General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

DAA / LANGLEY

OLD DOMINION UNIVERSITY RESEARCH FOUNDATION

DEPARTMENT OF MECHANICAL ENGINEERING AND MECHANICS
SCHOOL OF ENGINEERING
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA

NUMERICAL SOLUTIONS OF THREE-DIMENSIONAL NAVIER-STOKES EQUATIONS FOR CLOSED BLUFF-BODIES

By

Jamshid S. Abolhassani

and

S. N. Tiwari, Principal Investigator

Progress Report
For the period ending December 31, 1984

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Research Grant NCC1-68
Dr. Robert E. Smith, Jr., Technical Monitor
ACD-Computer Science and Applications Branch

(NASA-CR-175755) NUMERICAL SOLUTIONS OF 3-DIMENSIONAL NAVIER-STOKES EQUATIONS FOR CLOSED BLUFF-BODIES Progress Report (Old Dominion Univ., Norfolk, Va.) 19 p HC A02/MF A01 CSCL 12A G3/64 N85-26255 Unclas 21184

April 1985

DEPARTMENT OF MECHANICAL ENGINEERING AND MECHANICS
SCHOOL OF ENGINEERING
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA

NUMERICAL SOLUTIONS OF THREE-DIMENSIONAL NAVIER-STOKES EQUATIONS FOR CLOSED BLUFF-BODIES

By

Jamshid S. Abolhassani

and

S. N. Tiwari, Principal Investigator

Progress Report
For the period ending December 31, 1984

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

Under
Research Grant NCC1-68
Dr. Robert E. Smith, Jr., Technical Monitor
ACD-Computer Science and Applications Branch

Submitted by the
Old Dominion University Research Foundation
P.O. Box 6369
Norfolk, Virginia 23508

April 1985

FOREWORD

This is a progress report on the research project "Numerical Solutions of Three-Dimensional Navier-Stokes Equations for Closed-Bluff Bodies." The period of performance on this research was January 1 through December 31, 1984. The work is supported by the NASA/Langley Research Center through Cooperative Agreement NCC1-68, and monitored by Dr Robert E. Smith, Jr., of the Analysis and Computation Division (Computer Science and Application Branch), NASA/Langley, MS/125.

TABLE OF CONTENTS

	Page
FOREWORD	ii
SUMMARY	1
1. INTRODUCTION	2
2. GOVERNING EQUATIONS	4
3. METHOD OF SOLUTION	8
4. INITIAL AND BOUNDARY CONDITIONS	11
5. POST PROCESSING OF DATA AND DATA DISPLAY	13
6. PRESENT ACCOMPLISHMENTS	14
REFERENCES	15

NUMERICAL SOLUTIONS OF THREE-DIMENSIONAL NAVIER-STOKES EQUATIONS FOR CLOSED-BLUFF BODIES

By

J. S. Abolhassani[1] and S. N. Tiwari[2]

SUMMARY

With the present facilities at NASA/Langley Research Center, it is economically feasible to compute the three-dimensional flow about a complex configuration such as closed-bluff bodies (e.g., circular and elliptical cylinders) on a flate plate. If the body is sufficiently bluffed, there will exist a three-dimensional region close to and about the junction of the body with the surface.

In the present study, the Navier-Stokes equations are solved numerically. These equations are unsteady, compressible, viscous, and three-dimensional without neglecting any terms. The time dependency of the governing equations allows the solution to progress naturally for an arbitrary initial guess to an asymptotic steady state, if one exists. The equations are transformed from physical coordinates to the computational coordinates, allowing the solution of the governing equations in a rectangular parallelepiped domain. The equations are solved by the MacCormack time-split technique which is vectorized and programmed to run on the CDC VPS 32 computer. The codes are written in 32-bit (half word) FORTRAN, which provides an approximate factor of two decreasing in computational time and doubles the memory size compared to the 64-bit word size.

