University of Glasgow
DEPARTMENT OF
AEROSPACE
ENGINEERING

Simplified Procedure for Numerically Approximate Jacobian Matrix Generation in Newton's Method for Solving the Navier-Stokes Equations

X. Xu and B. E. Richards

GU Aero Report 9320, August, 1993

Simplified Procedure for Numerically Approximate Jacobian Matrix Generation in Newton's Method for Solving the Navier-Stokes Equations

X. Xu and B. E. Richards

GU Aero Report 9320, August, 1993

Department of Aerospace Engineering
University of Glasgow
Glasgow G12 8QQ

Simplified Procedure for Numerically Approximate Jacobian Matrix Generation in Newton's Method for Solving the Navier-Stokes Equations

X. Xu and B. E. Richards

December, 1992

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

Abstract

Since residual vector in a control volume is the linear combination of inviscid and viscous flux vectors on its all edges when a finite volume method is used, we can use flux vectors instead of residual vector in the formulation of generating Jacobian matrix elements. In the actual numerical calculation, after setting a variable perturbation in a control volume, we only calculate these flux vectors that according to the analysis they will change values because the variable perturbation, therefore we don't need to calculate all flux vectors which will compose of the residual vector. The new procedure decrease the amount of computation greatly, and also reduces the extent of the physical state variables used. For parallel computation the new procedure will short the storage of physical state variables in each subdomain.

CONTENTS

1. INTRODUCTION

Recent advances in high-speed supercomputers and parallel distributed memory multi-processors allow the Newton's method, a memory-intensive method, to be used in solving the Navier-Stokes equations. In practical applications it is very difficult to obtain the analytical Jacobian matrix of the nonlinear system when a high order high resolution scheme is used (it is almost impossible if turbulence or chemical reactions are involved). Different ways can be employed to handle this problem. One is to use the symbolic manipulation expert system MACSYMA to develop the exact Jacobian matrix [1,2], the other is to use numerically approximate methods to develop the approximate Jacobian [3,4,5]. By using the letter methods the quadratic convergence property of Newton's method is still retained.

2. NEWTON'S METHOD AND JACOBIAN MATRIX

2.1 Discretised Navier-Stokes equation

The structured grid, finite volume method, and Osher's upwind scheme are used to discretise the Navier-Stokes equation. The 2-dimensional flowfield is discussed as the example. In each control volume or cell (i,j) the discretised physical state variable vector and residual vector are

$$v_{cell(i,j)} = \begin{pmatrix} v^1_{(i,j)} \\ v^2_{(i,j)} \\ v^3_{(i,j)} \\ v^4_{(i,j)} \end{pmatrix}, \qquad R_{cell(i,j)}(V) = \begin{pmatrix} R^1_{(i,j)}(V) \\ R^2_{(i,j)}(V) \\ R^3_{(i,j)}(V) \\ R^4_{(i,j)}(V) \end{pmatrix} \tag{2.1}$$

where (i,j) = (1,1), ..., (I,J), and $V = \{ v_{cell(i,j)} \mid (i,j) = (1,1), ..., (I,J) \}$. Since in a cell (i,j) residual vector is composed of the inviscid and viscous flux vectors on its four interfaces

between the cell pairs (i-1,j), (i,j); (i,j), (i+1,j); (i,j-1), (i,j); and (i,j), (i,j+1) respectively, we have a vector formulation

$$R_{cell(i,j)}(V) = FI_{i+1/2,j}(V) - FI_{i-1/2,j}(V) + FI_{i,j+1/2}(V) - FI_{i,j-1/2}(V) + FV_{i+1/2,j}(V) - FV_{i-1/2,j}(V) + FV_{i,j+1/2}(V) - FV_{i,j-1/2}(V) \quad (2.2)$$

Therefore, the discretization of the Navier-Stokes equations results in a non-linear equations in a cell (i,j) as follows:

$$R_{cell(i,j)}(V) = 0 \quad (2.3)$$

The calculation of the flux vectors on the cell's edges will only use the discretised physical state variable vectors within its neighbouring cells, and when using a high order Osher scheme we have a 13 cell stencil as in Fig.2.1.

