ARTICLE NO. 79-1448R

J80-207

Performance of a Vectorized Three-Dimensional Navier-Stokes Code on the CRAY-1 Computer

20005
20010

J. S. Shang*
Air Force Wright Aeronautical Laboratories, Wright-Patterson Air Force Base, Ohio
P. G. Buning†
University of Michigan, Ann Arbor, Mich.
and
W. L. Hankey‡ and M. C. Wirth§
Air Force Wright Aeronautical Laboratories, Wright-Patterson Air Force Base, Ohio

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
C_p	= specific heat at constant pressure
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, Eq. (15)
k	= exponent in the stretched coordinates y, z
L	= length scale of eddy-viscosity, Eq. (6)
L_ξ, L_η, L_ζ	= differencing operator
M	= Mach number
n	= outward normal of solid contour
P	= static pressure
Pr	= Prandtl number
Pr_t	= turbulent Prandtl number
$\dot{q}$	= rate of heat transfer
q	= magnitude of velocity
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, $(\rho, \rho u, \rho v, \rho w, \rho e)$
$\bar{u}$	= velocity vector
u, v, w	= velocity components in Cartesian frame
x, y, z	= coordinates in Cartesian frame
x_L, y_L, z_L	= length scales of Cartesian coordinates
δ_{ij}	= Kronecker delta
ϵ	= eddy-viscosity coefficient
ξ, η, ζ	= transformed coordinate system, Eq. (14)
μ	= molecular viscosity coefficient
ρ	= density
$\bar{\bar{\tau}}$	= stress tensor

Superscripts

$\bar{\ }$	= denotes vector
$=$	= denotes tensor
n	= denotes time level

Subscripts

∞	= property evaluated at the freestream condition
0	= stagnation condition

Presented as Paper 79-1448 at the AIAA 4th Computational Fluid Dynamics Conference, Williamsburg, Va., July 23-24, 1979; submitted Aug. 16, 1979; revision received Feb. 19, 1980. This paper is declared a work of the U.S. Government and therefore is in the public domain.

Index categories: Computational Methods; Nozzle and Channel Flow.

*Aerospace Engineer, Aerodynamics and Airframe Branch.
†Research Scientist; presently with NASA Ames Research Center.
‡Senior Scientist, Aerodynamics and Airframe Branch.
§Computational Physicist, Computer Support Branch.

I. Introduction

IN the past decade, computational fluid dynamics has become firmly established as a credible tool for aerodynamics research. Significant results have been obtained for the inviscid-viscous interactions, including flow separation and flows over arbitrary aerodynamic configurations for a wide range of Reynolds numbers and Mach numbers.[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. From the viewpoint of computational fluid dynamics, the obstacles involve several key issues such as: the efficiency of computational algorithms, the numerical resolution, the automatic grid-point generation, the turbulence modeling, and the supporting computing facilities. Substantial progress has been made in these interlocked problem areas, but continuous efforts are still in demand.[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 flowfield is required. Time-dependent Navier-Stokes equations,[3-8] parabolized Navier-Stokes equations,[10,11] boundary-layer approximations, or inviscid assumptions have been developed to satisfy special needs. Efforts to develop more efficient finite differencing schemes have lead to a group of im-

plicit[12,13] and hybrid procedures.[14-16] 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.

Several efforts on vectorization of computer codes with varying degrees of success have been reported in the open literature.[17-20] The potential of increased speed in data processing rate is clear. Basically, the objective of the vectorization of a computer code is to construct a long-ordered data string according to the computer structure to be processed by the vector registers and to achieve optimum data flow.

A three-dimensional, time-dependent Navier-Stokes code using MacCormack's explicit scheme[21] 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 capability, 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. Floating point operations per second (FLOPS) may be used as a criterion for the measure of the central process or unit speed. The asymptotic rate for processing long vectors on the CRAY-1 is 160×10^6 FLOPS.[22] The combination of the selected algorithm and the CRAY-1 computer provides a bench mark for future development and a tool for current engineering evaluation.

