AIAA JOURNAL
Vol. 30, No. 8, August 1992

Temporal Adaptive Euler/Navier-Stokes Algorithm Involving Unstructured Dynamic Meshes

W. L. Kleb* and J. T. Batina†
NASA Langley Research Center, Hampton, Virginia 23665
and
M. H. Williams‡
Purdue University, West Lafayette, Indiana 47907

A temporal adaptive algorithm for the time integration of the two-dimensional Euler or Navier-Stokes equations is presented. The flow solver involves an upwind flux-split spatial discretization for the convective terms and central differencing for the shear-stress and heat flux terms on an unstructured mesh of triangles. The temporal-adaptive algorithm is a time-accurate integration procedure that allows flows with high spatial and temporal gradients to be computed efficiently by advancing each grid cell near its maximum allowable time step. Results indicate that an appreciable computational savings can be achieved for both inviscid and viscous unsteady airfoil problems using unstructured meshes without degrading spatial or temporal accuracy.

Introduction

In the last several years, significant progress has been made in developing computational fluid dynamics methods for aerodynamic analysis.[1] Recently, the efforts have been toward solving Euler and Navier-Stokes flows for full configurations. Many of these applications have been done using structured grids, i.e., grids that have a fundamental ordering to them such as grid cell 1 is known to lie next to grid cell 2 a priori. Maintaining this type of structure while adaptively refining or moving the mesh to provide more accurate solutions requires very involved methods. So, as a result, the emphasis of algorithm development has been shifting toward unstructured grid methodologies[2,3] that allow for simpler treatment of dynamic, adaptive meshes because the ordering restriction is circumvented.

Motivation

As a preliminary step toward efficiently solving unsteady flows in three dimensions, spatially adapting meshes have been implemented in two-dimensional flows.[4-7] These solvers place more grid cells in regions of interest (i.e., high-spatial gradients) and remove unnecessary cells from other areas of the flow. As a result of this spatial adaption, the meshes produced are very efficient in terms of the number of cells necessary to produce a predetermined solution accuracy. This adaption, however, has been almost exclusively limited to the spatial dimensions when in fact, adapting in time is also desirable if unsteady flows are considered. This statement is true for either implicit or explicit time-marching flow solvers. It is probably most evident that some type of temporal refinement, or adaption, is necessary for explicit time-marching schemes because these schemes have a numerical stability restriction tied to the global minimum time step usually dictated by the smallest of the cell volumes. This implies that the much larger cell volumes (usually many orders of magnitude larger than the size of the smallest cell) are being integrated in time at a fraction of their maximum allowable time step [a low Courant, or Courant-Friedricks-Lewy (CFL), number]. This is not only a waste of computational effort for the larger cell volumes, but it also introduces inaccuracies because net dissipative errors usually increase for a given scheme when a low Courant number is used. Implicit time-marching schemes do not have the numerical stability restriction of explicit schemes (e.g., CFL number), but they are designed to be most advantageous for flows where the small physical time scales throughout the flow can be ignored. In the case of chemically reacting flows for instance, one can easily envision the need to accurately resolve small time scales (i.e., rapidly varying chemical reaction source terms). This means that the global time step used in an implicit scheme might have to be so small that the extra expense of the implicit implementation is more than simply using an explicit time-marching formulation. With either scheme, however, the solution is always integrated in time with a globally determined time step. Thus, a viable approach to improve the efficiency of time-marching schemes for unsteady problems is through the use of some form of temporal adaption, whereby each grid cell is integrated according to the local flow physics and numerical stability. Time accuracy is maintained by bringing all cells to the same time level as determined by the step size of the largest cell.

