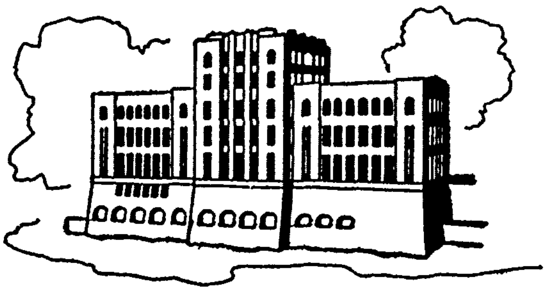
FINITE ANALYTIC NUMERICAL SOLUTIONS OF INCOMPRESSIBLE FLOW PAST INCLINED AXISYMMETRIC BODIES

by

Ching Jen Chen and Wu Sun Cheng



IIHR Report No. 308

Department of Mechanical Engineering and
Iowa Institute of Hydraulic Research
The University of Iowa
Iowa City, Iowa 52242-1585

April 1987

The research was partially supported by the Naval Sea Systems Command GHR Grant N000168-86-J-0019, administered by DTNSRDC

FINITE ANALYTIC NUMERICAL SOLUTIONS OF INCOMPRESSIBLE FLOW PAST INCLINED AXISYMMETRIC BODIES

by

Ching Jen Chen and Wu Sun Cheng

IIHR Report No. 308

Department of Mechanical Engineering and
Iowa Institute of Hydraulic Research
The University of Iowa
Iowa City, Iowa 52242-1585

April 1987

The research was partially supported by the Naval Sea Systems Command GHR Grant N00016S-S6-J-0019, administered by DTNSRDC

FINITE ANALYTIC NUMERICAL SOLUTIONS
OF
INCOMPRESSIBLE FLOW PAST INCLINED AXISYMMETRIC BODIES

by

Ching Jen Chen and Wu Sun Cheng

IIHR Report No. 308

Department of Mechanical Engineering and
Iowa Institute of Hydraulic Research
The University of Iowa
Iowa City, Iowa 52242-1585

April 1987

The research was partially supported by the Naval Sea Systems
Command GHR Grant N00168-86-J-0019, administered by DTNSRDC

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER IIHR Report No. 308	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Finite Analytic Numerical Solutions of Incompressible Flow Past Inclined Axisymmetric Bodies		5. TYPE OF REPORT & PERIOD COVERED Technical Report April 1987
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Ching-Jen Chen and Wu-Sun Cheng		8. CONTRACT OR GRANT NUMBER(s) N000167-86-K-0019
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Mechanical Engineering Iowa Institute of Hydraulic Research The University of Iowa, Iowa City, IA 52242		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS DTNSRDC Bethesda Maryland 20084		12. REPORT DATE April 1987
		13. NUMBER OF PAGES 349
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) DTNSRDC Bethesda Maryland . 20084		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

unlimited distribution

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

18. SUPPLEMENTARY NOTES

This research was partially supported by the Naval Sec. Systems Command GHR Program, administrated by the David W. Taylor Naval Ship R & D Center. Code 1505, Bethesda, Maryland 20084

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

ship model, viscous flow, turbulence model, computational fluid dynamics, boundary layer, separation, wakes

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

see attached *

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT

A finite analytic solution for three dimensional unsteady laminar and turbulent flow is derived on a curvilinear body-fitted coordinate system so that the flow past an arbitrary body shape can be predicted and solved. The general governing equations for turbulent flows are incompressible three-dimensional, ensemble-averaged Navier-Stokes equations. The Reynolds stresses are modeled by the k-ϵ turbulence model with Boussinesq eddy viscosity assumption.

In the numerical solution the velocity components and pressure are considered as primitive dependent variables and solved explicitly. A numerical program called FANS-3DEF (Finite Analytic Numerical Solution of Three Dimensional External Flow) is developed. In the FANS-3DEF program options are made available for users to select. They are (1) dimension, (2) grid system, (3) type of flow, and (4) turbulence models.

To verify the numerical accuracy and validity of the turbulence models, the finite analytic solution is first obtained for laminar and turbulent flow over a finite flat plate with or without angles of attack at Reynolds number 10^4, 10^5 and 2.48×10^6. Then finite analytic solutions for two axisymmetric bodies without an angle of attack at Reynolds number of 1.2×10^6 and 6.6×10^6 are obtained and compared with available experimental data. Good agreement between the predicted result and experimental data is obtained. Finally, the flow past an axisymmetric body with an ogival nose for three different angles of attack, 5, 10 and 15 degree at Reynolds number 3.7×10^6 is solved. Whenever possible the predicted solution are compared with either available numerical results or experimental data.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Page

Page

Page

LIST OF SYMBOLS

Alphabetical Symbols

A, B	constant coefficients in the linearized convective-transport equation
$a^{\phi}, b^{\phi}, \ldots$	constant coefficients in the transport equation for ϕ $(= u, v, w, k, \varepsilon)$
$a_U, a_D, \ldots$	finite-analytic coefficients for pressure and pressure-correction equations
C_1, C_2, C_3	constants in 11-point FA algebraic solution
$C_{\xi}, C_{\eta}, C_{\zeta}$	mean pressure-velocity linkage coefficients
C_f	$2\tau_{\omega}/\rho U_{\circ}^2$, friction coefficient in turbulent flow
C_{τ}	$\sqrt{Re}\,\tau_{\omega}/\rho U_{\circ}^2$, friction coefficient in laminar flow
C_{nb}	finite analytic coefficients for transport equations (nb=P, EC, NE, etc.)

C_p	pressure coefficient, or P normalized by $\rho U_o^2/2$
$C_m, C_{\varepsilon 1}, C_{\varepsilon 2}$	turbulence-model constants
D_U, D_V, D_W	FA pressure-velocity linkage coefficients
DS, DS'	source function in pressure and pressure-correction equations
E	integration constant, =9.
F1, F2, F3	grid control function in body-fitted coordinate system
h	grid size in finite-analytic local element
J, M	Jacobian
k	(1)dimensional turbulent kinetic energy (2)grid size in FA local element
k*	dimensionless turbulent kinetic-energy, normalized by U_o^2
L	length scale (plate or body length)
P	dimensional pressure
p	dimensionless pressure, or P normalized by ρU_o^2
p	guessed (imperfect) pressure field
p'	pressure-correction

q	magnitude of total velocity vector, normalized by U_0
Re	Reynolds number
S	source functions for transport quantities $(u, v, w, k, \varepsilon)$
T	dimensional time
t	(1)dimensionless time, or T normalized by L/U_0 (2)characteristic turbulent time scale
U, V, W	velocity components in Cartesian coordinates
u, v, w	dimensionless velocity components, normalized by U_0
u^*, v^*, w^*	velocities obtained from guessed (imperfect) pressure field p
$\hat{u}, \hat{v}, \hat{w}$	pseudovelocities, or velocities solution without pressure contribution
u', v', w'	velocity-corrections
U_0	constant free-stream (reference) velocity
U_1, U_2	resultant velocities parallel to the wall at first two nodal points

U_τ	$(\tau_\omega/\rho U_\infty^2)^{0.5}$, normalized friction (wall shear) velocity
$\overline{u_i u_j}$	dimensional Reynolds stress
$\overline{u'_i u'_j}$	dimensionless Reynolds stresses, normalized by U_∞^2
$\vec{V},\vec{V},\vec{V}$	velocities in the body-fitted coordinates
$\hat{V},\hat{V},\hat{V}$	pseudovelocities in the body-fitted coordinates
X,Y,Z	dimensionless Cartesian coordinates
x,y,z	dimensionless Cartesian coordinates, Or X, Y and Z normalized by L
Y_1,Y_2	distance normal to the wall at first two points
y^+	dimensionless distance measured in the direction normal to the wall

Greek Symbols

α	angle of attack
ε	dimensional rate of turbulent energy dissipation
ε^*	dimensionless rate of turbulent energy

	dissipation, normalized by U_0/L
κ	von Karman constant, 0.42
ν	kinematic viscosity
ν_t	turbulent eddy-viscosity,
ξ,η,ζ	body-fitted coordinates
η^*,ζ^*	transformed coordinates in linearized convective-transport equations
ρ	density
$\sigma_k,\sigma_\varepsilon$	turbulent Prandtl constants, for k and ε equations
τ_w	wall shear stress
ϕ	(1)transport quantities (u,v,w,k,ε) (2)mean quantity
ϕ'	fluctuating part
ϕ^*	instaneous quantity $\phi^* = \phi + \phi'$
∇	gradient
∇^2	laplacian
Δ	difference

Subscripts

EC,NE	east-central, north-east (similarly for WC,SC, NW,..etc.)
d,u,e w,n,s	downstream, upstream, east, west, north and south control surfaces (similarly for D,U,E,W,N,S)
nb	neighboring nodal points
P	interior node for numerical cells
ξ,η,ζ	first derivatives with respect to ξ,η,ζ
$\xi\xi,\eta\eta,\zeta\zeta$	second derivatives with respect to $\xi\xi,\eta\eta,\zeta\zeta$
x,y,z,t	first derivatives with respect to x,y,z,t
xx,yy,zz	second derivatives with respect to x,y,z

Superscripts

n,n-1	nth and (n-1)th time step
ϕ	transport quantities (u,v,w,k,ε)
i,i-1	ith and (i-1)th iteration

CHAPTER 1

INTRODUCTION

1.1 Motivation Of Research

Many fluid dynamic problems associated with flows over airplane, missile, ship, submarine and ground vehicle are three dimensional and turbulent. Because of the practical importance of these external flows in designing moving vehicles in the air, on ground and in the sea, the prediction of laminar and turbulent flow around a body has attracted considerably interest. To predict these flows is not a simple task since the flow past a simple body geometry can easily become three dimensional and turbulent if the approaching flow has an angle of attack to the body and when the Reynolds number is large. What is more is that the flow around the body may develop shedding if separation of flow is developed on the body. It is the aim of this study to develop a numerical prediction method for predicting complex laminar and turbulent flow past a two dimensional and axisymmetric body with or without angle of attack.

Although many numerical algorithms have been developed over the past decade to solve the three dimensional turbulent flow, most of these works are developed for external flow problems with governing equations of boundary layer flow type or for internal flow problems with governing equations of partially parabolized type. These works can not be implimented in prediction of the total flow field or flow with separation. This is due partly to the limitation of computer storage and computational time and partly to the lack of general turbulence model for prediction of complex flows involving flow separation and recirculation.

In the present study only external flows are considered. The prediction of external flow problems are indeed difficult and few solutions are available. However, many external flow problems are of great importance. In order to predict the complete flow past a body involving separation the complete Navier-Stokes equations are used in this study. For the case of turbulent flow the ensemble averaged process is used to obtain the averaged Navier-Stokes equations and the turbulence model based on second order correlation is adapted. As the prediction of complex three dimensional flows past three dimensional bodies is a formidable task. In the present study the prediction of flow past a finite flat plate from the upstream to the wake region is first made and then the flow past the finite length of axisymmetric body is predicted. Although the geometry of an axisymmetric body is simple in comparision with those practical configurations, the flow over the axisymmetric body at incidence is a complex three dimensional flow and contains most of the flow features observed on more complex geometries. Therefore, the prediction of flow past an inclined axisymmetric body is the first step in develcping numerical prediction capability for flow past more complicated geometry. The primary objective of the present study is then to develop a numerical scheme with some available turbulence models for prediction of flows past an axisymmetric body of finite length.

1.2 Previous Works

Before the detail of the present study is given, a brief review of the previous works on experimental study, numerical solutions for three dimensional turbulent flow are first made.

1.2.1 Experimental Study

Prandtl, the father of boundary layer theory, was the first to recognize the importance of three dimensionality in turbulent flow and had proposed a simple turbulent flow profile model [1] at the beginning of the 20th century. But Gruschwitz [2] (1935) was the first to conduct and publish the results of a comprehensive experiment involving three dimensional turbulent flows. He measured the free stream and the boundary layer mean flow field over many stations covering the flat end-wall of a curved two dimensional duct.

Since then the experimental studies in three dimensional turbulent flow grew. Unfortunately no turbulent stress data for three dimensional flows had been measured and published before 1967. Bradshaw and Terell [3] (1969) measured the turbulent stress on an 'infinite' sweep wing at Reynolds number around 6×10^4 which is believed to be the first detail study of turbulent flow in three dimensional boundary layer. They used this experiment to test Bradshaw's assertion that the ratio of turbulent stress to the turbulent kinetic energy is constant in the boundary layer. They found that the assertion is only approximately true.

Three dimensional turbulent experiments are painstaking and time consuming and definitely not abundant. Some experiments are conducted for greater understanding of the turbulent phenomena and can be used to develop suitable mathematical models for turbulent flow prediction. For an experiment to be useful in developing and testing the mathematical models the experimental data should provide adequate information for possible numerical simulation. In other words, in addition to the measurements of velocity, pressure and turbulent stress in the flow region, initial and boundary conditions must be carefully measured and documented. Since in this study the emphasis is placed on the turbulent flow past axisymmetric body that a brief review of experimental work pertaining to the flow past axisymmetric body with or without angle of attack is given.

Richmond [4] (1957) was probably the first to study the turbulent flow on a circular cylinder. He measured the velocity profile along the surface of a slender circular cylinder at several subsonic and hypersonic speeds. He obtained the law of the wall for the axisymmetric boundary layer by using Cole's streamline hypothesis. Later Yasuhara [5] (1959) measured a 20 mm diameter brass pipe, 1750 mm long with ogive-nose at Reynolds $1.2 \sim 1.8 \times 10^6$. Willmarth and Young [6] (1970) measured the boundary layer development for air flowing on a steel tube of 40 ft long and 3 in. diameter at 200 ft/sec free stream speed. In these experiments, although the velocity profile and pressure along the cylinder were measured, no turbulent quantities were measured.

