THE UNIVERSITY OF KANSAS

COMPUTATIONAL ANALYSIS FOR TIME DEPENDENT TWO DIMENSIONAL FLOW OF AN INCOMPRESSIBLE HOMOGENEOUS NEWTONIAN FLUID (SPIN UP PROBLEM)

by

KWANG S. KIM

B.S.M.E., Hanyang University, 1978, Korea

October 1982

Submitted to the Department of Mechanical Engineering and the Faculty of the Graduate School of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science

Redacted Signature

Professor in Charge

Redacted Signature

Redacted Signature

Committee Members

Redacted Signature

For the Department

ABSTRACT

A digital computer method and a program which employs optimized successive overrelaxation were developed to solve certain problems involving time-dependent two-dimensional flow of an incompressible homogeneous Newtonian fluid.

Bubble sort techniques were used in a boundary field and a vectorization of the inner-most Do-loops of the program was carried out using a CRAY-1 computer. A polygon region was selected for analysis purposes.

As a test program, "Spin-up problem" for a circular region with rotating solid boundary conditions was analyzed. The results of computations using a fine mesh were in good agreement with analytical results.

Finally, a flow contained in an infinite cylinder with part-moving boundaries is examined. It is to be applied to an analysis of a flow in a circular cavity.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES (continued)

LIST OF FIGURES (continued)

NOMENCLATURE

a	radius of curvature, ft.
const.	constant
E	allowable error
i	spatial location
j	spatial location
k	a constant
L	characteristic length of the problem, ft.
m	positive integer
n	time level
p	fluid pressure, lbf./ft^2.
P	a point
$\vec{q}$	fluid speed, ft./sec.
r	polar radius, ft.
Re	Reynolds number
s	distance along curve, ft.
t	time, sec.
$\vec{u}$	fluid velocity, ft./sec.
u	horizontal fluid velocity component, ft./sec.
U	fluid velocity parallel to the wall, ft./sec.
U_o	characteristic velocity of the problem, ft./sec.
U_p	fluid velocity, U, at the point P,ft./sec.
U_w	fluid velocity, U, at the wall, ft./sec.
v	vertical fluid velocity component, ft./sec.
x	horizontal coordinate
y	vertical coordinate

Greek Letters and Special Characters

β	aspect ratio
γ	unit weight of a fluid, lb./ft^3.
δ	grid space
∇	vector operator $(\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$ for two-dimensional problem
ϵ	thin layer of a fluid thickness, ft.
θ	polar angle in circular flow
θ_1	angular horizontal geometric coordinate
λ	mutual overlap, ft.
μ	dynamic (absolute) viscosity of a fluid, lb. sec./ft^2.
ν	kinematic viscosity of a fluid, ft^2./sec.
π	a polygon
ρ	fluid density, lbm./ft^3.
σ	boundary of a region of a two dimensional flow
ψ	stream function, ft^2./sec.
ψ_p	stream function at point P, ft^2./sec.
ψ_w	stream function at the wall, ft^2./sec.
$\forall$	"at all points"
ξ	vorticity, sec^{-1}

ACKNOWLEDGEMENT

I would like to express my sincere thanks to the people and advisors who have provided me with valuable help throughout this research. I would like to thank Dr. Charles Groth who originated this study, and provided great assistance. I also extend my deep appreciation to Dr. Bedru Yimer, Dr. John Crisp, and Dr. Louis Burmeister for their help and guidance in completing this work. Invaluable run time on a CRAY-1 computer was provided by United Computing Systems, Kansas City, Missouri. I want to thank the Ministry of National Defence of Korean Government, who has supported me in various ways.

Finally, I wish to thank my parents and my wife, Jin for their motivation and support. The completion of this research would not have been possible without the patience and encouragement of all the people mentioned.

CHAPTER 1

INTRODUCTION

The computational techniques of a computer program which seems to be effective for time-dependent two-dimensional viscous flow problems will be developed. The analysis is applied to a polygon region with solid moving or fixed boundary conditions. The method is outlined in Chapter 3, and more details concerning the test program are given in Chapter 4.

Mathematically, the problem is that of solving numerically a system of nonlinear partial differential equations. It is known that, even without the added complications of time variation or non-linearity, numerical techniques for multi-order equations tend to require substantial computer time. The presence of non-linear terms may not only accentuate the computer time problems, but may also tend to induce computational instabilities.

As might be expected from their potential importance, viscous flow computations have received considerable attention in the literature (Fromm& Harlow (4) ; Dix (2) ; Hellums & Churchill (5) ; Wilkes (15) and Thompson (14) ; Pearson (9), etc). In general, it appears that there have been difficulties with one or another of computational speed, stability, accuracy, or the ability to handle correct boundary conditions. In a problem of this general difficulty, it seems worth while to begin with the simplest possible problem possessing realistic boundary conditions. In this research, a rotating flow contained in an infinite cylinder, with moving boundaries (test program) and part-moving boundaries is treated.

In the process of developing computational techniques and a computer program, the treatment of boundary fields, which are very important in recent computational fluid dynamics is utilized. By employing the curvature term in the formula of Thom (12) and the computer programming algorithm (Bubble Sort, etc), the accurate calculations of the vorticities and stream functions are resolved and are in good agreement with those of analytical works in a test program. A vectorization in a relaxation loop which is the most time-consuming operation and in the other inner-most Do-loops results in an increase in speed by a factor of three on the CRAY-1 Computer. As an application, which can be used for the analysis of a flow in a circular cavity with part of its wall moving, the problems for the fluid in a part-rotating infinite cylinder of radius, a, are investigated. These computations are converged and stabilized after reaching the time desired, even though the exact analytical results are unknown. Throughout this research, calculations of pressure are omitted.

CHAPTER 2

GOVERNING EQUATIONS

In terms of vector notations and time $\bar{t}$, the Navier-Stokes equations are derived from Newton's second law of motion in the case of incompressible fluids ($\bar{\rho}$ and $\bar{\mu}$ are constant):

$$\vec{F} = m\,\vec{a}, \tag{1}$$

$$-\,\overrightarrow{\text{grad}\bar{p}} - \bar{\gamma}\,\overrightarrow{\text{grad}\bar{h}} + \bar{\mu}\bar{\nabla}^2\vec{\bar{u}} = \bar{\rho}\,\frac{D\vec{\bar{u}}}{D\bar{t}}\,, \tag{2}$$

Where $\bar{\rho}$ is density, $\bar{\gamma}$ is unit weight, $\bar{h}$ is elevation from a datum line and $\bar{\mu}$ is dynamic viscosity; $\frac{D}{D\bar{t}}$ is substantive derivative.

Applying the principle of conservation of mass, the equation of continuity may be written in the following form,

$$\bar{\nabla} \cdot \vec{\bar{u}} = -\,\frac{1}{\bar{\rho}}\,\frac{D\bar{\rho}}{D\bar{t}}. \tag{3}$$

In Cartesian coordinates ($\bar{x}$, $\bar{y}$), the equations relating pressure $\bar{p}$, and velocity components ($\bar{u}$, $\bar{v}$) for two-dimensional incompressible flow are

$$-\,\overrightarrow{\text{grad}\bar{p}} + \bar{\mu}\,\bar{\nabla}^2\vec{\bar{u}} = \bar{\rho}\,\frac{D\vec{\bar{u}}}{D\bar{t}}\,, \tag{4}$$

or

$$\bar{\rho}\,(\bar{u}_t + \bar{u}_x\bar{u} + \bar{u}_y\bar{v}) = -\,\bar{p}_x + \bar{\mu}\,(\bar{u}_{xx} + \bar{u}_{yy}), \tag{5}$$

$$\bar{\rho}\,(\bar{v}_t + \bar{v}_x\bar{u} + \bar{v}_y\bar{v}) = -\,\bar{p}_y + \bar{\mu}\,(\bar{v}_{xx} + \bar{v}_{yy}), \tag{6}$$

For two-dimensional incompressible flow the continuity equation may be written as

$$\bar{u}_x + \bar{v}_y = 0, \tag{7}$$

where subscripts denote partial differentiation, and the overbars represent dimensional quantities. A stream function that satisfies equation (4) may be defined in the following manner.

$$\bar{u} = \bar{\psi}_y, \qquad \bar{v} = -\bar{\psi}_x, \tag{8}$$

where $\bar{\psi}$ is the stream function. The vorticity $\bar{\xi}$ is defined by

$$\bar{\xi} = 2\bar{\omega} = -\,(\bar{u}_y - \bar{v}_x) = -\,(\bar{\psi}_{xx} + \bar{\psi}_{yy}), \tag{9}$$

where $\bar{\omega}$ is angular velocity. Then Poisson's equation is

$$\bar{\nabla}^2\,\bar{\psi}(x,\ y) = -\,\bar{\xi}(x,\ y), \tag{10}$$

Taking the curl of equation (4) yields

$$- \text{curl } \overrightarrow{\text{grad}\bar{p}} + \text{curl}\bar{\mu}\bar{\nabla}^2\vec{\bar{u}} = \text{curl}\,\bar{\rho}\,\frac{D\vec{\bar{u}}}{D\bar{t}} \tag{11}$$

$$\bar{\mu}\,\text{curl}\bar{\nabla}^2\vec{\bar{u}} = \bar{\rho}\,\text{curl}\,\frac{D}{D\bar{t}}\,\vec{\bar{u}}\,, \tag{12}$$

$$\bar{\mu}\,\bar{\nabla}^2\text{curl}\vec{\bar{u}} = \bar{\rho}\,\frac{D}{D\bar{t}}\,\text{curl}\vec{\bar{u}}, \tag{13}$$

$$\bar{\nu}\,\bar{\nabla}^2\bar{\xi} = \frac{D}{D\bar{t}}\bar{\xi} \tag{14}$$

or

$$\frac{\partial\bar{\xi}}{\partial\bar{t}} = -\,\vec{\bar{u}}\cdot(\bar{\nabla}\,\bar{\xi}) + \bar{\nu}\bar{\nabla}^2\bar{\xi}, \tag{15}$$

where $\bar{\nu} = \bar{\mu}/\bar{\rho}$ is the kinematic viscosity. Equation (14), the parabolic vorticity transport equation, can be thought of as the condition that a pressure function p exist, by taking the divergence of equation (4). An equation for p, the pressure equation, may be obtained from equation (4):

$$\bar{\nabla}\cdot\left(-\,\overrightarrow{\text{grad}\bar{p}} + \bar{\mu}\bar{\nabla}^2\vec{\bar{u}} = \bar{\rho}\,\frac{D\vec{\bar{u}}}{D\bar{t}}\right)\,, \tag{16}$$

$$\begin{aligned} &-\bar{\nabla}^2\bar{p} + \bar{\nu}\bar{\nabla}^2(\bar{u}_x + \bar{v}_y) \\ &= \frac{\partial}{\partial\bar{t}}(\bar{u}_x + \bar{v}_y) + \bar{u}_x\bar{u}_x + \bar{u}_y\bar{v}_x + \bar{u}\frac{\partial}{\partial\bar{x}}(\bar{u}_x + \bar{v}_y) \\ &\quad + \bar{v}_x\bar{u}_y + \bar{v}_y\bar{v}_y + \bar{v}\frac{\partial}{\partial\bar{y}}(\bar{u}_x + \bar{v}_y), \end{aligned} \tag{17}$$

$$\overline{\nabla}^2 \bar{p} = -(\bar{u}_x^2 + 2\bar{u}_y \bar{v}_x + \bar{v}_y^2) \ . \qquad (18)$$

The normalizing system used throughout the analysis is based on the advective time scale $\bar{L}/\bar{U}_0$, where $\bar{L}$ is a characteristic length and $\bar{U}_0$ is a characteristic velocity of the problem. The dimensionless quantities are defined as follows,

$$u = \bar{u}/\bar{U}_0 , \qquad v = \bar{v}/\bar{U}_0$$

$$x = \bar{x}/\bar{L} , \qquad y = \bar{y}/\bar{L}$$

$$\xi = \bar{\xi}/(\bar{U}_0/\bar{L}), \qquad t = \bar{t}/(\bar{L}/\bar{U}_0). \qquad (20)$$

Combining equations (15) and (10) yields

$$\frac{\partial \xi}{\partial t} = -\nabla \cdot (\vec{u}\,\xi) + \frac{1}{Re}\nabla^2 \xi, \qquad (21)$$

$$\nabla^2 \psi = -\xi , \qquad (22)$$

where Re is Reynolds number,

$$Re = \bar{U}_0 \bar{L}/\bar{\nu} \ . \qquad (23)$$

CHAPTER 3

COMPUTATIONAL APPROACH

- Procedure Methodology

Initially the fluid is at rest and the rotating boundary element assumes its rotational speed instantaneously. Thus, for the first time step the values of all variables, ψ, ξ, u, and v are zero with the exception of the finite velocities at mesh points on the rotating boundary. This constitutes the initial solution. A small time increment, Δt, is chosen according to equation (32) (Roach (12)). Then the computational cycle begins as the finite difference equation (FDE) form of the continuum equation for vorticity transport, equation (21), is used to calculate an approximation to $\frac{\partial \xi}{\partial t}$ at all interior points in the computational field. The new values of ξ are calculated at a new time level, increased by an increment Δt, by "marching" the vorticity transport equation forward in time. The next step in the computational cycle is to solve the FDE form of equation (22), Poisson's equation, for new value of the stream function ψ, using the new interior values of ξ. Since this Poisson solution for new ψ does not depend on the boundary values for new ξ , which are not yet known, the fact that the solution for the new ψ is itself iterative with the optimized successive over-relaxation factor is employed.

At this point, new velocity components can be evaluated by finite difference analogs of equation (8). Then the new values of ξ on the boundaries of the computational region are calculated as a last step. The computational cycle is repeated until the desired time is reached. The new values of vorticity, ξ, at all boundary points maybe calculated with high accuracy, using the appropriate form of the formula of Thom including the curvature term, and "Bubble Sort" techniques. The calculation of ψ for all interior points at each time step, which required the iteration of equation (10), is the most time consuming operation and in order to reduce computing time to a minimum successive over-relaxation is used to increase the speed of convergence of the iterative procedure, and a vectorization is carried out there. As in the point successive over-relaxation (S.O.R.) iterative process, the new value of ψ at each node is computed from the most recent values available at adjacent nodes, as indicated by equation (28). Schematically, the whole procedure is illustrated in Figure 3.1.1.

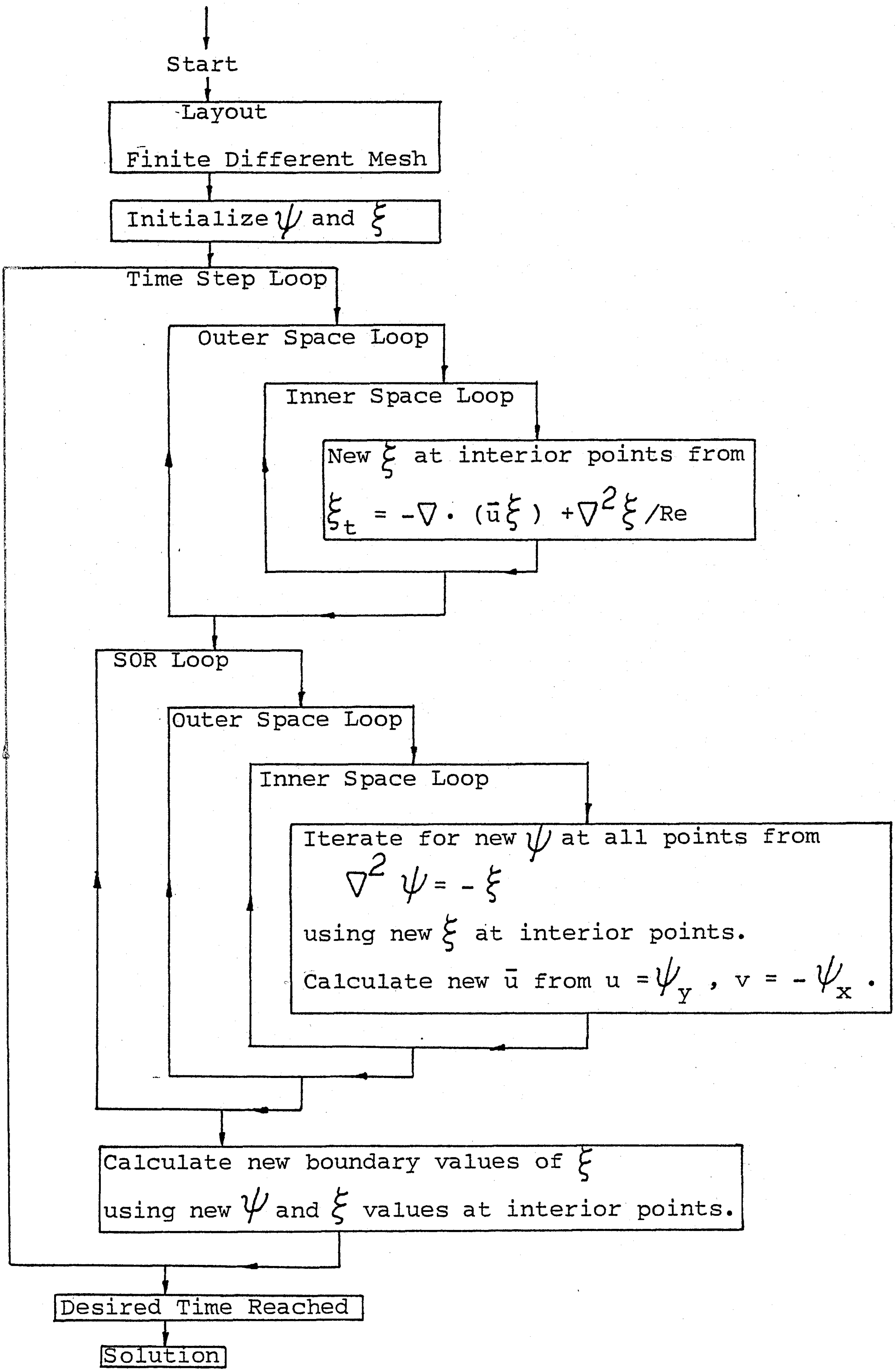

Figure 3.1.1 Main Program

2 - Finite Difference Equation

In the following analysis the governing equations are written in finite difference form. In finite difference formulations, spatial derivatives are represented by centered differences, whereas time derivatives are represented by forward differences (forward time and centered space method: FTCS) except at the advection terms. A one-step, explicit, two-time-level method which achieves static stability of the advection terms involves the use of one-sided, rather than space-centered, differencing. Backward differences are used when the velocities are positive, and vice-versa (Upwind Differencing Method: Lilly (8)). The governing differential equation (15) with symmetric "legs" is formulated in finite difference form as follows:

$$\xi^{n+1}_{(i,j)} = \xi^{n}_{(i,j)} + \Delta t(\nu\nabla^2\xi - \vec{u}\cdot\overrightarrow{\text{grad}}\,\xi)$$

$$= \xi^{n}_{(i,j)} + \nu\Delta t(\nabla^2\xi - \tfrac{1}{\nu}u\,\xi_x - \tfrac{1}{\nu}v\,\xi_y)$$

$$= \xi^{n}_{(i,j)} + \frac{\Delta t}{\delta^2 R}\Big[\xi^{n}_{(i+1,j)} + \xi^{n+1}_{(i-1,j)} + \xi^{n+1}_{(i,j+1)}$$

$$+ \xi^{n}_{(i,j-1)} - 4\,\xi^{n}_{(i,j)}\Big] - \Delta t(\bar{u}\xi_x + \bar{v}\xi_y)$$

$$= \xi^{n}_{(i,j)} + \tfrac{4}{R}\Delta t\,\frac{\langle\xi\rangle_{sur} - \xi^{n}_{(i,j)}}{\delta^2}$$

$$-\Delta t(u\,\xi_x + v\,\xi_y)$$

$$
\begin{aligned}
&= \xi^{n}_{(i,j)} + H1\ [\xi^{n}_{(i+1,j)} - \xi^{n}_{(i,j)} + \xi^{n}_{(i,j-1)} \\
&- \xi^{n}_{(i,j)} - \xi^{n}_{(i,j)} + \xi^{n}_{(i+1,j)} - \xi^{n}_{(i,j)} \\
&+ \xi^{n+1}_{(i,j+1)}] - H2 \{ [\psi^{n}_{(i,j-1)} - \psi^{n}_{(i,j+1)} \\
&+ |\psi^{n}_{(i,j-1)} - \psi^{n}_{(i,j+1)}|](\xi^{n}_{(i,j)} - \xi^{n}_{(i+1,j)}) \\
&+ [\psi^{n}_{(i,j-1)} - \psi^{n}_{(i,j+1)} \\
&- |\psi^{n}_{(i,j-1)} - \psi^{n}_{(i,j+1)}|](\xi^{n+1}_{(i-1,j)} - \xi^{n}_{(i,j)}) \\
&+ [\psi^{n}_{(i+1,j)} - \psi^{n}_{(i-1,j)} \\
&+ |\psi^{n}_{(i+1,j)} - \psi^{n}_{(i-1,j)}|](\xi^{n}_{(i,j)} - \xi^{n+1}_{(i,j+1)}) \\
&+ [\psi^{n}_{(i+1,j)} - \psi^{n}_{(i-1,j)} - |\psi^{n}_{(i+1,j)} - \psi^{n}_{(i-1,j)}|] \\
&(\xi^{n}_{(i,j-1)} - \xi^{n}_{(i,j)})\} \qquad (24)
\end{aligned}
$$

where: δ = grid space,

$$H1 = \frac{\Delta t}{\delta^2 R},$$

$$H2 = \frac{\Delta t}{4\delta^2},$$

and ξ^{n+1} denotes the value at time $t + \Delta t$. The subscripts i,j refer to points in the region of flow such that in the general co-ordinate system $X_i = i\Delta X$, $Y_j = j\Delta Y$. Then the governing differential equation (15) with assymmetric "legs" is

$$\xi^{n+1}_{(i,j)} = \xi^{n}_{(i,j)} + \frac{\Delta t}{R}(\nabla^2\psi) - \frac{\Delta t}{2}(2u\xi_x + 2v\xi_y),$$

$$\xi^{n+1}_{(i,j)} = \xi^{n}_{(i,j)} + \text{H5} * \text{ATLED}$$

$$- \text{H6}\Big\{\Big[\frac{\psi^n_{(i,j+1)} - \psi^n_{(i,j-1)}}{\delta_2 + \delta_4}$$

$$+ \Big|\frac{\psi^n_{(i,j+1)} - \psi^n_{(i,j-1)}}{\delta_2 + \delta_4}\Big|\Big]\Big[\frac{\xi^n_{(i,j)} - \xi_{(i-1,j)}}{\delta_3}$$

$$+ \frac{\psi^n_{(i,j+1)} - \psi^n_{(i,j-1)}}{\delta_2 + \delta_4}$$

$$- \Big|\frac{\psi^n_{(i,j+1)} - \psi^n_{(i,j-1)}}{\delta_2 + \delta_4}\Big|\Big]\Big[\frac{\xi_{(i+1,j)} - \xi^n_{(i,j)}}{\delta_1}\Big]$$

$$+ \Big[\frac{\psi^n_{(i-1,j)} - \psi^n_{(i+1,j)}}{\delta_1 + \delta_3} + \Big|\frac{\psi^n_{(i-1,j)} - \psi^n_{(i+1,j)}}{\delta_1 + \delta_3}\Big|\Big]$$

$$\Big[\frac{\xi^n_{(i,j)} - \xi_{(i,j-1)}}{\delta_4}\Big] + \Big[\frac{\psi^n_{(i-1,j)} - \psi^n_{(i+1,j)}}{\delta_1 + \delta_3}$$

$$- \Big|\frac{\psi^n_{(i-1,j)} - \psi^n_{(i+1,j)}}{\delta_1 + \delta_3}\Big|\Big]\Big[\frac{\xi_{(i,j+1)} - \xi^n_{(i,j)}}{\delta_2}\Big]\Big\} \qquad (25)$$

where: $$H5 = \frac{\Delta t}{R},$$

$$H6 = \frac{\Delta t}{2},$$

$$\begin{aligned}\text{ALTED} = & \frac{2\xi(i+1,j)}{\delta_1(\delta_1+\delta_3)} + \frac{2\xi(i,j+1)}{\delta_2(\delta_2+\delta_4)} \\ & + \frac{2\xi(i-1,j)}{\delta_3(\delta_1+\delta_3)} + \frac{2\xi(i,j-1)}{\delta_4(\delta_2+\delta_4)} \\ & - 2\left[\frac{1}{\delta_1(\delta_1+\delta_3)} + \frac{1}{\delta_2(\delta_2+\delta_4)} + \frac{1}{\delta_3(\delta_1+\delta_3)} + \frac{1}{\delta_4(\delta_2+\delta_4)}\right]\xi^{n}_{(i,j)}\end{aligned}$$

The expression of new or old values of time in the neighborhood of vorticities is omitted, since the scanning of the boundary points is different from that of the interior points. Bubble Sort techniques are used for the computation of vorticities at boundary points, using the formula of Thom (see Chapter 3.5 Boundary Treatment).

