AD-A081 579 MICHIGAN UNIV ANN ARBOR DEPT OF ELECTRICAL AND COMPU--ETC F/G 20/4
NUMERICAL SOLUTION OF THE 3-D NAVIER-STOKES EQUATIONS ON THE CR--ETC(U)
1979 J S SHANG, P G BUNING, W L HANKEY AFOSR-75-2812
UNCLASSIFIED AFOSR-TR-79-1075 NL

END
DATE FILMED
4-80
DDC

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) Unclassified

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFOSR-TR-79-1075	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER: Interim rept.
4. TITLE (and Subtitle): NUMERICAL SOLUTION OF THE 3-D NAVIER-STOKES EQUATIONS ON THE CRAY-1 COMPUTER		5. TYPE OF REPORT & PERIOD COVERED: Interim
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): J.S. SHANG, P.G. BUNING, W.L. HANKEY, M.C. WIRTH, D.A. CALAHAN, and W. AMES		8. CONTRACT OR GRANT NUMBER(s): AFOSR-75-2812
9. PERFORMING ORGANIZATION NAME AND ADDRESS: University of Michigan, Dept. of Elec. and Computer Eng.		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F 2304/A3
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Office of Scientific Research/NM, Bolling aFB, Washington, DC 20332		12. REPORT DATE: 1979 13. NUMBER OF PAGES: 8
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report): UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release;distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
fluid dynamics
aerodynamic simulation
vector processors
parallel processors

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A three-dimensional, time dependent Navier-Stokes code using MacCormack's explicit scheme has been vectorized for the CRAY-1 computer. Computations were performed for a turbulent, transonic, normal shock wave boundary layer interaction in a wind tunnel diffuser. The vectorized three-dimensional Navier-Stokes code on the CRAY-1 computer achieved a speed of 128 times that of the original scalar code processed by a CYBER 74 computer. The vectorized version

DD FORM 1 JAN 73 1473

20. Abstract continued.

of the code outperforms the scalar code on the CRAY computer by a factor of 8.13. A comparison between the experimental data and the numerical simulation is also made.

NUMERICAL SOLUTION OF THE 3-D NAVIER-STOKES EQUATIONS ON THE CRAY-1 COMPUTER

J. S. Shang,* P. G. Buning,** W. L. Hankey*
M. C. Wirth,* D. A. Calahan** and W. Ames**
*Air Force Flight Dynamics Laboratory
**University of Michigan

ABSTRACT

A three-dimensional, time dependent Navier-Stokes code using MacCormack's explicit scheme has been vectorized for the CRAY-1 computer. Computations were performed for a turbulent, transonic, normal shock wave boundary layer interaction in a wind tunnel diffuser. The vectorized three-dimensional Navier-Stokes code on the CRAY-1 computer achieved a speed of 128 times that of the original scalar code processed by a CYBER 74 computer. The vectorized version of the code outperforms the scalar code on the CRAY computer by a factor of 8.13. A comparison between the experimental data and the numerical simulation is also made.

NOMENCLATURE

c	Speed of Sound
Def	Deformation Tensor
e	Specific Internal Energy $c_v T + (u^2 + v^2 + w^2)/2$
$\bar{F}, \bar{G}, \bar{H}$	Vector Fluxes, Equation (15)
L_ξ, L_η, L_ζ	Differencing Operator
M	Mach Number
P	Static Pressure
$\dot{q}$	Rate of Heat Transfer
R_{ey}	Reynolds Number Based on Running Length $\rho_\infty u_\infty x/\mu_\infty$
T	Static Temperature
t	time
$\bar{U}$	Dependent Variables in Vector Form (ρ, ρu, ρv, ρw, ρe)
$\bar{u}$	Velocity Vector
u, v, w	Velocity Components in Cartesian Frame
x, y, z	Coordinates in Cartesian Frame
ξ, η, ζ	Transformed Coordinate System, Equation (14)
ρ	Density
$\bar{\tau}$	Stress Tensor

INTRODUCTION

In the past decade, computational fluid dynamics has become firmly established as a credible tool for aerodynamics research[1,2]. Aided by some rather crude and heuristic turbulence models, success has been achieved even for complex turbulent flows[3-8]. In spite of all these convincing demonstrations, the objective of a wide application of computational fluid dynamics in engineering design has yet to be achieved. The basic limitation is in cost effectiveness. A lower cost and systematic methodology needs to be developed[9].

