Three–Dimensional Incompressible Navier–Stokes Flow Computations about Complete Configurations using a Multiblock Unstructured Grid Approach

by

Chunhua Sheng, Daniel G. Hyams, Kidambi Sreenivas, J. Adam Gaither,
David L. Marcum, and David L. Whitfield

MSSU–COE–ERC–00–07

Computational Fluid Dynamics Laboratory

NSF Engineering Research Center

July 2000

Mississippi State University
P.O. Box 9627
Mississippi State, MS 39762

THREE-DIMENSIONAL INCOMPRESSIBLE NAVIER-STOKES FLOW COMPUTATIONS ABOUT COMPLETE CONFIGURATIONS USING A MULTIBLOCK UNSTRUCTURED GRID APPROACH

Chunhua Sheng*, Daniel G. Hyams†, Kidambi Sreenivas‡
J. Adam Gaither§, David L. Marcum¶, and David L. Whitfield¶
Computational Fluid Dynamics Laboratory
NSF Engineering Research Center, Mississippi State, MS 39762

W. Kyle Anderson#
NASA Langley Research Center, Hampton, VA 23681

Abstract

A multiblock unstructured grid approach is presented for solving three-dimensional incompressible inviscid and viscous turbulent flows about complete configurations. The artificial compressibility form of the governing equations is solved by a node-based, finite volume implicit scheme which uses a backward Euler time discretization. Point Gauss-Seidel relaxations are used to solve the linear system of equations at each time step. This work employs a multiblock strategy to the solution procedure, which greatly improves the efficiency of the algorithm by significantly reducing the memory requirements by a factor of 5 over the single-grid algorithm while maintaining a similar convergence behavior. The numerical accuracy of solutions is assessed by comparing with the experimental data for a submarine with stern appendages and a high-lift configuration.

Introduction

Unstructured grid technology offers several advantages for applications to computational field simulation (CFS). The geometric flexibility inherent in the approach makes it ideally suited for CFS applications with complex geometries and readily supports solution-adaptation methods, which can be essential for accurate simulations of complex flows with a minimal number of mesh points. In addition, unstructured grid generation is far more automatable than are the tasks associated with multiblock structured grid generation, resulting in considerable savings in the overall time required to perform simulations involving complex geometries. The primary disadvantage associated with the use of unstructured meshes is the inherent inefficiency of an unstructured solver. This inefficiency is caused by the need to explicitly store the data structure, and the fact that for tetrahedral meshes, many edges are coincident to each node. For implicit schemes, the problem is worsened because flux Jacobian matrices need to be stored for every edge in the mesh which is equivalent to storing the off-diagonal elements for a first-order linearization of the residual. The result is that the memory required for implicit schemes can be a limiting factor for performing large-scale turbulent flow computations on unstructured meshes.

The objective of this research is to develop an efficient, implicit, unstructured flow solver for computing three-dimensional, incompressible, high Reynolds number viscous flows about complete configurations, such as fully appended submarines. The primary focus is to reduce the memory required for viscous flow simulations by using an effective multiblock strategy. Here, the domain is divided into distinct blocks similar to the procedure often used for parallel computations. The solution is advanced from one time-step to the next by sequentially updating the solution in each block, where the Jacobians and metric quantities are computed only within each block. This technique has been previously demonstrated in Ref. [1] for two-dimensional computations and is extended to three-dimensions in the present work.

The basic solution algorithm is that of Anderson, Rausch, and Bonhaus, which utilizes the artificial compressibility form of the incompressible Navier-Stokes equations and is referred to as FUN3D [2]. A Spalart and Allmaras one-equation turbulence model [3] is used for simulating the effects of turbulence. The discretized scheme uses a node-based, finite volume scheme where the inviscid fluxes are evaluated using a second-order Roe scheme with a least squares procedure for data reconstruction, while the viscous fluxes are evaluated with a finite-volume formulation that is equivalent to a Galerkin type of approximation. The time advancement algorithm is based on the linearized backward Euler time-difference scheme, which yields

*Research Engineer, Member AIAA
†Graduate Research Assistant
‡Visiting Research Engineer, Member AIAA
§Research Assistant II
¶Professor, Member AIAA
#Senior Research Scientist, Member AIAA

a linear system of equations for the solution at each time step. The Gauss–Seidel procedure is used to solve the linear system of equations at each time step.

This paper is organized as follows. The artificial compressibility form of the three–dimensional Reynolds–averaged Navier-Stokes equations is first outlined, followed by the numerical procedures used in the current code. The multiblock algorithm is introduced next, which includes grid decomposition, data structure for the multiblock algorithm, and boundary treatment at block interfaces. Solutions of inviscid and viscous turbulent flows about appended submarines, and a turbulent flow about a high–lift configuration (Energy Efficient Transport), are presented to demonstrate the efficiency and accuracy of the current multiblock solver. Some conclusions are summarized in the last section.