---

[1]Graduate Research Assistant, Department of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia 23508.
[2]Eminent Professor, Department of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia 23508.

1. INTRODUCTION

The comprehension and analysis of three-dimensional fluid behavior around bluff bodies is of considerable importance for flight applications. The primary motivation to solve such problems is to understand the basic phenomenon of separated flows and to determine the associated forces on the object. The static and dynamic features of the unsteady flow field around the bluff-bodies are of interest to the designers as well. These flows are three-dimensional in the most realistic cases, and can be described by a set of Navier-Stokes equations. These equations can be solved numerically if proper grid distributions can be generated. This step (grid generation) is the essential first step in solving Navier-Stokes equations for complex configurations. There are presently only a few three-dimensional solutions of the Navier-Stokes equations documented, and the obvious reason for this is the complexity of the grid arrangements, numerical coding, and lack of sufficient numerical resolution (i.e., memory) to generate credible results. Most of the available solutions are for simple three-dimensional flow (e.g., three-dimensional corner, spherical dome, conical bodies, and spike-nosed bodies) which require simple grid arrangements. For many complex geometries, the constraints imposed by digital computers are less restrictive for two-dimensional computation than for three-dimensional ones. One major difference between numerical simulation of three-dimensional and two-dimensional flow is the overwhelming discrepancy in the data base. For a mesh with 100 points on a side, a two-dimensional problem requires a memory of about 80,000 words (counting 8 dependent variables at each mesh point in a typical solution of a compressible flow with arbitrary geometry), while a three-dimensional problem requires a memory of about 14,000,000 (14 variables) words. The former can be processed easily by any modern scientific

computer, however, this is certainly not possible for the latter case.

In order to study three-dimensional flows, the Navier-Stokes equations are being solved numerically. These equations are unsteady, compressible, viscous, and three-dimensional without neglecting any terms. The time dependency of governing equations allows the solution to progress naturally from an arbitrary initial guess to an asymptotic steady state, if one exists. The equations are transformed from physical coordinates to the computational coordinates, allowing the solution of the governing equations in a rectangular parallelepiped domain. The equations are solved by the MacCormack time-split technique which is vectorized and programmed to run on the CDC VPS 32 computer. The codes were written in 32-bit (half-word) FORTRAN, which provides an approximate factor of two decrease in computer time and doubles the memory compared to the 64-bit word size. Solutions will be obtained for an infinite right circular cylinder on a flat plate at high Mach and Reynolds numbers. In this study, the computational planes are perpendicular to the axis of the cylinder rather than the flow direction. Therefore, updating the boundary and initial conditions become more complex.

The algebraic method is used to generate the grids and they are distributed exponentially in the vertical direction (parallel to the axis of the cylinder). Grids are concentrated near solid boundaries and in the vicinity of the cylinder-plate junction. The circular cylinder geometry would require about 200,000-300,000 grid points and this will occupy 3-4.5 million 32-bit words of primary memory. It would take about 4-7 seconds to complete a time step of explicit MacCormack method. Also, it would take 3-4.5 seconds to transfer the restart file from primary memory to secondary memory.

2. GOVERNING EQUATIONS

The governing equations for a thermal fluid system are the conservation of mass, momentum, and energy. These equations are developed for an arbitrary region assuming the system is in continuum. Equations of motion for viscous, compressible, unsteady, heat conducting flow can be written as:

Continuity $$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \bar{u}) = 0, \tag{2.1}$$

Momentum: $$\frac{\partial (\rho \bar{u})}{\partial t} + \nabla \cdot (\rho \bar{u}\bar{u} - \tau) = 0, \tag{2.2}$$