The problem selected for evaluating the CRAY-1 performance was the experimental investigations of Abbiss et al.[23] and Seddon[24] 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 a Mach number of 1.51. The primary purpose of this 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.

II. Governing Equations

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

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

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

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

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 ($Pr_t=0.9$). An algebraic two-layer model of the Cebeci-Smith type was used where

$$\epsilon_i=\rho(k_1L)^2\{1-[\exp(-\rho\left|\frac{\partial q}{\partial\eta}\right|/\mu L^2)^{1/2}/26]\}^2\left|\frac{\partial q}{\partial\eta}\right| \tag{4}$$

$$\epsilon_0=\rho k_2 q_{\max}\int_0^y \max\,(1-q/q_{\max})\mathrm{d}y \tag{5}$$

k_1 is the von Kármán constant (0.40), and k_2 assumes the value of 0.0168. The length scale L is the asymptotic form given by Gessner for a rectangular duct[25]

$$L=2yz/[y+z+(y^2+z^2)^{1/2}] \tag{6}$$

The components of shear stress and heat flux vector thus can be given as

$$\tau_{ij}=(\mu+\epsilon)(\mathrm{Def}\,\tilde{u})_{ij}-[2/3(\mu+\epsilon)(\nabla\cdot\tilde{u})+p]\delta_{ij} \tag{7}$$

$$\dot{q}_i=-C_p\left(\frac{\mu}{Pr}+\frac{\epsilon}{Pr_t}\right)\frac{\partial T}{\partial x_i} \tag{8}$$

The equation of state, Sutherland's viscosity law, and assigned molecular Prandtl number (0.73) formally close the system of governing equations.

Since the wind-tunnel flowfield 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 straightforward (Fig. 1). In order to develop upstream conditions equivalent to the experiment, a separate computation is initiated with a freestream 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).[23] Then, the computed flowfield at this streamwise location is imposed as the upstream 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:

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

Upstream condition:

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

Downstream:

$$\left.\frac{\partial\bar{U}}{\partial x}\right|_{x\to x_L}=0 \tag{11}$$

On planes of symmetry:

$$\left.\frac{\partial\bar{U}}{\partial y}\right|_{y=y_L}=0 \quad\text{and}\quad \left.\frac{\partial\bar{U}}{\partial z}\right|_{z=z_L}=0 \tag{12}$$

On wind tunnel wall:

$$u=v=w=0 \tag{13a}$$

$$T_w=313.79\ \mathrm{K} \qquad \text{at } y,z=0 \tag{13b}$$

$$\nabla\cdot\bar{\bar{\tau}}=0 \tag{13c}$$

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

$$\xi=x/x_L \tag{14a}$$

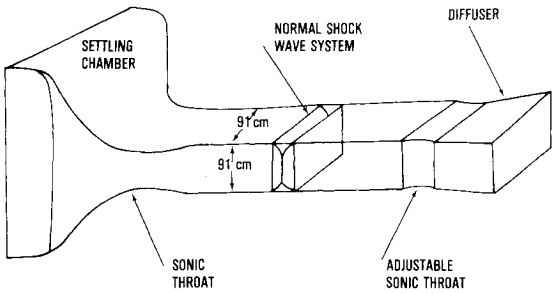

Fig. 1 Flowfield schematic.

$$\eta = 1/k\ell_n[1+(e^k-1)y/y_L] \tag{14b}$$

$$\zeta = 1/k\ell_n[1+(e^k-1)z/z_L] \tag{14c}$$

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

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

$$i=1,2,3 \tag{15}$$

where ξ_x, η_y, and ζ_z are the metrics of the coordinate transformation. For the present investigation, $\eta_x=\eta_z=\zeta_x=\zeta_y=0$. The definition of the conventional flux vectors $\bar{F}$, $\bar{G}$, and $\bar{H}$ can be found in Ref. 7.

