Journal of Engineering Mathematics **27**: 293–307, 1993.

Numerical aspects of a block structured compressible flow solver

B.J. GEURTS* and H. KUERTEN

Department of Applied Mathematics, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands
(**author for correspondence*)

Received 28 April 1992; accepted in revised form 20 January 1993

Abstract. A block structured compressible flow solver based on a finite volume approach with central spatial differencing is described and its performance in 2D on flow around an airfoil is studied. Variations in the number and dimensions of the blocks do not influence the convergence behavior nor the solution, irrespective of the relative positions of a possible shock and the block-interfaces. Mixed calculations, in which the governing equations, either Euler or Reynolds averaged Navier-Stokes, differ per block, give accurate results provided the Euler blocks are defined outside the boundary layer and or in the far field wake region. Likewise, extensive grid distortions near block interfaces can be allowed for outside the boundary layer. Finally, an unbalanced advancement in time, in which each block is advanced independently over several time steps gives no serious decrease in convergence rate.

1. Introduction

The Reynolds averaged Navier-Stokes equations form a suitable basis for simulations of complex flow fields in which viscous effects play an important role, such as occur e.g. in transonic compressible flow around an airfoil. Accurate predictions of quantities of aeronautical interest (e.g. drag- and lift coefficients) predominantly require the use of structured grids. In order to treat more complex geometries such as a multi-element airfoil, it is therefore natural to concentrate on a block structured flow solver. This allows for a total grid composed of several blocks, each of which is represented by a structured grid and appropriate boundary conditions. In this way flow problems of increased complexity can be envisaged. The construction and performance of such a block structured flow solver are described in this paper. At present, only block structured grids in which the grid lines are continuous over block boundaries are considered. It will be shown that the introduction of such a block structure on the grid does not seriously influence the convergence behavior when compared to a situation without a block structure. This remains true even if the position of a shock coincides with a block interface. Moreover, mixed calculations in which the Navier-Stokes equations are defined in the boundary layer and the Euler equations outside this region yield accurate simulation results. However, in such cases the simulation results depend sensitively on the exact position of the 'Euler-Navier-Stokes' interface. The mixed calculation results differ considerably from the Navier-Stokes results even if the Euler blocks have their interface with Navier-Stokes blocks just inside the boundary layer. On the other hand the simulation results fully agree, for any definition of the Euler blocks outside the boundary layer. Extensive, local grid distortions near block interfaces do not seriously influence the results provided these distortions are small inside the boundary layer. Finally, the blocks can be advanced in time independently over several time steps without causing the rate of convergence to decrease significantly and giving an accurate prediction of aerodynamic properties (relative errors smaller than 1 percent). A significant amount of calculation time can hence be saved through the inclusion of Euler blocks outside the

boundary layer and (to a lesser extent by) the independent advancement in time of the blocks during several time steps without affecting the solution. The latter property opens the possibility of parallel processing of these flow problems, to which study will be devoted in the future. Moreover, in order to increase the convergence towards the stationary state, an extension containing a multigrid acceleration is presently being considered. The present results indicate a multigrid-multiblock algorithm in which the multiblock loop is put inside the multigrid cycle.

The present block structured flow solver will be used in the more generally applicable three-dimensional ISNaS flow solver [1]. In this paper we restrict to algorithmic aspects. Details about software aspects can be found in ref. [2]. In the ISNaS project a flow solver based on the Reynolds averaged Navier-Stokes equations will be developed. Under transonic flow conditions turbulent flow in the near wing and wake regions dominates the predictions of the basic flow quantities. The inclusion of an eddy viscosity turbulence model (e.g. Baldwin-Lomax [3]) represents approximately the effects of small scale turbulent phenomena and renders the equations suitable for a numerical treatment. Without such a model, the resolution required to capture the small scale phenomena would make the problem intractable even for the largest computers available. The accuracy of the predictions, however, depends in part on how well the small scale turbulent phenomena are captured in the model. This is an area of active research which gradually may lead to more adequate models which are valid over a wide range of flow conditions. In particular Large Eddy Simulations (LES) and Direct Numerical Simulations (DNS) of turbulent flow in simple geometries can be used for calibrating the 'coarse grained' turbulence models which are incorporated in a Reynolds averaged Navier-Stokes treatment of a complex flow problem.

The highly nonlinear system of equations which result after the modeling of the Reynolds stresses is treated within a finite volume approach with central differencing in space and an explicit multistage Runge-Kutta scheme for the time-integration. This is described in detail in section 2 in which also some simulation results obtained with this 'monoblock' solver are presented. The construction of a block structured flow solver based on this method is sketched in section 3 and a general class of test-cases is introduced. This allows a detailed and separate analysis of the various extensions which are included in the block structured flow solver. The number and dimensions of the blocks as well as the governing equation per block are varied in the testcases. Moreover, a grid distortion procedure is introduced in order to study the effects of grid-irregularities near block interfaces. Finally, the effect of advancing the separate blocks independently in time over several time units is investigated. The simulation results will be considered in section 4 and we summarize our findings in section 5.

2. A monoblock flow solver

In this section we describe the numerical method used in the flow solver and show some results of the monoblock solver for comparison with the results obtained with the block structured flow solver, in section 4.