Energy: $$\frac{\partial (E)}{\partial t} + \nabla \cdot (E\bar{u} + \bar{q} - \bar{u} \cdot \tau) = 0. \tag{2.3}$$

where E is the total energy per unit volume given by $E = \rho\ (e + \frac{V^2}{2} +$ potential energy + ...) and e is the internal energy per unit volume. For Newtonian fluid, stress tensor can be related to the pressure and velocity components as:

$$\tau_{ij} = - P\ \delta_{ij} + \mu\ [(\frac{\partial U_i}{\partial X_j} + \frac{\partial U_j}{\partial X_i}) - \frac{2}{3}\ \delta_{ij} \frac{\partial U_k}{\partial X_k}] . \tag{2.4}$$

This equation is valid under negligible bulk viscosity. For an isotropic system, the heat flux in Eq. (2.3) can be expressed in terms of temperature gradient (Fourier's law of heat conduction) as:

$$q = - K \nabla T \tag{2.5}$$

where K is a coefficient of thermal conductivity. A common approximation

used for viscosity is based on the kinetic theory of gases using an idealized intermolecular-forces potential, the relation is:

$$\frac{\mu}{\mu_r} = \left(\frac{T}{T_r}\right)^{3/2} \frac{T_r + S_o}{T + S_o} \tag{2.6}$$

where

$$S_o = 198.6° \text{ R}$$

$$\mu_r = 0.1716 \text{ np}$$

coefficients of thermal conductivity K can be determined from Prandtl number

$$K = \frac{\gamma \mu C_\upsilon}{Pr} \tag{2.7}$$

where C_υ is the specific heat at constant volume and γ is the ratio of specific heats.

It is necessary to have a supplementary relation to close the system of equations (Eq. (2.1) - (2.3)). By neglecting intermolecular forces (a thermally perfect system), thermodynamic properties can be related as:

$$P = \rho RT \tag{2.8}$$

where R is the gas constant. Thermally perfect gas assumption allows expression of the internal energy (e) as a function of T only [$e = e(T)$]. In addition, assumption of calorically perfect gas [$e(0) = 0$] allows the following relation:

$$e = C_v T \tag{2.9}$$

A combination of Eqs. (2.8) and (2.9) results in

$$P = \rho e\ (\gamma - 1) \tag{2.10}$$

These equations (Eqs. (2.1) - (2.3)) are in conservative form. For simplicity, these equations can be expressed into a compact vector form as:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z} = 0, \tag{2.11}$$

where

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}, \qquad F = \begin{bmatrix} \rho u \\ \rho uu - \tau_{xx} + P \\ \rho uv - \tau_{xy} \\ \rho uw - \tau_{xz} \\ Eu + \dot{q}_x - \phi_x + Pu \end{bmatrix},$$

$$G = \begin{bmatrix} \rho v \\ \rho uv - \tau_{xy} \\ \rho vv - \tau_{yy} + P \\ \rho vw - \tau_{xz} \\ Ev + \dot{q}_y - \phi_y + Pv \end{bmatrix}, \qquad H = \begin{bmatrix} \rho w \\ \rho uw - \tau_{xz} \\ \rho vw - \tau_{yz} \\ \rho ww - \tau_{zz} + P \\ Ew + \dot{q}_z - \phi_z + Pw \end{bmatrix}.$$

For the sake of generality, we can transform these equations from a physical domain to a computational domain as

$$\frac{\partial U}{\partial t} + \begin{bmatrix} \xi_x \\ \xi_y \\ \xi_z \end{bmatrix} \left(\frac{\partial F}{\partial \xi}, \frac{\partial G}{\partial \xi}, \frac{\partial H}{\partial \xi} \right) + \begin{bmatrix} \eta_x \\ \eta_y \\ \eta_z \end{bmatrix} \left(\frac{\partial F}{\partial \eta}, \frac{\partial G}{\partial \eta}, \frac{\partial H}{\partial \eta} \right) \tag{2.12}$$

$$+ \begin{bmatrix} \zeta_x \\ \zeta_y \\ \zeta_z \end{bmatrix} \left(\frac{\partial F}{\partial \zeta}, \frac{\partial G}{\partial \zeta}, \frac{\partial H}{\partial \zeta} \right) = 0$$

