

N.A.S.A.
TM-112891
046509

Solution-Adaptive Cartesian Cell Approach for Viscous and Inviscid Flows

William J. Coirier and Kenneth G. Powell

Reprinted from

AIAA Journal

Volume 34, Number 5, Pages 938-945



A publication of the
American Institute of Aeronautics and Astronautics, Inc.
370 L'Enfant Promenade, SW
Washington, DC 20024-2518

Solution-Adaptive Cartesian Cell Approach for Viscous and Inviscid Flows

William J. Coirier*

NASA Lewis Research Center, Cleveland, Ohio 44135

and

Kenneth G. Powell†

University of Michigan, Ann Arbor, Michigan 48109-2118

A Cartesian cell-based approach for adaptively refined solutions of the Euler and Navier–Stokes equations in two dimensions is presented. Grids about geometrically complicated bodies are generated automatically, by the recursive subdivision of a single Cartesian cell encompassing the entire flow domain. Where the resulting cells intersect bodies, polygonal cut cells are created using modified polygon-clipping algorithms. The grid is stored in a binary tree data structure that provides a natural means of obtaining cell-to-cell connectivity and of carrying out solution-adaptive mesh refinement. The Euler and Navier–Stokes equations are solved on the resulting grids using a finite volume formulation. The convective terms are upwinded: A linear reconstruction of the primitive variables is performed, providing input states to an approximate Riemann solver for computing the fluxes between neighboring cells. The results of a study comparing the accuracy and positivity of two classes of cell-centered, viscous gradient reconstruction procedures is briefly summarized. Adaptively refined solutions of the Navier–Stokes equations are shown using the more robust of these gradient reconstruction procedures, where the results computed by the Cartesian approach are compared to theory, experiment, and other accepted computational results for a series of low and moderate Reynolds number flows.

I. Introduction

FOR complicated geometries, unstructured grids can be easier to generate than structured grids, which is directly responsible for their increasing popularity. Traditionally unstructured grids, with volume grids comprising triangles in two dimensions and tetrahedra in three dimensions, are typically generated by first discretizing the bounding surfaces and then filling the volume grid by, say, an advancing front method^{1,2} or by point insertion procedures where the site insertion and resulting triangulations are subject to certain geometric constraints.^{3,4} (A thorough review of current unstructured mesh generation and flow solution algorithms is available in Refs. 5 and 6.) The volume mesh generation for these approaches is constrained by the boundary surface discretization: Boundary cells are required to have faces that exactly match the discretized boundary surface. In the Cartesian approach considered here, the volume grid and surface description are not coupled in this manner. The computational boundaries are described functionally, and are cut out of the automatically generated, Cartesian cell-based volume grid, yielding polygonal cells near the boundaries, with a geometric, subcell resolution of the boundary. This gives a smooth grid away from boundaries, comprising mostly square, unit aspect ratio (Cartesian) conservation volumes and a nonsmooth grid near computational boundaries, where the cell cutting has faithfully preserved the geometry. The ease in grid generation has come at the cost of grid nonsmoothness near computational boundaries that necessitates special care in the treatment of the cut cells and cell-to-cell refinement boundaries.

The Cartesian cell-based approach has been used for solving a variety of problems, and variants of the approach have proven to be quite useful for computing both unsteady^{7–13} and steady^{14–20}

adaptively refined solutions to the Euler equations and for solutions to the transonic full potential equation.²¹ Recently, the usefulness of the automated gridding and adaptive mesh refinement capabilities of the solver have been demonstrated for adaptively refined solutions of the Euler equations about large aircraft configurations^{22,23} and for store separation problems.²⁴ The Cartesian cell-based approach has been coupled with a prismatic mesh approach in Ref. 25 and recently demonstrated for the Euler equations,²⁶ where a cursory demonstration for solving the Navier–Stokes equations was also made. In Ref. 9, the approach was also demonstrated for solving the Navier–Stokes equations, but the results there were only shown for a few geometrically simple cases. In Ref. 27 a more structured-like grid approach was taken for both the grid and underlying numerics, where the results indicated a sensitivity to grid smoothness. The recent philosophy of treating the Cartesian approach in a more unstructured-like manner, or the preferred hierarchically structured manner for the Euler equations, has proven to alleviate some of the problems found from a more traditional, structured-like grid approach. A hierarchically based data structure, as implemented here, allows adaptive mesh refinement by cell enrichment and is also used to determine cell-to-cell connectivity. This more modern approach has demonstrated the ability to compute adaptively refined solutions of the Euler equations with minimal user intervention and has provided an impetus for the work reported here. In this paper, the hierarchically structured, Cartesian cell-based methodology is investigated for obtaining adaptively refined solutions to the Euler and Navier–Stokes equations.

II. Grid Generation Procedure

The grid is generated by the recursive subdivision of a single cell, and during the creation of the grid, the hierarchical relation between newly created cells and their parents is stored in a binary tree data structure. The cut cells, which are the background Cartesian cells cut into polygons, are created automatically using many concepts borrowed from computer graphics applications. Since they are hierarchically related to their Cartesian parents, the cut cells are also stored in the tree. This procedure of cell cutting is a subject unto itself and its robustness is absolutely crucial for the utility of the approach. The cell cutting methodology used here is based on a polygon clipping algorithm,²⁸ where a subject polygon (the body of interest) is

Received April 3, 1995; revision received Sept. 8, 1995; accepted for publication Sept. 9, 1995. Copyright © 1995 by the American Institute of Aeronautics and Astronautics, Inc. No copyright is asserted in the United States under Title 17, U.S. Code. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Governmental purposes. All other rights are reserved by the copyright owner.

* Aerospace Engineer, Computational Fluid Dynamics Branch, Mail-Stop 5-11, 21000 Brookpark Road.

† Associate Professor, Department of Aerospace Engineering, 1320 Beal Avenue. Member AIAA.

clipped against a convex clipping polygon, which for this case is the Cartesian cell. The subject polygon can be formulated to include arbitrary functional descriptions of the boundary faces by the use of general basis functions defined between fixed control points, as is implemented here. This particular clipping operation yields the Boolean intersection between the clipping polygon and the subject polygon. The clipping operation will yield the correct cells when the subject polygon describes the outer boundary of a flow domain, but needs to be modified slightly when the subject polygon describes an inner boundary. When this is the case, the region needed is recovered from the clipping operation using a list directed vertex insertion procedure. Details behind the grid generation are presented in more detail in Ref. 29. Once a suitable geometric description of the computational boundaries is made, the grid generation is automatic, and since the procedure is recursive and tree based, it is also efficient.

