INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 22, 1153–1167 (1996)

EFFICIENT COMPUTATIONS USING UPWIND BIASED SCHEMES

C. DE NICOLA,* G. IACCARINO AND R. TOGNACCINI

Dipartimento di Progettazione Aeronautica, Università di Napoli, Federico II, Piazzale V. Tecchio 80, Napoli 80125, Italy

SUMMARY

A new multiblock unfactored implicit upwind scheme for inviscid two-dimensional flow calculations is presented. Spatial discretization is carried out by means of an upwind first-order method; an original extension to higher accuracy is also presented. The integration algorithm is constructed in a delta form that provides a direct derivation of the scheme and leads to an efficient computational method. Fast solutions of the linear systems arising at each time step are obtained by means of the bi-conjugate gradient stabilized technique. The computational results for super/hypersonic steady state flows illustrate the efficiency and accuracy of the algorithm.

KEY WORDS implicit multiblock algorithm; upwind scheme; local grid refinement

1. INTRODUCTION

There is now no shortage of numerical methods for the integration of fluid dynamic equations. In particular, significant innovations have been made in the numerical simulation of inviscid compressible flows. In the last few years, upwind methods have become quite popular and have been used for a variety of applications with considerable success.[1] The introduction of physical properties within the discretization of spatial derivatives yields a robust and accurate method (not requiring arbitrary smoothing parameters) for the analysis of discontinuous flow fields.

At present, one of the main drawbacks of modern shock-capturing schemes is their complexity when compared with classical central discretized schemes. In particular we were interested in the first-order upwind scheme of Pandolfi and Borrelli[2] (PB) designed for the computation of non-equilibrium hypersonic flows. The application of this scheme is limited for practical multidimensional steady state calculations by the slow convergence of its explicit one-step time integration.

A significant improvement in the computational efficiency can be achieved by combining different convergence acceleration techniques and computational approaches. In this work, in particular, it will be seen how recent progress in the solution of large sparse linear systems and a multiblock structured approach combined with local grid refinement can reduce computational costs by an order of magnitude for two-dimensional applications.

Implicit integration algorithms have usually been used with upwind discretizations, especially for steady state calculations. Explicit schemes such as multistage time stepping, well suited for central difference schemes, have not found favour since they are difficult to design for optimal performance with upwind methods. Implicit schemes have been praised for their improved stability and condemned

* Current address: CIRA (Centro Italiano Ricerche Aerospaziali), Via Maiorise, 81043, Capua, Italy.

CCC 0271–2091/96/121153–15

Received August 1994
Revised October 1995

for their large operation counts, in particular for multidimensional calculations. In the late 1970s the most significant achievement in this field was the introduction of the alternating direction implicit algorithm.[3] This technique has a Courant–Fredrichs–Lewy (CFL) limit due to the factorization error.

Interest has been spurred recently by the development of new iterative solvers of linear systems that allow efficient calculations with implicit unfactored schemes, giving better convergence and stability properties. There is a considerable literature concerning iterative solutions to linear systems. Although the conjugate gradient technique for the solution of a symmetric, positive definite system is well established, methods for solving non-symmetric problems are still evolving. The generalized minimum residual (GMRES) and conjugate gradient squared (CGS) techniques are efficient methods and have been used in a wide range of applications.[4,5] Another promising method, showing improved convergence properties and computational efficiency, was introduced in 1992 by Van der Voorst,[6] namely the bi-conjugate gradient stabilized (BiCGStab) algorithm.

In this paper we compute two-dimensional super/hypersonic flows by means of an original implicit unfactored upwind scheme based on the PB scheme; the linear system of equations arising at each time step is solved by the BiCGStab technique. A multiblock structured approach (MSA) is used in order to reduce the CPU time required to obtain numerical simulations. Historically the MSA has been introduced to simplify the grid generation process around complex three-dimensional geometries. In this context we are interested in its characteristic of enabling simple implementation of local grid refinement[7] and limiting the number of unknowns when the implicit procedure is employed with a fine grid.

The present method is completely defined without introducing any user parameter: damping terms, artificial viscosity, entropy fix, CFL bound. Its robustness makes this scheme well suited for complex high-speed flows with shock interactions and sonic transition.

In order to verify the effectiveness of the acceleration techniques for accurate compressible simulations, a novel extension of the PB scheme to second-order accuracy has been developed and presented.

A brief description of the PB scheme is presented in Section 2. The new implicit time integration and linearization procedures are shown in Section 3 together with the multiblock structured approach and local grid refinement technique. The numerical results are discussed in Section 4. Finally the symmetric PB (SPB) TVD scheme is presented in Section 5.

2. PANDOLFI–BORRELLI UPWIND SCHEME

We are interested in the numerical solution of inviscid steady flow fields by means of the unsteady Euler equations written in the conservative form

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega} \mathbf{U}\,\mathrm{d}\Omega + \oint_{\partial\Omega} (\mathbf{n}\cdot\mathbf{f})\,\mathrm{d}S = 0, \tag{1}$$

where $\mathbf{U}$ is the vector of conservative variables (density ρ, momentum $\mathbf{m}$, energy per unit volume e), $\mathbf{f}(\mathbf{U})$ is the flux of $\mathbf{U}$ and Ω is the physical domain ($\partial\Omega$ and $\mathbf{n}$ are respectively the border of Ω and its local outward unit normal vector).

Given a structured grid of the domain Ω, a finite volume discretization applied to equation (1) allows shock capturing in compressible flows; it leads to a set of non-linear first-order ordinary differential equations

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{u}_{i,j} + \frac{\mathbf{R}_{i,j}}{\Omega_{i,j}} = 0 \quad \forall\Omega_{i,j} \subset \Omega, \tag{2}$$

where $\mathbf{u}_{i,j}$ is an approximation to the solution in the grid cell volume $\Omega_{i,j}$ and $\mathbf{R}_{i,j}$ is the net flux out of the cell (e.g. in the two-dimensional case $\mathbf{R}_{i,j} = \mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2} + \mathbf{F}_{j+1/2} - \mathbf{F}_{j-1/2}$, where the subscripts $i \pm \frac{1}{2}$ and $j \pm \frac{1}{2}$ specify quantities computed on the border of each quadrilateral Ω and $\mathbf{F}$ is the numerical approximation to the flux $\mathbf{f}$).

