11th Australasian Fluid Mechanics Conference
University of Tasmania, Hobart, Australia
14-18 December 1992

A NUMERICAL STUDY OF HYPERSONIC LEEWARD FLOW OVER A DELTA WING USING A PARALLEL ARCHITECTURE SUPERCOMPUTER

E.R. MALLETT[1], D.I. PULLIN[2] and M.N. MACROSSAN[1]

[1]Dept of Mechanical Engineering, University of Queensland, QLD 4072 AUSTRALIA

[2]Graduate Aeronautical Laboratories, Caltech, Pasadena CA 91125, USA

ABSTRACT

We have implemented Pullin's kinetic theory based Equilibrium Flux Method (Pullin, 1980) on a parallel supercomputer (the Intel iPSC Touchstone Delta) to study the leeward flowfield of a blunt nosed delta wing at hypersonic speeds. Computational results are presented for a series of grids for both inviscid and viscous flows. Of particular interest are the vortex and shock structures in the leeward flow that are evident only with the high grid resolution afforded by the use of the parallel computer. In addition, some experiences of developing a CFD code for a parallel computer environment are discussed.

INTRODUCTION

The design of reusable spacecraft such as the Hermes (currently under development by the European Space Agency) will be enhanced by an accurate computational prediction of the flowfield during hypersonic re-entry into the atmosphere. This is particularly important on the leeward side of lifting surfaces where vortex phenomena have been judged responsible for intense local heating (Whitehead, 1970). The combined effects of the high spatial grid resolution necessary for accuracy, the complexity of model used and the three-dimensionality of the flow mean that these computations can only be performed in a reasonable amount of time by machines of the 'supercomputer' class.

The speed of a single processor has been shown to be asymptotically approaching a limit imposed by physical considerations (Denning, 1985). However, performance in excess of this limit can be achieved by dividing the necessary amount of computation between a network of processors which proceed to compute in parallel. The development of computer codes capable of being executed on such machines has been identified as one of the pacing items in CFD (Chapman, 1992; Steger & Hafez, 1992). However, it is claimed that software development has been lagging behind hardware development by 5 to 10 years (Peterson & Ballhaus, 1987).

In light of this claim, this work was conducted on a parallel supercomputer - the Intel iPSC Touchstone Delta at the California Institute of Technology. The Delta is an example of a MIMD (Multiple Instruction stream/Multiple Data stream) machine or 'multi-computer' (Flynn, 1966) and is constructed from 512 Intel i860 processors connected in a two-dimensional mesh. There is no shared memory therefore transfer of information between processors is accomplished using message passing. This computer is capable of a theoretical maximum speed of 32 Gflops (billion floating point operations per second) in 64 bit double precision mode and has 8 Gbytes of primary memory available with 200 Gbytes of disk storage.

NUMERICAL METHOD

The Equilibrium Flux Method (EFM) is an explicit finite volume shock-capturing method which may be applied to the solution of the Euler and Navier-Stokes equations for compressible fluids. EFM can be derived by consideration of the Boltzmann equation which provides a spatial and temporal description of the gas at the molecular level -

$$\frac{\partial}{\partial t}(nf)+\mathbf{c}\cdot\frac{\partial}{\partial \mathbf{r}}(nf)=\left[\frac{\partial}{\partial t}(nf)\right]_{collisions} \quad (1)$$

where the molecular distribution function in phase space is given by the product of the molecule number density n and the molecular velocity distribution function f, where $\mathbf{r}$ and $\mathbf{c}$ are the position and velocity of a molecule in physical space at time t.

A set of conservation equations for a gas obeying Boltzmann's equation can be obtained by taking moments of the equation using an algebraic vector Q consisting of the molecular quantities [m, $m\mathbf{c}$, $1/2m\mathbf{c}.\mathbf{c}$]. A moment of the phase space distribution function is given by

$$<Q>=\int_{\mathbf{c}}(nf)Q d\mathbf{c}. \quad (2)$$

Now consider an tetrahedral element or cell of physical space with volume V_i and surface area S_i,. We will define a local set of coordinate axes$(\hat{\mathrm{n}},\hat{\mathrm{p}},\hat{\mathrm{q}})$ such that $\hat{\mathrm{n}}$ is normal to the cell face and pointing outward, and $\hat{\mathrm{p}}$ and $\hat{\mathrm{q}}$ approximately lie in the plane of the face and form an orthogonal base set. Integrating the moments of the Boltzmann equation over the cell and applying the divergence theorem to the convection term gives

$$\frac{\partial}{\partial t}\oiiint_V U_Q dV+\oiint_S F_Q dS=0 \quad (3)$$

where U_Q is a set of conserved quantities per unit volume given by

$$U_Q=\int_{\mathbf{c}}(nf)Q d\mathbf{c} \quad (4)$$

and F_Q is a set of fluxed quantities,

$$F_Q=\int_{\mathbf{c}}(nf)Q\mathbf{c}\cdot\hat{\mathrm{n}}d\mathbf{c}. \quad (5)$$

