

AN ACCURATE MULTIBLOCK ENO DRIVEN NAVIER-STOKES SOLVER FOR COMPLEX AERODYNAMIC CONFIGURATIONS

Boris Epstein¹, Theodor Rubin², Stéphane Séror²

¹ Academic College of Tel-Aviv-Yaffo, Computer Science Department
4 Antokolsky St., Tel Aviv 64044, Israel, e-mail: epstein@mta.ac.il

² Israel Aircraft Industries, Aerodynamic Dept. CFD Group
Ben-Gurion International Airport, Lod, Israel, e-mail: sseror@comgate.iai.co.il

Keywords: Navier-Stokes, CFD, Transonic Flow, Turbulent Flows, Multiblock, Multigrid, ENO, Complex Configurations

Abstract

An CFD 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 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 non-oscillatory (ENO) scheme used to supply a defect correction 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 3-D turbulent flows around realistic aerodynamic configurations.

1 Introduction

This work is motivated by the need of an industrial CFD 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, high accuracy, reasonable turnaround times and high robustness.

Robust numerical schemes may possess relatively low accuracy as their robustness is usually achieved by inclusion of artificial dissipation into the scheme, either explicitly or implicitly. 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 relatively simple - from the geometrical viewpoint - configurations.

In this work an attempt is made to challenge the goal of combining together a Multiblock technique with a low dissipation numerical method incorporated into Multigrid framework.

In [1] a finite-volume numerical method based on the ENO approach [4, 5] was introduced and accurate results were achieved on relatively coarse grids without the need for any additional dissipation. The method and the appropriate com-

puter 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.

In order to perform large-scale computations employing millions of grid points, it was crucial to improve the efficiency of the method. A straightforward implementation of the Multigrid strategy employing FAS (Full Approximation Scheme [6]) in conjunction with ENO discretization was not satisfactory [3] and the solver of [1, 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. 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 [1, 2] with a comparatively small number of multigrid cycles needed to reduce the error below the level of truncation errors.

The present work focuses on the transformation of the single-block method of [1, 2, 3] to a structured multiblock code [7] capable of treating complex aerodynamic configurations efficiently. The data exchange among the neighboring blocks allows the treatment of convection terms of the Navier-Stokes equations in a "transparent" way. For viscous terms, minor changes to the numerical approximation were allowed. 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 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 two benchmarks:

1. 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.

2. A transport-type fuselage drag rise vs Mach study which was employed as a base-line calculation for a practical aerodynamic design case.

The above tests are compared with wind-tunnel experiment and available results by other authors.

2 Mathematical Background

For the sake of completeness a brief description of the numerical algorithm is given below. More details may be found in [1, 2, 3].

The choice of discretization for the convective part of the Navier-Stokes spatial operator is driven by the following requirements:

1. Applicability to 3-D reasonably smooth 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 due to 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.
7. Relatively low amount of computational work at given accuracy.

Theoretical considerations together with extensive numerical experiment showed that the ENO-based scheme first introduced in [1, 2] and incorporated in a Multigrid framework by means of the Defect Correction approach in [3] possesses the above properties.

We assume that all the block meshes are structured. By integrating over each cell separately we get a system of ODE's which can be solved

by a time-stepping procedure. Fluxes are approximated by a one-dimensional interpolation from nearby cell centers.

The ENO approach is implemented by choosing a template (typically consisting of 3 points in this work) which may change with iterations and determined separately in each field, primarily according to the sign of the corresponding eigenvalue and then according to the smoothness of the projected fluxes [5, 3]. The interpolated characteristic fluxes are projected back to get the Cartesian ones.

In the framework of the present method the above 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 relatively cheap upwind biased relaxation. 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 face is called sensitive if it belongs to the one-dimensional neighborhood of a sonic cell face. Natural finite differences are used to approximate the first derivatives along the local grid coordinates.

A 3-stage Runge-Kutta scheme is applied in a TVD preserving form [5] with theoretical CFL=1. To accelerate the convergence to steady-state, explicit residual smoothing is applied.

2.1 Multigrid Defect Correction Approach

For a given problem $L\mathbf{u} = \mathbf{F}$, the defect correction approach is defined by an iterative process

$$L_1\mathbf{u}^{(n+1)} = \mathbf{F} - [L_2\mathbf{u}^{(n)} - L_1\mathbf{u}^{(n)}]. \quad (1)$$

Here, \mathbf{u} and \mathbf{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 high order operator approximation (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 correction 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 opera-

tor 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 described above are added to the left-hand side of Eq. (1). L_1 is chosen as a low-order upwind-biased *one-point stencil* operator. L_2 is the ENO operator described above. 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 .