For the calculation of the stream functions in the grid centers, the iterative method of successive over-relaxation (SOR) was selected in order to have a very fast asymptotic rate of convergence. A five-point difference equation which represents the Poisson equation (10) can be found by expanding $\psi(x,y)$ in a Taylor series and substituting back into Poisson's equation. Neglecting terms with coefficients of order δ^2 or higher, the finite difference approximation is:

$$\frac{1}{\delta^2}[\psi(i+1,j) + \psi(i,j+1) + \psi(i-1,j) + \psi(i,j-1) - 4\psi(i,j)] = -\xi(i,j) \tag{26}$$

Rearranging, we have

$$\psi(i,j) = \langle\psi\rangle_{sur} + \frac{\xi\delta^2}{4} \tag{27}$$

where "$\langle\ \rangle_{sur}$" denotes the average of surrounding points.

Hockney (6) describes the method of odd/even SOR, in which the points on the grid mesh are corrected in a particular order:

First to be corrected are all the points for which i+j are even, followed by all points with i+j odd.

At each point, the corrected form of the stream function is

$$\psi^{n+1}(i,j) = ORF[\langle\psi\rangle_{sur} + \frac{\delta^2}{4}\xi^{n+1}(i,j)] + (1 - ORF)\psi^{n}(i,j) \tag{28}$$

ORF is called the over-relaxation factor, $1 < ORF < 2$, and it can be adjusted in order to speed up convergence of the system.

In this analysis, one more program was formulated which may calculate the optimized ORF with the calculations of the minimized error, being defined as

$$E = \sum_{i=1}^{NX}\sum_{j=1}^{NY}\left|\psi^{n+1}(i,j) - \psi^{n}(i,j)\right| \tag{29}$$

where:

NX is the number of x divisions, and NY is the number of y divisions.

Since the approach of Frankel (3) for the Dirichlet problem in a rectangular domain of size $(i - 1)\Delta x$ by $(j - 1)\Delta y$ with constant Δx and Δy it may be shown that

$$ORF = 2\left(\frac{1 - \sqrt{1 - \eta}}{\eta}\right) \tag{30}$$

where:

$$\eta = \left[\frac{\cos\left(\frac{\pi}{i-1}\right) + \left(\frac{\Delta x}{\Delta y}\right)^2 \cos\left(\frac{\pi}{j-1}\right)}{1 + \left(\frac{\Delta x}{\Delta y}\right)}\right],$$

The ORF may be approximated as an initial value for the program. Since $i = j$ and $\beta = \Delta x/\Delta y = 1$, then

$$ORF = 2\left(\frac{1 - 1 - \cos\left(\frac{\pi}{i-1}\right)}{\cos\left(\frac{\pi}{i-1}\right)}\right) \tag{31}$$

Let i-1 be large. Then, using $\cos\left(\frac{\pi}{i-1}\right) \approx 1 - \frac{\left(\frac{\pi}{j-1}\right)^2}{2!}$,

The initial guess (IORF $\approx$ 1.91 for 50 by 50 grid system), which is decreased by small decrements in order to get the minimized error may be thus obtained.

Using this program, the optimum over-relaxation factor was determined for the rectangular coordinates 50 by 50 grid system. The optimum over-relaxation factor was found to be ORF = 1.87 with the value of error $E = 8.5 \times 10^{-31}$.

Because of the odd/even fashion of stepping through the grid, the SOR routine can be put into vectorized form for use on the CRAY-1 computer. Vectorizing on the CRAY-1 refers to arranging the FORTRAN code in such a way that the computer can take advantage of its parallel processing capabilities, meaning that calculations normally done sequentially are done concurrently. Vectorizing is possible because the calculation of any single odd/even point does not depend on the previously computed odd/even point on that row. Vectorizing SOR results in an increase in speed by a factor of 3 on the CRAY-1. The program listed in Appendix C is the vectorized version.

A finite-difference equation is said to be stable if its solution remains bounded as $t \to \infty$. Furthermore, it is said to be convergent if the solution of the difference equation converges to the solution of the corresponding differential equation as the spatial and time increments tend to zero. Lax and Richtmeyer (7) show that stability implies convergence if the difference equations are consistent with the differential equations, i.e. if the truncation errors tend to zero as the spatial and time increments tend to zero. In this project, an analysis of a fluid filled rotating infinite cylinder was resolved with an error of $E \doteq 0.1$ %, which is in very good agreement with the computations of analytical work. (see Figure B in Appendix A). The convergence of a flow in a circular cavity with part of wall moving is attained for Reynolds numbers 3.16, 10, 31.6, 100, 316, 1000, and 3160. Considering the application of upwind differencing with diffusion terms in two-dimensional incompressible flow, Roach (12) showed that the time step, Δt, which is required for stability, is

$$\Delta t \leq \frac{1}{\frac{2}{Re}\left(\frac{1}{\Delta x} + \frac{1}{\Delta y}\right) + \frac{|u|}{\Delta x} + \frac{|v|}{\Delta y}} . \tag{32}$$

For CRAY-1, the half value of the time step, calculated in equation (32), was used in order to obtain the solution.

3 - Classification of Mesh Points

A polygon, π, was setup and periodic meshes covered, with mutual overlap (λ), by π (see Figure 3.3.1).

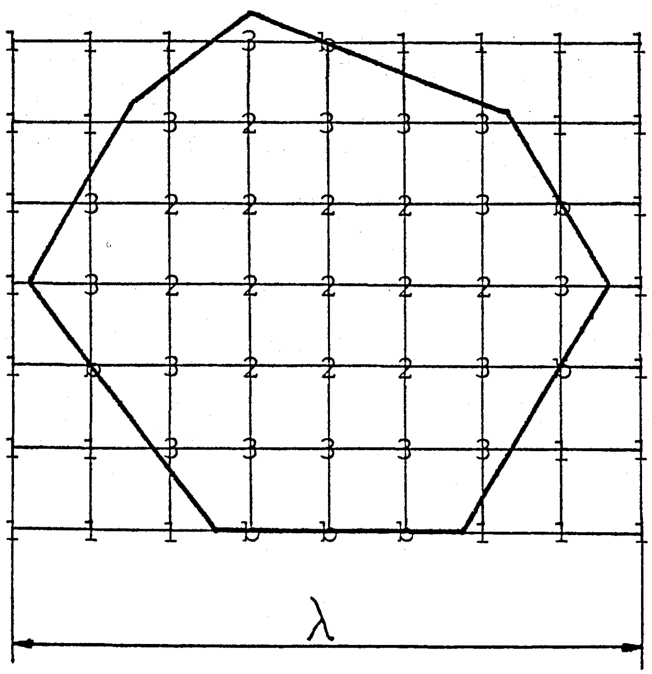

Figure 3.3.1 Classification of Mesh Points

Each mesh point is classified according to its relationship to π. Subroutine CLOSE takes as its input a point x in the plane, and an N sided polygon π in the plane, of vertices $[\vec{x}_i]$. It returns an output ICLOSE of "2" if $\vec{x}$ is inside π, and an output ICLOSE of "1" if $\vec{x}$ is outside π (see Figure 3.3.2).

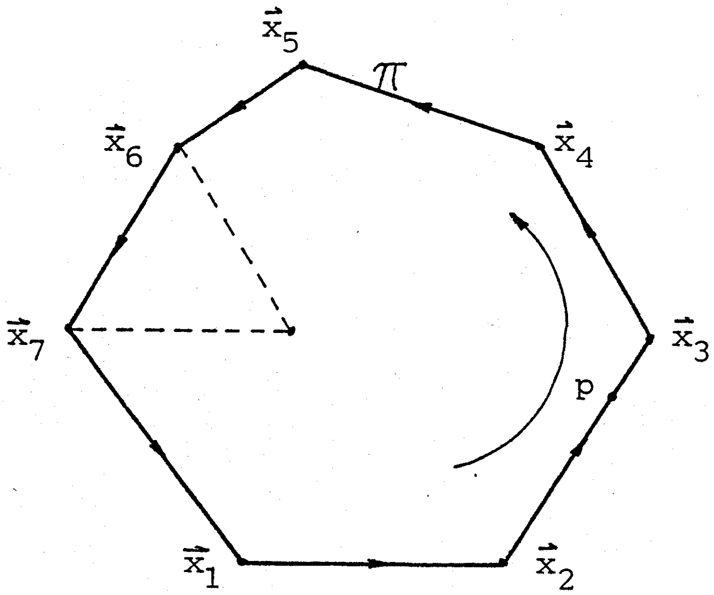

Figure 3.3.2 Moving Points Around a Polygon.

$\vec{x}_1$, shown in Figure 3.3.2 is both the start and the finish point for p as p moves around π. Subroutine NEAR determines whether a point p is near a polygon π. By "near" is meant within the small distance, using the matrix $|\Delta x| + |\Delta y| = d*$ of a vertex, or using the standard matrix $\sqrt{\Delta x^2 + \Delta y^2} = d*$ and requiring that a vertical line from p intersects the side. With the help of these two subroutines, there may be 4 catagories, 1, b, 2, and 3 (see Figure 3.3.1),

where:

1 : mesh points outside π

b : mesh points on Π

3 : mesh points inside Π but with a neighbor either on or outside Π

2 : mesh points inside Π all of whose neighbors are also inside Π.

4 - Treatment of Points According to Category

The mesh points on and outside of Π were skipped. At mesh points inside Π but with a neighbor either on Π or outside a stream function, ψ, is computed by assymetrical finite difference approximation to Poisson's equation (see Figure 3.4.1).

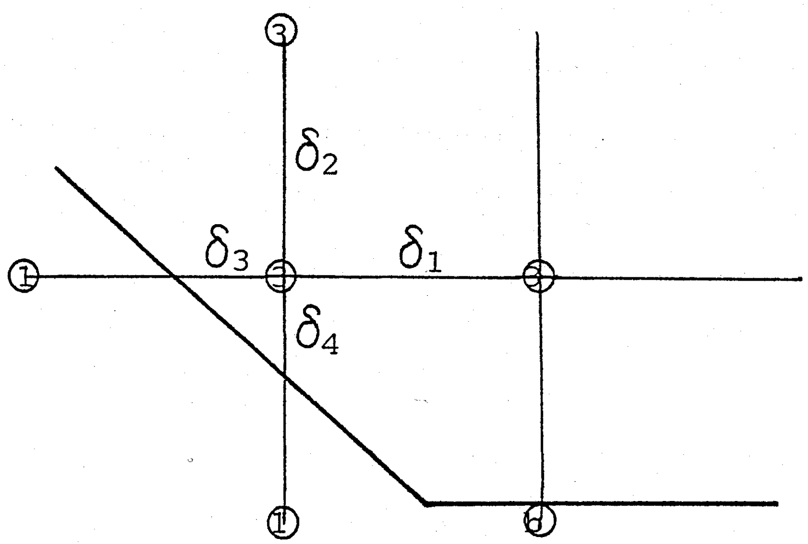

Figure 3.4.1 Treatment of Points

The length of the shortened "legs" are computed by subroutine POINT. The values of ψ at the "feet" of the shortened legs are given by the boundary conditions, $\psi = \psi(s)$ on Π. The computation of the finite difference approximation to Poisson's equation is accomplished by subroutine SIGMA.

A vorticity, ξ , is computed by the use of the no slip boundary condition according to the formula of Thom (Roach(12)). At strictly interior points (interior points with interior neighbors), a stream function is computed from symmetric finite difference approximation to Poisson's equation by the method of relaxation. A vorticity, ξ , is computed using weather differenced finite difference approximation to the vorticity transport equation (15).

5 - Boundary Treatment

For each boundary point (3-point) the arc length measured counter-clock-wise from the first vertex of π to the base of the normal from the boundary point to π is computed using the subtroutine APROCH, augmented by the subroutine VARC computing the arc length of the vertices (see Figure 3.5.1). In order to apply Bubble Sort technique along the perimeter of a polygon in the counter-clock-wise sence, the subroutine PRANK gives the rank from smallest to largest of the arc lengths. By employing this method, a more accurate analysis may occur because the lengths of 1 and n in the formula of Thom ($1 \not\approx n$) are differentiated.

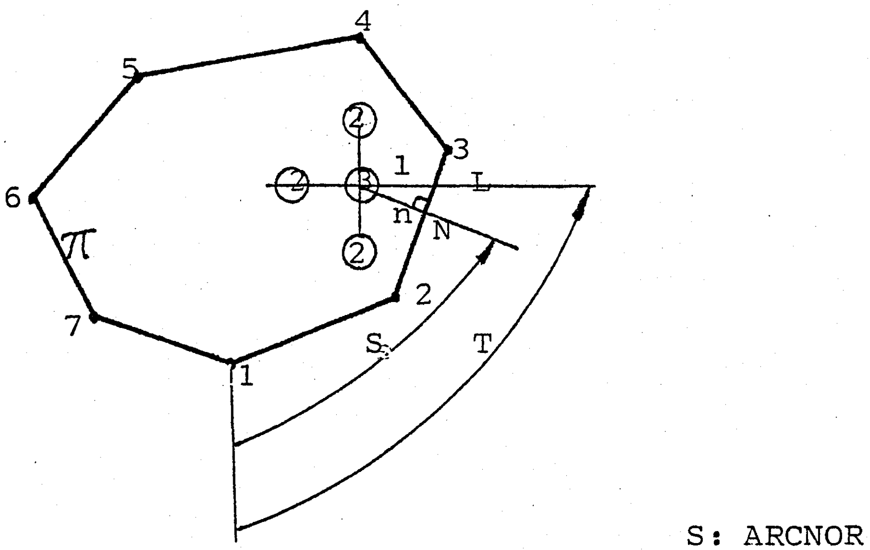

Figure 3.5.1 Method of Handling 3-Points

Vorticity at N is determined from the formula of Thom. Vorticity at L is determined by linear interpolation between such points as N, using the subroutine PERINT. The formula of Thom may be restated for no-slip conditions at a straight wall. Let point P be at a distance δ from an impermeable wall w which moves at speed U_w (see Figure 3.5.2). Let y be the coordinate normal to the wall (y = 0 on w).

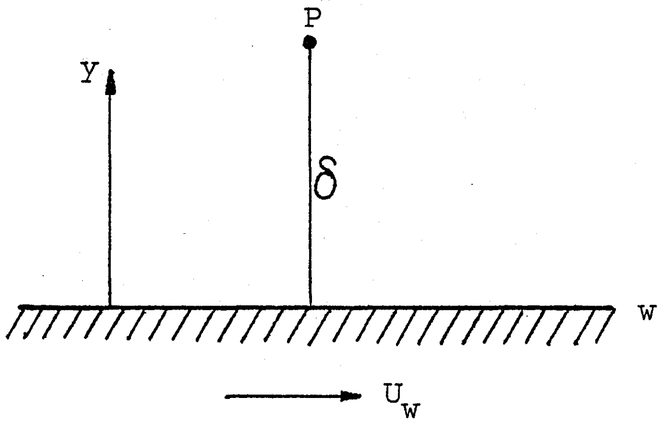

Figure 3.5.2 Point Near a Moving Wall

Let the stream function at the wall be ψ_w and let the stream function at P be ψ_p. Let the fluid velocity be U(y). Due to the no slip condition,

$$U(0) = U_w \tag{33}$$

Consider first the case of zero vorticity, in which

$$U(y) = U_w \qquad \forall y, \tag{34}$$

where "$\forall$" means "at all points."

Eq. (34) states that the fluid between w and P is all flowing parallel to the wall at speed U_w as shown in figure 3.5.3a.

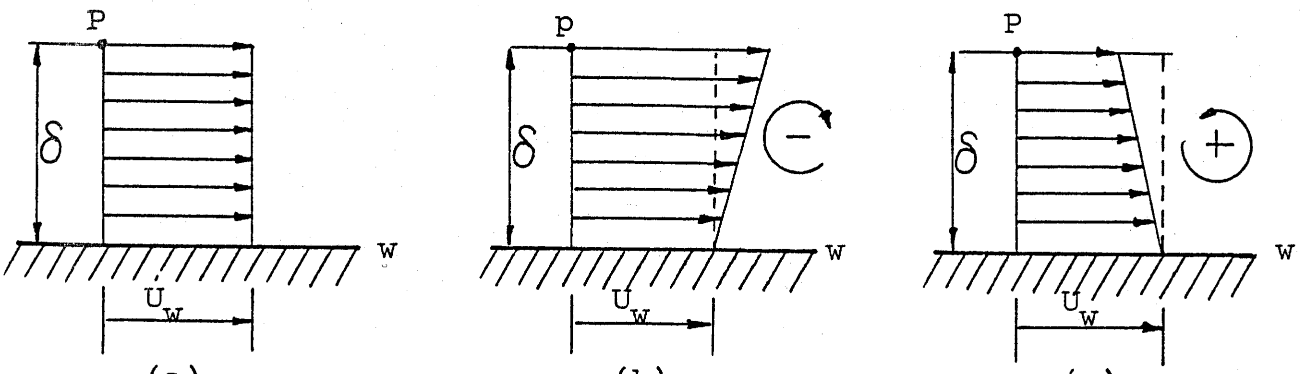

Figure 3.5.3 Relation Between ψ and ξ at w

Then the flow between w and P is given by

$$\psi_p - \psi_w = U_w \delta \tag{35}$$

Consider next the case where U(y) is not constant but is linear according to

$$U(y) = U_w + ky \tag{36}$$

where k is constant. Clearly

$$k = U'(y) = -\xi \tag{37}$$

where ξ here is assumed constant. Integrating Eq. (36) gives

$$\begin{aligned} \psi_p - \psi_w &= \int_0^\delta U(y)\, dy \\ &= U_w \delta - k \frac{\delta^2}{2} \end{aligned} \tag{38}$$

solving for ξ gives

$$\xi = \frac{2}{\delta^2}[(U_w\delta) - (\psi_p - \psi_w)] \tag{39}$$

which is called the formula of Thom. The first term in brackets in equation (39), $U_w\delta$, is recognized from equation (35) as the flow between P and w which would occur if the velocity were constant at U_w. The second term, $(\psi_p - \psi_w)$, is the flow which actually occurs. The formula of Thom, equation (39), implies that if there is more flow between w and P than would be provided by a constant velocity, then the vorticity will be negative (see Figure 3.5.3b). If there is less flow than $U_w\delta$, then the vorticity will be positive (see Figure 3.5.3c). The formula of Thom may also be derived for a curved wall (see Figure 3.5.4).

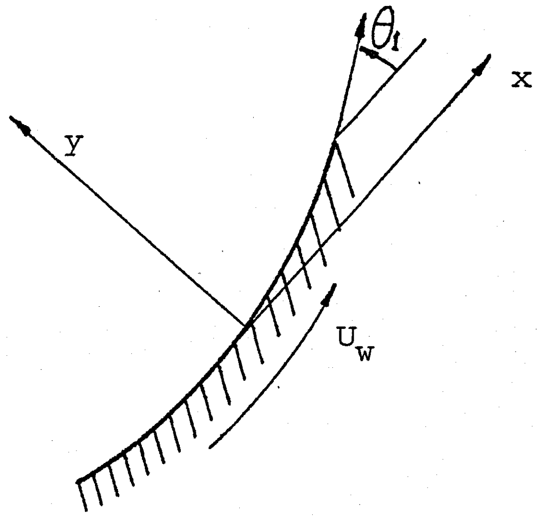

Figure 3.5.4 Formula of Thom for a Curved Wall

From the definition of vorticity, equation (9),

$$\xi = v_x - u_y$$

$$= \frac{U_w d\theta}{dx} - (\psi_y)_y$$

$$= U_w\left(\frac{d\theta}{dx}\right) - \left(\frac{\psi_p - \psi_w}{\delta} - U_w\right)/\delta\,/\,2.$$

Rearranging:

$$\xi = \frac{U_w}{a} + \frac{2}{\delta^2}\left[(U_w\delta) - (\psi_p - \psi_w)\right] \tag{40}$$

where a is the radius of curvature, or

$$a = \frac{(1 + y_x^2)^{3/2}}{y_{xx}} \tag{41}$$

CHAPTER 4

ANALYTICAL APPROACH FOR A TEST PROBLEM

Acylinder of internal radius, a, and infinite length is filled with a fluid of kinematic viscosity ν. Initially the fluid is at rest. At time zero the internal wall starts moving at peripheral speed U_w (Figure 4.4.1).

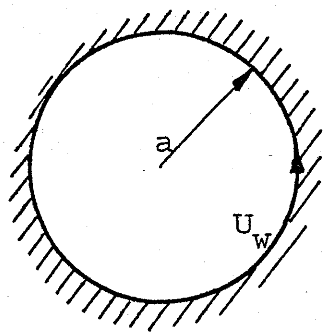

Figure 4.4.1 Rotating Infinite Cylinder Filled with a Fluid

The overbars are omitted for representing dimensional quantities in this section. The problem shall be formulated in terms of stream function and vorticity. Due to rotational symmetry these may be written, respectively, as $\psi(r,t)$ and $\xi(r,t)$, the ϕ being neglected. The differential equations for stream function and vorticity are, respectively

$$\psi_{rr} + \frac{1}{r}\psi_r = -\xi \qquad (42)$$

and

$$\nu\left(\xi_{rr} + \frac{1}{r}\xi_r\right) = \xi_t + \cancel{\vec{u}\cdot \mathrm{grad}\,\xi}^{\,0}\;{}^{*} \tag{43}$$

The boundary condition on the peripheral velocity q (r,t) is

$$q(a,\ t) = (-\psi_r)_{@r\,=\,a} = U_w \tag{44}$$

The initial condition of rest is given by

$$\psi(r,\ o) = 0, \tag{45}$$

and

$$\xi\ (r,\ o) = 0. \tag{46}$$

The problem of spinup is considered solved if q (r, t) are determined for all r and t. Starting with the vorticity equation (14) and applying separation of variables in ξ(r, t), according to

$$\xi\,(r,\ t) = R\ (r)\ T\ (t) \tag{47}$$

yields, upon substitution into equation (43),

$$\nu\ (R_{rr}T + \frac{1}{r}\ R_rT) = RT_t$$

$$\nu\left(\frac{R_{rr}}{R} + \frac{1}{r}\ \frac{R_r}{R}\right) = \frac{T_t}{T} = K = \text{const.} \tag{48}$$

---

* $\vec{u}\cdot \mathrm{grad}\,\xi = \begin{pmatrix}\cancel{u}\\ v\end{pmatrix}^{0} \cdot \begin{pmatrix}\xi_r\\ \cancel{\xi_\theta}^{\,0}\end{pmatrix} = 0$

This clearly resulted in two ordinary differential equations,

$$R_{rr} + R_r - \left(\frac{K}{\nu}\right)rR = 0 \qquad (49)$$

and

$$T_t - KT = 0. \qquad (50)$$

Equation (50) has the general solution

$$T = ce^{kt}. \qquad (51)$$

The vorticity is required to be bounded as $t \rightarrow \infty$ and therefore $K \leq 0$. Then,

$$k = -h; \qquad h \geq 0 \qquad (52)$$

Equation (49) is Bessels equation of order m, which is

$$xy_{xx} + xy_x + (x^2 - m^2) = 0 \qquad (53)$$

To transform equation (49) into equation (53) r is nondimensionlized according to

$$r_n = r/L \qquad (54)$$

Transformation of equation (49) to the dimensionless radius r_n proceeds as follows.

$$r_n L\, R_{r_n r_n} \frac{1}{L^2} + R_{r_n} \frac{1}{L} - \frac{k}{\nu} r_n L R = 0$$

$$r_n R_{r_n r_n} + R_{r_n} + (L^2 \frac{h}{\nu})\, r_n R = 0 \tag{55}$$

Selecting

$$L = \sqrt{\frac{\nu}{h}} \tag{56}$$

then

$$(L^2 \frac{h}{\nu}) = 1 \tag{57}$$

and

$$r_n^2 R_{r_n r_n} + r_n R_{r_n} + r_n^2\, R = 0 \tag{58}$$

Comparison of equation (58) with equation (53) shows equation (49) to be a Bessel equation of order zero. With the aid of equation (54) and equation (56) solution of equation (49) may be written as

$$R = J_o\, (r_n) = J_o(\sqrt{\frac{h}{\nu}} r) \tag{59}$$

The desired solution of the vorticity equation is given by an expansion, in terms of the form (47), as follows.

$$\xi = \sum_{i=0}^{\infty} c_i e^{-h_i t}\, J_o\, (\sqrt{\frac{h_i}{\nu}} r) \tag{60}$$

The constants $[c_i]$ and $[h_i]$ must be chosen so as to satisfy the boundary and initial conditions to determine the constants, an expression for the peripheral verocity $q = -\psi_r$ is derived. The equation for ψ, Eq. (42), may be written as

$$\frac{1}{r}\frac{\partial}{\partial r}(r\psi_r) = -\xi \tag{61}$$

The space dependent portion of each term of (60) is the form $J_o(\alpha_i r)$ so

$$\frac{1}{r}\frac{d}{dr}(r\psi_r^i) = -J_o(\alpha_i r) \tag{62}$$

which is integrated as follows.

$$r\psi_r^i = -\int_0^r rJ_o(\alpha_i r)dr$$

$$= -\frac{1}{\alpha_i^2}\int_0^r (\alpha_i r)J_o(\alpha_i r)d(\alpha_i r)$$

$$= -\frac{1}{\alpha_i^2}\int_0^{\alpha_i r}\theta\, J_o(\theta)d\theta$$

$$= -\frac{1}{\alpha_i^2}(\alpha_i r)J_1(\alpha_i r)$$

Thus

$$\psi_r^i = -\frac{1}{\alpha_i}J_1(\alpha_i r) \tag{63}$$

from which

$$q(r,\ t) = -\psi_r = \sum_{i=0}^{\infty} c_i e^{-h_i t} \sqrt{\frac{\nu}{h_i}}\, J_1(\sqrt{\frac{h_i}{\nu}}\, r) \qquad (64)$$

Now equations (44) and (60) are used to compute the constants $[h_i]$. Near the wall at early times the velocity jumps from 0 to U_w over a very thin layer of thickness ϵ (Figure 4.4.2).

