University of Glasgow

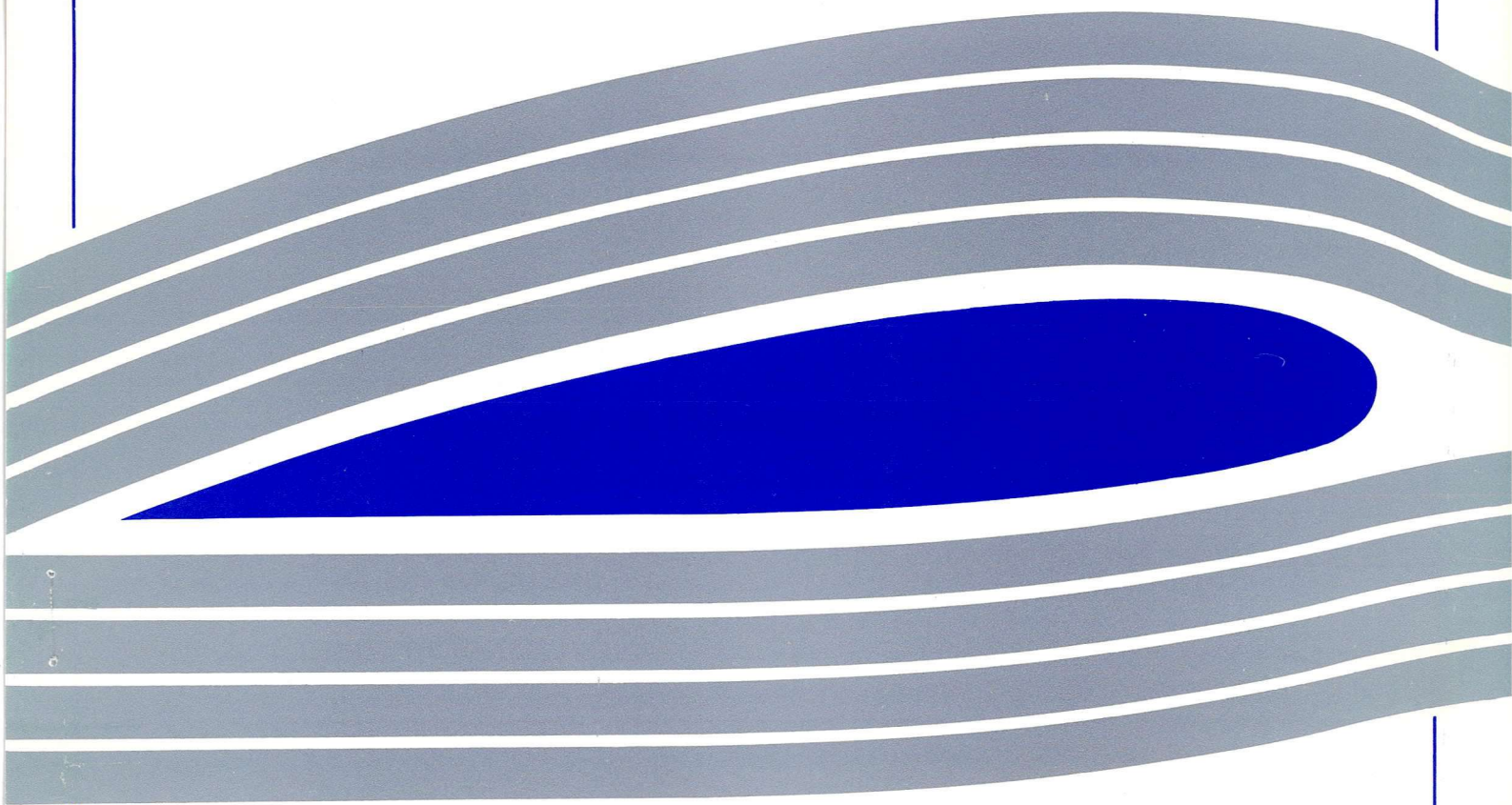
DEPARTMENT OF

AEROSPACE ENGINEERING

Applications and parallel implementation of continuation method for a fully implicit N-S solver

X. Xu and B.E. Richards

GU Aero Report 9416, October, 1994



Applications and parallel implementation of the continuation method for a fully implicit N-S solver

X. Xu and B.E. Richards

GU Aero Report 9416, October, 1994

Department of Aerospace Engineering
University of Glasgow
Glasgow G12 8QQ

Applications and parallel implementation of the continuation method for a fully implicit N-S solver

Xiao Xu and Bryan E. Richards

Department of Aerospace Engineering, University of Glasgow, Scotland, U.K.

