AIAA JOURNAL
Vol. 32, No. 9, September 1994

Implicit Multiblock Euler and Navier-Stokes Calculations

Carl B. Jenssen*
SINTEF–The Foundation for Scientific and Industrial Research at the Norwegian Institute of Technology, 7034 Trondheim, Norway

Implicit multiblock computations have been carried out for a large number of blocks using explicit coupling between the blocks. The convergence rate of this method is very sensitive to the block partitioning. For the Navier-Stokes equations the height of the blocks should be greater than the boundary-layer thickness. Also, excessively thin blocks will cause breakdown of the algorithm. For transonic calculations the best convergence rate was obtained using a red-black Gauss-Seidel approach. It is concluded that the method is well suited for massively parallel computers.

Introduction

In the last few years, massively parallel multiple-instruction multiple-data (MIMD) computers with distributed memory have become a competitive alternative to traditional vector supercomputers for large-scale flow calculations. For the majority of flow solvers, whether they are based on structured or unstructured grids, the most efficient way to utilize these computers is to use some sort of multiblock algorithm.

Explicit methods are straightforward to apply for multiblock computations, as the numerical scheme remains equivalent to that of the corresponding single-block algorithm. For implicit methods however, the final numerical scheme depends on how the implicit block interface conditions are treated. Although not in common use for fluid flow problems, domain decomposition methods have been used in a limited number of cases for both model problems[1] and realistic flows.[2] Since domain decomposition methods introduce additional work, implicit schemes are often considered as not well suited for parallel processing.

The simplest method is, however, to use explicit coupling between blocks, reducing the global nature of the implicit scheme, but causing no extra computational overhead. Such an approach has been used with success by several authors[3–7] for a moderate number of blocks, although little effort has gone into determining the effect of using explicit coupling compared with employing a fully coupled method. Indeed, for a moderate number of blocks these effects might be insignificant, but as the number of blocks is increased to utilize a large number of processors, the loss of convergence rate and stability can in some cases be severe and can no longer be ignored.

However, if the adverse effects on the convergence rate can be controlled, very high efficiency can be obtained by avoiding the use of a computationally expensive globally coupled solution procedure. One should also keep in mind that on parallel machines the communication overhead using such a scheme is much smaller than for an explicit scheme, as implicit schemes require many more operations per time step than explicit schemes.

In this paper we identify some important factors greatly affecting the convergence rate of implicit multiblock calculations with explicit interdomain coupling based on previous work by the author.[8] Since the limit of this method as the number of blocks increases is the point Jacobi method, we know in advance that a significant loss of efficiency will occur at some stage. Our aim is thus to get a clear idea about how many blocks can be used while keeping a reasonably high convergence rate. Specifically, we would like to see if calculations can be carried out using the number of blocks required for large parallel systems (i.e., the order of 100 blocks).

After an analysis of the stability and convergence rate for a model problem, the method is tested on realistic three-dimensional Euler and Navier-Stokes computations. Although subsonic cases can be considered more difficult due to the more elliptic or global nature of the equations, supersonic cases make it easier to isolate the various other effects that influence the convergence rate. Thus, two supersonic cases are first considered, before the problem of subsonic flow is highlighted by the computation of a transonic flowfield.

The flow solver is based on Roe's scheme with second-order total variation diminishing (TVD) extrapolation for the convective part of the equations. Laminar viscous terms are discretized using conventional central differencing. In each block the resulting linear system is solved using line relaxation. The resulting algorithm is thus a block Jacobi method, with approximate solution in each block.

Governing Equations

The Navier-Stokes equations can be found in any textbook on viscous fluid flow, and we thus give only a general symbolic form here. Consider a system of conservation laws written in integral form:

$$\frac{\partial}{\partial t}\int_\Omega U\,\mathrm{d}V + \int_{\partial\Omega} \boldsymbol{F}\cdot\boldsymbol{n}\,\mathrm{d}s = 0 \tag{1}$$

where U is the conserved variable, $\boldsymbol{F}$ the flux vector, and $\boldsymbol{n}$ the outward-pointing normal to the surface $\partial\Omega$ enclosing an arbitrary volume Ω. We assume that the flux vector can be split into one inviscid and one viscous part, $\boldsymbol{F}=\boldsymbol{F}_{\text{inv}}(U)+\boldsymbol{F}_{\text{vis}}(U, \nabla U)$, where the Jacobian of $\boldsymbol{F}_{\text{inv}}\cdot\boldsymbol{n}$ with respect to U has real eigenvalues and a complete set of eigenvectors, and furthermore that $\boldsymbol{F}_{\text{vis}}\cdot\boldsymbol{n}$ depends linearly on ∇U. In a Cartesian coordinate system x_1, x_2, x_3, we can then write

$$\boldsymbol{F}_{\text{vis}}\cdot\boldsymbol{n} = A_{n,x_1}\frac{\partial U}{\partial x_1} + A_{n,x_2}\frac{\partial U}{\partial x_2} + A_{n,x_3}\frac{\partial U}{\partial x_3} \tag{2}$$

where the matrices A_{n,x_l} are functions of U and $\boldsymbol{n}$. The Navier-Stokes equations for a Newtonian fluid satisfy the preceding assumptions, and in addition we make the usual assumptions that the fluid is a perfect gas with $\gamma=1.4$, that the viscosity is given by Sutherland's law, that Stokes' hypothesis is valid, and that thermal conductivity is proportional to the viscosity.

Space Discretization

The convective fluxes are discretized with a TVD scheme based on Roe's scheme with second-order monotonic upstream schemes

Received Nov. 17, 1993; presented as Paper 94-0521 at the AIAA 32nd Aerospace Sciences Meeting, Reno, NV, Jan. 10–13, 1994; revision received March 31, 1994; accepted for publication April 1, 1994.

