AIAA JOURNAL
Vol. 41, No. 4, April 2003

Accurate Multiblock Navier–Stokes Solver for Complex Aerodynamic Configurations

B. Epstein*
Academic College of Tel-Aviv-Yaffo, 64044 Tel Aviv, Israel
and
T. Rubin† and S. Séror‡
Israel Aircraft Industries, Ltd., 70100 Lod, Israel

A computational-fluid-dynamics tool capable to simulate accurately viscous flows around complex aerodynamic configurations is described. The method combines a multiblock technique with a low-dissipation numerical method incorporated into a multigrid framework. Structured subdomains (blocks) are united in multigrid/multiblock structures, and the blocks are treated independently at each stage of the numerical procedure, maintaining a regular information exchange between the neighboring blocks. In the numerical procedure the convection part of the equations is approximated by a low-order upwind-biased scheme employed for multigrid relaxation in combination with a higher-order essentially nonoscillatory scheme used to supply a defect to the right-hand side of the discrete equations on the finest multigrid level in a way ensuring the overall high accuracy of the scheme. Computational examples demonstrate the ability of the resulting method to perform accurate large-scale computations of complex three-dimensional turbulent flows around realistic aerodynamic configurations.

I. Introduction

THIS work is motivated by the need of an industrial computational-fluid-dynamics tool capable of providing accurate solutions of the Navier–Stokes equations for complex high-Reynolds turbulent flows around complex industrial configurations, including complete aircraft.

An engineering environment requires geometrical flexibility (in order to treat topologically complex geometrical configurations) and high accuracy (in order to get, for example, realistic estimates of sensitive aerodynamic coefficients such as drag and moments). Additional requirements include the demand for reasonable turnaround times and high robustness.

The preceding requirements are somewhat contradictory. Robust numerical schemes can possess relatively low accuracy as their robustness is usually achieved by inclusion of artificial dissipation into the scheme, either explicitly (as in central space discretization) or implicitly [as in total-variation-diminishing (TVD) schemes]. Higher-order-accurate numerical schemes are usually far less robust, and their incorporation into multigrid/multiblock framework is usually inefficient. As a result, the practical use of high-accuracy schemes is restricted (to the best of our knowledge) to relatively simple, from the geometrical viewpoint, configurations.

In this work an attempt is made to challenge the goal of combining together a multiblock technique (geometrically flexible and well suited for parallel computations) with a low-dissipation numerical method incorporated into multigrid framework, thus ensuring robustness and efficiency of the resulting method. This work is based on the previous publications[1–3] related to a single-block Navier–Stokes solver, NES.

Initially the work was motivated by insufficient accuracy of most existing Navier–Stokes codes. Striking examples for this inaccuracy in reference to drag estimates are given, for example, in Ref. 4 for the transonic flow over the well-known ONERA M-6 wing. Apparently the reason for this inaccuracy lies in the fact that in the transonic regime linearly stable high-order methods are often unstable and most standard methods add artificial dissipation in order to ensure the nonlinear stability. This leads to loss of accuracy as the details of flow near the shock (damped by artificial viscosity) might affect large-scale properties of the flow considerably. The examples given in Ref. 5 for the flow over a flat plate show that the artificial viscosity can also destroy the viscous drag calculation in the absence of shock.

Essentially nonoscillatory (ENO) schemes, introduced by Harten et al.[6] and further simplified by Shu and Osher,[7] use the idea of an adaptive stencil interpolation favoring regions where the solution is smoother, and thus achieve accuracy that is of a uniformly high order. In Ref. 1 a finite volume numerical method based on the ENO approach was introduced, and accurate results were achieved on relatively coarse grids without the need for any additional dissipation. The method was verified for a variety of flows around single-block geometries including turbulent transonic solutions with wide regions of separated flow. The method and the appropriate computer code did not employ any numerical parameters (coefficients of artificial viscosity or likewise), and thus, at a given numerical mesh, tuning of the code parameters is irrelevant.

To perform large-scale computations employing millions of grid points, it was crucial to improve the efficiency of the method while still retaining its high accuracy. As explained in detail in Ref. 3, a straightforward implementation of the multigrid strategy employing full approximation scheme (FAS)[8] in conjunction with ENO discretization was not satisfactory, as even for smooth solution fields ENO discretization does not produce stencils for which efficient smoothers can be constructed. To overcome this, the solver of Refs. 1 and 2 was changed to be based on a defect correction multigrid approach, where the target discretization is different from that used in the relaxation process of the multigrid cycle (for previous applications of this method, for example, see Refs. 9 and 10). A first-order-accurate driver is employed for relaxation, and a high-order ENO operator supplies a defect correction to the right-hand side of the discrete equations.

The resulting single-block multigrid method[3] retained the high accuracy of the ENO method of Refs. 1 and 2 with a comparatively small number of multigrid cycles needed to reduce the error below the level of truncation errors, thus allowing accurate estimates of sensitive flow characteristics on practical mesh densities.

Received 28 March 2002; revision received 11 November 2002; accepted for publication 11 November 2002.

*Professor and CFD Consultant, Computer Science Department, 4 Antokolsky Street; epstein@mta.ac.il.

†Head of CFD Group, Ben-Gurion International Airport; trubin@comgate.iai.co.il.

‡Aerospace Research Engineer, Ben-Gurion International Airport; sseror@comgate.iai.co.il. Member AIAA.

The present work focuses on the transformation of the single-block method of Refs. 1–3 to a structured multiblock code[11] capable of treating complex aerodynamic configurations efficiently, while retaining the high accuracy of its single-block predecessor.

The multiblock method and the corresponding computer code NES retain the numerical scheme of the single-block method based on the ENO approach and its multigrid/defect correction structure without introducing any artificial numerical parameters. The data exchange among the neighboring blocks allows the treatment of convection terms of the Navier–Stokes equations in a "transparent" way. This means that a division of a single block into several subblocks does not change the approximation of convection terms even in the vicinity of an interface. For viscous terms minor changes to the numerical approximation were allowed in order to simplify the interface treatment in the case of a complex domain decomposition. This leads to only insignificant loss of accuracy in the neighborhood of block interfaces and has almost no effect on the overall aerodynamic results.

The present multiblock version uses point-to-point matching across the interface of neighboring blocks, but a block face might adjoin several neighbors in order to simplify the global domain decomposition and to reduce the total number of blocks.

The communication overhead caused by the data exchange among neighboring blocks is negligible as it uses no complicated data management such as connectivity lists. As a result, the code is highly suitable for efficient parallelization on an almost "plug-in" basis.

The results include the following benchmarks: 1) consistency and convergence study, which includes turbulent viscous flow over a transonic wing case and inviscid flow over ONERA M-6 wing; 2) grid-independence study, which is the flow about a transport-type body; 3) a popular ARA M100 wing-body turbulent transonic test case with both surface data and aerodynamic forces data comparisons included over a wide range of conditions; and 4) a transport-type fuselage drag rise vs Mach study, which was employed as a baseline calculation for a practical aerodynamic design case. The preceding tests are compared with wind-tunnel experiment and available results by other authors.

II. Mathematical Background

For the sake of completeness, a brief description of the numerical algorithm is given next. More details can be found in Refs. 1–3.

A. Equations and Choice of Approximation Scheme

The Navier–Stokes equations in Cartesian coordinates can be written in the form

$$\boldsymbol{q}_t + \operatorname{div} \boldsymbol{C} = \operatorname{div}(\boldsymbol{V}) \tag{1}$$

where the tensor $\boldsymbol{C} = (\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{h})$ represents the convection terms; the tensor $\boldsymbol{V} = (\boldsymbol{r}, \boldsymbol{s}, \boldsymbol{t})$ represents the viscous terms; the vector $\boldsymbol{q} = (\rho, \rho\boldsymbol{u}, \rho v, \rho \boldsymbol{w}, E)$ depends on the density ρ, the velocity vector $(\boldsymbol{u}, v, \boldsymbol{w})$, and the energy E; t is the time; $\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{h}$ are the inviscid (convection) fluxes; and $\boldsymbol{r}, \boldsymbol{s}, \boldsymbol{t}$ are the viscous fluxes that depend in a nonlinear mode on $\boldsymbol{q}$.

The choice of discretization for the convective part of the spatial operator is driven by the following requirements: 1) applicability to reasonably smooth three-dimensional grids not necessarily defined by mapping function(s), but rather by a set of vertices; 2) applicability to high-aspect-ratio grids typical of the Navier–Stokes computations; 3) ability to "coexist" with viscous terms without damping them as a result of artificial viscosity effect; 4) high accuracy on aerodynamic level including computation of sensitive flow characteristics such as drag; 5) use of minimal number of numerical parameters; 6) robustness; and 7) relatively low amount of computational work at given accuracy.

Theoretical considerations together with extensive numerical experiments showed that the ENO-based scheme first introduced in Refs. 1 and 2 and incorporated in a multigrid framework by means of the defect correction approach in Ref. 3 possesses the preceding properties.

B. Spatial Approximation of Convection Terms

We assume that all of the block meshes are structured, consisting of cells that form an (i, j, k) structure for each block.