High order accuracy, high resolution numerical schemes are most desirable when solving increasingly complex physical problems in CFD. However this involves more complicated computational formulations and increased CPU times. For steady state solutions, a time-dependent approach is usually followed using the unsteady governing equations, which can be discretized in time by an explicit or an implicit method. The use of an explicit method is robust in the sense that non-physical states can easily be avoided as long as the initial flow field is physically defined, but it can be slow to converge due to the stability restrictions on time steps even if some acceleration techniques were employed, such as local time stepping, multigrid, and the use of approximate implicit operators. Using an implicit method, unconditional stability can be achieved and as the time step approaches infinity the method approaches the fully implicit (the Newton's) method for the solution of the non-linear system corresponding to the steady state problem.

It is well known that the Newton's method has a quadratic convergence property for solving the non-linear algebraic system, and many different discrete versions are available. However their robustness and efficiency are based on a 'good initial' guess, the Jacobian generation technique, and a very efficient linear solver. For high speed compressible flow governed by the N-S equations studied in this paper, many complicated physical phenomena are involved, therefore a high order high resolution numerical scheme, Osher upwind scheme with the third order MUSCL interpolation, is employed. In this case, an efficient linear solver becomes even more critical. The discrete Newton's method is also memory intensive since the Jacobian is normally needed to be stored. After spatial discretisation of the N-S equations, the non-symmetric non-linear system can be denoted as

$$R(V) = 0$$

where R, V are vectors composed by residuals and independent variables components in all discretized points.

For the laminar flow discussed in this paper, the Jacobian **A** of the discrete Newton's method is a large sparse non-symmetric block band matrix since the structured grid and cell centred finite volume method are used. For a 2-dimensional flow field the **A** is block 13 band matrix, and for a 3-dimensional flow field the **A** will be block 25 band matrix. The effect of using fluxes instead of residuals in the formulation of Jacobian elements generation, for both divided difference approximation and analytical methods, has increased the efficiency of the discrete Newton's method [1-3]. For the divided difference approximation method the column by column order is used in the procedure of generating Jacobian elements for the sake of saving computation time. However, the extent of the variables used in generating a element is at least 7 cells in one direction. Using analytical method the row by row order can be used, where the extent of the variables used is 5 cells in one direction. Using the latter method in parallel calculation, the amount of communication will be decreased by one-third for the generation of **A**. In parallel calculation, the matrix **A** and a vector x are divided as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_P \end{bmatrix} \quad , \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_P \end{bmatrix}$$

where P is the number of processors used. Only A_p and x_p are stored in processor p. One of the major operations, matrix-vector manipulation, $y = \mathbf{A}\, x$, can be carried out as:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_P \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_P \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_P \end{bmatrix} \quad \text{, and} \qquad y_p = [\mathbf{A}_p] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_P \end{bmatrix}$$

Since processor p only store x_p, communication is needed to provide all required elements of x in the calculation.

Since the discrete Newton's method needs a 'good initial' guess, some hybrid methods can be used such as using an explicit method to make a good initial guess followed by a switch to a discrete Newton's method [4]. For practical calculation, the procedure for seeking a good initial guess is quite time consuming even if other acceleration techniques have been used. Alternatively, in this paper the continuation method [5] is used to widen the domain of convergence of the discrete Newton's method, and therefore, decrease the requirement for the initial guess from the explicit method. Assume V^0 is an approximate result from the explicit method, by using continuation method, then one possible algorithm for the discrete Newton's method is

$$\begin{cases} \left(\frac{\partial R(V)}{\partial V}\right)^k \Delta^k V = -\left(R(V^k) + \left(\frac{k}{N} - 1\right) R(V^0)\right) , & k = 1, 2, ..., N_1 \\ \left(\frac{\partial R(V)}{\partial V}\right)^k \Delta^k V = -R(V^k) , & k = N_1+1, N_1+2, ... \end{cases}$$

where N and N_1 are two positive integral number, and $N > N_1$. The first formula in the above algorithm is a modified discrete Newton's method. In practice, the larger the value of N, the more robust the algorithm will be. However, when a choice of N1 relatively smaller than N, the above algorithm will switch to the discrete Newton's method earlier, and therefore, rapid convergence is achieved.