Past Efforts

The temporally adaptive procedure entails integrating individual cells according to their maximum allowable time step (as dictated by the time scales in the local flowfield or numerical stability) instead of a globally determined time step for the entire domain. In order to maintain a time-accurate solution, a method of accurately exchanging discrete information between two cells being integrated at disparate time steps needs to be developed. Osher and Sanders[8] introduced a technique to determine the interface values and proved that for scalar, monotone discretizations the solution using variable time step differencing converges to the correct physical solution. The scheme uses a predictor-corrector method of determining the interface values, which requires the storage of intermediate values; consequently, when this procedure is extended to more

Presented as Paper 90-1650 at the AIAA 21st Fluid Dynamics, Plasma Dynamics, and Lasers Conference, Seattle, WA, June 18-20, 1990; received Aug. 10, 1990; revision received Oct. 4, 1991; accepted for publication Nov. 13, 1991.

*Aerospace Engineer, Aerothermodynamics Branch, Space Systems Division. Member AIAA.
†Senior Research Scientist, Unsteady Aerodynamics Branch, Structural Dynamics Division. Senior Member AIAA.
‡Professor of Aerospace Engineering, School of Aeronautics and Astronautics. Member AIAA.

spatial dimensions with an arbitrary number of intermediate values, the storage becomes unwieldy. Löhner et al.[9,10] proposed a domain-splitting technique for hyperbolic problems in which domains of differing time steps are slightly overlapped to obtain the interface values. These references also propose an integration sequence for cells in regions where time steps are integer multiples of global minimum values. Pervaiz and Baron[11] and Pervais[12] continue in this vein with chemically reacting flows and go on to include a time step criteria that depends on the flow physics in addition to the inherent numerical stability restriction of explicit time-marching schemes. Berger[13] introduced a more general approach than that of Löhner that does not require that the domains of disparate time steps be overlapped in order to calculate temporal interface quantities. It is also shown that the interface equations for mesh refinement in time are stable and give good results when either a discontinuity or a rarefaction wave passes through the interface.

Present Study

A hybrid method of time integrating the Euler/Navier-Stokes equations is proposed for use with unstructured grids. The method used for determining temporal interface values is similar to that developed by Berger,[13] and the temporal integration is done in such a manner that it can be applied to many different types of evolution problems. In other words, the type of discretization used for determining the spatial flux balance is divorced from the temporal adaptive procedures. Particular attention was made to maintain a separation between the temporal and spatial operators, so that a particular computer architectural trait (e.g., vectorization or parallelization) for which the spatial discretization might have been tuned would not be undermined by the temporal adaption. The distribution of time steps throughout the domain is similar to that of Pervaiz and Baron,[11] Pervais,[12] and Löhner et al.[10] but the sequence of time integration is altered.

Analytical Discussion

In this section, the laminar Navier-Stokes equations are presented including the numerical algorithm used for their solution. The algorithm is an extension of the Euler solver of Batina,[14] which uses either flux-vector or flux-difference splitting for the spatial discretization and either explicit or implicit time marching.

Governing Equations

The unsteady two-dimensional laminar Navier-Stokes equations in Cartesian coordinates can be written in conservation law form as

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial y} = \frac{\partial \boldsymbol{R}}{\partial x} + \frac{\partial \boldsymbol{S}}{\partial y} \tag{1}$$

where the vector of conserved variables $\boldsymbol{Q}$ and the convective fluxes $\boldsymbol{F}$ and $\boldsymbol{G}$ are given by

$$\boldsymbol{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix}, \quad \boldsymbol{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (e +)u \end{bmatrix}, \quad \boldsymbol{G} = \begin{bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ (e + p)v \end{bmatrix} \tag{2}$$

and the viscous fluxes $\boldsymbol{R}$ and $\boldsymbol{S}$ are defined as

$$\boldsymbol{R} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - q_x \end{bmatrix}, \quad \boldsymbol{S} = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} - q_y \end{bmatrix} \tag{3}$$

The pressure p is given by the equation of state for a perfect gas such that

$$p = (\gamma - 1)[e - \tfrac{1}{2}\rho(u^2 + v^2)] \tag{4}$$

The components of the shear-stress tensor are defined using Stoke's hypothesis with the coefficient of viscosity given by Sutherland's formula; the heat fluxes are defined according to Fourier's law.