The present analysis addresses one of the key objectives in obtaining efficient numerical processing. To achieve this objective, two approaches seem obvious; either develop special algorithms designed for a particular category of problems according to the laws of physics or utilize an improved computer. In the case of special algorithms, a better understanding of the generic structure of the flow field is required. In general, these attempts have been successful and have achieved an order of magnitude improvement in computing speed. On the other hand, a class of computers designed for scientific computations; the CRAY-1, STAR 100 and ILLIAC IV among others, has become available. The most significant advance in computer hardware related to computational fluid dynamics is the vector processor which permits a vector to be processed at an exceptional speed. This option gives a new perspective; i.e., a drastic reduction in computing time[10, 11, 12].

A three-dimensional time dependent Navier-Stokes code using MacCormack's explicit scheme[13] has been vectorized for the CRAY-1 computer. The selection of this particular finite differencing scheme is based on its past ability to perform a large number of successful bench mark runs[2-7], its proven shock-capturing capa-

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF TRANSMITTAL TO DDC
This technical report has been reviewed and is approved for public release IAW AFR 190-12 (7b).
Distribution is unlimited.
A. D. BLOSE
Technical Information Officer

bility, and the inherent simplicity of the basic algorithms. The Cray-1 computer was chosen because at the present time, among all the available general purpose scientific processors, it provides the highest potential floating point computation rate in both the scalar and the vector mode[14]. The combination of the selected algorithm and the CRAY-1 computer provides a bench mark for future development and a tool for current engineering evalution.

The problem selected for evaluating the CRAY-1 performance was the experimental investigation of Abbiss[15,16] of a three-dimensional interaction of a normal shock with a turbulent boundary layer in a square wind tunnel diffuser at a Reynolds number of thirty million and Mach number of 1.51. The primary purpose of the paper is to determine the computational speed of the code, although a comparison with experimental data is presented to demonstrate the validity of the solution.

GOVERNING EQUATIONS

The time dependent, three dimensional compressible Navier-Stokes equations in mass-averaged variables can be given as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \bar{u}) = 0 \qquad (1)$$

$$\frac{\partial \rho \bar{u}}{\partial t} + \nabla \cdot (\rho \bar{u}\bar{u} - \bar{\bar{\tau}}) = 0 \qquad (2)$$

$$\frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho e \bar{u} - \bar{u} \cdot \bar{\bar{\tau}} + \bar{q}) = 0 \qquad (3)$$

The turbulent closure of the present analysis is accomplished through an eddy viscosity model. The effective thermal conductivity is also defined by the turbulent Prandtl number ($Pr_t = 0.9$). The equation of state, Sutherland's viscosity law and assigned molecular Prandtl number (0.73) formally close the system of governing equations.

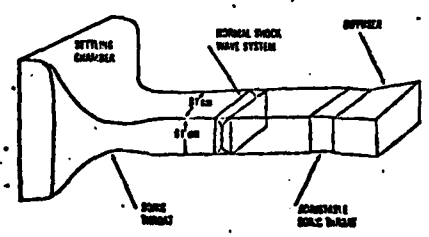

Figure 1. Flow Field Schematic

Since the wind tunnel flow field consisted of four symmetrical quadrants, only a single quadrant was computed. The boundaries of the computational domain contain two intersecting wind tunnel walls and two planes of symmetry for which the associated boundary conditions are straight forward (Figure 1). In order to develop upstream conditions equivalent to the experiment a separate computation is initiated with a free stream condition and permitted to develop a three-dimensional boundary layer along the corner region until the boundary layer duplicates the experimental observation ($\delta = 4.0$ cm, $x = 316$ cm)[15]. Then, the computed flow field at this streamwise location is imposed as the upsteam condition for the interaction computation. On the wind tunnel walls, the boundary conditions are no-slip for the velocity components and a constant surface temperature. The wind tunnel wall pressure is obtained by satisfying the momentum equation at the solid surface. On the planes of symmetry, the symmetrical boundary conditions are given for all dependent variables. The normal shock wave across the wind tunnel is then specified according to the Rankine-Hugoniot conditions. The far downstream boundary condition is the well known no-change condition. In summary:

INITIAL CONDITION:

$$\bar{u}(0, \xi, \eta, \zeta) = \bar{U}_\infty \qquad (9)$$

UPSTREAM CONDITION:

$$\bar{u}(t, 0, \eta, \zeta) = \bar{U}_\infty \qquad (10)$$

DOWNSTREAM :

$$\left.\frac{\partial \bar{u}}{\partial x}\right|_{x \to x_L} = 0 \qquad (11)$$

ON PLANES OF SYMMETRY:

$$\left.\frac{\partial \bar{u}}{\partial y}\right|_{y = y_L} = 0 \text{ and } \left.\frac{\partial \bar{u}}{\partial z}\right|_{z = z_L} = 0 \qquad (12)$$

ON WIND TUNNEL WALL:

$$u = v = w = 0 \qquad (13a)$$

$$T_w = 313.79°K \quad \text{at } y, z = 0 \qquad (13b)$$

$$\nabla \cdot \bar{\bar{\tau}} = 0$$

A coordinate system transformation is introduced to improve the numerical resolution in the viscous dominated region.

$$\xi = x/x_L \qquad (14a)$$

$$\eta = 1/k \ln[1 + (e^k - 1)\, y/y_L] \qquad (14b)$$

$$\zeta = 1/k \ln[1 + (e^k - 1)\, z/z_L] \qquad (14c)$$

The governing equations in the transformed space are of the following form:

$$\frac{\partial \bar{U}}{\partial t} + \xi_x \frac{\partial \bar{F}}{\partial \xi} + \sum_i \eta_{x_i} \frac{\partial \bar{G}}{\partial \eta} + \sum_i \zeta_{x_i} \frac{\partial \bar{H}}{\partial \zeta} = 0 \tag{15}$$

where ξ_x, η_y and ζ_z are the metrics of the coordinate transformation. The definition of the conventional flux vectors F, G, and H can be found in Ref. 7.

NUMERICAL PROCEDURE AND DATA STRUCTURE

The basic numerical method is the time-split or factorized scheme originated by MacCormack. The finite difference formulation in terms of the difference operator can be expressed as

$$\bar{U}^{n+2} = \sum_n L_\zeta \left(\frac{\Delta t}{2n}\right) \sum_m L_\eta \left(\frac{\Delta t}{2m}\right) L_\xi (\Delta t)$$

$$\sum_m L_\eta \left(\frac{\Delta t}{2m}\right) \sum_n L_\zeta \left(\frac{\Delta t}{2n}\right) \tag{16}$$

Each difference operator contains a predictor and corrector. During a specific numerical sweep, the flux vectors are approximated by a central, forward, and backward differencing scheme in such a fashion that after a complete cycle of the predictor and corrector operations all the derivatives are effectively approximated by a central differencing scheme. A graphic representation of these operations is given by Figure 2.

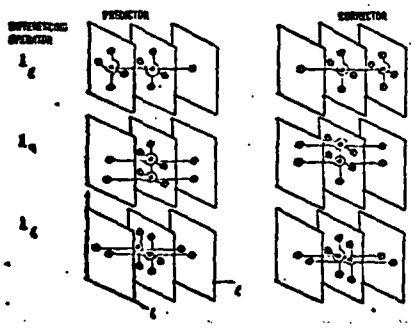

Figure 2. Grid Points Involved in the Time Step Sweep

When investigating flows with strong shock waves, it is necessary to employ numerical damping in a shock-capturing scheme. Fourth-order pressure damping was utilized which generates an artificial viscosity-like term.[17]

$$\Delta t \Delta \xi_i^3 \frac{\partial}{\partial \xi_i} \left[\frac{|u_i| + c}{4p} \frac{\partial^2 p}{\partial \xi_i^2} \right] \frac{\partial \bar{U}}{\partial \xi_i} \quad i = 1,2,3$$

The approximation of second order central differencing for the corrector step required additional grid point information beyond the immediately adjacent planes. The damping terms, however, are effective only in the presence of shock waves where the numerical resolution is degraded.