Other experimental studies on flow over an axisymmetric body without angle of attack with measured turbulent quantities are shown in the table 1. They are Chevray [7] (1967), Patel, Nakayama and Damian [8] (1974), Patel and Lee [9] (1977), Huang, Santelli and Belt [10] (1978), and Hung, Groves and Belt [11] (1980). Chevray's experiment was the first attempt to measure turbulence stress in the wake. In his experiment a small separation was observed ahead of the tail. The data provided detailed information far into the wake. This experiment was recommended as a test case at the 1980-81 Stanford Conference on Complex Turbulent Flows but,

Authors (Ref.)	Body Shape	Length (m)	Max. (D/L)	Re = UL/ν (x 10^6)	Range of Data X/L	Data
Chevray (7)	Shperoid	1.524	0.167	2.75	0.958 - 4.0	P, U_x, V_r, $\overline{u^2}$, $\overline{v^2}$, $\overline{w^2}$, $\overline{uv}$
Patel (8) Nakayama Damian	Modified Spheroid	1.578	0.161	1.26	0.662 - 0.99	P, U_s, V_n, $\overline{u^2}$, $\overline{v^2}$, $\overline{w^2}$, $\overline{uv}$
Patel (9) Lee	Low Drag	1.22	0.234	1.20	0.60 - 2.47	P, U_s, V_n, $\overline{u^2}$, $\overline{v^2}$, $\overline{w^2}$, $\overline{uv}$
Huang (10) Santoli Belt	Afterbody 1	3.066	0.091	6.80	0.775 - 1.182	P, U_x, V_r, $\overline{u^2}$, $\overline{v^2}$, $\overline{w^2}$, $\overline{uv}$
	Afterbody 2	3.066	0.091	6.60	0.84 - 1.182	P, U_x, V_r, $\overline{u^2}$, $\overline{v^2}$, $\overline{w^2}$, $\overline{uv}$
Huang (11) Groves Belt	Afterbody 5	2.91	0.096	9.30	0.74 - 0.987	P, U_x, V_r, $\overline{u^2}$, $\overline{v^2}$, $\overline{w^2}$, $\overline{uv}$
Ramaprian (12) Patel Choi	Hemispheroid with angle of attack = 15	1.37	0.25	2.1	0.176 - 0.87	P, U_s, V_n
Baek (13)	Hemispheroid with angle of attack = 15	1.37	0.25	1.86	0.176 - 0.87	P, U_s, V_n, W, $\overline{u^2}$, $\overline{v^2}$, $\overline{w^2}$, $\overline{uv}$, $\overline{uw}$, $\overline{u^3}$, $\overline{v^3}$, $\overline{uv^2}$, $\overline{u^2v}$, $\overline{w^3}$, $\overline{uw^2}$, $\overline{wu^2}$

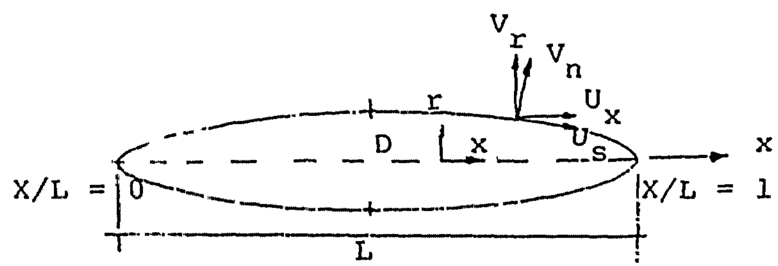


Table 1. Summary Of Experiments

to date, it has not been successfully predicted by any method. The remaining four experiments are quite well known and had been used extensively as test cases. None of these experimental studies involved separation, and all provided mean-flow and turbulence data in the stern boundary layer. The measurements were confined near the body therefore, the data did not provide sufficient information far into the wake. All of the above studies are restricted to axisymmetric flow i.e., without angle of attack.

Ramaprian, Patel and Choi [12] (1981) measured three dimensional flow past an inclined cylinder body. In this study only velocity profiles and pressure along the cylinder body were measured and no turbulent quantities were given. Baek [13] (1984) continued the previous experiment and concentrated in his work on the measurements of turbulent quantities. The other available sources of turbulent data pertaining to inclined cylinder is the experiments made by Meier et al. [14] (1984) at DEVLR in Gottingen, West Germany. However, all of these three dimensional turbulence data are measured only on the body between the region 0.2<x/L<0.9, where L is the body length, x=0 is the leading edge of the body and x=L is the tail of the body. No measurements involving the separation flow or inside the wake are yet available.

1.2.2 Numerical Approach

The primary difficulty in obtaining numerical solution of the three dimensional turbulent flow is that the governing Navier-Stokes and turbulent transport equations are non-linear and elliptic with respect to space variables. The numerical solution of the three dimensional flow problem must be found simultaneously in all three spatial directions. Generally speaking, the computer systems available in the academy or industry are still not large enough to store all the values associated with large number of discretized nodes and variables that are required in three dimensional flow calculation. Thus, to solve the three dimensional turbulent flow problem by using truely elliptic treatment is an impractical proposition at the present time. The computer storage requirements and the computational effort can be greatly reduced by the use of approximate equations whose applicability lies somewhere between that of the fully elliptic Navier-Stokes equations and the classical parabolic boundary layer equations. There are two different approaches in deriving the approximate equations. The first one is to simplify the Navier-Stokes equations by discarding certain terms and the second is to modify the boundary layer equations by introducing additional terms. Both approaches lead to the same equations. These intermediate equations representing an improvement over the classical boundary layer approximations and have been used by many authors [15]. In solving these approximate equations the integral as well as differential methods have been proposed. Generally, the integral method needs additional empricism to predict the crossflow across the boundary layer and this empricism varies from one problem to the other. A more general solution can be obtained if the differential solutions are solved numerically. Many available three dimensional solutions for the external flows problems are based on boundary layer equations rather than the Navier-Stokes equations. This approach has the major shortcoming of not being able to predict the flow separation. In order to develop a prediction scheme that is capable of predicting separation flow one must consider the Navier-Stokes equations. Here only the differential methods based on the Navier-Stokes equations are reviewed. Among the works based on simplified Navier-Stokes equations known as the partially-parabolic equations is perhaps the most popular one to date used in the three dimensional flow problem. The partially-parabolic approximations were first introduced by Patankar, Pratap and Spalding [16,17] (1972), to describe internal flows in a curved tube where the predominant flow direction is along the tube axis. The diffusion term is neglected in the approximation. Although there is no recirculation in the flow, significant pressure effects, unlike boundary layer assumption, propagate upstream. By neglecting the diffusion term in the axial direction the governing equation become parabolic in the axial direction for the velocity variables. It should be remarked that the pressure variable is governed by an elliptic equation which can be shown if the velocity variables in the continuity equation are expressed in pressure variable. Nevertheless, the numerical solution can be obtained by marching the solution from the upstream to the downstream instead of solving the whole velocity field that is required in the elliptic case. The approximation still enables one to compute a wide class of three dimensional flows of practical interest other than that of boundary layer flows.

Table 2 summarizes some numerical works using a partially parabolic approach on three dimensional external turbulent flow. They are Abdelmeguid, Markatos and Spalding [18] (1978), Muraoka [19] (1980), Huang and Chang [20] (1985), Chen and Patel [21,22] (1985). Similar to the published applications in internal flows [16,17], these studies used essentially the same numerical scheme based on the work of Patankar and Spalding et al. [23], and the k-ε turbulence model with specified wall functions for flow variable near the wall boundary. Some good success of numerical predictions is obtained in [20,21,22]. However,

Authors (Ref.)	Coordinates	Numerical Scheme	Turbulence Model	General Comment
Abdelmeguid (18) Markatos Spalding	Distorted cylindrical polar coordinates $\xi = x$ $\eta = (r-r_a)/(r_0-r_a)$, $r_a<r<r_0$ $\zeta = \theta/(\pi/2)$, $0<\theta<\pi/2$.	Spalding-Patankar finite-volume scheme and SIMPLE algorithm for pressure update via continuity equation.	2-equation $K-\epsilon$ with Wall function around the wall.	1) The first work using partially-parabolic approach for flow past ship hulls. 2) A coarse grid (25x12x12) in $0.75 \le X/L \le 1.2$. 3) The potential solution specified on the boundary around the hull. 4) Only the longiludinal velocity compared with the experiments.
Muraoka (19)	as same as above	as same as above	as same as above	1) The same method as the above. 2) Flow past body of revolution. 3) Very coarse grid (35x12x12) in $0.5 \le X/L \le 1.18$, $R_a < R < (R_a - 3.346L)$; 4) Quite encouraging results obtained in spite of a rather small solution domain.
Huang (20) Chang	Algebraic nonorthogonal streamline	as same as above	2-equation $K-\epsilon$ matches to mixing length in wall region. Simplified wall function below $y^+ = 30$.	1) Good agreement between the results and experimental data. 2) Reynolds stresses in the tail and rear-wake region underestimated. 3) Typically grid (37x34) in $0.6 \le X/L \le 4$; $R_a \le R \le 0.09L$.
Chen (21) Patel	Numerical body-fitted nonorthogonal coordinate	FA scheme for transport equations. SIMPLER algorithm for pressure.	2-equation $K-\epsilon$ 2-point wall function with stress-gradient correction	1) Good agreement between the solutions and experimental data. 2) Typically grid for axisymmetric flow (50x19), for 3D case (46x19x7), in $0.5 \le X/L \le 12.0$, $R_a < R \le 0.8L$ 3) Velocity and kinetic energy k overestimated around the tail and wake.
Chen (22) Patel	Numerical body-fitted nonorthogonal coordinate	FA Scheme for transport equations. Modified SIMPLER with global pressure solution procedure.	2-equation $K-\epsilon$ 2-point wall function with stress-gradient correction	1) The same method as the above. 2) Flow near real ship hulls. 3) The grid used (30x19x14) in $0.5 \le X/L \le 2.302$, $R_a \le R \le 1$. 4) Good agreement between the calculated velocity and experimental data.

Table 2. Summary Of Numerical Works

three dimensional flows that can be described by the approximated partially-parabolic equations are limited to those flows having no flow separation. In the present study the numerical solution is derived, in addition to the above flows, for prediction of separation flows. In order to achieve the prediction capability of separation flows, instead of neglecting the diffusion term in the direction of the predominant flow direction as in the partially-parabolic approximation, in this study the diffusion term is kept and cast into the source term, hence the fully elliptic governing equations are retained. The approximation is made only numerically to evaluate the longitudial diffusion term from the previous iteration. This approximation called the "semi-elliptic approximation" has an advantage that the fully elliptic solution is kept in the whole computational domain and that the computational effect and storage equals to that of the partially-parobolic approach. More details of the semi-elliptic solution procedure and numerical analysis will be explained later in chapter III.

1.3 Selection Of Methods And Models

As mentioned before, the numerical predictions of three dimensional turbulent flow are complicated and sometimes unreliable. These difficulties are not just with the geometry treatment, coordinates and numerical method adopted, but also in the turbulence models used to relate the Reynolds stresses to the mean flow. Therefore, in order to solve a complex three dimensional turbulent flow one must, in addition to an appropriate coordinate system and an accurate numerical method, have a turbulence model that is valid for three dimensional flow including flow separation. The selection of coordinate system, numerical method and turbulence model are outlined in the following sections.

1.3.1 Coordinate System

The simple and familiar coordinate systems that are usually used in the early numerical approach are Cartesian, Cylindrical and Spherical coordinates. However, it is obvious that these coordinate systems are appropriate only for the problem geometry having the coordinate lines as the boundaries. Consequently, the numerical solution based on the standard coordinate systems has difficulty in specifying the non-slip boundary conditions on the surface of arbitrary shaped bodies. In the present study the body-fitted coordinate system, such that the surface of the body is one member of a family of the coordinate surfaces, is used to avoid this difficulty.

The ideal body fitted coordinate system is the analytic, orthogonal curvilinear coordinate system that traces the problem boundary and satisfies the requirements of orthogonality in three dimensional space. In this system, the coordinate surfaces normal to the body must intersect the body surface in its lines of principal curvature. Thus, to find this coordinate system for a body of any given shape it is necessary to obtain the lines of curvature of the body surface, which can be mathematically derived as shown in the paper published by Miloh and Patel [24]. Unfortunately, the use of the lines of principal curvature to form the analytical body fitted coordinate system is not very convenient since the determination of these lines of principle curvature is rather complicated. Moreover, for an arbitrary shaped body the curvature of these lines may be quite large on some part of the body so that the numerical evaluation of coordinate value requires great care to attend sufficient accuracy.

To rectify the problem, many numerically-generated coordinate systems have been tried in the past decade. Investigators have constructed curvilinear meshes to span the whole physical region and some others have even tried different modifications of conformal transformation procedures [25]. The real breakthrough came from the elliptic-transformation procedure proposed by Thompson et al. [26]. In this method, one of the coordinate lines or surfaces is matched with the body identically and another with some outer boundaries, and internal points of the physical region are automatically generated on the computer by the solution of an elliptic system of partial differential equations. In contrast to a conformal mapping procedure, which is limited to bodies in two dimensions, the Thompson's procedure can be extended to three dimensional problems. The numerically-generated body fitted coordinates is not only proved to be mathematically sound, but also has the capacity of contracting the coordinate lines to a point or along some specified lines and surface in the physical region. Generally, the contraction of the grid lines to the surface of of body is required for obtaining more accurate results while solving the flow problem which has large gradient and rapid change of variables near the body. Therefore, in this study the body-fitted coordinate system based on Thompson [26] is used to generate the grid distribution in the computational domain. It should be remarked here that the use of the body-fitted coordinate systems which enables us to solve the flow past a body with irregular body shape has some disadvantages. The first is that many cross derivative terms are added to the governing equations after the transformation from the physical coordinates to the body-fitted coordinates. Therefore one must solve more complex governing equations in the body-fitted coordinates. The second is that numerical error due to difference approximation in those transformation functions relating the physical and body-fitted coordinates may decrease the accuracy of numerical solution. Third, the departure of the body-fitted coordinates from orthogonality may create additional difficulties and numerical error during the computation.

1.3.2 Numerical Method

Depending on how the algebraic representation of the differential equation is derived the numerical method may be classified as finite difference, finite volume, finite element or finite analytic. In the finite difference method [27] the discrete algebraic equation is obtained by Taylor-series expansion of differential terms while the finite volume method [23] derived the algebraic equation by formulating the conservation principle in a finite control volume without taking the limits to the infinitesimal volume. In the finite element method [28], the variational formulations and the method of weighted residuals are often used to derive an intergal form before an algebraic equation relating the nodal values in the element is obtained. The finite analytic method presented by Chen et al. [29-37] invokes another means of deriving the algebraic equations. Unlike the finite difference, finite volume or finite element method, the discretized algebraic equation is obtained from the analytic solution for each local element.

Finite element method which is used very successfully in the solid mechanics was first introduced to the fluid problem in the late of sixties. And the first conference on finite element in fluid mechanics was held in 1974 at Swansea [28]. During the last two decades the number of the applications of the finite element procedure for various areas of fluid mechanics had been increased. However the majority of applications of finite element in fluid mechanics occur in the slow viscous flow, wave phenomena and fluid-structure interaction. But for high Reynolds number or turbulent flow and external flow the finite element solutions are scarce. This is partly due to the fact that at high Reynolds number the finite element treatment of the convective term is often inadequate and the finite element solution can become unstable and inaccurate. Since in the present study the high Reynolds number external turbulent flows are going to be solved, then the finite element approach is not a suitable numerical method for this study.

The finite difference method is perhaps the most used numerical method in solving fluid flow. Various forms of finite difference methods had been used to solve fluid problems for a long time, and many successful solutions have been obtained. However there are still several difficulties in obtaining finite difference numerical solution. The first is the numerical instability in solving the system of algebraic equations that approximate the governing partial differential equations. The second is numerical error associated with the numerical method known as numerical or false diffusion which in some situations can become so severe as to completely overshadow the physical turbulent or viscous diffusion. The instability of the finite difference solution basically arises from the improper finite difference approximation of the original governing differential equation. If the original governing differential equation is well posed the numerical solution of the properly approximated finite difference equation must be stable. However the proper and accurate finite difference equation for the Navier-Stokes equation is not easy to derive. On the other hand it is known [27] that the finite difference equation based on the central difference approximation for the Navier-Stokes equations is unstable when the element Reynolds number is greater than two. To partially overcome the instability of the finite difference solution of Navier-Stokes equation, the upwind scheme was introduced [38] to preserve the proper characteristic of the original partial differential equation that is present. The upwind scheme uses some special formula to shift the weight of the difference scheme or nodal influence on the element to the points where the flow pass. However, if the formulation of upwind scheme is improperly implimented, the scheme may produce large numerical diffusion even though the solution is stable.