Governing Equations

The unsteady three–dimensional incompressible Reynolds–averaged Navier–Stokes equations without body forces are written in Cartesian coordinates and in conservative form. A pseudo–time derivative of pressure is added to the continuity equation. The resulting set of equations in integral form represents a system of conservation laws for a control volume that relates the rate of change of a vector of average state variables q to the flux through the volume surface, which can be written as

$$V\frac{\partial q}{\partial t} + \oint_{\partial\Omega} f_i \, . \, \hat{n} \, dl - \oint_{\partial\Omega} f_v \, . \, \hat{n} \, dl = 0 \qquad (1)$$

where $\hat{n}$ is the outward–pointing unit normal to the control volume V. The vector of dependent state variables q and the inviscid and viscous fluxes normal to the control volume f_i and f_v are given as

$$q = \begin{bmatrix} p \\ u \\ v \\ w \end{bmatrix}$$

$$f_i \, . \, \hat{n} = \begin{bmatrix} \beta\Theta \\ u\Theta + n_x p \\ v\Theta + n_y p \\ w\Theta + n_z p \end{bmatrix}$$

$$f_v \, . \, \hat{n} = \begin{bmatrix} 0 \\ n_x\tau_{xx} + n_y\tau_{xy} + n_z\tau_{xz} \\ n_x\tau_{xy} + n_y\tau_{yy} + n_z\tau_{yz} \\ n_x\tau_{xz} + n_y\tau_{zy} + n_z\tau_{zz} \end{bmatrix}$$

where β is the artificial compressibility parameter; u, v, and w are the Cartesian velocity components in the x, y, and z directions, respectively; Θ is the velocity normal to the surface of the control volume, where

$$\Theta = n_x u + n_y v + n_z w$$

and p is the pressure. Note that the variables in the preceding equations are nondimensionalized with the characteristic length, freestream values of velocity, density, and viscosity. Pressure is normalized using the following relationship $(p - p_\infty)/\rho_\infty V_\infty^2$, where the subscript denotes a freestream or reference value. The shear stresses in Eq. (1) are given as

$$\tau_{xx} = (\mu + \mu_t)\frac{2}{\mathrm{Re}}u_x$$

$$\tau_{yy} = (\mu + \mu_t)\frac{2}{\mathrm{Re}}v_y$$

$$\tau_{yy} = (\mu + \mu_t)\frac{2}{\mathrm{Re}}w_z$$

$$\tau_{xy} = \tau_{yx} = (\mu + \mu_t)\frac{1}{\mathrm{Re}}(u_y + v_x)$$

$$\tau_{xz} = \tau_{zx} = (\mu + \mu_t)\frac{1}{\mathrm{Re}}(u_z + w_x)$$

$$\tau_{yz} = \tau_{zy} = (\mu + \mu_t)\frac{1}{\mathrm{Re}}(v_z + w_y)$$

where μ and μ_t are the laminar and turbulent viscosities, respectively, and Re is the Reynolds number.

Solution Algorithm

Finite–Volume Scheme

The baseline flow solver is a node–based, finite volume implicit scheme based on unstructured meshes with tetrahedral elements. The computational domain is divided into a finite number of tetrahedral elements from which control volumes are formed that surround each vertex in the mesh. The flow variables are stored at the vertices of the element. Equation (1) is then numerically integrated over the closed boundaries of the control volumes surrounding each node. These control volumes are formed by connecting the dual faces of the edges, as shown in Figure 1. These nonoverlapping control volumes combine to completely cover the domain and are considered to form a mesh which is dual to the mesh composed of tetrahedral elements formed from the vertices.

Numerical Flux Evaluation

The numerical evaluation of the surface integrals in Eq. (1) is conducted separately for the inviscid and viscous contributions, and is evaluated on the dual face of each edge (an edge–based approach). The inviscid fluxes are obtained on the faces of each control volume with a second–order Roe scheme, while the viscous

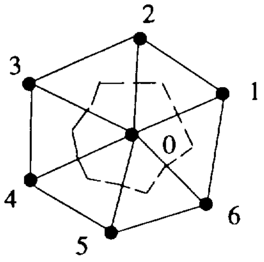

Figure 1. Control volume surrounding a node

terms are evaluated with a finite–volume formulation that is equivalent to a Galerkin type of approximation [2]. The inviscid fluxes on the boundaries of the control volumes are given by

$$\Phi = \frac{1}{2}\left(f_i(q^+;\hat{n}) + f_i(q^-;\hat{n})\right) - \frac{1}{2}|\bar{A}|(q^+ - q^-) \qquad (2)$$

where Φ is the numerical flux, and f_i is the flux vector given in Eq. (1). A nonsingular eigensystem for the matrix $|\bar{A}|$ in two–dimensional problems was reported in Ref. [2]. Quantities q^- and q^+ are the values of the dependent variables on the left and right side of the boundary of the control volume. For first–order accurate differencing, quantities q^- and q^+ are set equal to the data at the nodes lying on either side of the cell face. For higher–order differencing, these values are computed with a Taylor series expansion about the central node of the control volume

$$q_{face} = q_{node} + \nabla q \cdot \boldsymbol{r} \qquad (3)$$

where $\boldsymbol{r}$ is the vector that extends from the central node to the midpoint of each edge, and ∇q is the gradient of the dependent variables at the node and is evaluated with a least–squares procedure [2].

The viscous flux contribution to the residual is obtained using a finite–volume approach. In this approach, quantities such as velocity derivatives are first evaluated in each tetrahedral element of the mesh and the viscosity is computed as an average of the four nodes making up the tetrahedron.