The two-dimensional, compressible Navier-Stokes equations can be written in the integral form as

$$\frac{\partial}{\partial t}\left[\int\int_{\Omega} U \,\mathrm{d}x\,\mathrm{d}y\right] + \int_{\partial\Omega} (F\,\mathrm{d}y - G\,\mathrm{d}x) = 0\,, \tag{1}$$

where U represents the vector of dependent variables,

$$U = [\rho, \rho u, \rho v, E]^T , \tag{2}$$

with ρ the density, u and v the Cartesian velocity components, and E the total energy density. Further, Ω is an arbitrary part of the two-dimensional space with boundary $\partial\Omega$ and F and G are the Cartesian components of the total flux vector. This flux vector consists of two parts: the non-dissipative or 'convective' part and the dissipative or 'viscous' part, which describes the effects of viscosity and heat conduction, and involves first order spatial derivatives. In order to describe the effects of the small scale turbulent phenomena the Navier-Stokes equations are averaged over a sufficiently large time interval. Due to the nonlinear terms in the convective fluxes, the resulting Reynolds averaged Navier-Stokes equations involve averages of products of two velocity components. These terms are modeled by a suitable turbulence model. In the present paper the algebraic Baldwin-Lomax turbulence model, in which the unknown terms are modeled by eddy viscosity terms, is chosen [3]. In this way the flux vectors are given by

$$F = [\rho u, \rho u^2 + p - \sigma_{xx}, \rho uv - \sigma_{xy}, u(E + p) - u\sigma_{xx} - v\sigma_{xy} + q_x]^T , \tag{3}$$

$$G = [\rho v, \rho uv - \sigma_{xy}, \rho v^2 + p - \sigma_{yy}, v(E + p) - u\sigma_{xy} - v\sigma_{yy} + q_y]^T , \tag{4}$$

where the pressure p is related to the dependent variables by the equation of state, σ_{xx}, σ_{xy} and σ_{yy} are the components of the viscous stress tensor, and q_x and q_y are the components of the heat flux vector. The effects of turbulence are incorporated in the viscous stress tensor and the heat flux vector through a replacement of the laminar viscosity μ by the total viscosity $\mu + \mu_t$, where μ_t is predicted by the Baldwin-Lomax model in the present study.

The discretization of the Navier-Stokes equations follows the method of lines, i.e. the spatial discretization is performed first, and subsequently the resulting set of ordinary differential equations is integrated in time, until the steady state solution is obtained. First the computational domain is partitioned in quadrilateral cells with the help of a structured, boundary-conforming grid. For the spatial discretization a finite volume method is used in which the integral form of the Navier-Stokes equations is applied to a control volume Ω, consisting of a number of grid cells. The basis of the finite volume method used here is a cell vertex method with overlapping control volumes [4, 5, 6, 7]. In this method the variables are stored at the grid points and the control volume for each grid point consists of the four neighbouring grid cells. Since the variables are known at the grid points, the non-dissipative flux through the boundaries of a control volume can be approximated accurately using the trapezoidal rule. The control volume is sketched in Fig. 1. It is known that this method is more accurate than the cell-centered method on non-uniform grids [8].

As the viscous flux vector involves spatial derivatives of the state vector U, application of the same control volume for the calculation of the viscous fluxes would lead to a large control volume, which is impractical near block boundaries. Therefore, for the viscous fluxes an auxilary control volume, indicated in Fig. 1 by the area enclosed by the dashed lines, is used. The spatial derivatives of the state vector in the corner points of this control volume can be approximated with the use of Gauss' theorem on a grid cell. The total viscous flux can then be calculated using the trapezoidal rule. For example the quantity $\partial u/\partial x$, appearing in the viscous stress tensor, is in the point $(i + 1/2, j + 1/2)$ approximated by

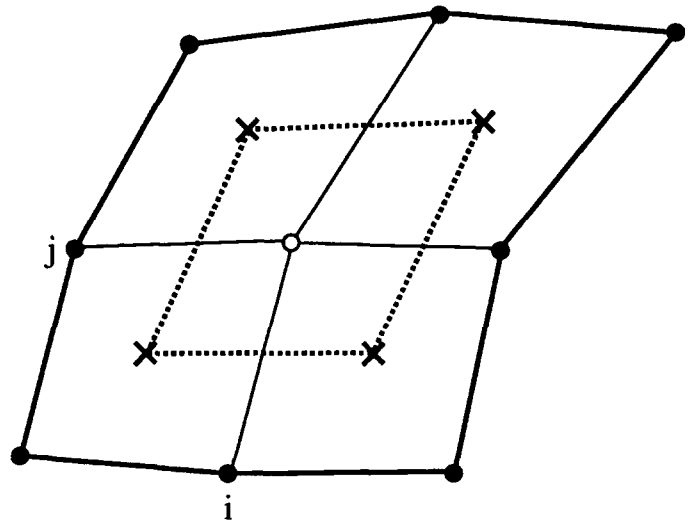

Fig. 1. Sketch of the primary control volume (full lines) and the auxilary control volume (dashed lines) around node (i, j). The primary control volume is the union of the four adjacent grid cells and the auxilary control volume is defined as the cell spanned by the centers of the four adjacent grid cells.

$$\frac{\partial u}{\partial x} \approx \frac{1}{\Omega_e} \int \int_{\Omega_e} \frac{\partial u}{\partial x} \, \mathrm{d}x \, \mathrm{d}y = \frac{1}{\Omega_e} \int_{\partial \Omega_e} u \, \mathrm{d}y \,, \tag{5}$$

where Ω_e is the grid cell with center $(i + 1/2, j + 1/2)$. Since the variables are located at the grid points, u can be calculated in the corner points of the line integral at the right hand side, and the integral is approximated using the trapezoidal rule.

The method of central differencing leads to an uncoupling of odd and even grid points and to oscillations near shock waves. Even for viscous flow calculations the presence of viscous dissipation is not sufficient to damp these instabilities outside shear layers. Therefore, nonlinear artificial dissipation is added to the basic numerical scheme. This artificial dissipation consists of two contributions: fourth order difference terms which prevent odd-even decoupling, and second order difference terms to resolve shock waves. The second order terms are controlled by a shock sensor, which detects discontinuities in the pressure. In the present flow solver the artificial dissipation in the boundary layers, where the viscous dissipation should be dominant, can be reduced by multiplication with the ratio of the local Mach number and free-stream Mach number. For laminar flow at low Reynolds numbers this reduction of the artificial dissipation is not necessary. However, in case of high Reynolds number flows, multiplication of the artificial dissipation in the normal direction with the ratio of the local and free stream Mach number may lead to a reduction in the drag of approximately 10 percent, and is always used in the calculations presented here. The role of the artificial dissipation in relation to the viscous dissipation is discussed in more detail in [9]. There it is shown that the artificial dissipation can be chosen in such a way that it is comparable with the inherent artificial dissipation in a third order upwind scheme, obtained with the MUSCL approach used on Roe's first order upwind scheme.