The use of this particular grid and data structure easily allows adaptive mesh refinement and, away from bodies, yields smooth grids. The binary tree data structure provides a logical means of finding cell-to-cell connectivity by logic-based tree traversals and allows a straightforward means to perform mesh refinement and coarsening via tree branch growth and pruning. There are many niceties afforded by this data structure and grid setup. The grid hierarchy is amenable to multigrid³⁰ and provides a natural means of domain decomposition that may be well suited for coarse grain parallel computations.

III. Solution of the Euler and Navier–Stokes Equations Using a Cartesian, Cell-Based Approach

The compressible Euler and Navier–Stokes equations are solved using a cell-centered, finite volume approach. The laws describing the conservation of mass, momentum, and energy in two dimensions for a control volume of area Ω with boundaries $\partial\Omega$ whose outward pointing unit normals are \hat{n}_j , (written in tensorial notation) is

$$\Omega \frac{\partial q_i}{\partial t} + \oint_{\partial\Omega} F_{ij} \hat{n}_j d(\partial\Omega) = 0 \quad (1)$$

The vector of conserved variables is $\mathbf{q} = (\rho, \rho u, \rho v, \rho E)$, and the flux tensor is split into purely convective and diffusive components, $F_{ij} = F_{ij}^c + F_{ij}^d$. The convective contributions to the conservation laws are

$$F_1^c = \begin{pmatrix} \rho u \\ \rho u^2 + P \\ \rho uv \\ \rho uH \end{pmatrix} \quad F_2^c = \begin{pmatrix} \rho v \\ \rho uv + P \\ \rho v^2 \\ \rho vH \end{pmatrix} \quad (2)$$

and the diffusive components are

$$F_1^d = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - q_x \end{pmatrix} \quad F_2^d = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ u\tau_{yx} + v\tau_{yy} - q_y \end{pmatrix} \quad (3)$$

where the shear stress tensor and heat flux vector are

$$\begin{aligned} \tau_{xx} &= 2\mu \frac{\partial u}{\partial x} - \frac{2}{3}\mu \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \\ \tau_{xy} = \tau_{yx} &= \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \quad q_x = -k \frac{\partial T}{\partial x} \\ \tau_{yy} &= 2\mu \frac{\partial v}{\partial y} - \frac{2}{3}\mu \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \quad q_y = -k \frac{\partial T}{\partial y} \end{aligned} \quad (4)$$

The equations are closed by providing a thermodynamic equation of state relating the conserved variables to the pressure, $P = (\gamma - 1)\rho e$, and Sutherland's law to relate the laminar viscosity and the thermal conductivity to the temperature. Only laminar solutions are considered here.

IV. Solution Adaptive Mesh Refinement

Each level of adaptive mesh refinement comprises two stages. In the first stage, refinement criteria are constructed for all cells on the mesh, and then in the second stage, cells are tagged for refinement or coarsening based on this criteria. After the mesh is enriched, a new calculation is made, converging the solution to a steady and hopefully more accurate solution. This process of refining the grid and converging the solution on the new grid is repeated in an automated fashion, a set number of times, until a given level of refinement is achieved.

The refinement criteria and grading procedure is based on that presented in Refs. 18 and 31, and is briefly shown here. The procedure of refinement and coarsening of the cells relies upon a statistical description of the cell size weighted velocity divergence and curl. The local velocity divergence is used to detect compressive phenomena, whereas the velocity curl is used to detect shear. Each of these is weighted by the local cell size so that smaller cells contribute less to the overall weighting, as suggested in Ref. 32. That is, for a cell with a characteristic dimension h , the compressibility and shear detectors are

$$\tau_C = |\nabla \cdot \mathbf{u}| h^{\frac{3}{2}} \quad \tau_R = |\nabla \times \mathbf{u}| h^{\frac{3}{2}} \quad (5)$$

Cells are refined or coarsened if the variance of the compressibility and rotationality detectors about zero is above or below a specified threshold. Cells are refined if

$$\tau_R > \sigma_R \quad \text{or} \quad \tau_C > \sigma_C \quad \text{and} \quad h > h_{\min} \quad (6)$$

where a minimum cell size h_{\min} is a user specified constant, and they are coarsened if

$$\tau_C < (\sigma_C/10) \quad \text{and} \quad \tau_R < (\sigma_R/10) \quad (7)$$

By using the binary tree data structure, cell refinement and coarsening is accomplished by performing growth and pruning operations, respectively, on the tree.

V. Convective Flux Discretization

The convective flux terms are discretized using an upwind formulation. A linear reconstruction of cell-averaged data is used to provide input states to a numerical flux function, yielding the flux through cell-to-cell interfaces. The numerical fluxes are computed in an upwind fashion using an appropriate approximate Riemann solver. These fluxes are then used to perform a flux balance on the conservation volume that is then used to advance the conserved variables in time. The procedure follows standard practice for a finite volume scheme. The solution procedure can be broken up into three stages; reconstruction, flux construction, and evolution.

The variation of the cell primitive variables in each cell is reconstructed using a linear reconstruction procedure, in the spirit of MUSCL interpolation, based on the minimum-energy reconstruction presented by Barth and Fredrickson.³³ The minimum-energy reconstruction minimizes the Frobenius norm of the differences between the cell averages of the reconstructing polynomial and the cell averages of the support set. This reconstruction procedure is K -exact, in the sense that if a polynomial is cell averaged on the mesh, the reconstruction procedure reconstructs the same polynomial. For a cell averaging operator defined as

$$A(f) = \frac{\int f dA}{\int dA} \quad (8)$$

the procedure minimizes the norm S ,

$$S = \sum_n \omega_n [A_n(u^K - u_n)]^2 \quad (9)$$

with respect to the expansion about the object cell using zero mean polynomials,

$$u^K = \bar{u} + \sum_j \alpha_j \Psi_j(x, y) \quad (10)$$

where the cell averages of the zero mean polynomials Ψ_j are, by definition, identically zero. This results in a linear system for the unknowns $L_{ij}\alpha_j = b_i$, where

$$\begin{aligned} L_{ij} &= \sum_n \omega_n A_n(\Psi_i) A_n(\Psi_j) \\ b_i &= \sum_n \omega_n (\bar{u}_n - \bar{u}) A_n(\Psi_i) \end{aligned} \quad (11)$$

The L_{ij} is dependent only on the geometry, so it can be inverted beforehand. For the linear reconstructions used here, this results in a symmetric 2×2 matrix, constructed from the zero mean polynomials $\Psi = [(x - \bar{x}), (y - \bar{y})]$. For the calculations shown here, no gradient limiting is performed.