The Euler equations are obtained from Eq. (1) by simply setting the dissipative fluxes $\boldsymbol{R}$ and $\boldsymbol{S}$ of Eqs. (3) to zero.

Spatial Discretization

A finite volume approach is used so that Eq. (1) is written in the integral form,

$$\frac{\partial}{\partial t}\int\int_A \boldsymbol{Q}\, dx\, dy + \oint_{\partial A} (\hat{\boldsymbol{F}}\, dy - \hat{\boldsymbol{G}}\, dx) = \oint_{\partial A} (\boldsymbol{R}\, dy - \boldsymbol{S}\, dx) \tag{5}$$

where the convective fluxes now have grid speed terms due to a deforming mesh:

$$\hat{\boldsymbol{F}} = \boldsymbol{F} - x_t\boldsymbol{Q}$$
$$\hat{\boldsymbol{G}} = \boldsymbol{G} - y_t\boldsymbol{Q} \tag{6}$$

The last two terms of Eq. (5) are boundary integrals around a control volume. This integral form of the conservation laws is solved for the flow variables in each of the control volumes, which are taken to be the triangles of the mesh as shown, for example, in Fig. 1. The scheme uses upwind flux-split spatial discretization for the convective terms and central differencing for the viscous terms. The scheme is second-order accurate in space. For more specific details about the solver see Ref. 14.

Temporal Discretization

As a first step toward a multistage Runge-Kutta time stepping scheme, a one-stage (i.e., backward Euler) scheme was used to integrate the solution time accurately,

$$q^{n+1} = q^n + \Delta t R(q^n) \tag{7}$$

where $R(q^n)$ is the residual computed using the solution stored at time level n. This method is first-order accurate in time.

Temporal Adaption

Temporal adaption can be thought of as time-accurate local time stepping. Local time stepping is typically used in a non-time-accurate manner to accelerate the convergence of the unsteady Euler or Navier-Stokes equations to steady state. Since only steady state is desired, it does not matter that every point in the flow is at a different time. This, of course, is not the case for a time-accurate problem, since each point in the flow for such a calculation must be on the same temporal level to maintain time accuracy. The problem is that, if all of the grid cells are marched at the same time step with either an explicit or an implicit time-marching scheme, the most restrictive time step must be used in order to maintain numerical stability for explicit time marching or resolve the unsteadiness of the flow with either method. Temporal adaption is a

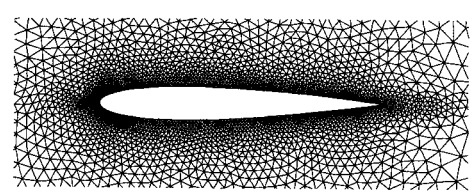

Fig. 1 Partial view of a typical unstructured mesh of triangles about a NACA 0012 airfoil.

method to march each cell at its own time step, although ultimately the flow variables in all cells reach the same point in time. Temporal adaption can be viewed as similar to spatial adaption in that small time steps should only be taken in localized areas governed by the flow physics, accuracy, or stability and not in the entire flowfield.

Illustrative Example

As a way of simplifying the discussion of the temporal adaption method, consider the one-dimensional linear wave equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0 \tag{8}$$

For simplicity, this model problem is discretized with a simple backward difference in time and upwind difference in space, as

$$u_i^{n+1} = u_i^n - \frac{\Delta t_i}{\Delta x_i}(u_i^n - u_{i-1}^n) \tag{9}$$

where the time step for a given cell is governed by

$$\Delta t_i = \nu \Delta x_i \tag{10}$$

where ν is the Courant number.