The transformation coefficients can be computed from a functional relation between the computational coordinates and the physical coordinates.

$$x = x(\xi,\eta,\zeta), \quad y = y(\xi,\eta,\zeta), \quad \text{and} \quad z = z(\xi,\eta,\zeta). \tag{2.13}$$

$$\xi = \xi(x,y,z), \quad \eta = \eta(x,y,z), \quad \text{and} \quad \zeta = \zeta(x,y,z). \tag{2.14}$$

If Eq. (2.14) is known, the transformation coefficients can be computed by direct differentiation. If the former relation is not known, after some algebraic manipulation, the transformation coefficients can be computed by:

$$\begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} & \frac{\partial \xi}{\partial z} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} & \frac{\partial \eta}{\partial z} \\ \frac{\partial \zeta}{\partial x} & \frac{\partial \zeta}{\partial y} & \frac{\partial \zeta}{\partial z} \end{bmatrix} = J = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{bmatrix}^{-1} \tag{2.15}$$

where

$$J = \frac{1}{|J^{-1}|} \begin{bmatrix} \left(\frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \eta}\right) & -\left(\frac{\partial x}{\partial \eta}\frac{\partial z}{\partial \zeta} - \frac{\partial x}{\partial \zeta}\frac{\partial z}{\partial \eta}\right) & \left(\frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \zeta} - \frac{\partial x}{\partial \zeta}\frac{\partial y}{\partial \eta}\right) \\ -\left(\frac{\partial y}{\partial \xi}\frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \xi}\right) & \left(\frac{\partial x}{\partial \xi}\frac{\partial z}{\partial \zeta} - \frac{\partial x}{\partial \zeta}\frac{\partial z}{\partial \xi}\right) & -\left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \zeta} - \frac{\partial x}{\partial \zeta}\frac{\partial y}{\partial \xi}\right) \\ \left(\frac{\partial y}{\partial \xi}\frac{\partial z}{\partial \eta} - \frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \xi}\right) & -\left(\frac{\partial x}{\partial \xi}\frac{\partial z}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial z}{\partial \xi}\right) & \left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}\right) \end{bmatrix} \quad (2.16)$$

$$|J^{-1}| = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \zeta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \zeta} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \zeta} \end{vmatrix}$$

$$= \frac{\partial x}{\partial \xi}\left(\frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \eta}\right) - \frac{\partial x}{\partial \eta}\left(\frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial z}{\partial \zeta}\right)$$

$$+ \frac{\partial x}{\partial \zeta}\left(\frac{\partial y}{\partial \xi}\frac{\partial z}{\partial \eta} - \frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \xi}\right)$$

In the present case, the planes of grid are parallel to x-direction thus allowing us to write:

$$x = x(\xi) \quad (2.17)$$

$$y = y(\eta,\zeta) \quad (2.18)$$

$$z = z(\eta,\zeta) \quad (2.19)$$

This reduces the metric coefficient from nine to five non-zero elements.

3. METHOD OF SOLUTION

A time marching method is used to compute the solution. This allows us to capture the possible transient feature. This method is an explicit

second-order accurate time-split predictor-corrector algorithm [3]. The governing equations (Eq. (2.12)) are discretized in computational directions. In a compact form, they can be expressed as

$$U^{n+1}_{i,j,k} = \left[L_\eta(\Delta t_\eta)\right] \left[L_\zeta(\Delta t_\zeta)\right] \left[L_\xi(\Delta t_\xi)\right] \left[L_\zeta(\Delta t_\zeta)\right] \left[L_\eta(\Delta t_\eta)\right] U^n_{i,j,k} \qquad (3.1)$$

where

$$\Delta t_\eta = \Delta t_\zeta = \frac{1}{2} \Delta t_\xi$$

and L_ξ, L_η, and L_ζ are the operators in ξ, η, and ζ directions, respectively. A time step is completed in this algorithm with the application of each operator applied symmetrically about the middle operator. For example, operator L_ξ can be defined as