For a particular cell (i, j, k) the following approximation is assumed:

$$\begin{aligned}(\Omega_{i,j,k}\boldsymbol{q}_{i,j,k})_t + [\boldsymbol{C}\cdot(S\boldsymbol{n})]_{i+0.5,j,k} - [\boldsymbol{C}\cdot(S\boldsymbol{n})]_{i-0.5,j,k} \\ + [\boldsymbol{C}\cdot(S\boldsymbol{n})]_{i,j+0.5,k} - [\boldsymbol{C}\cdot(S\boldsymbol{n})]_{i,j-0.5,k} + [\boldsymbol{C}\cdot(S\boldsymbol{n})]_{i,j,k+0.5} \\ - [\boldsymbol{C}\cdot(S\boldsymbol{n})]_{i,j,k-0.5} = [\boldsymbol{V}\cdot(S\boldsymbol{n})]_{i+0.5,j,k} - [\boldsymbol{V}\cdot(S\boldsymbol{n})]_{i-0.5,j,k} \\ + [\boldsymbol{V}\cdot(S\boldsymbol{n})]_{i,j+0.5,k} - [\boldsymbol{V}\cdot(S\boldsymbol{n})]_{i,j-0.5,k} + [\boldsymbol{V}\cdot(S\boldsymbol{n})]_{i,j,k+0.5} \\ - [\boldsymbol{V}\cdot(S\boldsymbol{n})]_{i,j,k-0.5}\end{aligned} \tag{2}$$

where $\Omega_{i,j,k}$ is the cell volume, $\boldsymbol{q}_{i,j,k}$ is some mean value of $\boldsymbol{q}$ over the cell, and S is the area of a cell side surface. Half-indices indicate from which side of the cell the flux (in square brackets) is taken. Fluxes with half-indices are approximated by a one-dimensional interpolation from the values of fluxes at nearby cell centers. That is, fluxes with subscripts $i+\frac{1}{2}, j, k$ and $i-\frac{1}{2}, j, k$ are approximated by fluxes at the cell centers with the same j and k and so on. The approximation and consistency properties are ensured, for example, by the following choice of the interpolation operator I:

$$[\boldsymbol{C}\cdot(S\boldsymbol{n})]_{i+0.5,j,k} = I\{\boldsymbol{C}_{l,j,k}\cdot(S\boldsymbol{n})_{i+0.5,j,k}\}$$
$$l = i, i+1, i-1, \ldots \tag{3}$$

with the direction vector frozen in the interpolant. The specific choice of the (nonlinear) operator I (described in the next section) determines the computation of convective fluxes.

C. Upwind-Biased and ENO Implementation

The ENO approximation[3,7] is implemented through the choice of an interpolation template in the right-hand side of Eq. (3), separately for each characteristic flux. To perform the characteristic decomposition at half-indexed points of the grid, an approximate Jacobian is calculated based on values of flow variables averaged over the neighboring cell centers in the appropriate direction. Then fluxes in the right-hand side (RHS) of Eq. (3) are projected into local characteristic fields by means of the preceding Jacobian.

In each characteristic field a template is chosen according to local characteristics and smoothness of the fluxes, which might change with iterations. An ENO interpolation template (which typically consists here of three points) is determined, primarily according to the sign of the corresponding eigenvalue and then according to the smoothness of the projected characteristic fluxes.[3,7] The interpolated characteristic fluxes are projected back to get the Cartesian ones.

In the present method the three-point ENO procedure is applied only for the defect correction calculation a very limited number of times (roughly equal to a number of multigrid cycles), and most of the computational work is performed using a one-point upwind-biased relaxation. This formally corresponds to the case of a one-point ENO template, and the interpolation proceeds as previously with the only point of a template chosen in each field according to sign of the relevant eigenvalue. In both cases if an eigenvalue changes sign between the cell centers, the corresponding flux is split into two partial fluxes as suggested in Ref. 7 (so-called entropy fix), and the interpolation is applied separately to each of the two fluxes.

D. Approximation of Viscous Terms

Natural finite differences are used to approximate the first derivatives along the local grid coordinates. The Cartesian derivatives are obtained by combining the preceding derivatives in a way that ensures exactness for a linearly changing flowfield.

E. Time Marching

A three-stage Runge–Kutta scheme is applied in a TVD-preserving form[7] with theoretical Courant–Friedrichs–Lewy (CFL) = 1. To accelerate the convergence to steady state, explicit residual smoothing is applied. It allows CFL numbers of about 1.5 (instead of 0.75, which is used in practice without the residual smoothing) for subsonic and transonic flows.

F. Sensitive Faces, Cells, and Use of a Linearly Stable Template

For subsonic and transonic flows a linearly stable template is applied in practice everywhere, except on "sensitive" faces where the variable ENO template is used. The cell face is called sonic if one of the sonic eigenvalues changes sign across the face. The face is called sensitive if it belongs to the one-dimensional neighborhood of a sonic cell face, a neighborhood being taken as the number of points in the ENO template minus one.

G. Multigrid Defect Correction Approach

For a given problem

$$L\boldsymbol{u} = \boldsymbol{F} \tag{4}$$

The defect correction approach is defined by an iterative process:

$$L_1\boldsymbol{u}^{(n+1)} = \boldsymbol{F} - \left[L_2\boldsymbol{u}^{(n)} - L_1\boldsymbol{u}^{(n)}\right] \tag{5}$$

Here, $\boldsymbol{u}$ and $\boldsymbol{F}$ are the solution and the given forcing fields, respectively; n is the iteration number; L is a (possibly nonlinear) operator; L_2 is typically a higher-order operator approximation (three-point ENO in our application) to L; and L_1 is an approximation to L that can be inverted efficiently by multigrid techniques. Thus, a defect, representing the difference between the values of operators L_1 and L_2 applied to the current solution field, is added as a forcing term to the RHS of the equations, and the actual relaxation process is performed using a stable and easy-to-invert operator L_1. In this work operators L_1 and L_2 only approximate the convection part of the equations differently, and the viscous terms approximated as just described are added to the left-hand side of Eq. (5). L_1 is chosen as a low-order upwind-biased operator, with characteristic fluxes interpolated by only a one-point stencil determined (in each characteristic field) according to the sign of the corresponding eigenvalue. L_2 is the ENO operator just described. The defect correction is applied only on the finest multigrid level; on other levels the relaxation process is simply driven by the low-order operator L_1.

In general, the convergence of the defect–correction iterations depends on the type of an approximation operator L_2 and also on the compatibility of L_1 and L_2. A simple convergence analysis for the defect–correction process, performed in a previous work,[3] showed that central difference discretizations can converge slowly because of weak damping of oscillatory errors. On the other hand, three-point ENO discretization necessarily produces asymmetric templates (e.g., outside sensitive regions, where sonic eigenvalues change sign, the flux is interpolated from one downstream and two upstream points). A model one-dimensional analysis in Ref. 3 showed that in these regions the residual convergence factor is equal to $\frac{2}{3}$, which can be further improved by overcorrection (that is, by multiplying the defect correction by a factor slightly greater than one). The same analysis showed that in sensitive regions a damping of the defect–correction iterations yields the residual convergence factor of $\frac{7}{9}$ in the worst-case analysis. It was found that satisfactory convergence behavior was obtained for three-dimensional flows computed by the multiblock code, similar to its single-block predecessor. The convergence issues are further discussed and illustrated in Sec. IV.B.

III. Multiblock Approach

In a multiblock approach the global domain is divided into smaller subdomains (blocks) for which computational meshes are easier to generate. It is presumed that an iteration process is applied separately to each block, with data exchange among the blocks responsible for the validity of boundary conditions on the block interfaces and their immediate vicinity. A number of numerical problems arise associated with the correct implementation of the preceding boundary conditions, conservativity of numerical fluxes in the vicinity of block interfaces, and multigrid interpolation. It is also vital to diminish the overhead as a result of the data exchange among neighboring blocks in order to achieve computational efficiency on serial computers on one hand and to make the code suitable for parallelization on the other.

A. Basic Features

The present multiblock method possesses the following basic features:

1) Similar to the single-block method of Refs. 1–3, the multiblock algorithm is based on an ENO numerical scheme, which is incorporated into multigrid defect–correction framework. The stability of the method is maintained by a nonlinear approximation scheme without introduction of any artificial numerical parameters.

2) The method employs no connectivity lists or similar data structures that can slow down the data transfer. This simplifies data organization and data management and thus reduces the communication overhead.

3) An appropriate multiblock data interchange ensures that the convection fluxes are approximated in the way identical to that of the single-block method. Slight changes are introduced to the approximation of viscous fluxes and to the multigrid interpolation in order to comply with the earlier requirement of a simple data management. The computational impact of these changes is negligible.

B. Multiblock/Multigrid Structure

The global domain comprising a given geometry is divided into smaller subdomains called blocks subject to the following requirements:

1) For each block i, j, k mesh structures are generated by means of standard grid generators (e.g., GRIDGEN).[12]

2) In the present version of the algorithm, there is a point-to-point matching of grid lines across the blocks interface. This restriction can be in principle lifted in order to allow more flexibility in the process of grid generation.

3) A block face might abut several neighboring blocks, each of which might also adjoin a number of different block faces (Fig. 1); this feature is further referred to as the "multiface" property.

The set of blocks thus constructed forms the finest multigrid level. Coarser multigrid levels are recursively built by comprising the blocks that are formed by merging neighboring computational volumes of the previous finer level. The numerical algorithm is embedded in the framework of the multigrid method, that is, the computational grids are arranged in hierarchical levels, starting from a coarse level, and then refining it so that each block contains the next-level refinement geometrically.

The FAS[8] drives the numerical algorithm, and the multigrid cycle is embedded in the so-called full multigrid, which supplies an initial approximation to the fine-grid solution by first solving the problem on a coarser grid and interpolating the solution to the fine grid. In the fine-to-coarse direction of a multigrid cycle, the weighted fine residuals are transferred, and the new RHS for the equations on the coarser level are computed. On the finest level the defect–correction technique is implied what is equivalent to the addition of a supplementary multigrid level on the top of the finest one. On this additional level a zero number of iterations and a higher-order ENO scheme are specified so that the multigrid residuals computed on this level represent the values of the high-order ENO operator applied to the current solution field.