From the symmetric differencing operator sequence of predictor and corrector steps, one detects that the dependent variables in the predictor level can be completely eliminated by retaining only the three cyclic pages currently in use (Figure 3). For a flow field requiring a large amount of data storage, this reduction in memory requirement is substantial. Meanwhile, the paging process is reduced from two sweeps to one. The predictor and corrector sequence is performed within one sweep by overlapping the corrector operation during one fractional time step.

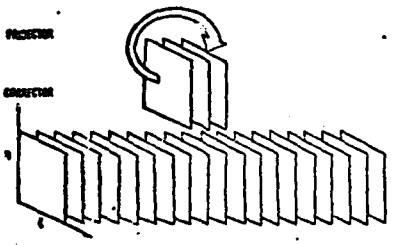

Figure 3. Data Storage and Data Flow Diagram

Once the planar or page storage is adopted, the vector length can be determined. Separate vectors are constructed for η and ζ directions, yielding vector lengths approximately equal to the number of grid points in each direction. In order to keep all solutions in the same page (η - ζ plane), the streamwise sweep (ξ sweep) is vectorized in the ζ direction.

For the present problem, the computational domain with the dimension of 356.3cm x 45.5cm x 45.5cm is partitioned into two streamwise sections of 64 pages each. Every page contains 33 x 33 grid points in η and ζ coordinates respectively. The problem is solved in two steps. The first computational section generates a three-dimensional boundary layer over a corner which becomes the in-flow boundary condition for the following shock-boundary interaction domain. Both contain 64 x 33 x 33 grid points, but a finer streamwise mesh spacing $\Delta x = 1.27$ cm was used for the interaction zone to gain a finer numerical resolution of the shock-

boundary layer interaction. The ratio between the fine and coarse streamwise grid spacing is 0.3063 of the local boundary-layer thickness (4.0cm)[15]. The cross flow plane grid-point distribution, however, remains identifical between the two overlapping segments. The memory requirement for each is about 0.545 million words.

The numerical solution is considered at its steady state asymptote when the maximum difference between two consecutive time levels of the static pressure in the strong interacting zone is less than 0.2 percent. In the leading computational domain the convergence criterion is established similarly but is based on the velocity profiles instead of pressure.

TIMING RESULTS

A portion of the present effort is aimed at making internal comparisons of the relative times for various types of functional unit processing and memory loading (I/O) for the vectorized code. A knowledge of relative time expenditure information is important to provide some insight into the program execution rate. Although this type of data is code dependent, the present example is deemed typical of a large class of Navier-Stokes solvers. The timing information is measured by vector operation counts[11] and shown in Figure 4. It is obvious that the relative usage of the memory path and functional units is dominated by memory loadings (34.6%) and floating point multiplication (33.3%). Within the functional units, the relative usage of the floating point addition and multiplication has the ratio of two to three. The relative usage of the reciprocal approximation is extremely rare, i.e. less the 2%. In spite of the high percentage of memory loading, a portion of the vectorized Fortran code has achieved an execution rate of 42.9 MFLOPS[11]. Further improvements still can be made either in Fortran or assembly language versions of the present code. However, we feel an overall execution rate greater than 60 MFLOPS on this size problem is unlikely.

A basic dilemma exists for the comparative investigation; namely in the process of vectorization significant changes were made either on the amount of computation performed or on the number of subroutine calls made. The final vectorized program usually bears little resemblence to the original scalar code[7, 11]. Substantial improvement in performance of the vectorized code on a scalar machine has also been reported. However, this improvement in performance can be considered as a contribution due to the vectorization process.