The integration of equation (2) is carried out by the classical one-step explicit updating (for the two-dimensional Euler equations)

$$\mathbf{u}_{i,j}^{n+1} = \mathbf{u}_{i,j}^{n} - \frac{\Delta t_{i,j}}{\Omega_{i,j}} R_{i,j}^{n} \quad \forall \Omega_{i,j} \subset \Omega, \tag{3}$$

where $\Delta t_{i,j}$ is the local time step computed according to the CFL stability limit.

Upstream differencing attempts to discretize the equations by using differences biased in the direction determined by the sign of the characteristic speed; the interface state is computed as a solution to a Riemann problem (RP).

The evaluation of numerical fluxes in equation (3) is based on Godunov's basic idea of flux difference splitting (FDS). A 'projection' stage makes it possible to identify two constant states at a given interface that are the initial values for the RP (Figures 1(a) and 1(b)).

At the 'evolution' stage the RP is solved to select the 'correct' interface state ($\mathbf{u}_{i+1/2,j} = \mathbf{u}_c$ in Figure 1(c)). In multidimensional FDS schemes the RP is solved in the direction normal to the cell interface as a one-dimensional RP.

Modern FDS methods solve the RP in an approximate but efficient way; historically, Roe[8] and Osher and Solomon[9] anticipated Pandolfi and Borrelli's approximate Riemann solver (ARS). Roe's ARS is based on a local linearization of the Euler equations, providing exact resolution of discontinuities but allowing steady solutions with expansion shocks, and requires a numerical parameter to satisfy the entropy condition (entropy fix). Osher's solver is more complicated and is based on the 'simple wave solution' of the RP. This leads to the exclusion of unphysical expansion discontinuities and provides a description of steady shocks with two interior cells and no overshoot.

The PB approach may be considered as a modification[10] of the Osher ARS. The numerical flux $\mathbf{F}$ is given by

$$\mathbf{F}_{i+1/2} = \mathbf{F}(\mathbf{u}_{i,j}, \mathbf{u}_{i+1,j}) = \frac{1}{2}\left(\mathbf{f}_{i,j} + \mathbf{f}_{i+1,j} - \int_{\mathbf{u}_{i,j}}^{\mathbf{u}_{i+1,j}} \left|\frac{\partial f}{\partial u}\right| \mathrm{d}u\right), \tag{4}$$

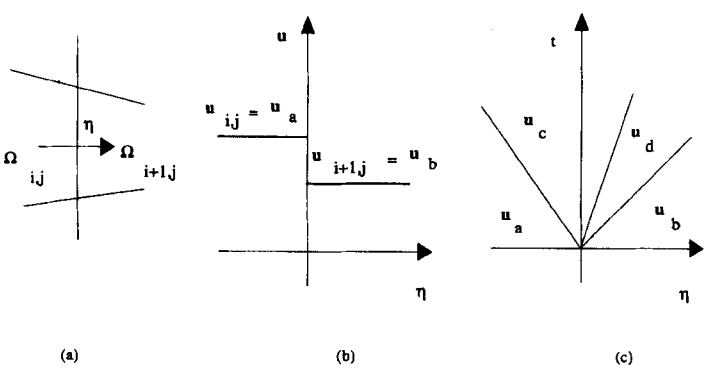

Figure 1. The Riemann problem

where $\mathbf{f}_{i,j} = \mathbf{f}(\mathbf{u}_{i,j})$. To evaluate the integral in equation (4), two intermediate points $\mathbf{u}_c$ and $\mathbf{u}_d$ are introduced ($\mathbf{u}_{i,j} = \mathbf{u}_a$, $\mathbf{u}_{i+1,j} = \mathbf{u}_b$). These points define three subpaths Γ_k ($k = 1, 2, 3$) along which the integral is computed and these are associated with an eigenvalue of the Jacobian matrix $\partial\mathbf{f}/\partial\mathbf{u}$ and a set of characteristic variables. These quantities remain constant along the subpath and this property is used in calculating the dependent variables at the intermediate points. For example,

$$\int_{\Gamma_2} \left|\frac{\partial f}{\partial u}\right| \mathrm{d}u = \int_{\mathbf{u}_c}^{\mathbf{u}_d} \left|\frac{\partial f}{\partial u}\right| \mathrm{d}u = \mathrm{sign}(\lambda_2)[f(u_d) - f(u_c)], \tag{5}$$

where λ_2 is the eigenvalue associated with the second subpath. The main difference between the Osher and PB solvers is in the ordering of the subpath in (4); in the present paper we use PB ordering with increasing eigenvalues. Another important difference lies in the solution of the RP, which is carried out by means of the Riemann invariants in the Osher ARS while in the PB approach the characteristic variables are used.

In Appendix I the complete algorithm for the Euler flux computation of the PB scheme is reported.

3. THE NEW TIME INTEGRATION ALGORITHM

Historically, multidimensional calculations using implicit schemes were carried out by means of factored techniques in which the differential operator was split into the product of one-dimensional operators.[11] In this way the computational cost and complexity of the integration scheme diminish, because simple and fast direct solvers of (block) tridiagonal matrices can be applied. The factorization error introduces a stability limit (bounded time step), reducing the efficiency for steady state calculations.

Another traditional technique for solving fluid dynamic equations is the Newton iterative algorithm,[12] which, unlike the time relaxation procedure, attempts to integrate the steady equations directly. The most interesting property of the Newton iterative approach is its 'superlinear' convergence to the solution, but an undoubted drawback is the excessive dependence on the initial guess.

In the past the use of an unfactored implicit time-marching procedure was limited owing to large memory requirements and operation counts, especially in the case of sparse (with very large band) linear systems. Recently, interest in the application of these methods has been revived thanks to the introduction of very fast linear solvers and the advent of large, powerful computers.

These advantages are amplified by employing an MSA technique both for the capability of introducing locally refined grids and for the reduction of the system unknowns.

