AIAA JOURNAL
Vol. 38, No. 8, August 2000

Local Block Relaxation Method for the Solution of Equations of Gasdynamics

Carlo de Nicola* and Renato Tognaccini†
University of Naples "Federico II," 80125 Naples, Italy
and
Vittorio Puoti‡
Centro Italiano Ricerche Aerospaziali, 81043 Capua (CE), Italy

A new explicit pseudo-time-integration algorithm, termed local block relaxation, for the solution of the equations of gasdynamics in block-structured domains is proposed. The aim of the method is to reduce the computational costs in obtaining the numerical solution of complex flowfields. Attention is given to the definition of proper interface boundary conditions that ensure stability even if the smoothing of the solution is not synchronized in time among the blocks. Once stability is guaranteed, oversmoothings can be performed in a restricted number of blocks, namely, those that exhibit a slower rate of convergence. The method proposed is closely coupled with a standard multigrid cycle. A theoretical analysis of the savings in computational work is presented so that the convergence speed up can be estimated a priori. The strategy is shown to be effective in both two-dimensional and three-dimensional Navier–Stokes simulations. The present algorithm can be extended to a broad variety of numerical schemes based on explicit time integrations.

Introduction

COMPUTATIONAL fluid dynamics applications are still limited by the large computational time required for the solution of complex three-dimensional flowfields. In spite of the development of faster and cheaper computers, the introduction of more sophisticated and complex flow models, for instance, higher-order turbulence closures, increases the demand on computational resources.

The numerical solution of complex flowfields on both parallel and sequential architectures is widely obtained by using multiblock structured methods (see Ref. 1 for an extensive literature survey). The success of this technique is due to the considerably simplified grid generation task and to the intrinsic parallel nature of the multiblock approach that enables a satisfactory parallel speed up and efficiency. The algorithm essentially consists in subdividing the flow domain into blocks in which the solution is independently computed and coupled among adjacent subdomains by appropriate internal boundary conditions (IBCs).

A new integration algorithm for multiblock structured domains, local block relaxation (LBR), is presented. This method increases the efficiency of classical explicit multistage time integration and can be easily adapted to other schemes. The basic idea is to perform time relaxations only in blocks with larger residuals, to reduce the global computational work while simultaneously converging to the steady solution.

Sawley and Tegner[2] exploited this idea for the simulation of a supersonic flowfield: in that case, due to the hyperbolic spatial nature of the problem, the initial flow conditions already represented the solution of the governing equations in many blocks, and both the implementation of the method and its effectiveness were straightforward and simply demonstrated.

When the equations of gasdynamics are solved to simulate more complex flows from the mathematical viewpoint such as in the transonic regime, the crucial point to overcome is given by the potential growth of unstable modes introduced by the IBCs at block interfaces. Whereas Ciment[3] studied the interface stability of a one-step explicit formula, de Nicola et al. provided examples of instabilities in the case of explicit multistage integration and verified them using the spectral analysis of the transitional operator[4] and the normal mode analysis.[5] In particular, they noted that stable multiblock computations by explicit Runge–Kutta (R–K) algorithms required further constraints on the Courant–Friedrichs–Lewy (CFL) condition. Some of these problems are also common in implicit schemes. Jenssen[6] reported strong convergence decays of an implicit upwind block structured method, whereas Lerat and Wu[7] performed a normal mode analysis of the interface matching problem of their central implicit scheme.

It is clear that these interface instabilities strongly limit the application of an LBR strategy. A general and robust method requires the definition of appropriate stable IBCs because the LBR eliminates the synchronization of the time-integration algorithm among blocks. In fact, providing the stability of the IBCs is guaranteed, each block can be independently relaxed by any number of time steps without the need for an exchange of information with the adjacent domain. In Ref. 8 a first LBR procedure was proposed. However, although overcoming stability constraints at block interfaces, the performance of the algorithm was not completely satisfactory. In the startup phase, when the residual was large in regions close to the geometry and small far from it, the method was very effective. Subsequently, the residual reached the same value almost everywhere; all of the blocks then needed relaxing to damp high-frequency oscillations, and consequently, the LBR was unsuccessful because the obtained computational speed up was not significant. A more effective approach, proposed here, is to insert an LBR in a multigrid V-cycle to retain the smoothing properties of the multigrid algorithm while reducing the computational cost of a single cycle.

In the present paper, after a brief description of the multiblock structured approach, second-order stable IBCs, based on a MUSCL scheme, are presented; subsequently, the LBR method is described. Finally, the results of a number of Navier–Stokes calculations are presented and the improvements in convergence speed are discussed.

Multiblock Structured Algorithm

Although block structured algorithms are widely used today, it is useful to recall the basic concepts, which may render a clearer understanding of the present method. We have to solve a mixed initial boundary value problem of the type

Received 16 February 1999; revision received 9 December 1999; accepted for publication 3 January 2000. Copyright © 2000 by the authors. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission.

*Associate Professor, Dipartimento di Progettazione Aeronautica, Piazzale Tecchio 80; denicola@unina.it.

†Staff Research Scientist, Dipartimento di Progettazione Aeronautica, Piazzale Tecchio 80; rtogna@unina.it.

‡Researcher, Dipartimento di Calcolo Numerico e Parallelo; v.puoti@cira.it.

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{A}\boldsymbol{u} = 0$$

$$\text{IC} \quad \boldsymbol{u}(\boldsymbol{r}, 0) = \boldsymbol{u}^0(\boldsymbol{r}), \qquad \forall \boldsymbol{r} \in \Omega$$

$$\text{BC} \quad \boldsymbol{u}(\boldsymbol{r}, t) = g(\boldsymbol{r}), \qquad \forall \boldsymbol{r} \in \partial\Omega \tag{1}$$

where $t \in [0, +\infty]$ is the independent time variable; $\boldsymbol{r} = [x, y, z]^T \in \Omega \subseteq X^3$ is the vector of space Cartesian coordinates defining the domain Ω, with boundary $\partial\Omega$; and IC and BC, respectively, stand for initial condition and boundary condition. $\boldsymbol{A}$ is a differential spatial operator, $\boldsymbol{u}$ is the vector of dependent flow variables, $\boldsymbol{u}^0$ is a given initial flow state, and g is a specified operator defining the values of $\boldsymbol{u}$ on $\partial\Omega$. Here we are interested in the steady-state solution computed by a pseudotransient method, that is, we do not require time consistency.