The inviscid numerical fluxes may be computed using a variety of approximate Riemann solvers: it is a simple matter to supply a different numerical flux function by replacing the approximate Riemann solver in the flow solver. For the work shown here, most of the computations have been performed using the advective upstream splitting method (AUSM) scheme of Liou and Steffen.³⁴ This novel flux function combines the efficiency of flux vector splitting with the accuracy of flux difference splitting. The derivation and use of this flux function is available in Ref. 34.

For simplicity, the semidiscrete form of the equations are advanced in time using a multistage scheme. A spatially varying time step is used, with a blending of inviscid and viscous time-step constraints. These are necessary since there is typically a many order variation in cell size across the mesh due to cell refinement and cutting, and a number of cut cells are viscous time step limited. The time step is formulated as

$$\Delta t = C_n \left(\frac{\Delta t_i \Delta t_v}{\Delta t_i + \Delta t_v} \right) \quad (12)$$

where the inviscid and viscous time step restrictions are obtained by evaluation of model equations for the convective and diffusive processes. For an arbitrary cell of area Ω , where each face has an outward pointing unit normal \hat{n} and face length of ΔS , the inviscid time step is found by

$$\frac{\Delta t_i}{\Omega} = \frac{1}{\sum_{\text{faces}} (|u\hat{n}_x + v\hat{n}_y| + a) \Delta S_{\text{face}}} \quad (13)$$

The viscous time step is represented as

$$\frac{\Delta t_v}{\Omega} = \frac{K_v}{-\alpha_0(\Omega\mu/\rho Re_\infty)} \quad (14)$$

The K_v in Eq. (14), inspired by Ref. 35, is taken to be $K_v = 0.25$. The term α_0 is the coefficient for the object cell in the discrete Laplacian obtained using the viscous gradient reconstruction procedure (Sec. VII). For the Green–Gauss–based reconstruction of the viscous gradients described in Sec. VII, the resulting coefficient of the object cell for the Laplacian stencil is

$$\alpha_0 = -\frac{1}{2\Omega} \sum_{\text{faces}} \frac{(\Delta S_f)^2}{\Omega_{\text{aux}}} \quad (15)$$

where the vertex weightings contributions to the α_0 have been neglected, ΔS_f is the length of the face, and Ω_{aux} is the area of the integration covolume.

A generic three-stage multistage scheme with stage coefficients (0.18, 0.5, 1.0) is used to advance the solution from the n th to the $(n + 1)$ th time level.

VI. Validation and Demonstration of the Euler Solver

The Cartesian, cell-based approach has been demonstrated extensively for a wide variety of inviscid flows,^{14–20,22} and has been carefully assessed for accuracy.³⁶ In Ref. 36, the approach was shown to compute transonic flows with a global second-order accuracy and a local accuracy between first- and second-order. The adaptively refined solution about a subcritical, multicomponent airfoil is computed here, demonstrating the adaptive-mesh refinement and mesh generation capability.

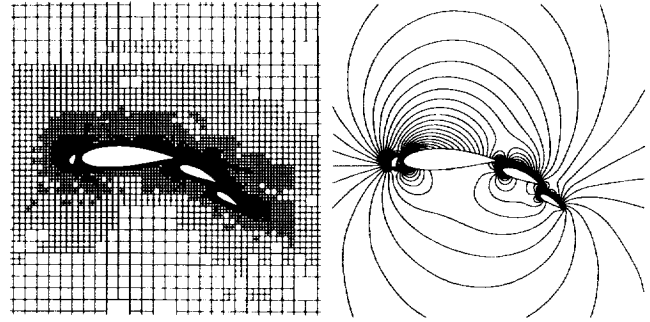


Fig. 1 Grid and pressure contours, Suddhoo–Hall airfoil.

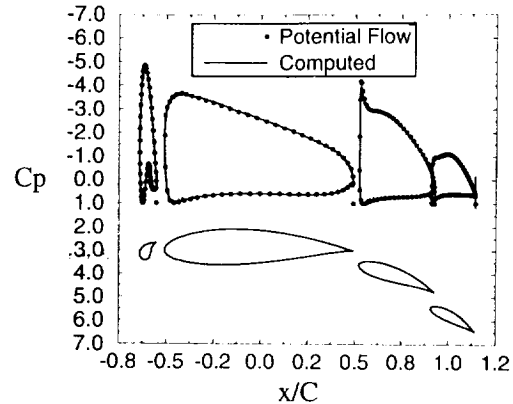


Fig. 2 Computed and exact pressure coefficients.

This four-element airfoil has been included in a series of test cases at the 1994 Institute for Computer Applications in Science and Engineering/NASA Langley Research Center Workshop on Adaptive Grid Methods. The geometry corresponds to that obtained by a conformal mapping technique by Suddhoo and Hall,³⁷ which has also yielded surface pressure data. The airfoil geometry has been curve fit using a cubic spline and made available on the world wide web by the workshop organizers. The freestream conditions corresponding to this case are $M_\infty = 0.2$ and $\alpha = 0$ deg. The mesh is generated automatically and adaptive-mesh refinement is performed for three levels beyond the base grid. A closeup of the final grid and the pressure contours are shown in Fig. 1, whereas the comparison with the analytical solution is shown in Fig. 2.

VII. Viscous Flux Discretization

The essence of the cell-centered viscous flux formulation is the reconstruction of the gradients of the velocity and temperature at the cell interfaces from the cell-averaged data of the cells around it. Once this has been completed, a numerical quadrature is performed over the cell faces, yielding the desired viscous fluxes through the interfaces. For cell-centered schemes, there are prevalently two separate classes of the viscous gradient reconstructions that have been analyzed in the Cartesian mesh framework in Ref. 29. The first and most widely used class is based on an application of the divergence theorem to a covolume surrounding the face where the flux is desired. These types of reconstructions are classified as Green–Gauss type reconstructions. Four reconstructions of this type have been analyzed in Ref. 29 and are delineated amongst themselves by the reconstruction covolumes and the procedures used to obtain the data at the vertices of the covolumes. A different class of gradient reconstructions based on expanding a polynomial about the face midpoint and then differentiating the polynomial to obtain the gradients has been proposed by Mitchell and Walters.³⁸ Linear and quadratic reconstructing polynomials are found using a Lagrangian type of interpolation and are also examined in detail in Ref. 29.