3.1. Linearizing procedure and system solver

The implicit time integration of equation (3) requires the evaluation of the numerical fluxes at the implicit level, leading to a system of non-linear equations. This can be solved in a non-iterative way after linearizing the fluxes with respect to time. They are computed by means of the PB upwind scheme; the linearization is carried out by Taylor expansion. For example,

$$\mathbf{F}_{i+1/2}^{n+1} = \mathbf{F}(\mathbf{u}_{i,j}^{n+1}, \mathbf{u}_{i+1,j}^{n+1}) = \mathbf{F}_{i+1/2}^{n} + \frac{\partial \mathbf{F}_{i+1/2}^{n}}{\partial \mathbf{u}_{i,j}^{n}} \Delta\mathbf{u}_{i,j} + \frac{\partial \mathbf{F}_{i+1/2}^{n}}{\partial \mathbf{u}_{i+1,j}^{n}} \Delta\mathbf{u}_{i+1,j}. \tag{6}$$

Analytical computation of the Jacobian matrices in (6) is very difficult;[13] we calculate the derivatives in a numerical way by solving a set of modified RPs. For example, the first Jacobian matrix $\mathbf{J} = [\mathbf{J}_m]$ $(m = 1, 2, 3, 4)$ in (6) can be obtained as

$$\mathbf{J}_m = \frac{\partial \mathbf{F}_{i+1/2}}{\partial (u_m)_{i,j}} = \frac{\mathbf{F}(\tilde{\mathbf{u}}_{i,j}, \mathbf{u}_{i+1,j}) - \mathbf{F}(\mathbf{u}_{i,j}, \mathbf{u}_{i+1,j})}{\Delta u}, \quad m = 1, 2, 3, 4, \tag{7}$$

where $\tilde{\mathbf{u}} = [u_n + \delta_{mn}\Delta u]^{\mathrm{T}}$ $(n = 1, 2, 3, 4; \delta_{mn}$ is the Kroneker symbol) and $\Delta u = 10^{-4}$.

The implicit linearized PB scheme can be written in a compact form as

$$\left[\frac{\Omega_{i,j}}{\Delta t_{i,j}} I - \left(\frac{\partial R}{\partial u}\right)^n_{i,j}\right] \Delta u_{i,j} + \sum_{s=1}^{4} \left(\frac{\partial R}{\partial u}\right)^n_s \Delta u_s = -R^n_{i,j} \quad \forall \Omega_{i,j} \subset \Omega, \tag{8}$$

where the index s indicates the neighbouring cells of the cell i, j. Equation (8) drives to a block pentadiagonal (each block being 4×4 in 2D) linear system; the solution is required at each time step.

It is interesting to note that this scheme is an 'augmented' Newton method (it becomes[5] a standard Newton iteration if $\Delta t \to \infty$) and that the obtained steady state is independent of the time step.

The linear system given by equation (8) is an approximate form of the PB non-linear implicit scheme (3). To obtain fast convergence, it has been found[4] that it is better to solve the linear problem to a moderate degree of precision and proceed to the next time step; thus iterative methods are well suited to the present framework.

In recent years the conjugate gradient squared method has been recognized as an attractive variant of the conjugate gradient iterative technique for the solution of non-symmetric linear systems. In many situations, however, one is faced with a quite irregular convergence behaviour of CGS, in particular when starting the iterations close to the solution. This is a common situation in the final stages of time-dependent problems and motivated our search for a smoothly converging variant of CG which did not forfeit the attractive speed of convergence of CGS. This proved to be the BiCGStab technique of Van der Voorst, accurately described in his original paper.

3.2. The multiblock structured approach

The physical domain is partitioned into *NB* regions (blocks) in which a structured grid is built and where the flow equations are integrated independently. Specifying by b the index of block B_b $(B_1 \cup \cdots \cup B_b \cup \cdots \cup B_{NB} = \Omega)$, the problem is reduced to

$$\forall b = \{1, 2, \ldots, NB\}: \quad \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{u}_{i,j,b} + \frac{\mathbf{R}_{i,j,b}}{\Omega_{i,j,b}} = 0 \quad \forall \Omega_{i,j,b} \subset B_b \tag{9}$$

Additional 'artificial' or internal boundary conditions need to be specified on the internal boundaries introduced by the block decomposition. They consist of defining a proper RP for each cell face belonging to the block interface. Their numerical implementation depends on the allowed structure of the grid across blocks. In fact, an interesting feature of the MSA is the possibility to built independent grids in each block where the grid size can be chosen according to the relevance of the flow region and to the expected flow gradient. In this way the total number of grid cells can be significantly reduced, preserving the target accuracy. In the work of Hessenius and Rai[14] a conservative boundary scheme for a discontinuous grid across the block interface was presented and applied to a first-order upwind solver. We propose here a simpler and more efficient approach that allows the computation of the numerical flux at an interface with 'partial' discontinuity in the distribution of grid points: the refinement (in the

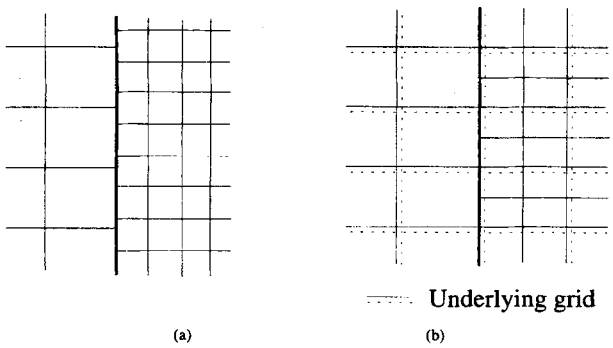

Figure 2. Discontinuous grid versus partially continuous grid

fine block) is applied to the underlying continuous grid through the interface (Figure 2). The algorithm can be better explained with the help of Figure 3. An extra layer of grid cells is added to the 'fine' block B_1. In these cells the flow state is computed by copying the states of the overlapping coarse cells:

$$\mathbf{u}_{NI_1+1,j_2+k-1,B_1} = \mathbf{u}_{1,j_2,B_2}, \quad \forall j_2 = 1,\ldots,NJ_2, \quad \forall k = 1,\ldots n \tag{10}$$

Hence the interface flux for block B_1 can be computed by solving a set of RPs by the same inner block algorithm. Finally, the fluxes of the 'coarse' block B_2 are computed by forcing their conservation across the internal face:

$$\mathbf{F}_{1/2,j_2,B_2} = \sum_{k=1}^{n} \mathbf{F}_{NI_1+1/2,j_2+k-1,B_1} \tag{11}$$

The proposed MSA modifies the time integration procedure, since the block decomposition splits the implicit upgrade of the flow state $\mathbf{u}$ in Ω in the solution of NB uncoupled linear systems and the applied internal boundary conditions have an explicit nature. For this reason, modifications in the convergence histories can be expected owing to the introduction of the MSA. The study of these effects is outside the scope of the present paper and will not be treated here.