Time Advancement Scheme

The time–advancement algorithm is based on the linearized backward Euler time–differencing scheme, which yields a linear system of equations for the solution at each time step

$$[A]^n \{\Delta q\}^n = -\{r\}^n \qquad (4)$$

where $\{r\}^n$ is the vector of steady–state residuals, $\{\Delta q\}^n$ represents the change in the dependent variables, and the solution matrix $[A]^n$ is written as

$$[A]^n = \frac{V}{\Delta t}I + \frac{\partial r}{\partial q} \qquad (5)$$

The solution of this system of equations is obtained by a relaxation scheme in which $\{\Delta q\}^n$ is obtained through a sequence of iterations, $\{\Delta q\}^i$, which converge to $\{\Delta q\}^n$. There are several variations of classic relaxation procedures which have been used in the past for solving this linear system of equations [4][5]. In this work, a point implicit Gauss–Seidel procedure as described in Ref. [2] is used. To clarify the scheme, $[A]^n$ is first written as a linear combination of two matrices representing the diagonal and off–diagonal terms

$$[A]^n = [D]^n + [O]^n \qquad (6)$$

and the solution to the linear system of equations is obtained by adopting a Gauss–Seidel type of strategy in which all odd–numbered nodes are updated first, followed by the solution of the even–numbered nodes. This procedure can be represented as

$$[D]^n \{\Delta q\}^{i+1} = \left[-\{r\}^n - [O]\{\Delta q\}^{(i+1)/i} \right] \qquad (7)$$

where $\{\Delta q\}^{(i+1)/i}$ is the most recent value of Δq, which will be at subiteration level i+1 for the odd–numbered nodes that have been previously updated and at level i for the even–numbered nodes. Normally 15–20 subiterations are adequate at each time step.

Turbulence Modelling

For the current study, the one–equation turbulence model of Spalart and Allmaras is used [3]. The model can be implemented in a straightforward manner because there is no need to define an algebraic length scale. The simple model [3] showed reasonably good prediction for most airfoil flows [1][2]. However, for the current submarine applications, it was found that the prediction of the axial force acting on the submarine was less than satisfactory. This is because the model overpredicts the level of eddy viscosity in the core of a vortex, which smears the vortex in the near field. This shortcoming can be worked around through modification of the production term as suggested in Ref. [6]. The production term $c_{b1}\omega\nu$ of the model [3] is multiplied by the "rotation function" f_{r1}

$$f_{r1}(r^*,\tilde{r}) = (1 + c_{r1})\frac{2r^*}{1 + r^*}\left(1 - c_{r3}\tan^{-1}(c_{r2}\tilde{r})\right) - c_{r1} \qquad (8)$$

Here, $\tilde{r} \equiv e/\sqrt{U_{ij}U_{ij}}$, $r^* \equiv \sqrt{2S_{ij}S_{ij}}/|\omega|$. e is the scalar measure of rotation and curvature effects which is evaluated with a complex expression (see Ref. [6]), $\sqrt{U_{ij}U_{ij}}$ is the norm of the whole tensor, and $\sqrt{2S_{ij}S_{ij}}$ is the strain rate. The constants are $c_{r1}=1$, $c_{r2}=12$, $c_{r3}=1$, which are based on the wingtip calculations of Dacles–Mariani et al. [7] and on curved and rotating boundary layers. The current work uses a simplified formula suggested by Spalart [8], which sets $\tilde{r} = 0$ in Eq. (8), and thus saves considerable effort in modifying the production term.

In the solution process, the equation for turbulent viscosity is solved using a backward Euler time–stepping scheme similar to that used for the flow variables, but separated from the flow equations. This results in a loosely coupled solution process that allows easy interchange with new turbulence models.

Multiblock Algorithm

Grid Generation

To generate a multiblock structured grid, the physical domain is first divided into several subdomains with prescribed block boundaries; the grid in each block is then generated separately. To generate a multiblock unstructured grid, however, the preceding process does not apply because prescribed block boundaries will degrade the grid quality in these regions. The multiblock unstructured grid is generated through a grid decomposition process. First, the computational grid of the entire physical domain is constructed within a single block using the grid generation software AFLR/SolidMesh [9] developed at the Engineering Research Center. This grid generation tool is very efficient in generating high aspect–ratio viscous unstructured meshes. A readily available partitioning software METIS [10] is then used to break the mesh into several sub–domains (or blocks) by selecting all the cells that fall into the region set for each subdomain. Each block contains the connectivity information required for that block, as well as the information to connect block–to–block interfaces.

Multiblock Strategy

The implementation of the multiblock algorithm on unstructured grids adopts a similar strategy as used in structured grids [11] and in two–dimensional unstructured work [1], i.e. a vertical mode in which a complete cycle is completed in each block before proceeding to the next block. The advantage of this approach is that the solution process (nonlinear and linear procedure) in each block is local and thus does not depend on the solution in other blocks. This nature of independence of the solution to other blocks offers great flexibility in both implementation and memory allocation for the algorithm, and also provides a natural platform for parallel implementations. (In fact, the main difference between the multiblock algorithm and the parallel implementation is that in the former, the solution in each block is performed sequentially in a prescribed order, while in the latter, all blocks are solved simultaneously). The disadvantage of the vertical mode is that the single block algorithm can never be recovered. Therefore if a grid has many blocks, the convergence rate could suffer because of the explicit nature of the data interchange between blocks. Since the main purpose of this work is to reduce the memory requirements of the unstructured grid algorithm, the memory allocation in the code must be done in a special way to achieve the best efficiency in both memory usage and CPU time. In the current work, all memory is allocated either locally or globally. For local memory, only the storage needed for the current block is allocated when the solution process enters that block, and this storage is freed at the time when the solution process leaves the block. The new storage is reallocated when the solution process moves to the next block. The local memory allocation is mainly for variables which do not need to be stored for all blocks, such as the flux Jacobian matrix (most costly part in the memory usage) which is updated after each time step in each block. On the other hand, the global memory allocation means that storage is allocated for all blocks and is not freed until the solution process is complete. Some data such as flow variables, grid coordinates and the distance to the wall surfaces, must be stored in a global way for all blocks. Therefore, by adopting the above strategy for memory allocation, the memory requirements of the multiblock algorithm will be much less than that of the single block algorithm where all memory is allocated globally.