The integral of the collision term is equal to zero as the molecular quantities Q are conserved in each collision. It is implicit in the derivation of F_Q, that both outward and inward fluxes are accounted for. For each cell face, the flux expression can be split into two parts: an outward or *forward* moving flux F_Q^+ and an inward or *backward* moving flux F_Q^- where

$$F_Q=F_Q^{+}+F_Q^{-}. \quad (6)$$

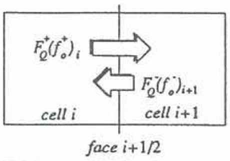

FIGURE 1. CELL FACE

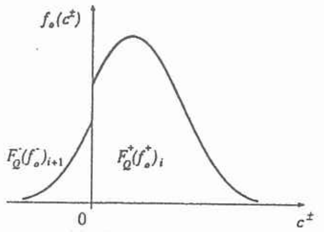

FIGURE 2. VELOCITY DISTRIBUTION

In a local coordinate system (figure 1), the set of forward and backward fluxes are then given by

$$F_Q^- = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{0}(n^- f_o^-) Q c_n dc_n dc_p dc_q, \tag{7}$$

and

$$F_Q^+ = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{0}^{\infty}(n^+ f_o^+) Q c_n dc_n dc_p dc_q. \tag{8}$$

For EFM it is assumed that the velocity distribution function on each side of the cell interface tends towards an equilibrium distribution dependent upon local flow conditions and given by Maxwell's equilibrium distribution function,

$$f = f_0 = \left(\frac{\beta^3}{\pi^{3/2}}\right)\exp\left(-\beta^2(\mathrm{c}-\bar{\mathrm{c}})^2\right) \tag{9}$$

where β is the reciprocal of the most probable thermal speed, and is defined by

$$\beta = \frac{1}{\sqrt{2RT}}. \tag{10}$$

The form of the distribution function for the normal component at the cell interface is shown in figure 2. The transport of mass, momentum and energy across a cell surface can be calculated analytically. A convenient form of the flux expressions is described by Macrossan et al. (1991).

After transforming flux expressions into a global cartesian system, the mass, momentum and energy in each cell is updated by using the product of the nett flux across a cell interface, the face area and a timestepΔt. The method is made second order accurate in space by using a *min-mod* strategy (van Leer, 1979) to estimate properties at cell interfaces. Simple Euler timestepping is used subject to the Courant-Friedrichs-Lewy (CFL) condition.

The gradients necessary for the estimation of viscous terms are obtained by using a secondary staggered grid, the vertices of which are the centroids of the primary grid. The use of the divergence theorem and central differences allows the gradients to be estimated at cell vertices in the primary grid. Sutherland's formula is used to estimate viscosity as a function of temperature.

FREESTREAM CONDITIONS AND BODY GEOMETRY

The body considered in this work was a thick blunt nosed delta wing of 70 degrees sweep, flying at an angle of attack of 30 degrees. The freestream Mach number was fixed at 8.7 with a freestream temperature of 55 K. Laminar flow was assumed for the viscous calculations with Reynolds numbers ranging from 10^5 to 10^6 and a fixed wall temperature of 300 K.

The original computational grid used was a C-O type grid consisting of 36 x 40 x 72 cells in the downstream, radial and azimuthal directions respectively. Due to symmetry about the x-y plane only one half of the body was modeled (z≥0) as shown in figure 3. This grid was subsequently truncated or refined using four point Lagrange interpolation to produce two further grids. Table I describes each of the grids.