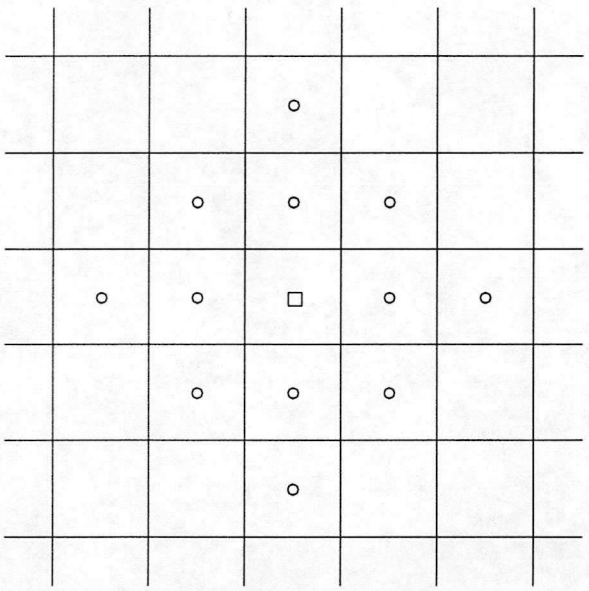


Fig. 2.1 13 cell stencil

Where we calculate the residual vector in cell marked square, and the discretised physical state variable vector need to be used within the cells marked square and circle.

Consolidating all the discretised Navier-Stokes equations in every cells we have a non-linear algebraic equation as follows:

$$\mathcal{R}(V) = 0 \tag{2.4}$$

where $\mathcal{R}(V) = \{ R_{cell(i,j)}(V) \mid (i,j) = (1,1), ..., (I,J) \}$. For a 2-dimensional flow the vectors V and $\mathcal{R}(V)$ are N-dimensional, N=I×J×4.

2.2 The Newton's method

For Eq. (2.4) the general Newton's method is

$$\left(\frac{\partial \mathcal{R}}{\partial V}\right)^k \Delta^k V = - \mathcal{R}^k (V)$$
$$V^{k+1} = V^k + \Delta^k V \tag{2.5}$$

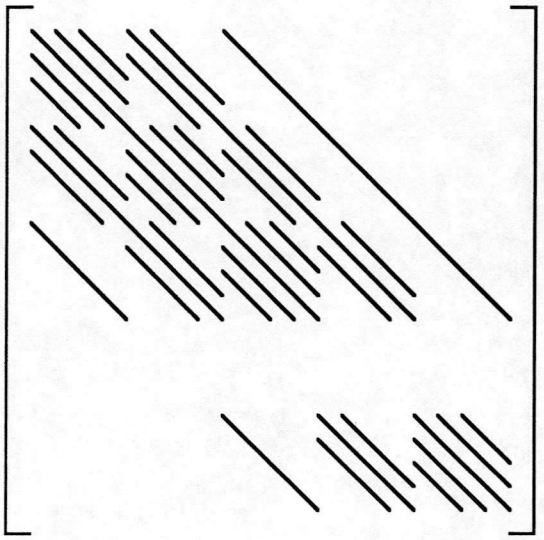
Since Eq. (2.4) is composed of Eq. (2.3) in each cell, we have the Newton's formulation in each cells as follows:

$$\left(\partial R_{cell(i,j)} (V) / \partial V\right)^k \Delta V^k = - R_{cell(i,j)} (V^k) \tag{2.6}$$

For any $v_{cell(l,m)}$, $\partial R_{cell(i,j)}/\partial v_{cell(l,m)}$ is a 4×4 sub-matrix, where i, l =1, ..., I, and j, m =1, ..., J. Since the 13 cell stencil above we know that Eq. (2.6) includes 13 4×4 sub-matrixes, which will form a block row of Jacobian matrix and are

$$\partial R_{cell(i,j)}/\partial v_{cell(i-2,j)},\ \partial R_{cell(i,j)}/\partial v_{cell(i-1,j-1)},\ \partial R_{cell(i,j)}/\partial v_{cell(i-1,j)},$$
$$\partial R_{cell(i,j)}/\partial v_{cell(i-1,j+1)},\ \partial R_{cell(i,j)}/\partial v_{cell(i,j-2)},\ \partial R_{cell(i,j)}/\partial v_{cell(i,j-1)},$$
$$\partial R_{cell(i,j)}/\partial v_{cell(i,j)},$$
$$\partial R_{cell(i,j)}/\partial v_{cell(i,j+1)},\ \partial R_{cell(i,j)}/\partial v_{cell(i,j+2)},\ \partial R_{cell(i,j)}/\partial v_{cell(i+1,j-1)},$$
$$\partial R_{cell(i,j)}/\partial v_{cell(i+1,j)},\ \partial R_{cell(i,j)}/\partial v_{cell(i+1,j+1)},\ \partial R_{cell(i,j)}/\partial v_{cell(i+2,j)}. \tag{2.7}$$