Unlike the finite difference method, the finite analytic method invokes the analytic solution of the governing equation in the local element in formulating the algebraic representation of the governing partial differential equation. The finite analytic method produces a stable solution because the analytic solution of a well posed problem is stable. Further more the finite analytic solution has the ability of automatically upwinding shift of the weight of the coefficients that are associated with the analytic algebraic equation. The finite analytic solution thereby minimizes the false numerical diffusion while providing a stable solution. The finite analytic method has been applied successfully in solving the vortex shedding, recirculation flow, free convection flow and laminar and turbulent flow at high Reynolds number [21,22,29-37]. From these published results the finite analytic solutions were shown to be indeed stable and accurate. Consequently the finite analytic method is adopted in this study.

1.3.3 Turbulence Model

Equations for describing the fluid motions, known as the Navier-Stokes equations, have been postulated and derived for over a century. However, it is difficult to solve these equations for both laminar and turbulent flows mainly due to the nonlinearlity of the equations. For turbulent flows, the difficulty is even more formidable to overcome because the turbulent fluid motion is irregular, random, time dependent and three dimensional. However, in most engineering applications, the detailed analysis of instantaneous turbulent motion is not necessary and the gross-parameters like mean velocity, average pressure and wall shear stress are often sufficient for engineering analysis and design.

In studying the turbulent flow O. Reynolds [39] proposed an averaging technique by assuming that the variable ϕ^* at any instant of time to consist of the mean quantity ϕ, an averaged value of ϕ^* during the long time period T, i.e.

$$\phi = \frac{1}{T}\int_0^T \phi^* \, dt$$

and a fluctuating part ϕ'. Hence,

$\phi^* = \phi + \phi'$ The time averaging process when applied to the Navier-Stokes equations, creats six additional unknowns $\overline{u_i u_j}$. These unknowns, although called Reynolds stress, are created from the convective or non-linear terms of the Navier-Stokes equations. Instead of time average a more general average, ensemble average, can be used to derive turbulent Navier-Stokes equations. In the ensemble average, the averaged value ϕ is now the average of many repeated same experiments or $\phi = \frac{1}{N} \Sigma \phi^*$

where N is the total number of the experiments, and ϕ' is the deviation of the instantaneous value ϕ^* from that of the ensemble averaged value ϕ. The advantage of considering ensemble average is that the process of ensemble average may be applied to unsteady turbulent flows, preserving the time dependence in the average value ϕ which the original Reynolds average can not do.

Many turbulence models have been proposed to evaluate the unknown Reynolds stress. All models have them coupled to the mean quantities through either algebraic or differential equations. Some are based on empirical relation and others on postulations.

In 1877, Boussinesq [40] proposed the concept of eddy viscosity which assumes that, in analogy to the viscous stresses in laminar flows, turbulent stresses are proportional to the mean velocity gradients. For general flow situations, it is expressed as

$$- \overline{u_i u_j} = \nu_t \left(\frac{\partial U_i}{\partial X_j} + \frac{\partial U_j}{\partial X_i} \right) - \frac{2}{3} \delta_{ij} k$$

Here ν_t is the turbulent or eddy viscosity which, unlike molecular viscosity, ν, is not a fluid property but depends on the state of turbulence. k represents the kinetic energy of the fluctuating motion or $\overline{u_i u_i}/2$. Boussinesq did not provide a general model for ν_t. In 1925, Prandtl [41] proposed a turbulence model called the 'mixing length' model. This model created a relation for the eddy viscosity, as a function of a length scale L characterizing the size of turbulent eddies and a suitable turbulent velocity scale, V. Since ν_t has the dimension of length squared over time, Prandtl proposed $\nu_t \propto VL$. Both the turbulent velocity scale, V, and the mixing length scale, L, could be reasonably approximated for many flows. However for each flow empirical constants were needed to prescribe a length scale. The flows that are most successfully modelled by the mixing length model are of thin shear flows such as boundary layer, jets, wake, mixing layer flows and pipe flows. The constants of the mixing length model were obtained by fitting the calculated results to experimental data of a particular flow under study. These mixing length model constants were found [42] to vary often from one flow to another. Consequently, the mixing length turbulence model is successful only in predicting turbulent flows in similar geometry and flow conditions but lacks the universality and predictability when the turbulent flow and geometry are different.

To overcome the lack of predictability and generality, several more complex models were developed during the 1940's and 1950's which employed differential transport equations for the turbulent quantities. However, these equations could not be solved directly as there were mathematical difficulties involved and numerical techniques and fast computers were not available. Alternatively, the governing partial differential equations for turbulent flows were often solved by integral method which reduced the governing partial differential equations to ordinary differential equations. These integral methods assumed some shape of mean profile and used some empirical relations for global behavior of turbulence. They lacked flexibility since the assumed profile must be approximately the same in the flow field and could not be applied for different flows.

Advances in computational facilities and numerical methods during the late 1960's and 1970's led to the use of more advanced models which solve complete partial differential equations for both mean flow and turbulent quantities. One of these models which solves the differential equation for the turbulent kinetic energy, k or $\overline{u_i u_i}/2$, is called the one-equation model as opposed to the zero-equation model proposed by Boussinesq or Prandtl where no differential equations are solved for turbulent quantities. With the kinetic energy known, the Boussineq's eddy viscosity can be written as

$$\nu_t = C_\mu K^{0.5} L$$

where C_μ is an empirical constant, k represents a turbulent velocity scale where k is solved from the modelled governing equation of the turbulent kinetic energy, and L the length scale is a varible which is obtained from simple empirical relations similar to those for the mixing layer. The one equation model was found [42] useful only in predicting thin shear flow since in many complex flows it is difficult to specify the length scale empirically. The logical extension of the turbulence modelling is that the length scale be obtained from a differential transport equation.

Models which solve differential equations for both turbulent velocity scale or turbulent kinetic energy k, and length sacle or alternatively the dissipation rate of turbulent kinetic energy ε ($= \nu \overline{\frac{\partial u_i}{\partial x_j} \frac{\partial u_i}{\partial x_j}}$) are known as two-equation models. The most popular one is the one suggested by Jones and Launder [43] which has $k^{1.5}/L$ instead of L. The term $k^{1.5}/L$ has physical significance as it has the same dimension as, ε, the dissipation rate of turbulent kinetic energy. Hence this model is usually called k-ε turbulence model. The conventional k-ε turbulence model which only uses k and ε to characterize the turbulent velocity ($\sqrt{k}$), length ($k^{1.5}/\varepsilon$) and time (k/ε) scale will be called one-scale k-ε turbulence model in this study. It was found that one-scale k-ε turbulence model can predict acceptable mean flow variables when flow geometry is not too complex. It also can predict a fair result for the turbulent transport properties. However, the one-scale k-ε turbulence model was found to be unsatisfactory in predicting the result for two dimensional and axisymmetric jets unless the turbulence constants which calibrated with experimental data are altered. In order to improve the prediction capability of turbulence model a two-scale k-ε turbulence model was proposed recently by Chen and Singh [44]. This model employs the concept of two different scales in characterizing the turbulent scales. One scale which is based on k and ε for the large energy containing eddies ($l=k^{1.5}$, $v= k$, $t=k/\varepsilon$) is used for modelling turbulent diffusion and other turbulent production phenomena and the other which is based on Kolmogrov's scale [45] ε and ν for the small eddies in the dissipation range ($l=(\nu^3/\varepsilon)^{0.75}$, $v=(\nu\varepsilon)^{0.25}$, $t=(\nu/\varepsilon)^{0.5}$) to model destruction of dissipation of turbulent kinetic energy and other turbulent dissipation phenomena. Based on the two-scale concept the ε equation is remodeled. It is found [44] that the two-scale k-ε turbulence model can predict many turbulent free shear flows and some recirculation flows without altering the turbulent constants including the turbulent two dimensional and axisymmetric jets and turbulent wakes and mixing phenomena. During the early stage of this study the two-scale k-ε turbulence model was tested for the boundary layer flow and found that the FANS-3DEF with the turbulent constants of the two-scale k-ε model often encounter numerical instability. It is found that the numerical instability started when the dissipation rate of the turbulent kinetic energy, ε, is larger than the production of turbulent kinetic energy. Therefore it is reasoned that the turbulent constants C_k, C_ε, $C_{\varepsilon 1}$ and $C_{\varepsilon 2}$ used in the two-scale turbulence model require further investigation. In the present study the one-scale k-ε model is choosen since the one-scale k-ε model in the present form was known to be stable in predicting the turbulent external flow although the model required further improvement in modeling.

1.4 Scope Of The Study

This study is undertaken to develop a prediction method capable of analyzing both laminar and turbulent flows past a finite or semi infinite two dimensional or axisymmetric body with and without an angle of attack. In chapter II, the partial differential equations governing the flow situation considered in the present study are described. The different turbulence models and the treatment of the boundary conditions near the wall are also discussed. In chapter III, the derivation of finite analytic formulation on the body-fitted coordinates, the formulation of pressure equation on a control volume and the description of numerical algorithm used in this study are presented. From chapters IV to chapter VI the prediction of the laminar and turbulent flow past two dimensional and axisymmetric bodies with and without angle of attack are given and discussed. In chapter VII, the conclusions of the present study are summarized and the recommendations for the future work are proposed.

The brief formulation for calculating the two dimensional finite analytic coefficients are given in appendix A. In appendix B, the brief introduction of the computer program FANS-3DEF (Finite Analytic Numerical Solution for Three Dimensional External Flow) and sample output on the interactive screen are outlined, and the complete program of FANS-3DEF is listed.

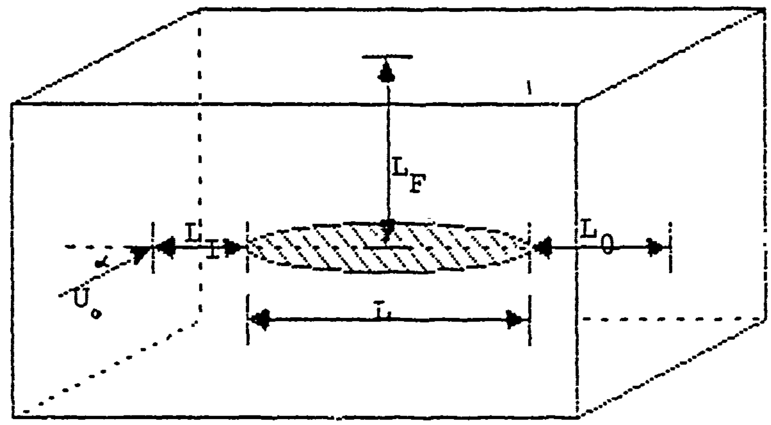
CHAPTER II

MATHEMATICAL FORMULATION

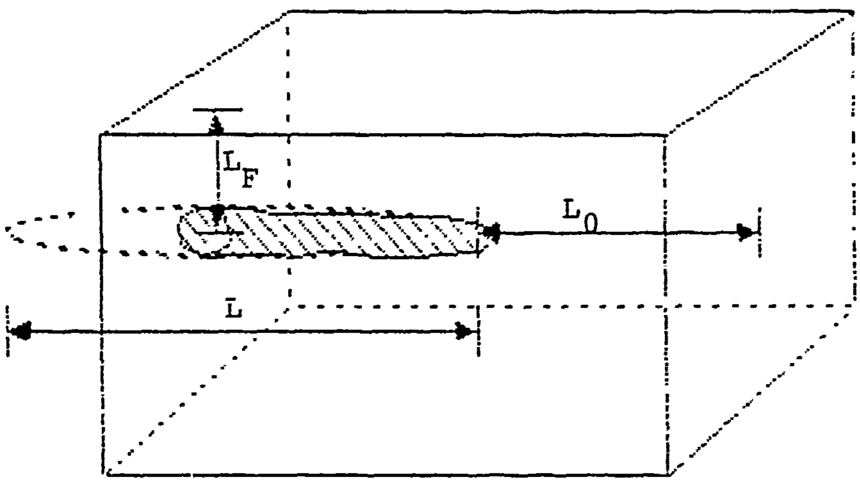
In this chapter a general mathematical formulation for predicting laminar and turbulent flow past a two dimensional and axisymmetric body with or without an angle of attack is introduced. The general governing equations for three dimensional turbulent flow are first formulated in Cartesian coordinates. The turbulence model based on the second order correlation for the Reynolds transport equation is then considered. The general features of boundary conditions are also stated to complete the mathematical formulation. Therefore, simple geometries like a flat plate or a cylindrical tube can be treated as a special case of the general formulation. All governing equations and boundary conditions are then transformed and rewritten in the body-fitted coordinate systems.

2.1 Governing Equations

Figure 1 depicts the whole computational domain to be considered in this study and a general geometry of a body which is subjected to an incoming flow U_0 with an angle of attack α. The body geometry can be thought to simulate an



(a) Total Body Domain



(b) Partial Body Domain

Figure 1. Computational Domain And Body Geometry

airborne object in air, a ground vehicle on the road or a submerged marine ship in the sea. For the total body domain the body which has a characteristic length of L is located in the center of the computational domain. The tip of the body is located at a distance L_I downstream of the inlet boundary, the body center is located at a distance L_F from the side boundary, and the rear end of the body is kept at a distance L_O from the downstream boundary. If a small computational domain is desired a partial body domain can also be considered as shown in figure 1(b).

For a three dimensional turbulent flow problem, the ensemble averaged incompressible Navier-Stokes equations in Cartesian tensor form are

$$\frac{\partial U_i}{\partial X_i} = 0 \tag{1}$$

$$\left(\frac{\partial U_i}{\partial T} + U_j \frac{\partial U_i}{\partial X_j} \right) = -\frac{1}{\rho} \frac{\partial P}{\partial X_i} + \frac{\partial}{\partial X_j} \left\{ \nu \left(\frac{\partial U_i}{\partial X_j} + \frac{\partial U_j}{\partial X_i} \right) - \overline{u_i u_j} \right\} \tag{2}$$

where $u_i u_j$ are turbulent Reynolds stresses. When the flow is laminar the Reynolds stresses are set equal to zero. Equations (1) and (2) are 4 independent equations governing 4 unknowns, U, V, W, P, and providing existence of solutions. Unlike laminar flow, if the flow is turbulent, equations (1) and (2) have 4 equations but with 10 unknowns. They are U, V, W, P, $\overline{uu}$, $\overline{vv}$, $\overline{ww}$, $\overline{uv}$, $\overline{uw}$, and $\overline{vw}$. Clearly, the closure of the turbulent problem requires additional information between the Reynolds stresses and the mean flow variables. The closure of turbulent flow equation can be done by the introduction of turbulence models for the Reynolds stresses, $\overline{u_i u_j}$, which is discussed in the next section.

2.2 Turbulence Model

In order to solve turbulent flow problems governed by Eqs. (1) and (2), the Reynolds stresses, $\overline{u_i u_j}$, must be known. In general, the exact equation for turbulent quantities, $\overline{u_i u_j}$, can be derived from Navier-Stokes equations. However, in these turbulent transport equations there exist additional unknown correlations other than $\overline{u_i u_j}$. Therefore, a turbulence model must be established to close the problem. The turbulence model may be classified according to how the Reynolds stresses that appear in the ensemble averaged Navier-Stokes equations are modelled. Generally, the more the number of differential transport equations are solved the more complete the turbulence model becomes. However, the effort in analyzing large numbers of differential equations will also increase. As mentioned in chapter I, the current trend in turbulent modelling is to model the Reynolds stresses by transport equations for the second order correlation. In the past ten years, the two equation k-ε turbulence model has become the most popular model in the turbulent flow calculation.