Block Boundary Condition

In the current multiblock solver, each block is surrounded by a set of phantom cells which connect to the block boundaries and lie in other blocks (Figure 2), similar to the previous work for two dimensional problems [1]. One difference is that in the current work, the data in both interior cells and phantom cells are stored in the same arrays, similar to the data structure used for a cell–centered approach on multiblock structured grids [11]. This allows the calculation of residuals at block interfaces to be treated in the same way as the interior nodes in one step, instead of two steps as in the previous two–dimensional work [1]. The new data structure not only reduces the arrays needed for storing the information on phantom cells, but also keeps the modification to the original code to a minimum.

Because the current multiblock solver employes a vertical mode, the phantom nodes can only be updated with the values obtained in the previous step from other blocks prior to each new iteration. This is the only time in the solution process when the information is exchanged among blocks. Since the updating of gradients

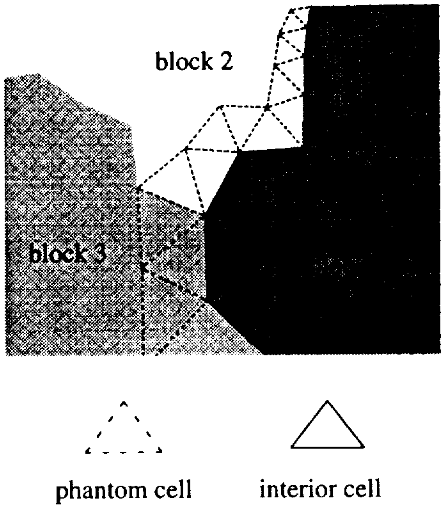

Figure 2. Interior cells and phantom cells for block 1

at the phantom nodes occurs such that they are “lagged” by one time step, and the correction to the dependent flow variables in other blocks is neglected during the Gauss–Seidel subiterations, the nodes on block interfaces can not be updated strictly at the correct point as done in the single–grid solver. However, the convergence behavior of the multiblock solver is not seriously degraded by explicitly updating phantom nodes, as shown in the subsequent section.

Results

To validate the current multiblock unstructured solver for large–scale simulations about complete configurations, three test cases are presented which include an inviscid flow around the SUBOFF model with a sail and four stern appendages, a viscous turbulent flow about the SUBOFF bare hull with four stern appendages, and a viscous turbulent flow about a high–lift configuration (Energy Efficient Transport). The computed force and moment coefficients for the SUBOFF bare hull with four stern appendages are compared with the solution obtained by a structured grid solver (UNCLE) [12] and experimental data [13]. The computed C_p distributions for the Energy Efficient Transport case are compared with experimental data [14]. All computations were carried out on a single UltraSPARC 333.6MHz processor with 2GB in–core memory and 4MB data cache size.

SUBOFF with Sail and Stern Appendages

The first case considers an inviscid flow around the SUBOFF model with a sail and four stern appendages. Figure 3 shows the unstructured grid for the SUBOFF configuration which is partitioned into 20 blocks; the shadowed colors represent different blocks. This grid contains approximately 216K nodes and 1.2M tetrahedral elements.

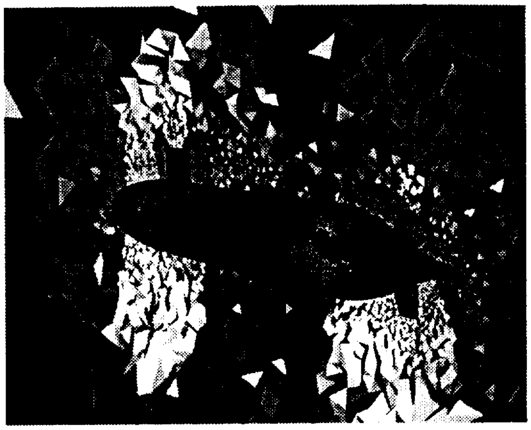

Figure 3. Twenty–block unstructured grid for fully appended submarine configuration

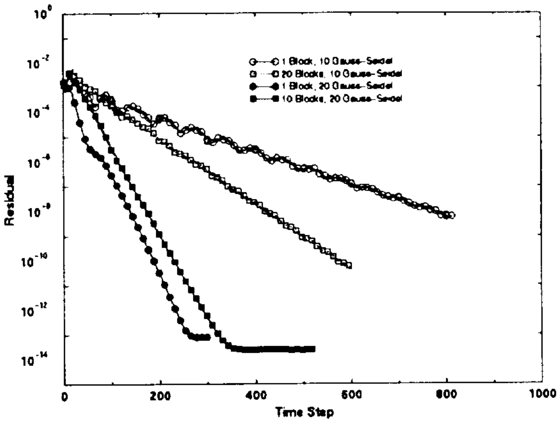

Figure 4. Convergence histories for inviscid flow about fully appended submarine

The Courant–Friedrichs–Lewy (CFL) number for the current computation has been linearly ramped from 20 to 200 over 100 iterations. The effect of the blocking strategy on the convergence rate can be seen in Figure 4 which shows the convergence histories for the multiblock and single–block solutions. When 10 Gauss–Seidel subiterations are used at each time step to obtain an approximate solution of the linear system, it is seen that the multiblock scheme converges somewhat faster than the single–block scheme. This may be due to the fact that when a mesh is decomposed into smaller sized blocks, the error signals of each block are damped faster during the subiterations, which thus improves the convergence of the global solution in the entire domain. For 20 subiterations, which is more typical for applications, the single–block scheme converges faster although the degradation due to the blocking is not significant. While the performance of the flow solver is not significantly effected by partitioning the mesh, the memory consummation is reduced by a factor of 5 compared to the single–block method. Similar benefits and performance were also obtained in the previous work on two–dimensional unstructured meshes [1] and three–dimensional structured multiblock grids [11]. Although not shown, computations performed on the multiblock and single–block meshes show no difference in the converged solutions.