C. Treatment of Block Interfaces: Data Structures

In the single-block code the concept of boundary arrays consisting of ghost cells was used in order to treat all kinds of boundary conditions (surface boundary conditions, far-field and symmetry boundary conditions). Note that the boundary arrays might be regarded as an extension, in each of six directions, of an actual

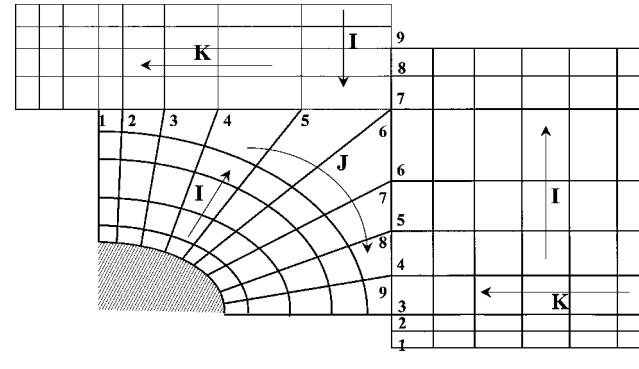


Fig. 1 Multiface block interface.

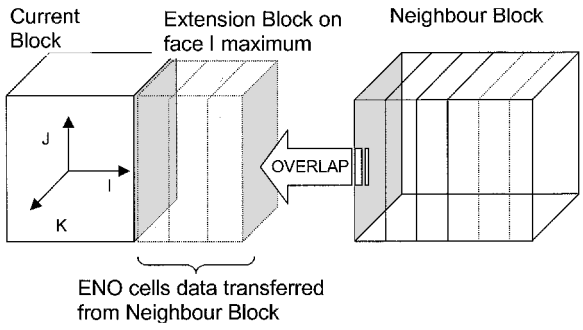


Fig. 2 Treatment of block interfaces: extension blocks.

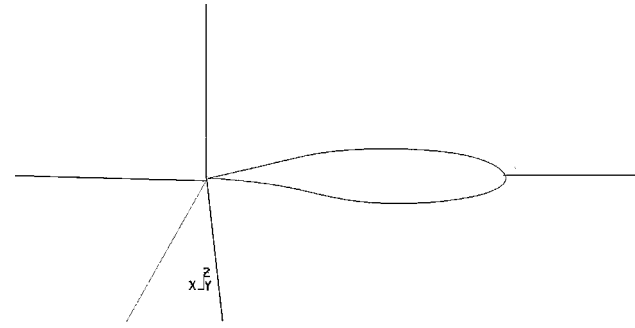


Fig. 3 Block junctions at the trailing edge.

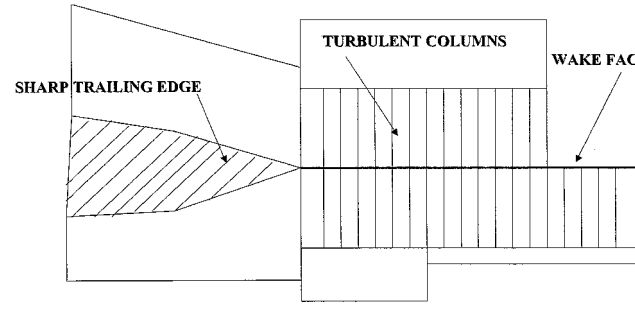


Fig. 4 Boundary faces identification and wake propagation.

grid that consists of the internal cells. The extensions do not include "corner cells" (that is, cells whose three-dimensional index includes more than one external one-dimensional subindex).

The same idea was used in order to implement data exchange between neighboring blocks through a block connectivity boundary condition, further referred to as the "merge" boundary condition. On each face corresponding to this condition, an "extension block" is built. The number of cells in such a block in the direction of extension is equal to the order of the currently applied ENO scheme. (This means that for all of the multigrid levels but the finest one the extension blocks have a width of one, as is also the case for boundary conditions other than the merge condition.) Each extension block is regularly filled by overlapping information from the neighboring block (or blocks in the multiface case; see Fig. 2).

Because of changeability of stencils, it is preferable (especially for a higher-order ENO) to transfer the values of basic variables (density, velocity, and energy), rather than the values of numerical fluxes. The preceding strategy of data transfer also enables the conservativity of the numerical scheme.

D. Spatial Approximation on Block Interfaces

As mentioned in Sec. III.B, block extensions represent "thin" blocks with a number of cells in the direction of extension equal to the current ENO order. In the remaining two directions a number of cells is equal to the number of internal cells in the corresponding directions of the basic blocks.

Everywhere, including the blocks interface, the convection terms of the Navier–Stokes equations are approximated as if the grid were a single block. Approximation of viscous fluxes and multigrid interpolations of the basic variables are performed in a way leading to only insignificant changes in the approximated values compared to the single-block case. As a result, for Euler computations the multiblock one-level results are identical to those of the original single-block code (where the comparison makes sense, for example, when dividing a single block into smaller subblocks). In the general case of multigrid viscous and inviscid computations, comparison between multiblock and single-block results shows that the impact of the changes is negligible.

A slight change in the approximation formulas was introduced in order to ensure the conservation properties of the overall numerical scheme without need for a complicated treatment of topologically complex block interfaces and thus to comply with the requirement of simple data management.

Convection fluxes are approximated one-dimensionally, so that the regular data refilling of the extension blocks ensures the full transparency of the flux interpolation. On the other hand, viscous fluxes require two-dimensional interpolation stencils. Thus, in order to retain the flux approximation formulas of the single-block scheme near a block interface it is necessary to use the velocity values at the corner cells of the block. Extension blocks defined in the preceding section do not contain corner cells as the extension is performed one-dimensionally across the face. Moreover, a correct determination of corner cells is not straightforward as they might come from a block that does not possess a face common with the block under consideration (and consequently is not normally identified as its neighbor). The situation is especially undetermined in the neighborhood of irregularities such as trailing edges of lifting surfaces, etc. (Fig. 3).

From the viewpoint of data organization, employment of corner cells requires the construction of connectivity lists. Even less desirable is the fact that it also involves the information exchange with distant blocks (i.e., blocks other than the immediate neighbors of the block).

The solution adopted was to use a lower-order approximation of the relevant viscous derivatives near the block corners, while keeping the viscous fluxes conservative.

The same line of reasoning can be applied to the fine-to-coarse multigrid interpolation of basic variables. Similarly, no corner cells are used, which slightly affects the overall accuracy of the interpolations.

In both cases only insignificant loss of accuracy has been observed, almost invisible in terms of pressure distribution. In a sensitive case of transonic flow-boundary-layer interaction, the multiblock computation retains several significant figures in lift and drag coefficients, for the same amount of multigrid iterations.

E. Treatment of Viscosity on Block Interfaces

To maintain the conservativity of viscous fluxes on block interfaces, the following algorithm is used. Turbulent viscosity coefficients are primarily determined at the cell centers in each block, and then the adjacent blocks exchange boundary viscosity values together with values of basic variables. This allows the interpolation of viscosity values on the common faces of neighboring blocks in the identical way and the ensuring of the conservativity of viscous fluxes.

F. User Interface

Manual specification of boundary conditions on the block faces can become a tedious task as the number of blocks increases. Hence the code includes a capability for automatic attribution of boundary conditions to the faces of blocks. All boundary conditions are fully determined by a single quick preprocessor run for a given configuration. This also includes the automatic identification of "wake" faces, which is essential, for example, for the standard Baldwin–Lomax[13] turbulence modeling on wakes. The preprocessor to the code automatically identifies the sharp edges that can produce a wake, labels the block faces that are geometrically induced by the preceding edges, and then recursively propagates the wake boundary condition to the neighboring blocks (Fig. 4). The approach removes

all topology dependent operations from the flow solver and thus enhances the versatility of the code.

IV. Computational Results

The first objective of the current study was to verify that the changes introduced to the numerical scheme, in order to adjust it to the multiblock structure, cause no significant loss of accuracy (Sec. IV.A). Robustness, accuracy, and convergence of the multiblock method are further discussed in Sec. IV.B, and grid-independence issues are addressed in Sec. IV.C.

The major objective of the computational tests was to check the ability of the modified method to allow large-scale calculations for realistic aerodynamic configurations while retaining its robustness and high accuracy (Secs. IV.D and IV.E).

In the computations the effects of turbulence are modeled through an eddy–viscosity hypothesis with the Baldwin–Lomax turbulence model, which, according to prior experience, might produce reasonably accurate representation of the flow in the conditions where the separation effects are not of primary importance. The results, which include surface pressure, lift/drag, and drag rise vs Mach data, were compared with a wind-tunnel experiment and available results by other authors.

The code uses no parameters associated with artificial viscosity treatment, explicitly or implicitly, and so, at a given computational grid, the only computational parameter subject to variation, is the CFL number. In all of the runs, the value of the CFL number was kept equal to 1.5. Thus all of the runs presented next are "first-shot" runs.

A. Consistency Check: Single-Block vs Multiblock Results

In this section the performance of the new multiblock solver is compared vs the single-block solver, by applying it to a cranked, swept wing with twisted and cusped profiles. The geometry of the wing and comparisons of the single-block numerical results with experiment are given in Refs. 2 and 3.

A C-O grid, which previously consisted of one block, was divided into smaller subgrids treated as the blocks of a multiblock structure. Different partitions included from four to six blocks. In the case of one-level transonic Euler computations, both methods produced identical results (as indicated in the preceding section), whereas multilevel multiblock Euler computations slightly differ from the corresponding single-block results. The negligible discrepancy is easily explained by the changes in multigrid interpolation on block interfaces, and it asymptotically vanishes.

The next step was to rerun the case on a multiblock set of grids in viscous transonic conditions $M_\infty = 0.80$, $Re = 13.1 \times 10^6$, at the angle of attack, which yielded the wing lift value of about 0.35. On the whole, the multiblock computations cause no significant loss of accuracy. It was also found that the most sensitive partitions contain block interfaces at the leading- and trailing-edges regions (indicated in Fig. 5), where the decomposition includes six blocks, whereas upper and lower surface partitions carried out outside the preceding regions have little effect on pressure and velocity of density distributions. The worst found discrepancy is presented in Figs. 6a and 6b, where computed surface pressure coefficients are presented at the span stations $2y/b = 0.30$ and 0.67, respectively.