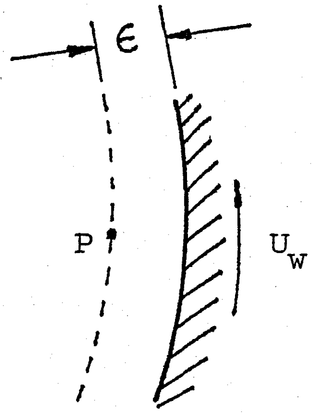

Figure 4.4.2 Thin Layer of Thickness ϵ

In that layer, for y = 0 and x = r,

$$\xi = v_x - u_y = 0(\frac{U_w}{\epsilon}) - 0(\frac{U_w}{a}) \approx v_r \qquad (65)$$

so

$$U_w = \int_{a-\epsilon}^{a} \xi\, dr = \text{const.}, \qquad (66)$$

where $0(\frac{U_w}{\epsilon})$ means the order of U_w/ϵ.

Thus, it can be seen that the vorticity distribution across the friction layer has a constant integral, U_w. The rate of change of that integral with respect to time is given by

$$0 = \frac{d}{dt}U_w = \frac{d}{dt}\int_{a-\epsilon}^{a} \xi\, dr = \begin{bmatrix}\text{vorticity rate of} \\ \text{increase in layer}\end{bmatrix}$$

$$= \begin{bmatrix} \text{vorticity diffusing} \\ \text{into the layer from} \\ \text{the wall at } r = a \end{bmatrix} - \begin{bmatrix} \text{vorticity diffusing} \\ \text{out of the layer at} \\ \text{inside edge of layer} \end{bmatrix}^{*}$$

$$= \nu \left(\frac{\partial \xi}{\partial r}\right)_{r=a} - \nu \left(\frac{\partial \xi}{\partial r}\right)_{r=a-\epsilon} \tag{67}$$

For $r \leq a-\epsilon$ the vorticity and all its derivatives are extremely small by equation (46) so equation (67) becomes

$$0 = \nu \left(\frac{\partial \xi}{\partial r}\right)_{r=a} - \nu \cdot 0$$

and

$$\left(\frac{\partial \xi}{\partial r}\right)_{r=a} = 0 \qquad \forall \text{ small } t. \tag{68}$$

Equations (50) and (68) can only hold for all small t if

$$\frac{d}{dr} J_o \left(\sqrt{\frac{h_i}{\nu}}\, r\right) = 0 \qquad @r = a$$

which implies that

$$J_1 \left(\sqrt{\frac{h_i}{\nu}}\, a\right) = 0 \qquad \forall i. \tag{69}$$

Thus the values of $\sqrt{\frac{h_i}{\nu}}\, a$ are zeros of J_1:

$$\sqrt{\frac{h_i}{\nu}}\, a = 0;\ 3.8317;\ 7.0156;\ 10.1735;\ 13.3237;\ 16.4706;\ \ldots\ldots \tag{70}$$

* This form of equation (14) is same as that of the energy equation.

From equation (70) the leading term of equation (64) is seen to be indeterminate. That term is evaluated by taking the limit as

$$\sqrt{\frac{h_o}{\nu}}\, a \rightarrow 0.$$

thus

$$\lim_{h_o \to 0} \left[\frac{J_1\left(\frac{h_o}{\nu} r \right)}{\sqrt{h_o/\nu}} \right] = \lim_{\alpha \to 0} \left[\frac{J_1(\alpha r)}{\alpha} \right] = \frac{r}{2} \tag{71}$$

From (64) and (71),

$$q\,(r,\, t) = \sum_{i=0}^{\infty} c_i e^{-h_i t} \frac{1}{\sqrt{h_i/\nu}}\; J_1\left(\sqrt{\frac{h_i}{\nu}}\, r\right)$$

or

$$q\,(r,\, t) = c_o \frac{r}{2} + \sum_{i=1}^{\infty} c_i e^{-h_i t} \sqrt{\frac{\nu}{h_i}}\, J_1\left(\sqrt{\frac{h_i}{\nu}}\, r\right) \tag{72}$$

At $r = a$, $q = U_w$ and $J_1\left(\sqrt{\frac{h_i}{\nu}}\, r\right) = 0$
so equation (72) gives

$$c_o = \frac{2U_w}{a}\,. \tag{73}$$

The dimensionless peripheral velocity q' is defined as follows.

$$q' = q/U_w \tag{74}$$

From equation (72) to equation (74)

$$q'(r', t') = r' - \sum_{i=1}^{\infty} a_i e^{-\alpha_i^2 t'} J_1(\alpha_i r') \qquad (75)$$

where the $[\alpha_i]$ are the zeros of J_1, $a_i = -\frac{c_i}{U_w}\sqrt{\frac{\nu}{h_i}}$, and the dimensionless radius and time are given by

$$r' = r/a \qquad (76)$$

and

$$t' = \frac{t\nu}{a^2} . \qquad (77)$$

Clearly as $t' \rightarrow \infty$ equation (75) gives the correct limiting peripheral velocity which is that of rigid body rotation (see curve labeled ∞ in Figure 4.4.3).

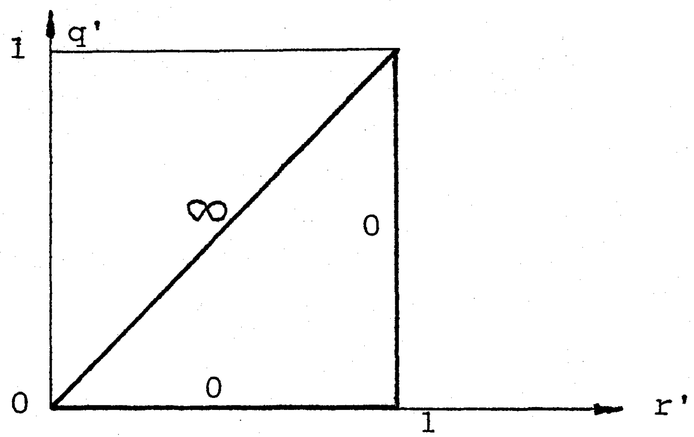

Figure 4.4.3 Dimensionless Peripheral Velocity vs. Dimensionless Radius.

There remains the task of computing the coefficients $[a_i]$ so that equation (75) will give the correct peripheral velocity at $t' = 0$ (see curve labeled 0 in Figure 4.4.3). Substituting $t' = 0$ into equation (75) gives

$$q'(r', 0) = r' - \sum_{i=1}^{\infty} a_i J_1(\alpha_i r') \tag{78}$$

or

$$\sum_{i=1}^{\infty} a_i J_1(\alpha_i r') = r' - q'(r', 0)$$

$$= q'(r', \infty) - q'(r', 0)$$

$$= \begin{cases} r' & ; r' < 1 \\ 0 & ; r' = 1 \end{cases} \tag{79}$$

The coefficients $[a_i]$ are obtained classically by convoluting equation (79) with successive components and applying the orthogonality conditions for Bessel functions. Thus both sides of equation (79) are multiplied by $r'J_1(\alpha_j r')$ and integrated between 0 and 1.

$$\sum_{i=1}^{\infty} a_i \int_0^1 r'J_1(\alpha_i r')J_1(\alpha_j r')dr'$$

$$\int_0^1 (r')^2 J_1(\alpha_j r')dr' \tag{80}$$

Now applying the orthogonality conditions

$$\int_0^1 rJ_1(\alpha_i r)J_1(\alpha_j r)dr = \begin{cases} 0 & ; i \neq j \\ \frac{1}{2}(J_1'(\alpha_i))^2 ; & i = j \end{cases} \tag{81}$$

to equation (80) gives

$$\frac{1}{2}\, a_j(J_1'(\alpha_j))^2 = \int_0^1 (r')^2 J_1(\alpha_j r')dr' \qquad (82)$$

The orthogonality equation has effectively isolated a single coefficient a_j for computation. The derivative $J_1'(\alpha_j)$ may be evaluated by use of the familiar Bessel function relationship

$$\frac{d}{dx}\,(x^m J_m(x)) = x^m J_{m-1}(x)$$

Thus setting m = 1 yields

$$\frac{d}{dx}(xJ_1(x)) = xJ_1'(x) + J_1(x) = xJ_o(x)$$

$$\alpha_j J_1'(\alpha_j) + J_1(\alpha_j) = \alpha_j J_o(\alpha_j).$$

But $J_1(\alpha_j) = 0$ so

$$\alpha_j J_1'(\alpha_j) + 0 = \alpha_j J_o(\alpha_j)$$

and

$$J_1'(\alpha_j) = J_o(\alpha_j) \qquad (84)$$

The integral appearing in equation (82) may be evaluated by another application of equation (83).

Setting m = 2 yields

$$\frac{d}{dx}\,(x^2 J_2(x)) = x^2 J_1(x)$$

which, upon integration yields

$$\int_0^{\alpha} x^2 J_1(x)dx = \alpha^2 J_2(\alpha).$$

thus

$$\int_0^1 (r')J_1(\alpha_j r')dr' = \frac{1}{\alpha_j^3}\int_0^1 (\alpha_j r')^2 J_1(\alpha_j r')d(\alpha_j r')$$

$$= \frac{1}{\alpha_j^3}\int_0^{\alpha_j} xJ_1(x)dx = \frac{1}{\alpha_j^3}\alpha_j{}^2 J_2(\alpha_j)$$

$$= \frac{1}{\alpha_j} J_2(\alpha_j) \tag{85}$$

From equations (82), (84), and (85) we write

$$a_j = \frac{2}{\alpha_j}\frac{J_2(\alpha_j)}{(J_o(\alpha_j))^2} \tag{86}$$

Manipulation of the formal series for the Bessel function, which is

$$J_m(x) = (\tfrac{x}{2})^m \frac{1}{m!}\Big\{1 - \frac{(x/2)^2}{(1)[(m+1)]} + \frac{(x/2)^4}{(1\cdot 2)[(m+1)(m+2)]}$$

$$- \frac{(m+2)^6}{(1\cdot 2\cdot 3)[(m+1)(m+2)(m+3)]} + \ldots\ldots\Big\}, \tag{87}$$

easily provides the relationship

$$J_2(\alpha_i) = - J_o(\alpha_i). \tag{88}$$

Equation (88) simplifies equation (86) to

$$a_j = \frac{-2}{\alpha_j J_o(\alpha_j)} \tag{89}$$

substitution of equation (89) into equation (75) gives

$$q'(r', t') = r' + \sum_{i=1}^{\infty} \frac{2e^{-\alpha_i^2 t'}}{i} \frac{J_1(\alpha_i r')}{J_o(\alpha_i)} \tag{90}$$

By assembling several of our results we may write the following expressions for the dimensionless vorticity and stream function.

$$\xi' = \frac{\xi}{(2U_w/a)} = 1 + \sum_{i=1}^{\infty} e^{-\alpha_i^2 t'} \frac{J_o(\alpha_i r')}{J_o(\alpha_i)} \tag{91}$$

$$\psi' = \frac{\psi}{(U_w a/2)} = (1 - r'^2) - 4 \sum_{i=1}^{\infty} \frac{e^{-\alpha_i^2 t'}}{\alpha_i^2} [1 - \frac{J_o \alpha_i r)}{J_o(\alpha_i)}] \tag{92}$$

A "Bessel" program is listed in Appendix B, which calculates values of q', ξ', and ψ' at various values of r' and t', using three subtoutines, SPEED which computes equation (90), VORT which computes equation (91), and FLOW which computes equation (92). The "Bessel" program was run on the Honeywell 66, using IMSL routines.

CHAPTER 5

NUMERICAL RESULTS AND DISCUSSION.

The stream function at center of cylinder in a test problem is plotted with respect to time. A comparison between Bessel function solution and numerical solution is shown in Figure B in Appendix A. Agreement is very good with the approximate error of 0.1%. There was a discrepancy due to omitting the curvature term in Thom's formula since the errors associated with the polygonal approximation propogate into the interior of the flow. The vorticities of the 26th column with 40 vertices are shown in Figure C in Appendix A. The results of Bessel function solution are shown by broken lines. It may be seen that the vorticity lines are becoming horizontal when time is increased, as is expected. The final value of vorticity at center is obtained after approximately 1600 time steps of 0.0002 each with the total dimensionless time of 0.32 secs (see Figure D in Appendix A).

The final values of vorticity at center, with respect to time steps, of a half-moving solid boundary problems ($\theta = 180^{\circ}$) are shown in Figure E to Figure K in Appendix A for the dimensionless Reynolds numbers, 3.16, 10, 31.6, 100, 316, 1000, and 3160 with the total dimensionless time, being converged. The cases of a quarter-moving solid boundary problems ($\theta = 90^{\circ}$) are plotted in Figure L to Figure O shown in Appendix A for the dimensionless Reynolds numbers, 100, 316, 1000, and 3160 with the total dimensionless time, being converged. The duration for convergence with respect to the dimensionless Reynolds numbers in logarithmic scale (see Figure P in Appendix A) may be calculated. Ultimate vorticities at column 26 are resolved in Figure Q, Figure R, and Figure S in Appendix A (Re = 100, 1000, and 3160, $\theta = 180^{\circ}$). Some final vorticities at center of circle and at center of vortex are shown in Figure T and Figure S in Appendix A. The advection effects may be seen from the equi-vorticity lines in Figure V and Figure W, and furthermore, these data would become valuable in analyzing more delicate and complicated phenomena.

LIST OF REFERENCES

(1) Courant, R. and Hilbert, D., Methods of Math. Phys., vol. 1, Interscience Pub., N.Y., 1937.

(2) Dix, D. M., J. Fluid Mech., vol. 15, 449, 1963.

(3) Frankel, S. P., "Convergence Rates of Iterative Treatments of Partial Differential Equations, Math. Tables and other Aids to Computation, vol. 4, pp. 65-75, 1950.

(4) Fromm, J. E. and Harlow, F. H., Phys. of Fluids, vol. 6, 975, 1963.

(5) Helliums, J. D. and Churchill, S. W., Proc. Int. Heat Transf. Conf., 985, 1961.

(6) Hockney, R. W., "The Potential Calculation and Some calculations", Methods in Computational Physics, vol. 9, eds. B. Alder, S. Fernbach, and M. Rotenberg, Academic Press, N.Y., 1970.

(7) Lax, P. D. and Richtmeyer, R. D., "Survey of the Stability of Linear Finite - Difference Equations", Comm. Pure Appl. Math., vol. 9, 267, 1956.

LIST OF REFERENCES (continued)

(8) Lilly, D. K., "On the Computational Stability of Numerical Solution of Time-Dependent Non-Linear Geophysical Fluid Dynamics Problems", U.S. Weather Bureau Monthly Weather Review, vol. 93, No. 1, pp. 11-26, 1965.

(9) Pearson, C. E., A Computational Method for Viscous Flow Problems, Sperry Rand Research Center, 1964.

(10) Peirce, B. O., A Short Table of Integrals, Dinn and Co., Boston, 1956.

(11) Reddick, H. W., and Miller, F. H., Adv. Math. for Engineers, John Wiley, N.Y., 1938.

(12) Roach, P. J., Computational Fluid Dynamics, Hermosa Publishers, Albuquerque, 1972.

(13) Schlichting, H., Boundary Layer Theory, McGraw-Hill, N.Y., 1979.

(14) Strawbridge, D. R. and Hooper, G. T. J., "Numerical solutions of the Navier - Stokes Equations for axisymmetric Flow", J. of Mechanical Engineering Science, vol. 10, No. 5, 1968.

LIST OF REFERENCE (continued)

(15) Thompson, P.D., Numerical Weather Analysis and Prediction, MacMillan, N.Y., 1961.

(16) Wilkes, J.O., Thesis, Univ. of Michigan, 1963.

APPENDIX A

FIGURES FOR MESH TREATMENT AND RESULTS

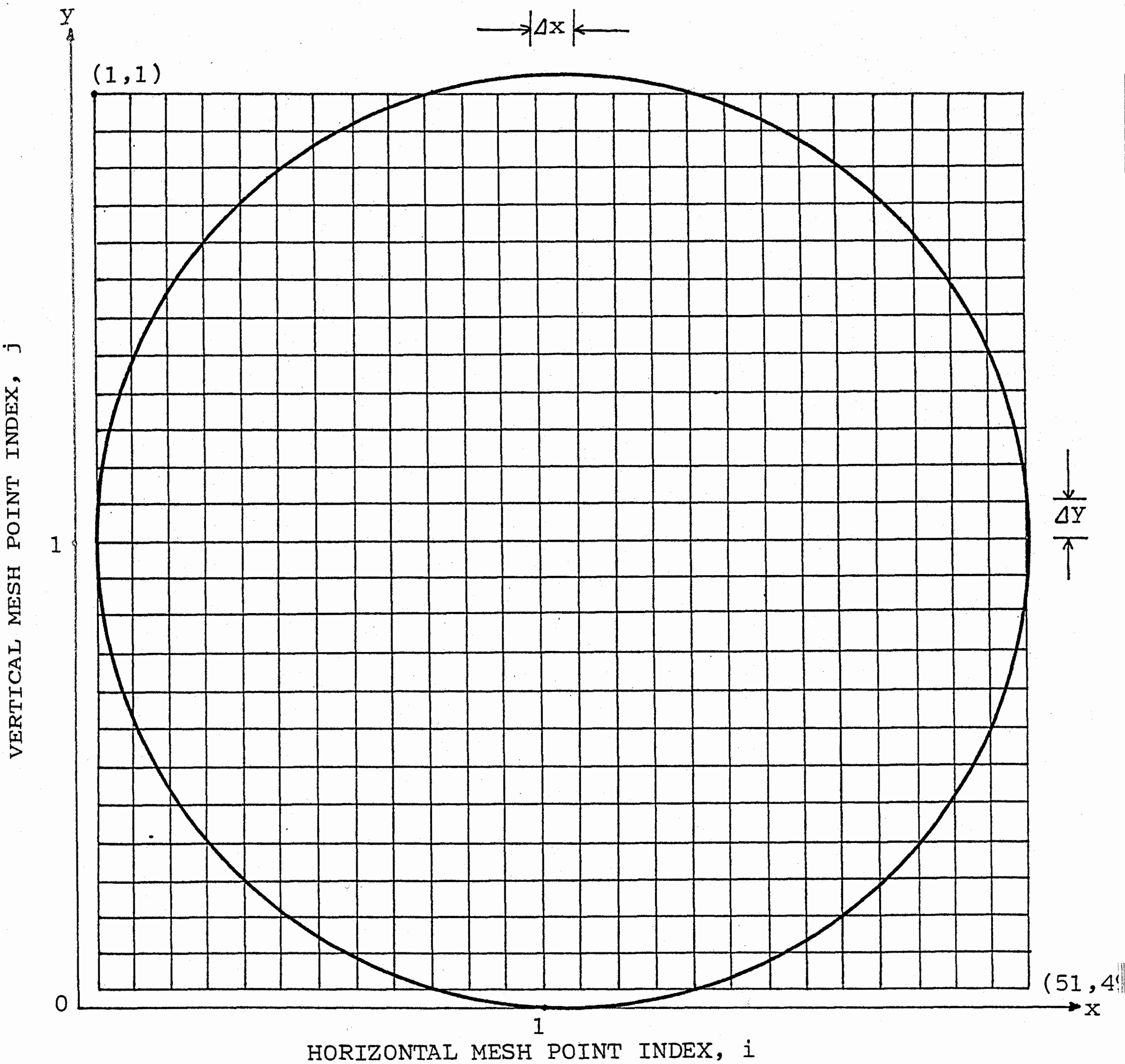

Figure A. Set Up 50 Mesh Spaces in Spin Up Problem.

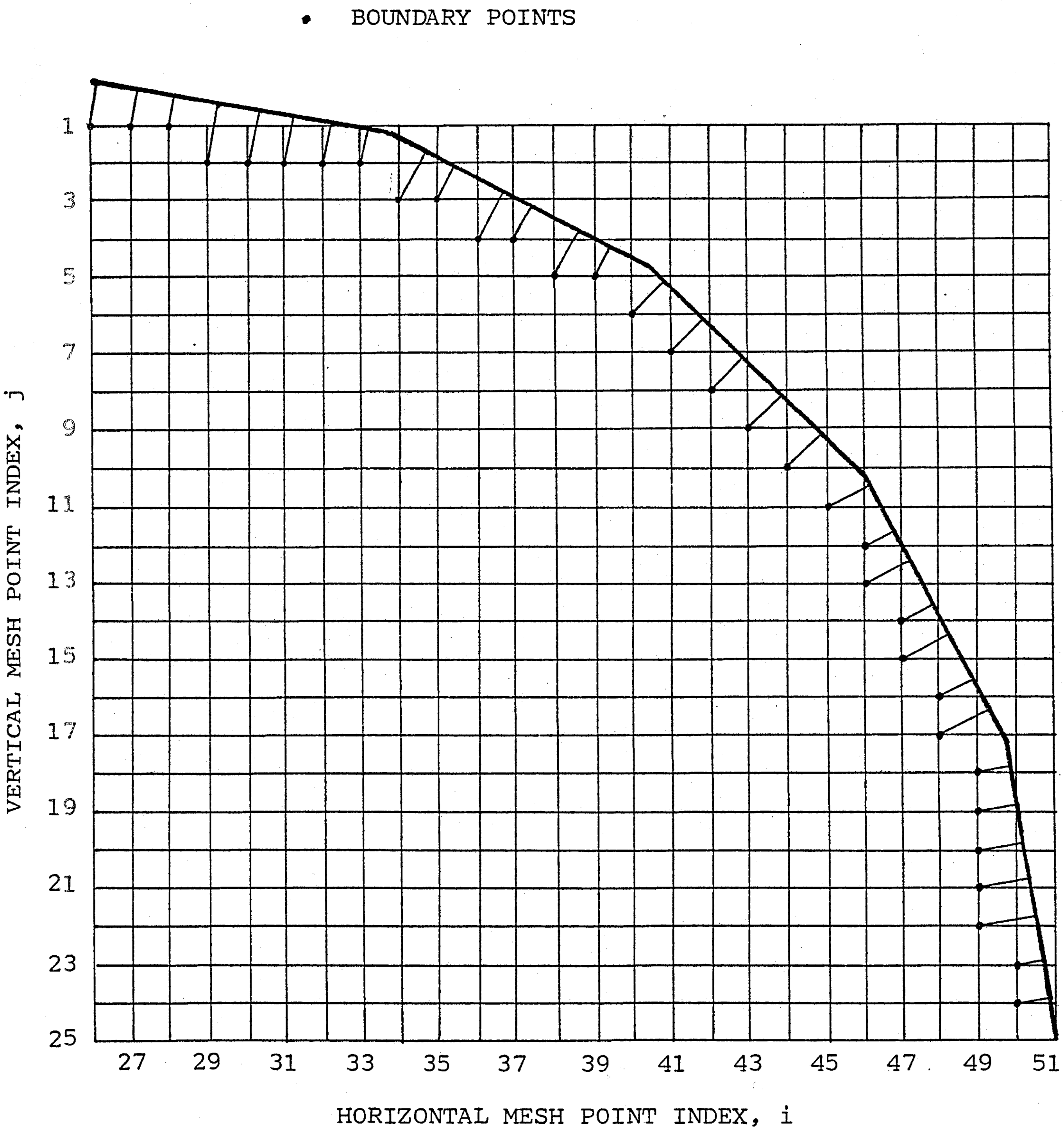

Figure A_1. Mesh Layout at Boundary Points with 20 Vertices for Spin-Up Problem(Upper-Half Region in Polygon).