3 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 above boundary conditions, conservativity of numerical fluxes in the vicinity of block interfaces and multigrid interpolation. It is also vital to diminish the overhead due to 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.

3.1 Basic features

The present multiblock method possesses the following basic features:

1. Similar to the single-block method of [1, 2, 3], the multiblock algorithm is based on an ENO numerical scheme which is incorporated into multigrid defect-correction framework.
2. The method employs no connectivity lists or similar data structures which may slow down the data transfer. This simplifies data organization and data management, and thus reduces the communication overhead.

3. 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 previous requirement of a simple data management. The computational impact of these changes is negligible.

3.2 Multiblock/Multigrid structure

The global domain decomposition is subject to the following requirements:

1. For each block, i, j, k mesh structures are generated by means of standard grid generators.
2. In the present version of the algorithm, there is a point-to-point matching of grid lines across the blocks interface.
3. A block face may abut several neighboring blocks each of which may also adjoin a number of different block faces, 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 which are formed by merging neighboring computational volumes of the previous finer level. The numerical algorithm is embedded in the framework of the multigrid method by means of the full-approximation scheme (FAS, [6]). On the finest level, the defect-correction technique is employed as explained in 2.1.

3.3 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 grid which 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 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. 1). Due to changeability of stencils, it is prefer-

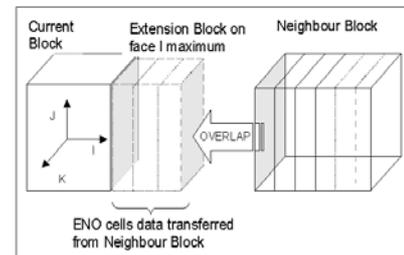


Fig. 1 Multiblock data management at merged faces

able (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 above strategy of data transfer also enables the conservativity of the numerical scheme.

3.4 Spatial approximation on block interfaces

As mentioned above in Section 3.2, 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.

A slight change in the approximation formulae was introduced in order to ensure the conservativity 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. 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).

On the other hand, viscous fluxes require two-dimensional interpolation stencils. Thus, in order to retain the flux approximation formulae 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 previous 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 may come from a block which 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.

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. With the same end in view, 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.

The same line of reasoning may 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.

3.5 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 e.g. for the standard Baldwin-Lomax [9] turbulence modeling on wakes. The preprocessor to the code automatically identifies the sharp edges which may produce a wake, labels the block faces which are geometrically induced by the above edges, and then recursively propagates the wake boundary condition to the neighboring blocks.

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

4 Computational Results

The major objective of the study was to check the ability of the modified method to allow (due to its multiblock structure) large scale calculations for realistic aerodynamic configurations while retaining its robustness and high accuracy. Solu-

tions for subsonic and transonic turbulent flow over different configurations in flight conditions where viscous effects are significant, were obtained.

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

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

4.1 Test-case ARA M100

4.1.1 Choice of a benchmark

The test-case selected is that of flow over ARA M100 wing at subsonic and transonic conditions. The flight conditions were mostly typical of cruise and the wing has transonic/transport design. The test-case was chosen due to the requirement for extensive and reliable experimental data on one hand, and due to the availability of available computations by other authors, on the other. The present case has been the subject of detailed studies in the past (see, e.g.[10, 11]).

4.1.2 Geometry and experiment

The configuration matched the geometry of a wind tunnel model tested in United Kingdom's Aircraft Research Association wind tunnel ([12]). The wing is mounted approximately centrally on the fuselage axis and has a 5° dihedral. The computational geometry used in grid generation was created by enriching the original airfoils

of [12]. A small cap was added to the wing tip (details of which were not available from [12]).

4.1.3 Grid Generation

The computational grid used here was generated using GRIDGEN V8 ([8]). To allow a fair comparison, an effort was made to generate a grid close to that constructed in [10, 11]. Similar to [10, 11], the grid topology is of C-O type, and it is stretched 16 mean chords upstream, 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, each of them of i, j, k structure. The total number of cells is about 900000. Two coarser multigrid levels were derived viz "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 [10, 11], an aeroelastic effect which caused a tip down twist was included into the geometry and computational grids, but the wing tip deflection mentioned in [10] was not applied. The fine level grid dimensions are very close to those of the grid designated in [10, 11] as a medium grid.