*Research Scientist, Department of Industrial Mechanics. Member AIAA.

for conservation laws (MUSCL) extrapolation.[9] Considering a system of nonlinear hyperbolic equations in one dimension, this scheme is briefly reviewed here. The extension to two or three dimensions is straightforward by applying the one-dimensional scheme independently in each coordinate direction.

For ease of notation we write $F(U)=\boldsymbol{F}_{\text{inv}}(U)\cdot\boldsymbol{n}$ and let $F_{i+1/2}$ denote the flux through the surface $\boldsymbol{S}_{i+1/2}$. By Roe's scheme, the numerical flux is then given by

$$F_{i+1/2} = (1/2)\,[F(U^-_{i+1/2}) + F(U^+_{i+1/2})]$$

$$-(1/2)\,(A^+_{i+1/2}-A^-_{i+1/2})(U^+_{i+1/2}-U^-_{i+1/2}) \tag{3}$$

where $U^-_{i+1/2}$ and $U^+_{i+1/2}$ are extrapolated values of the conserved variable at the left and right sides of the surface, and $A^{\pm}_{i+1/2}$ are given by

$$A^{\pm}_{i+1/2} = R_{i+1/2}\Lambda^{\pm}_{i+1/2}R^{-1}_{i+1/2} \tag{4}$$

where $R_{i+1/2}$ is the right eigenvector matrix of $A_{i+1/2}$, which is the Jacobian of F taken at $U_{i+1/2}$. The state $U_{i+1/2}$ is given by Roe's approximate Riemann solver defined by the equation

$$F(U^+_{i+1/2}) - F(U^-_{i+1/2}) = A(U_{i+1/2})(U^+_{i+1/2}-U^-_{i+1/2}) \tag{5}$$

where $A(U)$ is the Jacobian of $F(U)$. The diagonals $\Lambda^{\pm}_{i+1/2}$ are the split eigenvalue matrices corresponding to $R_{i+1/2}$. To avoid entropy violating solutions, the eigenvalues are split according to

$$\lambda^{\pm}_l = (1/2)\,(\lambda_l \pm \sqrt{\lambda_l^2+\varepsilon^2}) \tag{6}$$

where ε is a small parameter.

The first-order version of Roe's scheme is obtained by simply setting $U^-_{i+1/2}=U_i$ and $U^+_{i+1/2}=U_{i+1}$. Schemes of higher order are obtained by defining $U^-_{i+1/2}$ and $U^+_{i+1/2}$ by higher order extrapolation. The scheme used here is based on a second-order "minmod" limited extrapolation of the characteristic variables resulting in a scheme that is fully upwind.

The diffusive numerical fluxes are calculated using interpolation and central differencing of the conservative variables. By using the chain rule we can transform from primitive to conserved variables and from Cartesian to curvilinear coordinates, so the viscous flux can be expressed

$$\boldsymbol{F}_{\text{vis}}\cdot\boldsymbol{n} = \sum_l B_{n,\xi_l}\frac{\partial U}{\partial \xi_l} \tag{7}$$

with

$$B_{n,\xi_l} = \sum_k A_{n,x_k}\frac{\partial \xi_l}{\partial x_k} \tag{8}$$

In this form the derivatives $\partial/\partial\xi_l$ can be calculated directly on the computational mesh using interpolation and central differencing. By computing the various A_{n,x_k} from the interpolated value $U_{i+1/2}=(1/2)\,(U_i+U_{i+1})$, the viscous part of the numerical flux $F_{i+1/2}$ is then defined by Eqs. (7) and (8).

Finally, the thin-layer approximation has been used assuming that the solution varies predominantly along a single coordinate direction and ignoring the derivatives in the other directions. Denoting the only coordinate direction to be considered for ξ_{TL}, this amounts to setting B_{n,ξ_l} to zero for all values of ξ_l except ξ_{TL}, as well as setting the viscous fluxes to zero on surfaces other than those defined by constant ξ_{TL}. All Navier-Stokes calculations in this work have been carried out using the thin-layer approximation.

Time Integration

The steady-state solution to the discretized equations are obtained by implicit local time stepping. Since time accuracy is not required, several simplifications to the implicit operator can be made. First consider a fully implicit time discretization, again in one dimension, expressed in the so-called delta form:

$$\frac{V_i}{\Delta t}\Delta U_i + \Delta F_{i+1/2} - \Delta F_{i-1/2} = -(F^n_{i+1/2} - F^n_{i-1/2}) \tag{9}$$

where V_i is the volume of the grid cell i, and Δt the time step. Based on the first-order version of Roe's scheme, the flux vectors are approximately linearized, and the resulting linear system of equations is block septadiagonal of the form

$$C_0\Delta U_{i,j,k} + C_{-1}\Delta U_{i-1,j,k} + C_1\Delta U_{i+1,j,k}$$

$$+C_{-2}\Delta U_{i,j-1,k} + C_2\Delta U_{i,j+1,k}$$

$$+C_{-3}\Delta U_{i,j,k-1} + C_3\Delta U_{i,j,k+1} = RHS_{i,j,k} \tag{10}$$

where for the Navier-Stokes equations the various C are 5×5 matrices, and the unknowns $U_{i,j,k}$ and the right-hand side $RHS_{i,j,k}$ are vectors of size 5. The indices i, j, k of the various C have been dropped for clarity.

The preceding system is solved using a line Jacobi procedure that is a three-dimensional vectorizable extension of the conventional two-dimensional line Gauss-Seidel method.[10] Solving along all lines in a given coordinate direction simultaneously, the procedure can be vectorized across the tridiagonal systems yielding a vector length equal to the number of points in a computational plane. Assuming we have an approximate solution ΔU^m to Eq. (10), we can find a new approximate in three steps: First $\Delta U^{m+1/3}$ is found by solving along the i lines:

$$C_{-1}\Delta U^{m+1/3}_{i-1,j,k} + C_0\Delta U^{m+1/3}_{i,j,k} + C_1\Delta U^{m+1/3}_{i+1,j,k}$$

$$= RHS_{i,j,k} - C_{-2}\Delta U^m_{i,j-1,k} - C_2\Delta U^m_{i,j+1,k}$$

$$-C_{-3}\Delta U^m_{i,j,k-1} - C_3\Delta U^m_{i,j,k+1} \tag{11}$$

and then similar equations are solved for $\Delta U^{m+2/3}$ and ΔU^{m+1} along the j and k directions, always utilizing the latest available iterate of the solution on the right-hand side. A number of iterations can now be performed by repeating these three steps.

