GMRES ACCELERATION OF ITERATIVE IMPLICIT FINITE ELEMENT SOLVERS FOR COMPRESSIBLE EULER AND NAVIER–STOKES EQUATIONS

RÉMI CHOQUET

IRISA, Campus de Beaulieu, F-35042 Rennes, France

AND

PÉNÉLOPE LEYLAND AND TIANA TEFY

IMHEF, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland

SUMMARY

Implicit iterative schemes based on linearized and non-linear Newton methods are discussed, with resolution of a matrix subsystem or a matrix-free method by preconditioned GMRES algorithms. The defaults of convergence due to the locality of Newton algorithms can be partially overcome by using stabilizing descent techniques, restarting and global strategies such as line search backtracking procedures, or by tuning the iterations once the approximate Jacobians are closer to the exact ones. Comparison with a more conventional relaxation method and their implementation on parallel architectures are discussed.

KEY WORDS Newton; GMRES; descent direction; local preconditioner; dynamic sizing

1. INTRODUCTION

Implicit time-stepping schemes for steady state or unsteady computations of compressible flows on unstructured finite-element-type meshes lead to sparse matrices. The feasibility of direct methods for computations over complex geometries with a reasonable number of discretization nodes become rapidly prohibitive in memory. Point-by-point iterative methods are thus more usually employed. However, the global convergence properties and robustness of such methods are not always satisfactory. The local character of such schemes can lead to non-converging solutions in critical zones where the global nature of the flow field variables would be necessary. Convergence acceleration techniques such as multigrid can be an advantage in such cases, as can be other preconditioning techniques such as equation preconditioning or matrix preconditioning. As for many gradient techniques, a combination of domain decomposition and preconditioning is an important issue.

Non-linear Newton–GMRES algorithms can be seen as residual least squares problems for which descent techniques are stabilizing. Indeed, Newton iterations that are solved approximately by iterative methods (*inexact Newton methods*) require that the current iterate be a descent direction for the quadratic form associated with the system. Non-linear algorithms for non-linear systems of equations can be of two types. The first relies on an inner algorithm where a system of equations of 'the Jacobian matrix of the system times the vector iterate equals the previous residual' is to be solved (iteratively, directly), while the second involves truly non-linear approaches implying fixed point algorithms,

CCC 0271–2091/95/090957–11

descent methods, non-linear conjugate gradient iterations, etc. For the applications presented in this paper, only algorithms of the first type have been investigated. The eventual non-differentiability of the numerical method's Jacobians can be a drawback for the two types.

Linearized Newton methods have been extensively used in computational fluid dynamics for the last 12 years. In this case only an approximate linearized Jacobian is necessary together with a linear subsystem to be solved by either direct or iterative methods. This method is often used in so-called point implicit methods where the implicit scheme is written per time step per point, resulting in localized subsystems of the size of a maximum (*no neighbour nodes*) × (*n variables* × *n variables*). The major defaults of such methods are the *local* structure combined with the default of locality of Newton methods in themselves.

Krylov subspace methods minimize the inner iterate residuals on a sequence of subspaces of reduced dimension with respect to the total number of iterates. In this paper GMRES (generalized minimal residual) Krylov subspace methods are considered. Other projection methods such as Arnoldi's are also interesting.

In order to avoid initially high Krylov dimensions within the GMRES kernel, restarting procedures are often employed. However, a trade-off must then be made between the number of restarts (and thus a certain loss of precision) and convergence acceleration. The solution obtained by GMRES with a restarting procedure is shown to be a descent direction for the non-linear function which represents the system. Although the Newton method converges only locally, algorithms such as a line search backtracking strategy improve its global convergence. The combination of Newton, GMRES and line search backtracking proves to be a valuable convergence tool.

Let the discretized Navier–Stokes system of equations be written in the following symbolic form: find v_n at time t_n such that for given v_{n-1} and Δt, where $t_n = t_{n-1} + \Delta t$,

$$F(v_n, v_{n-1}, \Delta t) = 0. \tag{1}$$

Newton–GMRES algorithms present essentially two kinds of instabilities. The first is due to the fact that Newton methods do not necessarily converge to a global solution. Secondly, the sequence formed by $F^{\mathrm{T}}(v_n)F(v_n)$ is not always a monotone decreasing sequence, i.e.

$$F^{\mathrm{T}}(v_{n+1})F(v_{n+1}) \leq F^{\mathrm{T}}(v_n)F(v_n), \quad \forall n. \tag{1}$$

The convergence of the Newton–GMRES algorithm for such systems depends on the fact that the GMRES iterate is a descent direction for $f(v) = F^{\mathrm{T}}(v)F(v)$. In particular, in Reference 1 this property is shown in the case where the solution is initialized by a zero vector. This result has been extended to the case where the Newton–GMRES algorithm also presents several restarting procedures and to matrix-free approaches in Reference 2.