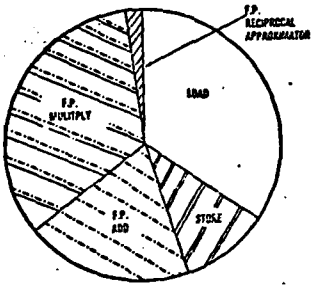

Figure 4. Vector Operation Counts in Percentage

In order to perform the comparative study, a criterion must be established. The ultimate evaluation of data processing rate is the computing time. The completely duplicated computations for an identical fluid mechanics problem are usually prohibited by the incore memory and the indexing limitations for various processors. Therefore, one has to accept the rate of data processing as the criterion. The rate of data processing is commonly defined as

RDP = CPU Time/(Total Number of Grid Points x Total Number of Iterations)

The particular rate of data processing is most suitable for numerical programs with similar algorithms and convergence rate. If the ratio between field grid points and boundary points can be maintained between two programs then the comparison is particularly meaningful.

In Table 1, the comparison of timing results between the scalar code and vectorized code on the CRAY-1 is presented.

Table 1

The Comparison of Scalar and Vector Processing on CRAY-1

VERSION OF CODE	RDP(Sec/Pts, ITERATIONS)
Scalar	4.761×10^{-4}
Vector	4.861×10^{-5}

The vectorized program outperforms the original scalar code by a factor of 8.13. In Table 2, the timing results of the scalar code and vectorized code perform-

ance for four different computers are given.

Table 2

Comparative Timing Results

COMPUTER		(RDP)	$\frac{\text{CYBER 74}}{\text{RDP}}$
CYBER 74	Scalar	7.48×10^{-3}	1.0
CDC 7600	Scalar	1.45×10^{-3}	5.2
CRAY-1	Scalar	4.76×10^{-4}	15.7
CRAY-1	Vector	5.86×10^{-5}	127.7
CRAY-1	Assembly	5.19×10^{-5}	144.2

A brief description of each running condition for which the timing results were obtained may help with the interpretion of the data. The computations conducted on CYBER 74 and CDC 7600 with a grid point system of (17 x 33 x 33) were performed in the early phase of the present task[7]. On the CYBER 74 computer the data storage problem was overcome by a data manager subroutine in conjunction with a random access disk file. The computation carried out on CDC 7600 used large core memory for all the dependent variables. The I/O requirement is substantial, particularly for the computation performed on the CYBER 74.

FORTRAN VS. ASSEMBLY LANGUAGE

The multiple functional units and memory hieracrchy of the CRAY-1 can be difficult for the Fortran compiler (CFT) to manage efficiently. Consequently, CRAY Assembly Language (CAL) versions of a number of subroutines which account for up to 78% of the computation time were written with the aid of a simulator [18]. These kernels were also vectorized in Fortran with the CRAY-1 architecture and compiler features in mind; however, non-ANSI standard utility functions [19] and unusual Fortran constructs [20] were not employed. The principle timing results follow.

1) Among 9 kernels, assembly language speedups ranged from 11% to 29% with vector lengths of 33 (= a grid dimension).

2) An overall speedup of 14.2% was achieved (Table 2), including the common 22% Fortran.

3) A detailed simulator-produced evaluation of a subroutine which accounts for ≃ 20% of the total computation time is given in [21]. The execution rate of ≃ 50 MFLOPS is 1/3 of the maximum practical rate of the processor. However, the memory path is busy 70% of the time for the Fortran code for a vector length of 63, and up to 90% for the CAL code, indicating the memory bound nature of the algorithm on the CRAY-1. Indeed, the 90% busy time is viewed as an excellent indicator of the optimality of the CAL code.

A more detailed comparative study of this code is given in [21].