After the topological decomposition of the domain Ω into structured blocks, the initial problem is subdivided into a number of mixed initial boundary value problems to be independently solved:

$$\frac{\partial \boldsymbol{u}_b}{\partial t} + \boldsymbol{A}\boldsymbol{u}_b = 0$$

$$\text{IC} \quad \boldsymbol{u}_b(\boldsymbol{r}, 0) = \boldsymbol{u}_b^0(\boldsymbol{r}), \qquad \forall \boldsymbol{r} \in \Omega_b$$

$$\text{BC} \quad \boldsymbol{u}_b(\boldsymbol{r}, t) = g_b(\boldsymbol{r}), \qquad \forall \boldsymbol{r} \in \partial\Omega_b$$

$$\Omega_1 \cup \Omega_2 \cup \cdots \cup \Omega_b \cup \cdots \cup \Omega_{nb} = \Omega \tag{2}$$

where the subscript b denotes one of the nb subdomains making up Ω. In this way proper IBCs need to be defined on the artificial block interfaces introduced by the domain decomposition. The IBCs can locally modify the numerical integration algorithm.

It is not aim of this paper to provide a detailed description both of the inner block method, applied to solve the equations of gasdynamics, nor of the algorithm implemented to define the IBCs. If interested, the reader should refer to Refs. 9 and 10, respectively. In Ref. 11, a more sophisticated algorithm for global discontinuous grids across blocks is described. The inner block scheme is the widely used central discretization with explicit multistage time-stepping integration incorporating several convergence accelerators, such as multigrid and residual averaging. Different turbulence models, such as Baldwin–Lomax and κ–ϵ can also be used. This method is extensively applied in multidomain calculations both in research and industrial environments.[12] In the case of Navier–Stokes equations, the differential operator $\boldsymbol{A}$ can be expressed as $\boldsymbol{A} = \nabla \cdot \boldsymbol{F}_t$, where $\boldsymbol{F}_t = \boldsymbol{F} - \boldsymbol{F}_v$ with $\boldsymbol{F}$ and $\boldsymbol{F}_v$, respectively, the inviscid and viscous part of the flux vector $\boldsymbol{F}_t$ (Ref. 13).

The IBCs algorithm, in the case of continuous grids across blocks, simply consists in computing the numerical fluxes at interfaces by using the same difference formulas used in the inner block scheme. From the computer scientist standpoint, this can be done by adding, at each interface of each block, a number of extra grid cell layers in which the required adjacent flow states are copied. If the extra block data are frozen at the values assumed at the initial stage of the multistage time stepping, the time integration is modified with respect to a single block computation. Frozen IBCs, in the following referred to as standard IBCs, are preferable when discontinuous grids across block interfaces are used because time-consuming interpolation formulas are required to compute the extra grid cell data to ensure the local accuracy as well as being required in parallel applications to minimize the interprocessor communications. Unfortunately, as stated in the Introduction, the main drawback is the reduction of the stability range of the time-integration scheme with further severe restrictions on the allowable Courant number (see Ref. 14 for practical examples) due to the creation of numerical perturbations at the block interfaces that do not allow the introduction of an LBR strategy.

Stable IBC

To implement an LBR method, numerical IBCs, preserving the stability range of the inner block scheme, have to be introduced.

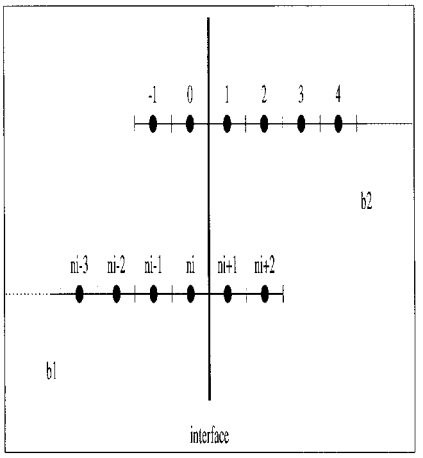

Fig. 1 One-dimensional example: interface separating blocks b_1 and b_2.

They will be referred as stable IBCs to enhance the difference with the standard IBCs.

To design stable IBCs it is necessary to formulate a mixed initial boundary value problem in each block that is stable, in the sense described by Gustafsson et al.,[15] during the time intervals in which extra grid cell states are frozen. Hyperbolic problems require that fluxes at boundaries (including block interfaces) are computed according to characteristic theory. This property is clearly not satisfied if the interface flux is computed by a central discretization that is symmetrical with respect to the block interface.

The proposed solution to this problem consists in computing the inviscid part of the numerical flux at block interfaces by using an approximate Riemann solver that strictly respects the boundary condition stability constraints.

First- and second-order numerical schemes for flux evaluation have been tested. The first-order IBCs have been implemented by adopting the flux difference splitting by Pandolfi and Borrelli.[16] Referring to Fig. 1, the interface inviscid flux $\boldsymbol{F}^{(m)}_{1/2,b2}$ of block b_2 at the mth stage of the R–K integration with M steps is given by

$$\boldsymbol{u}^{(0)}_{0,b2} = \boldsymbol{u}^{(0)}_{ni,b1}$$

$$\boldsymbol{F}^{(m)}_{\frac{1}{2},b2} = \boldsymbol{F}^{(m)}_{\frac{1}{2},b2}\left(\boldsymbol{u}^{(0)}_{0,b2}, \boldsymbol{u}^{(m)}_{1,b2}\right)$$

$$= \frac{1}{2}\left[f\left(\boldsymbol{u}^{(0)}_{0,b2}\right) + f\left(\boldsymbol{u}^{(m)}_{1,b2}\right) - \int_{\boldsymbol{u}^{(0)}_{0,b2}}^{\boldsymbol{u}^{(m)}_{1,b2}} \left|\frac{\partial f}{\partial \boldsymbol{u}}\right| \mathrm{d}\boldsymbol{u}\right], \qquad \forall m = 1, M \tag{3}$$

where $\boldsymbol{f}$ is the physical flux. The integral on the right-hand side is computed by using the characteristic variables along an appropriate path (Ref. 16). The interface flux $\boldsymbol{F}^{(m)}_{1/2,b1}$ of block b_1 is computed using a similar formula.

The second-order scheme, proposed here, has been derived using a MUSCL approach. The left A and right B states of the Riemann problem are obtained by extrapolating linearly to the interface the inner block flow:

$$\boldsymbol{u}_A = \frac{3}{2}\boldsymbol{u}^{(0)}_{ni,b1} - \frac{1}{2}\boldsymbol{u}^{(0)}_{ni-1,b1}, \qquad \boldsymbol{u}_B = \frac{3}{2}\boldsymbol{u}^{(m)}_{1,b2} - \frac{1}{2}\boldsymbol{u}^{(m)}_{2,b2}$$

$$\boldsymbol{F}^{(m)}_{\frac{1}{2},b2} = \boldsymbol{F}^{(m)}_{\frac{1}{2},b2}(\boldsymbol{u}_A, \boldsymbol{u}_B) = \frac{1}{2}\left[f(\boldsymbol{u}_A) + f(\boldsymbol{u}_B) - \int_{\boldsymbol{u}_A}^{\boldsymbol{u}_B} \left|\frac{\partial f}{\partial \boldsymbol{u}}\right| \mathrm{d}\boldsymbol{u}\right]$$

$$\forall m = 1, M \tag{4}$$

This scheme is very robust and simple. Note the absence of a limiting procedure in the flux evaluation; up to now, it has not been necessary to introduce it to preserve accuracy even if the artificial boundary is crossed by a shock.