In Ref. 29, these six schemes were analyzed for quality and positivity by local Taylor-series expansions of the stencils created for the Laplace equation on grids representative of the Cartesian approach. The importance of the K exactness of the interface gradient

reconstructions was shown. The analysis indicated that for arbitrarily distorted grids the only means to obtain a conservative, first-order accurate discretization requires that the gradients be found from quadratically preserving functions. It was also shown that if the gradient reconstruction procedure is not at least linearity preserving, stencils can be obtained that preclude grid convergence and will actually have an error that increases with mesh refinement, termed here as being mesh divergent. The robustness of all of the schemes on the distorted meshes caused by mesh refinement was gauged using a positivity analysis. Nonpositivity of a viscous reconstruction scheme can inhibit convergence and violates the discrete interpretation of the continuous maximum principle entertained by elliptic solutions.

The Taylor-series-based analysis in Ref. 29 presented no clear choice as to the best procedure of the two classes of reconstructions: divergence or polynomial based. None of the divergence-based schemes gave first-order accurate Laplacians on all of the model meshes and all created stencils that tended to be nonpositive. The quadratic-polynomial-based approach yielded first-order accurate stencils on all of the model meshes, yet these stencils tended to be more nonpositive and, hence, more destabilizing, when compared to the divergence-based approach. To determine which of the two viscous reconstruction schemes was better suited, the better of the divergence-based schemes and the quadratic-polynomial type scheme were both used to actually compute a series of low to moderate Reynolds number, adaptively refined solutions using the Cartesian approach.

The linearity preserving, diamond path scheme was chosen to represent the divergence-based schemes. This reconstruction is frequently referred to as a diamond path scheme since in two dimensions the path formed for the reconstruction covolume is in the shape of a diamond and is formed by joining the cell centers and face endpoints. Referring to Fig. 3, the data at the centroids L and R are known exactly, whereas the data at the vertices must be obtained by some interpolation procedure. In Ref. 39 a linearity preserving interpolation procedure is shown that finds the data at a vertex based on data surrounding it and is typically referred to as a linearity preserving Laplacian weighting. This procedure is necessary to ensure the viscous gradient reconstruction itself is also linearity preserving, by providing data at the vertices of the reconstruction covolume in a linearly exact manner. As indicated by the analysis,²⁹ although neither scheme guaranteed positive stencils, the diamond path scheme could be viewed as representing the more positive, yet less accurate of the two. In Ref. 29, a discrete accuracy and positivity analysis on the grids showed that the inconsistency incurred by the diamond path schemes is low whereas the computed results from both schemes were nearly identical. The inconsistency incurred by the diamond-path scheme is small, in a global sense, due to the regularity of the Cartesian grids, where the reconstruction procedure yields the desired stencils on most of the cells. The quadratic-polynomial-based scheme guaranteed consistency, which was also shown computationally in the discrete accuracy analysis, but yielded the most nonpositive stencils, which made the quadratic scheme the least robust. For arbitrarily cut cells, neither scheme was as robust as would be desired, but the diamond-path scheme was shown to be the more positive. Based on these conclusions, the linearity preserving, divergence-theorem based viscous reconstruction procedure is chosen as the most suitable approach.

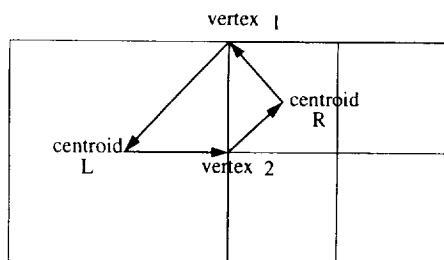


Fig. 3 Sample diamond path integration for a locally refined Cartesian mesh.

VIII. Viscous Computations

The adaptively refined, Cartesian approach is demonstrated in the following sections for low and moderate Reynolds number flows and is compared to theory, experiment, and accepted computational results. All computations shown here were performed using the non-limited, least-squares type reconstruction, the AUSM inviscid flux function, and the linearity-preserving, divergence-based viscous reconstruction.

A. Laminar Driven Cavity Flow

The laminar flow inside a square, driven cavity is computed and compared to the computed results of Ghia et al.⁴⁰ In Ref. 40, an incompressible formulation of the Navier–Stokes equations was solved using an implicit multigrid method. To compare with these incompressible results, the Mach number used here is taken to be $M_{\text{lid}} = 0.1$. The flow corresponding to two Reynolds numbers, based on lid velocity and cavity depth, were computed and compared to the tabulated results.

1. $Re = 100$

A uniform base grid of 1024 cells (32×32) is generated, and three levels of adaptive mesh refinement beyond the base grid are obtained. Adaptive mesh refinement improves the solution slightly, but the initial solution is quite good. Figure 4 shows the computed u - and v -velocity profiles along vertical and horizontal lines through the geometric center of the cavity. Although not shown here, there are secondary vortices predicted in the lower corners of the cavity: these vortices are not isolated by the refinement strategy, but rather, the large gradients of the velocity in the upper corners are refined. Overall though, the solution is predicted well with the initial grid.

2. $Re = 400$

This case is similar to the previous, although the Reynolds number is now 400. A coarse base grid is generated, and three levels of adaptive mesh refinement are performed beyond the base level. A comparison to the computational data of Ghia et al.⁴⁰ is shown for the u and v velocities on lines through the geometric center in Fig. 5. Here, the solution on the coarse, base grid is poor, and the solution is improved through the adaptive mesh refinement. The solution obtained is good on the final grids, and the mesh refinement is shown to have improved the solution.

B. Laminar Flow over a Backward Facing Step

The laminar flow over a backward facing step at two Reynolds numbers is next computed. The computed results are compared to the experimental data of Armaly et al.⁴¹ at the laminar Reynolds numbers. A parabolic velocity profile is specified at the inflow, and the exit pressure is specified. This ensures that the proper pressure gradient and mass flow is imposed on the flow. The results are qualitatively similar for both Reynolds numbers, although the higher

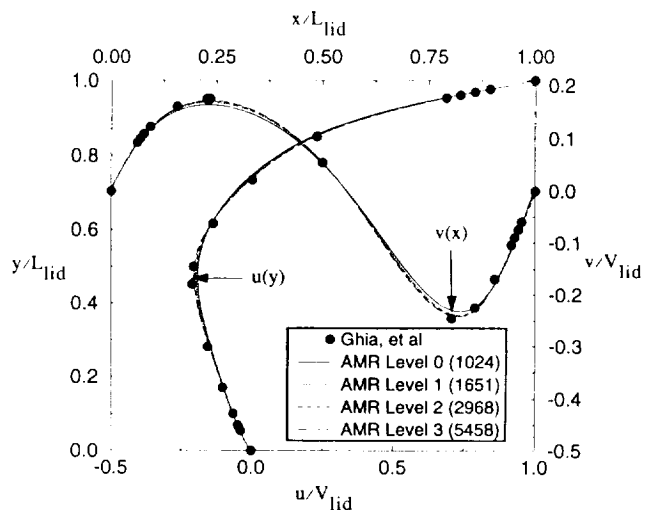


Fig. 4 Computed u - and v -velocity profiles through cavity centroid: $Re = 100$.