The somewhat high memory requirement of this method is controlled by solving blocks in a sequential manner, using the same work space. In a multiprocessor environment this can be achieved by mapping several blocks to each processor.

Stability and Convergence Rate

For the analysis of stability and convergence rate we consider the linear one-dimensional case, assuming exact solution of the equations within each block.

Applying the so-called matrix method, the difference equation is written in matrix form as $U^{n+1}=GU^{n+b}$, and the method is said to be stable if the solution to this equation is bounded. For increasing n on a fixed mesh size the stability criterion is that the spectral radius of G is less than or equal to unity.[11]

Since the usefulness of the implicit multiblock algorithm depends on the fact that the fast convergence of the implicit scheme is preserved, it is equally important to find out how the convergence rate of the split system compares with that of the unsplit system. The asymptotic rate of convergence is inversely proportional to the spectral radius, i.e., the largest of the moduli of the eigenvalues, of the iteration matrix.[11]

As a model equation we consider the scalar linear convection-diffusion equation:

$$\frac{\partial U}{\partial t} + a\frac{\partial U}{\partial x} - \mu\frac{\partial^2 U}{\partial x^2} = 0 \tag{12}$$

where both a and μ are nonnegative constants. Then we approximate this equation using evenly distributed grid points in accordance with the numerical scheme just described. To simplify the analysis, we consider only a first-order upwind scheme. For a mesh with N points we then get for $i=1,\ldots,N$,

$$\frac{U_i^{n+1}-U_i^n}{\Delta t}+a\,\frac{U_i^{n+1}-U_{i-1}^{n+1}}{\Delta x}$$

$$-\mu\,\frac{U_{i+1}^{n+1}-2U_i^{n+1}+U_{i-1}^{n+1}}{\Delta x^2}=0 \tag{13}$$

Imposing Dirichlet boundary conditions, this equation can be written in matrix form as

$$A_N U^{n+1}=U^n+b \tag{14}$$

where b represents the boundary conditions, and A_N is an $N\times N$ tridiagonal matrix given by

$$A_N = \text{tridiag}\,[-(\lambda+\gamma),\ 1+\lambda+2\gamma,\ -\gamma] \tag{15}$$

where $\lambda=a\Delta t/\Delta x$, i.e., the Courant-Friedrichs-Lewy (CFL) number, and $\gamma=\mu\Delta t/\Delta x^2=\lambda/R_\Delta$ where R_Δ is the mesh Reynolds number.

Employing the block Jacobi procedure is equivalent to splitting the matrix A_N into $A_N=B-C$ and writing

$$BU^{n+1}=(I+C)\,U^n+b \tag{16}$$

where

$$B=\begin{bmatrix} A_{N_1} & & & \\ & A_{N_2} & & \\ & & \ddots & \\ & & & A_{N_m} \end{bmatrix} \tag{17}$$

where m is the number of blocks, and N_k is the number of unknowns in block k with the obvious constraint $N_1+N_2+\ldots+N_m=N$; I is the identity matrix, and C is the matrix containing the explicit coupling between the blocks. The matrix C is thus a tridiagonal matrix with the form

$$C=\begin{bmatrix} & \gamma & & \\ \lambda+\gamma & & \gamma & \\ & \lambda+\gamma & \ddots & \\ & & \ddots & \gamma \\ & & \lambda+\gamma & \end{bmatrix} \tag{18}$$

where the only nonzero elements are located on the upper and lower diagonals. Rewriting Eq. (16) in the desired form, the iteration matrix G is given by

$$G=B^{-1}(I+C) \tag{19}$$

For the purely hyperbolic case, given by $\mu=0$, the eigenvalues of G can be found analytically by simple considerations and are equal $1/(1+\lambda)$ independent of how many blocks are used. For the mixed hyperbolic and parabolic case, given by $\mu\neq 0$, we have calculated the eigenvalues of G numerically for a few sample cases. A system with eight unknowns was considered, since for larger systems numerical errors were evident in the hyperbolic case when comparing the obtained results with the analytical solution.

In Fig. 1 the spectral radius for a system with two blocks is compared with that of the single-block case for a range of CFL and mesh Reynolds numbers. For a single block the results are as expected, showing a smaller spectral radius and therefore faster convergence for increasing CFL numbers throughout the range of mesh Reynolds numbers. In the multiblock case the spectral radius is close to unity for mesh Reynolds numbers less then one, indicating a convergence rate no better than for the equivalent explicit scheme. However, for increasing mesh Reynolds numbers, the spectral radius for the split systems decreases, and for $R_\Delta=10^6$ there is almost no difference between the split and unsplit systems.

As can be seen, also for the split system, the spectral radius is always less than or equal to one. This indicates that also for the mixed hyperbolic and parabolic case the multiblock method is unconditionally stable.

