Jameson

Department of Mechanical and Aerospace Engineering
Princeton University
Princeton N.J. 08544
U.S.A.

We present a parallel implementation of a fully-implicit multigrid driven algorithm, originally proposed and validated for both unsteady incompressible Euler and Navier-Stokes calculations in reference [1, 2]. The solver is parallelized by using the Message Passing Interface (MPI) Standard and implemented on a multiprocessor IBM SP2 computer.

Introduction

One of the major difficulties encountered in the calculation of incompressible flow is the enforcement of the time-independent constraint on the velocity field, imposed by the continuity equation. Lack of a pressure evolution term precludes the straightforward application of efficient time-marching algorithms which are available for hyperbolic problems. One of the approaches to circumvent this difficulty is to introduce, in the continuity equation, a pseudo-temporal evolution term for the pressure as in the well-known artificial compressibility method of Chorin [3]. This transforms the governing equations into a hyperbolic system at the expense of time accuracy. However, the time accuracy can be recovered by augmenting spatial residuals in the momentum equations with the discretized time derivatives, and by driving these modified residuals to zero at each mesh point and every time step. This generalized artificial compressibility approach ensures a direct coupling between the velocity and pressure fields upon convergence of the *pseudo-transient* at each time step. The use of this strategy for incompressible flow has been originally proposed by Rogers and Kwak [4]. Similar approach has also been used by Miyake et al. [5] using an explicit, up to a second order accurate discretization in time, and a rational Runge-Kutta scheme for the subiterations. In our approach, fast convergence to a steady state of the pseudo-transient is achieved by making use of a multigrid technique originally developed by Jameson for compressible flow [6]. Our method, which couples a second order accurate backward differencing of the temporal derivatives and a very efficient finite-volume multigrid strategy, has been described and validated in [1, 2]. The A-stable discretization in time allows the stability constraint on the physical time step to be relaxed, while standard convergence acceleration techniques such as local pseudo-timestepping and residual averaging are applied to the pseudo-transient iteration. Also, to alleviate the stiffness effects stemming from the unsteady source terms included in the residuals, a point-implicit five-stage Runge-Kutta scheme is constructed, following the guidelines given in [7]. Finally, the range of the characteristic wave speeds associated with the hyperbolic pseudotransient problem is optimized for better convergence by employing a suitable form of the local preconditioning [8, 2].

Discretization of the governing equations

The continuity equation and the time-dependent incompressible Navier-Stokes equations are chosen as the mathematical model (in integral form):

$$\mathbf{I^m} \cdot \frac{d}{dt} \iint_{\Omega} \mathbf{w}\, dx\, dy + \oint_{\partial\Omega} (\mathbf{f}\, dy - \mathbf{g}\, dx) = \oint_{\partial\Omega} (\mathbf{r}\, dy - \mathbf{s}\, dx), \tag{1}$$

where $\mathbf{w}$ is the vector of flow variables: P, u and v; $\mathbf{f}, \mathbf{g}$-are the convective and $\mathbf{r}, \mathbf{s}$-are the dissipative fluxes. The $\mathbf{I^m}$ is the modified identity matrix, $\mathbf{I^m}$= diag[0,1,1], annihilating the time derivative of the pressure from the continuity equation. The augmented residual $\mathbf{R}^*(\mathbf{w_{ij}})$ is formed at each cell using a cell-centered formulation in space and a backward difference discretization in time of the form

$$\frac{d}{dt} = \frac{1}{\Delta t} \sum_{q=1}^{k} \frac{1}{q} \left[\Delta^-\right]^q, \quad \Delta^- = \mathbf{w}^{n+1} - \mathbf{w}^n \tag{2}$$