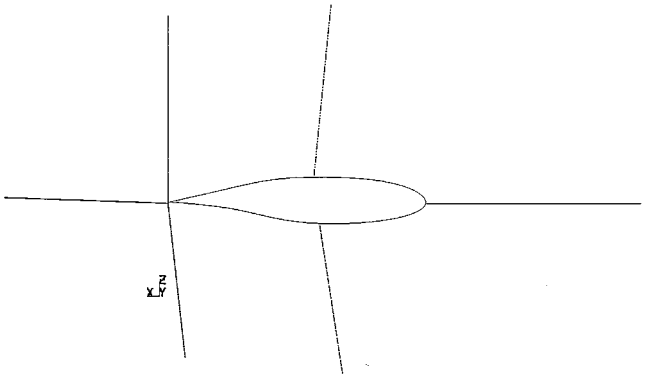


Fig. 5 Cranked, swept wing: the domain decomposition.

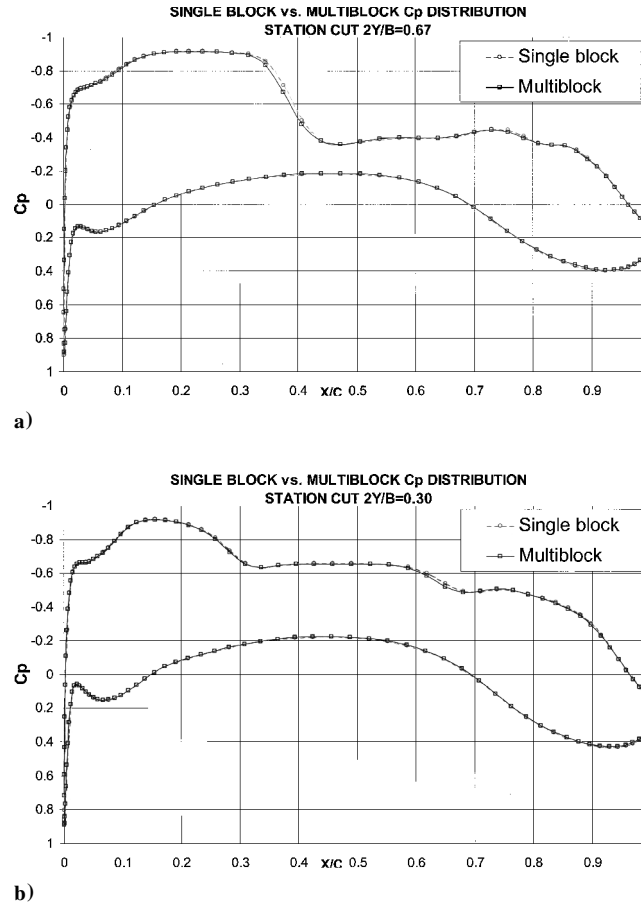


Fig. 6 Cranked, swept wing. Chordwise pressure distribution at a) $2y/b = 0.30$ and b) $2y/b = 0.67$.

The property of transparency of block interfaces with respect to the solution is further confirmed by investigating the behavior of residuals in the vicinity of interblock boundaries. The maximum (over a block) residual values are always located in the regions of complicated flow irrespective of block boundaries location, and the overall behavior of the multiblock run residuals is very similar to that of the single-block run. This is a general property observed in all of the runs performed by the multiblock code. It is mainly explained by the transparency of the characteristic interpolation applied to convective fluxes (see Sec. III.D).

B. Robustness, Accuracy, and Convergence of the Method

ENO numerical schemes maintain stability by varying the computational template. As a consequence, at a given grid resolution maximum residuals do not diminish to machine zero. Similarly to the single-block code,[2] the convergence is estimated by the reduction of residuals by several orders of magnitude and by the steadiness of the solution in terms of relevant flow characteristics.

As already stated in Ref. 2, the level of variations in the maximum residuals norm reflects the ability of the grid to resolve the flow properly. In all of the computational runs presented in the paper, the L_1 norm of the residual corresponding to the continuity equation was reduced by 3–4 orders of magnitude (apparently below the level of truncation errors). Similar to the convergence behavior of the single-block code, this allowed roughly 3–4 significant figures in major aerodynamic coefficients such as lift and drag.

The preceding is illustrated in Table 1, which summarizes the convergence history for the case of a transonic wing of the preceding section. The multiblock configuration consisted here of six blocks, and flight conditions were close to those of cruise: $M_\infty = 0.80$, $Re = 13.1 \times 10^6$, $C_L = 0.53$. It can be assessed from Table 1 that lift and drag coefficients stayed constant after roughly 30 multigrid cycles, whereas residuals continued to diminish in the course of another 30 multigrid cycles. Note that the convergence is

Table 1 Convergence history of aerodynamic coefficients and of the density residual; transport-type cranked wing

Multigrid cycles	C_L	C_D	Residual
0	——	——	$1.2E-5$
15	0.5312	0.02834	$1.1E-7$
30	0.5343	0.02845	$2.1E-8$
60	0.5344	0.02843	$2.1E-9$

Table 2 ONERA M-6 wing, $M_\infty = 0.60$: comparison between computed and theoretical pressure drag; inviscid computation on grids with high aspect ratio of cells

Angle of attack, deg	C_D (theory)	C_D (multiblock)	C_D (single-block)
0	0.00000	0.00004	0.00004
1	0.00045	0.00050	0.00049
2	0.00203	0.00208	0.00207

very close to that given in Ref. 3 for the same test case treated by the single-block method.

The next example is obtained by solving the Euler equations on grids with a high aspect ratio of cells typical for grids constructed for Navier–Stokes computations. The accuracy of such computations reflects the ability of the grid to resolve accurately the convection terms of the Navier–Stokes equations and the robustness of the method. The results for the symmetrical ONERA M-6 wing at $M_\infty = 0.60$, $\alpha = 0$, 1, 2 deg are compared in Table 2 to the theoretical induced drag and to the preceding computations by the single-block method.[3] The finest (at level 4) computational grid (slightly enlarged in comparison with that of Ref. 3) contained $192 \times 64 \times 32$ cells.

Similar to Sec. IV.A, the results of multigrid Euler computations by the two methods are almost identical, and both of them are very close to the theoretical values (with discrepancy not greater than 0.5 counts).

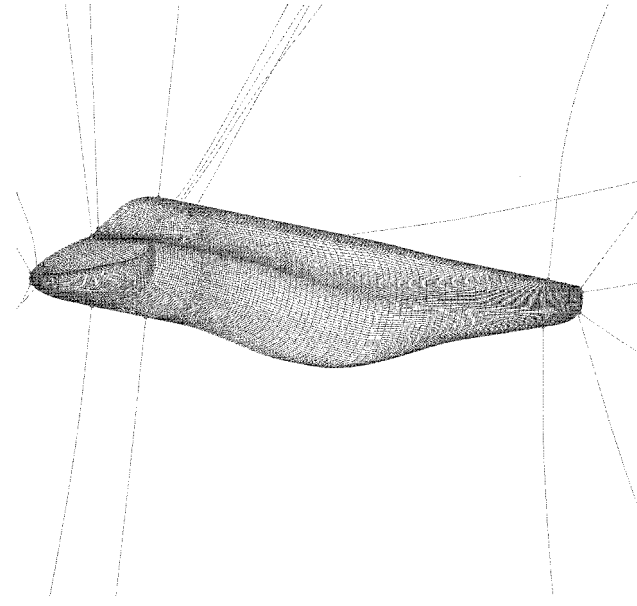
C. Grid-Independence Studies

To test grid-independence properties of the method, the following realistic case was studied. The flow around the body of a transport-type aircraft was computed by using two markedly different multilevel meshes. The first mesh was constructed for the isolated body. This allowed the construction of a smooth O-grid with low skewness (e.g., with grid lines normal to the surface in the corresponding direction). The finest (third) level contained nine blocks with a total of 1.34×10^6 cells (Fig. 7a). The second mesh of C-O type was derived from a previously constructed C-O grid for the preceding body in the presence of a wing. The body-alone mesh was derived by eliminating the wing and refilling the vacant space by additional computational blocks. In the wing-body grid it was necessary to comply with the requirement of noncrossing two-dimensional grid columns originating from the surface in the Baldwin–Lomax modeling. That resulted in highly skewed grid in the neighborhood of the wing-body junction, a property that was inherited by the final mesh (Fig. 7b, where the grid lines form a sort of "open book"). The finest (third) level consisted of 22 blocks with a total of 3.22×10^6 cells. The mesh is typical for cases when the body-wing interference is numerically studied in comparison with the flow over an isolated wing.

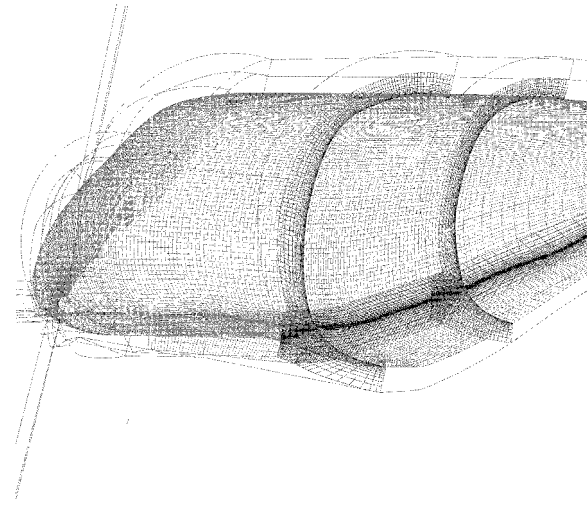
The computations were performed on two and three multigrid levels for both meshes. The lift and drag values at $M_\infty = 0.60$, $\alpha = 0$ deg are given in Table 3, and surface-pressure distributions on the center line of the body, taken from the fine level, are shown in Fig. 8a (O-grid) and Fig. 8b (C-O grid). It can be assessed from Table 3 that, on a much smoother O-grid, the two- and three-level solutions yield very close lift and drag values, whereas the two-level drag value on the C-O grid is of about 7–8 counts greater. Nevertheless, the finest (three-level) computation on the C-O grid yields the aerodynamic coefficients very close to those asymptotically achieved on O-grid, and the two pressure distributions on the finest multigrid levels are also close. The ability of the method to produce close results in terms of sensitive aerodynamic characteristics, on markedly different meshes, indicates the grid-independence property of the method.