Shown in Fig. 2 is a typical temporal stencil for an unevenly spaced mesh. The circles represent the solution as stored at the last complete time step n. The dashed squares are the locations of the $n+1$ time level for each point as determined by Eq. (10). (The actual values are shown below the diagram. From this distribution, a global minimum time step can easily be calculated. (For this case $\Delta t_{\min} = 0.10$.) Next, each point is assigned an integer corresponding to its power-of-two multiple of the global minimum time step via,

$$m_i = \text{int}\left[\frac{\log(\Delta t_i / \Delta t_{\max})}{\log(2)}\right] \tag{11}$$

with the values for this example shown on the third line. Computing the new time step for each point is determined by its multiple m_i and the global minimum time step;

$$\Delta t_i' = \Delta t_{\min} 2^{m_i} \tag{12}$$

The newly distributed time steps are represented by the squares, and the values are tabulated in the last line of Fig. 2.

The order of integration is performed according to the sequence shown in Fig. 3. This figure shows the specific order of time integration to be used to integrate the typical temporal stencil from Fig. 2 for indices 1–6. The solution is integrated

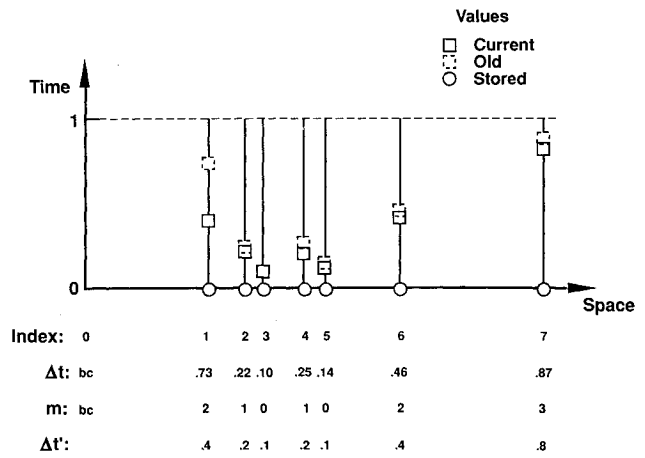

Fig. 2 Example of a temporal stencil for the one-dimensional wave equation.

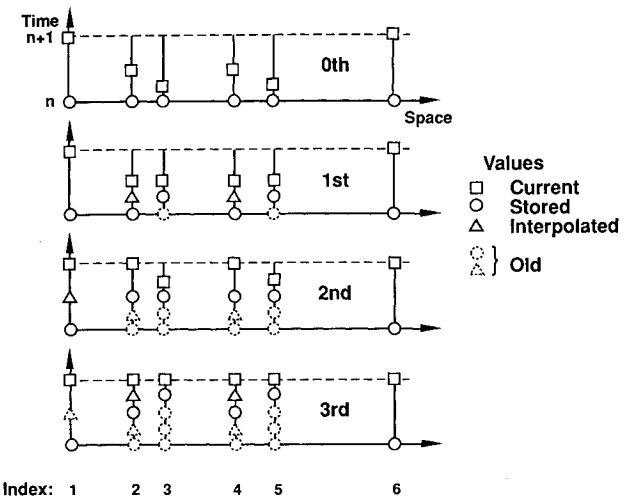

Fig. 3 Order of integration for the sample temporal stencil.

in this order so that all values necessary for the calculations at an intermediate level (triangles) are available at the proper times. On pass 0, the original solution is stored and all of the cells are integrated one time step. For subsequent passes, those cells that have time step multiples m_j, which are themselves multiples of the pass number, are integrated. Before each integration, however, the current solution in that cell is stored so that, if an intermediate value (triangle) is needed by an adjacent cell during future integration passes, it can be determined by interpolation between the two known values (a square and a circle).

Vectorization Considerations

In order to be competitive with the global time stepping procedure, the temporal adaptive procedure needs to be vectorizable. This is accomplished in a simple gathering and scattering operation. During each pass (0, 1, 2, etc.) a set of cells is marked for integration. These cells and the necessary information from neighboring cells are gathered into arrays, then these arrays are passed to the residual calculation routines. After the solution at the new time level is computed for each cell, the array values are scattered back to their respective locations.