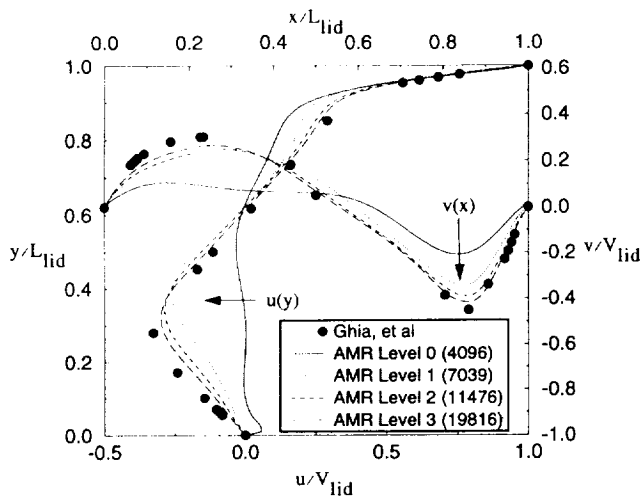


Fig. 5 Computed u - and v -velocity profiles through cavity centroid: $Re = 400$.

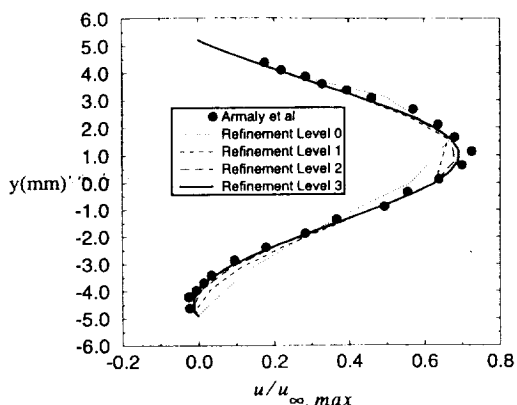


Fig. 6 Computed and experimental u velocities at a location 2.55 step heights downstream of back step, $Re = 100$.

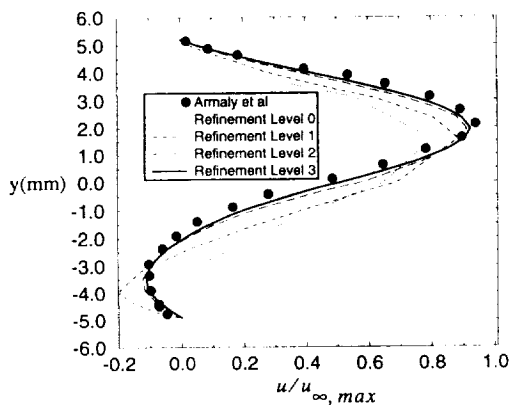


Fig. 7 Computed and experimental u velocities at a location 2.55 step heights downstream of back step, $Re = 389$.

Reynolds number flow indicates a mild separation on the upper wall. The computations for both Reynolds numbers begins with a very coarse mesh of 968 cells. Adaptive mesh refinement is performed for three levels beyond the initial mesh. Figures 6 and 7 compare the u -velocity profiles through refinement for both Reynolds numbers. As is seen from the figures, the initial solutions are poor, due to the coarse grids, but the adaptive mesh refinement automatically improves the solution quality. Although not shown here, the computed skin friction is quite oscillatory and is too noisy to discern an exact reattachment point.

C. Developing Laminar Flow over a Flat Plate

The developing, laminar flow over a flat plate that is aligned with the freestream is computed and compared to the theoretical solution.

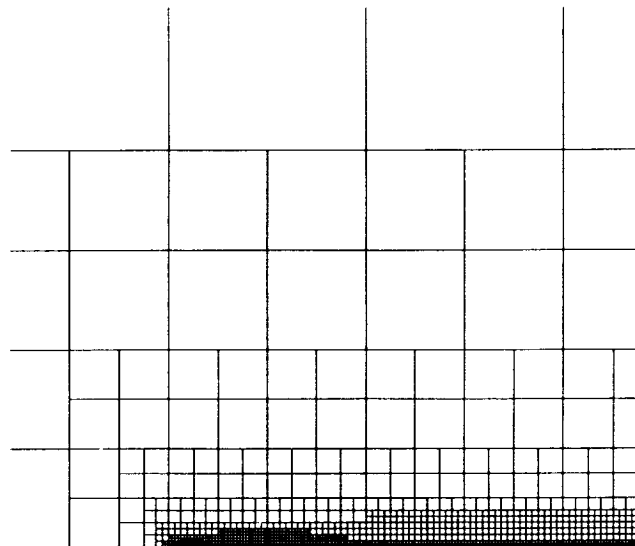


Fig. 8 Closeup initial grid for axis-aligned case.

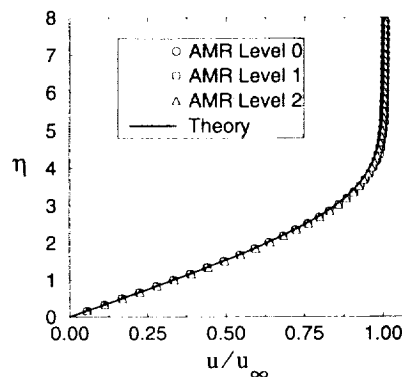


Fig. 9 Computed u velocity, axis-aligned plate.

To bring out the effect of cell cutting, two different sets of grids are used, where in the first (axis-aligned) case, the base axes of the grid are aligned with the plate, whereas in the second, the plate has been rotated 30 deg with respect to the x -axis. A uniform flow, aligned with the plate surface, is imposed ahead of the plate, and the flow is allowed to develop along the plate. Conditions are set so that the Reynolds number based on plate length is 10,000. The Mach number is taken to be $M_\infty = 0.2$, which eliminates the need for any compressibility transformation to compare to theory.