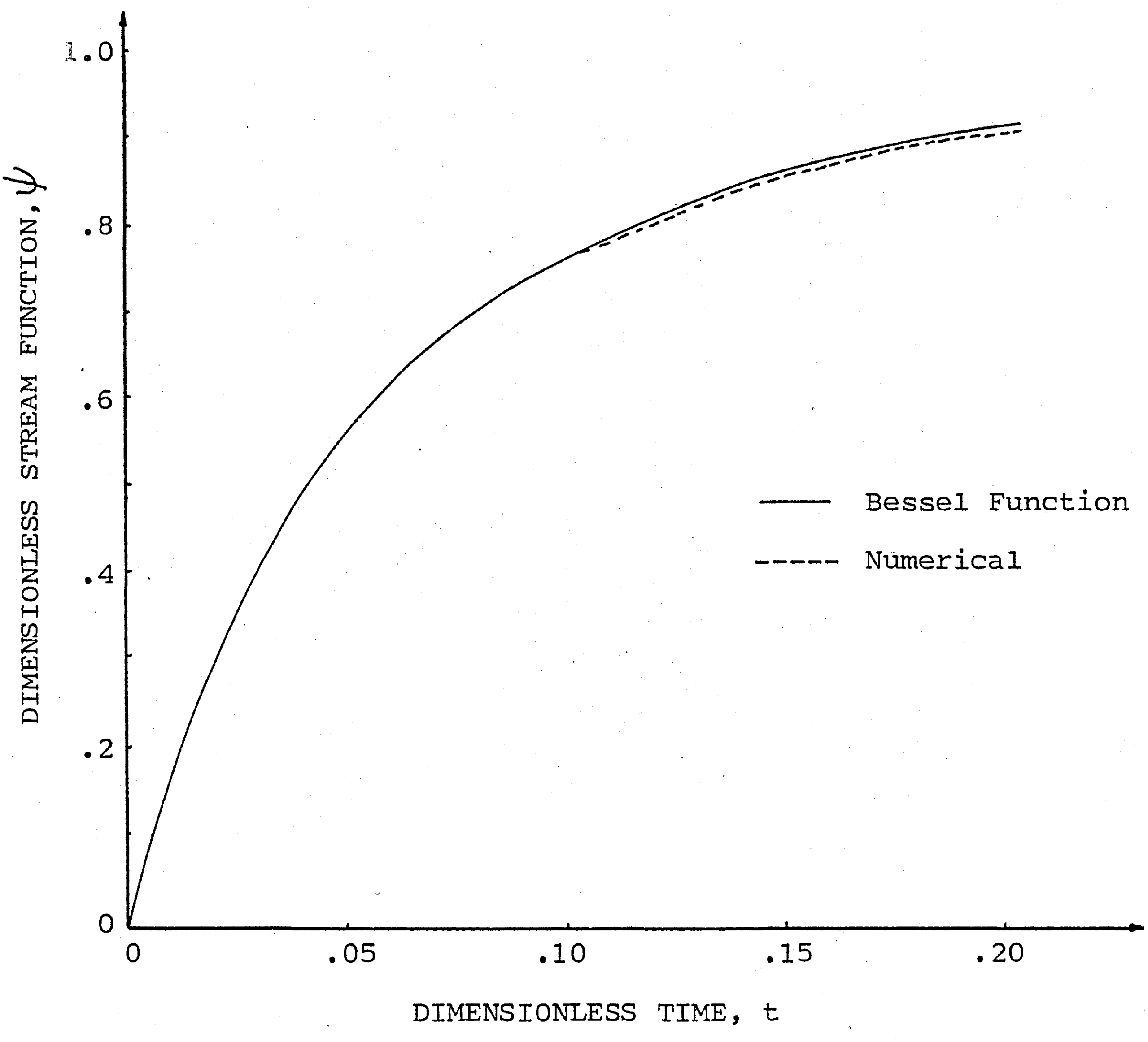

Figure B. Dimensionless Streamfunction vs. Dimensionless Time at Center of Cylinder in Spinup Problem-Comparison of Bessel Function Solution and Numerical Solution.

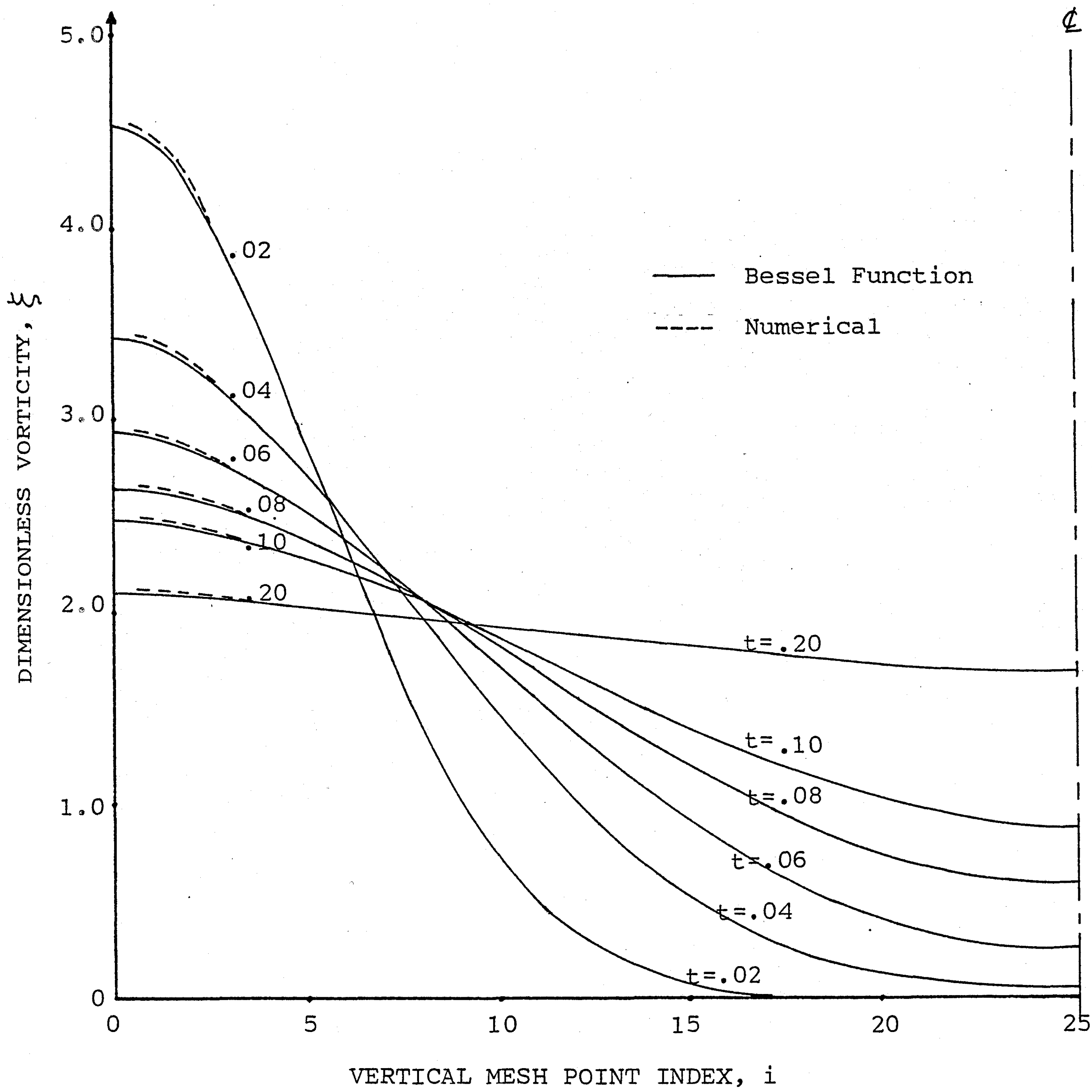

Figure C. Dimensionless Vorticity vs. Mesh Point at 26th Column at Various Dimensionless Time(Re=1, θ =360°)- Comparison of Bessel Function Solution and Numerical Solution.

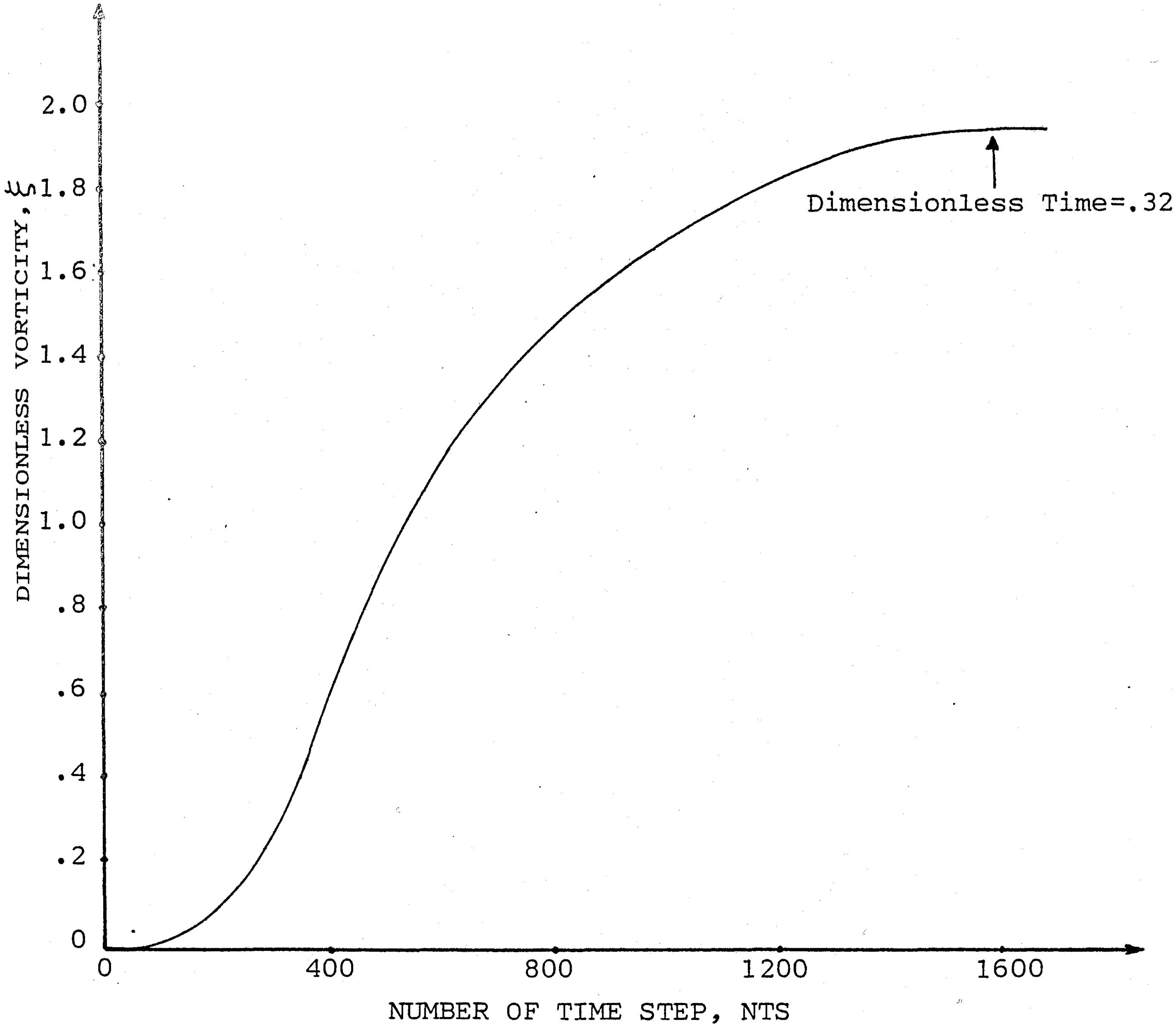

Figure D. Dimensionless Vorticity vs. Number of Time Step at Center(Re=1, θ=360°, 1 Time Step=0.0002).

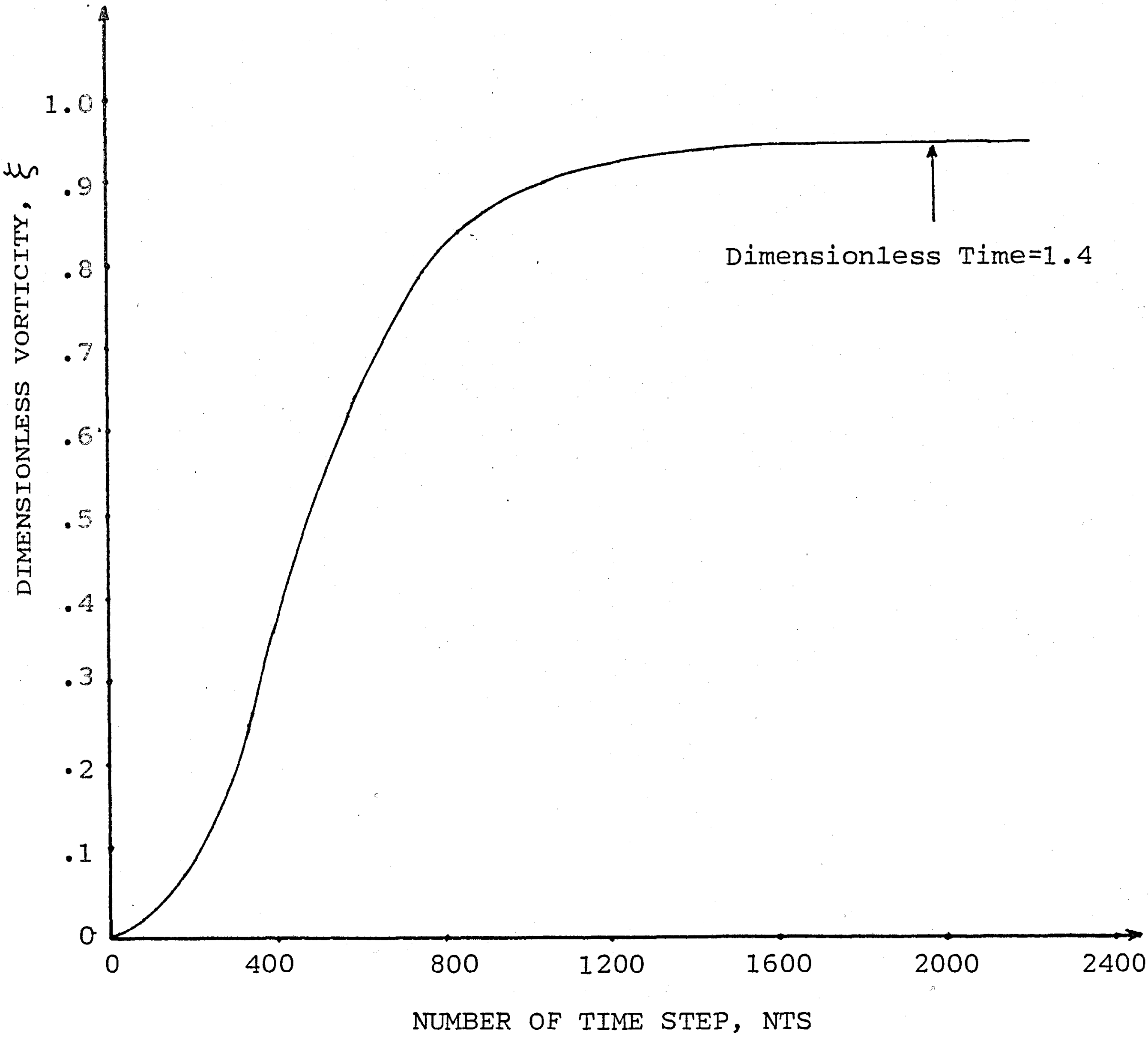

Figure E. Dimensionless Vorticity vs. Number of Time Step at Center(Re=3.16, θ=180°, 1 Time Step=0.0007).

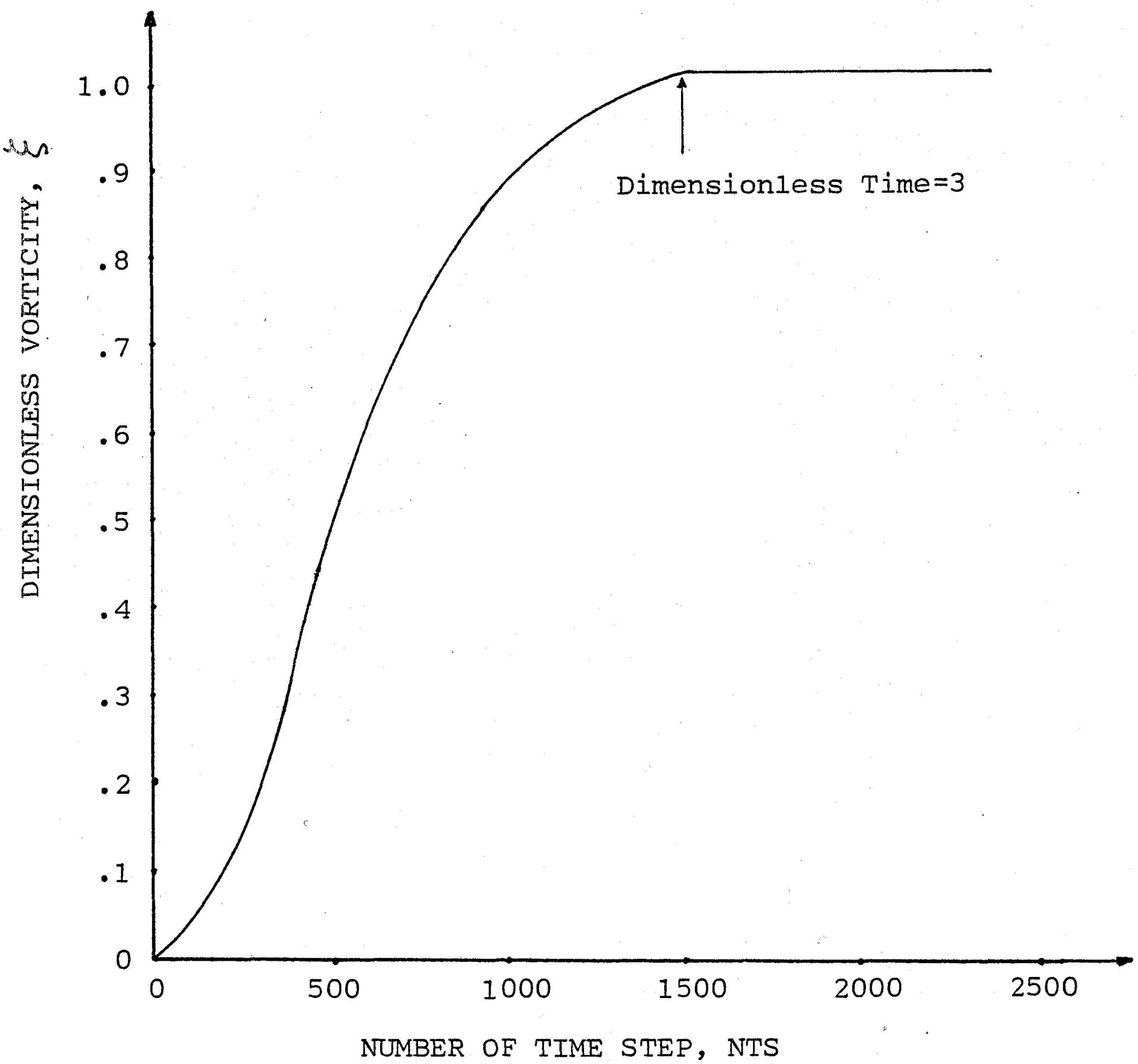

Figure F. Dimensionless Vorticity vs. Number of Time Step at Center(Re=10, θ=180°, 1 Time Step=0.002).

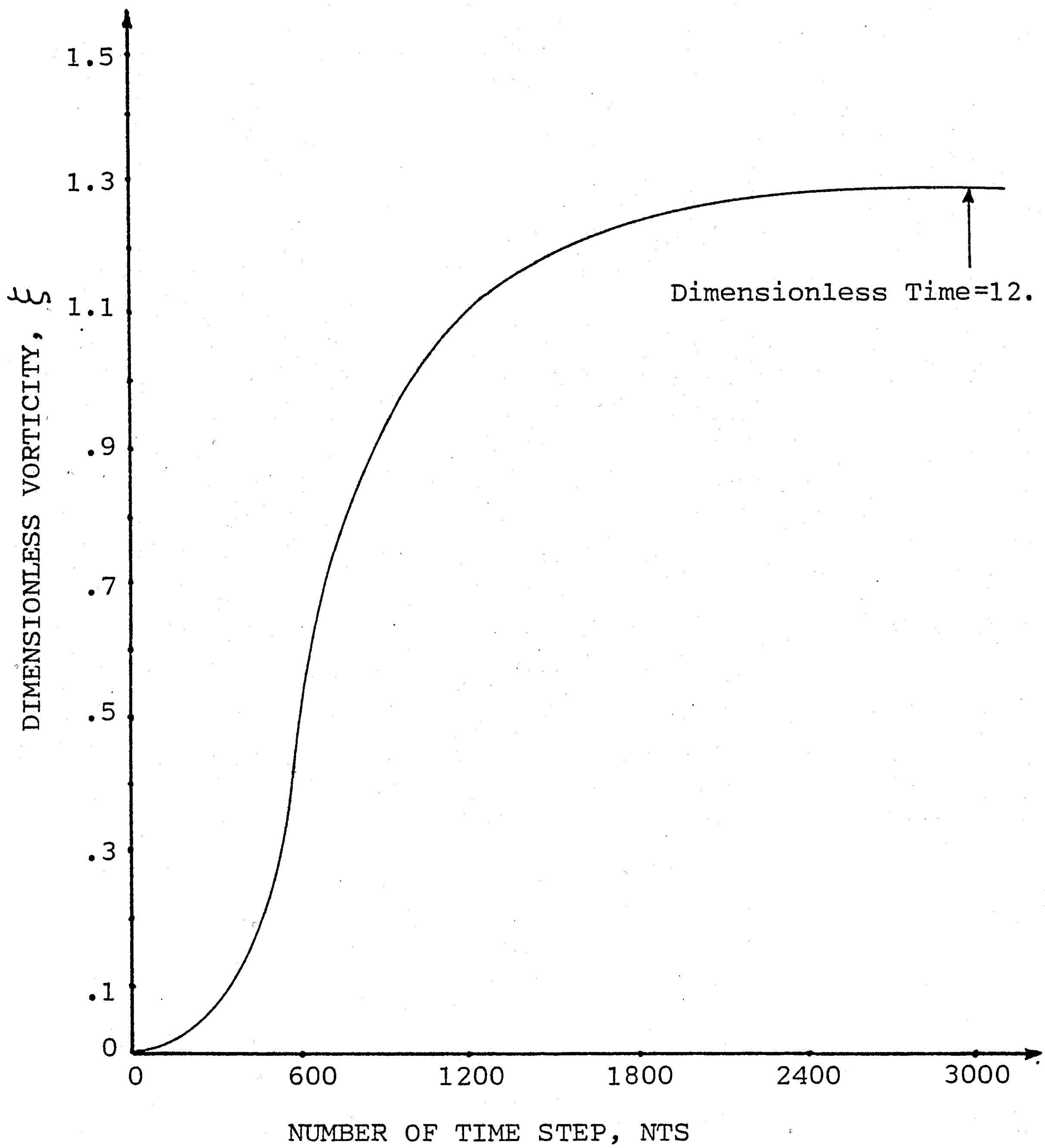

Figure G. Dimensionless Vorticity vs. Number of Time Step at Center(Re=31.6, θ=180°, 1 Time Step =0.004).

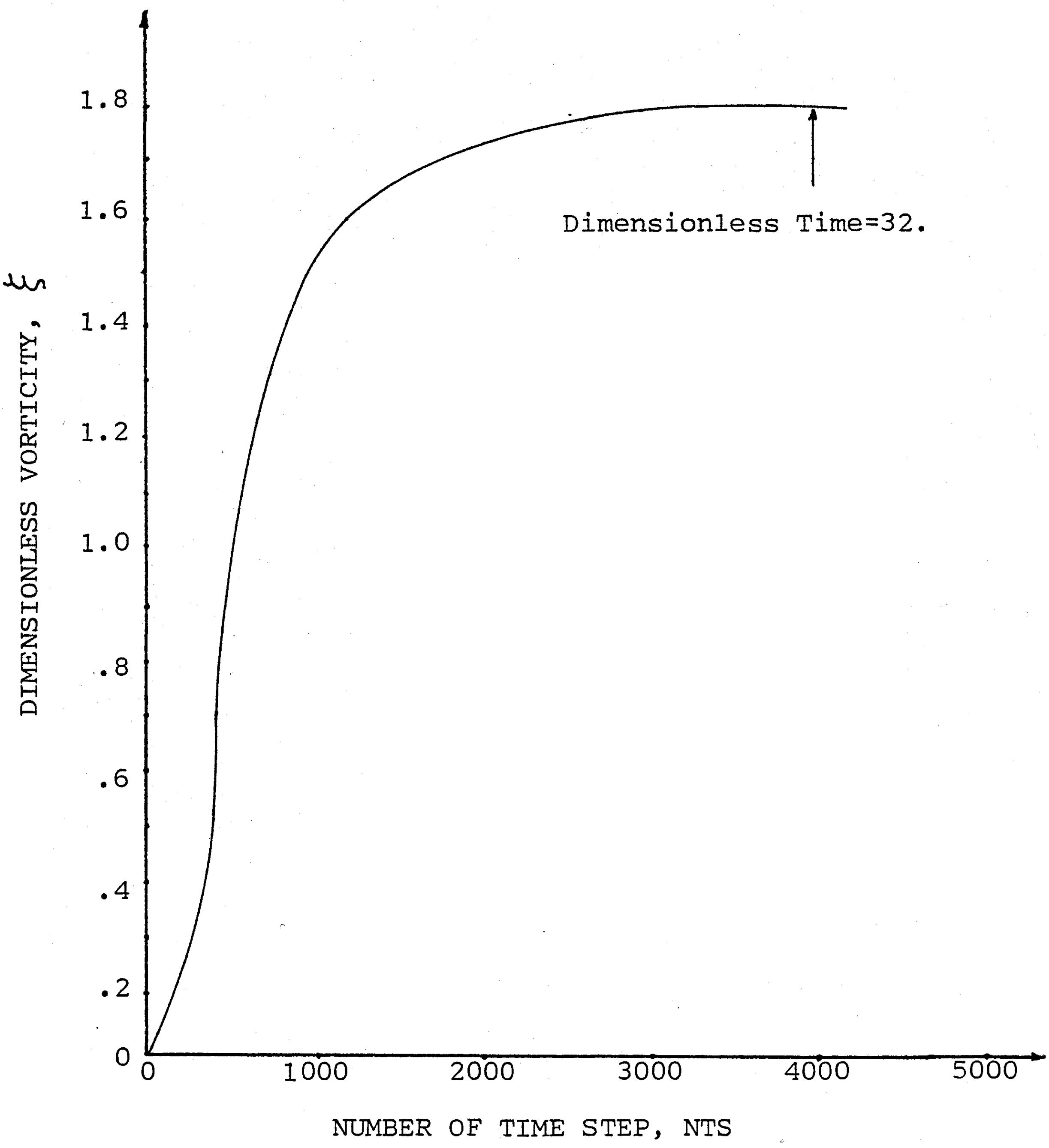

Figure H. Dimensionless Vorticity vs. Number of Time Step at Center(Re=100, θ=180°, 1 Time Step =0.008).