Table 3 Grid-independence study (lift and drag coefficients of a transport-type body computed on grids of different types and resolution; $M_\infty = 0.60$, $\alpha = 0$ deg)

Type of mesh/number of multigrid levels	C_L	C_D
O-grid/2 levels	−0.00270	0.00638
O-grid/3 levels	−0.00263	0.00650
C-O grid/2 levels	−0.00360	0.00714
C-O grid/3 levels	−0.00290	0.00640



a) O-grid



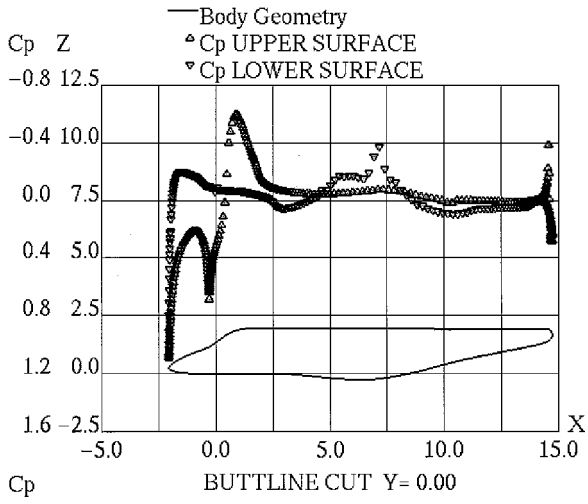
b) C-O grid

Fig. 7 Generic fuselage configuration.

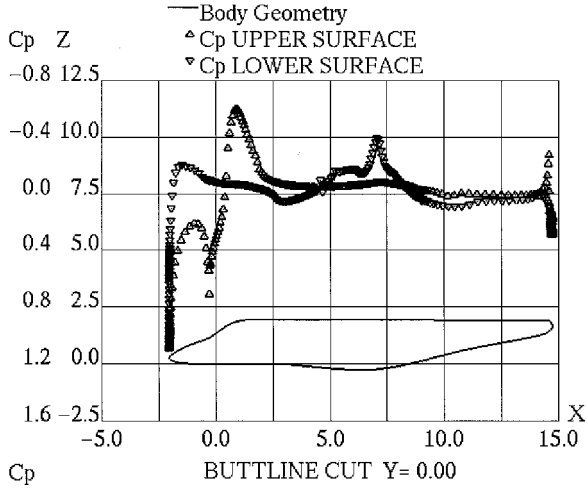
D. Test-Case ARA M100

1. Choice of a Benchmark

The test case selected is that of flow over ARA M100 wing at subsonic and transonic conditions. The wing has transonic/transport design. The test case was chosen because of the requirement for extensive and reliable experimental data on one hand and because of the



a) O-grid



b) C-O grid

Fig. 8 Pressure distribution on the centerline of a generic fuselage configuration.

availability of available computations by other authors on the other. The present case has been the subject of detailed studies in the past (e.g., see Ref. 14 and data online at http://hpccp-www.larc.nasa.gov) with both surface-pressure measurements and integral aerodynamic coefficients available from the experiment for a wide range of angles of attack.

2. *Geometry and Experiment*

The configuration matched the geometry of a wind-tunnel model tested in United Kingdom's Aircraft Research Association wind tunnel.[15] The wing is mounted approximately centrally on the fuselage axis and has a 5-deg dihedral. Detailed surface-pressure data and aerodynamic forces were measured for different Mach angle-of-attack conditions. The computational geometry used in grid generation was created by enriching the original airfoils of Ref. 15. A small cap was added to the wing tip (details of which were not available from Ref. 15).

3. *Grid Generation*

The computational grid used here was generated using GRIDGEN V8 (Ref. 12). To allow a fair comparison, an effort was made to generate a grid close to that constructed in Ref. 14 and online at http://hpccp-www.larc.nasa.gov. The grid topology is of C-O type, and it is stretched 16 mean chords upstream and 24 mean chords normal to the surface and the wing tip. The multigrid set contains three levels. The fine multiblock grid consists of 325 points around the configuration, 57 points normal to the surface, and 49 points in the spanwise direction. The computational grid comprises 24 blocks,

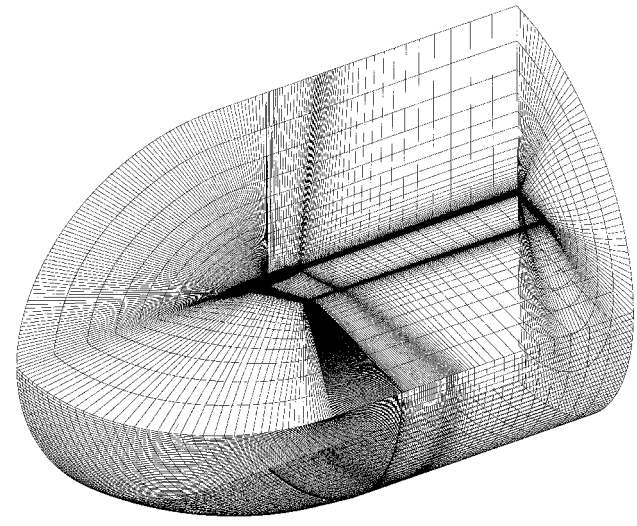


Fig. 9 ARA M100 configuration: grid topology.

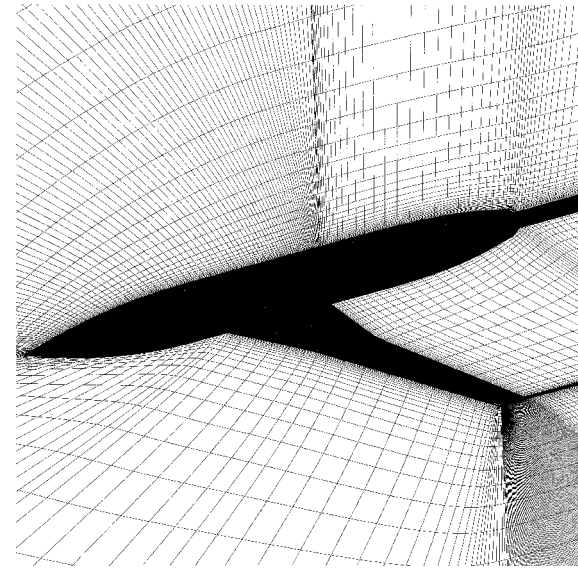


Fig. 10 ARA M100 configuration: detailed view of the computational grid.

each of them of i, j, k structure. The total number of cells is about 900,000. Two coarser multigrid levels were derived as medium and coarse by deleting every other coordinate line in each of the three directions. The first 15 points in the normal to surface direction were clustered to lie within the boundary layer in the way ensuring normal spacing of about 10^{-7} body length at the surface. As suggested in Ref. 14 and online at http://hpccp-www.larc.nasa.gov, an aeroelastic effect that caused a tip-down twist was included into the geometry and computational grids, but the wing-tip deflection mentioned in Ref. 14 was not applied. The fine-level grid dimensions are very close to those of the grid designated in Ref. 14 and online at http://hpccp-www.larc.nasa.gov as a medium grid. Grid topology is presented in Fig. 9, and a close view of the grid is shown in Fig. 10.

4. *Computational Runs*

A total of 28 computational runs was performed, aimed at achieving the following comparisons: 1) surface-pressure coefficients at the fixed freestream Mach number of $M_\infty = 0.80$ at different angles of attack; 2) lift vs angle-of-attack curve and drag polar C_L vs C_D at the same Mach number; and 3) the drag rise curve C_D vs Mach at a lift coefficient of about 0.40.

The solutions presented here possess the normal spacings at the wing surface, keeping $y+$ of about 1.5–2.5 on the finest meshes employed in order to allow reliable representation of viscous effects. In all of the computations, $Re = 13.1 \times 10^6$, and the flow was treated as fully turbulent.