COMPARISONS WITH EXPERIMENTAL DATA

In Figure 5, several velocity profiles across the wind tunnel at a Reynolds number of 3.0×10^7 are presented. This location represents the flow field condition at the end of the leading segment of the computational domain which is also the upstream condition for the following interaction zone. The present results agree reasonably well with the data of Seddon[16]. The data, however, were collected at a Reynolds number one decade lower than the present condition and at a slightly different Mach number (1.47 v.s. 1.51). At the range of Reynolds numbers considered, the Reynolds number dependence should be scaled out by the boundary layer thickness. An independent boundary-layer calculation using the exact simulated condition was performed that verified this contention. It was found that the difference in magnitude of velocity is a few percent. The present result underpredicts the measured boundary layer thickness[15] by about eight percent

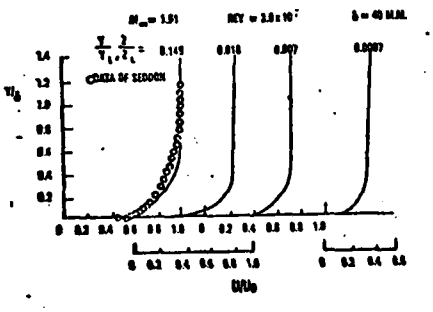

Figure 5. Velocity Profiles Along the Tunnel Wall

A direct comparison of several velocity distributions between the data of Abbiss et al[15] and the present calculation is presented in Figure 6 for the interaction region. The data are displayed for fixed x/δ and y coordinates away from the corner domain. The coordinate x is taken in the streamwise direction along the tunnel floor and y normal to the floor. Excellent agreement between the data and calculation is observed for the regions either deeply imbedded within the boundary layer or completely contained in the inviscid domain. The maximum discrepancy between data and calculation is in the lambda wave structure. The maximum desparity between data and calculations is about 10 percent.

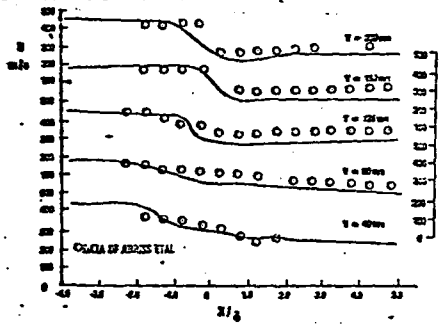

Figure 6. Comparison of the Flow Field Velocity in the Interactive Region

In Figure 7 the Mach number contour is presented in an attempt to compare with the flow field structure given by Abbiss et al[15] in Figure 8. The bifurcation of the normal shock wave is clearly indicated. The calculation nearly duplicates all of primary features of the experimental observation. However, a difference can be discerned in the dimension of the embedded supersonic zone between the experimental observation and calculation. The local supersonic zone emanates from the expansion due to the total pressure difference between the normal shock and the lambda shock structure and the rapid change in the displacement surface. A few percent disparity in predicting the magnitude of velocity lead to the distinguishable discrepancy in the definition of the embedded supersonic zone. A similar observation may be made for the work of Shea[22] in his investigation of the two-dimensional normal-shock wave turbulent boundary layer interaction.

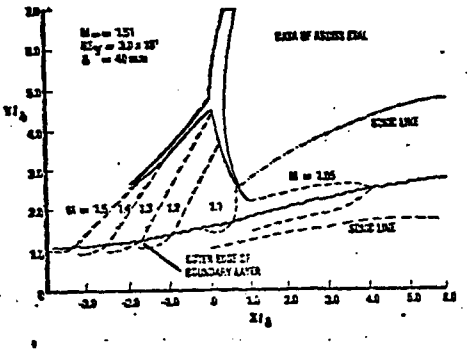

Figure 7. Experimentally Measured Flow Field Structure in the Plane of Symmetry

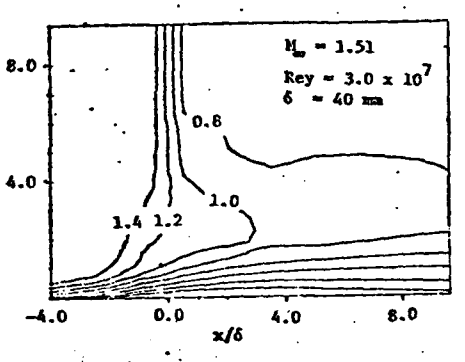

Figure 8. Computed Number Contour in the Plane of Symmetry

In Figure 9, the velocity distribution parallel to the wind tunnel side is given. A reverse flow is observed beneath the lambda shock wave system. The separated flow region begins about three boundary-layer thickness upstream of the normal shock and terminates at five boundary layer thickness downstream. The length of the separated domain is similar to the measurement of Seddon[16] and the numerical simulation by Shea[22].