In the k-ε turbulence model the turbulent kinetic energy k ($=\overline{u_i u_i}/2$) and its dissipation rate ε ($= \nu\overline{\frac{\partial u_i}{\partial X_j}\frac{\partial u_i}{\partial X_j}}$) are solved from two modelled differential transport equations. The Reynolds stresses $u_i u_j$ is then a function of k, ε and other known mean velocity quantity. Typically, the two equation k-ε turbulence model contains five empirical constants which are determined from some basic experimental configurations such as grid turbulence, homogeneous shear flow and boundary layer flow [42]. Although more effort is required in analyzing the two equation k-ε turbulence model than in other simpler models such as the mixing length model proposed by Prandtl [41], it is found [46] that the k-ε model or more generally the second order closure model with its empirical constants are less problem dependent. Therefore, some hope for predictability and universality of the turbulence model is established although the model still require further investigation and improvement.

In the present study the conventional one-scale k-ε turbulence model, known as the standard k-ε turbulence model, by Launder et al. [46] is considered. In the k-ε turbulence model the Reynolds stresses $\overline{u_i u_j}$ can be modeled either approximating the differential Reynolds stresses transport equation into an algebraic form or by an algebraic equation based on Boussinesq's assumption which relating Reynolds stresses to the gradients of mean velocities as

Here ν_t is the eddy viscosity and based on the dimensional analysis of (k,ε) we have

$$-\overline{u_i u_j} = \nu_t\left(\frac{\partial U_i}{\partial X_j} + \frac{\partial U_j}{\partial X_i}\right) - \frac{2}{3}\delta_{ij}\,k$$

where C_μ is an empirical constant (=0.09-0.128), k is the turbulent kinetic energy per unit mass ($k=\overline{u_i u_i}/2$), and ε is the dissipation rate of k ($= \overline{u_i u_i}/2$). In the present study generalized Boussinesq's equation (3) is adopted.

In addition to algebraic Reynolds stresses equation (3), two differential transport equations, namely, the turbulent kinetic energy and the rate of dissipation of turbulent kinetic energy are needed to close the problem. In this study the turbulent kinetic energy ,k, and its dissipation function are solved from following two modelled equations [46].

$$\frac{Dk}{DT} = \frac{\partial}{\partial X_i}\left\{\left(\nu + C_k\frac{k^2}{\varepsilon}\right)\frac{\partial k}{\partial X_i}\right\} - \left(\overline{u_i u_j}\right)\frac{\partial U_i}{\partial X_j} - \varepsilon \qquad (4)$$

$$\frac{D\varepsilon}{DT} = \frac{\partial}{\partial X_i}\left\{\nu + C_\varepsilon\frac{k^2}{\varepsilon}\right)\frac{\partial k}{\partial X_i}\right\} + \left\{C_{\varepsilon 1}\left(-\overline{u_i u_j}\right)\frac{\partial U_i}{\partial X_j} - C_{\varepsilon 2}\varepsilon\right\}\left(\frac{1}{t}\right) \qquad (5)$$

Here t in Eq. (5) is the characteristic turbulent time scale associated with the destruction of ε. If t is determined based on k and ε or $t=k/\varepsilon$ then the turbulence model is the conventional one-scale k-ε turbulence model. The model constants C_μ, C_k, C_ε, $C_{\varepsilon 1}$ and $C_{\varepsilon 2}$ in the one-scale k-ε

turbulence model can be determined from several basic experiments, namely isotropic grid turbulence, homogeneous shear flow, and boundary layer flow. The model constants suggested by Launder et al. [46] are:

$$C_\mu = 0.09, \quad C_k = 0.09, \quad C_\varepsilon = 0.07, \quad C_{\varepsilon 1} = 1.44, \quad C_{\varepsilon 2} = 1.92$$

With the introduction of above turbulence model, the problem for solving turbulent flow is closed. A unique solution of equations (1) to (5) can be obtained for U, V, W, P, $u_i u_j$, k and ε if the boundary conditions for U, V, W, P, k and ε are properly specified.

2.3 Boundary Conditions

In addition to the governing equations (1) through (5), the complete specification of external flow past a body requires an adequate prescription of boundary conditions. This means that the flow conditions must be specified at the inlet and outlet planes and at the lateral boundaries of the flow domain of interest (see figure 1). It should be noted that the boundary location may be placed arbitrarily with respect to the solid body by assigning different values of L_I, L_F and L_O for the computational domain.

(1) Inlet plane: The inlet plane is located at L_I distance upstream (Fig. 1(a)) or downstream (Fig. 1(b)) from the tip of the body. If L_I is placed far upstream from the tip of the body then the uniform velocity profile with or without

angle of attack is specified. In this study a constant ambient pressure, zero ambient turbulent kinetic energy k and its dissipation rate ε are assigned at the inlet plane. If inlet plane is placed at L_I distance downstream of the tip of the body then the distribution of the velocity components (U,V,W), pressure P and the turbulence quantities (k,ε) are prescribed at this plane either from detailed experimental measurements, boundary-layer calculation, or from simple flat-plate correlations.

(2) Outlet plane: Since this study includes the flow phenomena inside the wake region the outlet plane is always chosen to be far downstream of the body where the second derivatives of all variables are set equal to zero. This implies that the effect of diffusion from the outlet plane to the upstream locations are negligible.

(3) Lateral boundaries: There are three types, namely: walls, planes of symmetry and free stream boundaries.

(i) Wall boundaries: The wall of the body can be plane, cylinder or arbitrary cross section. For laminar flow, the numerical solutions are carried out upto the wall where the usual no-slip conditions, U=V=W=0, are imposed. For turbulent flows, since the turbulence model can not be employed in the viscous sublayer region, an alternative method should be used instead of applying no-slip conditions directly. In this study the two-node wall function is used to avoid the use of the low Reynold number turbulence model or a large number of grid points to resolve the large gradients in the near-wall region. The basic idea of the two-node wall function is the numerical solution from the wall to the first two nodal points near the wall and is replaced by a semi-analytic solution obtained from the turbulent inner layer equation for the near-wall region namely the log-law equation [47]. In doing so, the first two computational nodal points are placed at nondimensional distance y^+, y^+ away from the wall. Here the values of y^+ and y^+ should be arranged between 12 to 200 and y^+ is defined as

$$y^+ = \frac{U_\tau Y}{\nu}$$

where U_τ is the friction velocity or $(\sqrt{\tau_w}/\rho)$ with τ_w as total wall shear stress, Y is the distance away from the wall. If U_1, U_2 are respectively the resultant velocities parallel to the wall at first two nodal points as shown in figure 2, then wall boundary conditions can be specified through the log-law equation by the following steps.

(1) Using an initially assumed or update velocity U_2 through log-law equation to obtain U_τ.

(2) Using U_τ which just obtained from step (1) through log -law equation to obtain velocity U_1.

(3) U_1 is then used as the boundary condition for turbulent flow calculation.

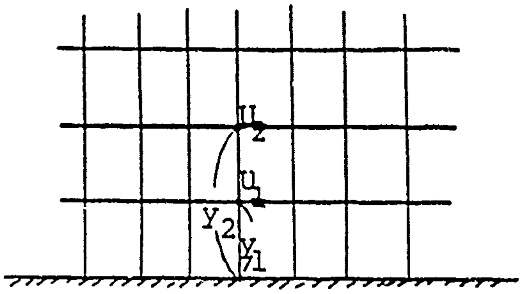


Figure 2. Two-Node Wall Function

In this study a two dimensional log-law euqtion for U velocity component based on fully developed or parallel flow assumption is used. It is [45]

$$\frac{U}{U_\tau} = \frac{1}{\kappa} \ln (E\, y^+) \qquad 12 < y^+ < 200 \tag{6}$$

with Karman constant κ=0.42 and integration constant E=9. The corresponding turbulent kinetic energy k and its dissipation rate ε at the first node are given [47] as

$$k_1 = \frac{U_\tau^2}{\sqrt{C_\mu}} \qquad \varepsilon_1 = \frac{U_\tau^3}{\kappa y_1} \tag{7}$$

Here C_μ=0.09 is determined empirically [42] and C_μ=0.128 if the value is obtained from the algebraic reynolds stress model [47].

It should be mentioned that the normal velocity componenent is taken to be zero at the first nodal point from the wall. This may not be the case when the flow separates near this node. At present there is no known wall function for flow near the separation. Thus the general practice is to use the same wall function, Eqs (6) and (7), even for the flow involved separation. Numerically the point of separation is unlikely to occur at one numerical node exactly. In other words τ_w or U_τ at those numerical nodes even close to the separation will have non zero values. Therefore the U_1 velocity at the first node y^+ will have a value, either positive or negative depending on the direction of τ_w. The positive value denotes the point before the separation while the negative value denotes the point behind the separation. In the region where the flow near the separation zone, either before or behind the separation the wall functions (6) and (7) for U_1 may still be approximately used since the flow vector is properly oriented and since the U_1 velocity at y^+ is small while near the separation. Although the use of wall function based on parallel flow assumption for the nodes near the separation is questionable, but this is currently done in just about every turbulent prediction calculations using a wall function. The weak justification of such a practice is that the number of nodes that are near the separation is far less than the total number of nodes where the wall functions (6) and (7) are applicable. Thus the error caused by the above practice may be confined only near the separation point. In the actual test from many calculations it seems to bear out that the separation phenomena can be reasonably predicted with this practice.

(ii) Symmetric planes: In some flow problems the symmetric condition may be used. For example, the flow past a symmetric body with no angle of attack. If no vortex shedding is expected in the axisymmetric flow the symmetric condition may be imposed on the line or plane of symmetry so that a smaller computational domain can be used to save computer time and storage. The velocity components which are normal to the line or plane of symmetry are set equal to zero, and there are no fluxes of any variable across symmetric planes.

(iii) Free stream boundaries: In figure 1 it is shown that the free stream boundaries were set at L_O distance away from the axis of the body. L_O is set far enough as numerically and computationally possible to avoid any unrealistic representation. The normal derivative of all vailables along the free stream boundary are set to be zero.

After specifying the boundary conditions along the boundary the mathematical description of the problem is complete. Since the exact mathematical solution can not be obtained, the numerical analysis of the problem is considered and discussed in the next chapter.

For the convenience of the numerical analysis the governing equations (1) to (5) are made dimensionless and summarized below

$$\frac{\partial u_i}{\partial x_i} = 0 \qquad (1)'$$

$$\left(\frac{\partial u_i}{\partial t} + u_j\frac{\partial u_i}{\partial x_j}\right) = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial}{\partial x_j}\left\{\left(\frac{\partial u_i}{x_j} + \frac{\partial u_j}{x_i}\right)\right\} - \frac{\partial}{\partial x_j}(\overline{u_i'u_j'}) \qquad (2)'$$

$$-\overline{u_i'u_j'} = C_\mu \frac{k^{*2}}{\varepsilon^*}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}\, k^* \qquad (3)'$$

$$\left(\frac{\partial k^*}{\partial t} + u_j\frac{\partial k^*}{\partial x_j}\right) = \frac{\partial}{\partial x_i}\left\{\left(\nu + C_k\frac{k^{*2}}{\varepsilon^*}\right)\frac{\partial k^*}{\partial x_i}\right\} + G - \varepsilon^* \qquad (4)'$$

$$\left(\frac{\partial \varepsilon^*}{\partial t} + u_j\frac{\partial \varepsilon^*}{\partial x_j}\right) = \frac{\partial}{\partial x_i}\left\{\left(\nu + C_\varepsilon\frac{k^{*2}}{\varepsilon^*}\right)\frac{\partial \varepsilon^*}{\partial x_i}\right\} + (C_{\varepsilon 1}G - C_{\varepsilon 2}\varepsilon^*)\left(\frac{1}{t^*}\right) \qquad (5)'$$

$$G = -\overline{u_i'u_j'}\frac{\partial u_i}{\partial x_j} \qquad t^* = \frac{k^*}{\varepsilon^*}$$

Where the variables are made dimensionless with the body length L and approaching velocity U_0 as the references. They are

$$u_i = \frac{U_i}{U_0}, \quad x_i = \frac{X_i}{L}, \quad k^* = \frac{k}{U_0^2}, \quad \varepsilon^* = \frac{\varepsilon L}{U_0^3},$$

$$Re = \frac{U_0 L}{\nu}, \quad p = \frac{P}{\rho U_0^2}, \quad \overline{u_i'u_j'} = \frac{\overline{u_i u_j}}{U_0^2}$$

There are five turbulence constants C_μ, C_k, C_ε, $C_{\varepsilon 1}$ and $C_{\varepsilon 2}$ that must be specified. They are

$$C_\mu = 0.09, \quad C_k = 0.09, \quad C_\varepsilon = 0.07, \quad C_{\varepsilon 1} = 1.44, \quad C_{\varepsilon 2} = 1.92$$

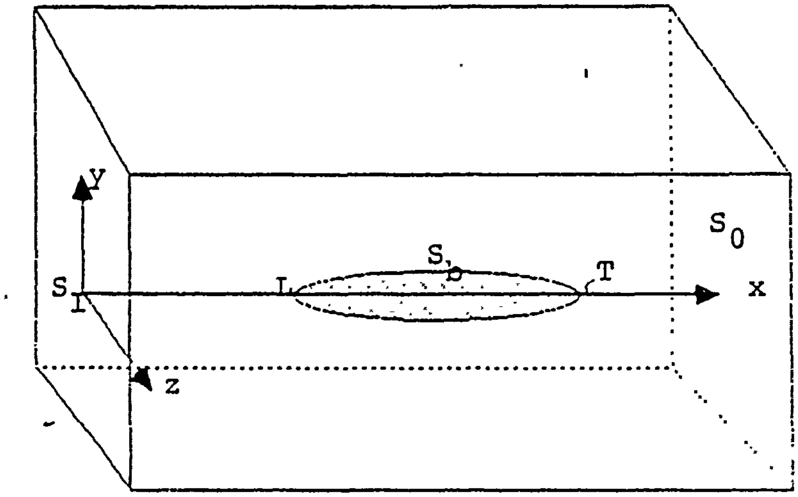
CHAPTER III

NUMERICAL ANALYSIS

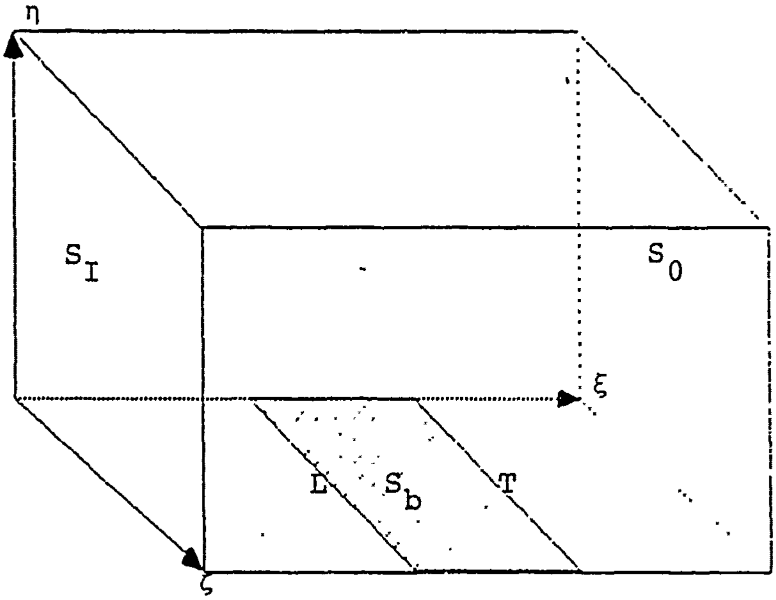
In the following sections the basic idea and principle of the numerical techniques for grid generation (body-fitted coordinate system), the numerical approach (FA numerical method) and the derivation of the pressure equation for modified SIMPLER algorithm are discussed.