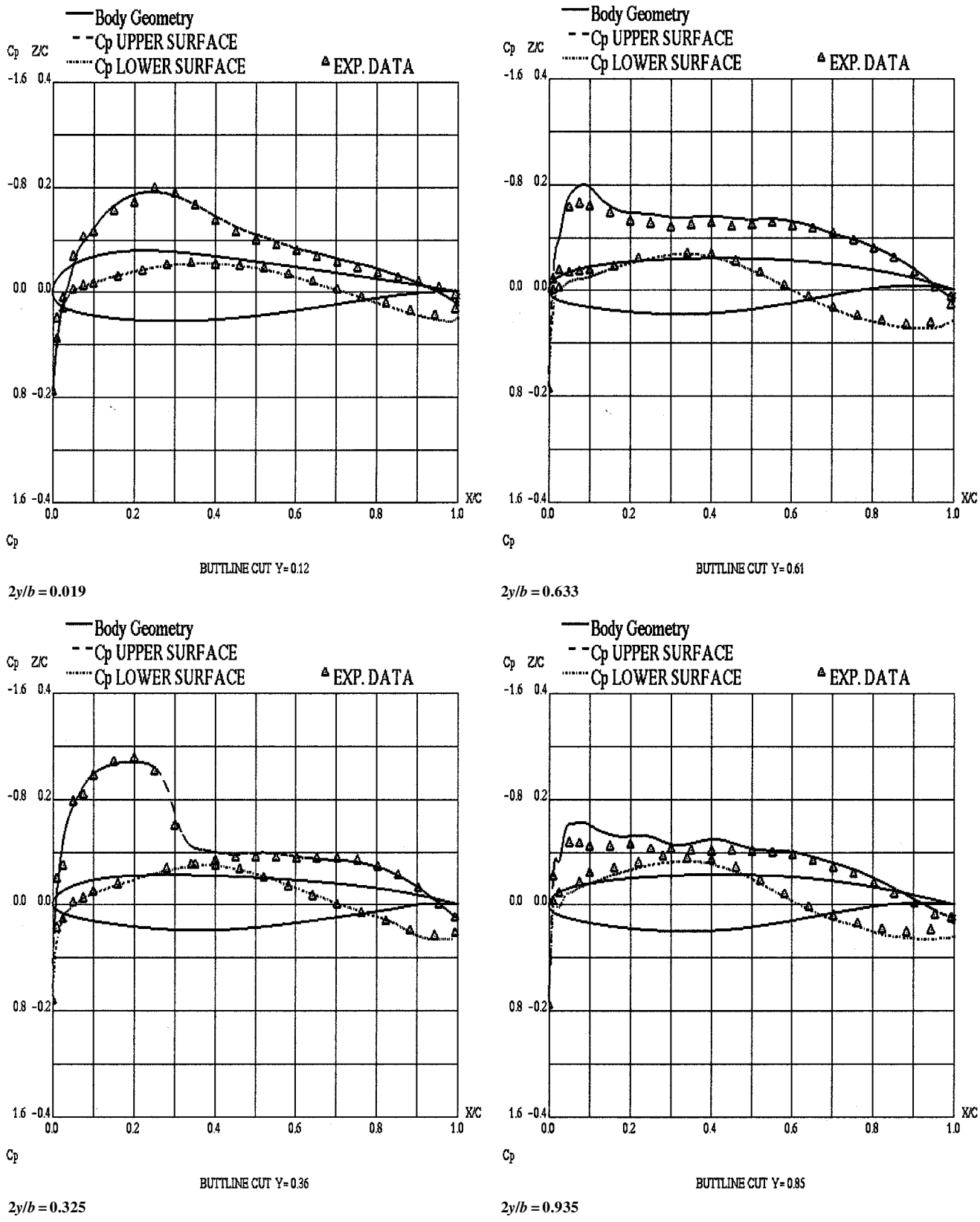


Fig. 11 ARA M100 configuration at $M = 0.80$ and $\alpha = -3.017$ deg. Chordwise pressure distribution at different span stations.

Most computational runs were performed in the framework of both two- and three-level multigrid in order to study an asymptotic behavior of the solution.

5. Surface-Pressure Comparisons

Wing surface-pressure comparisons were performed at the angle of attack, varied between −3.017 and 2.873 deg and at the freestream Mach number $M_\infty = 0.80$, $Re = 13.1 \times 10^6$, conditions that vary from low subsonic to transonic. The most challenging aspect of the computations was the shock-boundary-layer interaction at transonic flight conditions.

The wing surface-pressure comparisons are shown in Figs. 11–14 at the angles of attack $\alpha = -3.017$, −1.46, 0.148, and 2.873 deg, respectively. Each figure shows the pressure distribution at four span stations of the experiment. The surface-pressure coefficients are predicted reasonably well throughout the whole range of angles

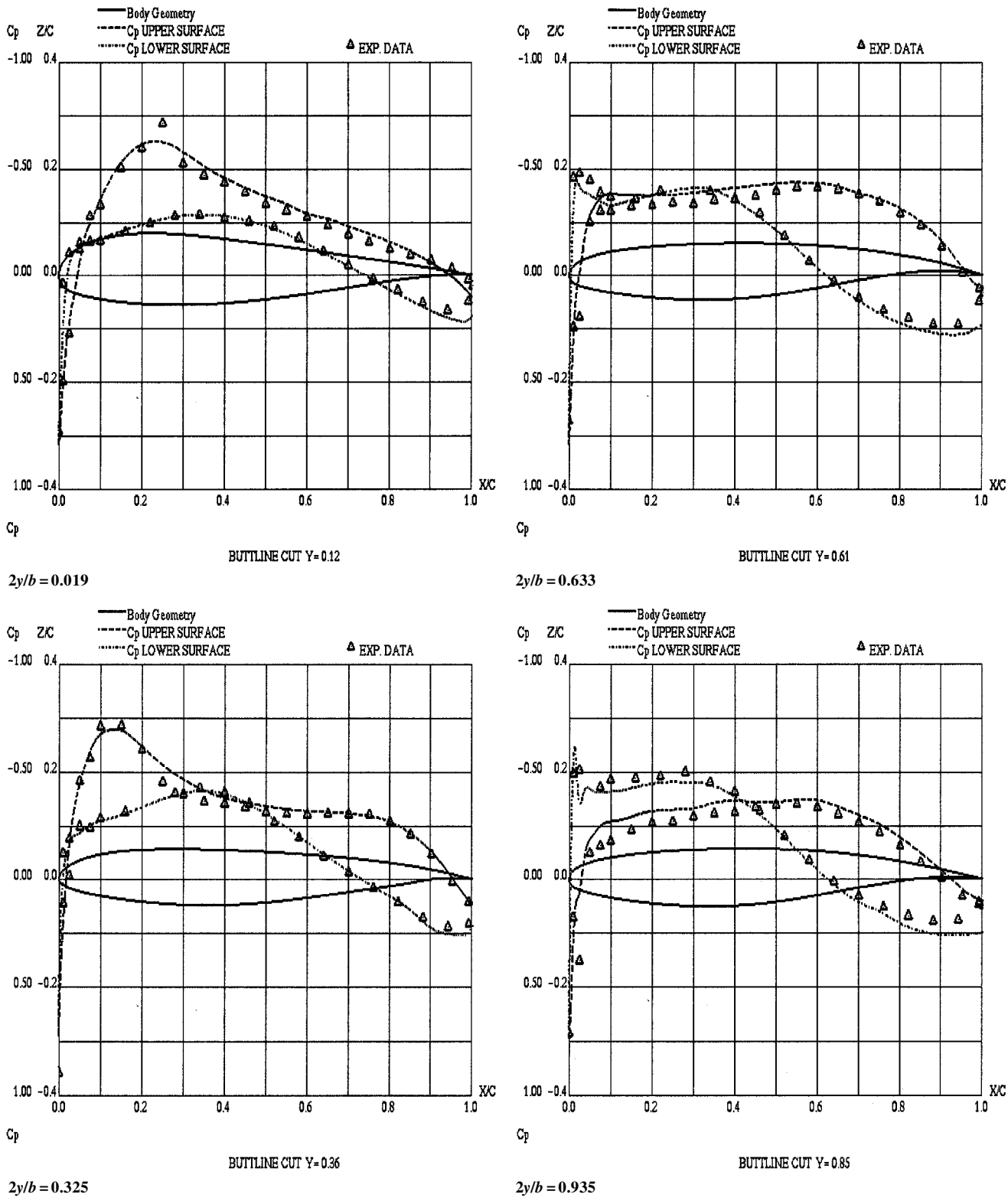


Fig. 12 ARA M100 configuration at $M = 0.80$ and $\alpha = -1.46$ deg. Chordwise pressure distribution at different span stations.

of attack. In general, the pressure distributions are close to those presented in Ref. 14 and online at http://hpccp-www.larc.nasa.gov. The computation seems to predict both the shock on the lower surface at $\alpha = -3.017$ deg and at the higher positive angles of attack. Somewhat surprisingly, the computation at $\alpha = 2.873$ deg predicts the shock location apparently better than the computation in Ref. 15, performed with the same turbulence model. This might be possibly attributed to a low-dissipation numerical scheme of the current method, which allows a reasonable prediction of shock-waves/boundary-layer interaction in the absence of strong separation by means of an algebraic turbulence modeling. Additionally a coarser, two-level multigrid computation was also carried out. In this case the finest grid (which is the medium grid of the three-level computations) contains slightly more than 115,000 computational cells. The effect of refinement is most significant at the highest computed angle of attack $\alpha = 2.873$ deg (Fig. 15), where the two pressure distributions are compared at the section $2y/b = 0.33$. Still the discrepancy is mostly confined to the shock region, where the finer computation exhibits a sharper shock pattern. On the whole, the two