III. CRAY-1 Architecture

In order to exploit the full capacity of the CRAY-1 computer, an understanding of the architecture is necessary. A detailed description of the central processing unit, memory section, and information of data flow can be found in Ref. 26. The control of data flow between the parallel functional units and hierarchically organized memories is of fundamental importance. The CRAY-1 memory section normally consists of 16 banks of bipolar 1024-bit LSI memory. The memory size can be as large as 1,048,576 words. Each word is 72 bits long and contains 64 data bits and 8 check bits. The memory cycle time is 50 ns (four clock periods). The access time (the time required to fetch an operand from memory to a scalar register) is 137.5 ns (11 clock periods). The maximum transfer rate for the intermediate register, intermediate scalar registers, and vector registers is 12.5 ns (one word per clock period). The address and operand data are transmitted along a single path between main and block register memory every clock period. The judicious usage of the relatively small block register memory essentially determines the effectiveness of the vector registers and allows concurrent operations (chaining) within one computer clock period. A major portion of the present effort was to organize the data storage and to reduce the total number of fetch/store data operations in the sweeping sequences.

The effect of I/O on program speed is minimized by overlapping data transfer with computations. CRAY-1 I/O section has twelve input and twelve output channels, so arranged that four simultaneous I/O operations can be performed, each at a rate of one word every four-clock period if there is no conflict. For each grid point per sweep, 18 words of I/O (five dependent variables, seven metrics, one eddy-viscosity coefficient, and the five updated dependent variables) must be executed. The buffered I/O and multiple independent I/O channels permit the concurrent I/O transfer and arithmetic operations. Based on our estimate, a $64\times47\times47$ grid-point system could have been solved on the one-million word memory unit of the CRAY-1 using the preceding procedure.

The vector processor generates results at rates greatly exceeding the rates of conventional scalar processing by performing floating point operations on sets of ordered data. The CRAY-1 has eight 64-element vector registers. All operands processed by the CRAY-1 are held in registers prior to and after being processed by the functional units. In general, the sequence of operations is to load one or more vector registers from memory and pass them to functional units. A result may be received by a vector register and reentered as an operand to another vector computation in the same clock period. The chaining of two or more vector operations allows the CRAY-1 to produce more than one result per clock period. Chain operation is automatically detected by the CRAY-1, but certain reordering of the code segments may improve the chain operation. A long vector which exceeds 64 elements is processed as one or more 64-element segments and a possible remainder of less than 64 elements. No rigid requirement is imposed to construct the vector loop.

From a code developer's viewpoint, the construction of a vectorized code hinges on how to avoid long or overly complicated loops, nonlinear indexing, and logical statement inside of a "do loop." A major improvement usually results if the scalar temporary variables which are encountered most frequently in the repetitious finite differencing scheme are replaced with the vector temporary variables. The recursive loop of which the output is propagated back into the input should be eliminated if the vector element is less than 64. The few simple rules suggested by Higbie were found to be extremely useful.[27]

IV. 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

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

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 Fig. 2. For a numerically simulated three-dimensional flowfield, the storage in two time-level memories of all dependent variables, metrics of the coordinate transformation, and the eddy-viscosity coefficient requires core storage usually exceeding the capacity of currently available computers. For vector processing, an even more stringent requirement on data organization is required.

In the scalar version of the code,[7] the core limitation on the CYBER 74 computer was remedied by organizing the data according to its streamwise location into planar storage (pages). Only the data in process were brought into the central memory core, while the remainder were retained on a random access disk file. Three pages of predictor level-dependent variables, four pages of corrector level-dependent variables, and two pages of transformation derivatives were required to process a planar sweep. The additional page for the corrector sweep is due to the numerical smoothing scheme originated by MacCormack.[28]

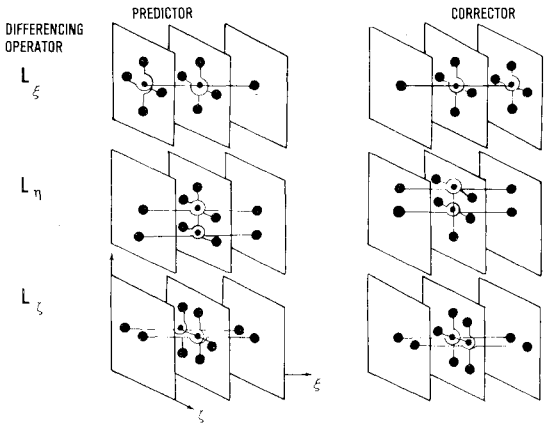

Fig. 2 Grid points involved in the time-step sweep.