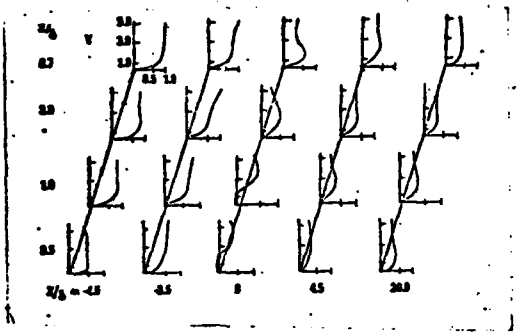

Figure 9. Computed Velocity Field in the Interaction Region

The entire flow field structure is presented in Figure 10 in terms of density contours at various streamwise locations. The shear layer over the corner region, the strong inviscid-viscous domain, and the subsequent readjustment of the flow field are easily detectable. A clear indication of substantial growth of the shear layer over the wind tunnel wall is also obvious.

CONCLUSIONS

A three-dimensional time dependent Navier-Stokes code using MacCormack's explicit scheme has been vectorized for the CRAY-1 computer achieved a speed of 128 time that of the original scalar code processed by a CYBER 74 computer. The vectorized code outperforms the scalar code on the CRAY-1 computer by a factor of 8.13.

The numerical simulation for a turbulent, transonic, normal shock-wave boundary-layer interaction in a wind tunnel has been successfully performed using a total 139,400 grid points. The numerical result indicates sufficient resolution for engineering purposes. Additional increase in speed by up to an order of magnitude through algorithm requirement also seems attainable.

ACKNOWLEDGEMENT

The authors wish to acknowledge the assistance of S. Arya and E. Sesek of the University of Michigan in preparation of the CRAY-1 program. This work was prepared in part under the auspices of Grant AF AFOSR Grant 75-2812. The authors also wish to express their appreciation to Cray Research, Inc. and Lawrence Livermore Laboratory for the use of their computer facility.

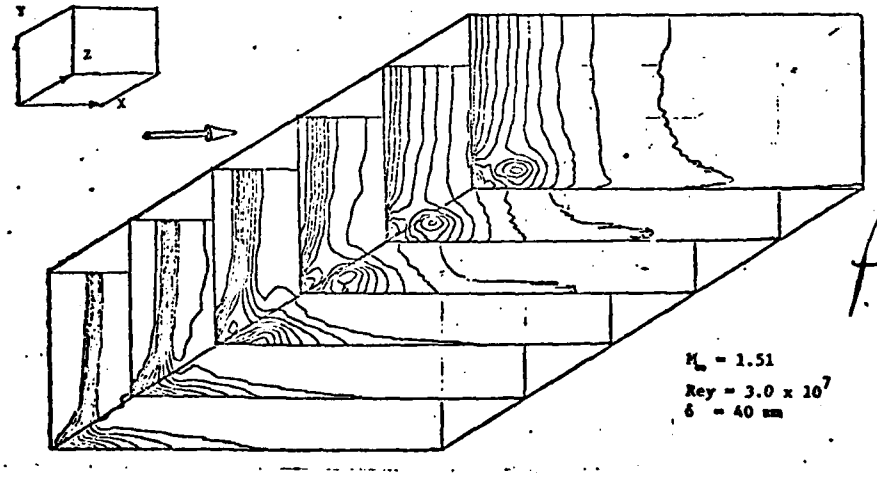

Figure 10. Perspective View of Density Contours

REFERENCES

1. Chapman, D. R., Dryden Lectureship in Research Computational Aerodynamics Development and Outlook, AIAA Paper 79-0129, January 1979.

2. Peyret, R. and Viviand, H. "Computation of Viscous Compressible Flows Based on the Navier-Stokes Equations," AGARDograph, No. 212, September 1975.