The spatial discretization leads to the system of equations:

$$S_{i,j} \frac{\mathrm{d}U_{i,j}}{\mathrm{d}t} + C_{i,j} + V_{i,j} + D_{i,j} = 0 \,, \tag{6}$$

where $S_{i,j}$ is the surface area of the union of the four neighbouring grid cells at node (i, j), $C_{i,j}$ is the discrete non-dissipative flux, $V_{i,j}$ is the viscous flux, and $D_{i,j}$ is the artificial dissipation. The artificial dissipation is chosen according to Martinelli [10]

$$D_{i,j} = D_{i+1/2,j} - D_{i-1/2,j} + D_{i,j+1/2} - D_{i,j-1/2}\,, \tag{7}$$

where for example

$$D_{i+1/2,j} = S_{i+1/2,j}[\epsilon^{(2)}_{i+1/2,j}(U_{i+1,j} - U_{i,j}) - \epsilon^{(4)}_{i+1/2,j}(U_{i+2,j} - 3U_{i+1,j} + 3U_{i,j} - U_{i-1,j})]\,, \tag{8}$$

in which

$$\epsilon^{(2)}_{i+1/2,j} = \kappa^{(2)} \max\{\nu_{i+1,j}, \nu_{i,j}\}\,, \tag{9}$$

$$\epsilon^{(4)}_{i+1/2,j} = \max\{0, \kappa^{(4)} - \epsilon^{(2)}_{i+1/2,j}\}\,, \tag{10}$$

and

$$\nu_{i,j} = \frac{|p_{i+1,j} - 2p_{i,j} + p_{i-1,j}|}{p_{i+1,j} + 2p_{i,j} + p_{i-1,j}} \tag{11}$$

is the shock sensor. Here, $\kappa^{(2)}$ and $\kappa^{(4)}$ are constants, and $S_{i+1/2,j}$ is a scaling factor, optionally multiplied with the ratio of the local and free-stream Mach numbers as described above.

At the solid wall boundaries the no-slip condition is used. The density and energy density in the grid points on a solid wall are calculated by solving their respective discrete conservation laws, using the two adjacent cells within the computational domain and their mirror images inside the wall as control volume. The values of the density and energy density in the grid points inside the walls are adjusted such that the adiabatic wall condition is approximated. The boundary conditions at a (subsonic) far-field boundary are based on characteristic theory. A reduction of the computational domain is possible if a single vortex is superimposed on the incoming free stream outside the computational domain [11].

The large system of ordinary differential equations (3), is integrated in time using a time-explicit multistage Runge-Kutta method. In the present flow solver a three-stage scheme is incorporated in which the dissipative terms (both viscous and artificial) are calculated only at the first stage in the three-stage scheme. In this way not only calculation time is saved, but the stability region of the method is increased as well. Extra calculation time is saved by advancing each grid point at the maximum time step according to its own stability limit. In this way the evolution from the initial solution to the steady state is no longer time accurate, but the steady state solution obtained, is unaffected. An improvement of the starting solution can be obtained by using 'grid sequencing'. The iteration process is started at a coarse grid and after several time steps the solution is interpolated to a finer grid. This procedure is repeated until the finest grid is reached. In this approach the solution on the coarser grids usually differs significantly from the final converged solution on the finest grids. Hence, grid-sequencing including coarse grids is not always very effective in turbulent calculations.

The capabilities of the present flow solver in a monoblock calculation can be seen from the steady state pressure distribution for transonic flow around a CAST7 profile as shown in Fig.

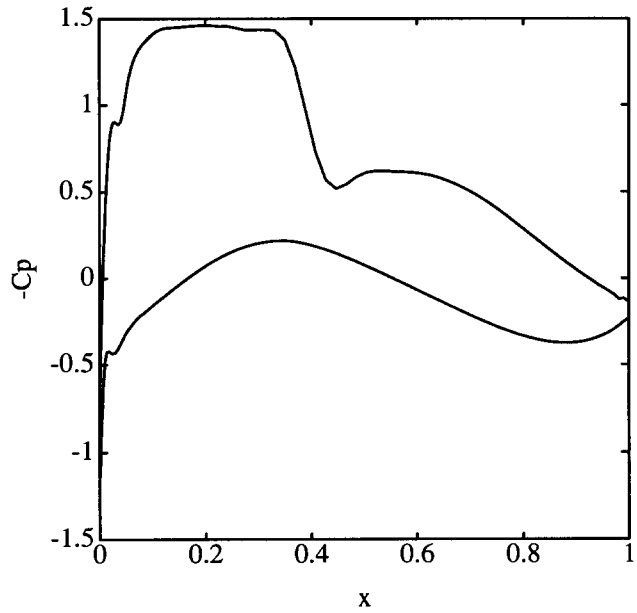

Fig. 2. Steady state pressure distribution around an airfoil obtained with the monoblock solver.

2. The calculation is performed at a Reynolds number of 4×10^6, free stream Mach number of 0.7 and angle of incidence of 2 degrees. A C-type grid with 257 grid points in streamwise direction, 193 of which are distributed along the airfoil, and 65 grid points in normal direction was employed. The grid points are non-uniformly distributed, with the outer boundaries at approximately 20 chord lengths away from the airfoil, and the first grid line in normal direction at a value of y^+ around 2. For this calculation the Baldwin-Lomax turbulence model with fixed transition at 7% chord length on both sides of the airfoil is used. The steady state solution is sufficiently approximated after about 10000 time steps on the finest grid. The total calculation time can be significantly reduced with the help of a multigrid technique and residual averaging [9], but in order to be able to compare the results of the monoblock solver with the multiblock solver, these acceleration techniques are not considered in this paper.