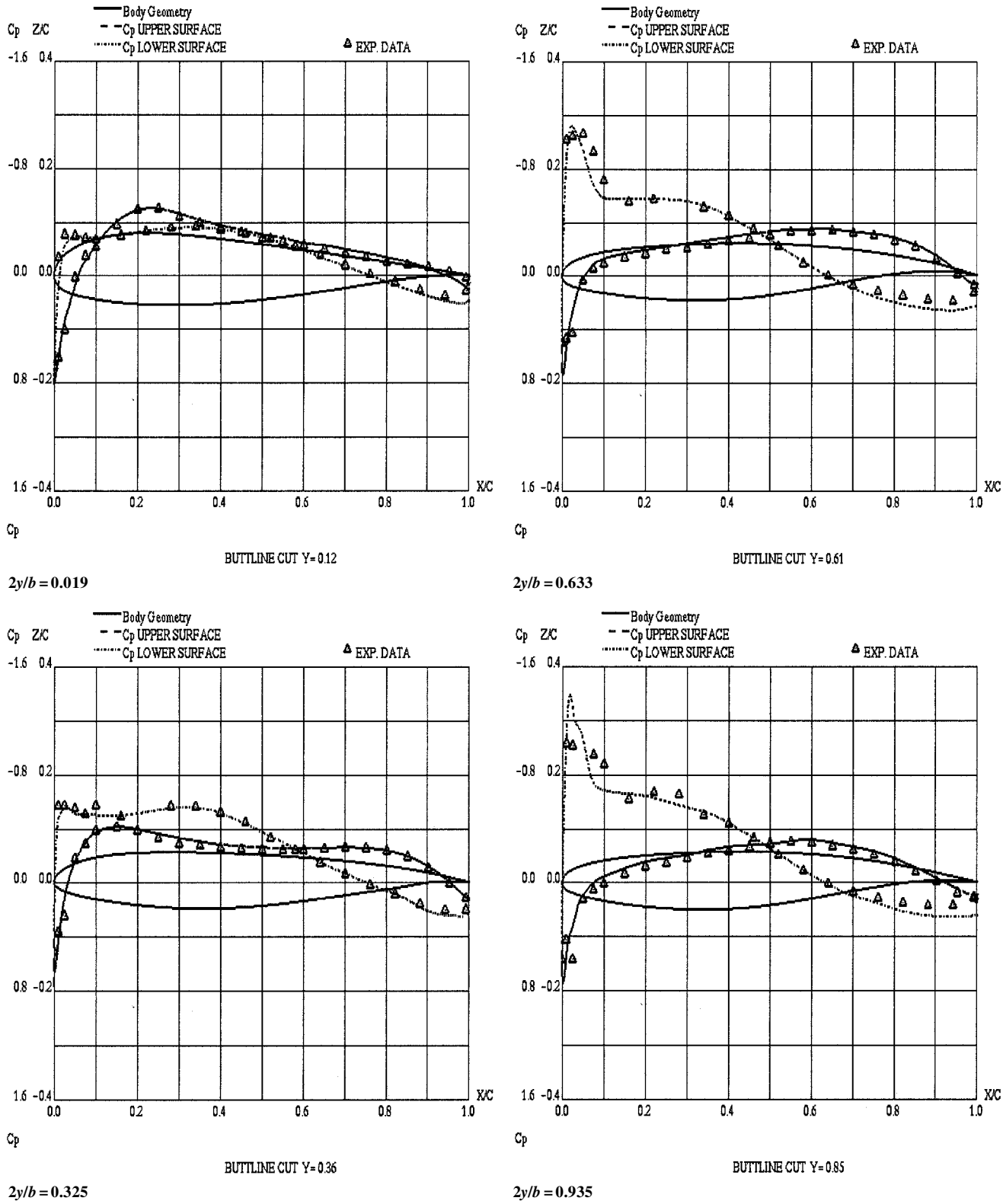


Fig. 13 ARA M100 configuration at $M = 0.80$ and $\alpha = 0.148$ deg. Chordwise pressure distribution at different span stations.

computations produce rather close results, which indicates a good grid convergence in terms of pressure distributions. The grid convergence in terms of lift and drag will be discussed in the next sections.

6. Aerodynamic Coefficients

Figure 16 shows the lift coefficient curve as function of angle of attack at $M_\infty = 0.80$. The medium- and fine-grid results by the current method are compared to experiment and to the results of Ref. 14 and data online at http://hpccp-www.larc.nasa.gov. The agreement with the lift curve slope is good, but the computation overpredicts a lift. Note that the fine-grid computation is nearer to the experiment than the medium-grid curve. Results of Ref. 14 and the data online at http://hpccp-www.larc.nasa.gov (at approximately the same grid resolution) are closer to the experimental curve than the current results, but a computation on a doubly fine grid yielded a much higher lift, which is attributed in the data online at http://hpccp-www.larc.nasa.gov to the oversensitivity of a specific turbulence modeling of the computation and to a need in additional aeroelastic corrections.

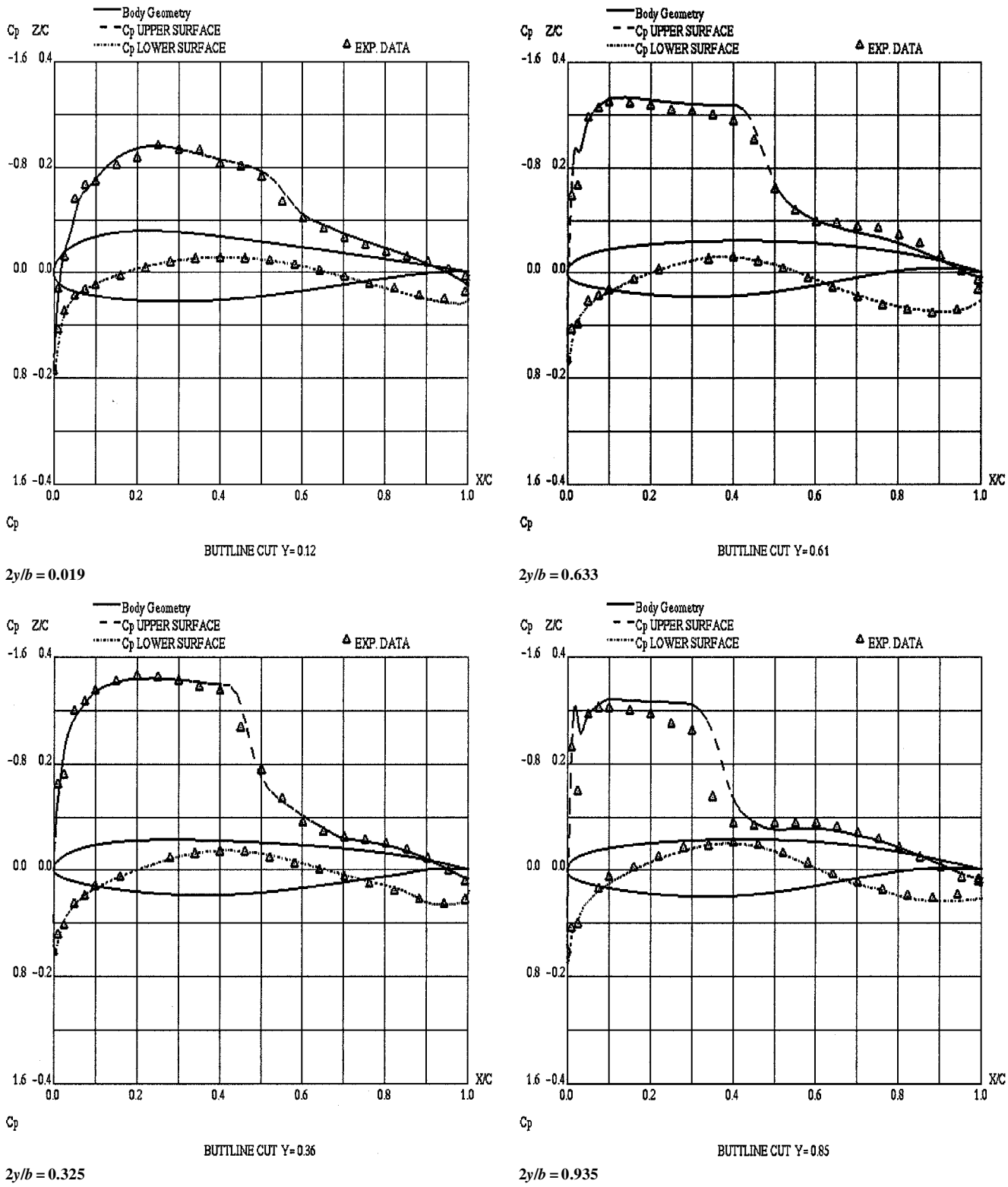


Fig. 14 ARA M100 configuration at $M = 0.80$ and $\alpha = 2.873$ deg. Chordwise pressure distribution at different span stations.

A three-level computation drag polar C_L vs C_D (Fig. 17) matches the experimental curve rather closely up to the highest computed angle of attack. The zero lift drag predicted by the computation is only slightly (about 7–8 counts or 0.0007–0.0008) lower than the experimental one. The drag values are clearly much nearer to the experiment than those reported in Ref. 14 for the same grid resolution. On a doubly fine grid of the data at http://hpccp-www.larc.nasa.gov, a much better matching is reported than in Ref. 14, especially when Menter's turbulence model is employed with the best zero lift drag estimate of about 12 counts higher than the experimental value. The zero lift drag values predicted on the medium (about 115,000 cells) and fine (about 900,000 cells) grids of the present computation differ significantly less than the estimates achieved in Ref. 14 and online at http://hpccp-www.larc.nasa.gov on the grids that comprise about 900,000 and close to 200,0000 points, respectively.

7. Drag Rise vs Mach at a Given Lift Coefficient

The drag rise curve C_D vs Mach number at a lift coefficient of about 0.40 (Fig. 18) illustrates the increase in drag caused by increasing shock strength with increasing Mach number. The three-level computation, which slightly underpredicts the experimental drag values, is also compared with the corresponding curve in

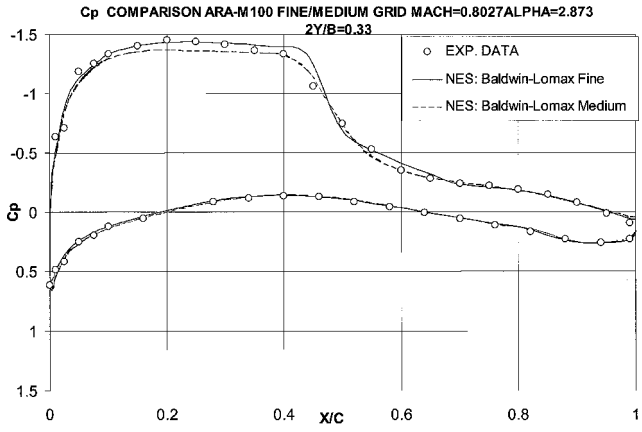


Fig. 15 ARA M100 configuration. Effect of refinement: comparison of pressure coefficients computed by two and three multigrid levels.