3.1 Body-Fitted Coordinates

In order to develop a prediction scheme for a three dimensional flow past an arbitrary two dimensional or axisymmetric body, the body-fitted coordinate system proposed by Thompson et al. [26] is used in this study. The basic idea of body-fitted coordinate system is to generate a curvilinear coordinate system as shown in figure 3 which has coordinate surfaces coincided with all the boundaries of a general multiply-connected body including the boundaries formed by solid walls and external boundaries. Thus, on the transformed domain, the numerical solution of the governing equations may be obtained on a fixed parallelepiped domain with a uniform mesh size. In this way no interpolation of the boundary variable is



(a) Physical Domain



(b) Transformed Domain

Figure 3. Physical And Transformed Domain

required regardless of the shape of the physical boundaries or the spacing of the curvilinear surfaces in the physical domain.

These body-fitted coordinates (ξ, η, ζ) can be generated from the solution of three partial differential equations with Dirichlet boundary conditions, to provide the contour of $\xi(x,y,z)$, $\eta(x,y,z)$ and $\zeta(x,y,z)$ on the physical plane (x,y,z). The partial differential equations are

$$\nabla^2\xi = F1, \qquad \nabla^2\eta = F2, \qquad \nabla^2\zeta = F3 \tag{8}$$

Here ∇^2 is Laplacian operator and F1, F2 and F3 are control functions which are used to concentrate the grid lines to the desired region.

When the flow problem with Eqs. (1) through (5) in (x,y,z) physical plane are transformed by Eq. (8) into the (ξ, η, ζ) transformed plane, the computational domain will become a simple rectangular domain which is shown in figure 3. Therefore it is more convenient to perform numerical calculation in uniform cubic grid in the transformed domain (ξ, η, ζ). It should be mentioned that in this study the ξ coordinate on the transformed domain for convenince is taken to coincide with the axial direction, x, in the physical domain. In order to solve Eqs. (1) through (5) in (ξ, η, ζ) coordinate system it is more convenient first to inverse Eq. (8) into a form of $x(\xi, \eta, \zeta)$, $y(\xi, \eta, \zeta)$ and $z(\xi, \eta, \zeta)$ so that the contour of ξ, η and ζ can be accurately identified on the physical coordinate (x,y,z).

If M is the Jacobian of the transformation from (x,y,z) coordinates to (ξ,η,ζ) coordinates then

$$M = \begin{vmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{vmatrix}$$

We further assume that the inverse transformation exists and is continuous. For this transformation, the Jacobian J is

$$J = \begin{vmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{vmatrix}$$

so that MJ=1.

Using the relationship between these two Jacobians, Eq. (8) can be inversely rewritten as

$$\begin{aligned} &\alpha_{11}x_{\xi\xi}+\alpha_{22}x_{\eta\eta}+\alpha_{33}x_{\zeta\zeta}+2\alpha_{12}x_{\xi\eta}+2\alpha_{13}x_{\xi\zeta}+2\alpha_{23}x_{\eta\zeta} \\ &\qquad +J^2(F1x_\xi+F2x_\eta+F3x_\zeta) = 0 \\ &\alpha_{11}y_{\xi\xi}+\alpha_{22}y_{\eta\eta}+\alpha_{33}y_{\zeta\zeta}+2\alpha_{12}y_{\xi\eta}+2\alpha_{13}y_{\xi\zeta}+2\alpha_{23}y_{\eta\zeta} \\ &\qquad +J^2(F1y_\xi+F2y_\eta+F3y_\zeta) = 0 \qquad (9) \\ &\alpha_{11}z_{\xi\xi}+\alpha_{22}z_{\eta\eta}+\alpha_{33}z_{\zeta\zeta}+2\alpha_{12}z_{\xi\eta}+2\alpha_{13}z_{\xi\zeta}+2\alpha_{23}z_{\eta\zeta} \\ &\qquad +J^2(F1z_\xi+F2z_\eta+F3z_\zeta) = 0 \end{aligned}$$

where $\alpha_{ij} = \sum_{k=1}^{3} \beta_{ki}\beta_{kj} \qquad 1 \leq i,j \leq 3$

is the transformation coefficient and

$$\beta_{11} = y_\eta z_\zeta - y_\zeta z_\eta, \quad \beta_{12} = y_\zeta z_\xi - y_\xi z_\zeta, \quad \beta_{13} = y_\xi z_\eta - y_\eta z_\xi$$
$$\beta_{21} = x_\zeta z_\eta - x_\eta z_\zeta, \quad \beta_{22} = x_\xi z_\zeta - x_\zeta z_\xi, \quad \beta_{23} = x_\eta z_\xi - x_\xi z_\eta$$
$$\beta_{31} = x_\eta y_\zeta - x_\zeta y_\eta, \quad \beta_{32} = x_\zeta y_\xi - x_\xi y_\zeta, \quad \beta_{33} = x_\xi y_\eta - x_\eta y_\xi$$

With suitable boundary conditions for the computational domain, Eq. (9) can then be solved by any stable numerical method to produce the coordinate relations between the physical and transformed domain. The detail of numerical procedures to obtain the body-fitted coordinates is further discussed in chapter IV and V.

After calculating the coordinate relationships, the governing equations (1) through (5) in the physical domain must also be transformed to the body-fitted coordinate system. The dimensionless continuity equation (1) in the transformed coordinate system becomes

$$\{J(\xi_x u+\xi_y v+\xi_z w)\}_\xi + \{J(\eta_x u+\eta_y v+\eta_z w)\}_\eta + \{J(\zeta_x u+\zeta_y v+\zeta_z w)\}_\zeta = 0 \qquad (10)$$

$$J = \begin{vmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{vmatrix}$$

where J is the Jacobian and the subscripts x, y, z, ... etc. mean the derivative with respect to x, y, z, ... etc.. It should be remarked that only the independent variables in the physical domain (x,y,z) are transformed, however the dependent variables u, v and w are not transformed since the problem can be solved on the transformed coordinates without the transformation of u, v and w. The momentum and turbulent transport equations (2), (4) and (5) can also be rewritten as

$$a^{\phi}\phi_t + b^{\phi}\phi_{\xi} + c^{\phi}\phi_{\eta} + d^{\phi}\phi_{\zeta} = e^{\phi}\phi_{\xi\xi} + f^{\phi}\phi_{\eta\eta} + g^{\phi}\phi_{\zeta\zeta} + h^{\phi} + i^{\phi} \tag{11}$$

where ϕ can be u, v, w, k and ε . The coefficients a^{ϕ}, b^{ϕ}, ... etc. in Eq. (11) respectively the coefficient for u, v, w, k and ε. They are listed in table 3. Again only the independent variables x, y and z are transformed to the body-fitted coordinates. The velocity components u, v and w are still the velocity components in the Cartesian x, y and z direction.

3.2 FA Formulation

The basic idea of FA method proposed by Chen et al. [29-37], is the incorporation of local analytic solution in the numerical solution of partial differential equations. In the finite analytic method, the whole region of the problem is divided into many small elements in which the governing equation is solved analytically. An algebraic equation which approximates the governing eeqation is then obtained when the analytic solution is evaluated in an interior node of

$$a^\phi \phi_t + b^\phi \phi_\xi + c^\phi \phi_\eta + d^\phi \phi_\zeta = e^\phi \phi_{\xi\xi} + f^\phi \phi_{\eta\eta} + g^\phi \phi_{\zeta\zeta} + h^\phi + i^\phi$$

ϕ	a^ϕ	b^ϕ	c^ϕ	d^ϕ	e^ϕ	f^ϕ	g^ϕ	h^ϕ	i^ϕ
u	Re	$\Omega_\xi - \xi_x \nu_{tx} - \nu_{eff} F1$	$\Omega_\eta - \eta_x \nu_{tx} - \nu_{eff} F2$	$\Omega_\zeta - \zeta_x \nu_{tx} - \nu_{eff} F3$	$\nu_{eff} A1$	$\nu_{eff} A2$	$\nu_{eff} A3$	$-RePx$	$-2/3\, Rek_x + 2\nu_{eff} [B1u_{\xi\eta} + B2u_{\eta\zeta} + B3u_{\xi\zeta}] + v_x \nu_{ty} + w_x \nu_{tz}$
v	Re	$\Omega_\xi - \xi_y \nu_{ty} - \nu_{eff} F1$	$\Omega_\eta - \eta_y \nu_{ty} - \nu_{eff} F2$	$\Omega_\zeta - \zeta_y \nu_{ty} - \nu_{eff} F3$	$\nu_{eff} A1$	$\nu_{eff} A2$	$\nu_{eff} A3$	$-RePy$	$-2/3\, Rek_y + 2\nu_{eff} [B1v_{\xi\eta} + B2v_{\eta\zeta} + B3v_{\xi\zeta}] + w_y \nu_{tz} + u_y \nu_{tx}$
w	Re	$\Omega_\xi - \xi_z \nu_{tz} - \nu_{eff} F1$	$\Omega_\eta - \eta_z \nu_{tz} - \nu_{eff} F2$	$\Omega_\zeta - \zeta_z \nu_{tz} - \nu_{eff} F3$	$\nu_{eff} A1$	$\nu_{eff} A2$	$\nu_{eff} A3$	$-RePz$	$-2/3\, Rek_z + 2\nu_{eff} [B1w_{\xi\eta} + B2w_{\eta\zeta} + B3w_{\xi\zeta}] + u_z \nu_{tx} + v_z \nu_{ty}$
k	Re	$\Omega_\xi - (1+\frac{\nu_t}{\sigma_k})F1$	$\Omega_\eta - (1+\frac{\nu_t}{\sigma_k})F2$	$\Omega_\zeta - (1+\frac{\nu_t}{\sigma_k})F3$	$(1+\frac{\nu_t}{\sigma_k})A1$	$(1+\frac{\nu_t}{\sigma_k})A2$	$(1+\frac{\nu_t}{\sigma_k})A3$		$2(1+\frac{\nu_t}{\sigma_k}) \cdot [B1k_{\xi\eta} + B2k_{\eta\zeta} + B3k_{\xi\zeta}] + (\nu_t G - \epsilon Re)$
ϵ	Re	$\Omega_\xi - (1+\frac{\nu_t}{\sigma_\epsilon})F1$	$\Omega_\eta - (1+\frac{\nu_t}{\sigma_\epsilon})F2$	$\Omega_\zeta - (1+\frac{\nu_t}{\sigma_\epsilon})F3$	$(1+\frac{\nu_t}{\sigma_\epsilon})A1$	$(1+\frac{\nu_t}{\sigma_\epsilon})A2$	$(1+\frac{\nu_t}{\sigma_\epsilon})A3$		$2(1+\frac{\nu_t}{\sigma_\epsilon}) \cdot [B1\epsilon_{\xi\eta} + B2\epsilon_{\eta\zeta} + B3\epsilon_{\xi\zeta}] + (C_{\epsilon 1}\nu_t G - C_{\epsilon 2}\epsilon Re)[\frac{1}{t}]$

$\Omega_\xi = Re\vec{V}\cdot\nabla\xi - \nabla(\frac{\nu_t}{\sigma})\cdot\nabla\xi$ where $\nabla = \frac{\partial}{\partial x}\hat{i} + \frac{\partial}{\partial y}\hat{j} + \frac{\partial}{\partial z}\hat{k}$, $\vec{V} = u\hat{i} + v\hat{j} + w\hat{k}$

$\Omega_\eta = Re\vec{V}\cdot\nabla\eta - \nabla(\frac{\nu_t}{\sigma})\cdot\nabla\eta$ $\quad \sigma = 1$ (for u, v, w) $\quad [\frac{1}{t}] = \frac{\epsilon}{k}$ for one-scale model; $\sqrt{\frac{\overline{\phi\epsilon}}{\phi\nu}}$ for two-scale model

$\Omega_\zeta = Re\vec{V}\cdot\nabla\zeta - \nabla(\frac{\nu_t}{\sigma})\nabla\zeta$ $\quad \sigma = \sigma_k$ (for k), $\sigma = \sigma_\epsilon$ for (ϵ)

$A1 = \nabla\xi\cdot\nabla\xi$, $A2 = \nabla\eta\cdot\nabla\eta$, $A3 = \nabla\zeta\cdot\nabla\zeta$, $B1 = \nabla\xi\cdot\nabla\eta$, $B2 = \nabla\eta\cdot\nabla\zeta$, $B3 = \nabla\xi\cdot\nabla\zeta$

$G = (\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})\frac{\partial u_i}{\partial x_j}$, $\quad \phi_x = \phi_\xi\xi_x + \phi_\eta\eta_x + \phi_\zeta\zeta_x$, $\quad \phi_y = \phi_\xi\xi_y + \phi_\eta\eta_y + \phi_\zeta\zeta_y$

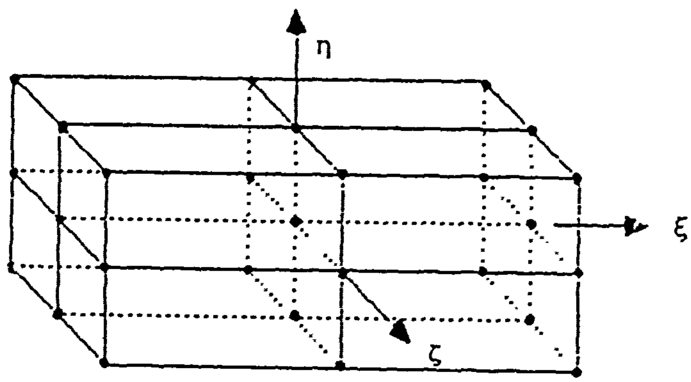
$\phi_z = \phi_\xi\xi_z + \phi_\eta\eta_z + \phi_\zeta\zeta_z$

Table 3

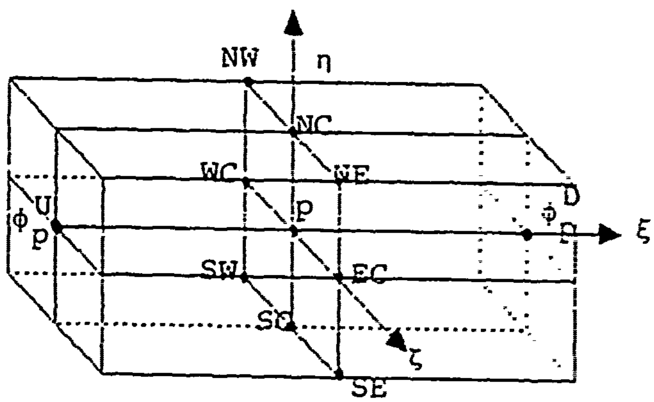
The Coefficients Of Momentum And Turbulent Transport Equations the element for numerical solution. The principle and procedures in obtaining these FA solutions are illustrated in detail in many published papers by Chen et al. [29-37]. Here we derive the finite analytic (FA) solution for a three dimensional unsteady flow. Detail of the FA coefficients which are used in this study are given in Appendix A.