3. A block structured flow solver; Test-cases

In this section we describe the extension of the monoblock solver to a block structured flow solver. This generalization will be worked out in detail for the two-dimensional case. A three-dimensional generalisation of this sketch is relatively simple. We restrict to cases in which only one type of boundary condition may be assigned to each edge of a block. Moreover, grid lines are assumed to be continuous at block boundaries. Subsequently, we introduce a set of test cases allowing a detailed and separate investigation of the various extensions.

The description of the monoblock solver given in section 2 suggests a natural extension to a multi-block solver. Each separate block is considered to be covered by a structured grid.

These blocks each contain 'inner' and 'boundary' points. The 'inner' points of each block can be treated in an entirely similar way as in the monoblock solver. The 'boundary' points can be classified as lying either in the interior of an 'edge' of a block or at a 'vertex' of a block. These 'boundary' points can be treated in the same way as in the monoblock solver, i.e. the variables are updated after each stage in the Runge-Kutta scheme according to the boundary condition which locally applies. The main extension lies in both the increased variety of conditions relevant at the boundaries and in the occurrence of a larger class of 'special' points associated with 'vertices' of the blocks. Also, the governing equation can be different for each individual block, and the time-advancement can, in principle, be chosen independently.

We introduce a *basic* block which in an actual calculation can have a multitude of different realisations. It is most convenient to discuss a basic block in computational domain. A basic block has points labeled with i and j for the streamwise and normal directions respectively. Such a block has four edges and four vertices and to each of these a specific boundary condition is imposed. Such a condition either corresponds to an actual condition corresponding to the full flow domain out of which the block was taken, or it is an internal boundary. The treatment of the boundary conditions and the application of the spatial discretisation scheme requires (at most) two additional 'layers' of dummies which, strictly speaking, are outside the range of the basic block. These layers of dummy variables are extracted from the blocks adjacent to the 'current' block, hence defining a 'dressed' block. Obviously, in order to fully specify a specific realisation of a basic block additional information should be added to the dressed block, indicating the actual boundary conditions which apply, the type of equation to be used (Euler or Navier-Stokes) and the 'time-lag', i.e. the number of time steps to be taken independently per block before turning to the next block.

Obviously, specific blocks that can be defined in the flow region can be mapped to a specific realisation of the basic block. The propagation of a particular block in a given flow problem then proceeds as follows. First, the grid-points and instantaneous values of the variables are extracted from the full grid. Then the block is 'dressed' with two layers of dummies taken from the appropriate adjacent blocks and the information defining the 'types' of edges and vertices as well as the 'type' of governing equation is collected. This realisation of the basic block is then treated in the same way as in the monoblock solver, i.e. the inner points are updated first and the values of the variables at the block boundaries are adjusted, according to the boundary condition imposed, afterwards. This is done in each stage of the Runge-Kutta scheme, and it is understood that during a full time-step the values of the dummy variables are updated in the same way as in the monoblock solver. Dummy-variables corresponding to 'inner' points of the full grid remain fixed to the instantaneous value assumed when the 'current' block was realised. Once each block in the total flow region has been treated in this way, the total propagated solution is updated, taking into account the fact that points at block-interfaces have been treated more than once. One particularly simple way to deal with this is to take the average of the propagated solutions at the block interfaces. A repeated application of the above algorithm gradually takes an initial solution over to a fully converged solution of the flow problem at hand. With this method the results and convergence behavior are independent of the order in which the blocks are treated. In order to study the operation of this solver, we next introduce a set of test-cases. Simulation results will be presented in section 4.

A flexible, relatively general class of test-cases can be based on an existing single block

grid. A variation in the number of blocks, the type of governing equation per block (i.e. Euler or Navier-Stokes), a distortion of the grid near block interfaces as well as the introduction of a time lag can be studied separately and in a controlled way.

We consider transonic flow around a two dimensional airfoil. This flow problem was treated successfully with the monoblock solver described in the previous section on a globally structured C-type grid. Based upon this grid a multi-block test case can be derived with which the solver as sketched above can be adequately tested and the performance of the full algorithm investigated. We consider cases with $2n$ blocks as sketched in Fig. 3. This simple structure allows for an easy variation of the number of blocks and the type of governing equation per block. In a natural way an 'inner' ($j < j_s$) and an 'outer' ($j > j_s$) region is introduced and variation of j_s can give some appreciation of the influence of Euler-blocks in the outer region on the total (Navier-Stokes) results.

The generation of a (good) multi-block grid in more realistic situations, such as a multi-element airfoil, is a complicated matter. One of the main problems to overcome is to generate a 'smooth' connection between the grid-lines in the various adjacent blocks. In general some 'non-smoothness' of the grid at block-interfaces must be anticipated and in order to simulate this in our test-cases a 'grid-distortion' near block interfaces at $j = j_s$ is considered. The distortion algorithm generates structured grids and, most importantly, an iterative application of this algorithm generates extensive grid-distortions globally. Figure 4 displays an example of a distorted grid. Of course, this (and any) distortion is 'ad hoc' but the resulting grid seems to reflect the basic multi-block grid problems in more realistic cases. One clearly observes the sharp 'kink' in the grid along the line $j = j_s$. In the next section, we consider simulations on both smooth and distorted grids. In addition, we will consider the effects related to the coupling of blocks in which different governing equations are defined, i.e. Euler or Navier-Stokes and finally turn to effects of a time lag in advancing the individual blocks.