SUBOFF with Stern Appendages

The second test case is for turbulent flows over the same SUBOFF bare hull with four stern appendages. The Reynolds number is 14 million based on the body length and freestream velocity. A number of grids, consisting of 250K to 1.14M nodes (1.4M to 6.8M elements), has been used for grid convergence studies. The spacing of the first mesh point off the surface is 0.6×10^{-6} relative to the length of the submarine; this spacing yields a y^+ value less than of 0.5 over the surface of the submarine. Figure 5 shows the surface grid in the region of the stern appendages for turbulent flow computations. Figure 6 shows the convergence histories of multiblock and single–block solutions on a mesh with 600K nodes and 3.5M elements. The convergence rate of the multiblock solution is close to that of the single–block solution; however, the memory required by the multiblock solution is less than 1/5 that of the single–block solution. It should be noted that the final solutions are independent of the blocking strategy.

The present work also compares computed results with the structured grid solution from Ref. [12] and the experimental data from Ref. [13]. The experimental data available for the SUBOFF configuration is not symmetrical about the negative and positive angles of attack. One has to take this into account when viewing the comparison of computed results with the experimental data. Figures 7–9 show computed axial force, lateral force, and pitching moment coefficients obtained on both structured and unstructured grids along with the experimental data. The effect of varying the unstructured mesh size on the computed forces and moments is seen in the figures. It was found that in order to resolve the lateral force, the unstructured grid needs at least 850K nodes for this configuration. However, the prediction of the axial force was still not satisfactory, even though the grid size was increased. To remedy this problem, a modified Spalart and Allmaras turbulence model [6] is examined. Figures 10–12 compare the effects of the original and modified models on grid with 850K nodes and at 0 and 10–deg angles of attack. The prediction of the force coefficients at moderate angles of attack (within 10–deg) is improved by using the modified model.

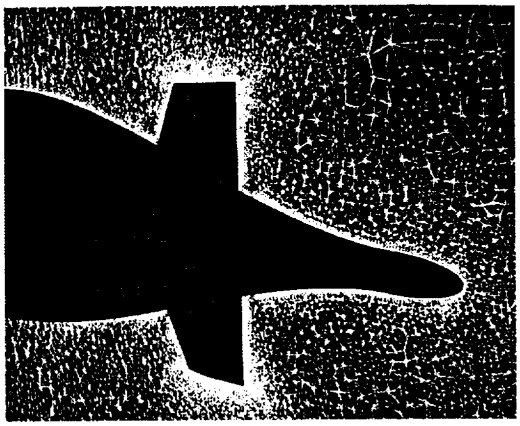

Figure 5. Twenty–block unstructured grid with 1.14M nodes for SUBOFF with stern appendages

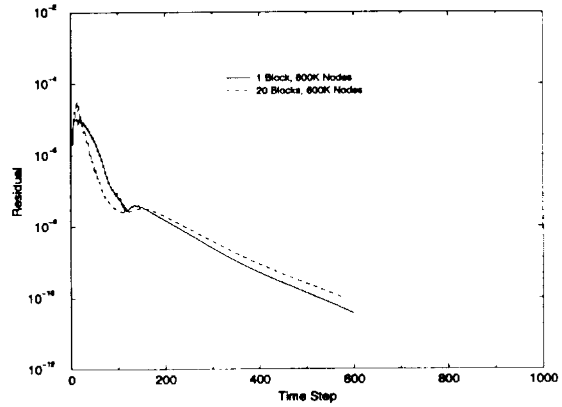

Figure 6. Convergence histories for turbulent flow about SUBOFF with stern appendages

Figures 13 and 14 show the convergence histories for the residual, forces and moments of the multiblock solution on the grid with 850K nodes at 10–deg angle of attack. The residual is reduced to the machine accuracy in 800 time steps, however, the force and moment coefficients are converged in just 500 time steps. The current multiblock solution required 600 MB in–core memory, and took 20 hours for every 100 time steps on the single processor mentioned before. This represents a memory savings of a factor of about 5 and no CPU time overhead over the single–block solution.

High–Lift Configuration

The last case is an example application for the computation over a high–lift configuration (Energy Efficient Transport). This is a fairly complex configuration with the slat, vane, and auxiliary flap attached to the