In particular, dropping the subscripts i, j for clarity, for a third order discretization in time one obtains

$$\mathbf{R}^*(\mathbf{w}) = \frac{1}{\Delta t} \left(p_1[\mathbf{w}\, V] + p_2[\mathbf{w}^n V^n] + p_3[\mathbf{w}^{n-1} V^{n-1}] + p_4[\mathbf{w}^{n-2} V^{n-2}]\right) + \mathbf{R}(\mathbf{w}), \tag{3}$$

where $\mathbf{p} = (11/6, -3, 3/2, -1/3)$, V is the cell volume and $\mathbf{R}(\mathbf{w})$ is the spatial residual, which includes convective and dissipative terms and the third order artificial dissipation term to prevent an odd-even decoupling. Application of the generalized artificial compressibility approach results in the system of coupled O.D.E.s to be solved to convergence on every time step

$$\frac{d(\mathbf{w}V)}{dt^*} + \mathbf{Pr} \cdot \mathbf{R}^*(\mathbf{w}) = \mathbf{0}, \tag{4}$$

where the diagonal preconditioning matrix is $\mathbf{Pr}$ =diag$[\Gamma^2, 1, 1]$ and $\Gamma^2 = \max(0.25, u^2 + v^2)$. The details of the point-implicit Runge-Kutta method which is used to drive the full approximation multigrid scheme can be found in reference [7].

A no-slip condition is imposed on the solid boundary and approximate nonreflecting boundary conditions are constructed on the far field boundary using method of characteristics as described in[2].

Parallel implementation

The acceleration techniques, such as multigrid and residual smoothing, which are built in our method rely on global information transfer within the computational domain. Thus, care must be taken to implement these inherently communication intensive techniques on a distributed memory computer. Parallel implementation of the solver follows the general guidelines outlined in[9]. The implementation is carried out on an IBM SP2 multiprocessor computer using the Message Passing Interface (MPI) Standard.

Domain decomposition

Typically, an O-mesh with $n_i \times n_j = 192 \times 128$ cells is used for the calculations. A static domain decomposition is employed at the beginning of each calculation to provide a nearly perfect load balancing. The computational domain is subdivided into n_p subdomains containing $\frac{n_i}{n_p} \times n_j$ cells each, where n_p is the number of processors used. Communication between the subdomains is achieved by passing a single or a double-level halo of the boundary variables.

Parallel multigrid

The multigrid acceleration used in the present method takes advantage of corrections, computed on a series of successively coarser meshes, driven by the residuals accumulated on the finest mesh. These corrections are then bilinearly interpolated on the finer mesh. The time stepping on the coarser meshes in a multigrid cycle effectively speeds up information transfer within the computational domain and provides fast smoothing of the long wavelength error modes on the finest mesh. The details of the multigrid strategy can be found in [10]. Unfortunately, the calculations on the coarser meshes increase the task granularity (the ratio of the number of bytes received by a processor to the number of floating point operations it performs) and, in a distributed memory environment, require memory-to-memory copying which has a negative effect on the parallel performance. The communication overhead could be reduced by updating flow variables on selected stages of the multistage scheme only, but, as suggested in [9], this leads to the prohibitive decrease in the convergence rate. In the present work, a full approximation multigrid is used which reproduces exactly the output of the original serial code, when the global residual smoothing is applied (see below). The double-level halo is passed on the finest mesh to provide all necessary values for the computation of the fourth order undivided differences entering the artificial dissipation term. The single-level halo is exchanged on the coarser meshes because the simpler, second order undivided differences are employed at the lower levels of the multigrid cycle.

Implicit residual averaging

The residual smoothing technique[11], allows a larger pseudo-time step to be used in the pseudo-transient calculation. This technique requires the solution of a tridiagonal system along each of the coordinate directions (*global* smoothing). When implemented in parallel, global smoothing requires unacceptably long idle times. Thus, in order to improve the parallel performance of the code, it pays off to confine the smoothing within each subdomain (*subdomain* smoothing). The convergence rate of the subdomain smoothing was found to match that of the global smoothing for the unsteady test calculations which are presented next.