The stable IBCs (4) enable the stability limits of the single block inner scheme to be retrieved, thereby eliminating the reduction of

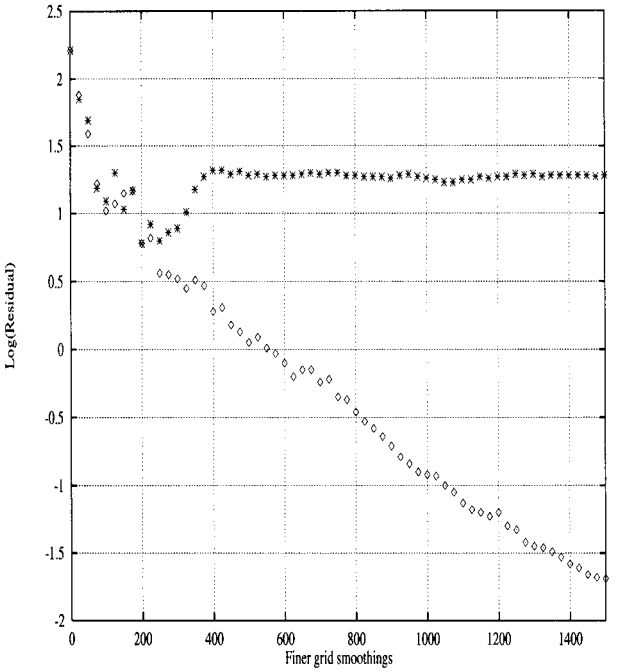

Fig. 2 Three-dimensional test: three R–K stages, CFL = 1.5, three multigrid levels, V-cycle, comparison of histories of convergence obtained with standard IBCs (∗) and stable IBCs (◇).

the stability range due to the block decomposition. Figure 2 shows the convergence histories of two numerical predictions of the viscous flow past a wing using a multiblock grid: the logarithm of the residual, that is, the L_2 norm of the time derivative in the continuity equation, is plotted against finer grid smoothings. The first result (asterisk in Fig. 2) has been obtained by adopting the standard IBCs and the maximum allowable Courant number for the given explicit multistage scheme. After 200 finer grid smoothings, the residual starts growing and reaches a plateau value, which is a typical behavior associated to the application of unstable boundary conditions. If the standard IBCs are replaced by the stable IBCs, as specified by Eq. (4), the residual (◇ symbol in Fig. 2) shows a convergent solution. (After 1500 finer grid smoothings the residual is dropped by 4 orders of magnitude.) Details about the geometry and flow conditions of the test will be given later in the Results section.

LBR Method

The stable IBCs enable multiblock computations to be performed without any further stability constraint. Once stability is guaranteed, the next step consists in an asynchronous computation of the solution in each block in the sense that the blocks where the solution has to be smoothed can be arbitrarily chosen. This is the basic idea of the LBR method.

The required user input consists of the number of blocks to be oversmoothed nbp and ν, the number of multigrid pre/postsmoothings. In the present algorithm the classical single multigrid V-cycle is performed as follows (the case of a two-level cycle is considered for sake of simplicity): 1) one presmoothing in all blocks, 2) computing the L_2 norm of the residual in each block, 3) $\nu - 1$ presmoothings in the nbp blocks with larger residual, 4) restricting the solution to the next coarser level, 5) ν smoothings in all blocks at this level, 6) prolongating the solution to the finest level, 7) one postsmoothing in all blocks at this level, and 8) $\nu - 1$ postsmoothings in the nbp blocks with the larger residual at this level.

Because the main idea governing the method consists in smoothing the solution in a restricted number of blocks of the domain, that is, those that exhibit larger values of the local residual, it is necessary to introduce a new parameter measuring the computational work of the multigrid cycle. The computational cost of a single smoothing on the finer grid level depends on the ratio of the actually smoothed grid cells to the number of cells in the whole domain:

$$\mathrm{WU} = \frac{\sum_{b=1}^{\mathrm{nbp}} NC_b}{\sum_{b=1}^{\mathrm{nb}} NC_b} \tag{5}$$

where WU is work unit and NC_b is the number of grid cells of block b. Equation (5) applies when the smoothing of the solution in each grid cell requires the same number of floating point operations. Should this requirement not be met, that is, different sets of equations are solved in different blocks, weighting coefficients have to be introduced and Eq. (5) changes slightly to

$$\mathrm{WU} = \frac{\sum_{b=1}^{\mathrm{nbp}} wc_b \cdot NC_b}{\sum_{b=1}^{\mathrm{nb}} wc_b \cdot NC_b}. \tag{6}$$

The wc_b values have been obtained from numerical experiments. They are as follows.

For Euler flow:

$$wc_b = 1.00 \tag{7a}$$

For thin-layer laminar flow:

$$wc_b = 1.25 \tag{7b}$$

For thin-layer turbulent flow with algebraic model:

$$wc_b = 1.55 \tag{7c}$$

For complete Navier–Stokes laminar flow:

$$wc_b = 1.63 \tag{7d}$$

For complete Navier–Stokes turbulent flow with algebraic model:

$$wc_b = 1.93 \tag{7e}$$

For complete Navier–Stokes turbulent flow with two-equation model:

$$wc_b = 3.79 \tag{7f}$$

If the LBR is applied only on the finer grid level, the computational cost of a complete multigrid cycle is

$$\mathrm{WU}_c = \frac{\nu \cdot \sum_{i=1}^{\mathrm{ngl}} 2^{d(1-i)} + (1/WU - 1) \cdot \left(1 + \nu \cdot \sum_{i=2}^{\mathrm{ngl}} 2^{d(1-i)}\right)}{(1/\mathrm{WU}) \cdot \nu \cdot \sum_{i=1}^{\mathrm{ngl}} 2^{d(1-i)}} \tag{8}$$

where ngl is the number of multigrid levels and d is the number of dimensions of the geometry. If the LBR is not applied, Eq. (8) returns $\mathrm{WU}_c = 1$. After c multigrid cycles the total computational work is

$$W_c = W_{c-1} + \mathrm{WU}_c, \qquad W_0 = 0 \tag{9}$$

The work saving (WS) obtained by applying the LBR is given by the ratio of the work performed without LBR to the work performed using LBR ($\mathrm{WS} = W_{c\,\mathrm{NOLBR}} / W_{c\,\mathrm{LBR}}$). It is possible to evaluate a priori the WS obtainable by applying the proposed LBR method. The estimation of WS is performed using the following simplifying hypothesis: it is assumed that there is no decay of convergence speed of the multigrid algorithm (to be verified a posteriori), and that the blocks chosen for oversmoothings do not change during convergence (which is not true in the practical algorithm). In this case the WS expression reduces to