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.

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

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 already degraded.

For the vector code, the organization of data must satisfy dual constraints. The dependent variables must be reduced to a minimum to satisfy an overall memory size restriction; and equally important, the partition of the immediately accessible data from the rest, which may reside on disk or other mass storage devices, must be optimized. The input/output traffic can significantly degrade the computer performance because the data flow to and from the disk or the mass storage requiring memory access is slower than the transfer from memory to the registers. The basic program structure is designed for a flowfield characterized by a dominant flow direction, and for the relatively short vector length requirement by the CRAY-1 vector registers, thus planar partitioning was chosen over the block partitioning. The relative merits have been discussed by Buning,[17] and will not be repeated here.

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 (Fig. 3). For a flowfield 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.

Once the planar or page storage is adopted, the vector length can be determined. Calculations can be performed over the complete page ($\eta-\zeta$ plane), thus yielding a vector length equal to the produce of grid points in the η and ζ coordinates. However, this particular arrangement may preclude a future requirement for a greater number of grid points to be retained in memory for vector temporary, i.e., immediately accessible vector storage. In addition, the CRAY-1 vector registers (and therefore vector operations) are limited to 64 elements, hence extremely long vector elements would not enhance the vectorization. Therefore, 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 ($\eta-\zeta$ plane), the streamwise sweep (ξ sweep) is vectorized in the ζ direction.

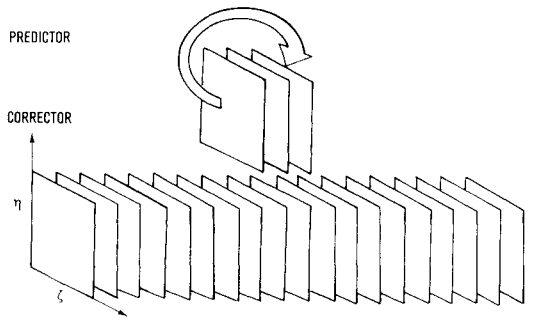

Fig. 3 Data storage and data flow diagram.

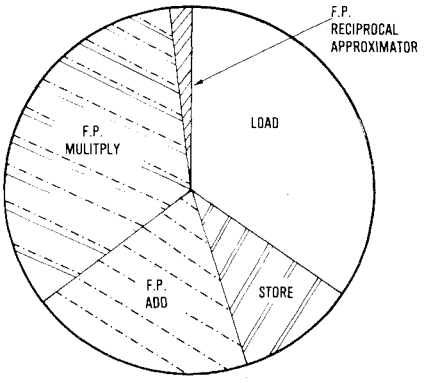

Fig. 4 Vector operation counts in percentage.

For the present problem, the computational domain with dimensions $356.3\times45.5\times45.5$ cm is partitioned into two streamwise sections of 64 pages each. Every page contains 33×33 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 inflow boundary condition for the following shock-boundary-layer interaction domain. Both contain $64\times33\times33$ 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 0.3958 of the local boundary-layer thickness (4.0 cm).[23] The crossflow plane grid-point distribution, however, remains identical 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%. In the leading computational domain, the convergence criterion is established similarly, but is based on the velocity profiles instead of pressure.