4. RESULTS

The selected 2D inviscid test cases are supersonic channel flow (test A) and hypersonic blunt body flow (test B).

Test A was proposed by Spekreijse,[15] who adopted a first-order Osher ARS and a second-order MUSCL Osher ARS. It consists of a supersonic channel ($M_{\text{inlet}} = 1{\cdot}4$) with a 4 per cent circular arc bump. The computational grid has 96 × 32 cells and is reported in Figure 4 together with the iso-Mach contours of the solution. In Figure 5 the Mach number distribution on the lower surface of the channel is compared with the results of Spekreijse and demonstrates a substantial agreement between the PB and the Osher ARS.

A comparison between linear system solvers has been carried out to evaluate the performance of the BiCGStab technique. All the computations have been performed on a Convex 3420 vector computer. In Figure 6 are reported the convergence histories (residual versus CPU time) of some iterative linear system solvers (SOR, CGS, BiCGStab) and one classical direct solver (Crout). The conjugate gradient methods (CGS, BiCGStab) provided the best performance. The Van der Voorst algorithm demonstrates its effectiveness especially in damping the higher frequencies, as can be seen from the final part of the convergence.

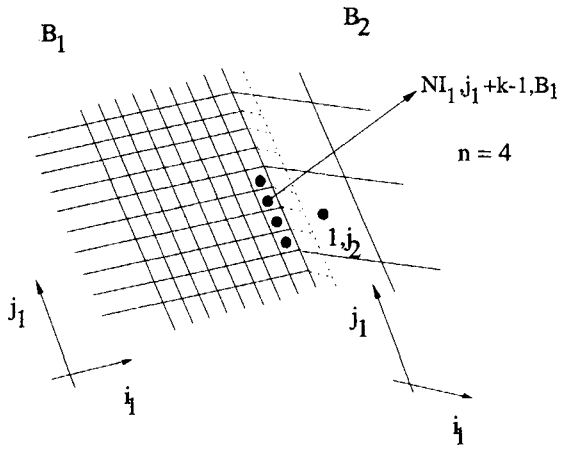

Figure 3. Computation of fluxes at an LGR interface

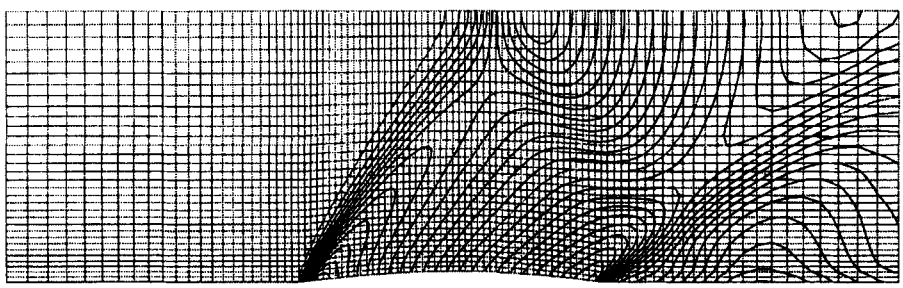

Figure 4. Test A: grid and iso-Mach contours ($\Delta M = 0{\cdot}1$)

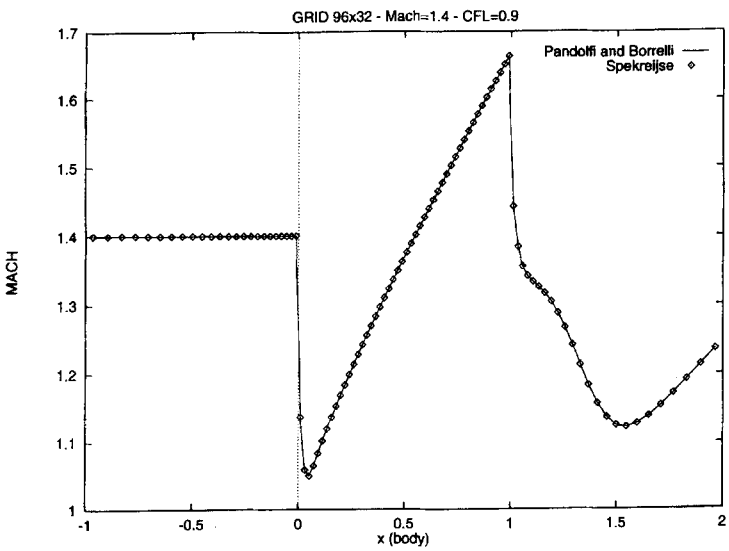

Figure 5. Test A: Mach number distribution on lower surface of channel

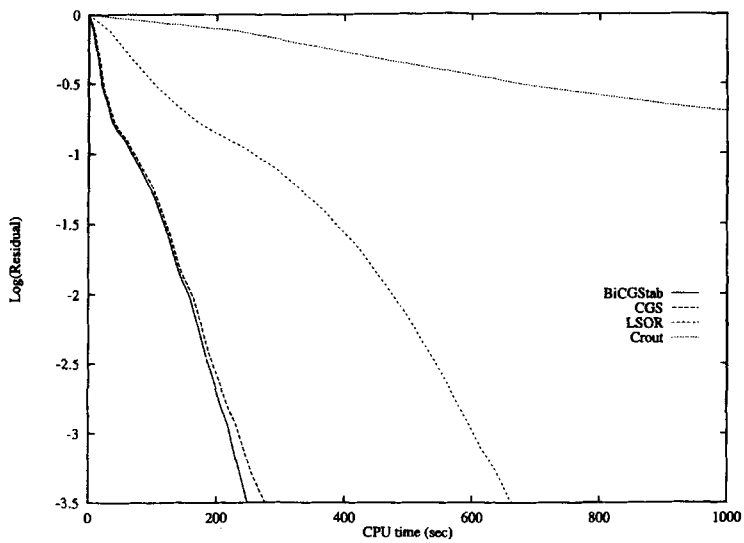

Figure 6. Test A: comparison of linear system solvers

The performance of the new implicit algorithm is presented with respect to the original PB explicit scheme in Figure 7. A drastic reduction in the number of time steps required is evident.

Table I lists the CPU times required for both explicit and implicit schemes: improvements of one order of magnitude are shown. In this table the speed-up is the ratio between the explicit and the implicit CPU time.