The significantly different behavior of the purely hyperbolic systems and systems with a strong parabolic influence is not surprising. In the purely hyperbolic case, with an infinite time step, the multiblock method would converge in a number of timesteps equal to the number of blocks. This is because information travels from left to right, so that the leftmost block would converge in the first time step, the second leftmost block in the second time step, and so on. For a mixed system, however, information travels both

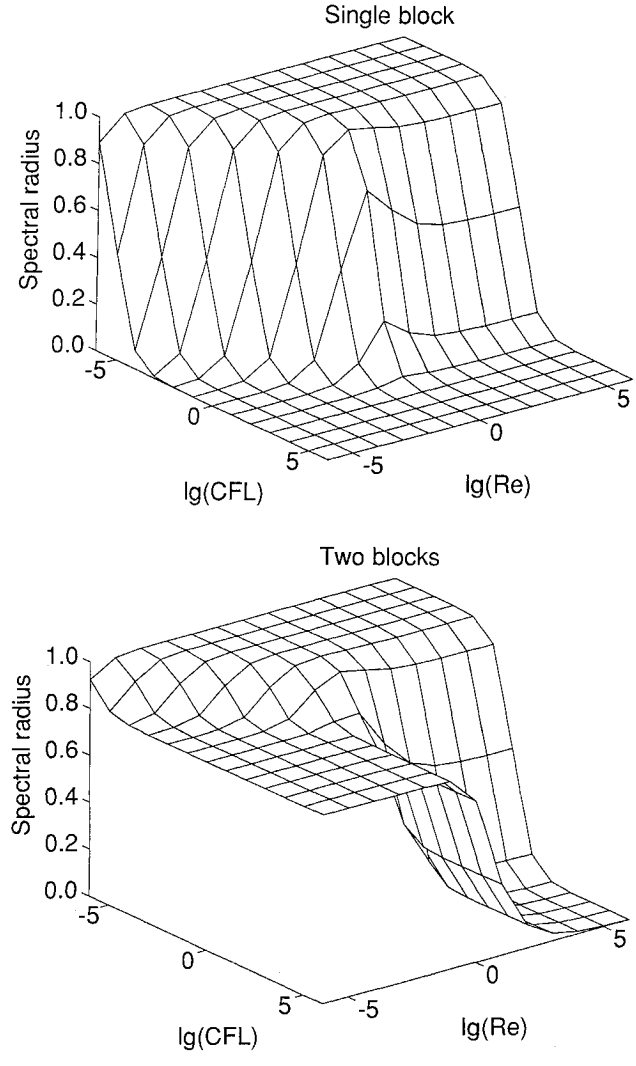

Fig. 1 Spectral radius of the iteration matrix for a mixed hyperbolic and parabolic system with eight unknowns using one or two blocks.

ways, and even for a splitting into just two blocks, the exact steady-state solution cannot in general be obtained in a finite number of time steps.

For small mesh Reynolds numbers we are thus left with what seems like a paradox: One of the main reasons for using an implicit scheme is often to be able to take long time steps in regions with viscous boundary layers. Here the stability criterion severely limits the time step of explicit schemes because of the clustering of grid points. But the clustering of grid points in the viscous boundary layer is exactly what causes a low-mesh Reynolds number and reduces the convergence rate of the implicit multiblock scheme to that of explicit schemes.

In the next section we will demonstrate that this problem can easily be circumvented as long as viscous effects are confined mainly in a boundary layer.

Test Cases

In the following the method is tested with practical calculations of realistic three-dimensional flowfields. To save CPU time and to enable single-block calculations, most of our computations have been on relatively coarse meshes. However, to confirm that the results carry over to larger problems, as well as to validate the correctness of the code, at least one fine mesh computation has been done for each case.

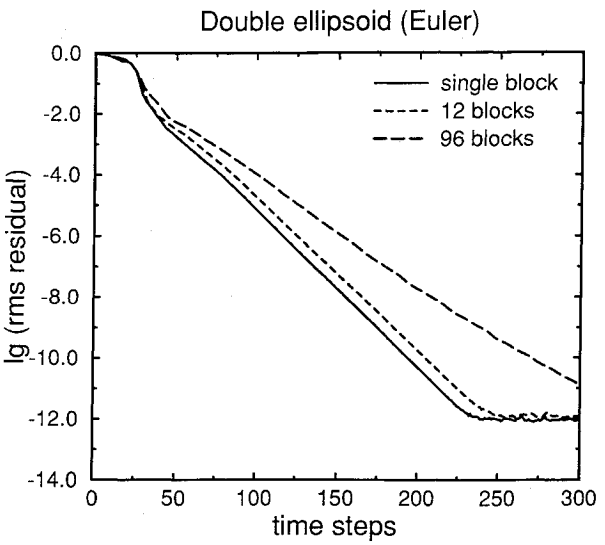

Fig. 2 Convergence histories for the solution of the Euler equations for the flow around the double ellipsoid.

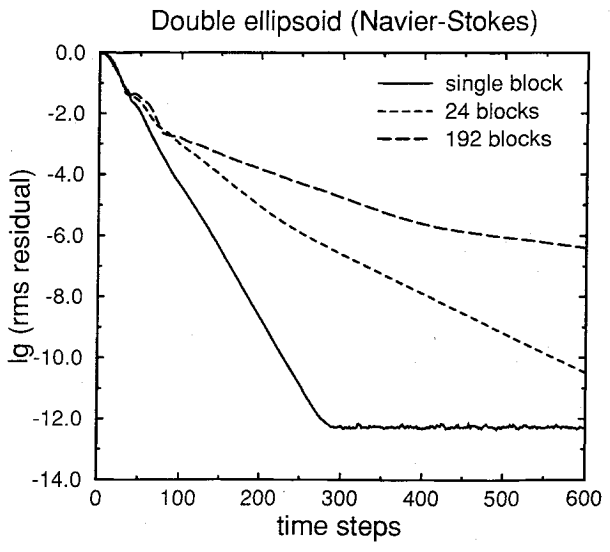

Fig. 3 Convergence histories for the solution of the Navier-Stokes equations around the double ellipsoid using blocks that are cubic in computational space.

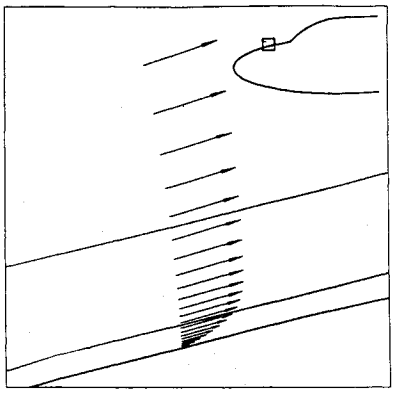

Fig. 4 Sample block interfaces using blocks that are cubic in computational space to solve the Navier-Stokes equations.