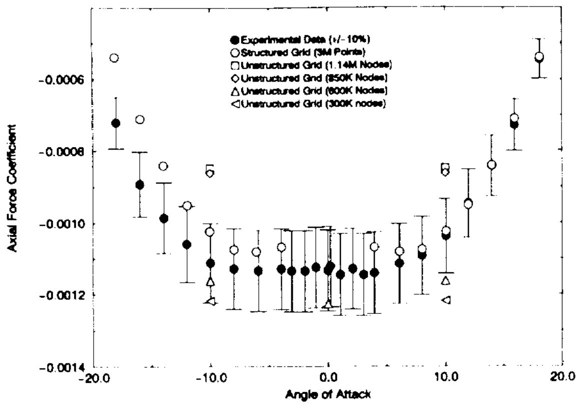

Figure 7. Comparison of axial force coefficient with different mesh sizes

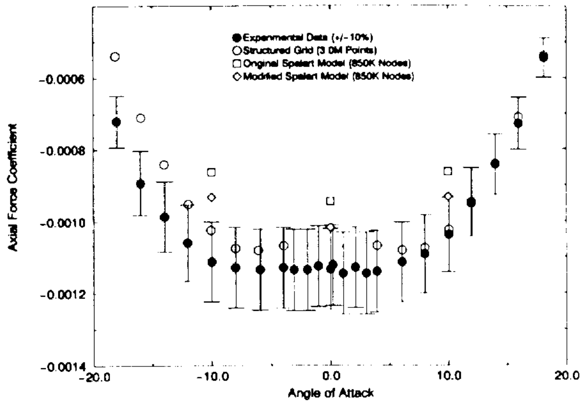

Figure 10. Comparison of axial force coefficient with simple and modified turbulence models

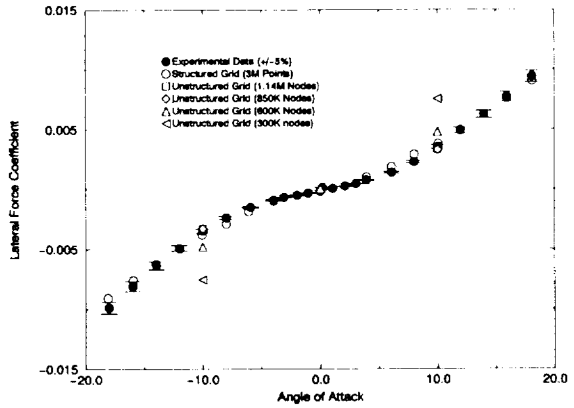

Figure 8. Comparison of lateral force coefficient with different mesh sizes

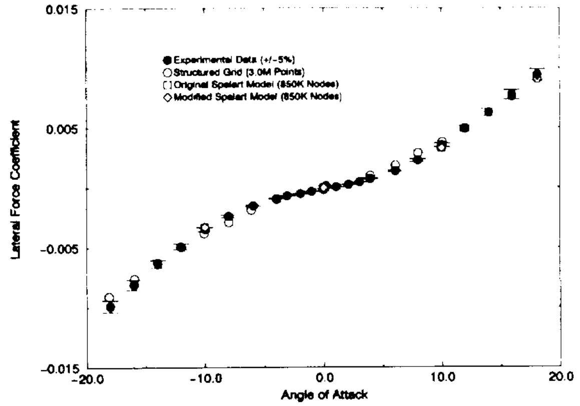

Figure 11. Comparison of lateral force coefficient with simple and modified turbulence models

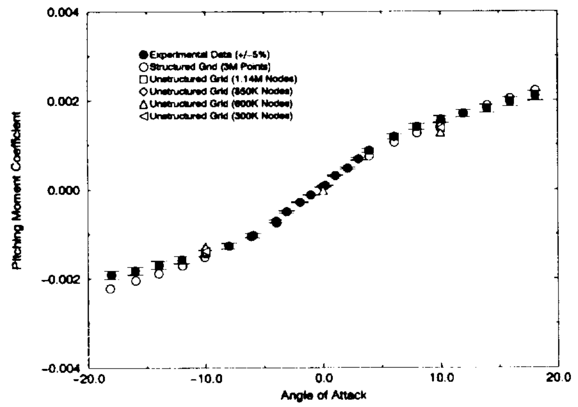

Figure 9. Comparison of pitching moment coefficient with different mesh sizes

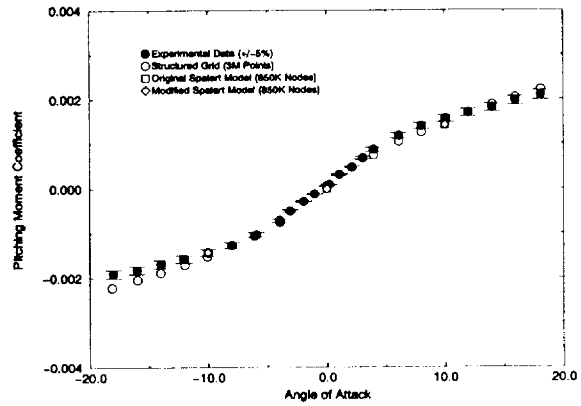

Figure 12. Comparison of pitching moment coefficient with simple and modified turbulence models

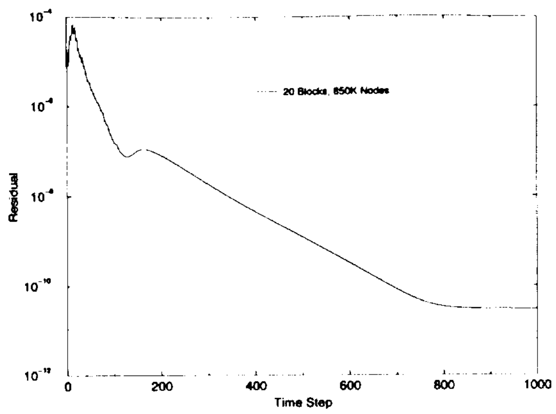

Figure 13. Convergence history of total residual for SUBOFF body with four stern appendages