The implicit PB scheme is always stable, but in practice there is no point in raising the CFL above 1000 because of the large discretization error introduced in the integration of the equation (scheme (8) is only first-order-accurate in time).

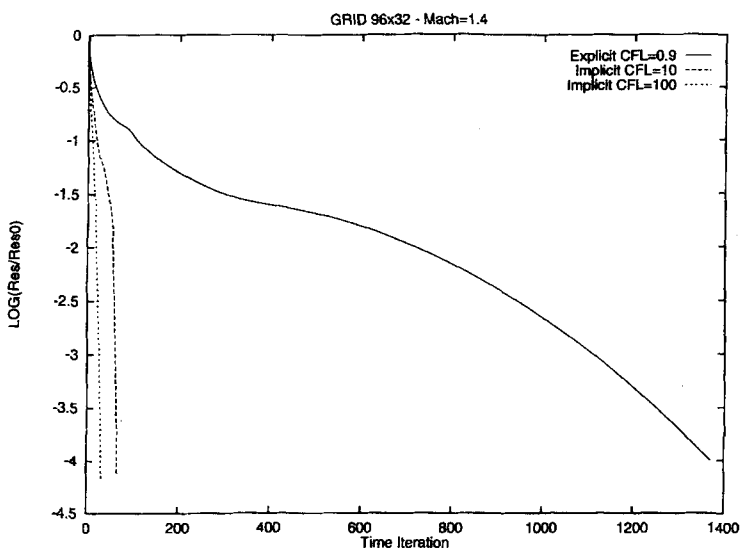

Figure 7. Test A: convergence history of implicit versus explicit PB scheme

Table 1. CPU times required for 2D wind tunnel test case on a Convex 3420

Time integration	CFL	Iterations	CPU time (s)	Speed-up
Explicit	0·9	1385	1634·4	1
Implicit	10	78	541·7	3·02
Implicit	100	48	257·3	6·35
Implicit	500	46	221·7	7·37
Implicit	1000	47	238·5	6·85
Implicit	2000	51	271·1	6·03

Test B consists of a hypersonic flow ($M_\infty = 5$, $\alpha = 30°$) past a double ellipse. In Figure 8 we show the solutions obtained using the original explicit upwind scheme (left) and the present method with a locally refined grid (right). The continuous one-block grid is built by 256 × 92 cells. The LGR grid is composed of 24 blocks with a total of 11,530 cells. The refinement strategy has allowed us to employ finer meshes in the nose region with subsonic flow (Figure 9) and in the canopy shock region. In this relatively simple case the LGR has been chosen *a priori*; for more complex applications an automatic adaptive LGR procedure could be implemented on the basis of the local flow gradient.

In Figure 10 the Mach number distribution on the solid wall is presented: the solutions for the one-block fine grid and for the LGR grid are equivalent in spite of a saving of 13,056 cells. It is interesting to analyse the behaviour of the solution at the LGR interfaces near the solid wall. The solutions are strictly

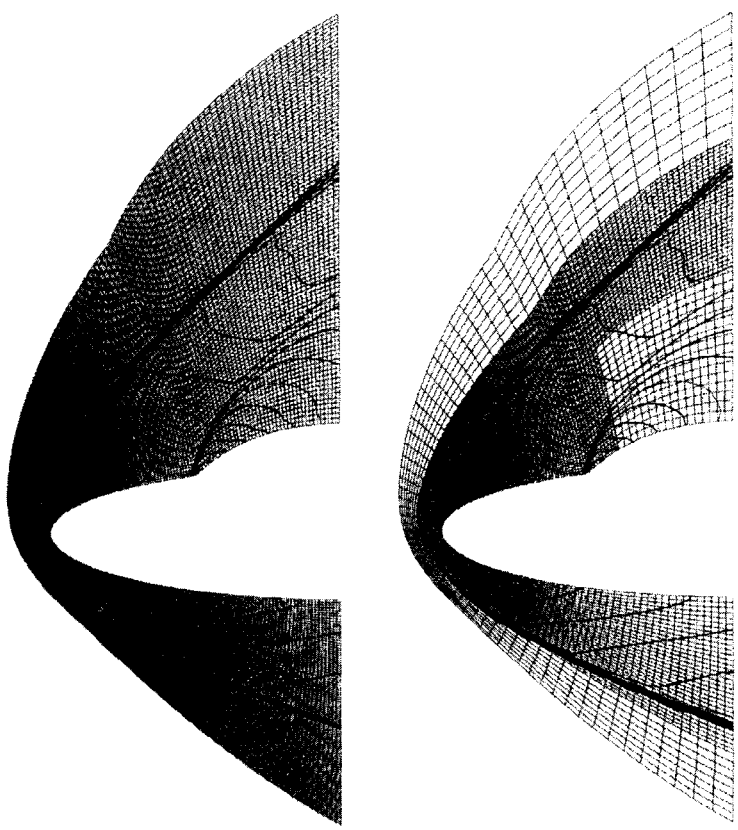

Figure 8. Test B: continuous grid versus LGR grid and iso-Mach contours ($\Delta M = 0{\cdot}25$)

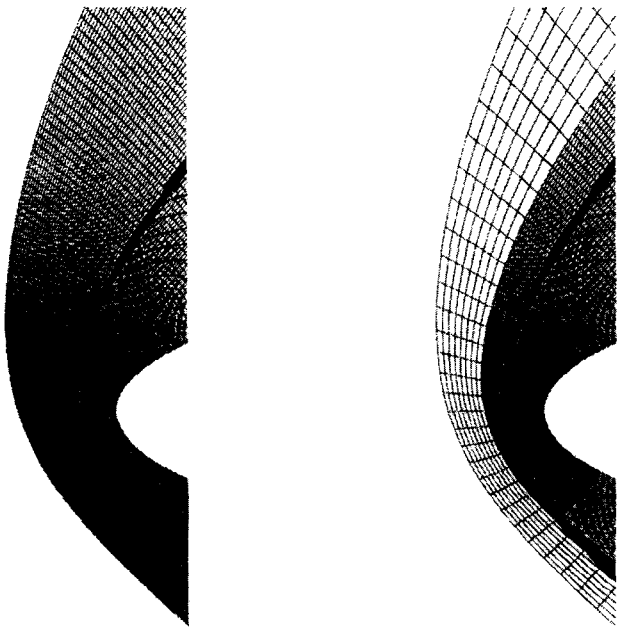

Figure 9. Test B: enlargement of nose region ($\Delta M = 0{\cdot}1$)

the same where the refined grid is equal to the fine grid; the local nominal accuracy of the scheme is respected across the interface and is slightly reduced in the coarser regions according to the local mesh size.