Mathematically Eq. (11) shown in the last section is a fully three dimensional elliptic partial differential equation in space. An accurate and complete finite analytic numerical solution for Eq. (11) can be derived [34] based on the principle of the FA method to obtain an finite analytic algebraic equation based on 27-node FA element as shown in figure 4(a). However the finite analytic solution based on the 27-node element requires large storage and at the present it is beyond the computer capacity that is available for the user. In order that the problem of three dimensional flow can be solved with the limited facilities, the unsteady three dimensional elliptic partial differential equation (11) is solved by a hybrid finite analytic-finite difference method as follows. Eq. (11) is first cast into Eq. (12) where the derivatives of dependent variables with respect to time t and the axial direction are shifted into the source term s^{ϕ} as shown in Eq. (12).

$$c^{\phi}\phi_{\eta} + d^{\phi}\phi_{\zeta} = f^{\phi}\phi_{\eta\eta} + g^{\phi}\phi_{\zeta\zeta} + s^{\phi} \tag{12}$$



(a) 27 Spatial Nodes



(b) 11 Spatial Nodes

Figure 4. Finite Analytic Element

$$s^{\phi} = h^{\phi} + i^{\phi} + e^{\phi}\phi_{\xi\xi} - b^{\phi}\phi_{\xi} + a^{\phi}\phi_t$$

In the hybrid finite analytic and finite difference method the terms in the source term of Eq. (12) $\phi_{\xi\xi}$, ϕ_ξ and ϕ_t are approximately expressed by the finite difference such that ϕ_t by the implicit or backward difference, $\phi_{\xi\xi}$ central difference and ϕ_ξ by the upwind difference, or

$$\phi_t = \frac{\phi^{t+1} - \phi^{t}}{\Delta t}$$

$$\phi_{\xi\xi} = \frac{\phi_U + \phi_D - 2\phi_P}{(\Delta\xi)^2}$$

$$\phi_\xi = \frac{\phi_P - \phi_U}{\Delta\xi} \qquad (\text{if } b^{\phi} \geq 0)$$

$$\phi_\xi = \frac{\phi_D - \phi_P}{\Delta\xi} \qquad (\text{if } b^{\phi} < 0)$$

Allspace derivatives ϕ_ξ, $\phi_{\xi\xi}$ are evaluated from the previous time step.

If we introduce the coordinate-stretching functions

$$\eta^* = \frac{\eta}{\sqrt{f^{\phi}}}, \qquad \zeta^* = \frac{\zeta}{\sqrt{g^{\phi}}}$$

Eq. (12) is reduced to the standard two dimensional convective transport equation described in Chen & Chen [34,37], i.e.,

$$\phi_{\zeta^*\zeta^*} + \phi_{\eta^*\eta^*} = 2A\phi_{\zeta^*} + 2B\phi_{\eta^*} - s^{*\phi}$$

with

$$A = \frac{d^{\phi}}{2\sqrt{g^{\phi}}}, \qquad B = \frac{c^{\phi}}{2\sqrt{f^{\phi}}}$$

for a local element with dimensions

$$\Delta\xi = 1$$

$$\Delta\eta = k = \frac{1}{\sqrt{f^{\phi}}}$$

$$\Delta\zeta = h = \frac{1}{\sqrt{g^{\phi}}}$$

This hybrid FA-FD formulation gives the 11-point algebraic solution of Eq. (12) for three dimensional time dependent flow for an element as shown in figure 4(b) as

$$\phi_P = \frac{\sum_{nb=1}^{8} C_{nb}\phi_{nb} + C_P^{\phi}\{[(e^{\phi}+C_2e^{\phi})\phi_D + (e^{\phi}+C_3b^{\phi})\phi_U + h^{\phi} + i^{\phi}]^{i-1} + \frac{a^{\phi}\phi^{t-1}}{\Delta t}\}}{1 + C_P^{\phi}(C_1b^{\phi} + 2e^{\phi} + \frac{a^{\phi}}{\Delta t})} \qquad (13)$$

Here if $b^{\phi} > 0$, $C_1 = 1$, $C_2 = 0$, $C_3 = 1$

if $b^{\phi} < 0$, $C_1 = -1$, $C_2 = -1$, $C_3 = 0$

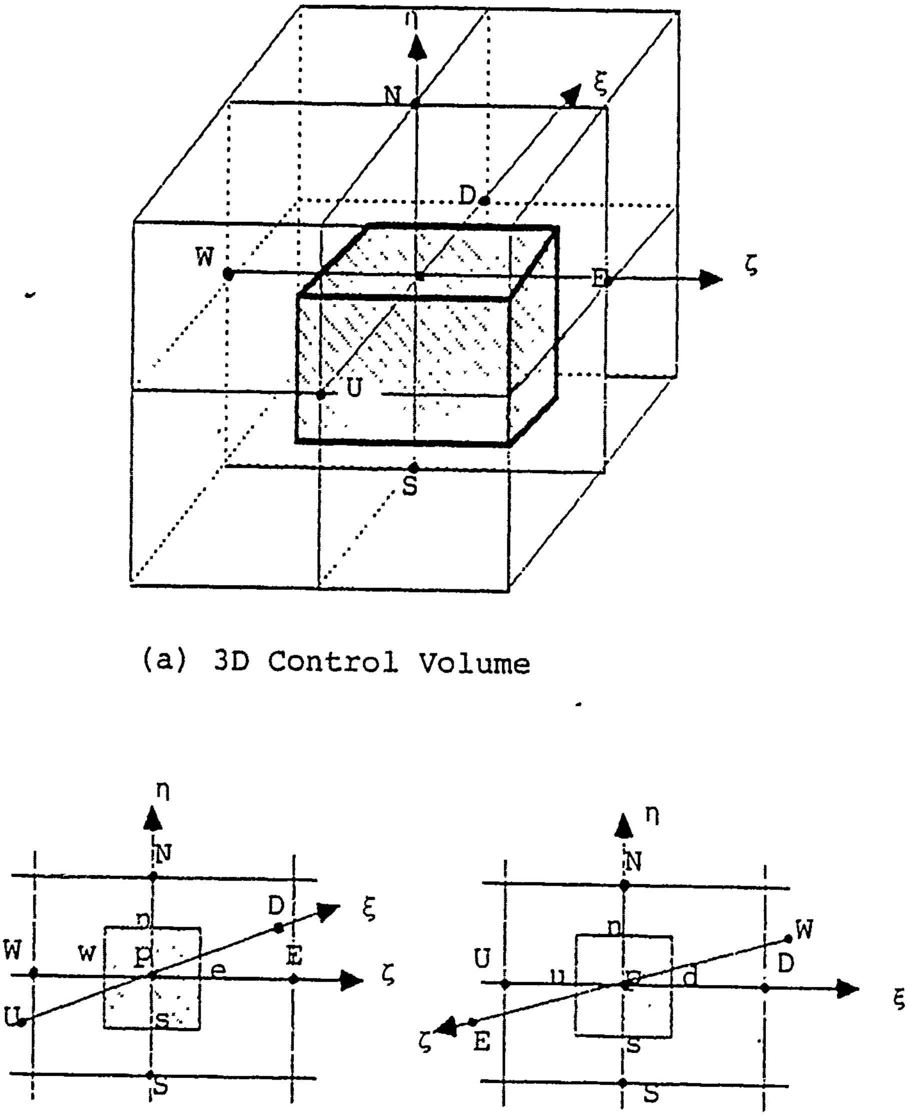
Figure 4(b) shows the relation between each of the 11 nodes. In Eq. (13) the superscript t-1 denotes the previous time step, and the term with the superscript i-1 means the value of previous iteration. On the same time step ϕ_U and ϕ_D are the values of node p at the upstream and downstream of the ξ coordinate and Δt is the time increment. The expressions of these FA coefficients C_{nb}, C_P are listed in Appendix A.

Physically, the above formulation preserves the three dimensional ellipticity and still allows the recirculation to exist. In Eq. (13) the calculation sweeps iteratively along the ξ direction. It should be remarked that since the numerical solution in the ξ direction is approximated by finite difference with only one upstream and one downstream node the prediction of the separation or the vortex formation in the ξ direction can only be regarded approximately. Therefore if one expects a flow problem which has strong recirculation in all three directions, then it is suggested that dense nodes must be arranged in the ξ axial direction or small spacing in the axial direction.

3.3 Pressure Equation

To complete the numerical solution, in addition to solving the finite analytic algebraic equations, i.e., Eq. (13), for variables u, v, w, k and ε, one more equation is needed for solving the unknown p. There are several ways to solve the pressure variable. For example, Roach [27] solved the pressure varible from the Poisson equation which is derived by taking the divergence of the momentum equation. In this approach, a velocity correction term is incorporated in the Poisson equation where velocity is corrected to satisfy the continuity equation. In the other methods, Patankar and Spalding et al. [23] proposed to use the continuity equation for the pressure variable. The basic idea of their approach is to express the velocity variable in the continuity equation in terms of pressure variables.

The equation for the pressure variables is obtained when the algebraic expressions for the velocity from the momentum equation are substituted into the continuity equation which is expressed either in a finite difference or finite volume expression. In the numerical procedure, the pressure variable is then updated in each numerical calculation such that the velocity components respectively solved from the momentum equation are made to satisfy the conservation of mass. Depending on the approximations made in updating pressure, different governing equations for pressure may be obtained. Among them, the pressure-update-Patankar (PUP) scheme [48] combined with Pantankar-Spalding p' equation (or called SIMPLER algorithm) gives the best result. Here p' is known as the pressure correction and defined as the difference between the true or exact pressure field and that of the approximate or incorrect pressure field. In the PUP scheme, instead of updating pressure gradually from the pressure correction p', a pseudovelocity, u_i, obtained by omitting the pressure gradient term in the momentum equation is introduced so that the pressure field can be obtained from a guessed velcoity field. The general procedure of SIMPLER (Semi-Implicit Method for Pressure-Linked Equations Revised) algorithm is adopted and modified in this study. Details of the derivation of pressure equation and the pressure correction equation are provided in the following sections.



(a) 3D Control Volume

(b) η-ζ Cross Section (c) ξ-η Cross Sction

Figure 5. Typical Control
Volume for Pressure Equation

Figure 5 shows a typical control volume (shaded area) that is used to derive the pressure equation based on mass conservation applied to the shaded element in the present study. It should be noted that the shaded element for the velocity variables has a smaller control volume ($\Delta\xi=\Delta\eta=\Delta\zeta=0.5$) in the transformed domain than that for the pressure variable. All the velocity components u, v and w are specified at the surface nodes e, w, n, s, u and d of the shaded control volume and are assumed known and stored. Then the pressure p is assigned at the surface nodes of unshaded control volume ($\Delta\xi=\Delta\eta=\Delta\zeta=1$) i.e., E, W, N, S, U and D. In order to replace the velocity variables in continuity equation Eq. (10) by pressure variable, we first decompose the actual velocity field (u, v, w) in the momentum equations Eq. (13) into two parts. They are

$$u = \hat{u} + D_u p_x, \quad v = \hat{v} + D_v p_y, \quad w = \hat{w} + D_w p_z \tag{14}$$

$$D_u = \frac{Re\, C_P^u}{D_P^u}, \quad D_v = \frac{Re\, C_P^v}{D_P^v}, \quad D_w = \frac{Re\, C_P^w}{D_P^w}$$

$$\hat{\phi} = \frac{\sum_{nb=1}^{8} C_{nb}\phi_{nb} + C_P^{\phi}\left\{\left[(e^{\phi}+C_2 e^{\phi})\phi_D + (e^{\phi}+C_3 b^{\phi})\phi_U + i^{\phi}\right]^{i-1} + \frac{a^{\phi}\phi^{t-1}}{\Delta t}\right\}}{D_P^{\phi}}$$

$\hat{\phi}$ for $\hat{u}$, $\hat{v}$, $\hat{w}$

where $$D_P^{\phi} = 1 + C_P^{\phi}\left(\frac{a^{\phi}}{\Delta t} + C_1 b^{\phi} + 2e^{\phi}\right)$$

Here all the notations have been described in Eq. (13). In Eq. (14) $\hat{u}$, $\hat{v}$, $\hat{w}$ are called pseudovelocities equal to the values that the velocities would have without the pressure contribution. Then substitution of Eq. (14) into the continuity equation Eq. (10) written in the pressure control volume i.e., the shaded control volume in figure 5, one obtains

$$(\bar{V}_\xi)_d - (\bar{V}_\xi)_u + (\bar{V}_\eta)_n - (\bar{V}_\eta)_s + (\bar{V}_\zeta)_e - (\bar{V}_\zeta)_w = 0 \qquad (15)$$

where the subscripts d, u, n, s, e and w denote downstream, upstream, north, south, east and west side of the shaded area. And

$$\bar{V}_\xi = \hat{V}_\xi + C_\xi p_\xi, \quad \bar{V}_\eta = \hat{V}_\eta + C_\eta p_\eta, \quad \bar{V}_\zeta = \hat{V}_\zeta + C_\zeta p_\zeta \qquad (16)$$

where

$$\hat{V}_\xi = J(\xi_x\hat{u}+\xi_y\hat{v}+\xi_z\hat{w}+D_{12}p_\eta+D_{13}p_\zeta)$$
$$\hat{V}_\eta = J(\eta_x\hat{u}+\eta_y\hat{v}+\eta_z\hat{w}+D_{12}p_\xi+D_{23}p_\eta)$$
$$\hat{V}_\zeta = J(\zeta_x\hat{u}+\zeta_y\hat{v}+\zeta_z\hat{w}+D_{13}p_\xi+D_{23}p_\eta)$$
$$D_{12} = D_u\xi_x\eta_x+D_v\xi_y\eta_y+D_w\xi_z\eta_z$$
$$D_{13} = D_u\xi_x\zeta_x+D_v\xi_y\zeta_y+D_w\xi_z\zeta_z$$
$$D_{23} = D_u\eta_x\zeta_x+D_v\eta_y\zeta_y+D_w\eta_z\zeta_z$$
$$C_\xi = J(D_u\xi_x^2+D_v\xi_y^2+D_w\xi_z^2)$$
$$C_\eta = J(D_u\eta_x^2+D_v\eta_y^2+D_w\eta_z^2)$$
$$C_\zeta = J(D_u\zeta_x^2+D_v\zeta_y^2+D_w\zeta_z^2)$$