Performance Prediction

One qualitative measure of determining whether temporal adaption is suitable for a given problem is to estimate the ratio of the work required for global time marching to the work required for temporally adaptive time marching. In this case, work is defined in terms of the number of residual calculations performed throughout the domain to reach a predetermined time level. For instance, if a particular flowfield or distribution of grid points indicates that all of the grid cells should be marched at a similar time step, then temporal adaption is of no benefit. If, however, only a small fraction of grid cells need to be marched at a very small time step compared to the other grid cells, whether for numerical stability or for resolving the local flow features, then temporal adaption is desirable because otherwise all cells would have to be integrated at the smallest time step, which is prohibitively expensive.

This performance prediction is a very simple one in that the only information it requires is a computational mesh. A time step for each cell is determined according to the stability restriction

$$\Delta t_i = \frac{\nu A_i}{s_i(|u_i| + a_i)} \tag{13}$$

which is based on the Courant number ν, a wave speed $(|u| + a)$, and a characteristic length (A_i/s_i) for each cell. For

this analysis, the wave speed and the Courant number are assumed to be constant throughout the computational domain so that the time step is simply proportional to the characteristic length of the cell

$$\Delta t_i \propto \frac{A_i}{s_i} \tag{14}$$

where A_i is the cell area and s_i is the longest side of a cell. Using this idea, the work required to reach a predetermined time T with global time marching with N points in the computational domain is given by

$$W_{\text{global}} = \sum_{i=1}^{N} \frac{T}{\Delta t_{\min}} = \sum_{i=1}^{N} \frac{T}{(A/s)_{\min}} \tag{15}$$

whereas the work for the temporal adaptive method (termed local for convenience) is given by

$$W_{\text{local}} = \sum_{i=1}^{N} \frac{T}{\Delta t_i} = \sum_{i=1}^{N} \frac{T}{(A_i/s_i)} \tag{16}$$

and so the ratio of the work required becomes

$$\frac{W_{\text{global}}}{W_{\text{local}}} = \frac{\sum_{i=1}^{N} [T/(A/s)_{\min}]}{\sum_{i=1}^{N} [T/(A_i/s_i)]} = \left[\frac{1}{N} \sum_{i=1}^{N} (A/s)_{\min}(A_i/s_i)\right] \tag{17}$$

This formulation does not include the overhead involved with the temporal adaptive scheme, and so this work ratio is an approximate upper bound on the performance.

Results and Discussion

Calculations were performed on several different meshes for a one-dimensional shock-tube problem and a NACA 0012 airfoil to assess the accuracy and efficiency of the temporal adaptive method. The results were obtained with the temporal adaptive method applied to the integration of the Euler and the Navier-Stokes equations. In these calculations, comparisons were made between results obtained using global time stepping, the temporal adaptive algorithm, and experimental data. For the shock-tube problem, comparisons are made with the exact solution. For the viscous solution, only a timing comparison was performed.

Shock Tube

An unsteady one-dimensional shock-tube problem was used as a means of testing the speed and accuracy of the temporal adaptive algorithm. The problem is a good test case since it contains all of the major types of flow features expected to occur in transient problems, including a moving shock wave, an expansion fan, and a contact surface. The test case involved a density ratio of five run on a mesh of random spacing. This is a challenging test for the adaptive algorithm since the local time step will vary considerably from grid cell to grid cell. The code was written so that it would simulate the two-dimensional unstructured mesh in that it contained a similar data structure involving indirect addressing.

Figure 4 shows the resulting density profiles for a 200-point mesh at three sequential times during the calculation (only every third point has been plotted for clarity): A Mach number 1.4 shock wave is followed by a contact surface traveling to the right while an expansion fan moves to the left. These profiles show good agreement between results obtained using the temporal adaptive method and global time stepping, which tends to verify the one-dimensional temporal adaptive method of the present study. Both calculated solutions also agree well with the exact solution. Furthermore, as indicated in Table 1, the solutions obtained using temporal adaption were between 6 and 8 times less expensive than the solutions obtained using global time stepping for 100- and 200-point meshes, respectively.