$$L_\xi(\Delta t_\xi) = U^{out}_{i,j,k} \qquad (3.2)$$

where

<u>Predictor step:</u>

$$\overline{U}_{i,j,k} = U^{in}_{i,j,k} - \frac{\Delta t_\xi}{\Delta \xi} \left[(F_i - F_{i-1}) \frac{\Delta \xi}{\partial x} i + (G_i - G_{i-1}) \frac{\partial \xi}{\partial y} i \right.$$

$$\left. + (H_i - H_{i-1}) \frac{\partial \xi}{\partial z} i \right]_{j,k}$$

Corrector step:

$$U^{out}_{i,j,k} = \frac{1}{2}\left(U^{in}_{i,j,k} + \overline{U}_{i,j,k} - \frac{\Delta t_\xi}{\Delta\xi}\left[(F_{i+1} - F_i)\frac{\partial\xi}{\partial x}\,i + (G_{i+1} - G_i)\frac{\partial\xi}{\partial y}\,i + (H_{i+1} - H_i)\frac{\partial\xi}{\partial z}\,i\right]_{j,k}\right)$$

This method has a time step stability limit, but there is no rigorous stability analysis available for this. A conservative time step that is commonly used is

$$\Delta t < \min\left[\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + \frac{|w|}{\Delta z} + c\sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} + \frac{1}{\Delta z^2}}\right]^{-1} \qquad (3.3)$$

where c is the local speed of sound.

In the supersonic region, there exists a large gradient which requires a very fine mesh to resolve it. If they are not resolved, they produce a large oscillation which eventually blows up the solution. These oscillations of "low frequency" can be suppressed by adding a fourth order dampening. A common dampening used is the pressure dampening. This can be expressed in physical coordinates as

$$-\overline{\alpha}_\ell \Delta t_\ell \delta_\ell^3 \frac{\partial}{\partial\delta_\ell}\left[\frac{|v_\ell| + c}{4P}\left|\frac{\partial^2 P}{\partial^2\delta_\ell}\right|\frac{\partial U}{\partial\delta_\ell}\right] \qquad \ell = 1,2,3 \qquad (3.4)$$

where $\delta_1 = \xi$, $\delta_2 = \eta$, and $\delta_3 = \zeta$.

4. INITIAL AND BOUNDARY CONDITIONS

In computational fluid dynamics the initial conditions usually correspond to a real initial situation for a transient problem, or a rough guess for a steady state problem. In practice, initial conditions are obtained from experiments, empirical relations, approximation theories, or previous computational results. An inappropriate initial guess may result in generating unphysically strong transient waves which propagate through the computational region dominating the flow field and eventually lead to a solution failure. In general, there are two important requirements that should be considered in the choice of initial conditions. First, they should be compatible with the fixed upstream boundary conditions. Secondly, the initial conditions should be as physically close as possible to the actual nature of the flow field in the region under study. The former will minimize the number of iterations required for convergence. An attractive approach is to initialize the entire flow field (including the upstream boundary and the body surface) with a crude and simple guess (e.g., free stream condition). Then, during the course of the computation, both body and upstream boundary conditions are changed in a gradual manner to their final values over a prescribed number of iterations. The former approach is applied in only one step which is equivalent to impulsive initial conditions.

It is equally important to implement a realistic, accurate, and stable method to determine boundary conditions. The application of certain conditions may cause numerical instability even though the flow is physically stable. There are neither mathematical nor physical justifications to implement a realistic boundary condition. Most of the boundary conditions currently implemented are drawn mainly upon intuition, wind tunnel experience, and computational experimentation. There are three general types of

boundary conditions. They are Dirichlet conditions (specified function value), Neumann conditions (specificed normal gradient), and Robin conditions (a combination of both). Four important factors should be considered in the selection of bundary conditions. They are convergence, stability, computer time, and above all the physical justification.