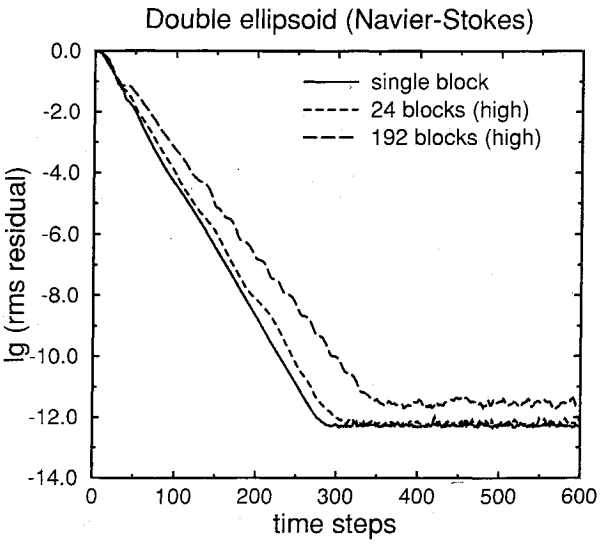

Fig. 5 Convergence histories for the solution of the Navier-Stokes equations around the double ellipsoid using high blocks.

Hypersonic Flow over the Double Ellipsoid

First we present results obtained from calculations of the flow over the double ellipsoid used at the Antibes workshop on hypersonic flows[12] as a model of a re-entry vehicle forebody. We have calculated the standard test case with Mach 8.15 and a 30-deg angle of attack.

To be able to compare the results with the results of the analysis of the model problem, the method was applied to the Euler equations as well as the Navier-Stokes equations. In all tests the initial conditions were obtained from a solution on a coarser mesh with half the number of points in each coordinate direction. The CFL number was initially set to 0.01 and increased with a constant factor each time step to reach 1000 in 50 steps, after which it was kept constant.

For the Euler equations, where clustering of mesh points close to the boundary is not needed, the mesh consisted of $24 \times 16 \times 16$ cells in the streamwise, normal, and circumferential directions. Test runs were carried out using 12 and 96 blocks consisting of $8 \times 8 \times 8$ and $4 \times 4 \times 4$ points, respectively. The convergence histories, based on the L_2 norm of the residual, for the aforementioned test runs and for the single-block case are given in Fig. 2. As can be seen, the use of 12 blocks gives a convergence almost identical to using a single block, whereas using 96 blocks gives just slightly slower convergence.

Returning to the Navier-Stokes equations, the same block sizes were tested corresponding to 24 and 192 blocks on a clustered

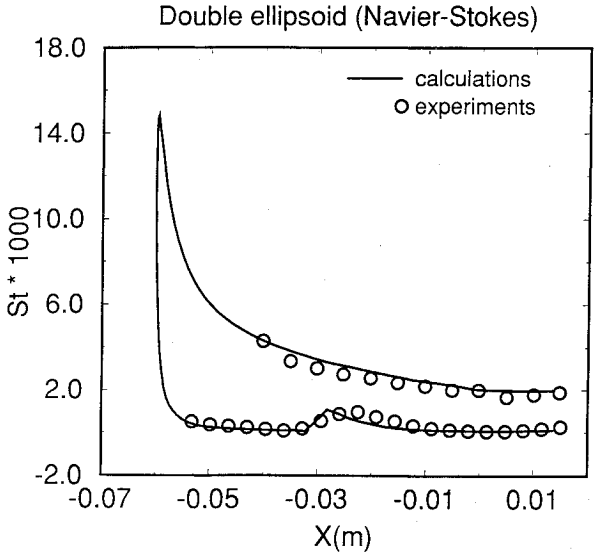

Fig. 6 Surface distribution of the Stanton number in the symmetry plane of the double ellipsoid.

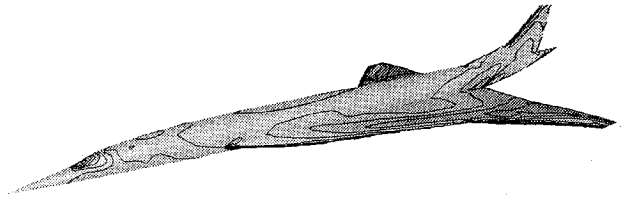

Fig. 7 Alliance geometry with computed Mach number contours.

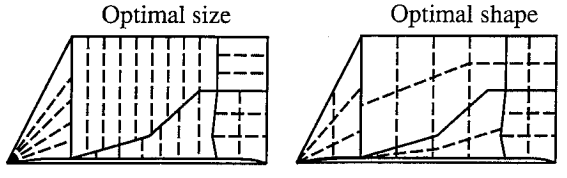

Fig. 8 Schematic view of different block partitionings based on a given multiblock mesh.

$24 \times 32 \times 16$ mesh. The convergence histories are given in Fig. 3, and we can see that for both 24 and 192 blocks the solution is converging significantly slower than for the single-block case, with the use of 192 blocks resulting in a convergence rate that can only be described as catastrophic. These results are in surprisingly good agreement with the theoretical predictions. Referring to Fig. 1, we were to expect only a very slight deterioration of the convergence rate when using the multiblock procedure on a hyperbolic equation, whereas for the Navier-Stokes equations the linear analysis indicates a decrease of the convergence rate with decreasing mesh Reynolds number. As can be seen from Fig. 4, the use of blocks with size $8 \times 8 \times 8$ for the Navier-Stokes equations produces block interfaces located in the outer part of the boundary layer. In this region we expect a strong viscous contribution to the equations, and the mesh is clustered, resulting in a low mesh Reynolds number. For blocks of size $4 \times 4 \times 4$, block interfaces will be located well inside the boundary layer where viscous forces are more dominating and the mesh even more clustered. Thus in this case we should expect an even lower convergence rate, which is indeed what we see in Fig. 3.