1. Axis-Aligned Plate

Figure 8 shows the closeup of a base grid near the leading edge where the initial resolution of the grid is made according to an estimated normal velocity scale variation deduced from theory. Two levels of adaptive mesh refinement are made beyond the base grid shown in Fig. 8, and the computed velocity component parallel to the plate and the skin friction is shown in Figs. 9 and 10, respectively. The quality of the computed results and the smoothness of the skin friction in Fig. 10 is due to the smoothness of the grid and to the adequate initial resolution given on the base grid. Since the root cell of the grid system has been located so that no cut cells are introduced along the surface of the flat plate, the only nonsmoothness is introduced near cell refinement boundaries. As seen in the figures, the mean flow quantities are predicted well by the proposed approach.

2. Nonaxis-Aligned Plate

Here, the plate is rotated 30 deg about the base axes and the identical flow is computed, bringing to light the effect of introducing cut cells along the plate boundary. The initial grid is shown in Fig. 11. To obtain a converged solution, it proved necessary to zero the weights of the linearity-preserving vertex weightings for all cut cells and

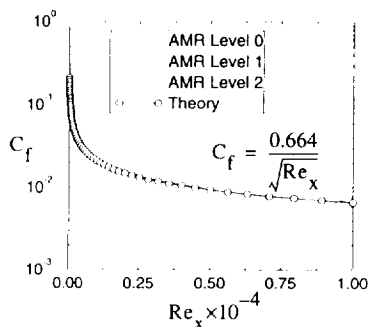


Fig. 10 Computed skin friction, axis-aligned plate.

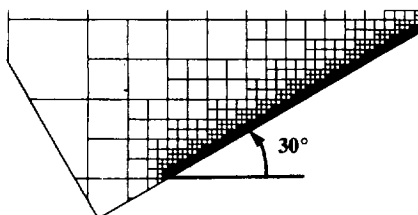


Fig. 11 Closeup of grid for rotated plate.

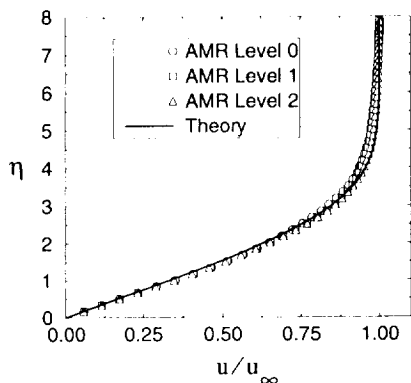


Fig. 12 Computed u velocity, rotated plate.

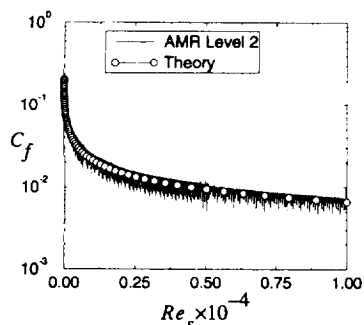


Fig. 13 Computed skin friction, rotated plate.

their neighbors, resulting in a less accurate near-wall treatment of the viscous fluxes. Regardless of this reduction in accuracy of the near-wall viscous discretization, the mean flow quantities are predicted well, as shown in Fig. 12. The skin friction is shown to be very oscillatory (Fig. 13), where the use of a higher-order reconstruction for only boundary faces smooths the skin friction somewhat, although it is still extremely nonmonotone.²⁹

D. Laminar Flow Through a Branched Duct with Cooling Fins

To demonstrate the approach for complex geometries, the flow in a stylized duct is computed. This duct geometry corresponds to an experiment conducted at the NASA Lewis Research Center designed to simulate, in a simplified manner, the flow in the cooling passages of a turbine blade.⁴² The calculations shown here in no way try to

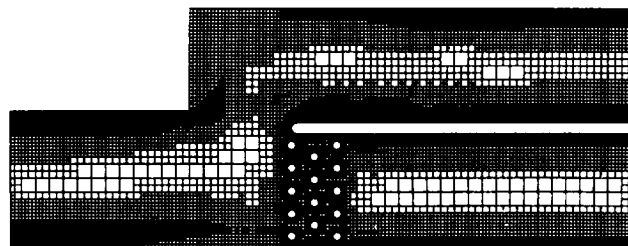


Fig. 14 Adapted grid, branched duct case.

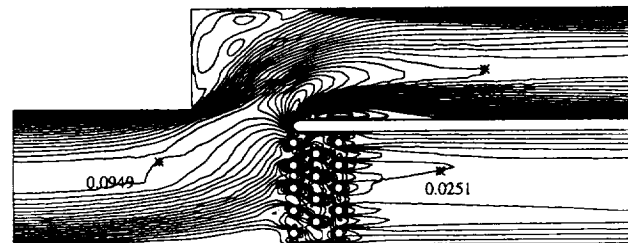


Fig. 15 Total velocity contours on adapted grid; numbers indicate contour levels.

simulate the experiment: the experimental conditions correspond to a three-dimensional, turbulent flow, whereas the calculations shown here are laminar. The single, entrance flow path branches into two passages, a primary and secondary passage, where there are 14 circular pin fins providing blockage in the secondary passage.

A fully developed profile is introduced at the inflow, and the flow is diverted into the primary passage by the blockage introduced by the pin fins in the secondary passage. Two different Reynolds numbers based on pin diameter and maximum velocity in the fully developed inflow profile were computed in Ref. 29. Only the lower Reynolds number results are shown here for brevity, where the Reynolds number based on maximum inflow velocity and pin fin diameter is $Re = 25$. Only one level of adaptive mesh refinement beyond the base grid level was obtained, due to stability problems in the second refinement level, even with the cut cell viscous flux modifications described in Sec. VIII.C. The final adapted grid and contours of total velocity are shown Figs. 14 and 15, respectively.

The basic flow features predicted here correspond to those in the experiment, although some important features are grossly under-resolved, such as the individual pin-fin wakes. The primary passage separation and reattachment along the splitter plate and the separation anchored at the back step portion are both properly predicted, as well as the upstream influence of the pin blockage upon the lower wall flow. Although many levels of refinement were not achieved, the larger scale flow features were adequately predicted, and their resolution was automatically improved by the mesh refinement procedure.