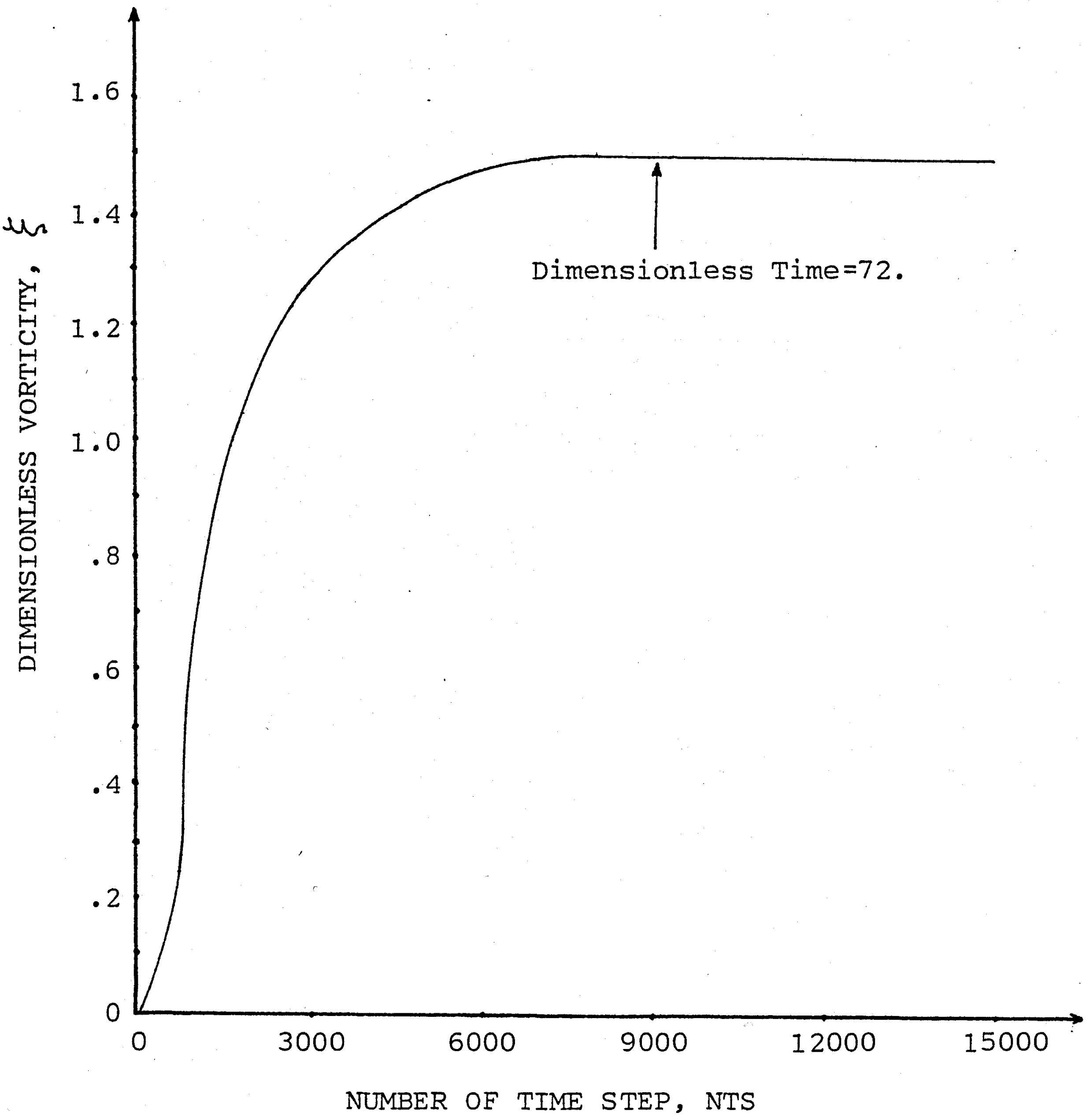

Figure I. Dimensionless Vorticity vs. Number of Time Step at Center(Re=316, θ=180°, 1 Time Step=0.008).

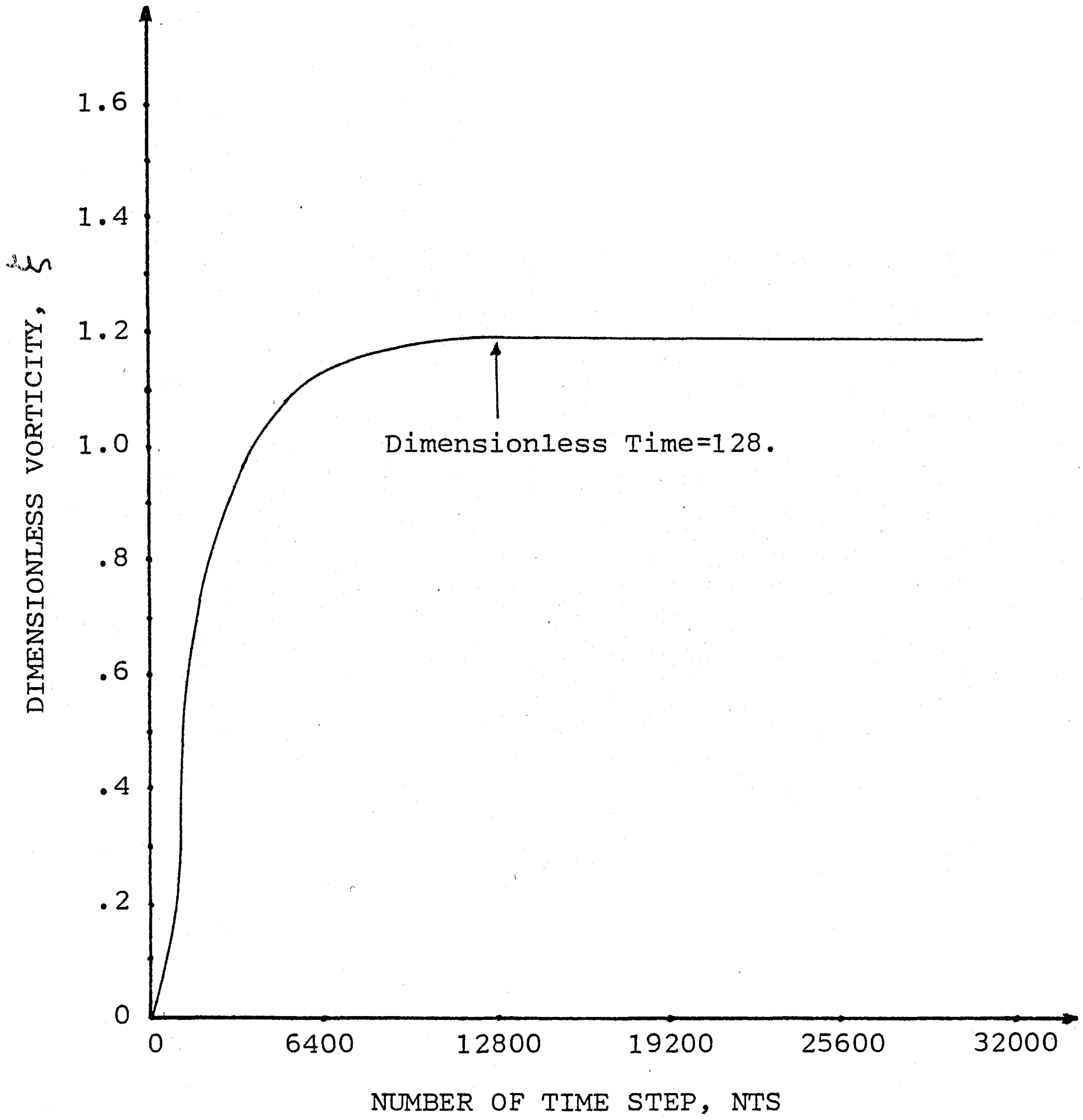

Figure J. Dimensionless Vorticity vs. Number of Time Step at Center(Re=1000, θ=180°, 1 Time Step=0.01).

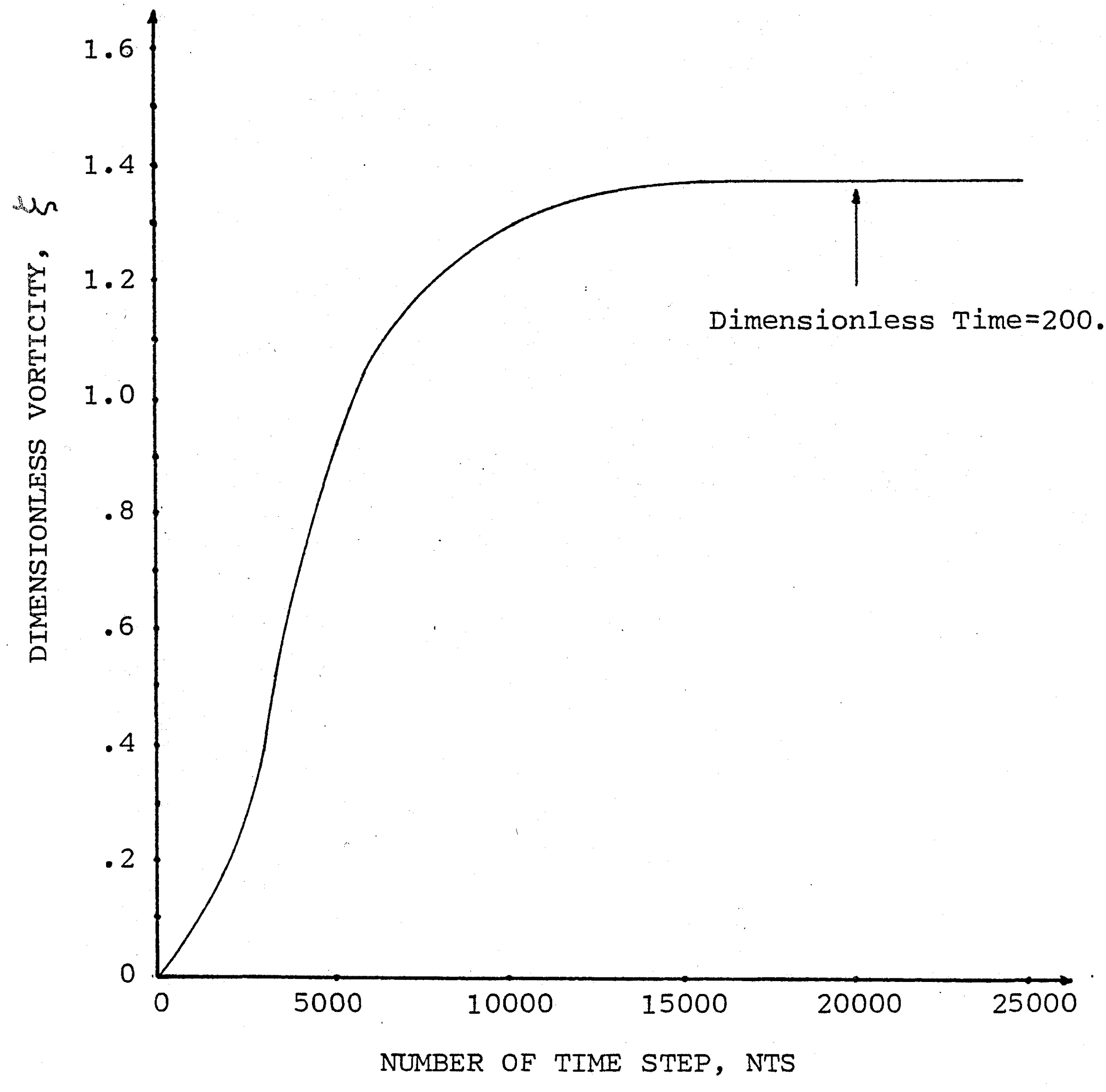

Figure K. Dimensionless Vorticity vs. Number of Time Step at Center(Re=3160, θ=180°, 1 Time Step=0.01).

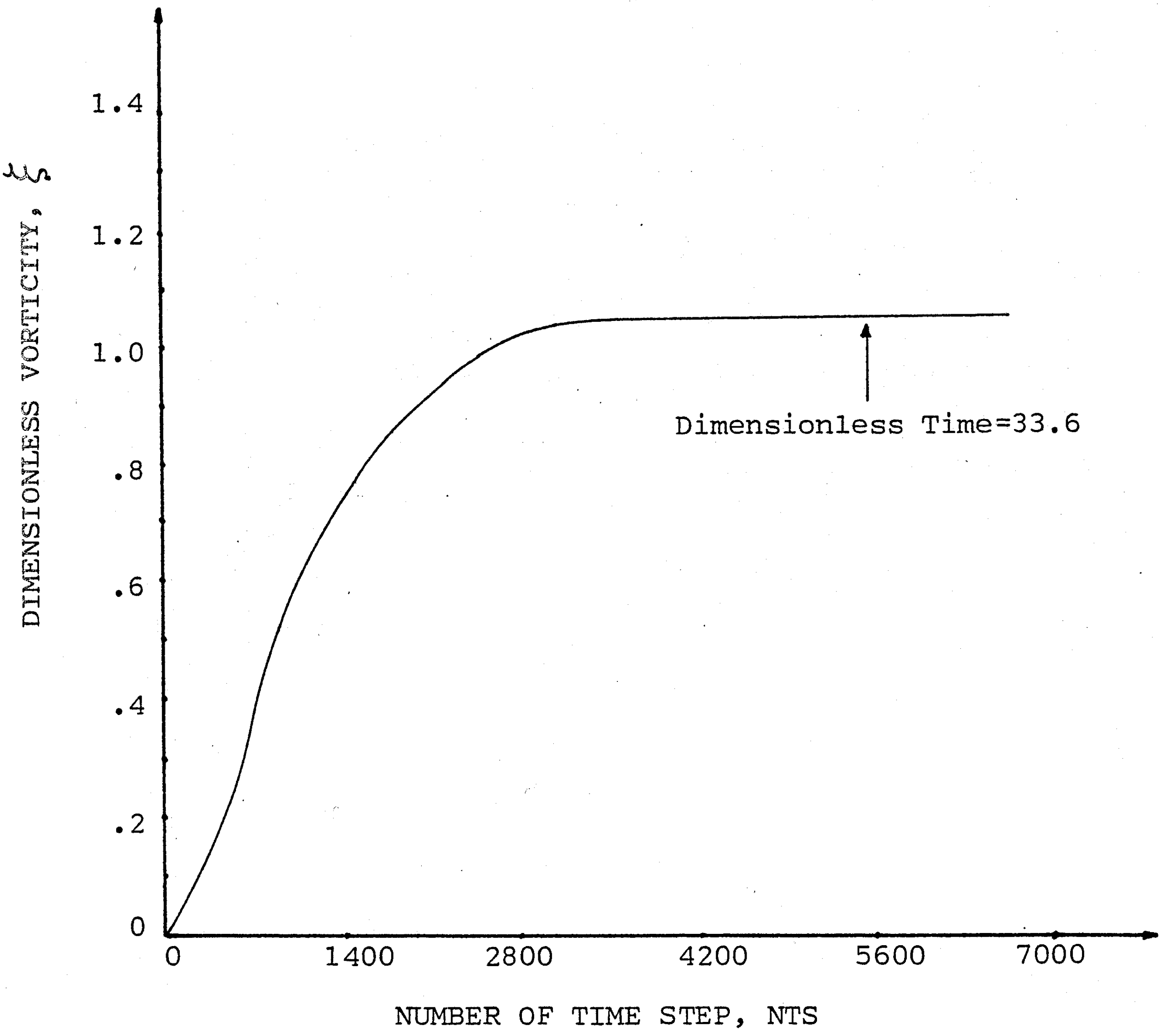

Figure L. Dimensionless Vorticity vs. Number of Time Step at Center(Re=100, θ=90°, 1 Time Step=0.006).

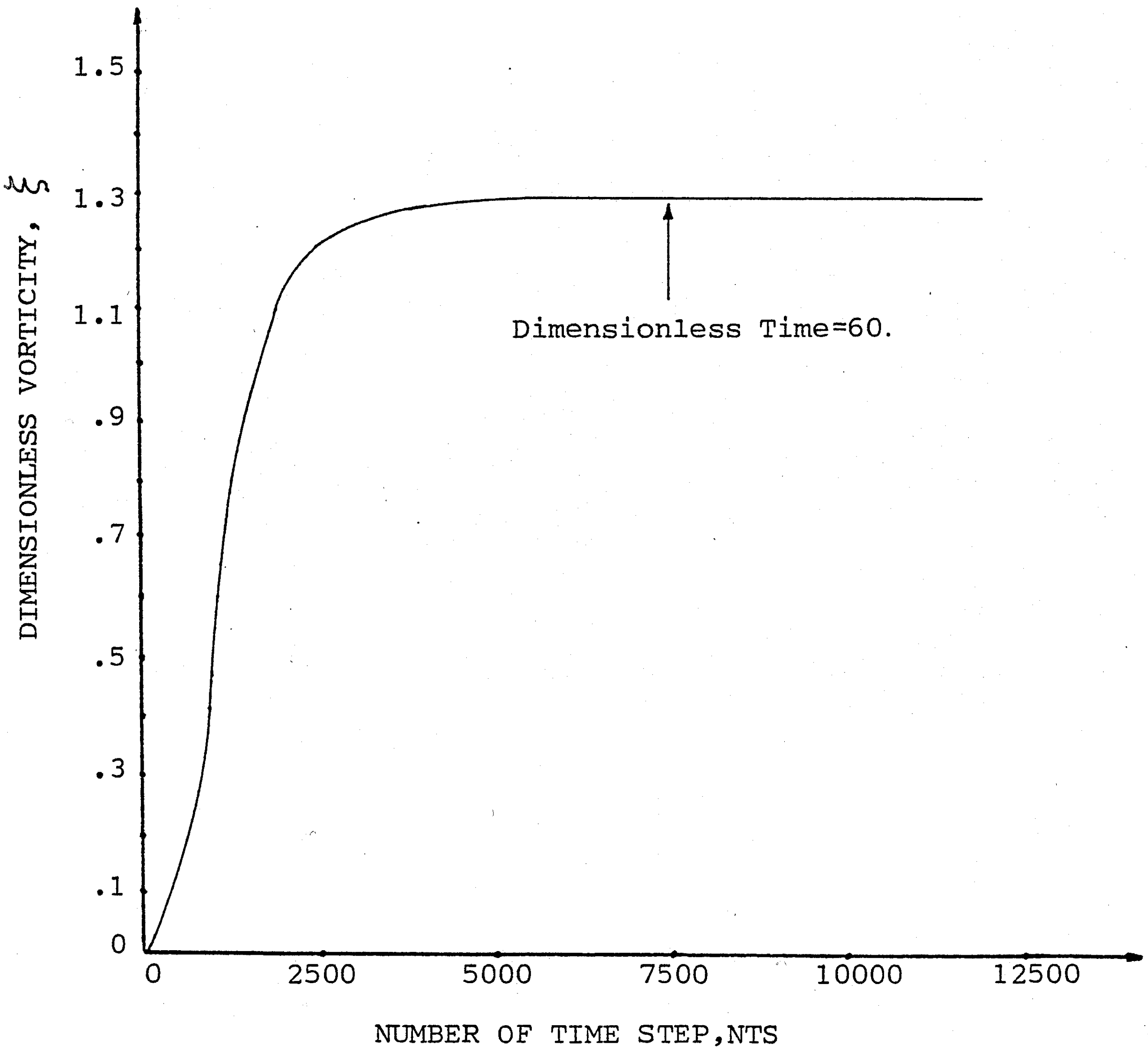

Figure M. Dimensionless Vorticity vs. Number of Time Step at Center(Re=316, θ=90°, 1 Time Step=0.008).

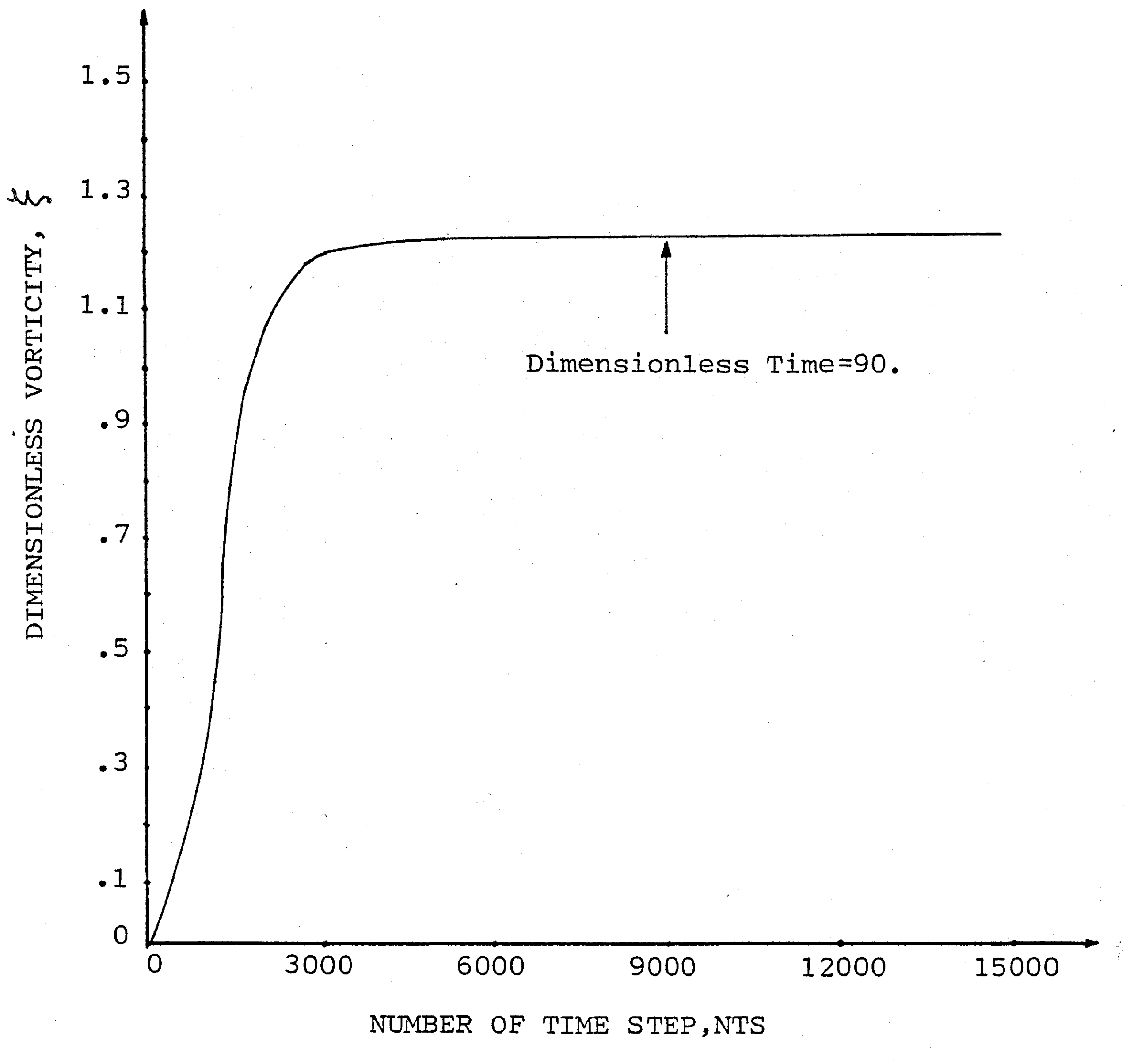

Figure N. Dimensionless Vorticity vs. Number of Time Step at Center(Re=1000, θ=90°, 1 Time Step=0.01).