Now, to circumvent this problem without resorting to larger blocks, we can use a different block partitioning with blocks thicker than the viscous boundary layer. Again consider the use of 24 or 192 blocks, but this time with sizes $8 \times 16 \times 4$ and $2 \times 16 \times 2$, respectively. The convergence histories using these block partitionings are given in Fig. 5 and clearly demonstrate how the convergence rate is sensitive to how the boundary layer is split. We notice that with this shape of the blocks, we have obtained practically the same convergence rate using 24 blocks as for a single block, and even for 192 blocks we have obtained a convergence rate that is almost as good as in the single-block case.

As code validation we now present results obtained for the same flow case on a finer mesh consisting of $48 \times 64 \times 32$ cells. Using 24 blocks with size $16 \times 32 \times 8$, a six decade reduction of the residual was reached after 242 time steps or about 1 h and 25 min of CPU time on one processor on a Cray X-MP. In Fig. 6, we show the Stanton number on the body in the symmetry plane. The present results are compared with wind-tunnel measurements.[13] As we can see, the computations agree well with the experiments.

Supersonic Flow over the Alliance

The Alliance is a supersonic aircraft design currently being studied by Aerospatiale and British Aerospace. Being a successor of the well-known Concorde, it will, if built, carry from 250 to 300 passengers at Mach 2 up to 8800 km. The inviscid flow around a wing body tail configuration of the Alliance seen in Fig. 7 has been calculated for Mach 2 and a 4-deg angle of attack.

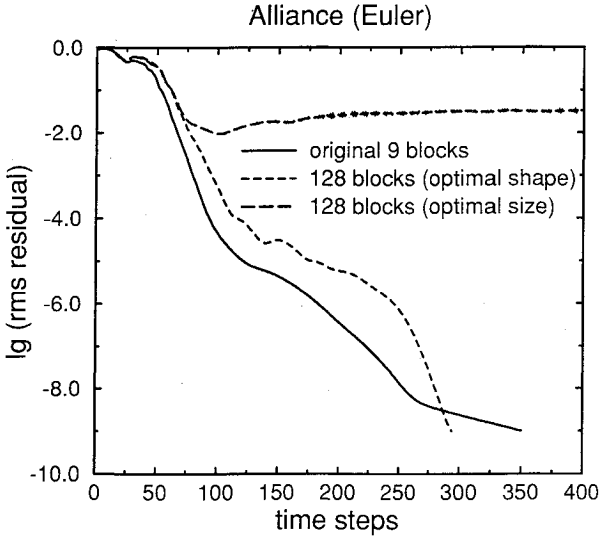

Fig. 9 Convergence histories for the solution of the Euler equations over the Alliance using different block partitionings.

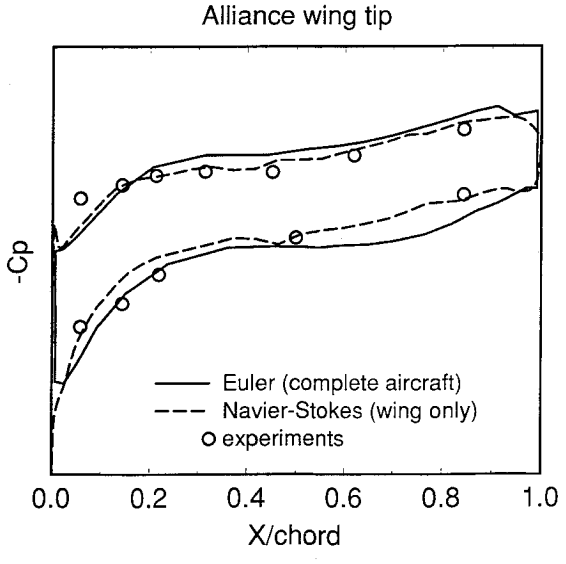

Fig. 10 Surface distribution of the pressure coefficient on the Alliance wing at a chordwise cut.

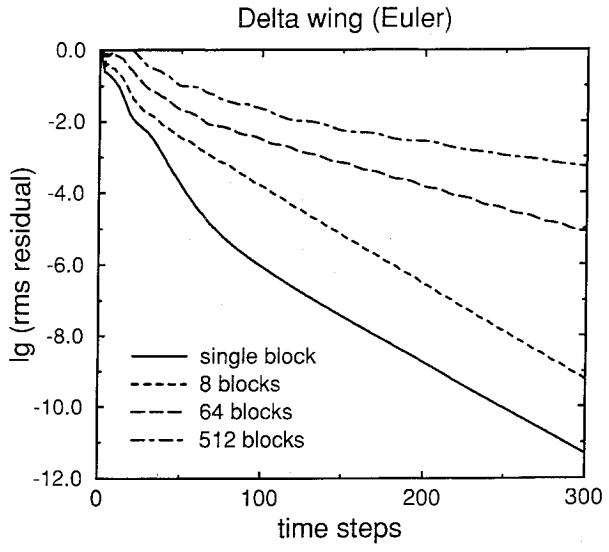

Fig. 11 Convergence histories for the solutions of the Euler equations over a delta wing using the block Jacobi procedure.