In terms of computational costs the LGR implicit calculation required 1071·2 s of CPU time while the fine grid computation with the PB explicit scheme took 15640·5 s. The convergence histories are reported in Figure 11.

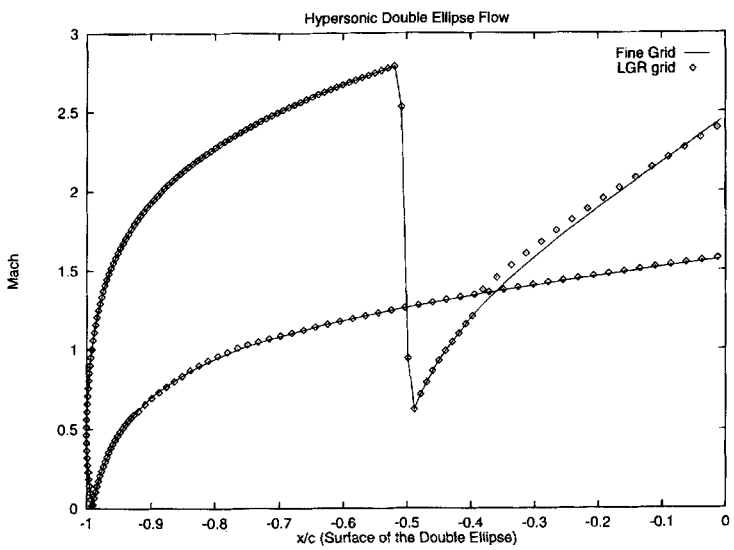

Figure 10. Test B: Mach number distribution on solid wall

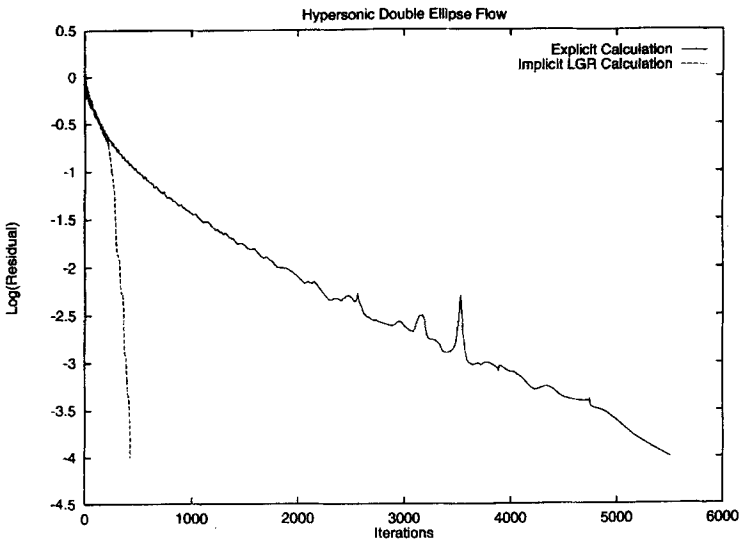

Figure 11. Test B: convergence history of implicit versus explicit PB scheme (CFL explicit = 0·95, CFL implicit = 200)

5. HIGH-ORDER UPWIND SCHEME

Owing to the large numerical dissipation introduced, first-order upwind methods do not allow accurate solutions of smooth flow regions. We therefore extended the PB scheme to second-order accuracy to discover whether the proposed techniques could provide sufficient speed-up in practical CFD applications.

The definition of high-resolution schemes is based on certain mathematical concepts such as monotonicity and TVD.[16] We are interested in TVD upwind schemes that can be classified on the basis of the principle used to compute interface fluxes in 'algebraic' or 'geometric' methods.

Algebraic schemes are constructed by adding low-order and high-order approximations of the numerical flux by means of a weight function that is computed on the basis of 'flux' limiters.[17] In the geometric approach, attempts are made to reconstruct dependent variables within each control volume subject to monotone constraints by using 'slope' limiters.

From a more general point of view we can upgrade the PB first-order scheme to second-order accuracy by modifying the projection stage or the evolution stage, obtaining respectively a MUSCL-like or a symmetric PB (SPB) scheme. In this context we show results concerning the application of the SPB scheme.

The basic idea is extensively discussed in the work of Yee.[18] Roughly speaking, we modify the numerical flux evaluation, equation (4), by introducing a limiting procedure, e.g.

$$\mathbf{F}_{i+1/2,j} = \mathbf{F}(\mathbf{u}_{i-1,j}, \mathbf{u}_{i,j}, \mathbf{u}_{i+1,j}, \mathbf{u}_{i+2,j}) = \frac{1}{2}\left(\mathbf{f}_{i,j} + \mathbf{f}_{i+1,j} - \sum_{k=1}^{3}(1 - Q^{(k)}) \int_{\Gamma_k} \left|\frac{\partial f}{\partial u}\right| \mathrm{d}u\right), \tag{12}$$

where $Q^{(k)} = Q(r^+_{(k)}, r^-_{(k)})$. With reference to Section 2 we can define, e.g. along Γ_2,

$$Q^{(2)} = Q(r^+_{(2)}, r^-_{(2)}) = Q\left(\frac{W^{(2)}_{i+3/2}}{W^{(2)}_{i+1/2}}, \frac{W^{(2)}_{i-1/2}}{W^{(2)}_{i+1/2}}\right) \tag{13}$$

where W is the integral along the subpath Γ_2 of the corresponding Riemann variable. Some limiter functions Q are listed in Reference 18; we use $Q = \text{MinMod}(1, r^+, r^-)$, which is a compromise between 'compressive' behaviour and simplicity.

In Appendix II the complete algorithm for the flux computation of the SPB scheme is presented.

In multidimensional flow calculations, SPB is more accurate than PB but too dissipative with respect to Osher MUSCL.[12] This is shown in Figure 12 for test A. Better results could be obtained using a more 'compressive' limiter, but in this context we are interested in the efficiency of the acceleration techniques with high-order schemes.