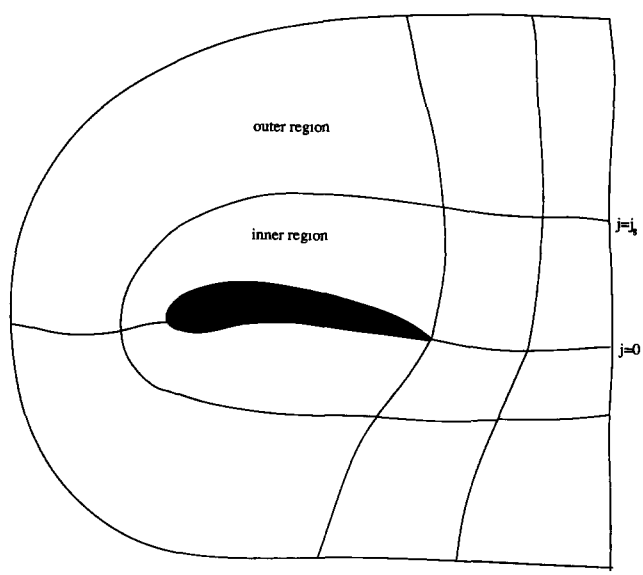

Fig. 3. Sketch of the subdivision of the flow region around an airfoil into an even number of blocks. An 'outer' and an 'inner' region are defined naturally at j_s.

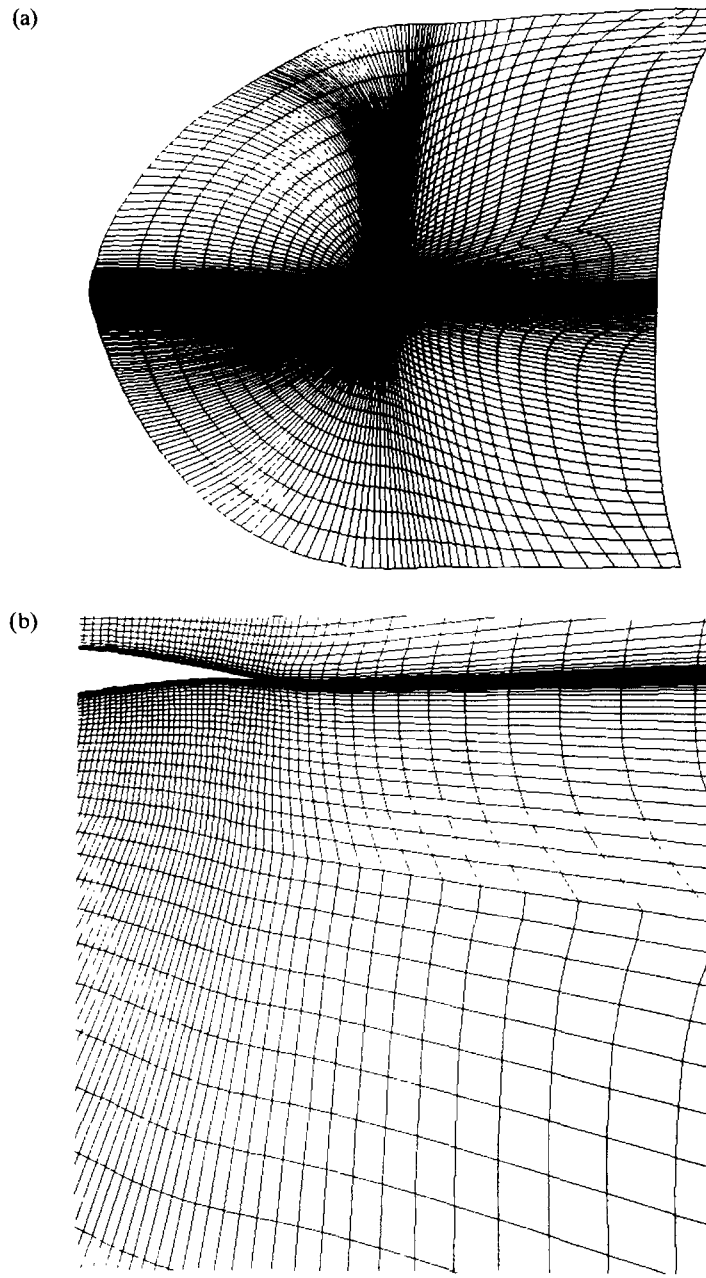

Fig. 4. Example of a distorted grid generated from the CAST7 grid, with a significant distortion near $j_s = 32$. In (a) a total view of the grid is shown and in (b) a portion of the grid is enlarged.

4. Simulation results in 2D

We describe some simulations performed on the test-cases introduced in the previous section in order to obtain an appreciation of the various extensions relevant to the multi-block

solver. First, we consider some basic simulations in which only a block structure is introduced on the flow domain. Then the effects of defining different governing equations per block will be presented and simulations performed on (highly) distorted grids are considered. Finally, the effect of a 'time-lag' will be studied. We will focus attention on the pressure distribution on the airfoil (C_p), the drag- lift- and momentum coefficients (C_d, C_l, C_m) and on the convergence behavior of the block structured flow solver.

Basic simulations

We first consider cases in which the Navier-Stokes equations are used in each block and concentrate on the effects related to the introduction of a block structure on the grid. A 3-stage Runge-Kutta scheme was adopted for the time integration and the parameters are the same as in section 2. Different settings of the total number of blocks, n_{blocks}, have been considered as well as variations in the location of the interfaces. The influence of such variations on the behavior of the solver is small in case the width in either direction is $\geqslant 8$. If this width was reduced to 4, convergence problems arise. This is due to the grid sequencing, by which use only one grid cell on the coarsest grid remains. With $n_{blocks} = 12$ we obtain the 'smallest' case containing all different types of edges and vertices arising in a multiblock treatment of a single airfoil and hence this value is adopted in the sequel. We considered in total five different test-cases; $j_s = 8$, 12, 16, 20 and 24 respectively. The first two correspond to cases in which a block-interface is defined within the boundary layer and the latter two represent examples of block-interfaces outside the boundary layer. The convergence history appears to be independent of the introduction of a block structure. This holds irrespective of the position of the shock in relation to the block boundaries. Notice that in the multiblock treatment a difference inside the Runge-Kutta stages occurs near the block interfaces. The 'neighbouring variables' of the interfaces lying inside the flow domain, but outside the 'current' block are not updated during the time step but remain equal to the value assumed when the current block was realised. This does not apply in a single block treatment. A study of the C_p-profiles corresponding to these cases revealed that each of these cases give simulation results which differ less than 'linewidth' from each other and fully agree with the monoblock result shown in Fig. 2. Finally, the aerodynamic coefficients do not show any dependence on the parameter j_s as well. In conclusion, the introduction of a block-structure on the full grid does not have any significant influence on any of the measures that we have concentrated upon, even in case a block interface coincides with the position of the shock on the upper part of the airfoil. These results were obtained with the Baldwin-Lomax eddy viscosity model. It is expected that the multiblock algorithm remains valid also if more advanced turbulence models (e.g. $k - \epsilon$ models) are used.