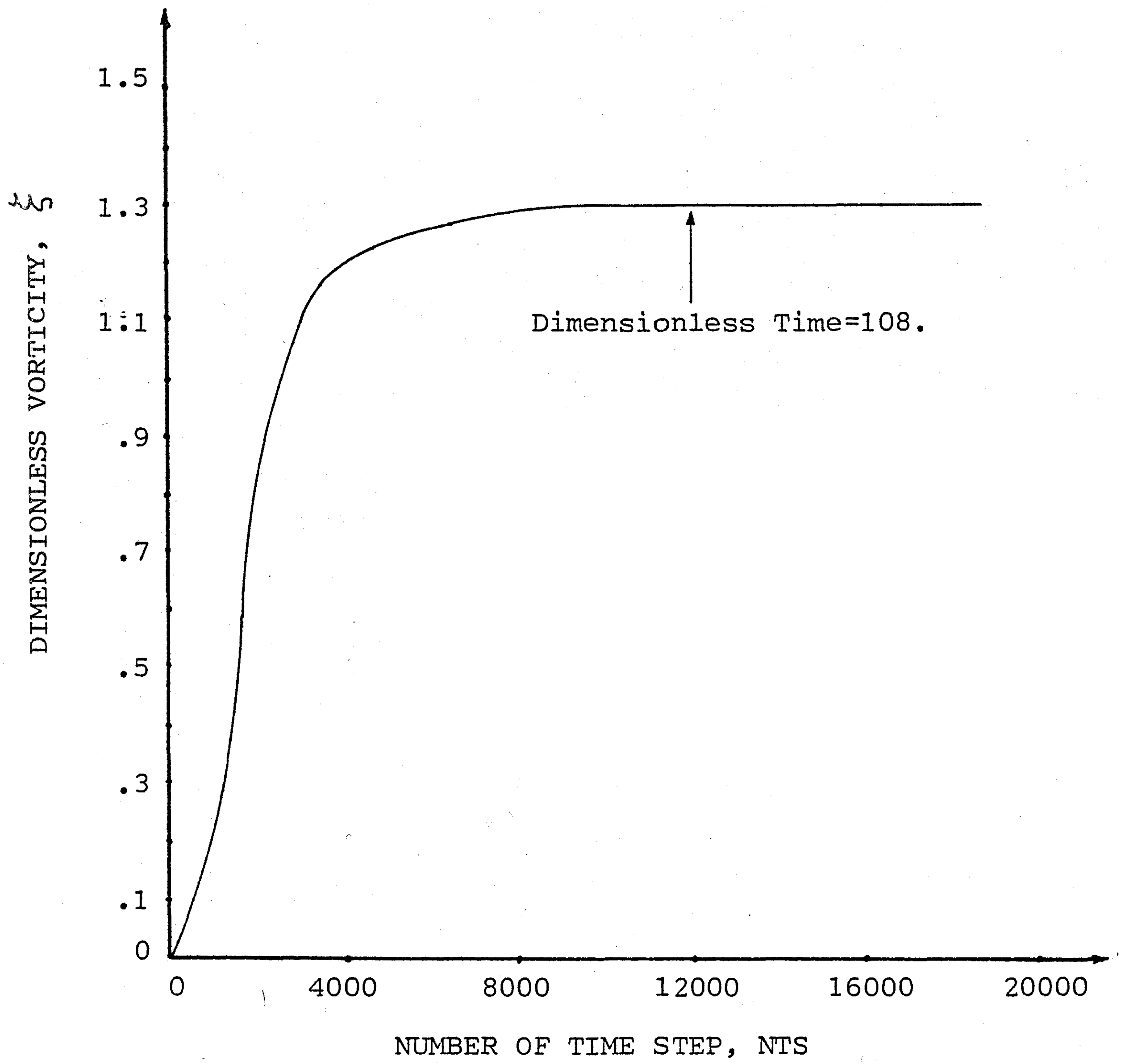

Figure 0. Dimensionless Vorticity vs. Number of Time Step at Center(Re=3160, θ=90°, 1 Time Step=0.009).

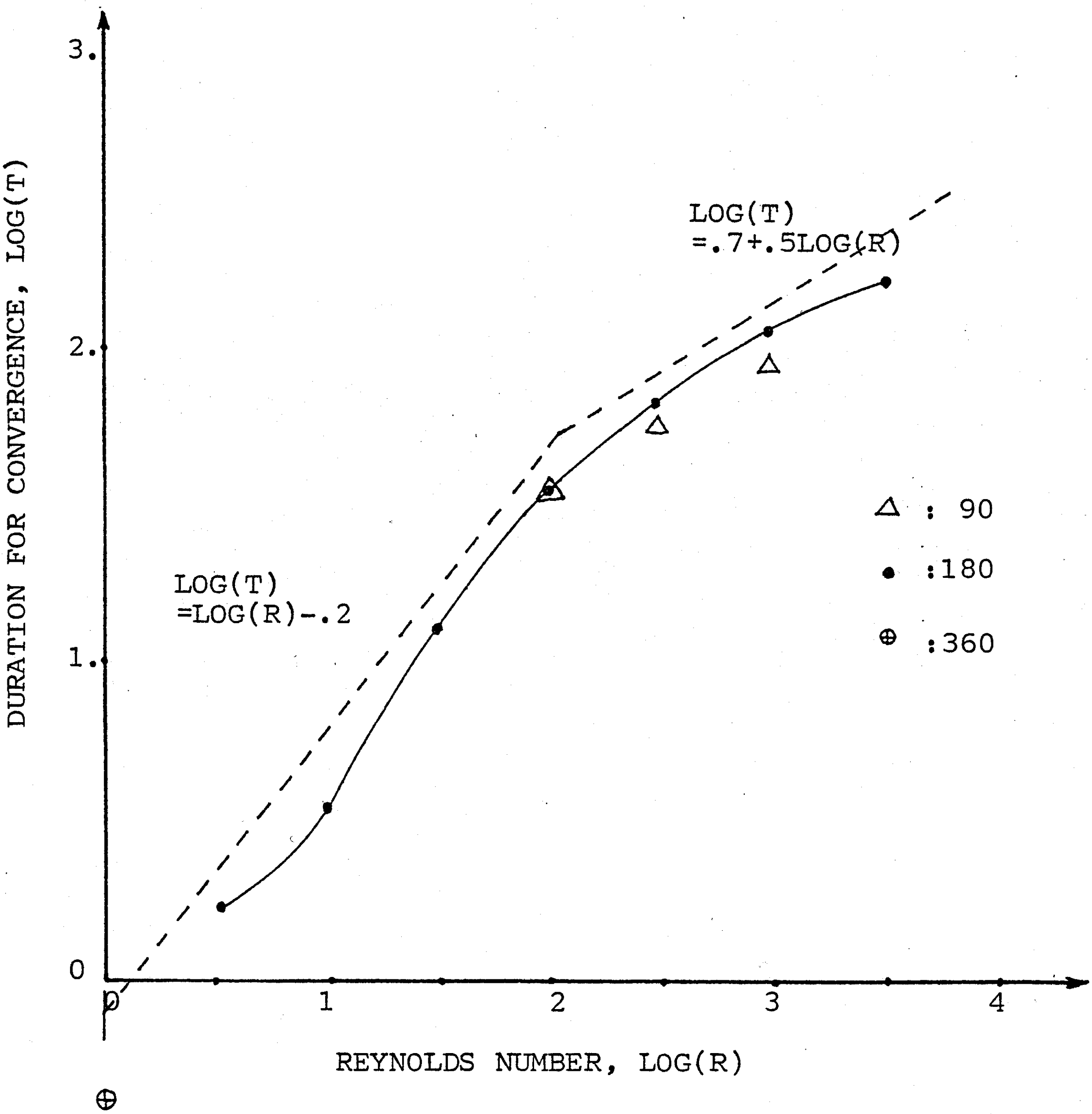

Figure P. Duration for Convergence(T) vs. Reynolds Number in Logarithmic Scale at $\theta=90^\circ$, 180°, 360°.

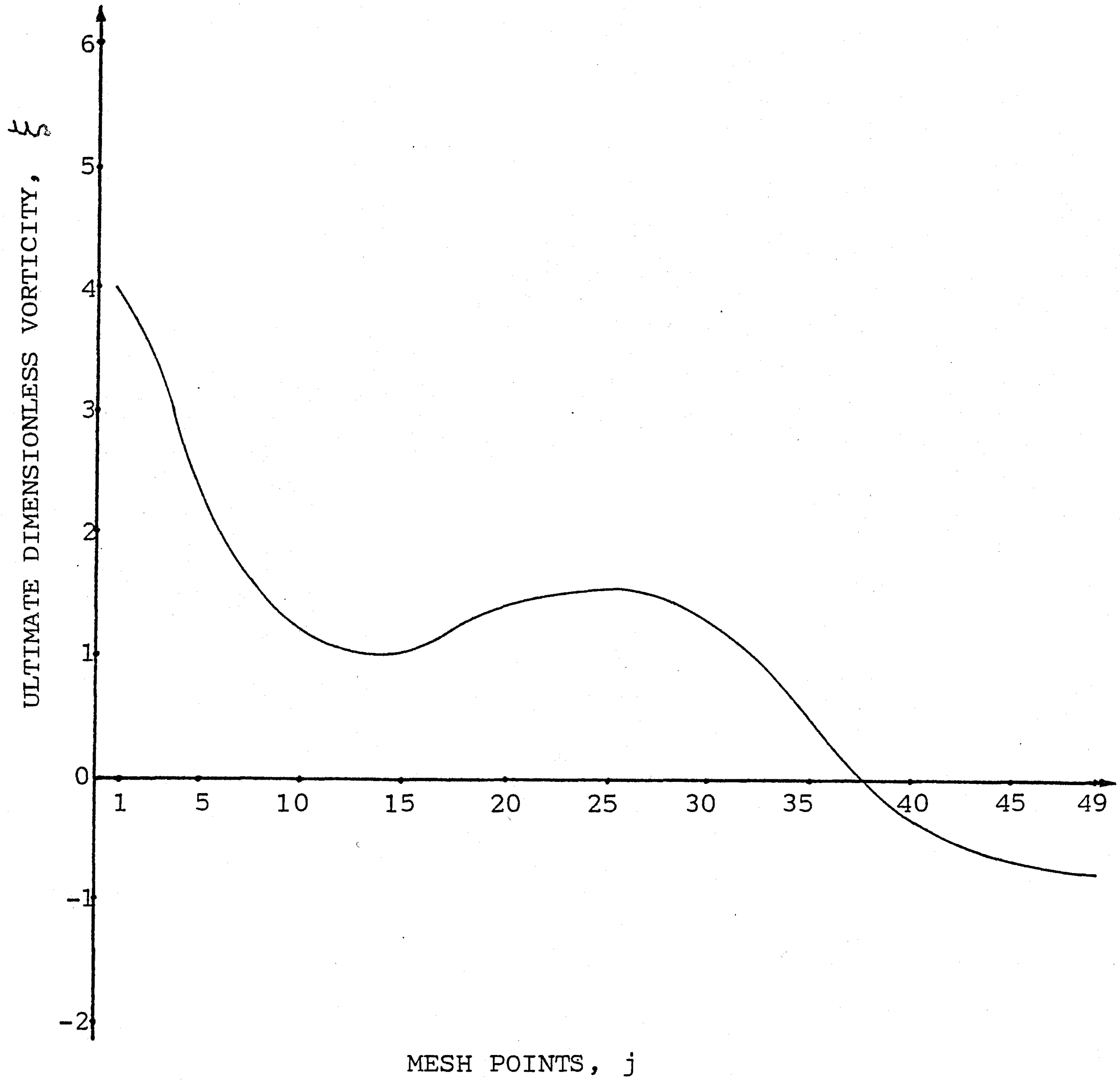

Figure Q. Ultimate Dimensionless Vorticity vs. Mesh Points at 26th Column(Re=100, θ=180°).

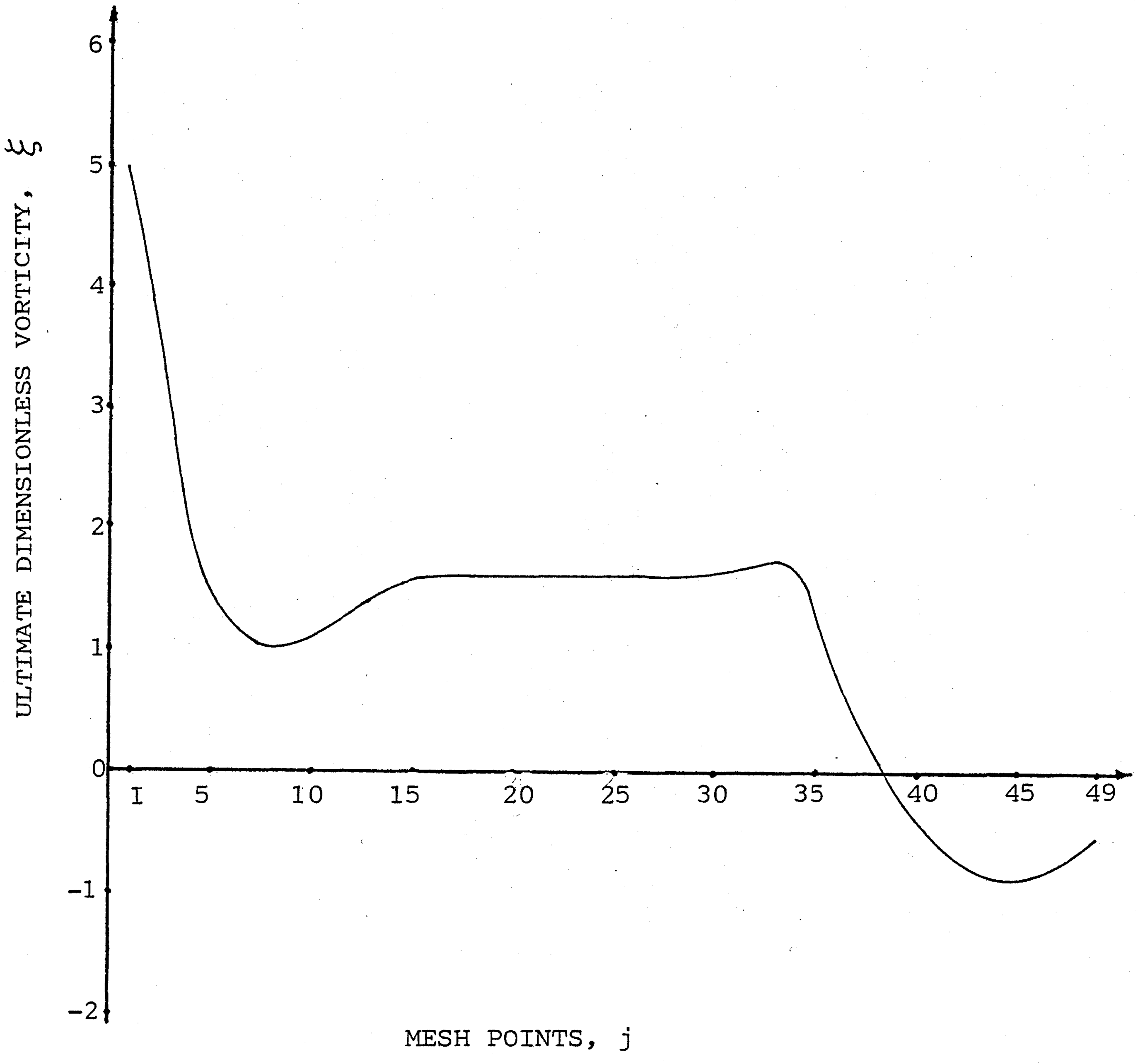

Figure R. Ultimate Dimensionless Vorticity vs. Mesh Points at 26th Column(Re=1000, θ=180°).

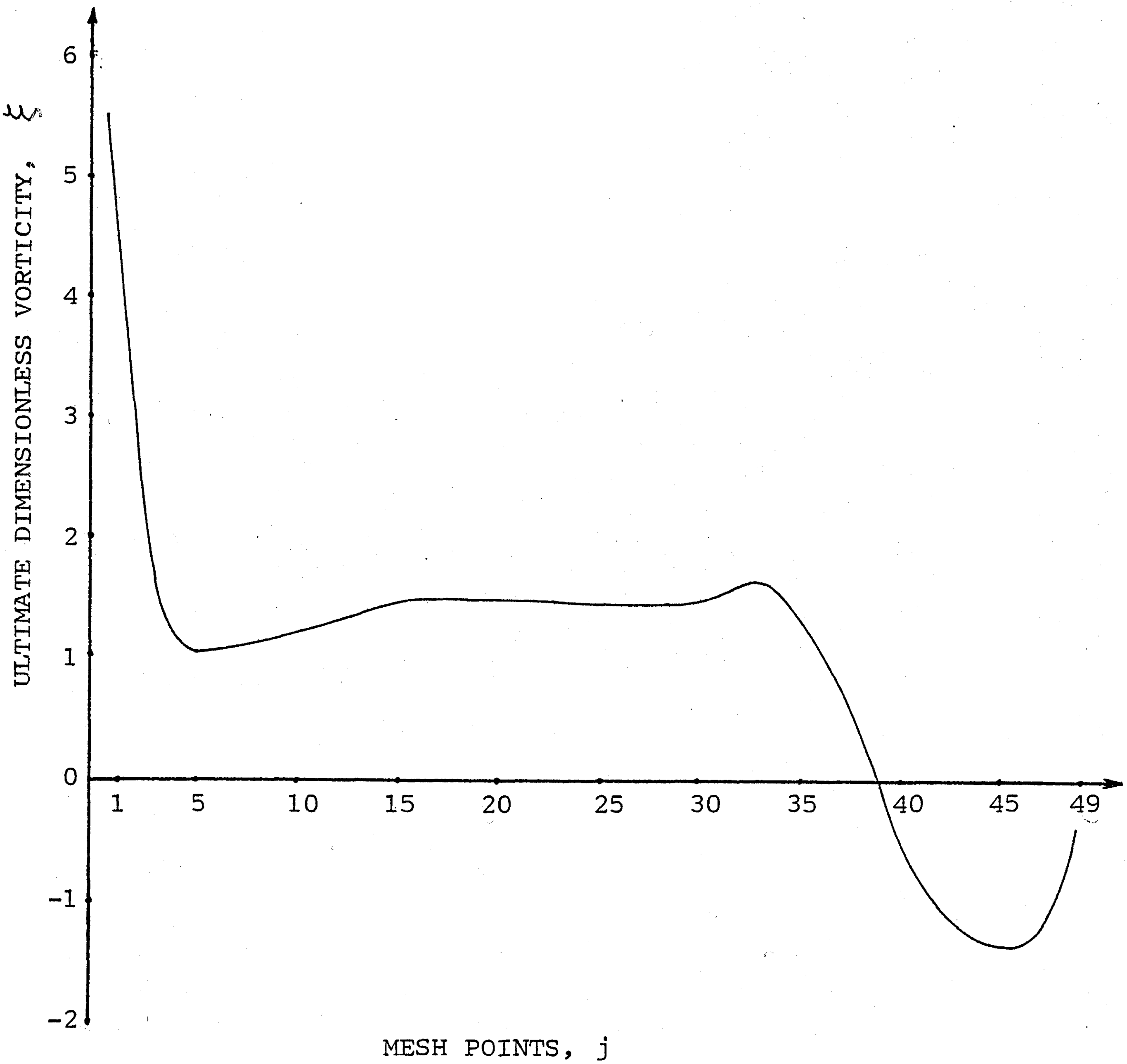

Figure S. Ultimate Dimensionless Vorticity vs. Mesh Points at 26th Column(Re=3160, θ=180°).

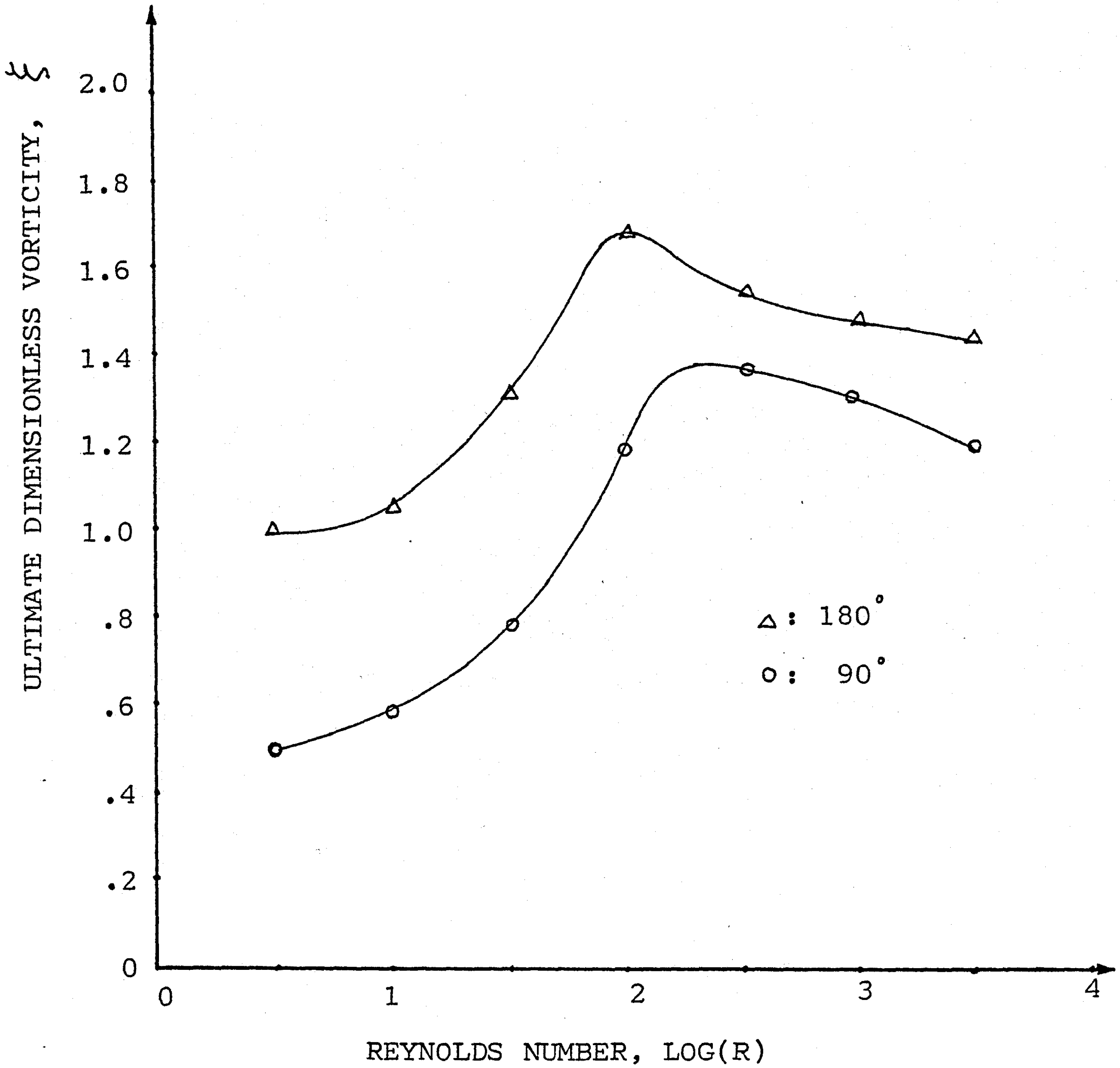

Figure T. Ultimate Dimensionless Vorticity at Center vs. Reynolds Number in Logarithmic Scale(θ=90°, 180°).

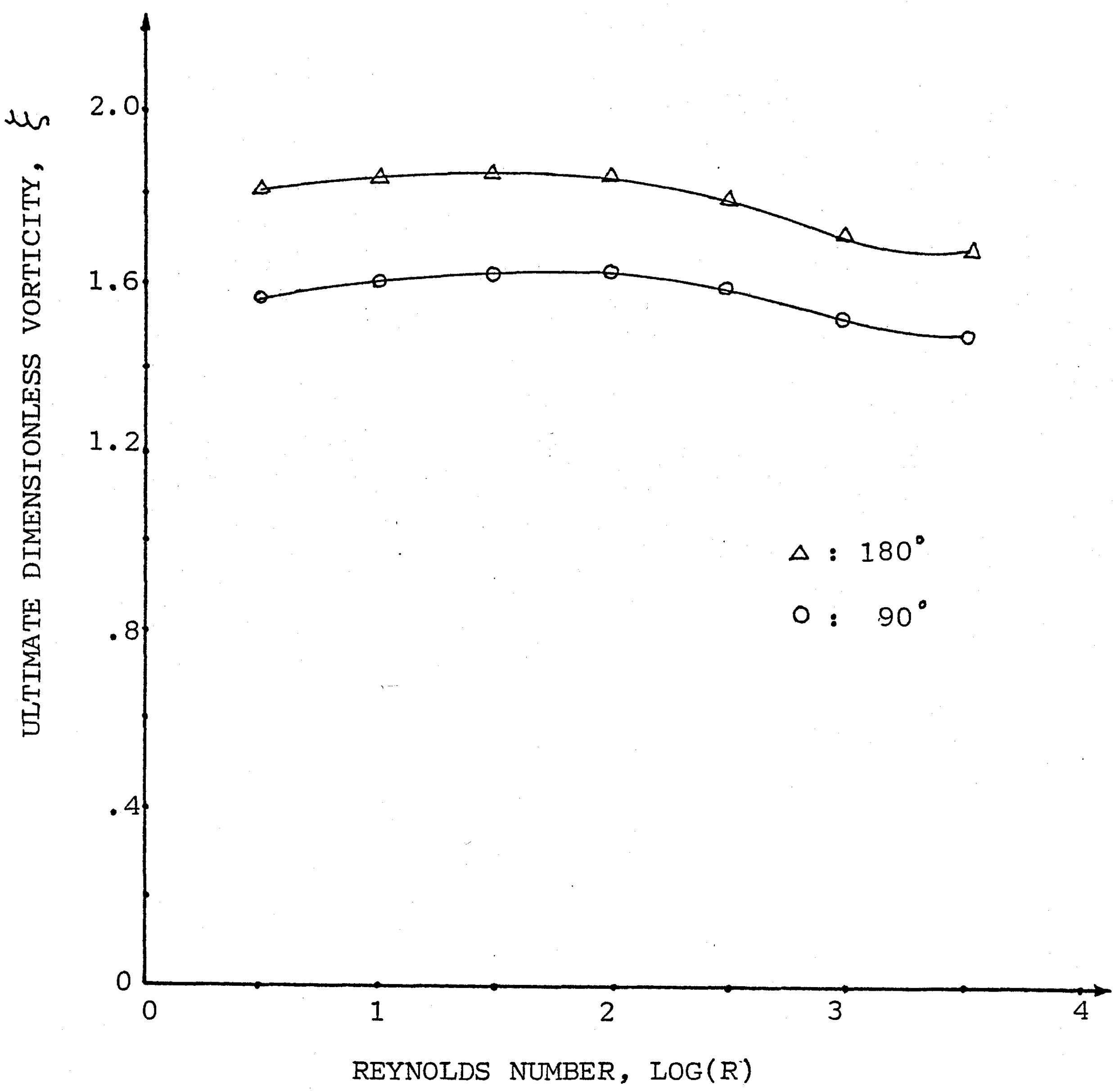

Figure U. Ultimate Dimensionless Vorticity at Center of Vortex vs. Reynolds Number in Logarithmic Scale (θ=90°, 180°).