Figure 15. High–lift configuration with 2 million nodes and 11 million elements

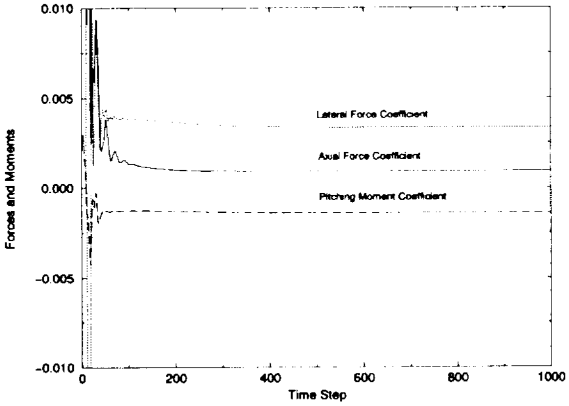

Figure 14. Convergence history of forces and moment for SUBOFF body with four stern appendages

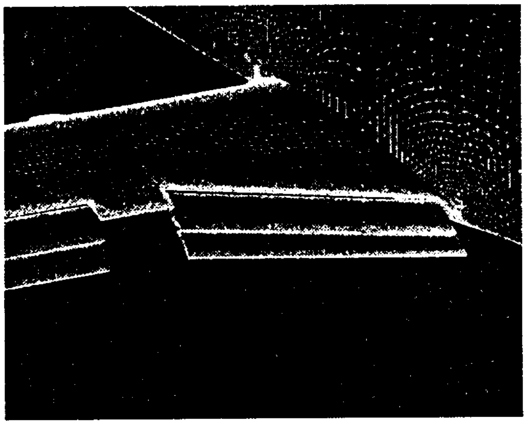

Figure 16. Surface grid in the wing region

main wing. The mesh for this configuration contains over 2 million nodes and 11 million elements for half of the physical space. The mesh is partitioned into 12 blocks for the multiblock computation. Figure 15 shows the configuration which has a symmetric plane at the center of the configuration. Figure 16 shows the surface grid in the wing region which indicates the complexity of the configuration.

The flow conditions for this computation are a 10–deg angle of attack and a Reynolds number of about 1.6 million based on the mean aerodynamic chord. The boundary conditions used in the computation are the characteristic variable boundary condition on the far–field, no–slip condition on the solid wall surface, and a symmetric flow condition on the central plane of the configuration. The CFL number was linearly ramped from 1 to 100 over 100 iterations, and 10 subiterations of a symmetric Gauss–Seidel method were used to solve the linear system. Figures 17–20 show the computed C_p distributions on the slat, main, vane, and flap elements at an inner section of the wing (27.6% of the span location from the root of the wing). It is seen that the computation is in reasonably good agreement with the experimental data [14].

The convergence history of the multiblock solution for the configuration is shown in Figure 21. The residual is reduced by 4 orders of magnitude in 1500 time steps. The current multiblock solver requires about 1 GB in–core memory for this configuration, using single precision definition for the flux Jacobian matrices. It should be pointed that there is no way to perform such a computation using the single–block solver, since the memory requirements are prohibitively high. However, the required CPU time for the multiblock solution is still very large, because of the size of the computational grid. Several procedures have been taking place at the Engineering Research Center to shorten the computational

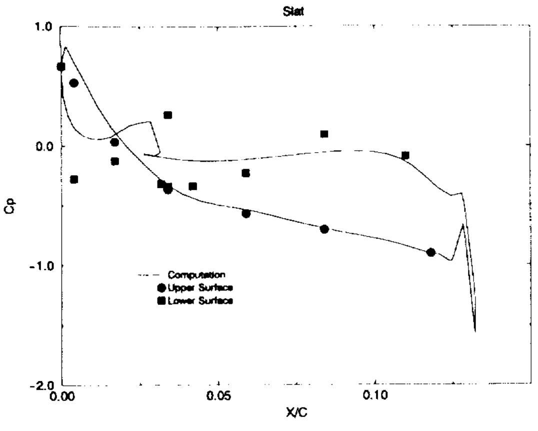

Figure 17. Comparison of Cp distribution on the slat element

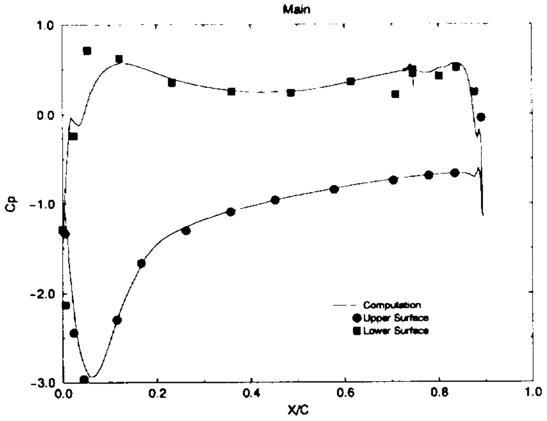

Figure 18. Comparison of Cp distribution on the main element

time. One is to use a mixed element method to reduce the number of mesh edges for the computations. Another procedure is to use the parallel algorithm to reduce the turnaround time since the present multiblock strategy readily supports the parallel implementations.