Numerical results

The two-dimensional, unsteady, viscous flow over a pitching NACA0012 airfoil is computed to assess efficiency of the parallel solver. This problem has potential applications to high-lift maneuver of aircraft [12]. The airfoil performs a harmonic pitch around it's midchord point between 10^o and 30^o angle of attack. The reduced frequency based on the airfoil chord and free stream velocity is chosen to be 1.0. The Reynolds number based on the chord length is 1000. The O-mesh is rigidly attached to the airfoil and uses 192×128 cells with the far field boundary placed approximately 15 chords away from the body.

The domain decomposition is illustrated in Figure 1. It can be clearly seen from Figure 2 that the inter-processor boundaries are fully transparent to the flow and that the instantaneous isobars remain smooth across subdomains. The parallel speedup curves for both global and subdomain residual averaging are shown in Figure 3. To ensure the unbiased measure, the number of levels used in the W-cycles was kept fixed as the number of processors was varied. Clearly, the straightforward application of the global smoothing hampers the parallel efficiency and should be avoided. The parallel efficiency is recovered up to 80% on 8 processors by using the subdomain smoothing. For a fixed size mesh, the speedup curves tail off for a large number of processors since the time (i.e. the wall-clock time) is mostly being spent on the exchange of information on a coarser meshes. Results in Figure 3 suggest that a large (over 8) numbers of processors could be efficiently used on the larger meshes which are needed for high Reynold's number calculations, and for 3-D simulations.

Calculations have been performed using second, third and fourth order backward differencing in time, all of which proved to be robust. However, the number of multigrid cycles required to converge pseudo-transient to the same level (as measured by the maximum residual) proved to increase somewhat with the order of the scheme. This is possibly due to the corresponding increase of the leading coefficient p_1 in Equation 3, from 3/2 for the second order to 25/12 for the fourth order scheme. This increase leads to an additional shift of the Fourier symbol of the augmented residual operator to the left along the negative real axis and, consequently, increases the stiffness of the pseudo-time stepping. This problem might have a particularly significant effect in the far field, when stretched meshes are used. The best choice of the scheme depends therefore on the particular application and on the time accuracy required. For the cases considered here, the 3rd order accurate scheme provided the best compromise between the accuracy of the results and the stability constraint, enabling the resolution of a full pitching cycle in as few as 32 steps.

Preliminary results for the vortex shedding from a 2-D half cylinder are presented next. An O-mesh with 192 × 256 cells was used. The Reynolds number based on the diameter is 250. The time-resolved calculation was performed with a third order accurate time discretization using 50 multigrid W-cycles per time step. This resulted, at each time step, in a maximum residual less than 10^{-4} and a total mass flux out of the computational domain[†] less than 10^{-3}. This level of convergence of the pseudo-transient iteration is crucial to resolve the far wake accurately. The computed instantaneous vorticity distribution is presented in Figure 4. An abrupt change in the initial periodic pattern of the vortex street is clearly observed for this flow. This change in pattern is attributable to the onset of a physical instability in the wake. A qualitatively similar phenomenon was earlier computed using an unstructured spectral element method[13, 14]. A clean capture of the secondary instability associated with the vortex pairing provides an additional validation of our method. A detailed grid refinement and domain truncation study is required and planned to further investigate this phenomenon.

Conclusions

A fully-implicit multigrid solver has been implemented on a multiprocessor computer using MPI (Message Passing Interface) Standard and a domain decomposition approach. Good parallel performance is demonstrated by using the full approximation multigrid while confining residual averaging within the grid subdomains. The tests performed so far suggest that the parallel implementation of our multigrid solver will enable routine solutions of unsteady incompressible three-dimensional viscous problems.

Acknowledgement

We are grately indebted to Juan J. Alonso for numerous helpful suggestions on the parallelization of the code. Support for this work was provided by the Office of Naval Research through Grant N00014-93-I-0079, under the supervision of Dr. E.P. Rood.

[†] For a detailed discussion of these local and global convergence criteria, please refer to reference [2].

References

[1] Belov, A., Martinelli, L. and Jameson, A., "A Novel Fully Implicit Multigrid Driven Algorithm for Unsteady Incompressible Flow Calculations," Computational Fluid Dynamics '94, Proceedings of the 2nd European Computational Fluid Dynamics Conference, Stuttgart, Germany, 5-8 September 1994, John Wiley & Sons, 1994.