In Section 2 the governing equations and their discretization are discussed. The linear and non-linear Newton–GMRES algorithms used are presented in Section 3. In Section 4 we discuss the possibilities of preconditioning and stabilizing initialization procedures: Newton methods only converge locally. The initial phase of convergence can require an excessive number of iterations in order to approach this condition. Finally some results for transonic and hypersonic viscous flows and implementation considerations are given.

2. GOVERNING EQUATIONS AND DISCRETIZATION

The Navier–Stokes equations for a non-reactive compressible flow can be written in conservative form as

$$W_t + F_1(W)_x + F_2(W)_y = \frac{1}{Re}(R(W)_x + S(W)_y), \tag{2}$$

where W is the state vector, which for a perfect gas in two dimensions is $W = (\rho, \rho u, \rho v, \rho E)^{\mathrm{T}}$, with ρ, u and v the density and the velocity components of the flow field respectively. The total energy E is defined as

$$E = e + \tfrac{1}{2}(u^2 + v^2 + w^2).$$

F_{k} are the convective fluxes:

$$F_1 = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(\rho E + p) \end{pmatrix}, \qquad F_2 = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho E + p) \end{pmatrix}.$$

Re is the local Reynolds number and (R, S) are the viscous fluxes. The Euler equations are obtained by putting $1/Re$ to zero. The whole system is of mixed hyperbolic/parabolic type, with the dominant type depending highly on the flow characteristics, which can be very localized, particularly for complex viscous flows where several different regimes are present. This means that the mathematical analysis is also local and produces disparate conditioning depending upon the *local* predominant flow structure.

For reactive flow the equations are augmented in dimension by additional continuity equations for the partial densities of the species, with source terms corresponding to the species net production rate by chemical reactions.

Another way of writing the governing equations is to use entropy variables. This automatically enforces the *entropy-preserving* condition for the hyperbolic part of the equations (second law of thermodynamics). The equations then become symmetric:

$$A_0 V_{,t} + A_i V_{,i} = (K_{ij} V_{,j})_{,i} + \mathbf{F}, \tag{3}$$

where $A_0 = W_{,V} = \partial W/\partial V$ is symmetric and positive definite, $A_i = F_{i,i}A_0$ is symmetric and $K_{ij} = (K^u_{ij} + K^h_{ij})A_0$ is symmetric and positive definite.

This formulation has been adopted by Hughes and co-workers.[3,4] The discretization is obtained by a Petrov–Galerkin finite element formulation where the test functions are upwinded in the downstream direction, introducing viscosity in the direction of gradients (SUPG). This approach has been adopted for implementation on connection machine architectures using a node/element data structure. For the conservative approach, equation (2), a hybrid finite element (FE)/finite volume (FV) method is used where the convective fluxes are evaluated over the geometrical dual space of control volumes of the finite element simplexes, allowing robust numerical flux functions coming from purely finite volume methods to be formulated as well as conserving the finite element approximation precision. Both formulations give $(p + \frac{1}{2})$-order convergence,[5–7] where p is the order of the FE approximation. The viscous terms are evaluated on the finite element simplexes in a standard finite element way depending on the degree of approximation.

The domain Ω is approximated by a mesh Ω_{h} of triangular elements with P^1 test functions ϕ_{h}.

Both finite element discretizations of the equations at time $t_n = n\Delta t$ can be written as

$$\int_\Omega h \frac{W^{n+1} - W^n}{\Delta t} \phi_h \mathrm{d}\Omega = \sum_{T \in \Omega_h} \int_T \mathscr{F}(W^{n+1}) \nabla \phi_h \mathrm{d}T = F(W^{n+1}, W^n, \Delta t) = 0 \tag{4}$$

for all test functions φ_{h} in the chosen finite element discretization.