Here C_ξ, C_η, C_ζ are the mean pressure-velocity linkage coefficients obtainable from the transformed momentum equation, and variables $\bar{V}_\xi$, $\bar{V}_\eta$, $\bar{V}_\zeta$ and $\hat{V}_\xi$, $\hat{V}_\eta$, $\hat{V}_\zeta$ are the velocities and pseudovelocities in the body fitted coordinates along the ξ, η and ζ directions. Similarly p_ξ, p_η and p_ζ respectively are the pressure gradients in the transformed domain. Using central difference for these pressure gradients, one may rewrite Eq. (15) and obtain the pressure equation as

$$a_P p_P = a_D p_D + a_U p_U + a_S p_S + a_N p_N + a_E p_E + a_W p_W + DS \qquad (17)$$

where

$$a_D = (C_\xi)_d,\quad a_U = (C_\xi)_u,\quad a_N = (C_\eta)_n,$$
$$a_S = (C_\eta)_s,\quad a_E = (C_\zeta)_e,\quad a_W = (C_\zeta)_w$$
$$a_P = a_D + a_U + a_N + a_S + a_E + a_W$$
$$DS = (\hat{V}_\xi)_d - (\hat{V}_\xi)_u + (\hat{V}_\eta)_n - (\hat{V}_\eta)_s + (\hat{V}_\zeta)_e - (\hat{V}_\zeta)_w$$

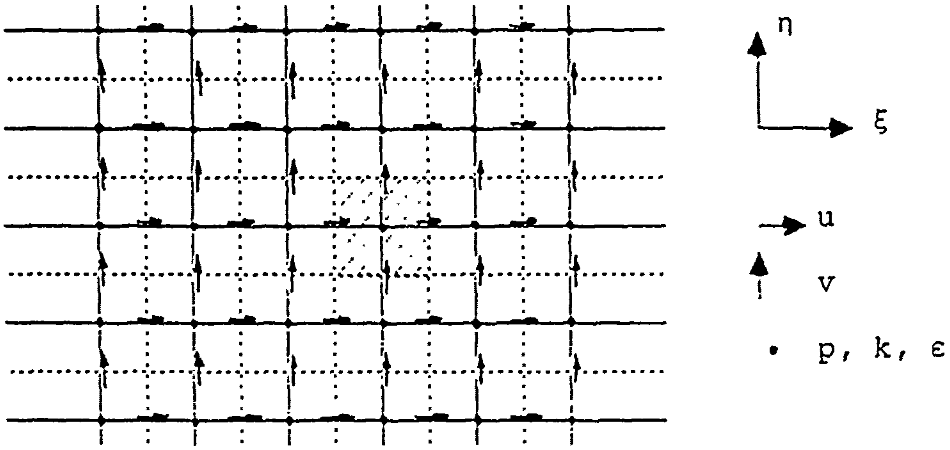
a_D, a_S, ... etc. are the coefficients of pressure equation (17).

In deriving the pressure equation Eq. (17) a proper choice of grid system is very important. There are two commonly used grid systems in the numerical calculation. One is the staggered grid system which distributes the variables at different nodes, the other is the regular grid system which solves all variables at the same node. In the following sections these two different grid systems will be investigated and discussed.

3.4 Staggered Grid System

Figure 6 is the general view of the staggered grid system [49]. If the dashed lines represent the control volume faces, the pressure and scalar variables such as k and are stored at the center of the control volume, while the velocity components are stored at midway between these nodes denoted with arrow "↑" for v component and "→" for u component. Here the velocity components are perpendicular to the control surfaces or in the direction of the coordinate lines.

In this study all equations are transformed and solved on the transformed domain (ξ, η, ζ), where the coordinate lines ξ, η, ζ in general are curvilinear and non-orthogonal in the physical (x,y,z) domain. The velocity components u, v and w that are defined in the x, y and z direction are neither perpendicular to the control surfaces nor in the direction of the coordinate lines ξ, η and ζ. Therefore the velocities in the ξ, η, ζ directions. must be projected from the velocity components u, v, w defined in the x, y and z directions. Consequently examining the source term of pressure equation Eq. (17), one finds a total of eighteen velocity components are needed, three velocity variables in



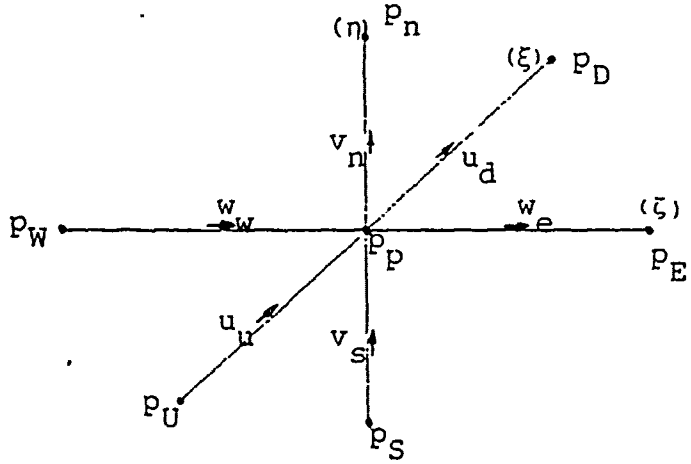


Figure 6. Typical Control Volume In Staggered Grid System

each of the six surfaces, in each small control volume as shown in figure 5. The projection of u, v, w velocities in the x, y and z coordinates to the ξ, η and ζ coordinates certainly provides numerical error either from interpolation or difference approximation. However, in the staggered grid system as shown in figure 6 the velocity component u at node points d and u, velocity component v at node points n and s, and velocity component w at node points e and w had been solved from the momentum equation or Eq. (13), thus six of eighteen velocity components, namely, u_d, u_u, v_n, v_s, w_e and w_w can be obtained directly from the surfaces. It is, therefore, only to approximate the remaining twelve by interpolation or difference approximation.

One way to reduce the numerical error is to reduce the use of interpolation or difference approximation. This can be achieved by letting one of the transformed coordinate lines, say coordinate to be just a function of x only. In this way the velocity component u is perpendicular to the η-ζ section, no other velocity components are needed in ξ direction. In other words the velocity component u in the x direction is identical to that in the ξ direction. Therefore in the source term of Eq. (17) only eight velocity components are still unknown on the η-ζ section and require interpolation. The coordinate arrangement of letting $\xi=\xi(x)$ is reasonable since in the present investigation of the flow past an axisymmetric body most of the experimental measurements are made along the section normal to the x axis of the body, that is on the y-z or η-ζ plane. Further more if the flow past the body is predominately along the x direction then the magnitude of the velocities v and w are in general smaller than that of the u velocity. Therefore, a simple linear interpolation can be used here to evaluate these values of v and w components from the velocity field known at the previous time step or iteration without causing too much error. In summary, under the present arrangement the source term of pressure equation Eq. (17) on the staggered grid system only needs the following eight approximations.

$$u_e = \frac{1}{4}\sum_1^4 u_{nb}, \quad v_e = \frac{1}{4}\sum_1^4 v_{nb}, \quad u_w = \frac{1}{4}\sum_1^4 u_{nb}, \quad v_w = \frac{1}{4}\sum_1^4 v_{nb},$$
$$u_n = \frac{1}{4}\sum_1^4 u_{nb}, \quad w_n = \frac{1}{4}\sum_1^4 w_{nb}, \quad u_s = \frac{1}{4}\sum_1^4 u_{nb}, \quad w_s = \frac{1}{4}\sum_1^4 w_{nb}$$

Where nb denotes the known neighbor nodes surrounding the unknown surface node e, w, n and s.

In order to impliment the arrangement of a staggered grid in computer programing it requires not only a large computer storage but also tedious work. As an alternative the regular grid system which solves all variables at the same node maybe used in the present computation. It is discussed in the next section.

3.5 Regular Grid System

Figure 7 shows a typical control volume in the regular grid system. Since all variables u, v, w, p, k and ε are stored and calculated at the same node in the unit control volume there are no velocity components at the surfaces of the small control volume shown as the shaded area in figure 7 for the pressure equation Eq. (17). It is, therefore, necessary to approximate all velocity components by interpolations. The interpolations are

$$
\begin{aligned}
(\hat{V}_\xi)_d &= \tfrac{1}{2}[(\hat{V}_\xi)_D+(\hat{V}_\xi)_P], & (\hat{V}_\xi)_u &= \tfrac{1}{2}[(\hat{V}_\xi)_U+(\hat{V}_\xi)_P], \\
(\hat{V}_\eta)_n &= \tfrac{1}{2}[(\hat{V}_\eta)_N+(\hat{V}_\eta)_P], & (\hat{V}_\eta)_s &= \tfrac{1}{2}[(\hat{V}_\eta)_S+(\hat{V}_\eta)_P], \\
(\hat{V}_\zeta)_e &= \tfrac{1}{2}[(\hat{V}_\zeta)_E+(\hat{V}_\zeta)_P], & (\hat{V}_\zeta)_w &= \tfrac{1}{2}[(\hat{V}_\zeta)_W+(\hat{V}_\zeta)_P] \\
(C_\xi)_d &= \tfrac{1}{2}[(C_\xi)_D+(C_\xi)_P], & (C_\xi)_u &= \tfrac{1}{2}[(C_\xi)_U+(C_\xi)_P], \\
(C_\eta)_n &= \tfrac{1}{2}[(C_\eta)_N+(C_\eta)_P], & (C_\eta)_s &= \tfrac{1}{2}[(C_\eta)_S+(C_\eta)_P], \\
(C_\zeta)_e &= \tfrac{1}{2}[(C_\zeta)_E+(C_\zeta)_P], & (C_\zeta)_w &= \tfrac{1}{2}[(C_\zeta)_W+(C_\zeta)_P]
\end{aligned}
$$

where all the notations have been described in Eq. (15).

Although the regular grid system may commit interpolation error, the use of the regular grid system when compared with the staggered grid system can save

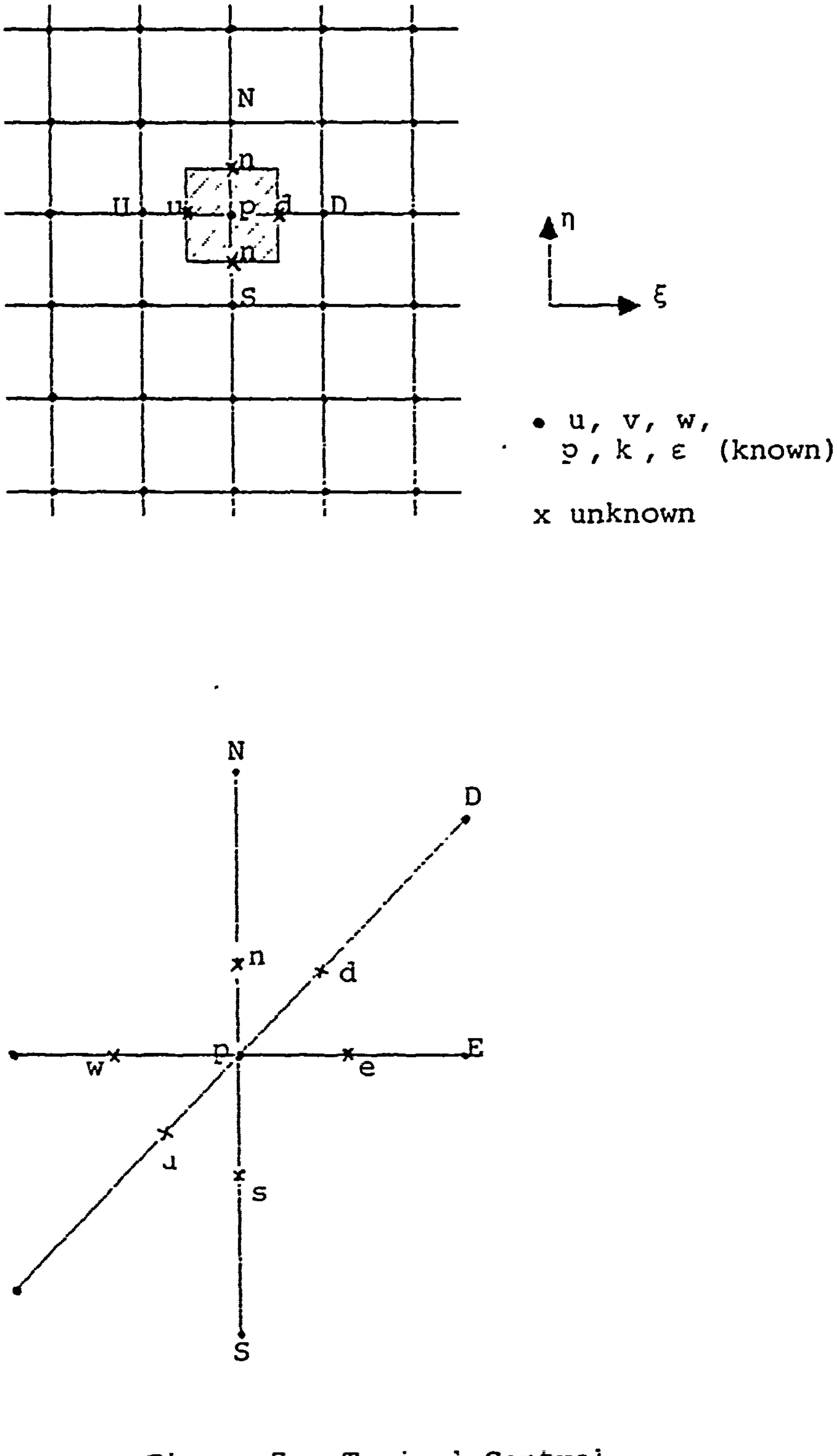


Figure 7. Typical Control
Volume In Regular Grid System

computational time and storage. Therefore in the present study the regular grid system is used for laminar flow over a finite flat plate with or without angle of attack.

3.6 Pressure Correction Equation

The governing equations formulated in chapter 2 are Eqs. (1) to (5). These equations are recasted into Eqs. (10) and (11) in the transformed domain. We are thus required to solve Eq. (11) for u, v, w, k and ε and Eq. (10) for p. The corresponding algebraic equations for Eqs (11) and (10) are Eqs. (13) and (17). The system of these nonlinear equations are solved iteratively in the present study. In this section we derive a scheme to ensure that the iterative procedure leads to a converged solution. Before we derive the pressure correction equation it should be noted that either with staggered grid or regular grid systems when both the momentum and continuity equations Eqs. (13) and (17) are exactly satisfied the value of DS on the right hand side of pressure equation Eq. (17) will be zero. However during the iterations because momentum and mass are not conserved in the volume element there exists some error in u, v, w and therefore DS in Eq. (17) is nonzero. The pressure correction equation is derived to improve the convergence of the solution. The following are the derivation and steps considered in this study for solving Eqs. (13) and (17) iteratively.