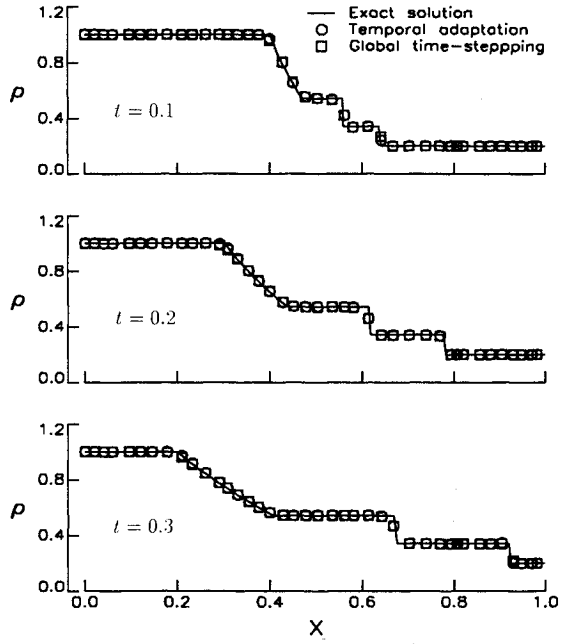

Fig. 4 Comparison of density distributions in a shock tube for a sequence of times (0.1, 0.2, and 0.3) for a 200-point mesh that has random spacing (every third point plotted for clarity).

Unsteady Airfoil

Inviscid results were obtained for a pitching NACA 0012 airfoil to assess the accuracy and efficiency of the temporal adaptive time integration. These results were obtained using various unstructured grids generated via the advancing front method.[3] The finest mesh contains 3828 nodes and 7420 triangles (a partial view is shown in Fig. 1). This mesh had the outer boundary located at a 20-chord radius from the midchord with 209 points around the airfoil. The unsteady results were performed for the airfoil pitching about the quarter chord with an amplitude of $\alpha_1 = 2.51$ deg and a reduced frequency based on semichord of $k = 0.0814$ at $M_\infty = 0.755$ and $\alpha_0 = 0.016$ deg. For this pitching case, the first part of the cycle is dominated by a strong upper surface shock, and the lower surface is predominately subcritical; conversely, the second half of the cycle shows the development of a lower surface shock with predominately subcritical flow above the airfoil. Instantaneous pressure distributions at eight points in time during the fourth cycle of motion are shown in Fig. 5. For each pressure plot, the instantaneous pitching angle $\alpha(\tau)$ and the angular position in the cycle $k\tau$ are noted, where τ is a nondimensional time. The solutions computed using both temporally adaptive integration and conventional integration with global time stepping match very well, demonstrating that temporal adaption preserves the time accuracy of the solution. Both results also compare well with experimental data.[15] Small discrepancies with experiment are evident near the shocked regions of flow as can be expected for an inviscid calculation (i.e., the shock boundary-layer interaction is not resolved).

Timing Comparisons

Timing studies for various meshes were performed using both temporal adaption and global time stepping. An initial estimation of performance was computed using Eq. (17) for each mesh, followed by the actual performance.

Inviscid calculations were performed using the pitching airfoil case as described in the proceeding section. Timing comparisons were computed with three different meshes typically used for an Euler calculation, as shown in Table 2.