For this problem there are five different boundary conditions. They are upstream, downstream, lateral, top, and solid boundary. The upstream boundary conditions are the undisturbed free stream conditions and are located at a grid space away from the leading edge, i.e.,

$$\bar{U}\Big|_{\text{upstream}} = \bar{U}_{\infty} \tag{4.1}$$

A zero gradient in y-direction (parallel to the primary direction of flow) is assumed for the downstream boundary, i.e.,

$$\frac{\partial \bar{U}}{\partial y}\Bigg|_{\text{downstream}} = 0 \tag{4.2}$$

The lateral boundaries are located far enough to avoid any influence on the interaction region. A boundary-layer profile can be prescribed on the lateral boundaries. These profiles can be obtained from their corresponding points of a flow over a flat pate. Presently, a zero gradient in z-direction is assumed for these boundaries, i.e.,

$$\frac{\partial \bar{U}}{\partial z}\Bigg|_{\text{lateral}} = 0 \tag{4.3}$$

The cylinder is assumed infinite in height, therefore, the flow at the top of the cylinder would be two-dimensional. Consequently, a zero-gradient boundary condition is imposed in x-direction for the top boundaries, i.e.,

$$\left.\frac{\partial \bar{U}}{\partial x}\right|_{Top} = 0 \tag{4.4}$$

The wall is assumed impermeable and no slip boundary conditions are applied, therefore, all velocity components are assumed to be zero. The wall is also assumed to have a constant temperature Tw. A zero normal pressure gradient is assumed for the solid surface, i.e.,

$$\left.\frac{\partial P}{\partial n}\right|_{solid} = 0 \tag{4.5}$$

This evaluation may appear to be based on the boundary-layer approximation (zero normal pressure gradient). In fact, it is a much milder approximation, since constant pressure is not applied through the boundary layer but over one grid line in the boundary layer. This approximation has yielded stable computations for both non-separated and separated boundary layers [2].

5. POST PROCESSING OF DATA AND DATA DISPLAY

The solution of a three-dimensional flow can produce upto 1,000,000 data. This vast amount of data needs to be analyzed and necessary information extracted. Storage and manipulation of the huge amount of data becomes a serious problem. Color computer graphics provide adequate solutions to

the problem. Data which sometimes is over 1,000 pages of computer print-out can be compacted and displayed as a digital image on color monitors. Color computer graphics offer several advantages for technical presentations, documentations, and easy and rapid analysis. A digital image of a flow field variable (e.g. density) can be created by transforming node points from the computational grids into the object space to the image plane and fill the void between the image node point with a color that varies according to the magnitude of the variables.

6. PRESENT ACCOMPLISHMENTS

The equations of motion are transformed from the physical coordinates to the computational coordinates allowing the solution of the governing equations in a rectangular parallelepiped domain. The equations are being solved by the MacCormack time-split technique which is vectorized and programmed to run on the CDC VPS 32 computer. The codes are written in 32-bit (halfword) FORTRAN, which has provided an approximate factor of two decrease in computer time and doubles the memory size compared to the 64-bit word size.

The algebraic method is used to generate a O-type grid around the cylinder under study, and they are distributed exponentially in the vertical direction (parallel to the axis of the cylinder). Grids are concentrated near solid boundaries and in the vicinity of the cylinder plate junction. Further accomplishments will be reported in a subsequent progress report.

REFERENCES

1. Smith, R.E., "Two-Boundary Grid Generation for the Solution of the Three-Dimensional Compressible Navier-Stokes Equations," NASA Technical Memorandum 83123, May 1981.

2. Rache, P.J., Computational Fluid Dynamics, Hermosa publisher, 1972.

3. MacCormack, R.W. and Bladwin, B.S., "A Numerical Method for Solving the Navier-Stokes Equations with Application to Shock-Boundary Layer Interactions," AIAA Paper 75-1, AIAA 13th Aerospace Sciences Meeting, Pasadena, CA, January 20-22, 1975.