Euler Navier-Stokes coupling

The block-structure on the computational domain allows for a variation of the governing equation per block. The influence of the introduction of Euler-blocks in the 'outer' region on the simulation results will be considered next. Intuitively speaking, this influence will be small if the Euler blocks are defined in regions of the flow outside the boundary layer. We will quantify this by considering the cases presented in the previous section, in which we adopt the Euler equations in case $j > j_s$. We will also consider the additional definition of the Euler-equations in the blocks in the far field wake region behind the airfoil.

Table 1. j_s-dependence of C_x-coefficients in mixed calculations.

j_s	$C_d^{(1)}$	$C_d^{(2)}$	$C_l^{(1)}$	$C_l^{(2)}$	$C_m^{(1)}$	$C_m^{(2)}$
8	0.011272	0.011265	0.342001	0.342005	−0.048803	−0.048797
12	0.010934	0.010930	0.402167	0.402266	−0.045788	−0.045790
16	0.015136	0.015132	0.835702	0.835732	−0.109931	−0.109935
20	0.015313	0.015310	0.843383	0.843434	−0.111077	−0.111080
24	0.015315	0.015312	0.843493	0.843518	−0.111089	−0.111088

Table 1 shows the 'C_x-coefficients', i.e. drag-, lift and momentum coefficients C_d, C_l and C_m respectively. These values should be compared with $C_d = 0.0153\ldots$, $C_l = 0.843\ldots$ and $C_m = -0.111\ldots$ which are obtained in the monoblock calculation. The index (1) corresponds to a definition of Euler blocks for $j > j_s$ only and the index (2) implies the additional definition of Euler blocks in the far field wake region. Since this region is well separated from the trailing edge, the differences in the coefficients in these two cases are negligible. The value of j_s is more important. The results are quite accurate if Euler blocks are defined outside the boundary layer; in this case corresponding to, say, $j_s = 16$. Observe the sharp transition when passing from $j_s = 12$ (just inside the boundary layer) to $j_s = 16$ (just outside the boundary layer). Thus, the acceleration of the calculation by the introduction of Euler blocks in the 'outer' flow field which is feasible in the multiblock solver, should be considered with some care. The application of Euler instead of Navier-Stokes leads to negligible changes only if all the Euler-blocks are defined outside the boundary layer and/or in the far field wake region.

In Fig. 5 we present the influence on the C_p-profiles and the convergence histories of the various cases showing the influence of the introduction of the Euler-blocks. The effect of the inclusion of an extra pair of Euler blocks in the far field wake region is negligible. We observe that the results corresponding to $j_s = 8$ and 12, i.e. Euler blocks defined well within the boundary layer, are not acceptable. At small j_s the convergence in these mixed calculations is quite bad, which is not surprising since at the block interfaces the viscous fluxes do not match. This effect becomes more prominent as j_s decreases. Moreover, convergence in these cases is much slower than in the other three cases, and this difference becomes more prominent on the finer grids. Finally, corresponding to the results summarized in Table 1, the differences in the results obtained with $j_s \geq 16$ are relatively small. At $j_s = 16$ only the position of the shock is predicted with a small error and the rest of the profile corresponds in close agreement with the 'all Navier-Stokes' results. In conclusion, the introduction of Euler-blocks outside the boundary layer does not seriously affect the simulation results. However, it does result in a reduction of the computation time required for the simulation.

Distorted grid results

The occurrence of Euler blocks well within the boundary layer was found to influence the simulation results. As a further complication for the solver, we consider the effects related to an additional grid-distortion, both in the 'near-wing' as well as in the 'far-field' region. We consider two different cases, $j_s = 16$ and 32, which in view of the previous discussion correspond to a case just outside the boundary layer and a 'far field' case respectively. Extensive grid distortion in the boundary layer is generated at $j_s = 16$ whereas at $j_s = 32$ the grid in the boundary layer is almost unchanged. Again, we concentrate on the convergence