The decay of the convergence rate for the explicit SPB method with respect to PB is shown in Figure 13. A limit cycle results with periodic oscillations of the residual. This is a consequence of the highly non-linear nature of the SPB scheme. The implicit method preserves its fast convergence property even if conjugated with second-order flux evaluation (in the LHS of equation (8) we simply took $Q = 0$ and therefore the shape of the implicit operator is preserved).

6. CONCLUSIONS

A new multiblock implicit algorithm for the solution of the two-dimensional Euler equations is presented. The method is based on the upwind formulation of Pandolfi and Borrelli. The implicit unfactored method is written in 'delta' form and leads to an efficient and compact computational method. The linear systems arising at each time step are solved by a modern technique belonging to the conjugate gradient class, BiCGStab.

The present formulation has allowed us to rule out all the tuning parameters (artificial viscosity, entropy fix, CFL bound) in such a way as to obtain a robust and efficient method well suited for super/hypersonic shock-dominated problems.

Numerical experiments were performed to validate this scheme. One selected test case was a 2D hypersonic blunt body flow. The CPU time required is reduced by a factor of 14 when the implicit time-marching procedure is used together with a locally refined grid.

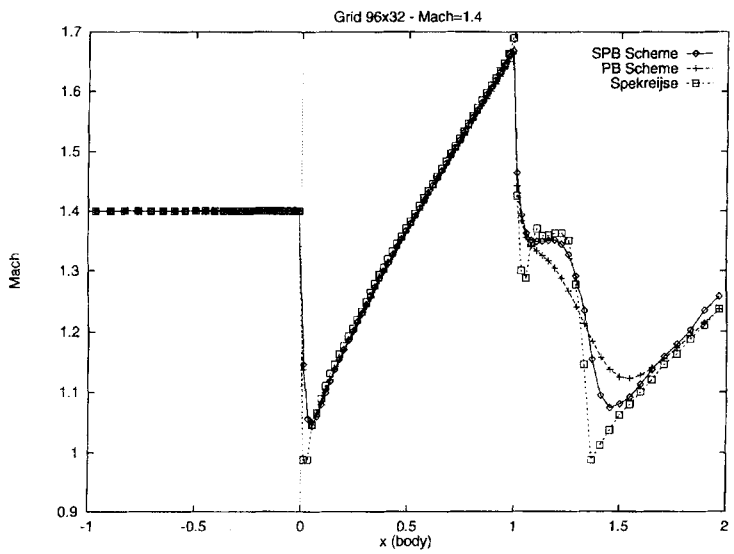

Figure 12. Test A: Mach number distribution on lower surface of channel

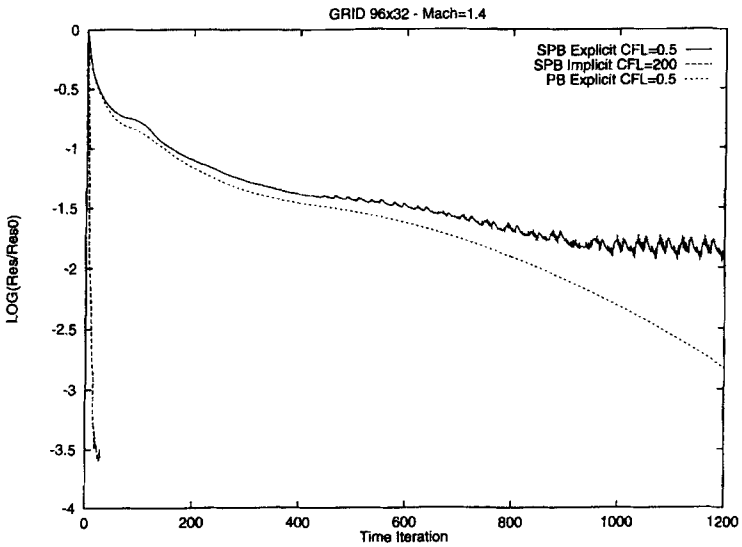

Figure 13. Test A: convergence history of implicit versus explicit SPB scheme (CFL explicit = 0·95, CFL implicit = 200)

The original first-order spatial discretization has been upgraded to second- order accuracy to verify that the developed time integration can be used together with the new generation of TVD schemes.

ACKNOWLEDGEMENTS

The authors wish to thank Alfonso Matrone and Vittorio Puoti of CIRA (Centro Italiano Ricerche Aerospaziali), who provided the linear system solvers tested in this paper, and Salvatore Borrelli of CIRA for fruitful discussions.

APPENDIX I: NUMERICAL FLUX EVALUATION WITH PB SCHEME

The procedure to calculate the numerical flux $\mathbf{F}_{i+1/2}$ can be divided into six steps.

1. Projection stage:

$$\mathbf{u}_a = [\rho_{i,j} v_{i,j} p_{i,j}]^T = [\rho_a, v_a, p_a]^T, \qquad \mathbf{u}_b = [\rho_{i+1,j}, v_{i+1,j}, p_{i+1,j}]^T = [\rho_b, v_b, p_b]^T.$$

2. Set RP:

$$p_c + (\rho_a a_a) v_c = p_a + (\rho_a a_a) v_a, \qquad h_c - p_c/\rho_a = h_a - p_a/\rho_a,$$
$$p_c = p_d, \qquad v_c = v_d,$$
$$p_d - (\rho_b a_b) v_d = p_b - (\rho_b a_b) v_b, \qquad h_d - p_d/\rho_b = h_b - p_b/\rho_b.$$

3. Solve RP:

$$\mathbf{u}_c = [\rho_c, v_c, p_c]^T, \qquad \mathbf{u}_d = [\rho_d, v_d, p_d]^T.$$

4. Compute flux **f**:

$$\mathbf{f}_a = \mathbf{f}(\mathbf{u}_a), \qquad \mathbf{f}_b = \mathbf{f}(\mathbf{u}_b), \qquad \mathbf{f}_c = \mathbf{f}(\mathbf{u}_c), \qquad \mathbf{f}_d = \mathbf{f}(\mathbf{u}_d).$$

5. Compute eigenvalues:

$$\lambda_1 = v_a - a_a, \qquad \lambda_2 = v_c, \qquad \lambda_3 = v_b + a_b.$$

6. Evolution stage:

$$\mathbf{F}_{i+1/2} = (\mathbf{f}_a + \mathbf{f}_b)/2 - [\text{sign}(\lambda_1)(\mathbf{f}_c - \mathbf{f}_a) + \text{sign}(\lambda_2)(\mathbf{f}_d - \mathbf{f}_c) + \text{sign}(\lambda_3)(\mathbf{f}_b - \mathbf{f}_d)]/2.$$

Here p is the pressure, ρ is the density, $a = (\gamma p/\rho)^{1/2}$ is the speed of sound (γ is the constant of the gas), $h = a^2/(\gamma - 1)$ is the enthalpy and v is the velocity normal to the considered interface.