IX. Conclusions

An adaptively refined, finite volume solution procedure for the Euler and Navier–Stokes equations using a Cartesian, cell-based approach has been presented. The grid generation procedure is automated and is able to generate grids about complicated geometries with minimal user intervention. This strategy is based on the recursive, isotropic subdivision of a Cartesian cell that encompasses the domain. Where the resulting Cartesian cells span boundaries of the domain, polygonal cells are cut out of the background Cartesian mesh using a modified polygon-clipping algorithm. The hierarchy of the grid generation process is stored in a binary tree that provides a natural means of finding cell-to-cell connectivity and provides a straightforward means of adaptively refining the grid. The utility of the Cartesian approach has come about by sacrificing grid smoothness near boundaries to achieve a near automation of the grid generation. Application of this approach for the Euler equations has shown promise and has provided an impetus for evaluating the approach for the Navier–Stokes equations.

A finite volume, upwind-based scheme has been selected for treatment of the convective terms in the Euler/Navier–Stokes equations and has been implemented in the Cartesian, cell-based framework.

Extension of the Cartesian approach for solving the Navier–Stokes equations has necessitated a careful investigation of candidate viscous flux formulations, which has been summarized here. The two viscous flux formulations that were investigated represent, respectively, a divergence-theorem-based reconstruction procedure and a quadratic-polynomial-based reconstruction procedure. Analysis and practice indicated that the divergence-theorem-based procedure, where data at the face vertices were found in a linearity-preserving manner, was adequate and was also the more robust of the two, although neither could guarantee positivity on arbitrarily distorted meshes. The nonpositivity induced by the cut cells caused convergence problems and yielded quantities derived from wall gradient data, such as skin friction and heat transfer, for the most part unusable, although the mean values of these quantities were shown to be adequate. The divergence-based scheme, commonly referred to as a diamond path reconstruction using a linearity-preserving weighting, has been used here to compute adaptively refined solutions to a variety of low and moderate Reynolds number flows. Comparisons were made to accepted computational results, to experimental data and to theory for a range of problems, where the Cartesian, cell-based approach is shown capable to accurately predict these flows.

An important conclusion from this study is that current viscous flux functions rely heavily upon grid smoothness and orthogonality to obtain accuracy and positivity, and that these two properties are in direct competition with each other: it is difficult to get both positive and accurate viscous stencils on nonorthogonal/nonsmooth meshes. On the interior of the mesh, the majority of the cells produce accurate stencils, due to the regularity and orthogonality of the Cartesian cells. These qualities do not hold at refinement boundaries and are violated near cut cells. The nonsmoothness of the cut cells have negative implications for the smoothness of aerodynamic parameters that rely on derivative quantities at walls, such as skin friction and heat transfer, and can also have a detrimental effect on convergence.

Another drawback of the approach is due to the anisotropy of the Navier–Stokes equations at high Reynolds numbers and the isotropy of the Cartesian cells: stretched, anisotropic cells are needed for the economic resolution of higher Reynolds number flows. A promising approach currently under investigation²⁶ hopes to alleviate these shortcomings, by combining the Cartesian approach with locally stretched, prismatic cells. This approach appears promising, in that it can use the type of stretched, nearly orthogonal cells in viscous layers and move the cut, Cartesian cells into a convection dominated region. Regardless of these comparatively negative findings, the approach shown here has proven to be useful and can give accurate, automatically gridded, adaptively refined solutions of the Navier–Stokes equations for low and moderate Reynolds number flows.

Acknowledgments

For the multielement airfoil, the authors gratefully acknowledge the Institute for Computer Applications in Science and Engineering workshop organizers for the geometry, Kyle Anderson of NASA Langley Research Center for the Suddho–Hall reference, and Darren DeZeeuw of the University of Michigan for the electronic form of the actual pressure coefficient data. The geometry for the coolant passage and the branched duct was graciously supplied by Erlendur Steinthorsson of the Institute for Computational Mechanics. The first author gratefully acknowledges many discussions with P. C. E. Jorgenson and C. J. Steffen Jr.