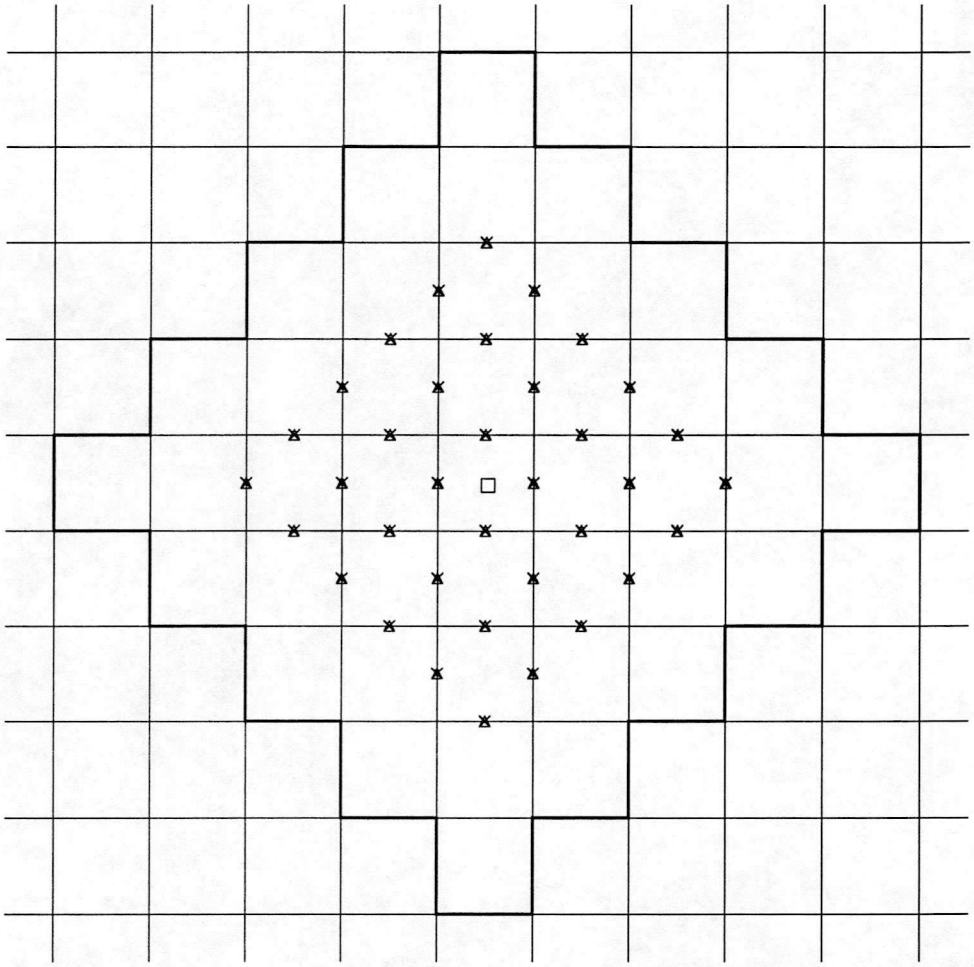
Therefore the Jacobian matrix of Eq. (2.5) will be a block 13-point diagonal matrix which can be denoted as