Table I Grid Descriptions

Grid	Resolution	Dimensions	Cells
1	coarse	18 x 20 x 36	12,960
2	medium	36 x 40 x 72	103,680
3	fine	72 x 80 x 144	829,440

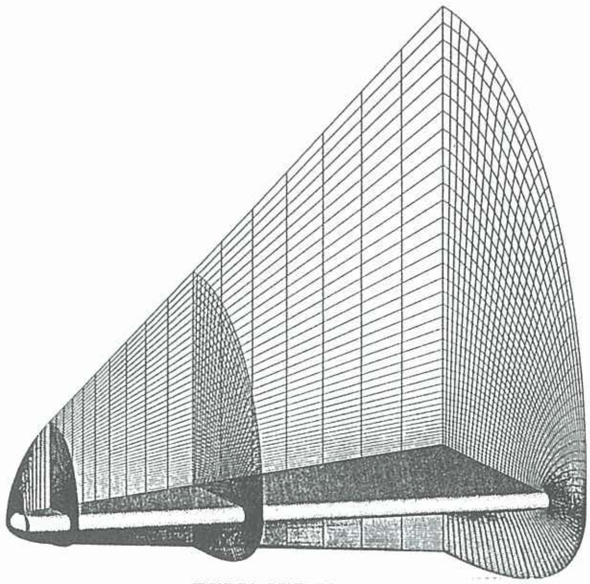
FIGURE 3 DELTA WING GRID

PARALLEL IMPLEMENTATION

The Parallel General Purpose 3-Dimensional (PGP3D) code was written in standard FORTRAN 77 and uses the communication calls of the Parasoft Express operating system which allows portability between a number of parallel machines. In addition, the use of dummy libraries enables the code to be run on sequential type machines without modification. Only one version of the code needs to be maintained, with identical copies of this code being loaded into every processor at runtime.

Parallelization was achieved by distributing the physical domain of the grid amongst the available processors in such a way that the boundary surface area of each sub-grid was minimized. This reduces the amount of communication between processors as compared to computation and thus the overhead associated with using this machine architecture.

A series of ghost cells are used around the perimeter of each sub-grid to implement boundary conditions. Before each timestep the values of primitive variables in these cells are updated by communicating with the neighboring processor which has the recently calculated value in memory. Thus the algorithm consists of alternate periods of computation and communication. The communication is carried out synchronously with each processor waiting for all others to finish computing before passing values. Consequently overall speed is dictated by the speed of the slowest processor. It is therefore extremely important to balance the computing load between the available processors.

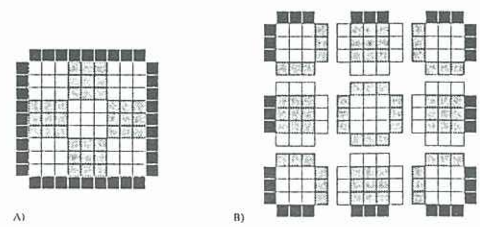

FIGURE 4 DOMAIN DECOMPOSITION

RESULTS

Figure 5 shows streamlines and density contours at the nose of the delta wing in the plane of symmetry for the inviscid solution on the finest grid. The bow shock is clearly detached and the position of the stagnation point can also be seen. Figure 6 shows surface streamlines for the same case. At the nose and close to the plane of symmetry the crossflow velocity component is small. Further along the leading edge the curvature is such that the crossflow velocity is large and supersonic. Because of the symmetry condition in the x-y plane a shock is produced at the surface which causes the flow to separate. This produces a vortex which causes the fluid between the shock and the plane of symmetry to flow in an outboard direction.

Figure 7 shows a comparison of the pressure field of the inviscid solution on the three grids. The comparison is made at the downstream station $x/L = 0.8$. Although the bow shock position is about the same for each case, the larger number of cells in the radial direction of grid 3 clearly allows greater resolution of the shock. The flow undergoes a large expansion around the leading edge but remains attached. The crossflow component of velocity must go to zero at the plane of symmetry which causes the inbound flow to undergo a compression. For the flow close to the surface this occurs approximately midway along the semi-span.

In the solution for the highest resolution grid (grid 3) a shock system is evident. This structure is similar in many respects to that computed by Marconi (1989) for the flow over a circular cone under similar freestream conditions. The flow structure can be seen more readily in figures 8 and 9. The crossflow shock is of sufficient strength to cause the flow to separate with a contact surface rolling up into a vortex. Just away from the surface the crossflow shock is oblique and the crossflow velocity remains supersonic as it passes through the shock. Again the symmetry condition requires this supersonic flow to pass through another crossflow shock that sits approximately above the vortex core. The fluid close to the plane of symmetry is accelerated by the presence of the vortex sufficiently to produce a supersonic cross flow velocity in the direction of the wing surface and is again shocked. As the flow is turned outboard and parallel to the surface it undergoes a series of expansions and compressions before being turned away from the surface on the inboard side of the separation streamline. The appearance of these shocks about the periphery of the vortex has led Marconi to describe such systems as 'pinwheel shocks'.