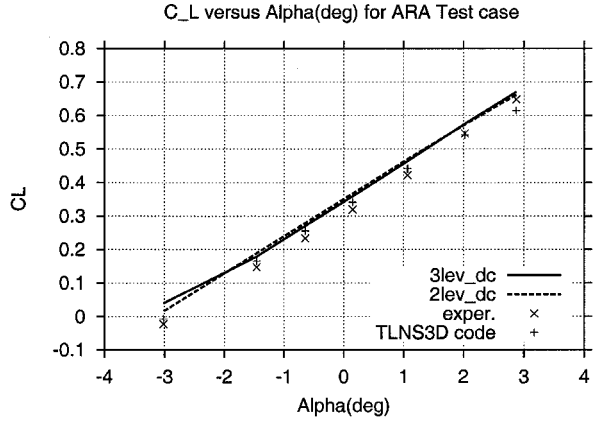


Fig. 16 ARA M100 configuration. Lift coefficient curve at $M = 0.80$.

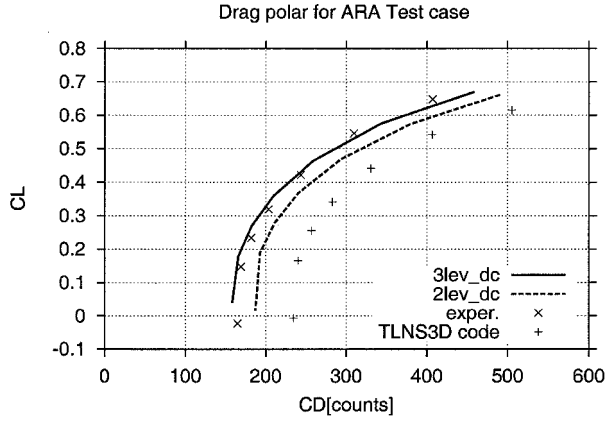


Fig. 17 ARA M100 configuration. Drag polar at $M = 0.80$.

Ref. 14 and online at http://hpccp-www.larc.nasa.gov (which employs a grid of similar resolution) and with the present two-level computation. The coarser grid computation is also rather close to the experimental curve, and the discrepancy between the two computational curves by the current method is relatively low, once more indicating a good grid convergence. To the best of our knowledge, no results by the method of Ref. 14 and the preceding Website on the grids of a finer resolution are available.

E. Drag Rise for a Transport-Type Fuselage

This case was designed to investigate the ability of the code to predict the transonic drag rise characteristics of a realistic fuselage, and the computational results served as a baseline computation for a further aerodynamic design. The body is typical of a transport-type aircraft. Reliable experimental data are available for the case

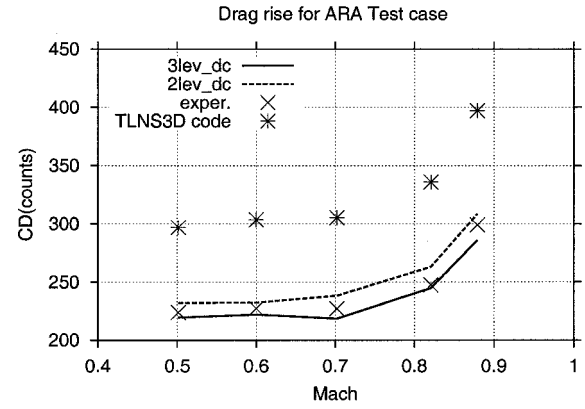


Fig. 18 ARA M100 configuration. Drag rise vs Mach number at $C_L = 0.40$.

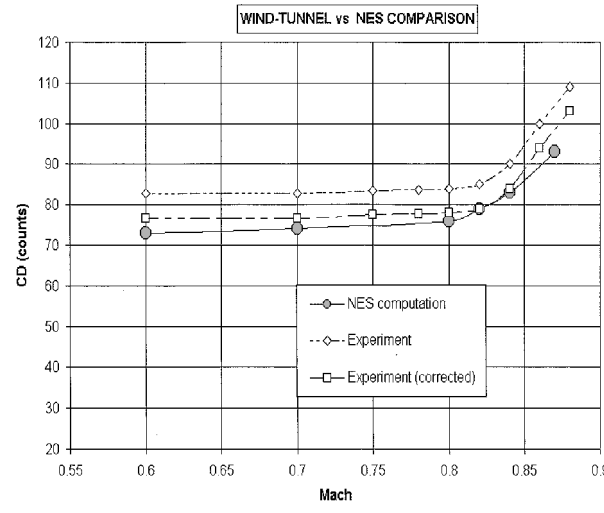


Fig. 19 Transport-type fuselage. Drag rise vs Mach number at $\alpha = 0$ deg.

including the experimental drag values correction caused by the wind-tunnel installation conditions.

Figure 19 shows a plot of drag coefficient vs Mach number at an angle of attack of 0 deg. The computational data set is compared to the raw experimental data of Ref. 16 and to a corrected experimental curve where an estimated correction of about six counts was applied. The computational and experimental drag values compare both qualitatively and quantitively, thus indicating the ability of the code to predict sensitive characteristics for complex aerodynamic shapes.

V. Conclusions

A multigrid solver has been developed and applied to a three-dimensional model of turbulent compressible flow at high Reynolds number.

The code extends the capabilities of its predecessors by the introduction of the multiblock framework, incorporated into the multigrid/defect correction method, and, consequently, by its applicability to large-scale Navier–Stokes computations.

The computational tests show the ability of the modified method to achieve accurate results on relatively coarse meshes, thus retaining the low-dissipation property of the single-block method. Sensitive aerodynamic characteristics, such as transonic drag rise or zero lift drag, were correctly predicted for reasonably complex aerodynamic configurations. Comparison of computations performed on grids with different space resolution indicates good grid convergence and an asymptotic grid independence of results.

The interblock data exchange treats convection terms of the Navier–Stokes equation in a transparent way, thus ensuring the stability of the multiblock method. No artificial parameters have been

introduced to the numerical scheme, and the changes caused by a multiblock implementation resulted in only insignificant loss of accuracy compared to the single-block method.

The exchange of information between the blocks is constructed in a modular way, which allows an almost "plug-in" implementation of the code on parallel multiprocessors.

High accuracy and robustness of the method, lack of artificial numerical parameters, and a simple user interface allow the code to be used for practical large-scale aerodynamic analysis and design in the engineering environment.

Acknowledgment

Partial results of this paper were presented at the 23rd Congress of the International Council of the Aeronautical Sciences in Toronto, Ontario, Canada, September 2002.

References

[1]Epstein, B., and Nachshon, A., "An ENO Navier–Stokes Solver Applied to 2-D Subsonic, Transonic and Hypersonic Aerodynamic Flows," AIAA Paper 94-0303, Jan. 1994.

[2]Epstein, B., Jacobs, A., and Nachshon, A., "Aerodynamically Accurate Three-Dimensional Navier–Stokes," *AIAA Journal*, Vol. 35, No. 6, 1997, pp. 1089–1091.

[3]Epstein, B., Averbuch, A., and Yavneh, I., "An Accurate ENO Driven Multi-grid Method Applied to 3D Turbulent Transonic Flows," *Journal of Computational Physics*, Vol. 168, No. 2, 2001, pp. 316–338.

[4]Wong, K. J., Ayers, T. K., and van Dam, C. P., "Accurate Drag Prediction—a Prerequisite for Drag Reduction Research," Society of Automotive Engineers, Paper 932571; also *SAE Transactions*, Sec. 1, Vol. 102, 1993, pp. 1882–1891.

[5]Reddy, S., and Papadakis, M., "Artificial Viscosity Models for the Navier–Stokes Equations and Their Effect in Drag Prediction," AIAA Paper 93-0193, Jan. 1993.

[6]Harten, A., Engquist, B., Osher, S., and Chakravarthy, S., "Uniformly High Order Accurate Non-Oscillatory Schemes, I," *Journal of Computational Physics*, Vol. 71, 1987, pp. 231–303.

[7]Shu, C.-W., and Osher, S., "Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes," *Journal of Computational Physics*, Vol. 83, No. 32, 1989, pp. 32–78.

[8]Brandt, A., "Multi-Level Adaptive Computations in Fluid Dynamics," *AIAA Journal*, Vol. 18, No. 10, 1980, pp. 100–108.

[9]Koren, B., "Multigrid and Defect Correction for the Steady Navier–Stokes Equations," *Journal of Computational Physics*, Vol. 87, No. 1, 1990, pp. 25–46.

[10]Mulder, W. A., "A High-Resolution Euler Solver Based on Multigrid, Semi-coarsening, and Defect Correction," *Journal of Computational Physics*, Vol. 100, No. 1, 1992, pp. 91–104.

[11]Séror, S., Rubin, T., and Epstein, B., "Construction of a Multiblock 3D Full Navier–Stokes Code for Practical Aerodynamic Computations," *Proceedings of 41th Israel Annual Conference on Aerospace Sciences*, Vol. 1, Technion Press, Haifa, Israel, 2001, pp. 231–241.

[12]Steinbrenner, J. T., Chawner, J. R., and Foust, C. L., "The GRIDGEN3D, Multiple Block Grid Generation System," WDC-TR-90-3022, Vol. I and II, Wright–Patterson AFB, OH, 1990.

[13]Baldwin, B. S., and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turblent Flows," AIAA TP 78-257, 1978.

[14]Marconi, F., Siclary, N., Carpenter, G., and Chow, R., "Comparison of TLNS3D Computations with Test Data for a Transport Wing/Simple Body Configuration," AIAA Paper 94-2237, June 1994.

[15]Carr, M. P., and Pallister, K. C., "Pressure Distributions Measured on Research Wing M100 Mounted on an Axysymmetric Body," AGARD, AR-138 Addendum, 1984.

[16]Ernst, Y., "IAI WW1125-E1 1:14 Model NLR Wind-Tunnel Results," Israel Aircraft Industries, Rept. 18931, Lod, Israel, 1978.

P. Givi
Associate Editor