[2] Belov, A., Martinelli, L. and Jameson, A., "A New Implicit Algorithm with Multigrid for Unsteady Incompressible Flow Calculations," AIAA Paper 95-0049, AIAA 33rd Aerospace Sciences Meeting, Reno, Nevada, January 1995.

[3] A. J. Chorin, "A Numerical Method for Solving Incompressible Viscous Flow Problems," Journal of Computational Physics, **2**, 12-26, 1967.

[4] S. E. Rogers, and D. Kwak, "Upwind Differencing Scheme for the Time-Accurate Incompressible Navier-Stokes Equations," AIAA Journal, Vol. 28, No. 2, 253-262, 1990.

[5] Miyake, T., Sakamoto, Y., Tokunaga, H. and Satofuka, N., "Numerical Solution of Incompressible Flow Using Two-Step, One-Stage Runge-Kutta Time Integration Scheme," Presented at the 1st European Computational Fluid Dynamics Conference, Brussels, Belgium, 7-11 September, 1992.

[6] A. Jameson, "Time Dependent Calculations Using Multigrid, with Applications to Unsteady Flows Past Airfoils and Wings," AIAA 10th Computational Fluid Dynamics Conference, AIAA-91-1596, 1991.

[7] Melson, N.D., Sanetrik, M.D. and Atkins, H.L., "Time Accurate Navier-Stokes calculations with Multigrid Acceleration," presented at the 6th Copper Mountain Conference on Multigrid Methods, Copper Mountain, 4-9 April, 1993.

[8] Rizzi, A. and Eriksson L.-E., "Computation of Inviscid Incompressible Flow with Rotation," Journal of Fluid Mechanics, Vol. 153, pp. 275-312, 1985.

[9] Alonso, J.J., Mitty, T.J., Martinelli, L. and Jameson, A., "A two-dimensional multigrid-driven Navier-Stokes solver for multiprocessor architectures," in Proceedings of the Parallel CFD '94 Conference, Kyoto, Japan, May 1994.

[10] A. Jameson, "Multigrid Algorithms for Compressible Flow Calculations," In W. Hackbusch and U. Trottenderg, editors, Lecture Notes in Mathematics, **1228**, 166-201. Proceedings of the 2nd European Conference on Multigrid Methods, Cologne, 1985, Springer-Verlag, 1986.

[11] Martinelli, L., "Calculations of Viscous Flows with a Multigrid Method," Ph.D. Thesis, MAE Dept., Princeton University, October 1987.

[12] Mittal, S. and Tezduyar, T.E., "A Finite Element Study of Incompressible Flows Past Oscillating Cylinders and Aerofoils," Intern. J. for Numer. Meth. in Fluids, Vol. 15, 1073-1118, 1992.

[13] Henderson, R.D., "Unstructured Spectral Element Methods: Parallel Algorithms and Simulations," Ph.D. Thesis, MAE Dept., Princeton University, June 1994.

[14] Karniadakis, G.E. and Brown, G.L., "Vorticity Transport in Modeling Three-Dimensional Unsteady Shear Flows," Phys. of Fluids, 1995.

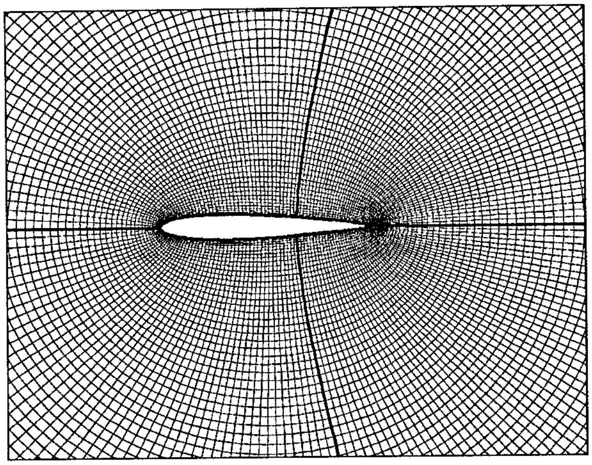

Figure 1: Detail of an O-Mesh and the Domain Decomposition.