For a discretization point i and for j a neighbouring point of i the conservative formulation of (4) thus becomes

$$\begin{aligned} 0 = M_i \frac{W_i^{n+1} - W_i^n}{\Delta t_i} &+ \int_{\partial C_i} (\vec{F}_1(W_{IJ}^{n+1}) \cdot \vec{n}_{ij} + \vec{F}_2(W_{ij}^{n+1}) \cdot \vec{n}_{ij} \mathrm{d}\sigma \\ &+ 1 \int T_i(R(W^{n+1}, \nabla W^{n+1}) + S(W^{n+1}, \nabla W^{n+1}))\mathrm{d}T^i. \end{aligned} \tag{5}$$

Here M_i denotes the lumped mass matrix and the second term is evaluated by the convective numerical flux function $\Phi(W_i, W_j, \vec{n}_{ij})$ along the direction $\vec{n}_{ij}$; ∂C_i denotes the boundary of the dual cell associated with i.[5,8]

3. NEWTON–GMRES ALGORITHMS

To solve the non-linear system (4) with an implicit time-stepping procedure, two Newton-type algorithms have been explored.

The first is a linearized Newton algorithm. Following (5), the Taylor series expansion for the fluxes around the solution at time t_n, W^n, gives

$$\mathcal{F}(W^{n+1}) = \mathcal{F}(W^n) + \frac{\partial\mathcal{F}}{\partial W}(W^n)(W^{n+1} - W^n) + o((W^{n+1} - W^n)^2).$$

We introduce the notation $\delta W^n = W^{n+1} - W^n$. Then the resolution of the system (5) leads to the resolution of

$$\frac{M_i}{\Delta t_i}\delta W_i^n + \frac{\partial\mathcal{F}}{\partial W}(W^n)\delta W^n = -\mathcal{F}(W^n). \tag{6}$$

The Jacobian matrices J of $\mathcal{F}$ are approximated either by linearization, or in the case of a differentiable numerical flux by an exact Jacobian, or finally by finite difference approximations of the product $J(u)\delta u$.[1,2] (See also in the non-linear method.) Both the linearized and exact Jacobian methods have been implemented. In the first case the linearized Jacobians for the convective fluxes are approximated by a Q-scheme related to the characteristic decomposition, where the term

$$\frac{d\mathcal{F}}{dW}\cdot n_{ij} = \frac{\partial F}{\partial W}n_{ij} + \frac{\partial G}{\partial W}n_{ij} = T_{ij}^{-1}\Lambda_{ij}T_{ij}. \tag{7}$$

Λ_{ij} denotes the eigenvalue matrix and T_{ij} the transformation matrix of right eigenvectors in the direction n_{ij}. In the second case an Osher approximate Riemann numerical flux has been chosen since it is differentiable in most situations. The flux interface evaluation of the second term in (5) is approximated by Φ_{Osher} $(WW_i^n, W_j^n; \vec{v}_{ij})$; thus

$$\int_{\partial C_i\cap\partial C_j}\left(\frac{\partial F}{\partial W}(W^n)\cdot\vec{v}_{ij}\right)\delta W^n \mathrm{d}\sigma \approx \frac{1}{2}\left(\frac{\partial\Phi}{\partial W_i}(W_i^n, W_j^n; \vec{v}_{ij})\delta W_i^n + \frac{\partial\Phi}{\partial W_j}(W_i^n, W_j^n; \vec{v}_{ij})\delta W_j^n\right).$$

A pseudo-Newton time-stepping procedure can be formulated as follows:

```
let $u^0 = u^n \equiv W^n$
    Calculate local time step $\Delta t_i$(steady stateflow)
    For $n = 1, \dots,$ inewtonmax
        Calculate the residual $-\mathcal{F}(u^n)(6)$
        Evaluate $J = \frac{1}{\Delta t} + \frac{\mathrm{d}\mathcal{F}(u^n)}{\mathrm{d}u}|_{u^n_{in}}$
        Solve $J\delta u = -\mathcal{F}(u^n)$
    $u^{n+1} = \delta u^n + u^n$
$u^{n+1} \Rightarrow u^n$
```

The resolution of the linear system is performed by either a relaxation method such as point/block Jacobi or by a diagonally preconditioned GMRES algorithm. The Jacobians are evaluated at each node point and consist of blocks of 4×4 per node for a perfect gas and 9×9 for modelization of a five-species reacting gas (air).[8] The above algorithm can be looked at as a pseudo-Newton method when $n \rightarrow \infty$.