The first mesh was generated using Delaunay triangulation of a structured O-mesh, whereas the second two were gener-

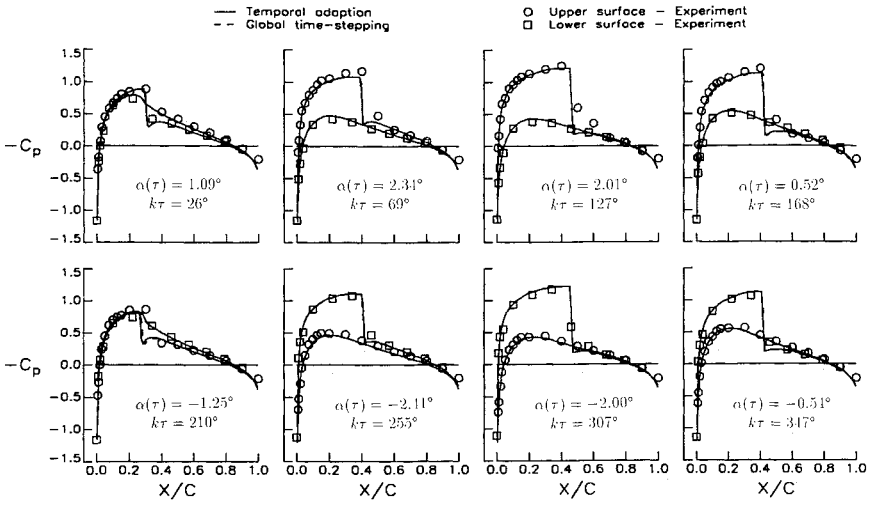

Fig. 5 Comparison of instantaneous pressure distributions for a NACA 0012 airfoil pitching at $M_\infty = 0.755$, $\alpha_0 = 0.016$ deg, $\alpha_1 = 2.51$.

Table 1 Comparison of normalized CPU times for shock-tube meshes

Points	Estimated performance	CPU time ratio, global/local
100	5.7	6.2
200	6.3	8.4

Table 2 Comparison of normalized CPU times for Euler airfoil meshes

Triangles	Estimated performance	CPU time ratio, global/local
3968	42.9	9.2
6466	4.3	3.4
7420	4.5	4.0

Table 3 Comparison of normalized CPU times for Navier-Stokes airfoil meshes

Triangles	Estimated performance	CPU time ratio, global/local
25,964	7.1	7.2
30,720	5.7	6.1

ated with an advancing front method.[3] These results indicate that the simple expression for determining the benefit of temporal adaption provides a good indication of the actual performance, provided that the spatial distribution of the mesh was done in a reasonable fashion. The performance of the first mesh is significantly overpredicted, which is primarily due to the excessive clustering at the leading and trailing edges. Only a handful of cells are being integrated on the smallest time level, which introduces a significant increase in overhead due to the very short vector lengths. Overall, the results indicate that an unsteady, inviscid solution is between 3 and 9 times less expensive using the temporal adaptive method rather than global time stepping.

Timing comparisons were also made for the NACA 0012 airfoil with a freestream Reynolds number of 5×10^3 at $M_\infty = 0.5$ and $\alpha_0 = 0.0$ with two Navier-Stokes meshes. Since this is not a physically unsteady problem, the timings were done by marching both schemes time accurately from freestream initial conditions. Asymptotic time marching was done in lieu of fully unsteady calculations because of resource limitations, but the mesh and the flow provide suitable conditions for this type of study. The predicted and actual performance are shown in Table 3 for two meshes.

The first mesh was generated using an adaptive Delaunay triangulation[6] of a C-type mesh that eliminates high aspect ratio cells in the far field, and the second mesh was generated by simply triangulating a C-type mesh. The results in Table 3 indicate that a Navier-Stokes solution is between 6 and 7 times less expensive using the temporal adaptive method rather than the global time-stepping method.

Concluding Remarks

A temporal-adaptive algorithm for the time integration of the two-dimensional Euler or Navier-Stokes equations was presented. The flow solver involved an upwind flux-split spatial discretization for the convective terms and central differencing for the shear-stress and heat flux terms on an unstructured mesh of triangles along with temporal-adaptive time integration. The temporal adaptive algorithm is a time-accurate local time-stepping procedure that integrates each grid cell near its maximum allowable time step according to numerical stability. Results obtained using the Euler and Navier-Stokes equations demonstrate that temporal adaption is between 4 and 10 times less expensive than the solutions obtained using global time stepping, depending on the distribution of cell sizes throughout the mesh.