On the other hand, a block column of Jacobian matrix will be

$$\begin{aligned}
&\partial R_{cell(i-2,j)}/\partial v_{cell(i,j)},\ \partial R_{cell(i-1,j-1)}/\partial v_{cell(i,j)},\ \partial R_{cell(i-1,j)}/\partial v_{cell(i,j)},\\
&\partial R_{cell(i-1,j+1)}/\partial v_{cell(i,j)},\ \partial R_{cell(i,j-2)}/\partial v_{cell(i,j)},\ \partial R_{cell(i,j-1)}/\partial v_{cell(i,j)},\\
&\qquad\qquad \partial R_{cell(i,j)}/\partial v_{cell(i,j)},\\
&\partial R_{cell(i,j+1)}/\partial v_{cell(i,j)},\ \partial R_{cell(i,j+2)}/\partial v_{cell(i,j)},\ \partial R_{cell(i+1,j-1)}/\partial v_{cell(i,j)},\\
&\partial R_{cell(i+1,j)}/\partial v_{cell(i,j)},\ \partial R_{cell(i+1,j+1)}/\partial v_{cell(i,j)},\ \partial R_{cell(i+2,j)}/\partial v_{cell(i,j)}.
\end{aligned} \tag{2.8}$$

From (2.8) we can see that in order to generate a block column elements of the Jacobian matrix we need calculate the residual vectors with 13 cells, which have the same stencil as in Fig.2.1. For obtaining these residual vectors we should calculate 36 inviscid and viscous flux vectors, which are shown in Fig.2.2. Fig.2.2 also shows that the cell extent of physical state variable vectors are needed in this procedure.



×: inviscid flux vector needed, Δ: viscous flux vector needed,

square: the perturbation position. Physical state variable vectors used lie within bold line

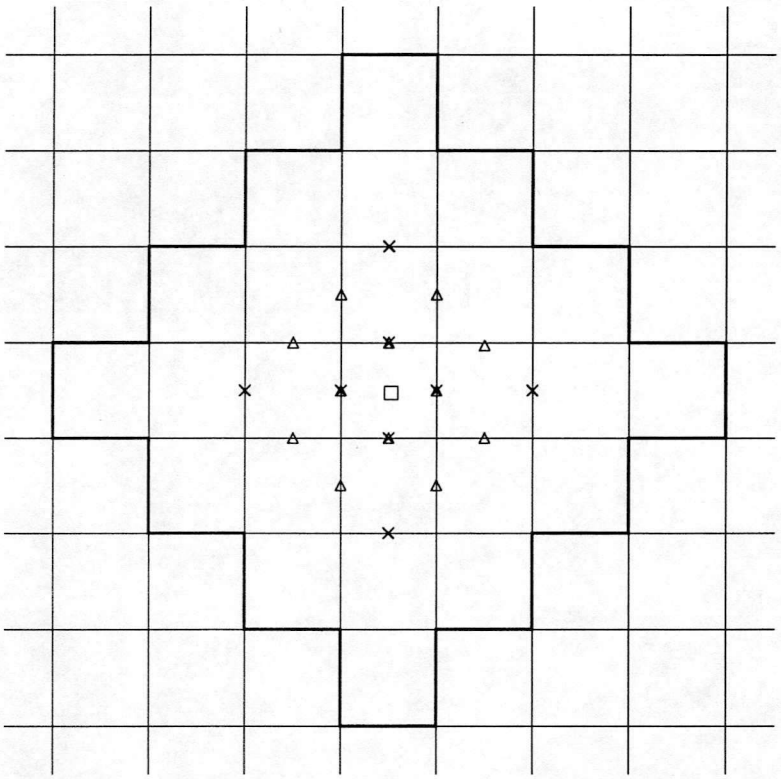
Fig. 2.2 The fluxes needed in the original formulation

A new simplified procedure can be implemented by substituting (2.2) to (2.8). We have

$$\partial R_{cell(l,m)}/\partial v_{cell(i,j)} = \begin{cases} (\partial FI_{l+1/2,m} - \partial FI_{l-1/2,m} + \partial FI_{l,m+1/2} - \partial FI_{l,m-1/2} \\ + \partial FV_{l+1/2,m} - \partial FV_{l-1/2,m} + \partial FV_{l,m+1/2} - \partial FV_{l,m-1/2}) \\ /\partial v_{cell(i,j)} \end{cases}$$

where (l,m) will be one of the 13 cells in Fig.2.1. After changing $v_{cell(i,j)}$ the flux vectors changed are only 8 inviscid and 12 viscous flux vectors, which are shown in Fig.2.3. Therefore the amount of calculations of inviscid and viscous flux vectors decrease from 36 to

8 and 36 to 12 respectively. Fig.2.3 also shows the cell extent of physical state variable vectors needed in this new procedure.