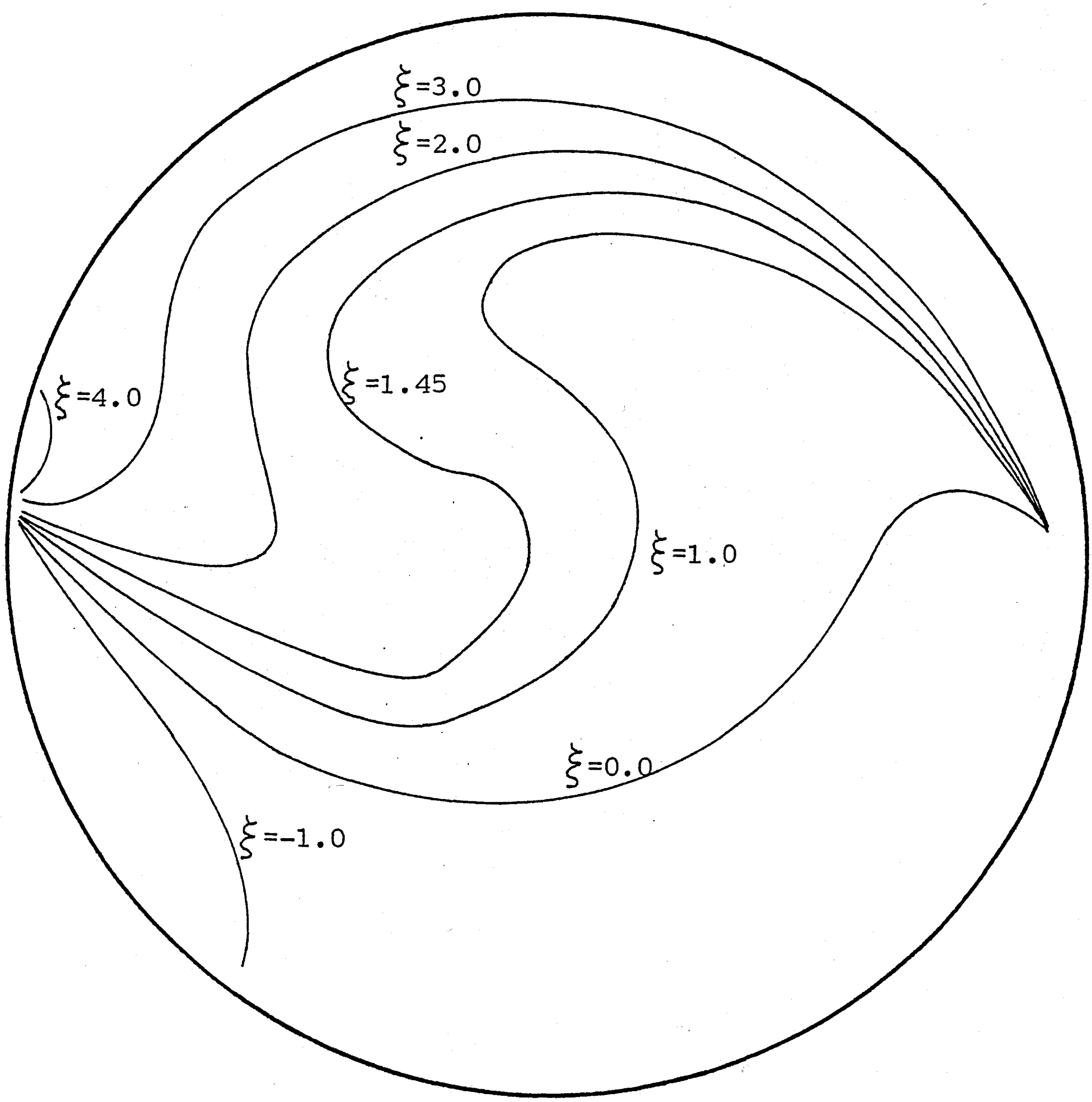

Figure V. Dimensionless Equivorticity Line at R=100 and θ=180°.

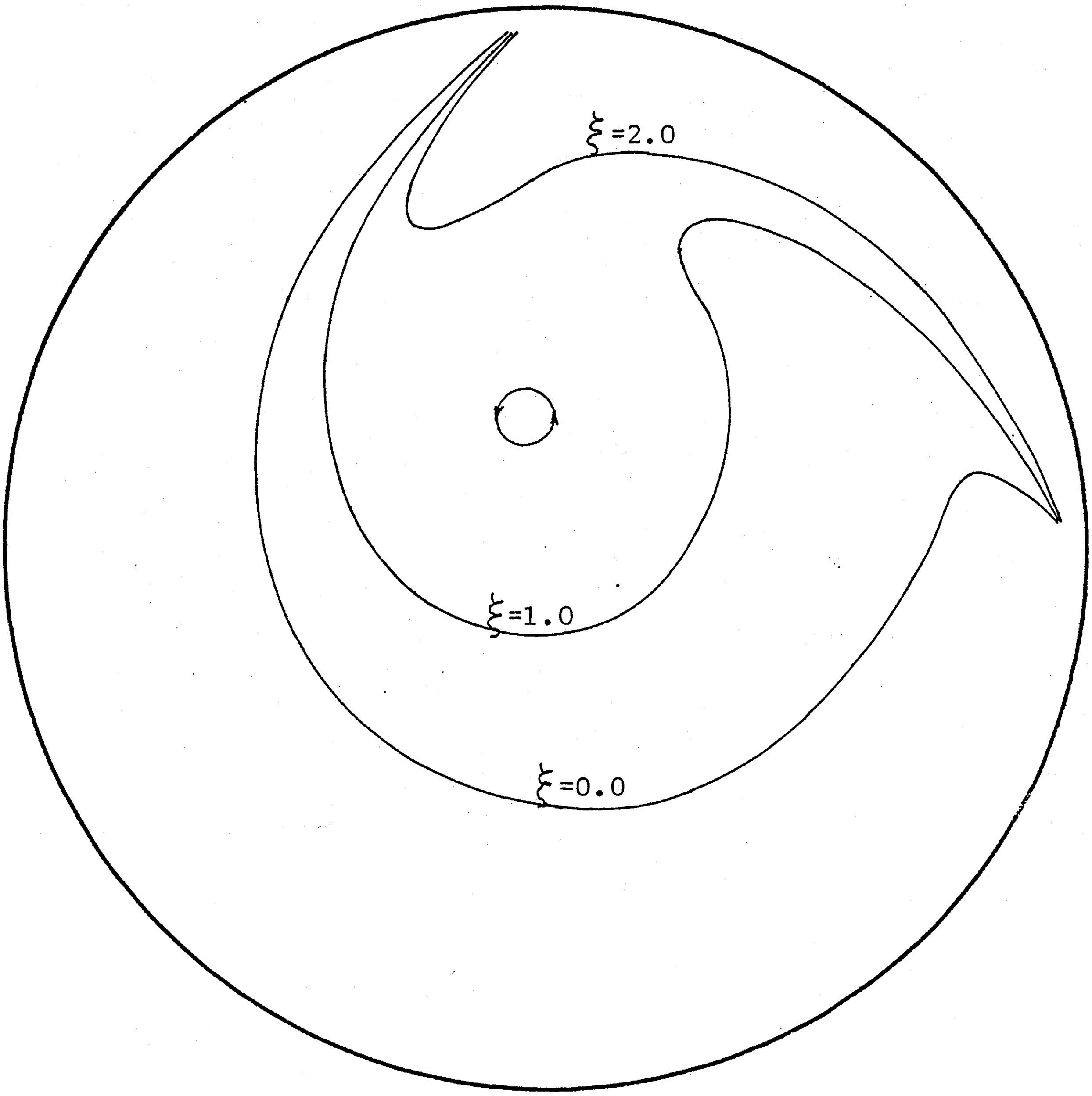

Figure W. Dimensionless Equivorticity Line at R=316 and θ=90°.

APPENDIX B

COMPUTER PROGRAM FOR ANALYTICAL SOLUTION

The program which calculates the dimensionless stream function and vorticities for the two-dimensional incompressible spin up problem ($\theta = 360^{\circ}$, Re = 1) by analytical works is listed below. The principal symbols used in the program and their definitions are as follows:

ALPHA(100)	zeros of J_1
ALPJO(100)	$J_o(\alpha)$
ALPJOR(100,20)	$J_o(\alpha r)$
ARGO	argument of $J_o(\alpha r)$
MMBSJO	Bessel function of order 0
MMBSJ1	Bessel function of order 1
PI	π, the geometry constant
R	radius
XJJO	temporary value of $J_o(\alpha r)$
XJJ1	temporary value of $J_1(\alpha r)$
XJO	temporary value of $J_o(\alpha)$

```
      DIMENSION ALPHA(100),R(21),ALPJO(100),ALPJOR(100,21)
      DOUBLE PRECISION ALPHA,ARGO,XJJ1,XJJO,XJO,PI,R
     &,MMBSJO,MMBSJ1,ARG
      DATA PI,T/3.14159D0,0.0/
      EXTERNAL MMBSJ1,MMBSJO,SPEED,VORT,FLOW
      DATA ALPHA/3.8317D0,7.0156D0,10.1735D0,13.3237D0,
     &16.4706D0,19.6159D0,22.7601D0,25.9037D0,29.0468D0,
     &32.1897D0,35.3323D0,38.4748D0,41.6171Do,44.7593D0,
     &47.9015D0,51.0435D0,54.1856D0,57.3275D0,60.4695D0,
     &63.6114D0,66.7532D0,69.8951D0,73.0369D0,76.1787D0,
     &79.3205D0,82.4623D0,85.6040D0,88.7458D0,91.8875D0,
     &95.0292D0,98.171D0,101.3127D0,104.4544D0,107.5961D0,
     &110.7378D0,113.8794D0,117.0211D0,120.1628D0,123.3045D0,
     &126.4461D0,60*1.00D0/
      DO 30 II=1,60
      ALPHA(40+II)=ALPHA(39+II) + PI
30    CONTINUE
      R(1)=0.0
      DO 20 NNN=1,20
      R(NNN+1)=R(NNN)+.05D0
20    CONTINUE
      DO 50 N=1,100
      ARG=ALPHA(N)
      XJO=MMBSJO(ARG,IER)
      ALPJO(N)=XJO
50    CONTINUE
       DO 55 LL=1,100
       DO 55 LLL=1,21
       ARGO=ALPHA(LL)*R(LLL)
       XJJO=MMBSJO(ARGO,IER)
       ALPJOR(LL,LLL)=XJJO
55     CONTINUE
       DELTIM=.01
       DO 60 IMET=1,15
       CALL SPEED(R,T,ALPJO,ALPHA)
       CALL VORT(R,T,ALPJOR,ALPJO,ALPHA)
       CALL FLOW(R,T,ALPJOR,ALPJO,ALPHA)
       T=T+DELTIM
60     CONTINUE
       STOP
       END
       SUBROUTINE SPEED(R,T,ALPJO,ALPHA)
       DIMENSION ALPJO(100),ALPHA(100),R(21)
       DOUBLE PRECISION ALPHA,R,XJJ1,ARG1,MMBSJ1
       EXTERNAL MMBSJ1
       DO 80 K=1,21
       VELSUM=0.0
```

```
      DO 85 KK=1,100
      ARG1=ALPHA(KK)*R(K)
      XJJ1=MMBSJ1(ARG1,IER)
      EXARG=ALPHA(KK)*ALPHA(KK)*T
      IF(EXARG.LE.88.0)GO TO 83
      VEL=0.0
      GO TO 84
83    VEL=(2.*EXP(-EXARG)/ALPHA(KK)*(XJJ1/ALPJO(KK))
84    VELSUM=VELSUM+VEL
85    CONTINUE
      VELTOT=VELSUM+R(K)
      WRITE(5,86)T,R(K),VELTOT
86    FORMAT('T=',F8.2,'R=',F8.2,'     VELOCITY=',F20.8)
80    CONTINUE
      RETURN
      END
      SUBROUTINE VORT(R,T,ALPJOR,ALPJO,ALPHA)
      DIMENSION R(21),ALPJOR(100,21),ALPJO(100),ALPHA(100)
      DOUBLE PRECISION ALPHA,R
      DO 90 J=1,21
      VORSUM=0.0
      DO 95 JJ=1,100
      AXARG=ALPHA(JJ)*ALPHA(JJ)*T
      IF(AXARG.LE.88.0)GO TO 93
      VORTT=0.0
      GO TO 94
93    VORTT=EXP(-AXARG)*(ALPJOR(JJ,J)/ALPJO(JJ))
94    VORSUM=VORSUM+VORTT
95    CONTINUE
      VORTOT=1.+VORSUM
      WRITE(5,100)T,R(J),VORTOT
100   FORMAT('T=',F8.2,'    R=',F8.2,'   VORTICITY=',F20.8)
90    CONTINUE
      RETURN
      END
      SUBROUTINE FLOW(R,T,ALPJO,ALPHA)
      DIMENSION R(21),ALPJOR(100,21),ALPJO(100),ALPHA(100)
      DOUBLE PRECISION ALPHA,R
      DO 110 I=1,21
      PSISUM=0.0
      DO 120 III=1,100
      BXARG=ALPHA(III)*ALPHA(III)*T
      IF(BXARG.LE.88.0)GO TO 103
      PSI=0.0
      GO TO 104
103   PSI=(EXP(-BXARG)/(ALPHA(III))**2.)*(1.-ALPJOR(III,I)
     &/ALPJO(III))
```

```
104   PSISUM=PSISUM+PSI
120   CONTINUE
      PSITOT=(1.-(R(I))**2.)-4.*PSISUM
      WRITE(5,130)T,R(I),PSITOT
130   FORMAT('T=',F8.2,'    R=',F8.2,'  STREAM FCN=',F20.8)
110   CONTINUE
      RETURN
      END
```

APPENDIX C

COMPUTER PROGRAM FOR NUMERICAL SOLUTION

The program which calculates the dimensionless stream functions and vorticities for the two-dimensional incompressible spin up problem by computational works is listed below. The principal symbols used in the program and their definitions are as follows:

ARCLEG(K,L)	arc length from the 1st vertex to the Lth leg of Kth 3 point
ARCNOR(J)	arc length from the 1th vertex to the bases of the normals of the 3 points in computer-clock-wise order around
ARCNOR(K)	arc length from the 1st vertex to the Kth normal
ARCVER(I)	arc length from the 1st vertex to the Ith vertex
DCALL(L)	4 distances for calling SUBROUTINE SIGMA
DELTA	mesh space
DELTAT	time increment
DSIG(K,L)	distances to π right of, above, leftof, and below the Kth point near σ (boundary)
I	grid-point subscript
ICLASS(I,J)	classification of mesh points 1-out; 2-in; 3-on σ

IDISPLY(51)	mesh display matrix
IPOFR(J)	integer position of occurance in the mesh scan of the Jth 3-point going C.C.W. around π(position of ranks J)
IROFP(I)	ranks of the Ith element of ARCNOR
ISSIG(K,L)	side index to right of, above, to left of, and below the Kth point near σ according as whether L is respectively 1, 2, 3 and 4
J	grid-point subscript
K	boundary point index
L	mesh points leg index
NSIG	number of points near σ
NTS	number of time steps
NV	number of vertices
NX	number of columns in space mesh
NY	number of rows in space mesh
OMEGA(I,J)	mesh vorticity
OMNEW(I,J)	mesh vorticity storage place
OMWALL(I)	vortivity at base of normal of the Ith 3-point C.C.W. around π
PSI(I,J)	mesh stream function
R	Reynolds number
SCALL(L)	4 stream functions for calling SUBROUTINE SIGMA
SMALL	0.25 DELTA
SPI(I)	stream function on π
SRTRN	value of stream function returned by