V. 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[17] and shown in Fig. 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 func-

Table 1 Comparative timing results

Computer	Version of code	RDP	$(RDP)_{CYBER\ 74}/RDP$
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
STAR 100	Vector (64 bit)	1.50×10^{-4}	49.9
STAR 100	Vector (32 bit)	6.00×10^{-5}	129.7

tional 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 than 2%. Access to a limited number of vector registers is required for temporary storage of intermediate results in vector operations. When these are not available, additional memory loadings result. 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.[17] Further improvements still can be made either in Fortran or assembly language versions of the present code. However, the authors feel an overall execution rate greater than 60 MFLOPS on this size problem is unlikely.

The timing comparison between the scalar and vectorized version of the same numerical algorithm is very important for the projection of future developments in computational fluid dynamics. Equally important, a comparison between several high-speed processors and the CRAY-1 on the three-dimensional aerodynamics simulation program is highly desirable in the evaluation of the vector performances. 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 resemblance to the original scalar code.[7,17] 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.

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 no. of grid points $\times$ total no. of iterations). The particular rate of data processing is most suitable for numerical programs with similar algorithms and convergence rates. If the ratio between field grid points and boundary points can be maintained between two programs, then the comparison is particularly meaningful.

The vectorized program outperforms the original scalar code by a factor of 8.13. In Table 1, the timing results of the scalar code and vectorized code performance for four different computers are given.

A brief description of each running condition for which the timing results were obtained may help with the interpretation of the data.[7] The computations conducted on CYBER 74 and CDC 7600 with a grid-point system of $17\times33\times33$ 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.

The comparison of timing results between the CRAY-1 and STAR 100 needs special attention. The vector code developed by Smith at NASA Langley Research Center[18] contains the full complement of the Jacobian of the coordinate transformation (9 metrics), thereby permitting more complex configurations to be simulated. The present code (7 metrics), however, is designed for a flowfield containing a definite orientation bias for aerodynamical engineering application. The aforementioned requirement in formulation definitely will lead to more computations to be performed than for the present simplification. This difference in code definitely contributes to the rather significantly lower data processing rate than the present result. For the explicit numerical scheme, the numerical results either by the 64- or 32-bit arithmetic are nearly identical.[18] Therefore, the accurate assessment of the rate of data processing between the CRAY-1 and STAR 100 should be comparable for aerodynamics engineering simulations.

VI. Comparisons with Experimental Data

Two experimental investigations[23,24] on the interaction of a normal-shock wave with a turbulent boundary layer were used to compare with the present vectorized CRAY-1 computations. The data of Seddon[24] were collected for the arrangement of a suspended flat plate and a shock generator. A substantial amount of the measurements within the boundary layer were recorded by a series of pressure probes. The velocity flowfield data of Abbiss et al.[23] however, were obtained by means of a laser anemometer. The test arrangement is simply a wind tunnel in which a normal-shock wave is held across the whole test section by an adjustable sonic throat positioned far downstream. This particular configuration is simulated by the present analysis. For the purpose of comparison, the combined usage of the two sets of data becomes necessary.

For the normal-shock wave and turbulent boundary-layer interactions, the only significant length scale is the turbulent boundary-layer thickness at the initial interception of the shock wave. The streamwise coordinate in the present effort is therefore presented in dimensionless units based upon this boundary-layer thickness.

In Fig. 5, several velocity profiles across the wind tunnel at a Reynolds number of 3.0×10^7 are presented. This location represents the flowfield condition at the end of the leading segment of the computational domain which is also the upstream condition for the following interaction zone. An orderly structure in these velocity distributions can be observed reflecting the symmetrical development of the viscous layer over the corner. In the plane normal to the axis of the wind tunnel and at a distance greater than the boundary-layer thickness (4 cm) away from the adjacent wall, the two-dimensional boundary-layer structure is clearly exhibited. The present results agree reasonably well with the data of

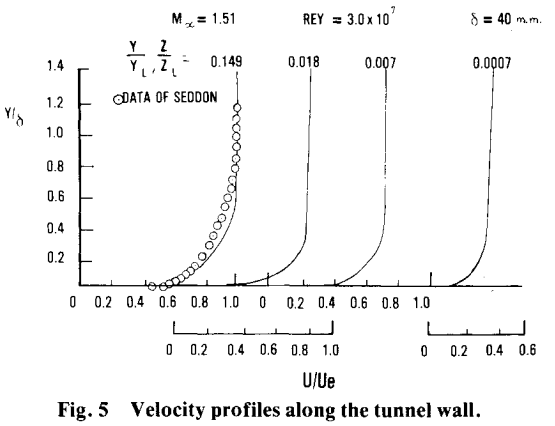

Fig. 5 Velocity profiles along the tunnel wall.