$$\mathrm{WS} = 1/\mathrm{WU}_c \tag{10}$$

Figure 3 shows WS against 1/WU for different values of ν and d, given ngl. Satisfactory values of WS require large values of 1/WU, the latter depending on the input value of nbp, the domain topology, the grid point distribution, and on the flow models used in the blocks. Because WS tends to an asymptotic limit when $1/\mathrm{WU} \to \infty$, it is useless to stress the LBR to very high values of 1/WU:

$$\lim_{1/\mathrm{WU} \to \infty} \mathrm{WS} = \frac{\nu \cdot \sum_{i=1}^{\mathrm{ngl}} 2^{d(1-i)}}{\left(1 + \nu \cdot \sum_{i=2}^{\mathrm{ngl}} 2^{d(1-i)}\right)} \tag{11}$$

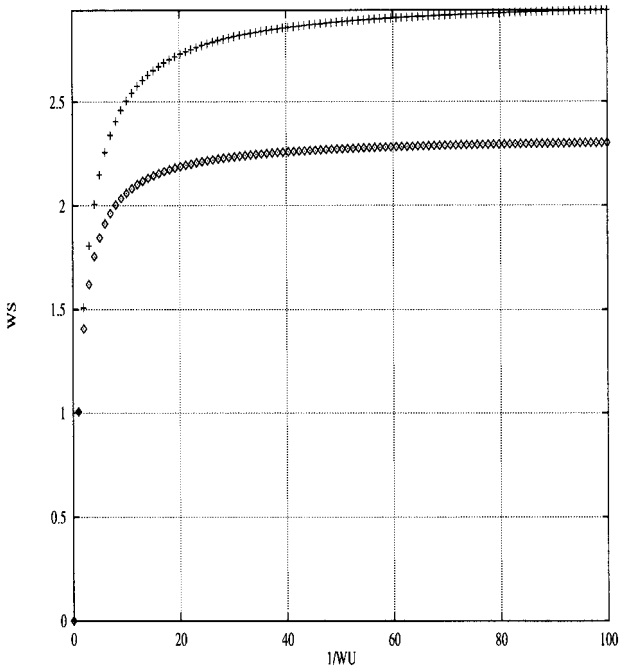

Fig. 3 LBR method on the finest grid, predicted WS vs 1/WU, ngl = 3: ◇, $\nu = 4$ and $d = 2$; and +, $\nu = 8$ and $d = 3$.

WS is upper bounded also for $\nu \to \infty$:

$$\lim_{\nu \to \infty} \mathrm{WS} = \frac{1/\mathrm{WU} \cdot \sum_{i=1}^{\mathrm{ngl}} 2^{d(1-i)}}{\sum_{i=1}^{\mathrm{ngl}} 2^{d(1-i)} + (1/\mathrm{WU} - 1) \cdot \sum_{i=2}^{\mathrm{ngl}} 2^{d(1-i)}} \tag{12}$$

Numerical experiments indicated that the extension of the LBR to the coarser level of the multigrid cycle was not effective. In particular, while the scheme was still converging, the smoothing factor of the multigrid cycle was not retained.

Results

In this section results and remarks concerning the simulation of transonic, viscous flows past two-dimensional and three-dimensional geometries are presented.

The initial flowfield was set to freestream conditions for all of the tests. They were performed with the same numerical input: 1) three multigrid levels, 2) three-stage R–K scheme with local time stepping, and 3) $k^{(2)} = 1$, $k^{(4)} = 0.04$, and CFL $= 1.5$, where $k^{(2)}$ and $k^{(4)}$ are, respectively, the second- and fourth-order artificial dissipation coefficients. The coefficients represent the upper stability limit of the R–K scheme used. When LBR is activated, the parameters nbp $= 4$, and nbp $= 1$, with $\nu = 4$ were set for the two-dimensional case, whereas nbp $= 6$ and 1 with $\nu = 8$ were used for the three-dimensional case.

The two-dimensional viscous flow was computed past the RAE2822 airfoil by solving the thin-layer Navier–Stokes equations. The flow conditions were $M_\infty = 0.730$, $\alpha = 2.780$ deg, and $Re = 6.5 \times 10^6$. A C-type topology was used consisting of 20 blocks grouped in three layers in which different sets of equations were solved. In the first layer, close to the airfoil and to the wake, the thin-layer Navier–Stokes equations were solved utilizing the Baldwin–Lomax turbulence model, in the second layer the laminar flow model was used, whereas in the outer layer the inviscid flow model was applied (see Fig. 4). The grid consisted of 65,000 grid cells.

Given the domain topology, the grid point distribution, and the input parameters, that is, ngl and ν, the WS obtainable with the LBR can be predicted a priori by using either Eq. (10) or Fig. 3. To compute WS in Eq. (10), a guess is needed for 1/WU. It has been computed using Eq. (6) assuming that the nbp blocks are those close to the airfoil. The predicted values of WS are

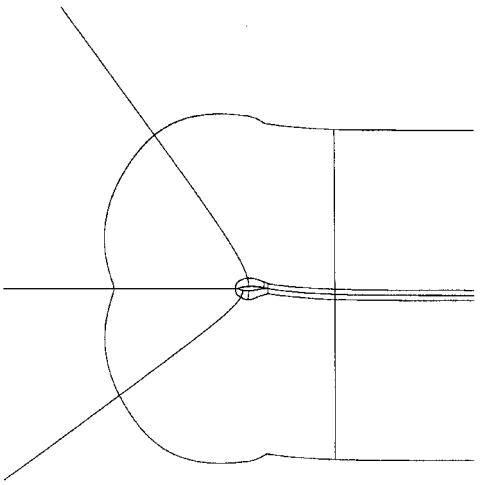

Fig. 4 Two-dimensional test: domain topology.