The GMRES (and also the CGS) linear solver was used to solve the linear system in the discrete Newton's method. The block incomplete lower and upper factorization (BILUF) technique was used for obtaining an efficient preconditioner. Since the factorization is incomplete, thus some of information will be lost in the procedure. Therefore, the means by which the procedure of generating the preconditioner keeps the most important information will play a key role for creating a very efficient linear solver. One method is to arrange a suitable order of the cells in the mesh [6], such as the elements of Jacobian, which correspond to the most rapid change in physical phenomenon concentrating at its centre.

The generation of the lower and upper matrices, which have the same stencil as the Jacobian, and forward-backward substitution in the linear solver do suffer from sequential bottle-neck. Since the proportion of the CPU time has been reduced for these sequential steps, parallel implementation does show a reasonably good result and high parallel efficiency. Different parallel computers, including iPSC/860, CRAY-T3D, heterogeneous clustered workstations with PVM environment, have been tested.

Fig. 1, 2 show the convergence histories for the calculation of the locally conical N-S (LCNS) equations for the hypersonic flow over a cone at 24^o incidence. In the figures, by using the continuation method the switch points from explicit to implicit can be set much earlier, the CPU times shown are those for implementation on an IBM RS6000 320H workstation. Fig. 3 shows the parallel calculation of the parabolised N-S (PNS) equations for the high speed viscous flow over a Ogive cylinder at an incidence. The parallel results are obtained in iPSC/860.