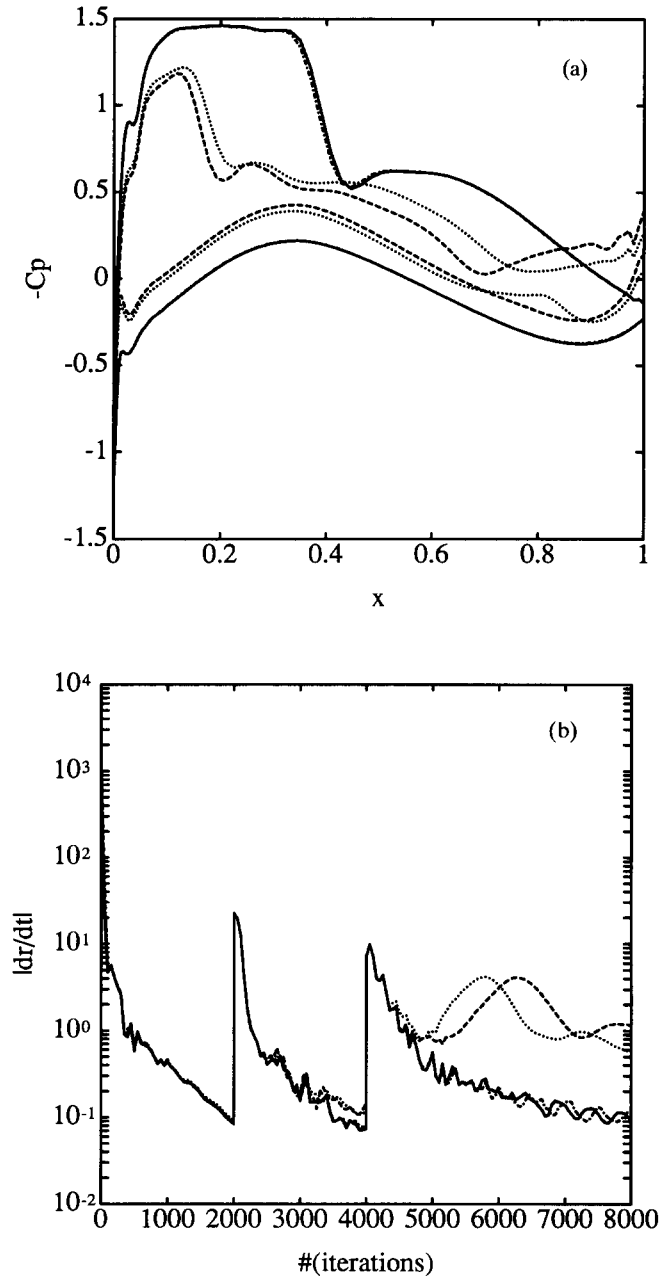

Fig. 5. The C_p-profiles (a) and the convergence history of $|d\rho/dt|$ (b) for the mixed simulations. The results corresponding to $j_s = 8$, 12 and 16 are shown as dashed, dotted, chain-dotted respectively and the results for $j_s = 20$, 24 are drawn as full curves showing complete agreement between the latter two settings.

history, the C_p-profile and the C_x-coefficients obtained for these distorted grid cases, both in a mixed and a full Navier-Stokes simulation.

In Fig. 6 we present the C_p- profiles both for the mixed and Navier-Stokes calculations. The C_p-profiles corresponding to the distorted cases at $j_s = 32$ agree with the full Navier-Stokes results on the original smooth grid to within 'line-width', i.e. no effect of grid-

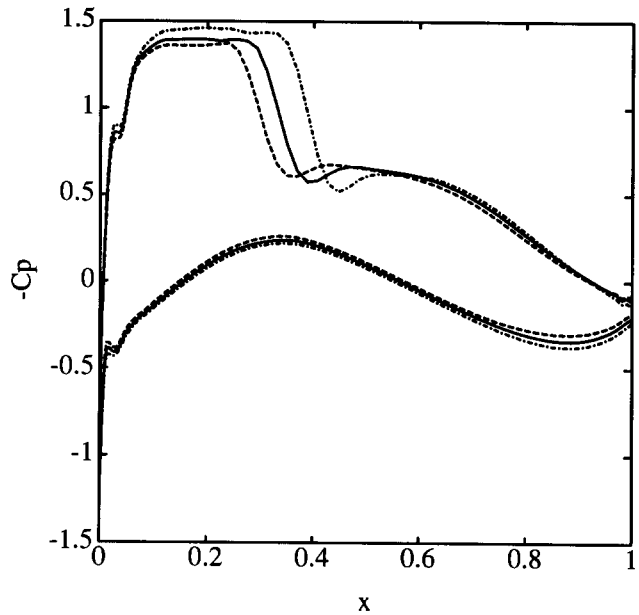

Fig. 6. The C_p-profiles for the distorted grid simulations. The results corresponding to $j_s = 16$ are shown as solid (full Navier-Stokes) and dashed curves (mixed calculation) whereas the results for $j_s = 32$ are drawn as dotted curves (full Navier-Stokes) and chain-dotted curves (mixed calculation), showing full agreement between the latter two.

Table 2. j_s-dependence of C_x-coefficients in mixed and Navier-Stokes calculations on distorted grids.

j_s	C_d	C_l	C_m	Remark
16	0.018969	0.767161	−0.101493	Navier-Stokes
16	0.016984	0.705367	−0.090708	Euler-Navier-Stokes
32	0.015339	0.843091	−0.111086	Navier-Stokes
32	0.015339	0.843091	−0.111086	Euler-Navier-Stokes

distortion is observed. This is completely different at $j_s = 16$ in which case both the magnitude of C_p as well as the position of the shock is predicted with large error. This indicates that the vertex based treatment of the viscous fluxes is more sensitive to grid distortion than the treatment of the convective fluxes. The convergence behavior for these cases is, roughly speaking, unaffected by these changes, and moreover, agrees quite well with the 'smooth-grid' results presented above. In conclusion, if the grid is distorted considerably and close to the boundary layer the simulation results do not correctly represent the physical solution. Distortions of this type in the 'far field' region were shown to give negligible effects on the results. This information is further clarified in Table 2 in which the C_x-coefficients are summarized.

Effects of time-lag

In order to further show the 'robustness' of the multi-block solver we present the effects of a 'time-lag' in the advancement of the individual blocks. In particular, we allow n time steps to be performed per block, independently, keeping the dummies frozen to the values assigned when the block was realized. We concentrate on cases in which the Navier-Stokes equations are defined in each block, as well as on mixed calculations.