References

- ¹Pirzadeh, S., "Recent Progress in Unstructured Grid Generation," AIAA Paper 92-0445, Jan. 1992.
- ²Mavripilis, D. J., "An Advancing Front Delaunay Triangulation Algorithm Designed for Robustness," Inst. for Computer Applications in Science and Engineering, ICASE Rept. 92-49, Oct. 1992.
- ³Barth, T. J., "Steiner Triangulation for Isotropic and Stretched Elements," AIAA Paper 90-0013, Jan. 1990.
- ⁴Jorgenson, P. C. E., "An Implicit Numerical Scheme for the Simulation of Internal Viscous Flows on Unstructured Grids," Ph.D. Thesis, Dept. of Mechanical Engineering, Iowa State Univ., Ames, IA, 1993.
- ⁵Anon., "Special Course on Unstructured Grid Methods for Advection Dominated Flows," AGARD-R-787, 1992.
- ⁶Venkatakrishnan, V., "A Perspective on Unstructured Flow Solvers," AIAA Paper 95-0667, Jan. 1995.
- ⁷Berger, M. J., and Colella, P., "Local Adaptive Mesh Refinement for Shock Hydrodynamics," *Journal of Computational Physics*, Vol. 82, 1989, pp. 64–84.
- ⁸Berger, M. J., "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations," *Journal of Computational Physics*, Vol. 53, 1984, pp. 484–512.
- ⁹Quirk, J. J., "An Adaptive Grid Algorithm for Computational Shock Hydrodynamics," Ph.D. Thesis, College of Aeronautics, Cranfield Inst. of Technology, 1991.
- ¹⁰Quirk, J. J., "An Alternative to Unstructured Grids for Computing Gas Dynamic Flows Around Arbitrarily Complex Two-Dimensional Bodies," Inst. for Computer Applications in Science and Engineering, ICASE Rept. 92-7, Feb. 1992.
- ¹¹Bayyuk, S. A., Powell, K. G., and van Leer, B., "An Algorithm for the Simulation of 2-D Unsteady Inviscid Flows Around Arbitrarily Moving and Deforming Bodies of Arbitrary Geometry," AIAA Paper 93-3391, July 1993.
- ¹²Chiang, Y. L., "Simulation of Unsteady Inviscid Flow on an Adaptively Refined Cartesian Grid," Ph.D. Thesis, Dept. of Aerospace Engineering, Univ. of Michigan, Ann Arbor, MI, 1992.
- ¹³Pember, R., Bell, J., Colella, P., Crutchfield, W., and Welcome, M., "Adaptive Cartesian Grid Methods for Representing Geometry in Inviscid Compressible Flow," AIAA Paper 93-3385, July 1993.
- ¹⁴Clarke, D. K., Salas, M. D., and Hassan, H. A., "Euler Calculations for Multielement Airfoils Using Cartesian Grids," *AIAA Journal*, Vol. 24, No. 3, 1986, pp. 353–358.
- ¹⁵Berger, M. J., and LeVeque, R. J., "An Adaptive Cartesian Mesh Algorithm for the Euler Equations in Arbitrary Geometries," AIAA Paper 89-1930, 1989.
- ¹⁶Berger, M. J., and LeVeque, R. J., "A Rotated Difference Scheme for Cartesian Grids in Complex Geometries," *Computing Systems in Engineering*, Vol. 1, 1990, pp. 305–311.
- ¹⁷DeZeeuw, D., and Powell, K. G., "An Adaptively Refined Cartesian Mesh Solver for the Euler Equations," *Journal of Computational Physics*, Vol. 104, 1993, pp. 56–58.
- ¹⁸DeZeeuw, D., and Powell, K. G., "Euler Calculations of Axisymmetric Under-Expanded Jets by an Adaptive-Refinement Method," AIAA Paper 92-0321, Jan. 1992.
- ¹⁹Gooch, C., and Oksuzoglu, H., "Extension of State-Vector Splitting to the Navier–Stokes Equations," AIAA Paper 93-3374, July 1993.
- ²⁰Melton, J., Enomoto, F., and Berger, M., "Three-Dimensional Automatic Cartesian Grid Generation for Euler Flows," AIAA Paper 93-3386, July 1993.
- ²¹Young, D. P., Melvin, R. G., Bieterman, M. B., and Bussoletti, J. E., "A Locally Refined Rectangular Grid Finite Element Method: Application to Computational Fluid Dynamics and Computational Physics," *Journal of Computational Physics*, Vol. 92, 1991, pp. 1–66.
- ²²Melton, J. E., Berger, M. J., Aftosis, M. A., and Wong, M. D., "3D Applications of a Cartesian Grid Euler Method," AIAA Paper 95-0853, Jan. 1995.
- ²³Tidd, D. M., Strash, D. J., Epstein, B., Luntz, A., Nachshon, A., and Rubin, T., "Multigrid Euler Calculations over Complete Aircraft," *Journal of Aircraft*, Vol. 29, No. 6, 1992, pp. 1080–1085.
- ²⁴Welterlen, T. J., and Karman, S. L., "Rapid Assessment of F-16 Store Trajectories Using Unstructured CFD," AIAA Paper 95-0354, Jan. 1995.
- ²⁵Melton, J. E., Pandya, S. A., and Steger, J. L., "3D Euler Flow Solutions Using Unstructured Cartesian and Prismatic Grids," AIAA Paper 93-0331, Jan. 1993.
- ²⁶Karman, S. L., "SPLITFLOW: A 3D Unstructured Cartesian/Prismatic Grid CFD Code for Complex Geometries," AIAA Paper 95-0343, Jan. 1995.
- ²⁷Frymier, P. D., Hassan, H. A., and Salas, M. D., "Navier–Stokes Calculations Using Cartesian Grids: I. Laminar Flows," *AIAA Journal*, Vol. 26, No. 10, 1988, pp. 1181–1188.
- ²⁸Sutherland, I. E., and Hodgman, G. W., "Reentrant Polygon Clipping," *Communications of the ACM, Graphics and Image Processing*, Vol. 17, No. 1, 1974, pp. 32–42.
- ²⁹Coirier, W. J., "An Adaptively-Refined, Cartesian, Cell-Based Scheme for the Euler and Navier–Stokes Equations," Ph.D. Thesis, Dept. of Aerospace Engineering, Univ. of Michigan, Ann Arbor, MI; also NASA TM 106754, Jan. 1994.
- ³⁰DeZeeuw, D. L., "A Quadtree-Based Adaptively-Refined Cartesian-Grid Algorithm for Solution of the Euler Equations," Ph.D. Thesis, Dept. of Aerospace Engineering, Univ. of Michigan, Ann Arbor, MI, 1993.
- ³¹DeZeeuw, D., and Powell, K. G., "An Adaptively-Refined Cartesian Mesh Solver for the Euler Equations," AIAA Paper 91-1542, June 1991.
- ³²Warren, G., Anderson, W. K., Thomas, J., and Krist, S., "Grid Convergence for Adaptive Methods," AIAA Paper 91-1592, June 1991.

³³Barth, T. J., and Frederickson, P. O., "Higher-Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction," AIAA Paper 90-0013, Jan. 1990.

³⁴Liou, M. S., and Steffen, C. J., Jr., "A New Flux Splitting Scheme," *Journal of Computational Physics*, Vol. 107, 1993, pp. 23-39.

³⁵Mavriplis, D. J., and Jameson, A., "Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes," *AIAA Journal*, Vol. 28, No. 8, 1990, pp. 1415-1425.

³⁶Coirier, W. J., and Powell, K. G., "An Accuracy Assessment of Cartesian-Mesh Approaches for the Euler Equations," *Journal of Computational Physics*, Vol. 117, March 1995, pp. 121-131.

³⁷Suddhoo, A., and Hall, I. M., "Test Cases for the Plane Potential Flow Past Multi-Element Aerofoils," *Aeronautical Journal*, Dec. 1985, pp. 403-414.

³⁸Mitchell, C. R., and Walters, R. W., "K-Exact Reconstruction for

the Navier-Stokes Equations on Arbitrary Grids," AIAA Paper 93-0536, July 1993.

³⁹Holmes, D. G., and Connell, S. D., "Solution of the 2D Navier-Stokes Equations on Unstructured Adaptive Grids," AIAA Paper 89-1932, 1989.

⁴⁰Ghia, U., Ghia, K. N., and Shin, C. T., "High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method," *Journal of Computational Physics*, Vol. 48, 1982, pp. 387-411.

⁴¹Armaly, B. F., Durst, F., Pereira, J. C. F., and Schonung, B., "Experimental and Theoretical Investigation of Backward-Facing Step Flow," *Journal of Fluid Mechanics*, Vol. 127, 1983, pp. 473-496.

⁴²Russell, L. M., Thurman, D. R., Simonyi, P. S., Hippensteele, S. A., and Poinsatte, P. E., "Measurements and Computational Analysis of Heat Transfer and Flow in a Simulated Turbine Blade Internal Cooling Passage," AIAA Paper 93-1797, June 1993.