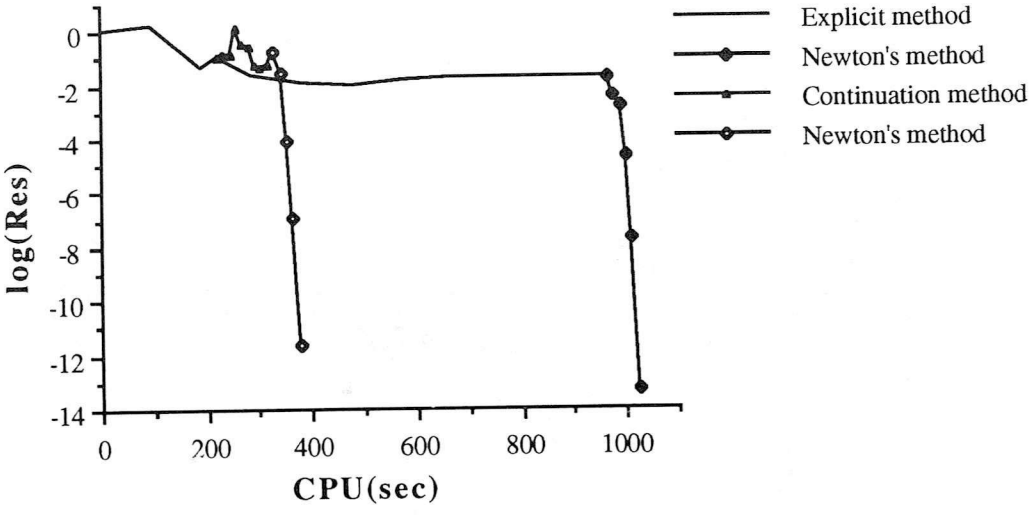


Fig. 1 Convergence history for LCNS solution 34×34 grid

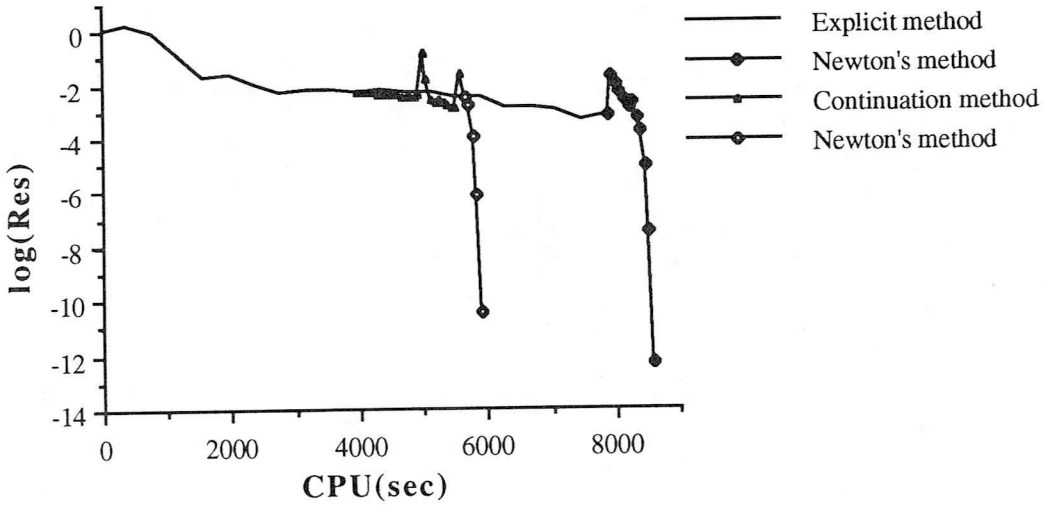


Fig. 2 Convergence history for LCNS solution 66×66 grid

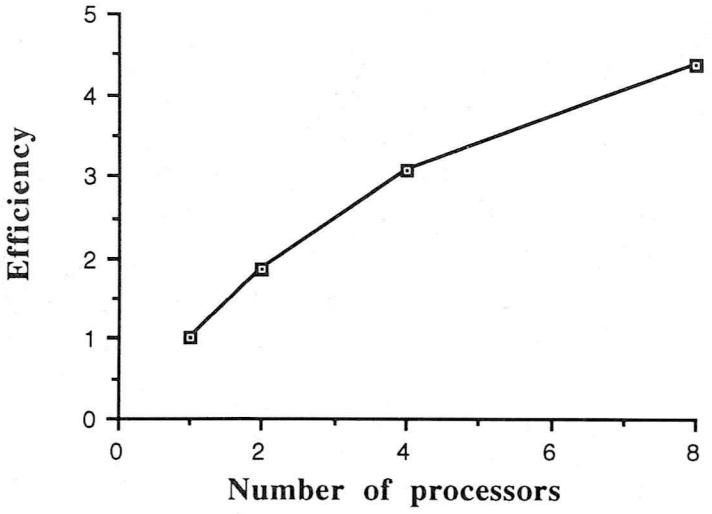


Fig. 3 Parallel efficiency for 34×34 grid

The authors acknowledge the financial support given by the Engineering and Physical Science Research Council, UK, through Grant GR/H48156.

References:

1. Xu, X. and Richards, B.E., "Simplified Procedure for Numerically Approximate Jacobian Matrix Generation in Newton's Method for Solving the Navier-Stokes Equations", G.U. Aero Report No.9320, 1993.

2. Jin, X.S., "Derivation of analytical Jacobian matrix for Newton method to 3D N-S equations", G.U. Aero Report No.9318, 1993.

3. Badcock, K., "Computation of turbulent pitching aerofoil flows", G.U. Aero Report No.9323, 1993.

4. Xu, X., Qin, N., and Richards, B.E., "Parallelising Explicit and Fully Implicit Navier-Stokes Solutions for Compressible Flows", *Proc. on Parallel CFD '92 Conf.*, Rutgers University, New Jersey, May 1992. Elsevier, Amsterdam, 1993.

5. Ortega, J. and Rheinboldt, W., "Iterative Solution of Nonlinear Equations in Several Variables", Academic Press, New York, 1970.

6. Xu, X. and Richards, B.E., "Numerical Advances in Newton Solvers for the Navier-Stokes Equations", Computational Fluid Dynamics '94, ed., Wagner, S., ed. Wiley, 1994.