3. Knight, D. D., "Numerical Simulation of Realistic High-Speed Inlets Using the Navier-Stokes Equations," AIAA J., Vol. 16, June 1978.

4. Levy, L. L. "Experimental and Computational Steady and Unsteady Transonic Flow About a Thick Airfoil," AIAA J., Vol. 16, June 1978.

5. Mikhail, A. G., Hankey, W. L. and Shang, J. S., "Computation of a Supersonic Flow Past An Axisymmetric Nozzle Boattail with Jet Exhaust," AIAA Paper 78-993, July 1978.

6. Hung, C. M. and MacCormack, R. W. "Numerical Solution of Three-Dimensional Shockwave and Turbulent Boundary-Layer Interaction," AIAA J., Vol. 16, No. 10, October 1978.

7. Shang, J. S., Hankey, W. L. and Petty, J. S., "Numerical Solution of Supersonic Interacting Turbulent Flow Along a Corner," AIAA Paper 78-1210, July 1978.

8. Pulliam, T. H. and Lomax, H., "Simulation of Three-Dimensional Compressible Viscous Flow on the Illiac IV Computer," AIAA Paper 79-0206, January 1979.

9. "Future Computer Requirements for Computational Aerodynamics," A Workshop held at NASA Ames Research Center, Oct 4-6, 1977, NASA Conference Proceeding 2032.

10. J. S. Shang, Buning, P. G., Hankey, W. L., and Wirth, M. C., "The Performance of a Vectorized 3-D Navier-Stokes Code on the CRAY-1 computer, AIAA Paper 79-1448, 1979.

11. Buning, P. G., "Preliminary Report on the Evaluation of the CRAY-1 as a Numerical Aerodynamic Simulation Process," Presented at AIAA 3rd Computational Fluid Dynamics Conference, Open Forum, June 1977.

12. Smith, R. E. and Pitts, J. I., "The Solution of the Three-Dimensional Compressible Navier-Stokes Equations on a Vector Computer," Third IMACS International Symposium on Computer Methods for Partial Differential Equations," June 1979, Lehigh University, PA, and Private Communication.

13. MacCormack, R. W., "Numerical Solutions of the Interactions of a Shock Wave with a Laminar Boundary-Layer," Lecture Notes in Physics, Vol. 8, Springer-Verlag, 1971.

14. Calahan, D. A., "Performance of Linear Algebra Codes on the CRAY-1, "Proceedings SPE Symposium on Reservior Simulation, Denver, CO, 1979.

15. Abiss, J. B., East, L. F., Nash, C. R., Parker, P., Pike, E. R. and Swayer, W. G., "A Study of the Interaction of a Normal Shock-wave and a Turbulent Boundary Layer Using a Laser Anemometer," Royal Aircraft Establishment, England, TR 75151, February 1976.

16. Seddon, J., "The Flow Produced by Interaction of Turbulent Boundary-Layer with a Normal Shock Wave of Strength Sufficient to Cause Separation," Royal Aircraft Establishment, England, Rand M 3502, March 1960.

17. MacCormack, R. W. and Baldwin, B. S., "A Numerical Method for Solving the Navier-Stokes Equations with Application to Shock-Boundary Layer Interactions," AIAA Paper 75-1, January 1975.

18. Orbits, D. A., "A CRAY-1 Simulator," Report #118, Systems Engineering Laboratory Univ. of Michigan, September 1, 1978.

19. CRAY-1 Fortran (CFT) Reference Manual, Pub. #2240009, Cray Research, Inc., 1978.

20. Higbie, Lee, "Speeding Up Fortran (CFT) Programs on the CRAY-1," Technical Note Pub. #2240207, Cray Research, Inc., 1978.

21. Ames, W. G., Arya, S. and Calahan, D. A., "An Evaluation of the Fortran Compiller on the CRAY-1," Report #134, Systems Engineering Laboratory, University of Michigan, October 1, 1979.

22. Shum, J. R., "A Numerical Study of Transonic Normal Shock-Turbulent Boundary Layer Interactions," AIAA Paper 78-1170, July 1978 and Private Communication.