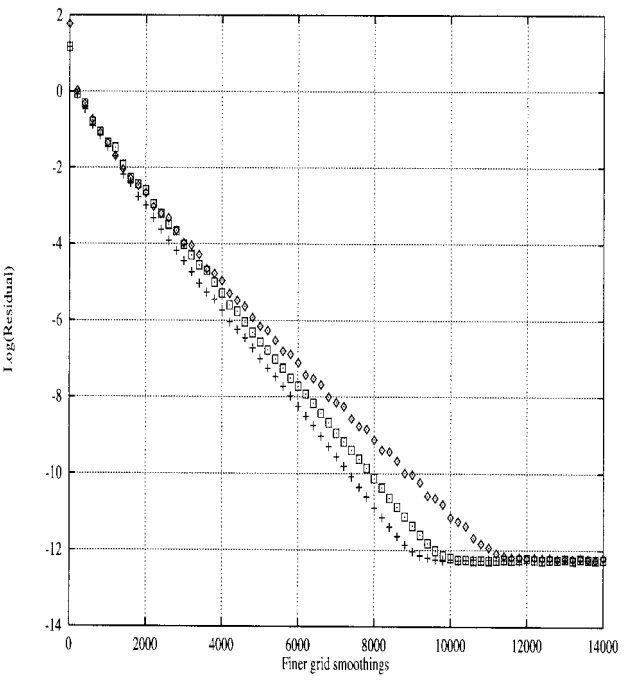

Fig. 5 Two-dimensional test; stable IBCs, comparison of convergence histories against finer grid smoothings obtained with LBR: +, nbp = 4 and $\nu = 4$; □, nbp = 1 and $\nu = 4$; and ◇, without LBR.

$$\mathrm{nbp} = 4 \to 1/\mathrm{WU} = 3.70 \to \mathrm{WS} = 1.71$$

$$\mathrm{nbp} = 1 \to 1/\mathrm{WU} = 18.45 \to \mathrm{WS} = 2.17 \tag{13}$$

The convergence histories of the numerical simulations, obtained with and without the LBR, are shown in Fig. 5 in terms of finer grid smoothings. In this diagram the speed up of the LBR cannot be shown because the total number of finer grid smoothings is equal for all cases, but it demonstrates a fairly similar behavior in the drop of the residual to machine zero; hence the first hypothesis made in the preceding section related to Eq. (10) is confirmed. In fact the convergence with LBR, in terms of total smoothings, was actually slightly faster.

The gain, obtained by utilizing the proposed LBR method, can be measured by plotting the same results for residual against work done as defined by Eq. (9) and shown in Fig. 6. The values of WS computed, for instance, when the residual has dropped by six orders of magnitude, that is, when log(res) $= -4$, are

$$\mathrm{nbp} = 4 \to \mathrm{WS}_+ = 1.96, \qquad \mathrm{nbp} = 1 \to \mathrm{WS}_\square = 2.25 \tag{14}$$

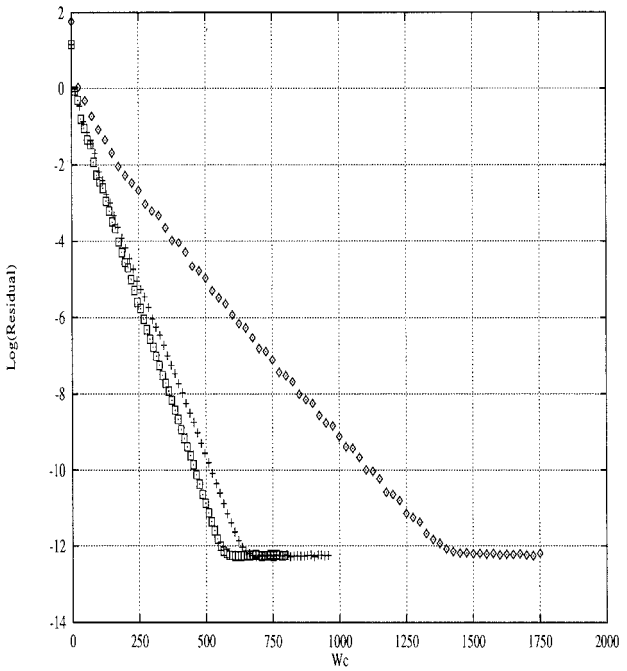

Fig. 6 Two-dimensional test; stable IBCs, comparison of convergence histories against W_c obtained with LBR: +, nbp = 4 and $\nu = 4$; □, nbp = 1 and $\nu = 4$; and ◇, without LBR.

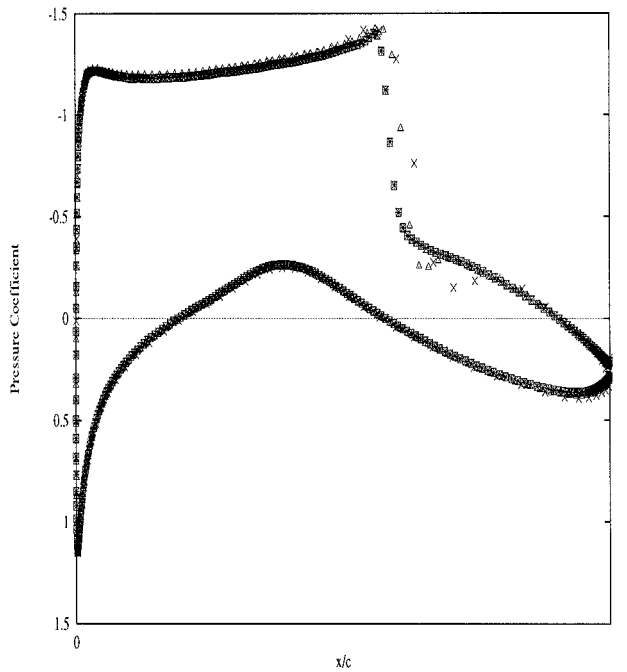

Fig. 7 Two-dimensional test; pressure coefficient obtained with standard IBCs on different size meshes: ×, coarse; △, medium; ∗, fine; and □, with LBR, nbp = 1 and $\nu = 4$.

Note that the measured values of WS were reasonably similar to those predicted using Eq. (10). The small differences are mainly because the second assumption for the theoretical estimate of WS was not satisfied: The blocks that were overrelaxed changed during convergence.

The values of WS are an estimate of the CPU time saving (CPUS) defined as the ratio of the CPU time required for the calculation without the LBR to the CPU time required with the LBR. The following values of CPUS were obtained:

$$\text{nbp} = 4 \rightarrow \text{CPUS}_{+} = 2.05, \qquad \text{nbp} = 1 \rightarrow \text{CPUS}_{\square} = 2.21 \tag{15}$$

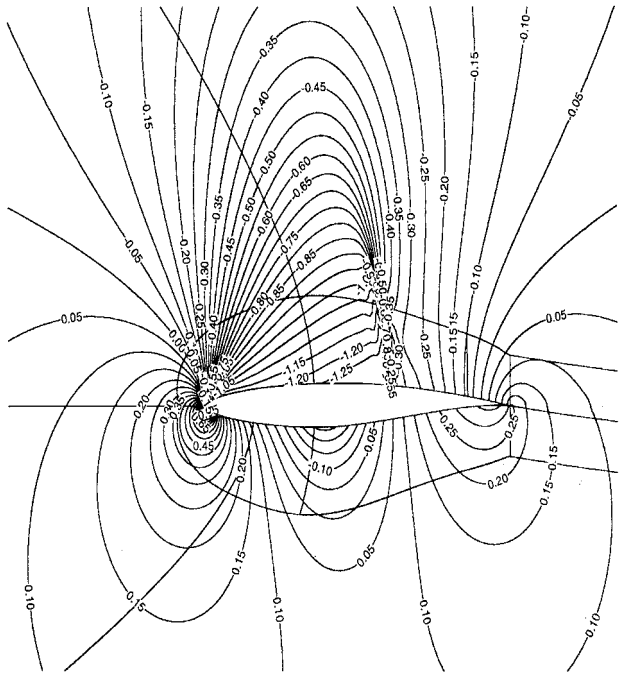

Fig. 8 Two-dimensional test, stable IBCs, pressure coefficient distribution obtained with LBR, nbp = 4 and $\nu = 4$.