Seddon.[24] 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 vs 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[23] by about 8%.

A direct comparison of several velocity distributions between the data of Abbiss et al.[23] and the present calculation is presented in Fig. 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 the lambda wave structure. One of the factors contributing to the disparity is that the present variable mesh distribution does not match perfectly with the experimental data collecting location, thereby requiring an interpolation. Nevertheless, in general, the agreement between data and calculation is very good. The maximum disparity between data and calculation is about 10%.

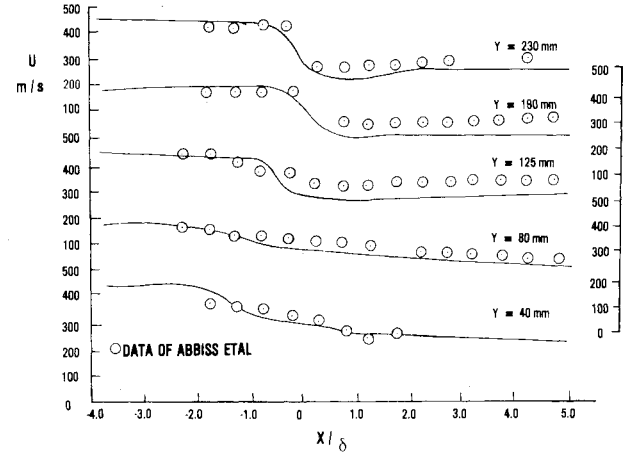

Fig. 6 Comparison of the flowfield velocity in the interaction region.

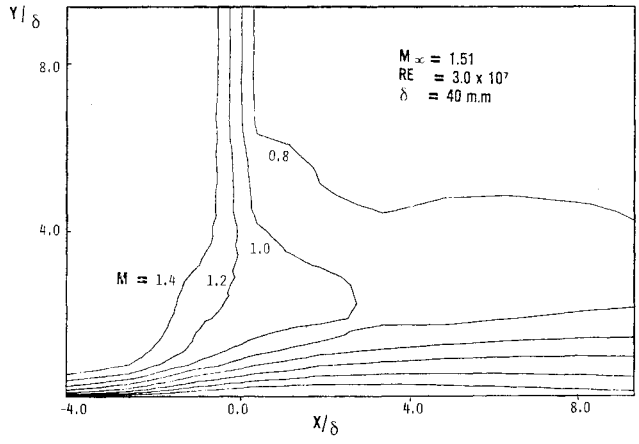

Fig. 7 Computed Mach number contour in the plane of symmetry.

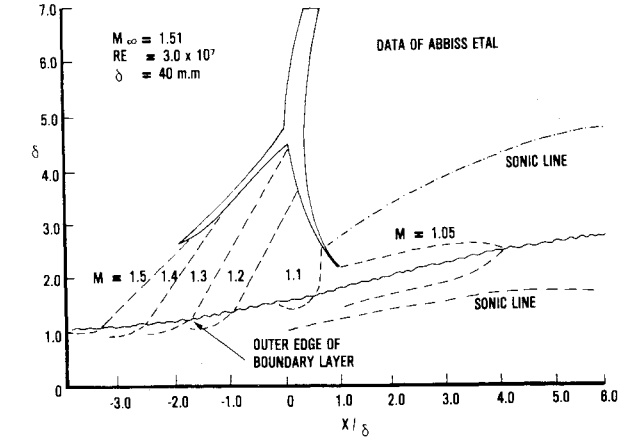

Fig. 8 Experimentally measured flowfield structure in the plane of symmetry.