APPENDIX II: NUMERICAL FLUX EVALUATION WITH SPB SCHEME

This procedure is almost the same as the one presented in Appendix I. It differs only in introducing limiters in the evolution stage. To calculate $\mathbf{F}_{i+1/2}$, we have to compute $\mathbf{Q}_{i+1/2}$ (see equation (12)). The kth component is

$$Q^{(k)} = Q(r^+_{(k)}, r^-_{(k)}) = Q\left(\frac{W^{(k)}_{i+3/2}}{W^{(k)}_{i+1/2}}, \frac{W^{(k)}_{i-1/2}}{W^{(k)}_{i+1/2}}\right)$$

where $\mathbf{W} = [p_c - p_a - \rho_a a_a(v_c - v_a),\ h_d - h_c,\ p_b - p_d - \rho_b a_b(v_b + v_d)]^T$ is the vector of Riemann variables. To evaluate the flux across the interface $i + \frac{1}{2}$, we must compute $\mathbf{W}$ at $(i - \frac{1}{2}, i + \frac{1}{2}, i + \frac{3}{2})$ and $\mathbf{Q}$. Then we can limit the evolution stage:

$$\mathbf{F}_{i+1/2} = (\mathbf{f}_a + \mathbf{f}_b)/2 - [(1 - Q^{(1)})\text{sign}(\lambda_1)(\mathbf{f}_c - \mathbf{f}_a) + (1 - Q^{(2)})\text{sign}(\lambda_2)(\mathbf{f}_d - \mathbf{f}_c)$$

$$+(1 - Q^{(3)})\text{sign}(\lambda_3)(\mathbf{f}_b - \mathbf{f}_d)]/2.$$

We use the two-argument limiter defined in Reference 18:

$$Q = \text{MinMod}(1, r^+, r^-) = \text{Max}(0, \text{Min}(1, r^+, r^-)).$$

APPENDIX III: ABBREVIATIONS

ARS	approximate Riemann solver
BiCGStab	bi-conjugate gradient stabilized
CFL	Courant–Fredrichs–Lewy
CGS	conjugate gradient squared
FDS	flux difference splitting
GMRES	generalized minimum residual
LGR	local grid refinement
LHS	left-hand side
MSA	multiblock structured approach
MUSCL	monotone upstream scheme for conservation laws
PB	Pandolfi–Borrelli
RHS	right-hand side
RP	Riemann problem
SPB	symmetric PB
TVD	total variation diminishing

REFERENCES

1. A. Rizzi and B. Engquist, 'Selected topics in the theory and practice of computational fluid dynamics', *J. Comput. Phys.*, **72**, 1–69 (1987).
2. M. Pandolfi and S. Borrelli, 'An upwind formulation for hypersonic non-equilibrium flows', in *Modern Research Topics in Aerospace Propulsion*, Springer, Berlin, 1991.
3. R. M. Beam and R. F. Warming, 'An implicit factored scheme for the compressible Navier–Stokes equations', *AIAA J.*, **16**, (1978).
4. P. Sonneveld, 'CGS, a fast Lanczos-type solver for nonsymmetric linear systems', *SIAM J. Sci. Stat. Comput.*, **10**, (1989).
5. V. Venkatakrishnan, 'Preconditioned conjugate gradient methods for the compressible Navier–Stokes equations', *AIAA J.*, **29**, (1990).
6. H. Van der Voorst, 'BiCGStab: a fast and smoothly converging variant of the BiCG for the solution of nonsymmetric linear systems', *SIAM J. Sci. Stat. Comput.*, **13**, (1992).
7. A. Kassies and R. Tognaccini, 'Boundary conditions for Euler equations at internal block faces of multiblock domain using local grid refinement', *AIAA Paper*, 90–1590, 1990.
8. P. Roe, 'Approximate Riemann solvers, parameters vector and difference schemes', *J. Comput. Phys.*, **43**, 357–372 (1981).
9. S. Osher and F. Solomon, 'Upwind difference schemes for hyperbolic systems of conservation laws', *Math. Comput.*, **38**, (1982).
10. M. Pandolfi, 'A contribution to the numerical prediction of unsteady flows', *AIAA J.*, **22**, 602–610 (1984).
11. T. H. Pulliam and J. L. Steger, 'Implicit finite-difference simulations of three-dimensional compressible flow', *AIAA J.*, **18**, (1980).
12. V. Venkatakrishnan, 'Newton solution of inviscid and viscous problems', *AIAA J.*, **27**, (1989).
13. M. M. Rai and S. R. Chakravarty, 'An implicit form for the Osher upwind scheme', *AIAA J.*, **22**, (1987).
14. K. A. Hessenius and M. M. Rai, 'Applications of a conservative zonal scheme to transient and geometrically complex problems', *Comput. Fluids*, **14**, 43–58 (1986).
15. S. P. Spekreijse, 'Multigrid solution of the steady Euler equations', *Ph.D. Thesis*, CWI, Amsterdam, 1987.
16. A. Harten, P. D. Lax and B. Van Leer, 'On upstream differencing and Godunov-type schemes for hyperbolic conservation laws', *SIAM Rev.*, **25**, (1983).
17. P. K. Sweby, 'High resolution TVD schemes using flux limiters', talk given at *AMS–SIAM Summer Seminar on Large Scale Computations in Fluid Mechanics*, La Jolla, CA, 1983.
18. H. C. Yee, Construction of explicit and implicit symmetric TVD schemes and their applications', *J. Comput. Phys.*, **68**, 151–179 (1987).
19. A. Harten, R. F. Warming and H. C. Yee, 'Implicit total variation diminishing schemes for steady state calculations', *J. Comput. Phys.*, **57**, 327–360 (1985).
20. S. Borrelli, A. Matrone and P. Schiano, 'A multiblock hypersonic flow solver for massively parallel computer', *Proc. Conf. on Parallel CFD '92*, New Brunswick, NJ, May 1992.