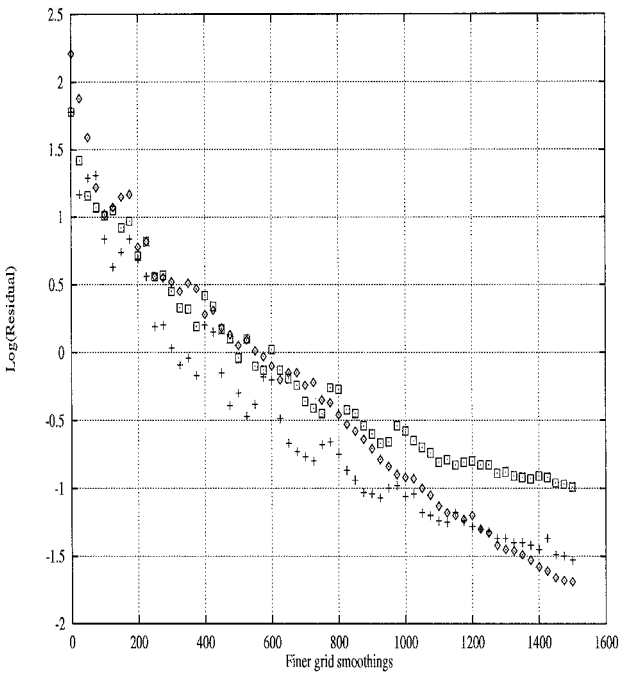

Fig. 9 Three-dimensional test; stable IBCs, comparison of convergence histories against finer grid smoothings obtained with LBR: +, nbp = 6 and $\nu = 8$; □, nbp = 1 and $\nu = 8$; and ◇, without LBR.

The discrepancies between Eqs. (14) and (15) are because the values of wc_b used do not take into account that the turbulent eddy viscosity is not recomputed on the coarser grid levels.

As far as the solution is concerned, it is important to verify its accuracy, the numerical flux evaluation being locally modified by the upwind biased IBCs. The lift and drag coefficients are reported in Table 1 varying the IBCs and the LBR parameters. The results show that lift and drag do not change in practice. The pressure coefficient distribution obtained by the LBR method matches that obtained via the standard IBCs, as shown in Fig. 7. This test is very interesting for the purpose of verifying how the modifications introduced in

Table 1 Two-dimensional test: aerodynamic coefficients

Coefficient	Standard IBCs	Stable IBCs	LBR (nbp = 4)	LBR (nbp = 1)
c_l	0.889	0.889	0.889	0.889
c_d	0.0170	0.0169	0.0169	0.0169

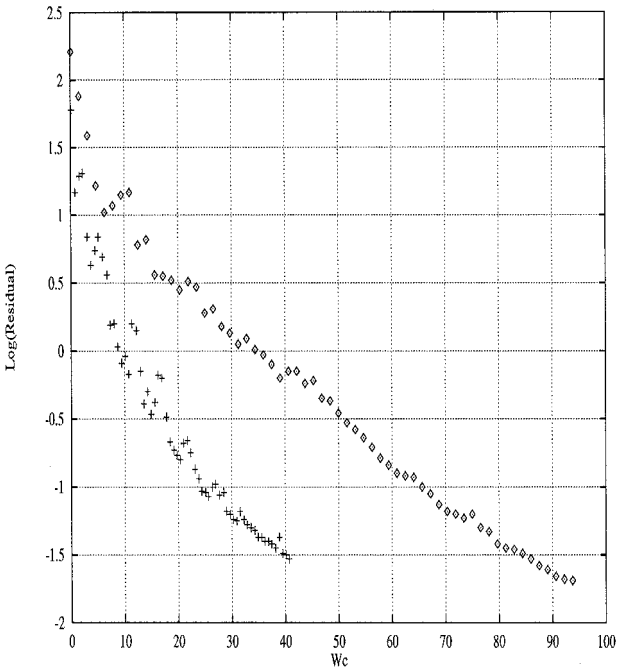

Fig. 10 Three-dimensional test; stable IBCs, comparison of convergence histories against W_c obtained with LBR: +, nbp = 6 and $\nu = 8$; and ◇, without LBR.

the numerical scheme at block interfaces can affect the solution as a strong normal shock is observed that crosses the block interface (see Fig. 8).

To complete the analysis of the accuracy of present calculations, the pressure distributions obtained for the coarser and medium grids are also presented in Fig. 7. In the regions of smooth flow, the solution is already converged on the coarsest mesh, whereas as could be expected, grid resolution influences the results in the shock region. Similar results that have also been obtained in the three-dimensional test are discussed next.

The three-dimensional viscous flow was computed past the RAE2155 wing solving the thin-layer Navier–Stokes equations, the geometry of which was provided to all of the participants in the EUROVISC Workshop held on 28–29 October 1994. The flow conditions were $M_\infty = 0.744$, $\alpha = 2.500$ deg, and $Re = 4.1 \times 10^6$. A C-type topology was generated around the geometry in the symmetry plane and an H-type topology in the span direction. The thin-layer Navier–Stokes equations together with the Baldwin–Lomax turbulence model were solved in the 6 blocks lying adjacent to the wing solid boundary, and laminar flow conditions were simulated in the remaining 16 blocks spanning the rest of the flow domain. The grid consisted of 1,128,534 points.