The second method is known as the non-linear Newton–GMRES algorithm. At each Newton iteration a resolution of the subsystem is solved by Krylov subspace methods. These latter methods look for the approximate solution to the system of equations

$$\min_{z \in \mathscr{K}} b - J(u_0 + z)\|$$

in the Krylov space $\mathscr{K}$ of dimension $k \leq N$ (where N denotes the size of the problem) associated with the initial residual $r_0 = b - Ju_0$ and the matrix J,

$$\begin{aligned} \mathscr{K} &= \operatorname{span}\{r_0, Jr_0, J^2 r_0, \dots, J^{k_{\max}-1} r_0\} \\ &= \operatorname{span}\{u_1, u_2, \dots, u_{k_{\max}}\}, \end{aligned}$$

which corresponds to solving an over determined system of equations in the least squares sense. The method is efficient if there exists a $k_{\max} \ll N$. If a sufficient approximation cannot be reached within a predetermined dimension of the Krylov space, the algorithm is *restarted* with the new approximate solution. The algorithm combining Newton, GMRES and line search backtracking for the resolution of problem (4) can be summarized as follows:

```
$u_0$ given
for $m = 1, \dots$
    Solve $J(u^m)\delta u^m = -F(u^m)$
    $u^{m+1} = u^m + \alpha\delta u^m$
$\alpha \in [0, 1]$
$\alpha$ is calculated by line search backtracking
to decrease $f(u) = \frac{1}{2}F^t(u)F(u)$
```

Newton methods have at least local quadratic convergence. However, their global convergence is not guaranteed. The backtracking technique is to ensure that the quadratic functional $F^{\mathrm{T}}F$ is a monotone decreasing one, i.e.

$$F^{\mathrm{T}}(v_{n+1})F(v_{n+1}) \leq F^{\mathrm{T}}(v_n)F(v_n), \quad \forall n. \qquad (8).$$

Hence it is natural to consider descent methods to enforce this condition (8), as proposed in Reference 1. Indeed, the above algorithm is efficient if the GMRES iterate δu^m is a descent direction at u^{m+1}, i.e.

$$F^{\mathrm{T}}J\delta u^m < 0; \text{then } \exists\, \alpha \text{ such that } (F^{\mathrm{T}}F)(u + \alpha\delta u) < (F^{\mathrm{T}}F)(u).$$

Note that $\nabla f(u) = J^{\mathrm{T}}(u)F(u)$, so p is a descent direction for f at u if $F^{\mathrm{T}}(u)J(u)p < 0$. It is difficult to evaluate *a priori* an optimal value of a maximal Krylov dimension given a certain stopping criterion. Also, the memory storage required for the above method depends linearly on the size of this maximal Krylov dimension k. However, a small value of k will not lead to a sufficiently accurate solution. A restarting procedure is thus introduced fixing a certain $k_{\max}$, which in general is small. If the initial estimate of $u_{k_{\max}}$ is insufficient, the algorithm is restarted initializing with this last vector. Several stages of restarting can be introduced to avoid increasing the dimension $k_{\max}$.

```
$u$ given, $R = F(u), J = \frac{\partial F}{\partial W}(u)$
$\delta u = 0$
FOR $l = 1, \ldots, l_{\max}$DO
    solve $-J\delta u = R$ by GMRES
    $V_m$ is an orthogonal subspace of $\mathcal{K}$
    $u = u + V_m y_m$
```

In Reference 2 the GMRES iterate is shown to be again a descent direction when there is *restarting*, making non-linear Newton–GMRES a promising algorithm to solve complex systems such as (4).

A particular advantage of GMRES is that the actual Jacobian matrix J does not need to be assembled explicitly; the algorithm only requires matrix–vector products. It is thus possible to conceive '*matrix-free*' Newton–GMRES algorithms with a simplified finite difference approximation of the product $J(u)\delta u$:

$$J(W)p \approx \frac{F(W + \varepsilon p) - F(W)}{\varepsilon},$$

where ε is a small scalar estimated by the consistency and the cut-off errors.[9,10]

4. PRECONDITIONING, RESTARTING AND DYNAMIC SIZING

All the low situations presented here have a convection dominance present somewhere within the field. The convergence of linear Newton implicit schemes often suffers from a certain stagnation of the residuals within a certain tolerance, beyond which convergence no longer proceeds. The more complex the flow and the higher the number of discretization nodes, the more this stagnation effect is evident. It is thus necessary to precondition the Jacobian when using gradient methods to ensure convergence. Two preconditioners are used here, namely diagonal preconditioning and an ILU-type per block. The latter is not possible for matrix-free methods, where right preconditioning has been adopted. In the latter case the step $J\delta u = -F$ becomes $JB^{-1}B\delta u = -F$.