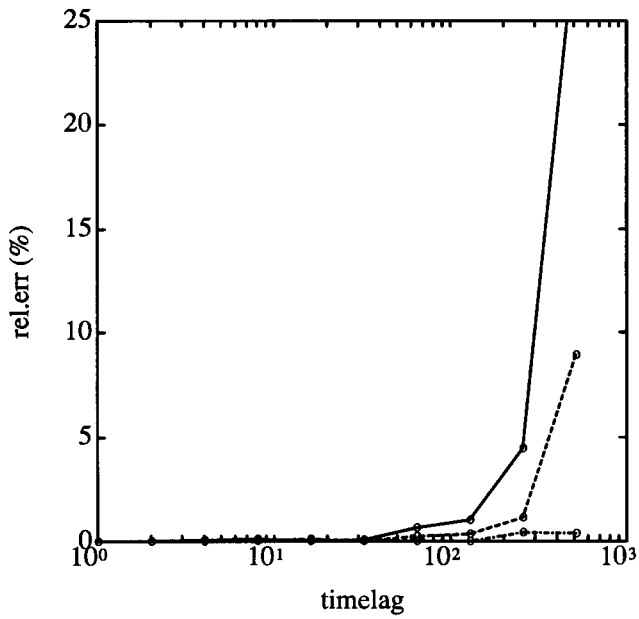

Fig. 7. The relative error in the C_x-coefficients as a function of time-lag after 2000, 2000 and 4000 steps on the coarsest, middle and finest grid, respectively. The results for C_d, C_l and C_m are shown as full, dashed and chain-dotted curves respectively. The Navier-Stokes equations were adopted in each block and $j_s = 24$.

The simulations indicate that a time-lag of up to ≈ 64 is allowed without significant reduction of the convergence on the finest grids. In Fig. 7 we plotted the relative error in the C_x-coefficients as a function of time-lag at a stage at which the solution with time-lag = 1 was converged. We observe that the accuracy of the predictions is almost unchanged after the same number of time steps if the time-lag is smaller than ≈ 64. A sudden decrease in accuracy is present at higher values of the time-lag. It was further noticed that the convergence behavior on the coarsest grid differed considerably as a function of time-lag. Moreover, for time-lag around 16 the grid-sequencing procedure is not effective on the coarsest grid. At these values of the time-lag the convergence on the coarsest grid is virtually absent and the iteration process can best be started at the next finer grid level. At values of the time-lag much smaller or much larger than 16 this behavior was not observed and the grid-sequencing does provide an improved starting solution for the iteration on the finest grid. A similar observation can be made for the mixed calculations.

5. Concluding remarks

We presented a description of the construction of a block structured flow solver and considered the performance of this solver on a number of 2D compressible turbulent flow cases around an airfoil. Variations in the number of blocks as well as the dimensions of the blocks gave no significant effects in the behavior of the solver nor in the numerical solution even if the block interfaces are defined at or near the position of the shock. A combination of Navier-Stokes blocks and Euler blocks does show a significant effect in case the Euler blocks are defined well within the boundary layer around the airfoil. Most surprising is the

sharp transition between totally unacceptable and physically relevant solutions if the Euler blocks are defined in the neighbourhood of the boundary layer. Quite extensive grid distortions do not influence the simulation results, provided these are concentrated outside the boundary layer. A significant time lag can be introduced in the advancement of the individual blocks without causing unphysical solutions. The time-advancement in combination with a block structure on the grid is robust and hence a multigrid acceleration technique in combination with a multiblock solver seems promising. This is subject of further research. We also investigated the behavior of the convergence process and the solution in laminar simulations on a NACA0012 grid. Comparable results were obtained in these cases, i.e. mixed calculations, on distorted grids, with the inclusion of time-lag can be performed without yielding unrealistic solutions, provided the Euler blocks and the grid distortions are outside the boundary layer, and the time-lag is not too large. Thus, multiblock simulations of both laminar and turbulent flow are possible and yield the same results as in the monoblock case, even in mixed simulations on a highly distorted grid, provided both the Euler blocks as well as the grid distortions are situated outside the boundary layer.

Acknowledgements

The authors are greatly indebted to Frans Brandsma for several useful and stimulating discussions. Part of this research was sponsored by the Stichting Nationale Computerfaciliteiten (National Computing Facilities Foundation, N.C.F.) for the use of supercomputer facilities, with financial support from the Nederlandse Organisatie voor Wetenschappelijk Onderzoek (Netherlands Organisation for Scientific Research, N.W.O.)

References

1. F.J. Brandsma, M.E.S. Vogels, J. van der Vooren, D. Dijkstra and J.G.M. Kuerten, Pre-design document of the ISNaS compressible flow simulator. *ISNaS* 88.04.027 (1988).
2. M.E.S. Vogels, Technical design of the ISNaS compressible flow simulator. *ISNaS* 90.02.062 (1990).
3. B. Baldwin and H. Lomax, Thin layer approximation and algebraic model for separated turbulent flow. *AIAA*-78-257 (1978)
4. J.G.M. Kuerten, Numerical definition document for the explicit flow solver. *ISNaS* 88.10.031 (1988).
5. V.N. Vatsa and B.W. Wedan, Development of an efficient multigrid code for 3D Navier-Stokes equations. *AIAA*-98-1791 (1989).
6. A. Jameson, Numerical solution of the Euler equations for compressible inviscid fluids. Report *MAE* 1643 (1985).
7. D.A. Caughey and E. Turkel, "Effects of numerical dissipation on finite-volume solutions of compressible flow problems. AIAA-88-06212 (1988).
8. C. Rossow, Comparison of cell centred and cell vertex finite volume schemes, Proc. of 7th GAMM conference on Numerical Methods in Fluid Mechanics, Louvain-La-Neuve (1987).
9. J.G.M. Kuerten, B.J. Geurts, J.W. van der Burg, A.W. Vreman and P.J. Zandbergen, Development and applications of a 3-D compressible Navier-Stokes solver. M. Napolitano and F. Saletta (Eds.) pp. 529–533 (1993) Springer Verlag.
10. L. Martinelli, Calculations of viscous flow with a multigrid method. Ph.D. Thesis, Princeton University (1987).
11. R. Radespiel, A cell-vertex multigrid method for the Navier-Stokes equations. NASA Technical Memorandum 101557 (1989).