4.1.4 Computational runs

The total of 28 computational runs were performed based on the experimental data available in [12]. They were aimed at achieving the following comparisons:

1. Surface pressure coefficients at the fixed free-stream Mach number of $M_\infty = .80$, 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.
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 the computations, $Re = 13.1 \times 10^6$ and the flow was treated as fully turbulent.

4.1.5 Surface pressure comparisons

Wing surface pressure comparisons were performed at the angle of attack varied between -3.017° and 2.873° , and at the free-stream Mach number $M_\infty = .80$, $Re = 13.1 \times 10^6$, conditions which 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 Fig. 2 to Fig. 3 and in Fig. 4 to Fig. 5 at the angles of attack $\alpha = -3.017^\circ$, and $\alpha = 2.873^\circ$ respectively.

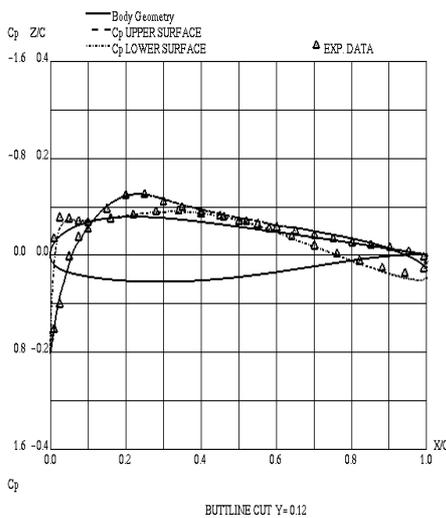


Fig. 2 ARA Cp profiles Station 1 - $\alpha = -3.017^\circ$
 $M_\infty = .80$, $Re = 13.1 \times 10^6$

The surface pressure coefficients are predicted reasonably well throughout the whole range of angles of attack. In general the pressure distributions are close to those presented in [10, 11]. The computation seems to predict both the shock

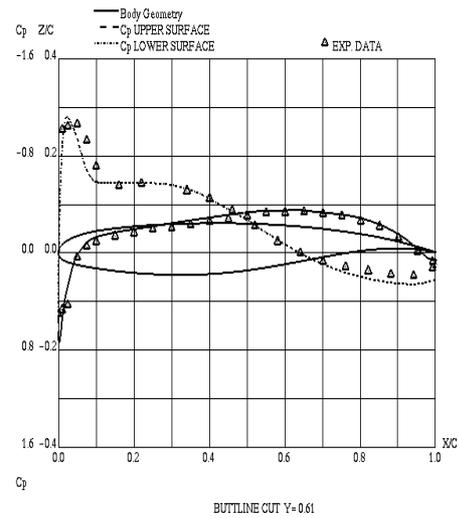


Fig. 3 ARA Cp profiles Station 3 - $\alpha = -3.017^\circ$
 $M_\infty = .80$, $Re = 13.1 \times 10^6$

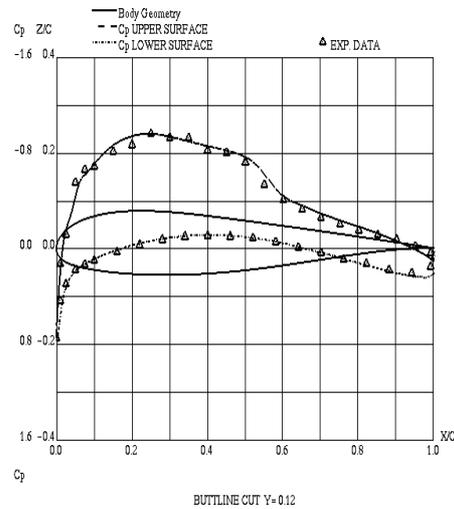


Fig. 4 ARA Cp profiles Station 1 - $\alpha = 2.873^\circ$
 $M_\infty = .80$, $Re = 13.1 \times 10^6$

on the lower surface at $\alpha = -3.017^\circ$ and at the higher positive angles of attack. Somewhat surprisingly, the computation at $\alpha = 2.873^\circ$ predicts the shock location apparently better than the computation in [11, 12], performed with the