Newton convergence is at least linear if each subsystem is solved exactly. Satisfying such a requirement is impossible both because of finite precision computer arithmetic and because of the complexity of a non-symmetric linear system solver. The tolerance parameters introduced within the iterative stages must take account of these problems. Also, as explained above, the choice of $k_{\max}$ is not evident. The GMRES complexity increases as the square of $k_{\max}$ and an insufficient $k_{\max}$ will not approach the solution sufficiently, thus leading to poor convergence. This problem increases with increasing problem size. Restarting and low $k_{\max}$ will also lead to loss of information. A dynamic strategy is used to optimize k and the stopping criterion.[2]

If r^m denotes the non-linear residual $\{-F(u^m)\}$ and ε_n denotes the ending criterion for Newton, assuming a linear convergence for r^m, the number of iterations for convergence can be estimated by

$$\frac{\log(\|r^m\|/\|r^{m-1}\|)}{\log(\varepsilon_n)}. \tag{9}$$

Then $k_{\max}$ and ε_n are updated to give bounds for criteria.

Let $r^{(j)}$ be the residual after j restarts of GMRES. We can define a similar criterion to (9) for $k_{\max}$ by

$$\frac{\log(\|r^{(j)}\|/\|r^{(j-1)}\|)}{\log(\varepsilon/\|r^{(0)}\|)}. \tag{10}$$

According to this number, the sizes of $k_{\max}$ and ε_n are updated to give bounds for these criteria. A dynamic strategy to update $k_{\max}$ and ε is then established. In Figure 1 the Krylov dimension tends to a constant and the dimension of the maximal number of restarts tends to unity once convergence is established. This property is conserved upon going to a mesh with almost five times as many points (Figure 1, right). This ensures a higher performance ratio.

As introduced above, the Newton–GMRES algorithm with restarting can be further stabilized by using a line search backtracking technique which enforces condition (8) and thus global convergence (Figure 2).

5. RESULTS AND PERFORMANCE ANALYSIS

Results are given for a comparison of a point Jacobi iterative scheme versus a preconditioned GMRES algorithm, both on the CM200, CM5 and on a CRAY YMP with the various constructions for the implicit matrix as described in detail above. On both machines the results were comparable for standard Euler calculations. On the CM200 GMRES took longer because of the scalar product operation being slower, whereas on the YMP both algorithms were comparable in performance and CPU time.[11]

The choice of $k_{\max}$ is dependent upon the problem size. For small meshes (less than 3000 nodes) a value of $k_{\max} = 10$ was sufficient for the convergence of linear Newton methods. For more complex problems with a large number of discretization points ($\approx 25{,}000$) values of $k_{\max}$ 25–40 were necessary. Values of $k_{\max} = 25$–50 were required for the non-linear scheme (but with an overall reduced number of iterations). The number of restarting loops should be restricted as much as possible to avoid growth of errors; the balance $k_{\max}/l_{\max}$ is avoided by the dynamic strategy proposed above.

An interesting comparison also reveals the dependence on the CFL number for steady state calculations when using the relaxation method. The GMRES algorithm is less dependent and allows for higher CFL numbers. This is particularly interesting for Navier–Stokes calculations, where the maximal value of CFL can be quite low. This allows for a considerable gain in CPU time.