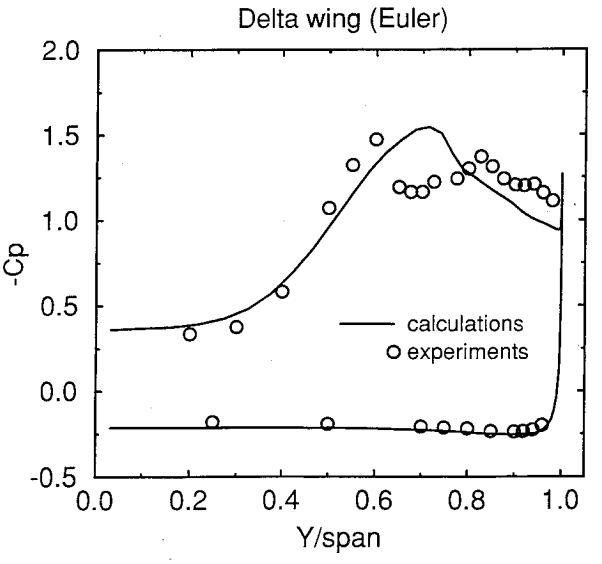

Fig. 13 Surface distribution of the pressure coefficient on the delta wing at a spanwise cut at 80% of the root chord.

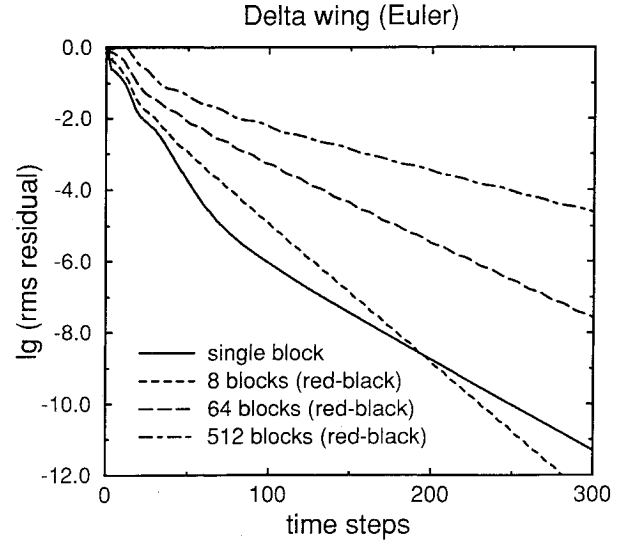

Fig. 12 Convergence histories for the solutions of the Euler equations over a delta wing using a red-black block Gauss-Seidel procedure.

As the flow is supersonic in the complete computational domain, this represents an interesting test for the block Jacobi method. In this case all information in the main flow direction is propagated from left to right, and we can therefore expect to see clearly the effect that increasing the number of blocks has on the hyperbolic part of the equations.

The mesh used was provided by Aerospatiale, containing nine blocks with a total of 313,994 points on one side of the symmetry plane. Here we were faced with the nontrivial problem of splitting an existing multiblock mesh into a larger number of blocks. Two different block topologies were generated, each consisting of 128 blocks. In the first topology, the mesh was partitioned to achieve load balance on a parallel computer, resulting in nearly the same number of points in each block. As indicated in Fig. 8, optimizing only the size of the blocks runs the risk of creating excessively thin blocks. The second block topology was therefore generated by optimizing the size of the sides of the blocks, resulting in blocks that were close to cubic in computational space. As it turns out, this also results in blocks with approximately the same size.

As before, the CFL number was initially set to 0.01 and increased with a constant factor each time step to reach 1000 in 50 steps, after which it was kept constant. Freestream values were used as initial conditions.

From Fig. 9 we can see that the two different block partitionings result in drastically different convergence rates. Whereas for blocks with optimal shape there is indeed very little difference from the convergence rate obtained with the original nine blocks, for blocks optimized only with respect to their size, the algorithm effectively breaks down. In the latter case several of the original blocks were split into blocks with just two cells in one of the coordinate directions.

As a consequence, algorithms for creating blocks to be calculated in parallel must take into account the shape of the blocks to retain as much as possible the fast information transport of the implicit scheme. This requirement comes in addition to minimizing work load and communication, which are the most important factors for explicit schemes.

Validation results showing the pressure coefficient at a chordwise cut close to the wing tip can be seen in Fig. 10. These results are based on a nine decade reduction of the residual, taking 3 h and 31 min on one processor of a Cray Y-MP using the original nine blocks. In addition to the Euler results we also show results from a Navier-Stokes calculation over the wing only. Here 40 blocks were used on a C mesh with 576,000 cells. As before the block interfaces parallel to the wall in the boundary layer were avoided. A six decade reduction of the residual was obtained after 256 time steps taking approximately 4 h, 30 min on one Y-MP processor. The calculated results are compared with experiments provided by Aerospatiale. Although the numerical values on these curves are withheld at the request of Aerospatiale, we can once again see that the computations agree well with the experiments.

Transonic Flow over a Delta Wing

The final test case is the transonic flow over a delta wing at Mach 0.85 and a 20-deg angle of attack. The wing geometry for this well-known test case is defined by a delta wing with NACA 64A005 chord sections having a 65-deg sweep and cropped at 85% of the root chord.

As this case is more strongly elliptic than the preceding supersonic cases, we can expect a stronger deterioration of the convergence rate even for the Euler equations as the number of blocks is increased.

This is confirmed by Fig. 11 showing convergence histories using 1, 8, 64, or 512 blocks. The mesh used was made up of $24 \times 16 \times 32$ cells in the streamwise, normal, and circumferential directions. Although convergence is just slightly slower in the 8 block case, the result for 64 blocks is quite discouraging while

using 512 blocks causes unacceptably slow convergence. It turns out that better results can be obtained using a red-black Gauss-Seidel procedure[14] rather than a Jacobi procedure as shown in Fig. 12. As we can see, using 64 blocks (which can be calculated in parallel on 32 processors) requires about twice as many time steps as the single-block method. In our opinion this is an acceptable result, considering that no extra work is done to solve the inter-domain coupling.

Various other block iterative Gauss-Seidel-type algorithms were also tested, but none of these resulted in a convergence rate that was significantly different from the one obtained using red-black ordering, nor did an attempt to enhance the information transport by using long and thin blocks parallel to the main flow direction. It is therefore our belief that to improve the convergence rate further an implicit coupling of the blocks has to be introduced.