For the viscous calculations, the position and strength of the vortex was found to be similar to that of the inviscid case, even though the flow separated much further outboard. No cross flow shock structures were observed in the viscous cases, but in the region where the vortex first appears a region of heat transfer to the wing surface was seen.

PARALLEL PERFORMANCE

The PGP3D code was run on different numbers of processors to gauge the efficiency of running on multiple processors. For the viscous calculations, the measured efficiency for all 512 processors was approximately 85% (80% for the inviscid runs), resulting in a speedup of about 400. Even with a conservative estimate of 10 Mflops per processor, this constitutes performance of several gigaflops.

CONCLUDING REMARKS

The high computing performance afforded by parallel computers allows a depth of detail and turnaround times unavailable with sequential type computers. It is hoped that these results encourage others to expend the additional time and effort required to develop codes for the parallel computer environment.

ACKNOWLEDGMENTS

We would like to acknowledge and thank Prof. Hans Hornung, Director of GALCIT, for making arrangements so that this work could be carried out at Caltech; the Caltech Concurrent Supercomputing Facilities (CCSF); the Concurrent SuperComputing Consortium (CSCC); and Prof. Chien-Peng Li of the NASA Johnson Space Center for providing the grid. This work was supported by the Australian Research Council under Grant Number A89031403 and by the Walter and Eliza Hall Traveling Scholarship.

REFERENCES

CHAPMAN, D R (1992) A perspective on aerospace CFD. Aerospace America, January 1992.

DENNING, P J (1985) The science of computing - 1985. Research Institute for Advanced Computer Science, RIACS TR.85.12.

FLYNN, M J (1966) Very high speed computing systems. Proceedings of the IEEE, 54, 12 (December), pp. 1901-1909.

MACROSSAN, M N (1989) The Equilibrium Flux Method for the calculation of flows with non-equilibrium chemical reactions. Journal of Computational Physics, vol. 80, no. 1, January 1989.

MACROSSAN, M N, MALLETT E R & PULLIN D I (1991) Flow calculations for the Hermes delta wing at an angle of attack using a second order kinetic theory based Euler solver. Proceedings of the Workshop on Hypersonic Flows for Reentry Problems, Antibes, France, 1991, Institut National de Recherce en Informatique et en Automatique, Antibes, pp. 85-101.

MARCONI, F (1989) Complex shock patterns and vortices in inviscid supersonic flows. Computers & Fluids, vol. 17, no. 1, pp. 151-163.

PERRY, A E & HORNUNG, H (1984) Some Aspects of Three-Dimensional Separation. Z. Flugwiss. Welt. vol. 8, no. 2.

PETERSON, V L & BALLHAUS, W F Jr. (1987) History of the Numerical Aerodynamic Simulation Program. Supercomputing in Aerospace, NASA CP-2454.

PULLIN, D I (1980) Direct simulation methods for compressible inviscid ideal-gas flow. Journal of Computational Physics, vol. 34, pp. 231-244.

STEGER, J L & HAFEZ, M M (1992) CFD goes to school. Aerospace America, January 1992.

VAN LEER, B (1979) Towards the ultimate conservative difference scheme: a second order sequel to Godunov's method. Journal of Computational Physics, 32, pp. 101-136.

WHITEHEAD, A H Jr. (1970) Effect of vortices on delta wing leeward flow at Mach 6. AIAA Journal, vol. 8, no. 3, March 1970.