Conclusions

A multiblock unstructured flow solver is presented to solve the three–dimensional incompressible Reynolds–averaged Navier–Stokes equations. The multiblock technique has been previously demonstrated in the two–dimensional work, and is now extended to three–dimensional problems. Results presented show that, by properly allocating the memory for the code, the multiblock solution may reduce the memory requirements by a factor of 5 compared to the single–block method. Solutions about an appended submarine and a

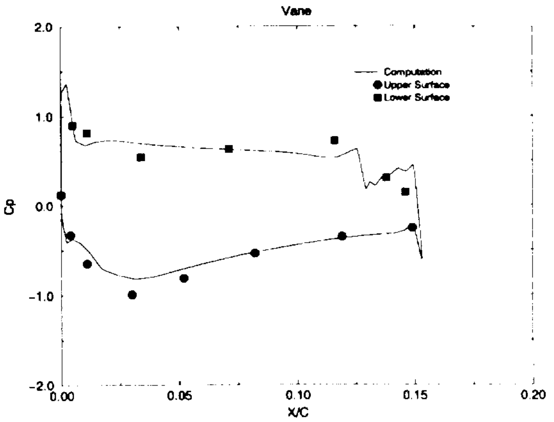

Figure 19. Comparison of Cp distribution on the vane element

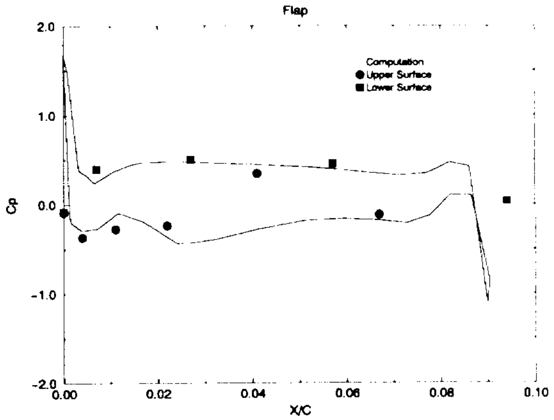

Figure 20. Comparison of Cp distribution on the flap element

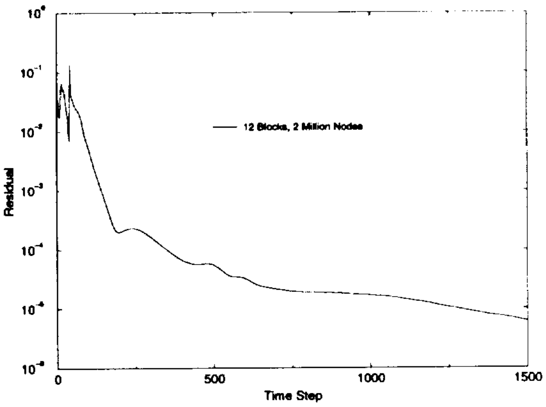

Figure 21. Convergence history of multiblock solution for high–lift configuration

high–lift configuration are presented to demonstrate the capability of the current solver for large–scale complex flow simulations, which is otherwise impractical to implement due to the large memory requirements of the unstructured flow solver. Future work of adding the capability to handle the mixed elements and parallel implementation will further improve the efficiency of the unstructured solver by greatly reducing the computational time and memory requirements, and thus provide a highly efficient and cost–effective tool to predict large–scale realistic complex flows about complex configurations.

Acknowledgement

This research was supported by NASA Langley Research Center under the NASA Grant NAG–1–1990. This support is gratefully acknowledged.

References

[1] Sheng, C., Whitfield, D.L., and Anderson, W.K., "A Multiblock Approach for Calculating Incompressible Fluid Flows on Unstructured Grids," AIAA Paper 97–1866, June 1997.

[2] Anderson, W.K., Rausch, R.D., and Bonhaus, D. L., "Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids," AIAA Paper 95–1740, June 1995.

[3] Spalart P., and Allmaras, S., "A One–Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92–0439, January 1991.

[4] Anderson, W.K., "Grid Generation and Flow Solution Method For Euler Equations on Unstructured Grids," NASA Technical Report TM–4295, April 1992.

[5] Batina, J.T., "Implicit Flux–Split Euler Schemes for Unsteady Aerodynamic Analysis Involving Unstructured Dynamic Meshes," AIAA Paper 90– 0936, January 1990.

[6] Spalart, P.R., and Shur, M., "On the Sensitization of Turbulent Models to Rotation and Curvature," *Aerospace Science and Technology*, May 1997. pp. 297–302.

[7] Dacles–Mariani, J., Zilliac, G.G., Chow, J.S., and Bradshaw, P., "Numerical/Experimental Study of a Wingtip Vortex in the Near Field," *AIAA Journal*, Vol. 33, No. 9, September 1995. pp. 1561–1568.

[8] Spalart, P.R., private conversations, NSF Engineering Research Center for Computational Field Simulation, Mississippi State, MS, October 1998.

[9] Marcum, D.L., "Generation of Unstructured Grids for Viscous Flow Applications," AIAA Paper 95–0212, January NV.

[10] Karypis, G., and Kumar, V., "METIS – A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill–Reducing Orderings of Sparse Matrices," Version 3.03, University of Minnesota, Department of Computer Science / Army HPC Research Center, Minneapolis, MN, November 1997.

[11] Sheng, C., Taylor, L.K., and Whitfield, D.L., "Multiblock Multigrid Solution of Three–Dimensional Incompressible Turbulent Flows About Appended Submarine Configurations," AIAA Paper 95–0203, January 1995.

[12] Jonnalagadda, R., Taylor, L.K., and Whitfield, D.L., "Multiblock Multigrid Incompressible RANS Computation of Forces and Moments on Appended SUBOFF Configurations at Incidence," AIAA Paper 97–0624, January 1997.

[13] Roddy, R.F., "Investigation of the Stability and Control Characteristics of Several Configurations of the DARPA SUBOFF Model," Department Report, Ship Hydromechanics Department, David Taylor Research Center, Maryland, September 1990.

[14] Anderson, W.K., private conversations, NSF Engineering Research Center for Computational Field Simulation, Mississippi State, MS, November 1998.