In Fig. 7, the Mach number contour is presented in an attempt to compare with the flowfield structure given by Abbiss et al.[23] in Fig. 8. The bifurcation of the normal-shock wave is indicated. The strong viscous-inviscid zone extends about four undisturbed boundary-layer thicknesses upstream of the normal-shock wave. As the boundary layer thickens through the continuous compression of the lambda shock wave system, the local boundary layer exceeds twice the value of the initial boundary-layer thickness. One can detect the outward displacement of the Mach contour either from Fig. 7 or Fig. 8. The calculation nearly duplicates all of the 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 leads to the distinguishable discrepancy in the definition of the embedded supersonic zone. A similar observation may be made for the work of Shea[29] in his investigation of the two-dimensional normal-shock wave and turbulent boundary-layer interaction.

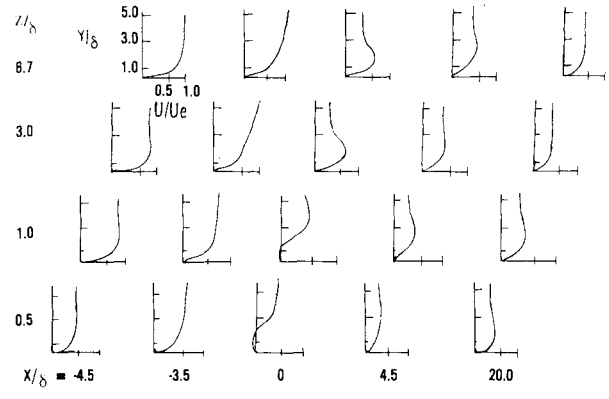

Fig. 9 Computed velocity field in the interaction region.

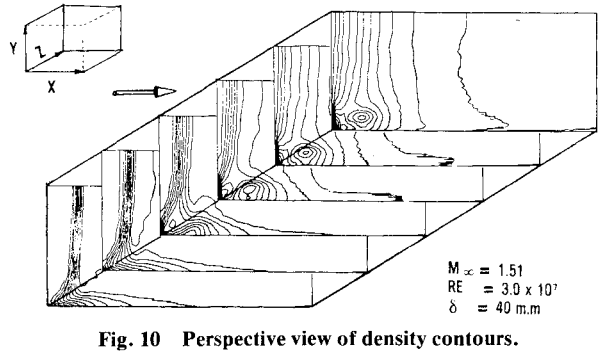

Fig. 10 Perspective view of density contours.

In Fig. 9, the velocity distribution parallel to the wind tunnel side walls is given. The velocity distribution near the plane of symmetry displays a structure similar to that found in Seddon's data,[24] however, for different test conditions. Therefore, quantitative comparisons cannot be made except to point out some of the salient features of the entire flowfield. 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[24] and the numerical simulation by Shea.[29]

The entire flowfield structure is presented in Fig. 10 in terms of density contours at various planes parallel to the side wall of the wind tunnel. The shear layer over the corner region, the strong inviscid-viscous domain, and the subsequent readjustment of the flowfield are easily detectable. A clear indication of substantial growth of the shear layer over the wind tunnel wall is also obvious.

VII. Conclusions

A three-dimensional, time-dependent Navier-Stokes code using MacCormack's explicit scheme has been vectorized for the CRAY-1 computer. The vectorized 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 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 of 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.

Acknowledgments

The authors wish to acknowledge the assistance of D. A. Calahan, W. Ames, P. Goshgarian, and E. Sesek of the University of Michigan in preparation of the CRAY-1 program. This work was performed in part under the auspices of Grant AF AFOSR 75-2812 while P.G. Buning was at the University of Michigan. The authors also wish to express their appreciation to CRAY Research, Inc., and Lawrence Livermore Laboratory for the use of their computer facility.

References

[1] Chapman, .R., "Dryden Lectureship in Research Computational Aerodynamics Development and Outlook," AIAA Paper 79-0129, Jan. 1979.