Acknowledgments

The work constitutes a part of the first author's M.S. thesis at Purdue University and was supported by the NASA Langley Graduate Aeronautics Program under Grant NAG-1-372. The authors would like to thank Ken Morgan of the University College of Swansea, Wales, and Jaime Peraire of the Imperial College of Science, Technology, and Medicine, London, England, for providing the advancing front method grid generation program that was used to generate the grid for the NACA 0012 airfoil for the Euler solutions, and Dimitri Mavriplis of the Institute for Computer Applications in Science and Engineering, Hampton, Virginia, for one of the meshes used for the Navier-Stokes comparisons. The authors would also like to express thanks to William J. Usab of the School of Aeronautics and Astronautics, Purdue University, West Lafayette, Indiana, for various ideas leading to the method of performance prediction presented within.

References

[1]Jameson, A., "Successes and Challenges in Computational Aerodynamics," AIAA Paper 87-1184, Jan. 1987.

[2]Löhner, R., "Finite Element in CFD: What Lies Ahead," *International Journal for Numerical Methods in Engineering*, Vol. 24, No. 10, 1987, pp. 1741-1756.

[3]Morgan, K., and Peraire, J., "Finite Element Methods for Com-

pressible Flow," Von Kármán Institute for Fluid Dynamics Lecture Series 1987-04, Computational Fluid Dynamics, Von Kármán Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, March 1987.

[4]Batina, J. T., "Vortex-Dominated Conical-Flow Computations Using Unstructured Adaptively-Refined Meshes," AIAA Paper 89-1816, June 1989.

[5]Dannenhoffer, J. F., and Baron, J. R., "Robust Grid Adaption for Complex Transonic Flows," AIAA Paper 86-0495, Jan. 1986.

[6]Mavriplis, D. J., "Adaptive Mesh Generation for Viscous Flows Using Delaunay Triangulation," Institute for Computer Applications in Science and Engineering, Rept. 88-47, Hampton, VA, Aug. 1988.

[7]Oden, J. T., Strouboulis, T., Devloo, P., Spradley, L. W., and Price, J., "An Adaptive Finite Element Strategy for Complex Flow Problems," AIAA Paper 87-0557, Jan. 1987.

[8]Osher, S., and Sanders, R., "Numerical Approximations to Nonlinear Conservation Laws with Locally Varying Time and Space Grids," *Mathematics of Computation,* Vol. 40, No. 164, 1983, pp. 321-336.

[9]Löhner, R., Morgan, K., and Zienkiewicz, O. C., "The Use of Domain Splitting with an Explicit Hyperbolic Solver," *Computer Methods in Applied Mechanics and Engineering,* Vol. 45, Nos. 1-3, 1984, pp. 313-329.

[10]Löhner, R., Morgan, K., and Peraire, J., "Finite Element Methods for High Speed Flows," AIAA Paper 85-1531, July 1985.

[11]Pervaiz, M. M., and Baron, J. R., "Spatio-Temporal Adaption Algorithm for Two-Dimensional Reacting Flows," AIAA Paper 88-0510, Jan. 1988.

[12]Pervaiz, M. M., "Spatio-Temporal Adaptive Algorithm for Reacting Flows," Ph.D. Dissertation, Massachusetts Inst. of Technology, Cambridge, MA, May 1988.

[13]Berger, M. J., "On Conservation at Grid Interfaces," *SIAM Journal of Numerical Analysis,* Vol. 24, No. 5, 1987, pp. 967-984.

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

[15]Landon, R. H., "NACA 0012. Oscillating and Transient Pitching," Data Set 3 in AGARD-R-702 Compendium of Unsteady Aerodynamic Measurements, Aug. 1982.