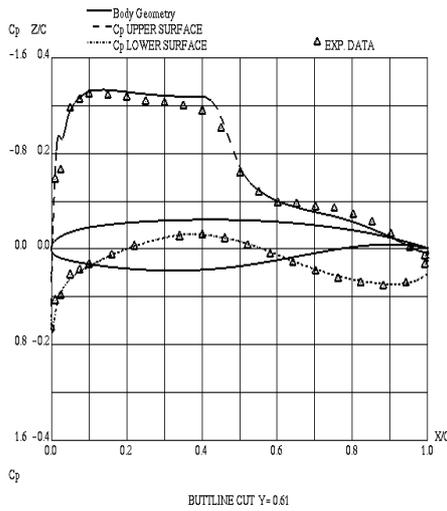


Fig. 5 ARA Cp profiles Station 3 - $\alpha = 2.873^\circ$
 $M_\infty = .80, Re = 13.1 \times 10^6$

same turbulence model. This may 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 115000 computational cells. The effect of refinement is most significant at the highest computed angle of attack $\alpha = 2.873^\circ$ (see Fig. 6) where the two pressure distributions are compared at the section $2y/b = .33$. Still the discrepancy is mostly confined to the shock region where the finer computation exhibits a sharper shock pattern. On the whole, the two 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.

4.1.6 Aerodynamic coefficients

Fig. 7 shows the lift coefficient curve as function of angle of attack at $M_\infty = .80$. The medium and

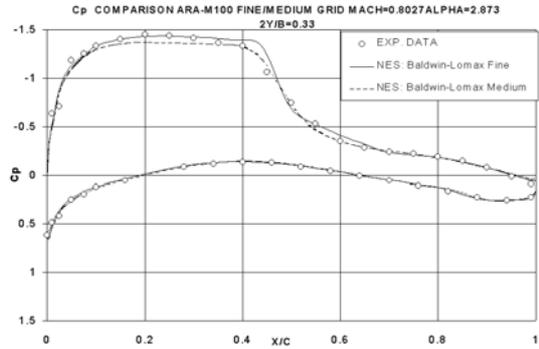


Fig. 6 medium versus fine grid result

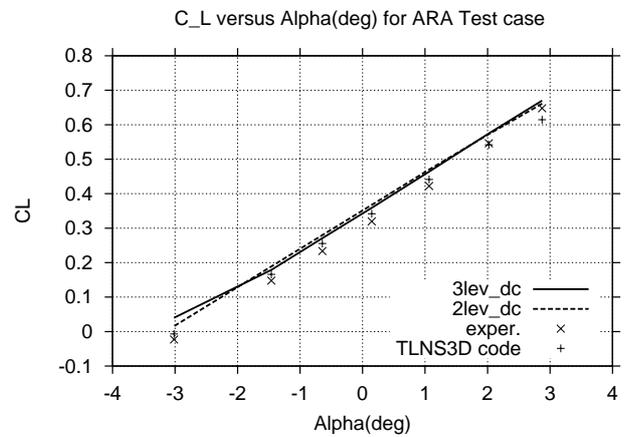


Fig. 7 ARA lift curve

fine grid results by the current method are compared to experiment and to the results of [10, 11]. 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 [10, 11] (at approximately the same grid resolution) are closer to the experimental curve than the current results but a computation on a doubly fine grid([11] yielded a much higher lift, which is attributed in [11] to the over sensitivity of a specific turbulence modeling of the computation and to a need in additional aeroelastic corrections. A three-level computation drag polar C_L vs C_D (Fig. 8) 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