The WS predictions for this case are

$$\text{nbp} = 6 \rightarrow 1/\text{WU} = 3.02 \rightarrow \text{WS} = 2.05$$

$$\text{nbp} = 1 \rightarrow 1/\text{WU} = 21.17 \rightarrow \text{WS} = 3.72 \qquad (16)$$

Tests with these values of nbp were performed in a manner similar to those of the two-dimensional simulations. Figure 9 contains the plot of the residual against finer grid smoothings obtained with no LBR and with nbp = 6 and 1. In this test the LBR with nbp = 1 did not perform well because the rate of convergence is slower if compared with nbp = 6 or 22. The plot of the residual in terms of work is shown in Fig. 10. The values of WS and CPUS are

$$\text{nbp} = 6 \rightarrow \text{WS}_+ = 2.11, \qquad \text{nbp} = 6 \rightarrow \text{CPUS}_+ = 2.97 \qquad (17)$$

By also activating LBR on the medium grid, it was possible to increase the value of WS without any decay in the convergence history (Fig. 11). In this case (Fig. 12) we obtained

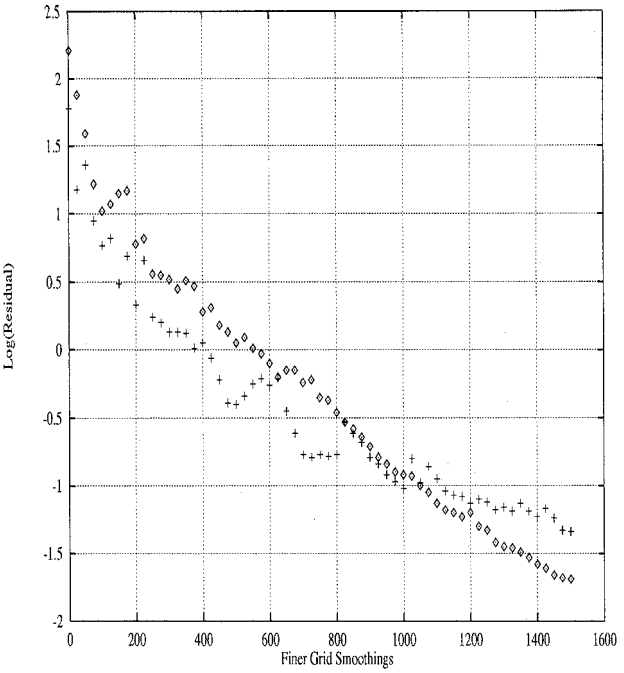

Fig. 11 Three-dimensional test; stable IBCs, LBR on both medium and fine grids, comparison of convergence histories against finer grid smoothings obtained with LBR: +, nbp = 6 and $\nu = 8$; and ◇, without LBR.

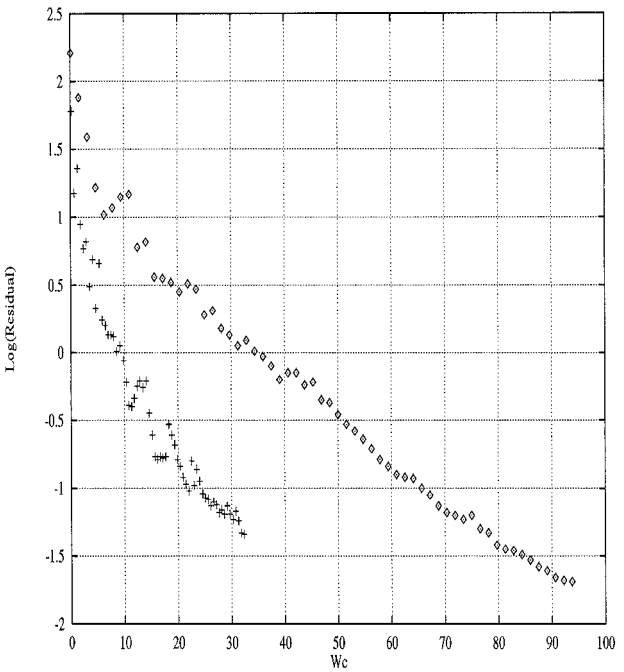

Fig. 12 Three-dimensional test; stable IBCs, LBR on both medium and fine grids, comparison of convergence histories against W_c obtained with LBR: +, nbp = 6 and $\nu = 8$; and ◇, without LBR.

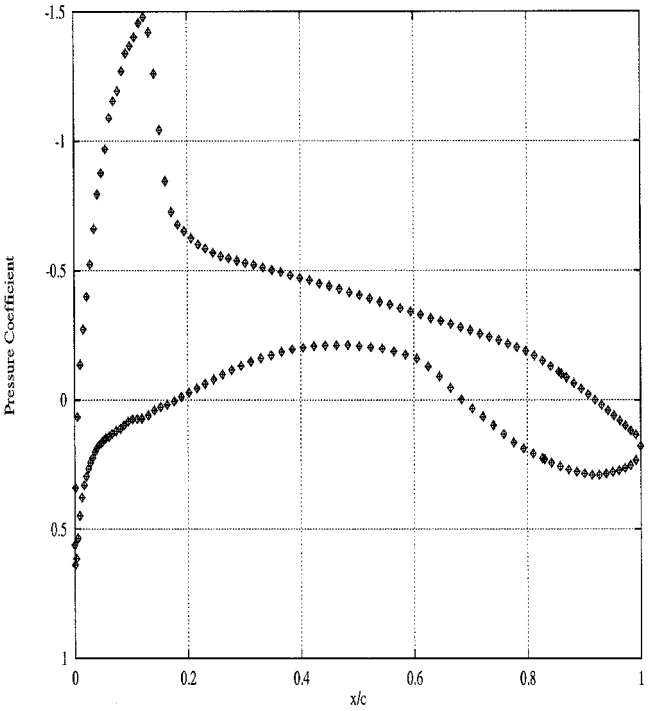

Fig. 13 Three-dimensional test; stable IBCs, $y/(b/2) = 0.20$, pressure coefficient obtained with LBR on the finest grid: +, nbp = 6 and $\nu = 8$; ◇, and without LBR.

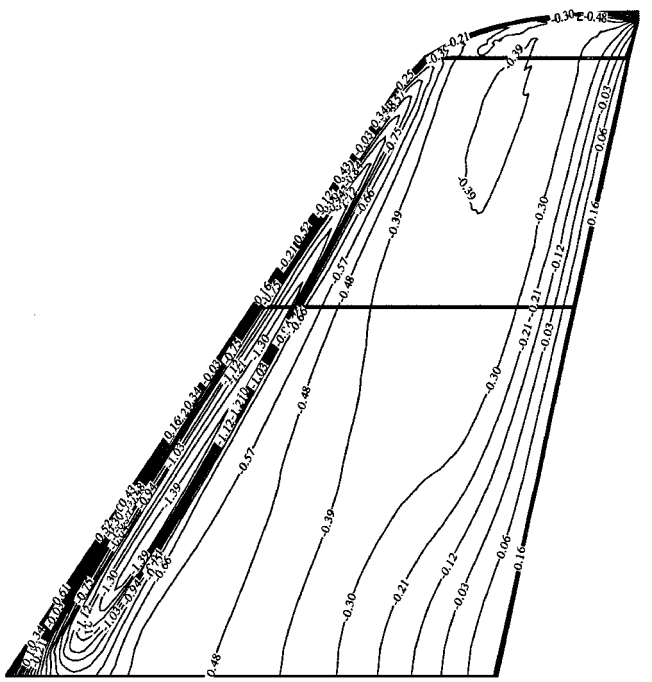

Fig. 14 Three-dimensional test; stable IBCs, pressure coefficient distribution obtained with LBR on the finest grid, nbp = 6 and $\nu = 8$.

$$\text{nbp} = 6 \rightarrow \text{WS}_+ = 2.46, \qquad \text{nbp} = 6 \rightarrow \text{CPUS}_+ = 3.35 \quad (18)$$

Figure 13 shows the pressure coefficient distributions at the spanwise station $(y/b/2) = 0.20$ with and without LBR. The isocurves of the pressure coefficient on the upper surface of the wing are shown in Fig. 14.