[2] Peyret, R. and Viviand, H., "Computation of Viscous Compressible Flows Based on the Navier-Stokes Equations," AGARDograph 212, Sept. 1975.

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

[4] Levy, L.L., "Experimental and Computational Steady and Unsteady Transonic Flow about a Thick Airfoil," *AIAA Journal,* Vol. 16, June 1978, pp.

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

[6] Hung, C. M. and MacCormack, R.W., "Numerical Solution of Three-Dimensional Shock Wave and Turbulent Boundary-Layer Interaction," *AIAA Journal,* Vol. 16, Oct. 1978, pp.

[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, Jan. 1979.

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

[10] Lin, T.C. and Rubin, S. G., "A Numerical Model for Supersonic Viscous Flow over a Slender Reentry Vehicle," AIAA Paper 79-0205, Jan. 1979.

[11] Schiff, L. B. and Steger, J. L., "Numerical Simulation of Steady Supersonic Viscous Flow," AIAA Paper 79-0130, Jan. 1979.

[12] Briley, W. R. and McDonald, H., "Solutions of the Three-Dimensional Compressible Navier-Stokes Equations by an Implicit Technique," *Lecture Notes in Physics,* Vol. 35, Springer-Verlag, New York, 1975, pp. 105-110.

[13] Warming, R. F. and Beam, R. M., "On the Construction and Application of Implicit Factored Schemes for Conservation Laws," *SIAM-AMS Proceedings,* Vol. 11, Nov., 1978.

[14] MacCormack, R. W., "A Rapid Solver for Hyperbolic Systems of Equations," *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics,* Springer-Verlag, New York, 1976, pp. 307-317.

[15] Li, C. P., "A Mixed Explicit-Implicit Method for the Compressible Navier-Stokes Equations," *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics,* Springer-Verlag, New York, 1976, p. 285.

[16] Shang, J. S., "Implicit-Explicit Method for Solving the Navier-Stokes Equations," *AIAA Journal,* Vol. 16, May 1978, pp. 496-502

[17] Buning, P. G., "Preliminary Report on the Evaluation of the CRAY-1 as a Numerical Aerodynamics Simulation Processor," presented at AIAA 3rd Computational Fluid Dynamics Conference, Open Forum, June 1977.

[18] 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 of Computer Methods for Partial Differential Equations,* Lehigh Univ., Pa., June 1979; also private communication.

[19] Keller, J. D. and Jameson, A., "Preliminary Study of the Use of the STAR 100 Computer for Transonic Flow Calculations," NASA TM 74086, Nov. 1977.

[20] Redhed, D.D., Chen, A.W., and Hotovy, S. G., "New Approach to the 3-D Transonic Flow Analysis Using the STAR 100 Computer," *AIAA Journal,* Vol. 17, Jan. 1979.

[21] 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, New York, 1971, pp. 151-163.

[22] Calahan, D. A., "Performance of Linear Algebra Codes on the CRAY-1," *Proceedings of the SPE Symposium on Reservoir Simulation,* Denver, Col., June 1979.

[23] Abbiss, 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 75141, Feb. 1976.

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

[25] Gessner, F. B. and Po, J. K., "A Reynolds Stress Model for Turbulent Corner Flows, Pt. I: Comparisons between Theory and Experiment," *Journal of Fluid Engineering, Transactions of ASME,* Vol. 98, Ser. 1, No. 2, June 1976.

[26] *CRAY-1 Computer System,* Hardware Reference Manual 2240004, CRAY Research, Inc., Minn.

[27] Higbie, L., "Speeding up Fortran (CFT) Programs on the CRAY-1," CRAY Research, Inc., Minn., Pub. No. 2240407, Oct. 1978.

[28] 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, Jan. 1975.

[29] Shea, J. R., "A Numerical Study of Transonic Normal Shock-Turbulent Boundary-Layer Interactions," AIAA Paper 78-1170, July 1978.