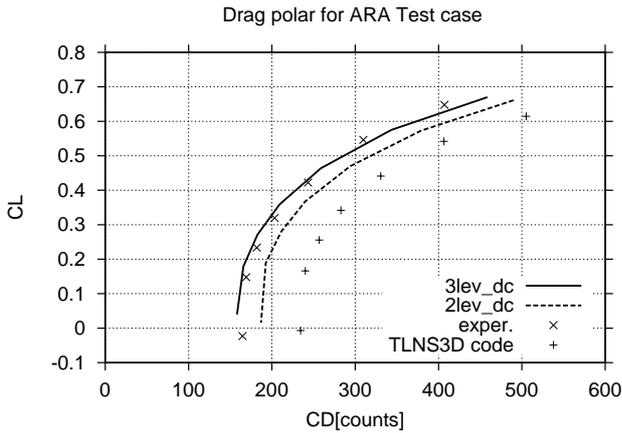


Fig. 8 ARA drag polar

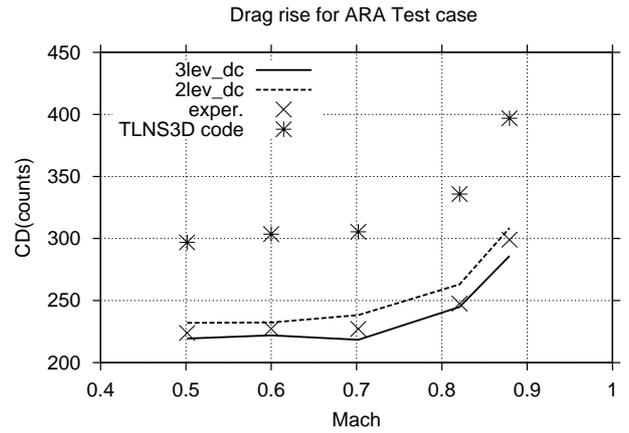


Fig. 9 ARA drag divergence rise

.0007 – .0008) lower than the experimental one. The drag values are clearly much nearer to the experiment than those reported in [10] for the same grid resolution. On a doubly fine grid of [11], a much better matching is reported than in [10] especially when Menter’s turbulence model is employed, with the best zero lift drag estimate of about 12 counts higher than the experimental value. It can be noted that the zero lift drag values predicted on the medium (about 115000 cells) and fine (about 900000 cells) grids of the present computation differ significantly less than the estimates achieved in [10, 11] on the grids which comprise about 900000 and close to 2000000 points, respectively.

4.1.7 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. 9) illustrates the increase in drag due to 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 [10, 11] (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.

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

Fig. 10 shows a plot of drag coefficient vs Mach number at an angle of attack of 0° . The com-

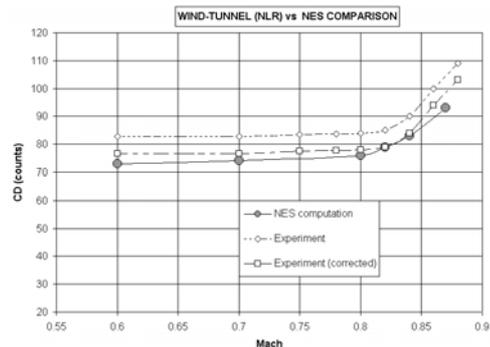


Fig. 10 Fuselage drag rise

putational data set is compared to the raw experimental data of [13] and to a corrected experimental curve where an estimated correction of about 6 counts was applied. The computational and experimental drag values compare both qualita-

tively and quantitatively thus indicating the ability of the code to predict sensitive characteristics for complex aerodynamic shapes.

5 Conclusions

A multigrid solver has been developed and applied to a 3D 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 a good grid convergence.

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 allows the code to be used for practical large-scale aerodynamic analysis and design in the engineering environment.

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.

- [2] Epstein, B., Jacobs, A., and Nachshon, A., Aerodynamically Accurate Three-Dimensional Navier-Stokes. *AIAA Journal*, Vol. 35(6), pp 1089-1091.
- [3] Epstein B., Averbuch A., and Yavneh I, An Accurate ENO Driven Multigrid Method Applied to 3D Turbulent Transonic Flows. *J. of Computational Physics*, Vol. 168, pp 316-338, 2001.
- [4] Harten, A., Engquist, B., Osher, S., and Chakravarthy, S., Uniformly High Order Accurate Non-oscillatory Schemes. *I. J. Comput. Phys.*, Vol. 71, pp 231-303, 1987.
- [5] Shu, C.-W., and Osher, S., Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes. *J. Comput. Phys.*, Vol. 83, pp 32-78, 1989.
- [6] Brandt, A., Multi-Level Adaptive Computations in Fluid Dynamics. *AIAA Journal*, Vol.18, No. 10, pp 100-108, 1980.
- [7] Séror, S., Rubin, T. & Epstein, B., Construction of a Multiblock 3D Full Navier-Stokes Code for Practical Aerodynamic Computations. *Proc of 41th Israel Annual Conference on Aerospace Sciences*, pp 231-241, 2001.
- [8] 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 Air Force base, 1990.
- [9] Baldwin, B.S., Lomax, H., Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows. *AIAA paper*, 78-257.
- [10] 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.
- [11] website [http : /hpccp_www.larc.nasa.gov](http://hpccp_www.larc.nasa.gov)
- [12] Carr, M.P. and Pallister,K.C., Pressure distributions measured on Research Wing M100 Mounted on an Axysymmetric Body. AGARD AR-138 Addendum, 1984.
- [13] Ernst, Y., IAI WW1125-E1 1:14 model NLR Wind-Tunnel results, report 18931, IAI, 1978.