×: inviscid flux vector needed, Δ: viscous flux vector needed,

square: the perturbation position. Physical state variable vectors used lie within bold line

Fig. 2.3 The fluxes needed in the new formulation

Therefore we have the following formulations:

$$\partial R_{cell(i-2,j)}/\partial v_{cell(i,j)} = \partial FI_{i-3/2,j}/\partial v_{cell(i,j)}$$

$$\partial R_{cell(i-1,j-1)}/\partial v_{cell(i,j)} = (\partial FV_{i-1/2,j-1} + \partial FV_{i-1,j-1/2})\ /\partial v_{cell(i,j)}$$

$$\partial R_{cell(i-1,j)}/\partial v_{cell(i,j)} =$$
$$(\partial FI_{i-1/2,j} - \partial FI_{i-3/2,j} + \partial FV_{i-1/2,j} + \partial FV_{i-1,j+1/2} - \partial FV_{i-1,j-1/2})\ /\partial v_{cell(i,j)}$$

$$\partial R_{cell(i-1,j+1)}/\partial v_{cell(i,j)} = (\partial FV_{i-1/2,j+1} - \partial FV_{i-1,j+1/2})\ /\partial v_{cell(i,j)}$$

$$\partial R_{cell(i,j-2)}/\partial v_{cell(i,j)} = \partial FI_{i,j-3/2}/\partial v_{cell(i,j)}$$

$$\partial R_{cell(i,j-1)}/\partial v_{cell(i,j)} =$$
$$(\partial FI_{i,j-1/2} - \partial FI_{i,j-3/2} + \partial FV_{i,j-1/2} + \partial FV_{i+1/2,j-1} - \partial FV_{i-1/2,j-1})/\partial v_{cell(i,j)}$$

$$\partial R_{cell(i,j)}/\partial v_{cell(i,j)} = \begin{cases} (\partial FI_{i+1/2,j} - \partial FI_{i-1/2,j} + \partial FI_{i,j+1/2} - \partial FI_{i,j-1/2} \\ + \partial FV_{i+1/2,j} - \partial FV_{i-1/2,j} + \partial FV_{i,j+1/2} - \partial FV_{i,j-1/2}) \\ /\partial v_{cell(i,j)} \end{cases}$$

$$\partial R_{cell(i,j+1)}/\partial v_{cell(i,j)} =$$
$$(\partial FI_{i,j+3/2} - \partial FI_{i,j+1/2} - \partial FV_{i,j+1/2} + \partial FV_{i+1/2,j+1} - \partial FV_{i-1/2,j+1})/\partial v_{cell(i,j)}$$

$$\partial R_{cell(i,j+2)}/\partial v_{cell(i,j)} = -\partial FI_{i,j+3/2}/\partial v_{cell(i,j)}$$

$$\partial R_{cell(i+1,j-1)}/\partial v_{cell(i,j)} = (\partial FV_{i+1,j-1/2} - \partial FV_{i+1/2,j-1})/\partial v_{cell(i,j)}$$

$$\partial R_{cell(i+1,j)}/\partial v_{cell(i,j)} =$$
$$(\partial FI_{i+3/2,j} - \partial FI_{i+1/2,j} - \partial FV_{i+1/2,j} + \partial FV_{i+1,j+1/2} - \partial FV_{i+1,j-1/2})/\partial v_{cell(i,j)}$$

$$\partial R_{cell(i+1,j+1)}/\partial v_{cell(i,j)} = (-\partial FV_{i+1/2,j+1} - \partial FV_{i+1,j+1/2})/\partial v_{cell(i,j)}$$

$$\partial R_{cell(i+2,j)}/\partial v_{cell(i,j)} = -\partial FI_{i+3/2,j}/\partial v_{cell(i,j)}$$

Using the above formulations we prefer to generate the elements of Jacobian matrix according to the order of column by column.