For large-scale problems such as complex Navier–Stokes calculations, GMRES was found to provide valuable convergence acceleration, as the relaxation method tends to stagnate convergence after a certain limit (Figures 3 and 4). However, in these cases the Newton–GMRES algorithm proved

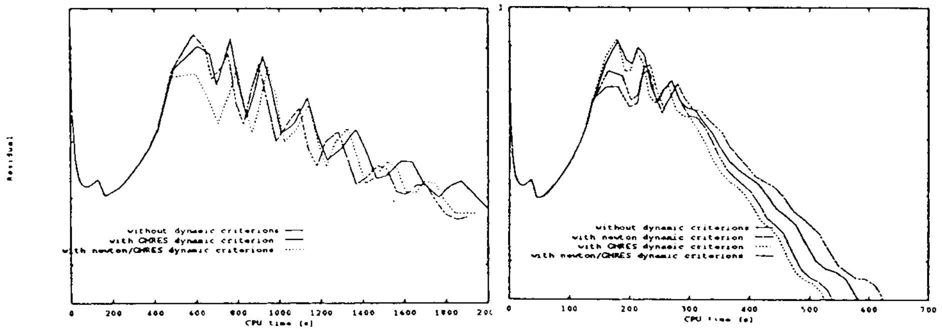

Figure 1. Dynamic decision of $k_{\max}$; initial iterations of a supersonic flow over a cylinder with the SUPG scheme. Left: mesh with 2256 nodes and 4096 elements. Right: mesh with 9399 nodes and 16,384 elements

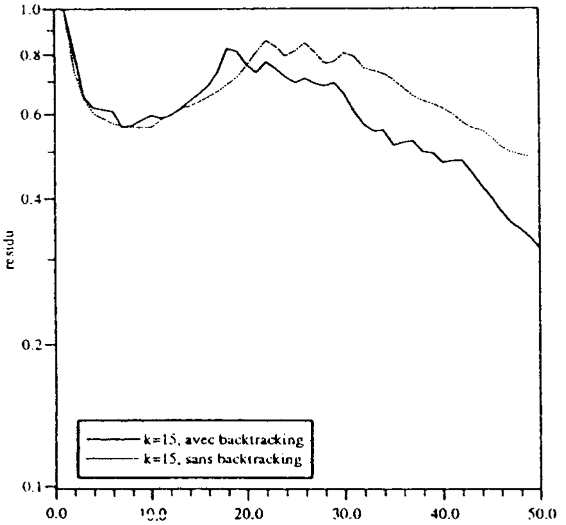

Figure 2. Effect of backtracking for the initial phase convergence

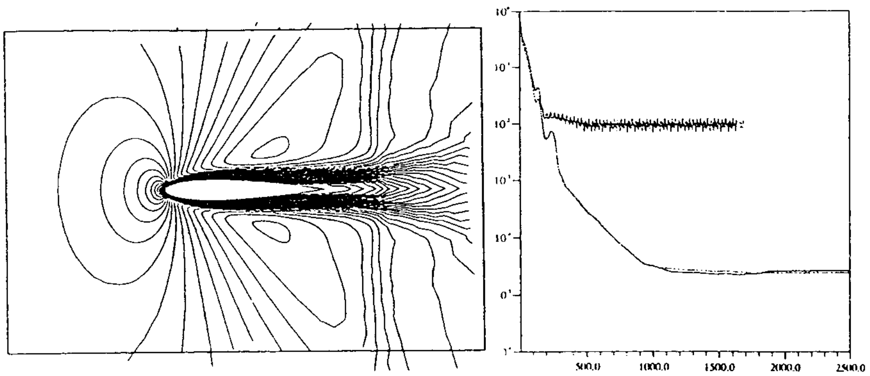

Figure 3. Transonic viscous flow over an NACA 0012 aerofoil, $M_{\infty} = 0{\cdot}5$, $Re = 1000$, and convergence acceleration due to ILU preconditioning in the wake. The transonic shock is thrown to the afterbody flow

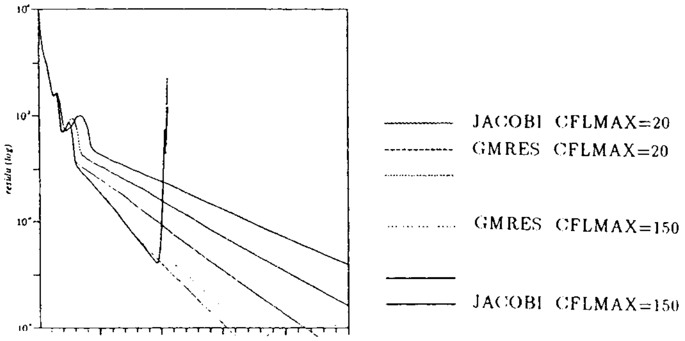

Figure 4. Jacobi versus linearized Newton-GMRES; convergence of the transonic viscous flow over an NACA 0012 aerofoil as in Figure 3