The present results show the possibility of obtaining large CPU saving by applying the LBR. However, the practical speedups depend on the grid topology and block decomposition. A proper block decomposition for LBR purposes requires the definition of different block layers from the configuration toward the far field. In fact, the residual is expected to be larger in the blocks that have part of the configuration as a boundary. On the contrary, in the far-field region the initial freestream state practically matches the flow solution. The proposed method requires the input of two numerical parameters: nbp and ν. It is not useful to apply too high values of ν and too low values of nbp because the convergence rate could be affected without substantial improvement in WS. In practice recommended values of ν are 4 (two-dimensional tests) and 8 (three-dimensional tests) and nbp should not exceed the number of blocks close to the configuration.

Conclusions

A new explicit time-integration algorithm for the solution of the equations of gasdynamics in block structured domains has been presented, with the objective to improve the convergence speed.

The method is based on the definition of new second-order conditions imposed on the artificial boundaries introduced in the computational domain by the block decomposition. The boundary conditions are capable of preserving the stability constraints of the inner block scheme even if the flow states at block interfaces are frozen for a large number of smoothings. Therefore, the implementation of LBR, which consists of performing a larger number of smoothings in blocks where the rate of convergence is slower, is straightforward and effective if a standard multigrid algorithm is also applied as described in this paper.

The set of internal boundary conditions developed do not locally affect the accuracy of the solution.

The theoretical analysis of the obtainable speed up shows that the efficiency of the LBR depends on the domain topology and the grid point distribution. In practical applications, CPU time savings greater than a factor of 3 have been obtained.

The strategy and the set of internal boundary conditions developed are loosely coupled with the inner block numerical method and can, therefore, be extended to a wide class of schemes.

Acknowledgments

The authors wish to thank L. Paparone of Centro Italiano Ricerche Aerospaziali (CIRA) for his support through the whole activity and useful discussions and A. D. French of CIRA for his endless patience during the correction of the paper.

References

[1]de Nicola, C., Tognaccini, R., and Visingardi, P., "Multiblock Structured Algorithms in Parallel CFD," *Parallel Computational Fluid Dynamics: Implementations and Results Using Parallel Computers*, North-Holland Elsevier, Amsterdam, 1996, pp. 1–8.

[2]Sawley, M. L., and Tegner, J. K., "A Data Parallel Approach to Multiblock Flow Computations," *International Journal for Numerical Methods in Fluids*, Vol. 19, No. 8, 1994, pp. 707–721.

[3]Ciment, M., "Stable Matching of Difference Scheme," *SIAM Journal on Numerical Analysis*, Vol. 9, No. 4, 1972, pp. 695–701.

[4]de Nicola, C., Pinto, G., and Tognaccini, R., "On the Numerical Stability of Block Structured Algorithms with Applications to 1-D Advection-Diffusion Problems," *Computers and Fluids*, Vol. 24, No. 1, 1995, pp. 41–54.

[5]de Nicola, C., Pinto, G., and Tognaccini, R., "A Normal Mode Stability Analysis of Multiblock Algorithms for the Solution of Fluid-Dynamics Equations," *Applied Numerical Mathematics*, Vol. 19, No. 4, 1996, pp. 419–431.

[6]Jenssen, C. B., "Implicit Multiblock Euler and Navier–Stokes Calculations," *AIAA Journal*, Vol. 32, No. 9, 1994, pp. 1808–1814.

[7]Lerat, A., and Wu, Z. N., "Stable Conservative Multidomain Treatments for Implicit Euler Solvers," *Journal of Computational Physics*, Vol. 123, No. 1, 1996, pp. 45–64.

[8]de Nicola, C., Puoti, V., and Tognaccini, R., "A Local Block Processing Strategy for Multiblock Flow Computations," *Parallel Computational Fluid Dynamics: Algorithms and Results Using Advanced Computers*, Elsevier, New York, 1997, pp. 493–500.

[9]Martinelli, L., "Calculations of Viscous Flows with a Multigrid Method," Ph.D. Dissertation, Mechanical and Aerospace Engineering Dept., Princeton Univ., Princeton, NJ, 1987.

[10]Kassies, A., and Tognaccini, R., "Boundary Conditions for Euler Equations at Internal Block Faces of Multiblock Domains Using Local Grid Refinement," AIAA Paper 90-1590, 1990.

[11]Amendola, A., de Nicola, C., Paparone, L., and Tognaccini, R., "Flow Equation Boundary Conditions at General Discontinuous Grid Interfaces,"

Atti del XII Congresso Nazionale dell'Associazione Italiana Di Aeronautica ed Astronautica, Vol. 1, AIDAA, Como, Italy, 1993, pp. 101–112.

[12]Amato, M., Iaccarino, G., and Paparone, L., "Adaptive Local Grid Refinement for Multiblock Solvers," *Proceedings of the 20th Congress of the International Council of the Aeronautical Sciences*, Vol. 2, AIAA, Reston, VA, 1996, pp. 1426–1433.

[13]Hirsch, C., *Numerical Computation of Internal and External Flows*, Vol. 2, Wiley, New York, 1990, pp. 597–598.

[14]de Nicola, C., Pinto, G., and Tognaccini, R., "Stability of Two-Dimensional Model Problems for Multiblock Structured Fluid-Dynamics Calculations," *Computers and Fluids*, Vol. 26, No. 1, 1997, pp. 43–58.

[15]Gustafsson, B., Kreiss, H.-O., and Sundstrom, A., "Stability Theory of Difference Approximations for Mixed Initial Boundary Value Problems. II," *Mathematics of Computation*, Vol. 26, No. 119, 1972, pp. 649–686.

[16]Pandolfi, M., and Borrelli, S., "An Upwind Formulation for Hypersonic Non-Equilibrium Flows," *Modern Research Topics in Aerospace Propulsion in Honor of Corrado Casci*, edited by G. Angelino, L. De Luca, and W. A. Sirignano, Springer-Verlag, Berlin, 1991, pp. 213–226.

M. Sichel
Associate Editor