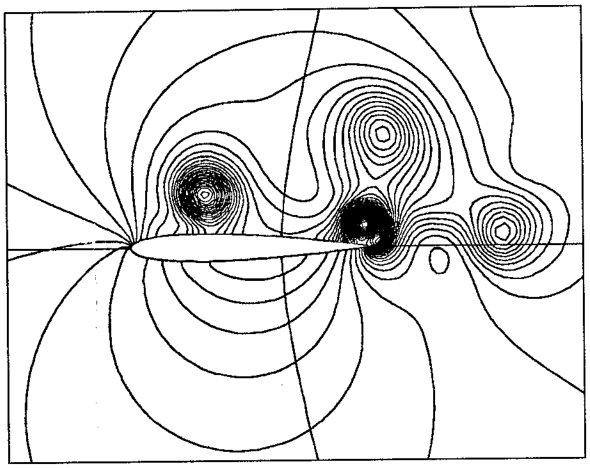

Figure 2: Instantaneous Pressure Field Around the Pitching NACA0012 Airfoil.

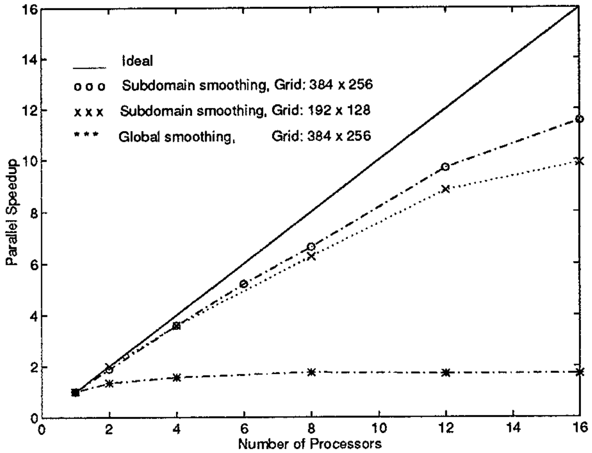

Figure 3: Parallel Performance of the Implicit Algorithm for Unsteady Incompressible Navier-Stokes Calculations.

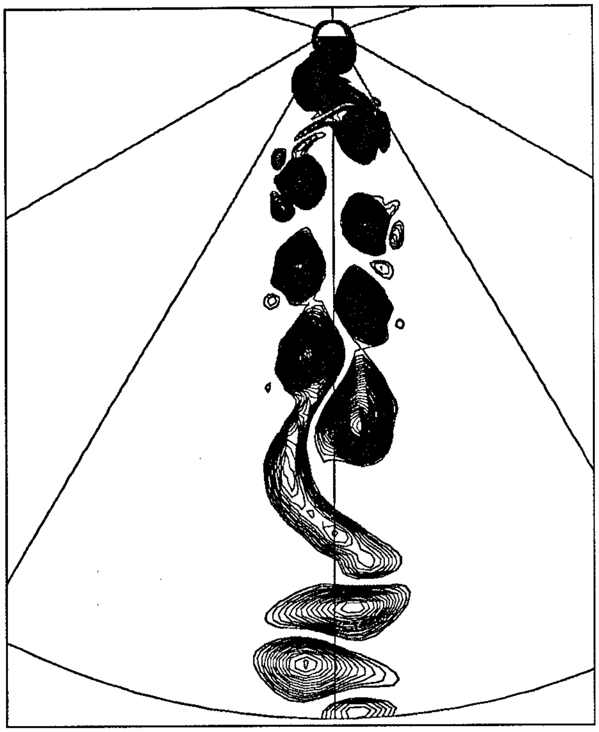

Figure 4: Computed Wake on a Half-Cylinder. Instanteneous Vorticity Contours