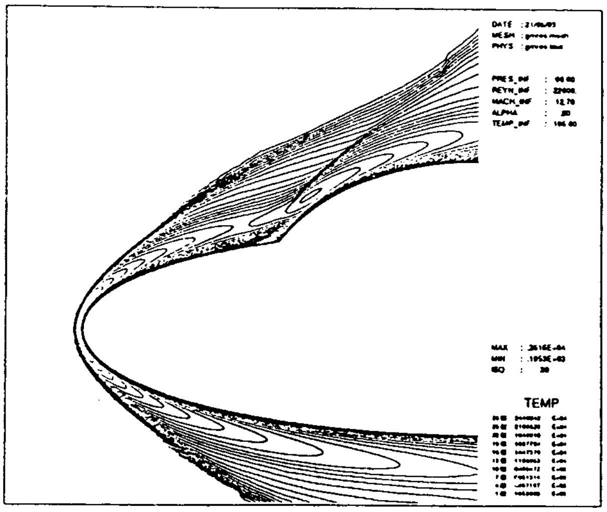

Figure 5. Isotherms of a hypersonic reactive viscous flow over a generic canopy (mesh with 23,477 nodes and 49,876 elements), $M_\infty = 12{\cdot}7$, $Re{\approx}10^5$

(as expected with such methods) only to be robust once the approximate Jacobians had approached the solution Jacobians and was inefficient for the initializing stages. Once a satisfactory order of convergence had been reached by the straightforward Jacobi relaxation method, switching on the Newton–GMRES method enhanced the convergence further from where the former method had saturated (Figures 5 and 6). Another spectacular result in this sense concerned the convergence of a hypersonic flow over a ramp with an induced separation bubble at the hinge.[8] The relaxation iterative methods gave stagnating convergence and the length of the separation bubble was still insufficient. By turning on the Newton–GMRES method, relaxation stagnation was attained leading to a converged solution with the separation bubble correctly captured (Figure 6, right).

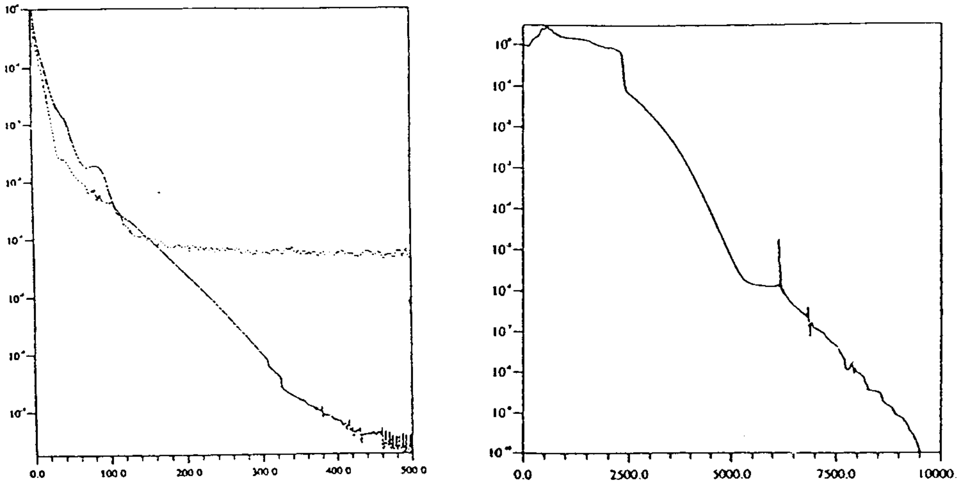

Figure 6. Left: convergence of the hypersonic reactive viscous flow of Figure 5 when the external non-linear Newton algorithm is turned on after two orders of convergence. Right: convergence of a hypersonic compression ramp computation where after 6000 Jacobi iterations the solution was insufficiently converged; by turning on Newton–GMRES for a further 1500 iterations, convergence was attained