2.3 Numerically Approximate Jacobian matrix

An approximation for generating the above elements of the Jacobian matrix can be obtained by replacing derivative with a finite difference formulation as follows:

$$\partial F/\partial v_{cell(i,j)} = \frac{F(v + \Delta v_{cell(i,j)}) - F(v)}{\Delta v_{cell(i,j)}} \qquad (2.9)$$

where F is FI or FV, which presents the inviscid or viscous flux vector. Various ways of selecting $\Delta v_{cell(i,j)}$ have been suggested in the numerical analysis literature. When choosing $\Delta v_{cell(i,j)} = h\, e_{(i,j)}$, where $e_{(i,j)}$ are unit vectors, Dennis and Schnabel [6] pointed out that if a sequence $\{h_k\}$ is used for the step size h, and if this sequence is properly chosen, then the quadratic convergence property of Newton's method is retained and Newton's method using finite differences is 'virtually indistinguishable' from Newton's method using analytic derivatives. In this paper the h is chosen as $\varepsilon \times v_{cell(i,j)}$, where $\varepsilon \approx \sqrt{}$ [machine epsilon] and $v_{cell(i,j)}$ are the state variable vector.

3. NUMERICAL TESTS

The foregoing numerical tests have been carried out on the test case of a laminar Mach 7.95 flow around a sharp cone with a cold wall (T_w =309.8K) and at an angle of attack of 24°. The Reynolds number is 4.1×10^6 and the flow temperature is 55.4K. The numerical solution is simplified to 2-D flow case since conical flow theory is used.

A structured grid and cell-centred finite volume method are used. The spatial discretization scheme used is the Osher flux difference splitting scheme. The formal accuracy is third order for the convective fluxes and second order for the diffusive fluxes. The linearization iterative method is Newton's method. The block element of the Jacobian matrix is a 5×5 sub-matrix since there are five state variables in each cell. The streamwise flux need to be calculated. However the simplified procedure reduced these calculations from in 13 cells to just in one cell. The computer used is the IBM RISC 6000/320H workstation.

Table 1 gives the comparison of the cpu time needed for the two procedures for generating the Jacobian matrix.

Table 1: CPU time (second) for generating the Jacobian matrix

Grids	Original formulation	New formulation
34×34	30 sec	6 sec
66×34	65 sec	13 sec
66×66	135 sec	28 sec
66×130	314 sec	54 sec

Table 2 gives the comparison of the cpu time needed for generating the Jacobian matrix using one step of an explicit iteration as the basic scale.

Table 2: CPU time needed for generating Jacobian matrix

Grids	Original formulation	New formulation
34×34	27.3	5.4
66×34	29.5	5.9
66×66	31.2	5.6
66×130	36.1	6.2

5. CONCLUDING REMARKS

The new simplified procedure presented in this paper creates significant economy in generating a Jacobian matrix. For the test cases the cpu time needed is only about 6 times of one step of an explicit iteration. This new formulation also reduces the extent of physical state variables used, so that for parallel calculation the new formulation can decrease the storage of these variables in each subdomain.

REFERENCES

1. Orkwis, P.D. and McRae, D.S. Newton's Method Solver for High-Speed Viscous Separated Flowfields. *AIAA J.* Vol. 30, (1), 1990, pp. 78-85.
2. Orkwis, P.D. and McRae, D.S. A Newton's Method Solver for the Axisymmetric Navier-Stokes Equations. AIAA Paper 91-1554.
3. Qin, N. and Richards B. E. Sparse Quasi-Newton method for Navier-Stokes Solutions. *Notes in Numerical Fluid Mechanics*, Vol. 29, 1990, pp. 474-483.
4. Xu, X., Qin, N., and Richards, B.E. Parallelization of discretized Newton method for Navier-Stokes equations. To appear in *Proc. on Parallel CFD* '92. Rutgers University, New Brunswick, New Jersey, 18-20, May 1992.
5. Whitfield, D.L. and Taylor, L.K. Discretized Newton-Relaxation Solution of High Resolution Flux--Difference Split Schemes. AIAA Paper 91-1539.
6. Dennis, J.E., Jr. and Schnabel, R.B. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice Hall, Englewood Cliffs, N.Y. 1983.
7. Xu, X., Qin, N., and Richards, B.E. α-GMRES: A new parallelizable iterative solver for large sparse non-symmetric linear system arising from CFD. *Int. J. Numer. Methods Fluids*. 15, 615-625 (1992).