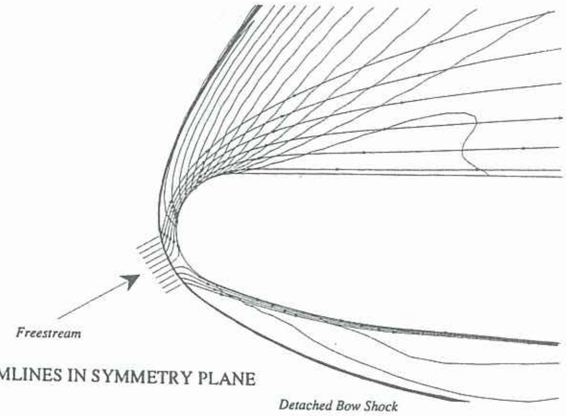

FIGURE 5 STREAMLINES IN SYMMETRY PLANE

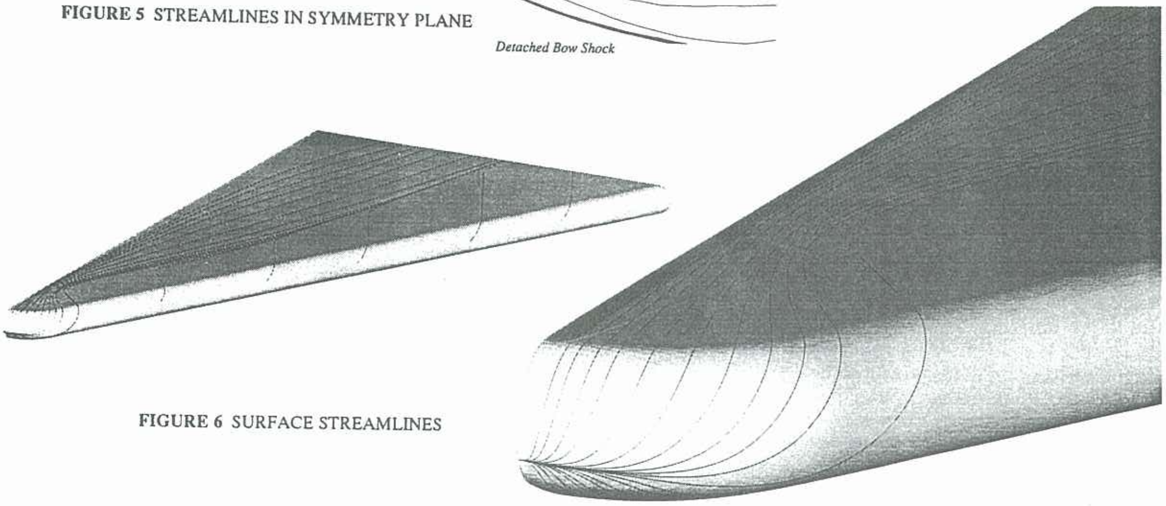

FIGURE 6 SURFACE STREAMLINES

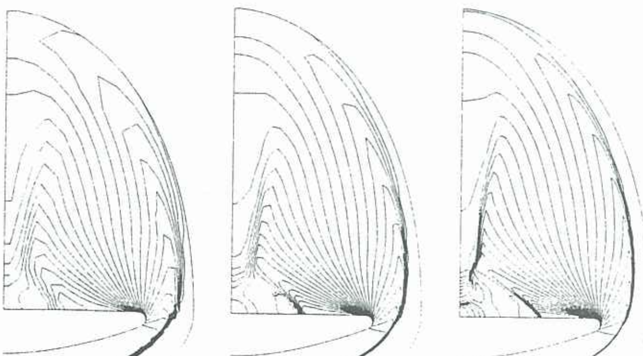

FIGURE 7 PRESSURE FIELD AT X/L = 0.8

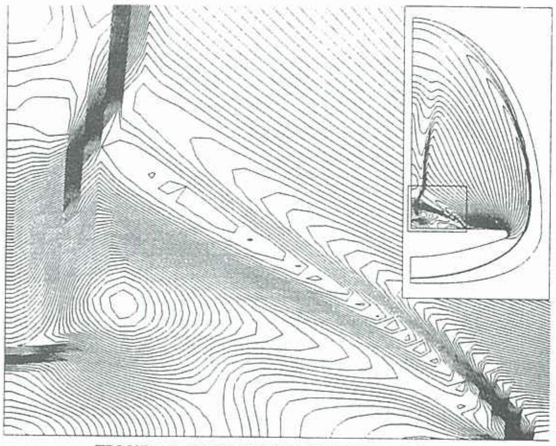

FIGURE 8 DENSITY FIELD AT X/L = 0.8

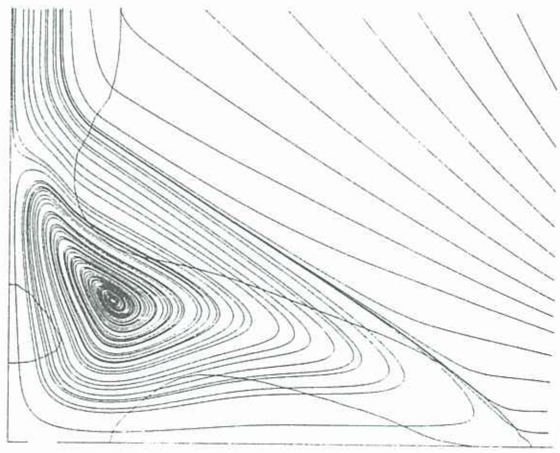

FIGURE 9 SECTIONAL STREAMLINES AT X/L = 0.8