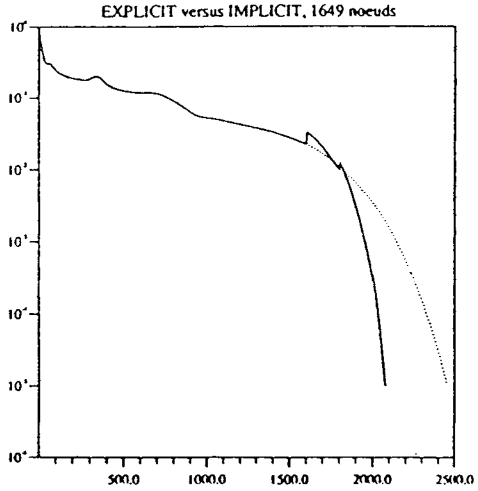

Figure 7. Convergence acceleration of the Newton algorithm using an exact Jacobian

Another test was performed on the preconditioning. Straightforward diagonal preconditioning is only really appropriate for convection-dominated non-complex flows. In the case of, for instance, a low transonic flow past an aerofoil, a shock is swept back at the trailing edge. A zone surrounding half the aerofoil to three chords in the wake was designated and an ILU preconditioning was applied. Elsewhere the diagonal one was maintained. A net improvement in convergence was obtained.

Finally, a novel comparison is shown in Figure 7, where the method with an exact Jacobian is compared with one using the finite difference approximation (see Section 2). The Newton method with the exact Jacobian did not converge when initializing with uniform flow conditions. Indeed, Newton methods require that the solution Jacobian be close to the tangent of the exact solution to ensure convergence and in the initial phases this is extremely hard to ensure. However, once the solution has started to converge, such that the exact Jacobian becomes close to the tangent, turning on the Newton method gives a spectacular increase in the rate of convergence. This tendency increases with increasing problem size.

6. CONCLUSIONS

In conclusion, linearized implicit or non-linear implicit methods for solving the complex systems of compressible flow can be performed with reasonable success by the use of Newton–GMRES algorithms. However, the highly localized physical phenomena give rise to very disparate conditioning of the block matrices and these global iterative methods applied in this point implicit way can thus result in unstable convergence properties. The possibility of coupling such methods to domain decomposition ones with specific conditioning per block seems to be promising.

ACKNOWLEDGEMENTS

This work is part of a Human Capital for Mobility programme of the EEC, with support from the Swiss Federal Office for Higher Education.

REFERENCES

1. P. N. Brown and Y. Saad, 'Hybrid Krylov methods for nonlinear systems of equations', *SIAM J. Sci. Stat. Comput.*, **11**, 450–481 (1990).
2. R. Choquet and P. Leyland, 'Viscous flow computations on the connection machine by a finite element Petrov–Galerkin scheme', *Proc. Parallel CFD '93*, Paris, May 1993.
3. T. Hughes, L. Franca and M. Mallet, 'A new finite element formulation for CFD, I', *Comput. Methods Appl. Mech. Eng.*, **54**, 223–234 (1986).
4. Z. Johan, T. Hughes and F. Shakib, 'A globally convergent matrix-free algorithm for implicit time-marching schemes arising in finite element analysis in fluids', *Comput. Methods Appl. Mech. Eng.* (1990).
5. K. Baba and T. Tabata, 'On a conservative upwind finite element scheme for convective diffusion equations', *RAIRO Numer. Anal.*, **15**, 3–25 (1981).
6. C. Johnson, U. Nävert and J. Pitkaranta, 'Finite element methods for linear hyperbolic problems', *Comput. Methods Appl. Mech Eng.*, **45**, 285–312 (1984).
7. E. Oñate, *ECCOMAS*, 1992.
8. P. Leyland, 'Shock wave/boundary layer interactions at hypersonic speeds by means of an implicit Navier–Stokes solver', *AIAA Paper 93-2938*, 1993.
9. R. Choquet, Résolution d'une formulation éléments finis par Newton/GMRES sur la machine aux connections, *IMHEF TM-93-3*, 1993.
10. Z. Johan, 'Data parallel finite element techniques for large-scale computational fluid dynamics', *Ph.D. Thesis*, Stanford University, 1992.
11. L. Bomholt and P. Leyland, 'Implementation of unstructured finite element codes on different parallel computers', *Proc. Parallel CFD '93*, Paris, May 1993.
12. J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, 1983.
13. Y. Saad and H. Schultz, 'GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems', *SIAM J. Sci. Stat. Comput.*, 7, 856–869 (1986).
14. T. Tefy, 'Solveurs implicites avec préconditionnement pour la resolution des equations de Navier–Stokes compressibles', *Master's Thesis*, Rennes, 1992; *IMHEF TM-92-25*, 1992.