Finally, in Fig. 13 we show the computed pressure coefficient on the surface at a cut at 80% of the chord length compared with wind-tunnel measurements.[15] These computations were carried out on a $64 \times 36 \times 80$ mesh using 64 blocks with red-black ordering. A six decade reduction of the residual was obtained after 426 time steps corresponding to 7 h and 45 min total parallel CPU time on a four-processor Cray 2. The computational results compare with the measurements as expected for an Euler solution that is able to capture a primary vortex induced by the sharp leading edge but not secondary vortices created by viscous effects.

Conclusion

We have shown that very efficient implicit multiblock calculations can be carried out even for a large number of blocks using a block Jacobi or block red-black Gauss-Seidel procedure. This shows that, contrary to what is often argued, implicit methods are well suited for massively parallel systems.

As we have seen, the efficiency of our method is strongly dependent on the block partitioning. For the Navier-Stokes equations it is of extreme importance that the height of the blocks is greater than the boundary-layer thickness. This will insure that the equations are not decoupled in regions with strong parabolic dominance. Also, excessively thin blocks will inhibit the fast information transport of implicit schemes and cause breakdown of the algorithm, at least for the high CFL numbers used in this work. As a consequence, when creating blocks to be computed in parallel, insuring fast convergence is just as important as achieving good load balance and low communication ratio.

For transonic computations, where the equations are more strongly globally coupled, the loss of convergence is more significant than for supersonic cases. In this case using a red-black Gauss-Seidel procedure, that still can be executed in parallel, gives a clear advantage over using the Jacobi method, even if this means that only half as many processors can be utilized. Although we feel that the loss of convergence that was observed in this case is acceptable as long as the number of blocks is not too high, this is clearly the case where there might be most to gain by using more sophisticated domain decomposition methods.

Since our solver is based on a rather expensive line relaxation procedure allowing for unlimited CFL numbers and convergence in relatively few time steps, it is reasonable to believe that our results carry over to the majority of implicit methods in practical use today.

Finally, it can be concluded that the block Jacobi or red-black Gauss-Seidel procedure with line relaxation within each block is a good alternative to conventional relaxation techniques. The method is straightforward to vectorize and parallelize, and the memory requirement is controlled by varying the number of blocks, making the solver portable to a wide range of computational platforms.

Acknowledgments

This work is based on the author's Ph.D. thesis at the Norwegian Institute of Technology supported by the Royal Norwegian Council for Scientific and Industrial Research and on research carried out while the author was employed by CERFACS (European Center for Research and Advanced Training in Scientific Computing), Toulouse, France. The author wishes to thank Aerospatiale for providing grid and experimental data for the Alliance.

References

[1]Chan, T., "Domain Decomposition Algorithms and Computational Fluid Dynamics," *The International Journal of Supercomputer Applications*, Vol. 2, No. 4, 1988, pp. 72–83.

[2]Keyes, D., "Domain Decomposition Methods for the Parallel Computation of Reacting Flows," *Computer Physics Communications*, Vol. 53, May 1989, pp. 181–200.

[3]Flores, J., "Simulation of Transonic Viscous Wing and Wing-Fuselage Flows Using Zonal Methods," *Proceedings of the First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1988, pp. 381–416.

[4]Walters, R., and Thomas, J., "A Patched-Grid Algorithm for Complex Aircraft Configurations," *Proceedings of the Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1990, pp. 397–409.

[5]Yadlin, Y., and Caughey, D., "Block Multigrid Implicit Solution of the Euler Equations of Compressible Fluid Flow," *AIAA Journal*, Vol. 29, No. 5, 1991, pp. 712–719.

[6]Sahu, J., and Steger, J., "Numerical Simulation of Three-Dimensional Transonic Flows," *International Journal for Numerical Methods in Fluids*, Vol. 10, No. 8, 1990, pp. 855–873.

[7]Schmatz, M., Monnoyer, F., and Wanie, K., "Numerical Simulation of Transonic Wing Flows Using a Zonal Euler, Boundary-Layer, Navier-Stokes Approach," *Zeitschrift Flugwissenschaften und Weltraumforschung*, Vol. 13, No. 6, 1989, pp. 377–384.

[8]Jenssen, C., "The Design and Implementation of an Implicit Multiblock Navier-Stokes Solver," Ph.D. Thesis, Dept. of Mechanical Engineering, Norwegian Inst. of Technology, Trondheim, Norway, 1992.

[9]Chakravarthy, S., "High Resolution Upwind Formulations for the Navier-Stokes Equations," *Proceedings of the von Kármán Institute for Fluid Dynamics Lecture Series 1988-05*, von Kármán Inst. for Fluid Dynamics, Brussels, Belgium, March 1988.

[10]MacCormack, R., "Current Status of Numerical Solutions of the Navier-Stokes Equations," AIAA Paper 85-0032, Jan. 1985.

[11]Smith, G., *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Oxford Univ. Press, Oxford, England, UK, 1985.

[12]*Proceedings of the Workshop on Hypersonic Flows for Reentry Problems* (Antibes, France), Pt. I, Institut National de Recherche en Informatique et en Automatique, Sophia Antipolis, France, Jan. 1990.

[13]Aymer, D., de Rocquefort, T. A., Carlomagno, G., and de Luca, L., "Experimental Study of Flow over a Double Ellipsoid," *Proceedings of the Workshop on Hypersonic Flows for Reentry Problems* (Antibes, France), Pt. I, Institut National de Recherche en Informatique et en Automatique, Sophia Antipolis, France, Jan. 1990.

[14]Golub, G., and Ortega, J., *Scientific Computing, An Introduction with Parallel Computing*, Academic Press, New York, 1993.

[15]Hirdes, R., "US/European Vortex Flow Experiment: Test Report of Wind Tunnel Measurements on the 65° Wing in NLR High Speed Wind Tunnel HST," National Aerospace Laboratory, NLR TR 85046 L, Amsterdam, The Netherlands, 1985.