SUBROUTINE SIFMA

SSIG(K,L)	stream function on right of, above, left of, and below Kth point
UPI(I)	velocity on π
VCALL	value of OMEGA(I,J) for calling SIGMA
VOR3L(L)	vorticity at the Lth leg of a 3-point either from an interior point or interpolated at a boundary
VORSUR(L)	surrounding vorticity at the Lth leg of a temporary assymetric spider
WL	mutual overlap λ
XPI(I)	vertex coordinates of π
YMAX	greatest y-coordinates reached by π
YMIN	least y-coordinates reached by π
YPI(I)	vertex coordinates of π

```
      DIMENSION DSIG(2000,4),ISSIG(2000,4),SSIG(2000,4),ARCLEG(2000,4)
      DIMENSION DCALL(4),ISCALL(4),VORSUR(4),SCALL(4)
      DIMENSION H3(2000),H4(2000),ARCNOR(2000),OMWALL(2000)
      DIMENSION ARCALL(2000),IPOFR(2000),IROFR(2000)
      DIMENSION XPI(200),YPI(200),UPI(200),SPI(200),ARCVER(200),RO(200)
      DIMENSION PSI(51,49),OMEGA(51,49),PSINEW(49)
      DIMENSION JSTART(600),JEND(600)
      COMMON/COM1/OMNEW(51,49)
      DIMENSION ISIG(2000),JSIG(2000)
      COSH(S)=0.5*(EXP(S)+EXP(-S))
      SINH(S)=0.5*(EXP(S)-EXP(-S))
    5 FORMAT(8I10)
   10 FORMAT(8F10.5)
   15 FORMAT(1H ,125I1)
   20 FORMAT(1H-)
   25 FORMAT(1H0)
   30 FORMAT(1H1)
   32 FORMAT(38H STREAM FUNCTION AFTER INITIALIZATION:)
   35 FORMAT(22H STREAM FUNCTION AFTER,15,12H TIME STEPS:)
   37 FORMAT(32H VORTICITY AFTER INITIALIZATION:)
   40 FORMAT(16H VORTICITY AFTER,15,12H TIME STEPS:)
   50 FORMAT(1H ,8I10)
   60 FORMAT(1H ,8F12.7)
   65 FORMAT(16H WALL VORTICITY:)
   70 FORMAT(26H BOUNDARY POINT VORTICITY:)
   75 FORMAT(30H PRINCIPAL VORTICITY DIAGONAL:)
   80 FORMAT(35H ANTI-PRINCIPAL VORTICITY DIAGONAL:)
      TIME1=SECOND(EARLY)
C     READ 5,NPI,NX,NY,NTS,NRELAX,INTDIS,IE
      READ (5,5)NPI,NX,NY,NTS,NRELAX,INTDIS,IE
C     READ 10,WL,R,YSTART,DELTAT,ORF
      READ (5,10)WL,R,YSTART,DELTAT,ORF
      PRINT 50,NPI,NX,NY,NTS,NRELAX,INTDIS,IE
      PRINT 60,WL,R,YSTART,DELTAT,ORF
C     READ 10,XPI(I),YPI(I),UPI(I),SPI(I),RO(I)
      DO 100 I=1,NPI
      XPI(I)=1.+COS(6.283185*(I-1)/NPI)
      YPI(I)=1.+SIN(6.283185*(I-1)/NPI)
      UPI(I)=0.
      IF(I.LE.IE)UPI(I)=1.
      SPI(I)=0.
      RO(I)=1.
  100 PRINT60,XPI(I),YPI(I),UPI(I),SPI(I),RO(I)
      CALL VARC(NPI,XPI,YPI,ARCVER)
      PRINT 60,(ARCVER(I),I=1,NPI)
      PERIM=ARCVER(I)
      DELTA=WL/(NX-1)
```

```
      SMALL=DELTA*.25
      H1=DELTAT/(DELTA*DELTA*R)
      H2=DELTAT/(DELTA*DELTA*4.)
      H5=DELTAT/R
      H6=DELTA*.5
      ORFTG=ORF*DELTA*DELTA/4.
      ORFD4=ORF/4.
      ORFM1=ORF-1.
      Y=YSTART
      DO 720 J=1,NY
      X=0.
      DO 710 I=1,NX
      CALL CLOSE(NPI,XPI,YPI,X,Y,IRTRN)
      IF(IRTRN.EQ.0)GO TO 700
      CALL NEAR(NPI,XPI,YPI,X,Y,SMALL,NYE)
      IF(NYE.NE.0)GO TO 700
      ICLASS(I,J)=2
      GO TO 710
  700 ICLASS(I,J)=1
  710 X=X+DELTA
  720 Y=Y-DELTA
      DO 730 I=1,NX
      IF(ICLASS(I,1).EQ.2)ICLASS(I,1)=3
      IF(ICLASS(I,NY).EQ.2)ICLASS(I,NY)=3
  730 CONTINUE
      NYM1=NY-1
      DO 740 J=2,NYM1
      DO 740 I=1,NX
      IL=I-1
      IF(IL.LT.1)IL=NX
      IR=I+1
      IF(IR.GT.NX)IR=1
      IF(ICLASS(I,J).EQ.2.AND.(ICLASS(IR,J).EQ.1.OR.ICLASS
     &(I,J-1).EQ.1
     +                    .OR.ICLASS(IL,J).EQ.1.OR.ICLASS
     &(I,J+1).EQ.1))
     +      ICLASS(I,J)=3
  740 CONTINUE
      DO 745 I=1,NX
      JSTART(I)=0
      DO 745 J=1,NY
      IF(ICLASS(I,J-1).NE.2.AND.ICLASS(I,J).EQ.2)JSTART(I)=J
      IF(ICLASS(I,J+1).NE.2.AND.ICLASS(I,J).EQ.2)JEND(I)=J
  745 CONTINUE
      DO 750 J=1,NY
      PRINT 15,(ICLASS(I,J),I=1,NX)
  750 CONTINUE
```

```
      GO TO 758
      PTINT 25
      JBELOW=NY-10
      DO 755 J=JBELOW,NY
      PRINT 15,(ICLASS(I,J),I=1,NX)
  755 CONTINUE
  758 K=0
      AA=DELTA/10000.
      BB=10000.*DELTA
      X=0.0
      DO 840 I=1,NX
      Y=YSTART
      DO 820 J=1,NY
      IF(ICLASS(I,J).NE.3)GO TO 810
      K=K+1
      ISIG(K)=I
      JSIG(K)=J
      CALL APROCH(DELTA,X,Y,NPI,YPI,L,DSQ,ARCN)
      H3(K)=2./DSQ
      H4(K)=SPI(L)+UPI(L)*(SORT(DSQ)+1./(RO(L)*H3(K)))
      IF(L.NE.1)ARCN+ARCVER(L)
      ARCNOR(K)=ARCN
      ARCALL(K)=ARCN
      CALL POINT(AA,BB,X,Y,NPI,XPI,YPI,DCALL,ISCALL,SCALL)
      CALL NEAR(NPI,XPI,YPI,X,Y,AA,NR)
      DO 760 L=1,4
      DSIG(K,L)=DCALL(L)
      ARCADD=0.
      IF(ISCALL(L).NE.1)ARCADD=ARCVER(ISCALL(L))
      ARCLEG(K,L)=SCALL(L)+ARCADD
      II=ISCALL(L)
      ISSIG(K,L)=II
      SSIG(K,L)=SPI(II)
  760 CONTINUE
      IF (NR.EQ.0)GO TO 770
      ISSIG(K,1)=-ISSIG(K,1)
      GO TO 810
  770 I1=I+1
      IF(I1.GT.NX)I1=1
      IF(ICLASS(I1,J).NE.1)ISSIG(K,1)=0
      IF(J.EQ.1)GO TO 790
      IF(ICLASS(I,J-1).NE.1)ISSIG(K,2)=0
  790 I3=I-1
      IF(I3.LT.1)I3=NX
      IF(ICLASS(I3,J).NE.1)ISSIG(K,3)=0
      IF(J.EQ.NY)GO TO 810
      IF(ICLASS(I,J+1).NE.1)ISSIG(K,4)=0
```

```
810 Y=Y-DELTA
820 CONTINUE
830 X=X+DELTA
840 CONTINUE
    NSIG=K
    CALL PRANK(NSIG,ARCALL,IPOFR,IROFP)
    PRINT 50,(ISIG(IPOER(K)),K=1,NSIG)
    PRINT 50,(JSIG(IPOFR(K)),K=1,NSIG)
    PRINT 50,(IPOFR(I),I=1,NSIG),(IROFP(I),I=1,NSIG)
    PRINT 50,NSIG
    DO 850 L=1,20
    PRINT 860,(DSIG(L,J),J=1,4),(ISSIG(L,J),J=1,4),(SSIG(L,J),
   &J=1,4)
    PRINT 60,H3(L),ARCNOR(L)
    PRINT 860,(ARCLEG(L,J),J=1,4)
850 CONTINUE
860 FORMAT(1H ,4F7.3,4I3,4F7.3)
    DO 853 K=1,NSIG
853 ARCALL(K)=ARCNOR(K)
    DO 857 K=1,NSIG
857 ARCNOR(K)=ARCALL(IPOFR(K))
    Y=YSTART
    CPSI=.033333333
    DO 880 J=1,NY
    BIGPSI=Y*(-1.+Y*(1.3333333-.44444444*Y))
    S=3.1622777*(1.-Y)
    HYPSNS=SINH(S)
    F=-.1468*HYPSNS-.03392*S*COSH(S)
    X=0.0
    DO 870 I=1,NX
    PSI(I,J)=0.0
    OMEGA(I,J)=0.0
    OMNEW(I,J)=0.0
    GO TO 870
    COSINE=-COS(3.1622777*X)
    XT=X+WL*.5
    YT=Y
    CALL RETRO(XT,YT)
    OMEGA(I,J)=2.6666667*(YT-1.)
    PSI(I,J)=BIGPSI+CPSI*F*COSINE
    OMNEW(I,J)=PSI(I,J)
870 X=X+DELTA
880 Y=Y-DELTA
    GO TO 1031
    DO 1030 IRLX=1,500
    DO 930 I=1,NX
    I1=1+MOD(I,NX)
```

```
      I3=NX-MOD(NX+1-I,NX)
      JS=JSTART(I)
      IF(JS.EQ.0)GO TO 930
      JE=JEND(I)
      JSBLAC=JS
      JSRED=JS+1
      JEBLAC=JE-MOD(JE-JS,2)
      JERED=JE-1+MOD(JE-JS,2)
      DO 910 J=JSBLAC,2
  910 PSINEW(J)=ORFD4*(PSI(I1,J)+PSI(I,J+1)+PSI(I3,J)+PSI(I,J-1))
     ++ORFIG*OMEGA(I,J)-ORFM1*PSI(I,J)
      DO 920 J=JSBLAC,JEBLAC,2
  920 PSI(I,J)=PSINEW(J)
      DO 923 J=JSRED,JERED,2
  923 PSINEW(J)=ORFD4*(PSI(I1,J)+PSI(I,J+1)+PSI(I3,J)+PSI(I,J-1))
     ++ORFTG*OMEGA(I,J)-ORFM1*PSI(I,J)
      DO 927 J=JSRED,JERED,2
  927 PSI(I,J)=PSINEW(J)
  930 CONTINUE
      DO 1020 K=1,NSIG
      I=ISIG(K)
      J=JSIG(K)
      I1=1+MOD(I,NX)
      I3=NX-MOD(NX+1-I,NX)
      IF(ISSIG(K,1).GE.0)GO TO 950
      PSI(I,J)=SSIG(K,1)
      GO TO 1020
  950 DO 1010 L=1,4
      IF(ISSIG(K,L).EQ.0)GO TO 960
      DCALL(L)=DSIG(K,L)
      SCALL(L)=SSIG(K,L)
      GO TO 1010
  960 DCALL(L)=DELTA
      GO TO(970,980,990,1000),L
  970 SCALL(L)=PSI(I1,J)
      GO TO 1010
  980 SCALL(L)=PSI(I,J-1)
      GO TO 1010
  990 SCALL(L)=PSI(I3,J)
      GO TO 1010
 1000 SCALL(L)=PSI(I,J+1)
 1010 CONTINUE
      VCALL=OMEGA(I,J)
      CALL SIGMA(DCALL,SCALL,VCALL,SRTRN)
      PSI(I,J)=SRTRN
 1020 CONTINUE
 1030 CONTINUE
```

```
1031 CONTINUE
     DO 1035 J=1,NY
     DO 1032 I=1,NX
     OMNEW(I,J)=OMEGA(I,J)
1032 CONTINUE
1035 CONTINUE
     PRINT 30
     PRINT 37
     CALL DISCRA(NX,NY,2,1,1,.001)
     PRINT 30
     PRINT 32
     DO 1440 J=1,NY
     DO 1445 I=1,NX
     OMNEW(I,J)=PSI(I,J)
1445 CONTINUE
1440 CONTINUE
     CALL DISCRA(NX,NY,2,1,1,.001)
     TIME2=SECOND(LATER)
     ELAPSE=TIME2-TIME1
     PRINT 1447,ELAPSE
1447 FORMAT(40H TIME FOR INPUT,MESH,AND INITIALIZATION=,E10.3)
     DO 3000 ITS=1,NTS
     TIME1=SECOND(EARLY)
     DO 1532 I=1,NX
     I1=1+MOD(I,NX)
     I3=NX-MOD(NX+1-I,NX)
     JS=JSTART(I)
     IF(JS.EQ.0)GO TO 1532
     JE=JEND(I)
     DO 1530 J=JS,JE
     J2=J-1
     J4=J+1
     U2D=PSI(I,J2)-PSI(I,J4)
     V2D=PSI(I3,J)-PSI(I1,J)
     OMEGA0=OMEGA(I,J)
     OMEGA1=OMEGA(I1,J)
     OMEGA2=OMEGA(I.J2)
     OMEGA3=OMEGA(I3,J)
     OMEGA4=OMEGA(I,J4)
     OMXD1=OMEGA1-OMEGA0
     OMXD3=OMEGA0-OMEGA3
     OMTD2=OMEGA2-OMEGA0
     OMYD4=OMEFA0-OMEGA4
     U4DP=U2D+ABS(U2D)
     U4DM=U2D-ABS(U2D)
     V4DP=V2D+ABS(V2D)
     V4DM=V2D-ABS(V2D)
     OMNEW(I,J)=OMEGA0
```

```
    ++H1*(OMXD1+OMYD2-OMXD3-OMYD4)
    +-H2*(U4DP*OMXD3+U4DM*OMXD1+V4DP*OMYD4+V4DM*OMYD2)
1530 CONTINUE
1532 CONTINUE
     DO 1535 K=1,NSIG
     I=ISIG(K)
     J=JSIG
     OMWALL(IROFP(K))=H3(K)*(H4(K)-PSI(I,J))
1535 CONTINUE
     DO 1780 K=1,NSIG
     I=ISIG(K)
     J=JSIG(K)
     I1=1+MOD(I,NX)
     I3=NX-MOD(NX+1-I,NX)
     IF(ISSIG(K,1).GE.0)GO TO 1550
     OMNEW(I,J)=OMWALL(IROFP(K))
     GO TO 1780
1550 DO 1610 L=1,4
     IF(ISSIG(K,L).EQ.0)GO TO 1560
     DCALL(L)=DSIG(K,L)
     SLEG=ARCLEG(K,L)
     CALL PERINT(NSIG,PERIM,ARCNOR,OMWALL,SLEG,VLEG)
     VORSUR(L)=VLEG
     SCALL(L)=SSIG(K,L)
     GO TO 1610
1560 DCALL(L)=DELTA
     GO TO(1570,1580,1590,1600),L
1570 VORSUR(L)=OMEGA(I1,J)
     SCALL(L)=PSI(I1,J)
     GO TO 1610
1580 VORSUR(L)=OMEGA(I,J-1)
     SCALL(L)=PSI(I,J-1)
     GO TO 1610
1590 VORSUR(L)=OMEGA(I3,J)
     SCALL(L)=PSI(I3,J)
     GO TO 1610
1600 VORSUR(L)=OMEGA(I,J+1)
     SCALL(L)=PSI(I,J+1)
1610 CONTINUE
     VCENTER=OMEGA(I,J)
     CALL ASLAP(VORSUR,DCALL,VCENTER,ATLED)
     D1PD3=DCALL(1)+DCALL(3)
     D2PD4=DCALL(2)+DCALL(4)
     USIG=(SCALL(2)-SCALL(4)/D2PD4
     VSIG=(SCALL(3)-SCALL(1)/D1PD3
     OMXR=(VORSUR(1)-VCENTER/DCALL(1)
     OMXL=(VCENTER-VORSUR(3))/DCALL(3)
```

```
      OMYA=(VORSUR(2)-VCENTER)/DCALL(2)
      OMYB=(VCENTER-VORSUR(4))/DCALL(4)
      U2SIGP=USIG+ABS(USIG)
      U2SIGM=USIG-ABS(USIG)
      V2SIGP=VSIG+ABS(VSIG)
      V2SIGM=VSIG-ABS(VSIG)
      OMNEW(I,J)=VCENTER+H5*ATLED
     +           -H6*(U2SIGP*OMXL+U2SIGM*OMXR+V2SIGP*OMYB+V2SIGM*
     &OMYA)
 1780 CONTINUE
      DO 1792 J=1,NY
      DO 1790 I=1,NX
      OMEGA(I,J)=OMNEW(I,J)*ORFTG
 1790 CONTINUE
 1792 CONTINUE
      TIME2=SECOND(LATER)
      ELAPSE=TIME2-TIME1
      IF(ITS.NE.NTS)GO TO 1800
      PRINT 1795,ELAPSE
 1795 FORMAT(37H TIME FOR SOLVING VORTICITY EQUATION=,E10.3)
 1800 CONTINUE
      TIME1=SECOND(EARLY)
      DO 2020 IRELAX=1,NRELAX
      DO 1930 I=1,NX
      I1=1+MOD(I,NX)
      I3=NX-MOD(NX+1-I,NX)
      JS=JSTART(I)
      IF(JS.EQ.0)GO TO 1930
      JE=JEND(I)
      JSBLAC=JS
      JSRED=JS+1
      JEBLAC=JE-MOD(JE-JS,2)
      JERED=JE-1+MOD(JE-JS,2)
      DO 1910 J=JSBLAC,JEBLAC,2
 1910 PSINEW(J)=ORFD4*(PSI(I1,J)+PSI(I,J+1)+PSI(I3,J)+PSI(I,J-1))
     ++OMEGA(I,J)-ORFM1*PSI(I,J)
      DO 1920 J=JSBLAC,JEBLAC,2
 1920 PSI(I,J)=PSINEW(J)
      DO 1923 J=JSRED,JERED,2
 1923 PSINEW(J)=ORFD4*(PSI(I1,J)+PSI(I,J+1)+PSI(I3,J)+PSI(I,J-1))
     ++OMEGA(I,J)-ORFM1*PSI(I,J)
      DO 1927 J=JSRED,JERED,2
 1927 PSI(I,J)=PSINEW(J)
 1930 CONTINUE
      DO 2020 K=1,NSIG
      I=ISIG(K)
      J=JSIG(K)
```

```
      I1=1+MOD(I,NX)
      I3=NX-MOD(NX+1-I,NX)
      IF(ISSIG(K,1).GE.0)GO TO 1950
      PSI(I,J)=SSIG(K,1)
      GO TO 2020
 1950 DO 2010 L=1,4
      IF(ISSIG(K,L).EQ.0)GO TO 1960
      DCALL(L)=DSIG(K,L)
      SCALL(L)=SSIG(K,L)
      GO TO 2010
 1960 DCALL(L)=DELTA
      GO TO(1970,1980,1990,2000),L
 1970 SCALL(L)=PSI(I1,J)
      GO TO 2010
 1980 SCALL(L)=PSI(I,J-1)
      GO TO 2010
 1990 SCALL(L)=PSI(I3,J)
      GO TO 2010
 2000 SCALL(L)=PSI(I,J+1)
 2010 CONTINUE
      VCALL=OMNEW(I,J)
      CALL SIGMA(DCALL,SCALL,VCALL,SRTRN)
      PSI(I,J)=SRTRN
 2020 CONTINUE
      TIME2=SECOND(LATER)
      ELAPSE=TIME2-TIME1
      IF(ITS.NE.NTS)GO TO 2027
      PRINT 2025,NRELAX,ELAPSE
 2025 FORMAT(18H TIME REQUIRED FOR,I4,13H RELAXATIONS=,E10.3)
 2027 DO 2030 J=1,NY
      DO 2035 I=,NX
      OMEGA(I,J)=OMNEW(I,J)
 2035 CONTINUE
 2030 CONTINUE
      IMPRTR=0
      RINT=(ITS+.25)/INTDIS
      IINT=RINT
      DIFRI=RINT-IINT
      HALFOI=.5/INTDIS
      IF(DIFRI.LT.HALFOI)IMPRTR=1
      IF(IMPRTR.NE.1)GO TO 3000
      PRINT 30
      PRINT 40,ITS
      CALL DISCRA(NX,NY,2,1,1,.001)
      PRINT 30
      PRINT 35,ITS
      DO 2040 J=1,NY
```

```
      DO 2045 I=1,NX
      OMNEW(I,J)=PSI(I,J)
 2045 CONTINUE
 2040 CONTINUE
      CALL DISCRA(NX,NY,2,1,1,.001)
      PRINT 65
      PRINT 60,(OMWALL(K),K=1,NSIG)
      PRINT 70
      PRINT 60,(OMEGA(ISIG(IPOFR(K)),JSIG(IPOFR(K)),K=1,NSIG)
      PRINT 75
      PRINT 60,(OMEGA(I,I-1),I=2,50)
      PRINT 80
      PRINT 60,OMEGA(I,51-I),I=2,50)
 3000 CONTINUE
      STOP
      END
      SUBROUTINE POINT(R,B,XP,YP,NV,XV,YV,D,IS,S)
      DIMENSION XV(200),YV(200),X(200),Y(200),D(4),IS(4),S(4)
      DO 40 L=1,4
      D(L)=B
      IS(L)=1
      S(L)=0
      DO 40 I=1,NV
      J=MOD(I,NV)+1
      XI=XP-XV(I)
      XJ=XP-XV(J)
      YI=YP-YV(I)
      YJ=YP-YV(J)
      DO 10 K=1,L
      AI=XI
      XI=YI
      AJ=XJ
      XJ=YJ
      YI=-AI
   10 YJ=-AJ
      IF(ABS(XI).LT.R.OR.ABS(XJ).LT.R)GO TO 20
      IF(XI*XJ.GT.0)GO TO 40
      YINT=YI-(YJ-YI)*XI/(XJ-XI)
      IF(YINT.LE.0..OR.YINT.GE.D(L))GO TO 40
      D(L)=YINT
      S(L)=SORT(XI*XI+(YI-YINT)*(YI-YINT))
      GO TO 30
   20 IF(YI.LE.0..OR.YI.GE.D(L).OR.ABS(XI).GE.R)GO TO 40
      D(L)=YI
      S(L)=0
   30 IS(L)=I
   40 CONTINUE
```

```
      RETURN
      END
      SUBROUTINE APROCH(DELTA,XP,YP,N,X,Y,IS,DSQ,S)
      DIMENSION X(200),Y(200)
      DSMIN=DELTA+DELTA
      DSQMIN=DSMIN*DSMIN
      DO 100 I=1,N
      J=I+1
      IF(J.GT.N)J=1
      DXIJ=X(J)-X(I)
      DYIJ=Y(J)-Y(I)
      DXIP=XP-X(I)
      DYIP=YP-Y(I)
      DOT=DXIJ*DXIP+DYIJ*DYIP
      IF(DOT.LT.0.)GO TO 50
      SLSQ=DXIJ*DXIJ+DYIJ*DYIJ
      IF(DOT.GT.SLSQ)GO TO 100
      CROSS=DXIJ*DYIP-DXIP*DYIJ
      DSQ=CROSS*CROSS/SLSQ
      IF(DSQ.GE.DSOMIN)GO TO 100
      DSQMIN=DSQ
      SSQ=DOT*DOT/SLSQ
      IS=I
      GO TO 100
   50 DSQ=DXIP*DXIP+DYIP*DYIP
      IF(DSQ.GE.DSQMIN)GO TO 100
      DSQMIN=DSQ
      SSQ=0.0
      IS=I
  100 CONTINUE
      DSQ=DSQMIN
      S=SQRT(SSQ)
      RETURN
      END
      SUBROUTINE CLOSE(N,XV,YV,XE,YE,I)
      DIMENSION XV(200),YV(200),ANGEL(200)
      DO 120 L=1,N
      X=XV(L)-XE
      Y=YV(L)-YE
      A=ABS(X)+ABS(Y)
      IF(A.LE.1.E-30)GO TO 140
      ANGLE(L)=1.-X/A
      IF(Y.LT.0.)ANGLE(L)=4.-ANGLE(L)
  120 CONTINUE
      SUM=0.
      DO 130 L=1,N
      LNEXT=L+1
```

```
      IF(L.EQ.N)LNEXT=1
      CHANGE=ANGLE(LNEXT)-ANGLE(L)
      SINGE=1.
      IF(CHANGE.LE.0.)SINGE=-1.
      SUBTAN=CHANGE*SINGE
      IF(SUBTAN.GT.2)CHANGE=-(4.-SUBTAN)*SINGE
      SUM=SUM+CHANGE
  130 CONTINUE
      I=0
      IF(ABS(SUM).GT.2.)I=1
      RETURN
  140 I=0
      RETURN
      END
      SUBROUTINE NEAR(NV,XV,YV,XP,YP,A,NYE)
      DIMENSION XV(200),YV(200)
      NYE=0
      DO 100 I=1,NV
      XIP=XP-XV(I)
      YIP=YP-YV(I)
      DIP=ABS(XIP)+ABS(YIP)
      IF(DIP.GT.A)GO TO 10
      NYE=I
      RETURN
   10 J=I+1
      IF(J.GT.NV)J=1
      XIJ=XV(J)-XV(I)
      YIJ=YV(J)-YV(I)
      RIPOIJ=XIP*XIJ+YIP*YIJ
      IF(RIPOIJ)100,100,20
   20 XJP=YP-YV(J)
      RJIOJP=-XIJ*XJP-YIJ*YJP
      IF(RJIOJP)100,100,30
   30 DLJ=ABS(XIJ)+ABS(YIJ)
      V=ABS(XIP*YIJ-XIJ*YIP)/DIJ
      IF(V.GT.A)GO TO 100
      NYE=I
      RETURN
  100 CONTINUE
      RETURN
      END
      SUBROUTINE DISCRA(M,N,KBILD,KTAB,KREICH,SMALL)
      DIMENSION IDSPLY(51)
      COMMON/COM1/A(51,49),ICLASS(51,49)
      DATA KD1/' '/,KD2/'8'/,KD3/'+'/
    5 FORMAT(7H COLUMN,I4,2H :)
    7 FORMAT(4H ROW,I4,2H :)
```

```
10 FORMAT(1H)
20 FORMAT(1H ,7E10.3)
25 FORMAT(50H PICTURE NOT SHOWN BECAUSE FUNCTION REMAINS
  &WITHIN,E10.2
  +,25H OF ITS' MINIMUM VALUE OF,E10.2)
30 FORMAT(1H ,80A1)
35 FORMAT(1H1)
   IF(KTAB.EQ.0)GO TO 70
   I=26
   PRINT 10
   PRINT 5,1
   PRINT 20,(A(I,J),J=1,N)
40 CONTINUE
   J=25
   PRINT 10
   PRINT 7,J
   PRINT 20,(A(I,J),I=1,M)
50 CONTINUE
   PRINT 10
70 IF(KBILD.EQ.0.AND.KREICH.EQ.0)GO TO 140
   PRINT 35
   AMAX=-100000.
   AMIN=100000.
   DO 80 J=1,N
   DO 80 I=1,M
   IF(ICLASS(I,J).EQ.1)GO TO 80
   AIJ=A(I,J)
   IF(AIJ.LE.AMAX)GO TO 75
   AMAX=AIJ
   IMAX=I
   JMAX=J
75 IF(AIJ.GE.AMIN)GO TO 80
   AMIN=AIJ
   IMIN=I
   JMIN=J
80 CONTINUE
   IF(KREICH.EQ.0)GO TO 90
   PRINT 83,AMIN,IMIN,JMIN,AMAX,IMAX,JMAX
83 FORMAT(5H MIN=,E10.3,2HAT,2I5,4HMAX=,E10.3,2HAT,2I5)
90 IF(KBILD.EQ.0)GO TO 140
   RANGE=AMAX-AMIN
   IF(RANGE.GT.SMALL)GO TO 95
   PRINT 25,SMALL,AMIN
   RETURN
95 DO 130 J=1,N
   DO 120 I=1,M
   STRIPE=KBILD+KBILD
```

```
      L=STRIPE*(A(I,J)-AMIN)/RANGE+1
      IF(L.EQ.2.OR.L.EQ.6.OR.L.EQ.8.OR.L.EQ.10)GO TO 100
      IDSPLY(I)=KD2
  100 IF(J.EQ.1.OR.J.EQ.N.OR.I.EQ.1.OR.I.EQ.M)GO TO 110
      IDSPLAY(I)=KD1
      GO TO 120
  110 IDSPLY(I)=KD3
  120 CONTINUE
      PRINT 30,(IDSPLY(I),I=1,M)
  130 CONTINUE
  140 RETURN
      END
      SUBROUTINE SIGMA(D,S,VOR,PSI)
      DIMENSION D(4),S(4)
      D1D3=D(1)*D(3)
      D2D4=D(2)*D(4)
      PSI=(D1D3*(D(2)*S(4)*S(2)+D(4))+D2D4*(D(3)*S(1)+D(1)*S
     C(3)/(D(1)+D(3)+VOR*D1D3*D2D4/2.)/(D1D3+D2D4)
      RETURN
      END
      SUBROUTINE RETRO(X,Y)
      DIMENSION A(4)
      DATA A(1),A(2),A(3),A(4)/2.69324,-.018087,.0001809,
     &-.000001855/
      DO 10 ITS=1,500
      CALL SPEED(4,A,3.16227701,X,Y,U,V)
      X=X-U*.000002
      Y=Y-V*.000002
   10 CONTINUE
      RETURN
      END
      SUBROUTINE SPEED(N,A,ALPHA,X,Y,U,V)
      DIMENSION A(4)
      U=0.0
      V=0.0
      DO 100 I=1,N
      U=U-A(I)*ALPHA*COSH(I*ALPHA*(1.-Y))*SIN(I*ALPHA*X)
      V=V-A(I)*ALPHA*SINH(I*ALPHA*(1.-Y))*COS(I*ALPHA*X)
  100 CONTINUE
      RETURN
      END
      SUBROUTINE PRANK(N,S,K,L)
      DIMENSION K(2000),S(2000),L(2000)
      DO 100 I=1,N
      K(I)=1
      SMIN=S(1)
      DO 50 J=2,N
```

```
      IF(S(J).GE.SMIN)GO TO 50
      K(I)=J
      SMIN=S(J)
   50 CONTINUE
      INDIR=K(I)
      L(INDIR)=I
      S(INDIR)=1000000000.
  100 CONTINUE
      RETURN
      END
      SUBROUTINE VARC(N,X,Y,S)
      DIMENSION X(200),Y(200),S(200)
      SUM=0.
      DO 100 I=1,n
      J=MOD(I,N)+1
      DXLJ=X(J)-X(I)
      DYIJ=Y(J)-Y(I)
      SIDEL=SORT(DXIJ*DXIJ+DYIJ*DYIJ)
      SUM=SUM+SIDEL
      S(J)=SUM
  100 CONTINUE
      RETUTN
      END
      SUBROUTINE PERINT(N,P,X,Y,YIN,YOUT)
      DIMENSION X(2000),Y(2000)
      IF(XIN.GE.X(1))GO TO 10
      XI=X(N)-P
      YI=Y(N)
      XJ=X(1)
      YJ=Y(1)
      GO TO 55
   10 IF(XIN.LT.X(N))GO TO 20
      XI=X(N)
      YI=Y(N)
      XJ=P+X(1)
      YJ=Y(1)
      GO TO 55
   20 DO 45 JJ=2,N
      XJ=X(JJ)
      IF(XIN.GE.XJ)GO TO 45
      I=JJ-1
      XI=X(I)
      YI=Y(I)
      YJ=Y(JJ)
      GO TO 55
   45 CONTINUE
   55 YOUT=YI+(YJ-YI)*(XIN-XI)/(XJ-XI)
```

```
RETURN
END
SUBROUTINE ASLAP(SUR,D,CENTER,ATLED)
DIMENSION SUR(4),D(4)
D1PD3=D(1)+D(3)
D2PD4=D(2)+D(4)
R1=2./(D(1)*D1PD3)
R2=2./(D(2)*D2PD4)
R3=2./(D(3)*D1PD3)
R4=2./(D(4)*D2PD4)
SUMR=R1+R2+R3+R4
ATLED=R1*SUR(1)+R2*SUR(2)+R3*SUR(3)+R4*SUR(4)-SUMR*CENTER
RETURN
END
```