During iterations we first compute the velocity field from Eq. (13) with a guessed pressure field p*. With the guessed pressure field p* given, one can solve the momentum equation Eq. (13) directly and obtain the guessed velocity field u_i^*. At the beginning since p* is not a correct solution so that u_i^* do not satisfy the continuity equation. Then one may assume the velocity correction u_i' which when added to u_i^* will satisfy the continuity equation Eq. (10). Thus substituting u_i ($=u_i^*+u_i'$) into continuity equation Eq. (10), one obtains

$$
\begin{aligned}
&[J(\xi_x u'+\xi_y v'+\xi_z w')]_\xi + [J(\eta_x u'+\eta_y v'+\eta_z w')]_\eta \\
&\qquad + [J(\zeta_x u'+\zeta_y v'+\zeta_z w')]_\zeta = -ERR \qquad (18) \\
&ERR = [J(\xi_x u^*+\xi_y v^*+\xi_z w^*)]_\xi + [J(\eta_x u^*+\eta_y v^*+\eta_z w^*)]_\eta \\
&\qquad [J(\zeta_x u^*+\zeta_y v^*+\zeta_z w^*)]_\zeta
\end{aligned}
$$

Since there are three velocity corrections (u', v', w') in one equation, it is impossible to solve Eq. (18) directly. Alternatively, one may assumes that p is the correct pressure field to produce correct velocity u_i then the correct pressure p may be written as the sum of the pressure correction p' and the inaccurate pressure p* or p=p*+p'. Substituting both velocity and pressure expressions into h^ϕ term of the momentum equation Eq. (13), one obtains the equations for velocity correction u_i' and pressure correction p'. They are

$$u' = \frac{\sum_1^8 C_{nb} u'_{nb}}{D_P^u} + \frac{C_P^u}{D_P^u} s'_u + D_u p'_x \tag{19}$$

$$v' = \frac{\sum_1^8 C_{nb} v'_{nb}}{D_P^v} + \frac{C_P^v}{D_P^v} s'_v + D_v p'_y$$

$$w' = \frac{\sum_1^8 C_{nb} w'_{nb}}{D_P^w} + \frac{C_P^w}{D_P^w} s'_w + D_w p'_z$$

If the first two terms of equation (19) were retained, they would have to be expressed in terms of the pressure corrections and the velocity corrections at the neighbors of u_i'. These neighbors would, in turn, bring their neighbors, and so on. Ultimately, the velocity correction formula would involve the pressure correction at all grid points in the calculation domain, and the resulting pressure correction equation would become unmanageable. Therefore in the present study the first two terms of equation (19) will be neglected. Eq.(19) is simplized to

$$u' = D_u p'_x, \quad v' = D_v p'_y, \quad w' = D_w p'_z \tag{20}$$

The omission of the first two terms enables us to cast the relation between velocity correction and pressure correction in a much simpler form. The omission of the first two terms in Eq. (19) can be justified since if p' is zero u_i' will be zero too which give the right asymptotic behavior for convergence. In fact the converged solution should not be influenced by any approximations made in correction equations during iterations. In other words, all formulations of the correction equations should give the same final solution if the formulation leads to a converged solution. However the rate of convergence of the solution will depend on the particular formulation of the correction equations used. If a too simplistic formulation is used, divergence may result.

From the above formulation it is clear that if the pressure correction p' can be solved then the guessed velocity u_i^* can be approximately corrected by the velocity correction u_i' to satisfy the continuity equation Eq. (10). To derive an equation for the pressure correction p', the same procedure from Eq. (14) to Eq. (17) can be followed by dividing the velocity field u_i into guessed velocity u_i^* and velocity correction u_i' which is expressed by Eq. (20), i.e. $u=u^*+D_u p'$, $v=v^*+D_v p'$, $w=w^*+D_w p'$. Substituting these expressions into the continuity equation Eq. (10), one had the pressure correction equation which is similar to the pressure equation Eq.(17).

$$a_P p'_P = a_D p'_D + a_U p'_U + a_S p'_S + a_N p'_N + a_E p'_E + a_W p'_W + DS' \qquad (21)$$

Here a_D, a_U, ... etc. are as the same as Eq. (17) and DS' is same as DS in Eq. (17) except that the values of $\hat{u}$, $\hat{v}$, $\hat{w}$, and p* in the DS are replaced by u*, v*, w*, and p'.

After using pressure correction p' to correct the velocity field to satisfy continuity equation, the next step is to update the pressure field by solving pressure equation Eq. (17) with the velocity u_i which had just been corrected. Although the velocity field ($u_i=u_i^*+u_i'$) had been corrected to satisfy continuity equation they may not satisfy momentum equation unless the velocity correction u_i' are zero, i.e. $u_i=u_i^*$ and u_i^* satisfied the momentum equation already. Therefore, in order to have the solutions of p and u_i that both satisfy the momentum and continuity equations simultaneously, we need an iteration procedure to ensure the convergence of the solution. From Eq. (20), it shows that if the pressure correction p' is zero then the convergent solutions of p and u_i will satisfy both the momentum and continuity equqtions simultaneously. Thus, the convergent criterion in this study is based on the value of the pressure correction p' that tends to be zero. Generally, if the value of pressure correction p' is smaller than one percent of value of pressure p the solution is considered as a convergent solution.

3.7 Algorithm

Accuracy and efficiency are two major considerations in designing the algorithm of a numerical program. In this study a modification to SIMPLER algorithm [23] is made so that it is more efficient in computational time and storage and more accurate in compucational results. The overall numerical procedure for a three dimensional case used in this study may be summarized as follows.

1. Start the inlet (or present) station with the guessed pressure p* and velocity distribution u_i^*.
2. Calculate FA coefficients from Eq. (12) with the guessed velocity u_i^* or best velcoty available u_i then solve starred-velocity u_i^* from Eq. (13) with the guessed pressure p*.
3. Calculate pressure correction p' from Eq. (21) with the starred-velocity u_i^* in DS'.
4. Calculate velocity correction u_i' from Eq. (20) with the pressure correction p'.
5. Obtain the correct velocity u_i by combining the starred-velocity u_i^* and the velocity correction u_i' for the present iteration.
6. Calculate the pseudovelocity $\hat{u}_i$ as defined in Eq. (14) with the correct velocity u_i.
7. If it is turbulent flow solve k and ε from Eq. (13) with the correct velocity u_i.
8. Repeat from step 2 to step 7 until the last station was reached. This repeatition is called the inner loop.

9. Calculate pressure field p from Eq. (17) with the pseudovelocity $\hat{u}_i$ in the whole computational domain based on the correct velocity u_i. The resulting pressure field is considered as the updated pressure p*.
10. Start from the inlet station in step 1 with the update pressure field p and correct velocity u_i. This part is called the outer loop.
11. Stop if the steady state criterion is achievbed, or the time t exceeds the maximum time period assigned.

It should be remarked here that the line by line tridiagonal scheme is adopted to solve pressure equation Eq. (17) and pressure correction equation, Eq. (20), while the modified strongly implicit MSI procedure [50], which uses lower and upper triangle matrices to solve 9-point difference scheme at the same time, is adopted to solve Eq. (13) for other variables u, v, w, k and ε.

3.8 FANS-3DEF Program

In the present study a computer program called FANS-3DEF is developed. FANS-3DEF (Finite Analytic Numerical Solution of Three Dimensional External Flow) consists of a preprocessor and a main solver. This program includes options for (1) two or three dimensional flow, (2) staggered or regular grid system, (3) incompressible laminar or turbulent flow and (4) two types of turbulence model. It is compiled by FORTRAN 77 compiler, and has been implemented and tested on PRIME 750 at the University of Iowa. In this section a brief introduction of this program is given. The detailed discussion of the whole program, the flow chart of main program, I/O system and two examples of how to control I/O system will be given in appendix B.

The main structure of a general program should contain (1) data input module (preprocessor) (2) analysis and solution (solver) (3) output module (postprocessor). This is illustrated in the following figure.

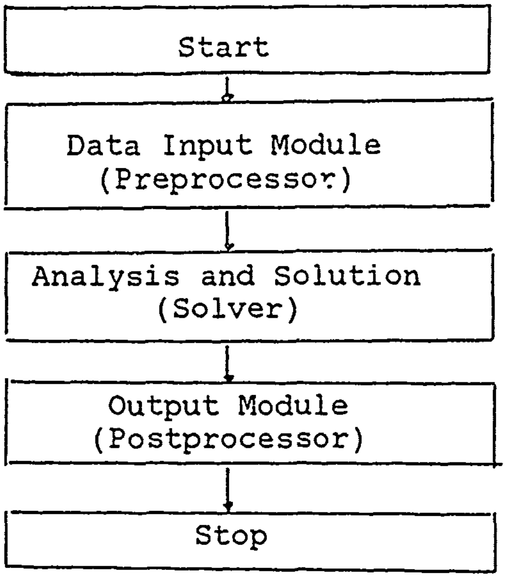


Figure 8. Simplified
Schematic Of A General Program

In the present FANS-3DEF program the output module (postprocessor) is not included. This is partly because at the present the graphic package is highly hardware orinted and partly because there are many professional graphic packages readily available. For example, at the University of Iowa a graphic package called 'GCS' is available and can be adopted as the output module.

In the FANS-3DEF program before the solver can be activated to solve the problem, sufficient information must be transmitted by the user to the data input module (preprocessor). This input system is described in detail in appendix B. put is completed then may initiate the problem the main program of FANS-3DEF. The computer program FANS-3DEF has been employed to calculate a variety of two-dimensional, axisymmetric and three-dimensional flows. In the next three chapters some representative examples and solutions are given to illustrate the capability of the numerical method used in this study. Suggestions for future applications are also given.

CHAPTER IV

TWO DIMENSIONAL FLOW OVER A FINITE FLAT PLATE

In this chapter the flow over a finite flat plate is considered. This is an important and fundamental external flow involving the development of the boundary layer flow on the plate and the evolution of the wake behind the plate. Although the geometry is simple in this case, and Cartesian coordinates can be used to solve the flow directly in the physical plane, the body-fitted coordinates in the FANS-3DEF (Finite Analytic Numerical Solution of Three Dimensional External Flow) are still used in order to verify the technique and program of the grid-generation. The numerical solution of this case can provide a useful test of the numerical method and the modified SIMPLER solution procedure for computing pressure and velocities during the iteration and a verification of turbulence models. In the following sections the solutions for both laminar and turbulent flows over a finit flat plate are given. The solution of laminar flow is first examined to verify the numerical algorithm and numerical scheme used in the FANS-3DEF. The solution of turbulent flow is then considered. The turbulent solution may be used to examine the predictability of the turbulence models.

4.1 Numerical Grid System

Figure 9 shows the computational domain of the finite flat plate. If the Cartesian coordinate (x,y) in the physical plane is chosen with distances x and y normalized by the plate length L, then x=0 and 1 correspond to the leading and trailing edges respectively, and y is the normal distance to the plate. Since the solution of variables u, v, w, k and ε vary rapidly in the neighborhood of the leading and trailing edges than other places, more grids are needed around these two regions. A desired grid distribution can be arranged by stretching and condensing the grids along the x, y coordinates in the physical plane. In this study a nonuniform rectangular grid is generated using the body-fitted coordinate technique as described in chapter III. With $x=x(\xi)$, $y=y(\eta)$, Eq. (9) is simplified to

$$\begin{aligned} \alpha_{11}x_{\xi\xi} + J^2F1x_{\xi} &= 0 \\ \alpha_{22}y_{\eta\eta} + J^2F2y_{\eta} &= 0 \end{aligned} \tag{22}$$

where

$$\alpha_{11} = y_\eta^2, \quad \alpha_{22} = x_\xi^2, \quad J = x_\xi y_\eta$$

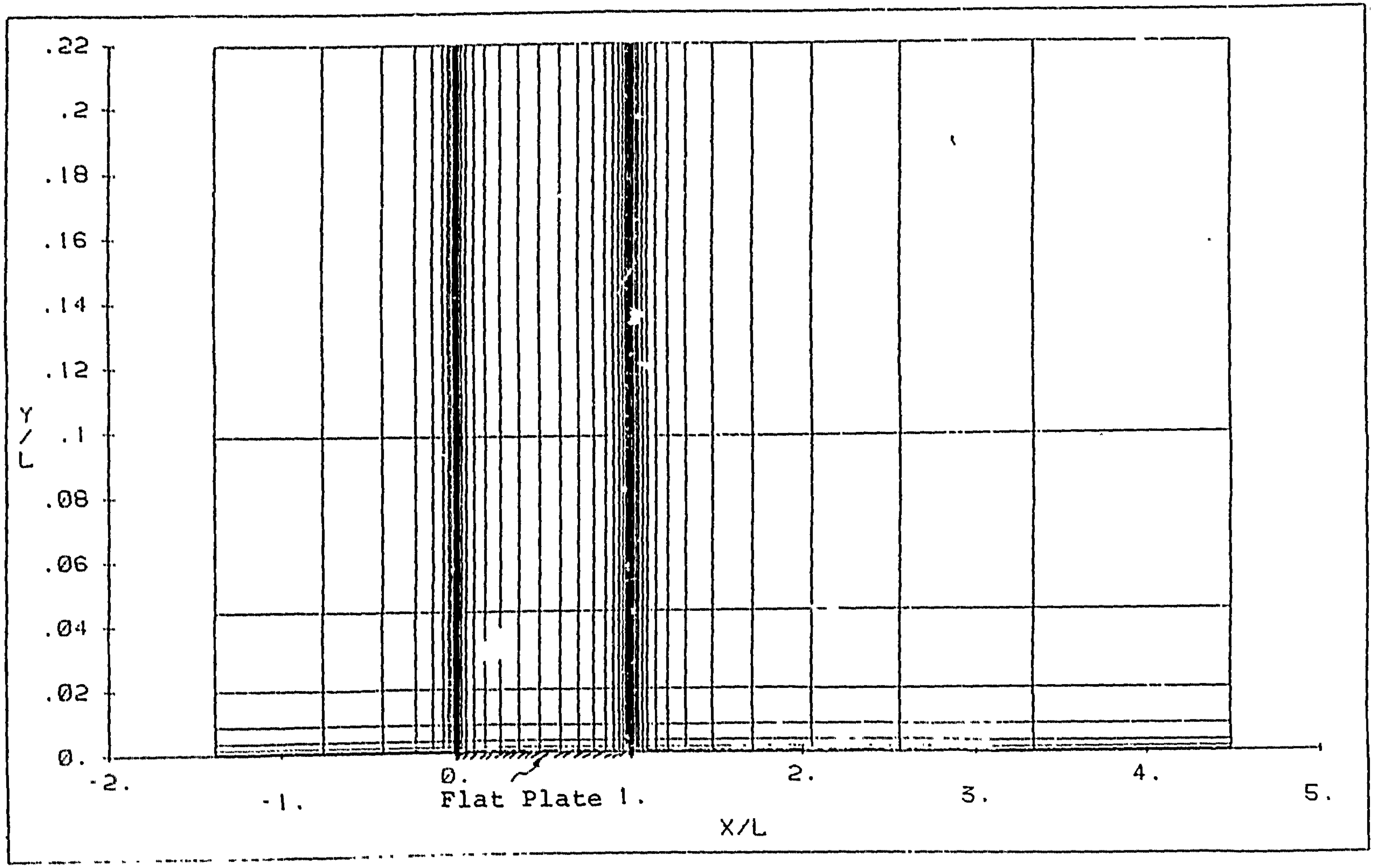


Figure 9. The Numerical Grid For Laminar Flow Over A Flat Plate

If the control functions F1 and F2 are prescribed a priori, then Eq. (22) can be solved for the coordinate variables (x,y) as a function of the uniform body fitted coordinstes (ξ,η). In analysing Eq. (22) one may choose the control functions F1 and F2 to remain constant within each numerical cell, thus Eq. (22) is solved analytically with $x(1,0)=x_D$, $x(-1,0)=x_U$, $y(0,1)=y_N$ and $y(0,-1)=y_S$.

$$x_P = \frac{e^{a}x_U + e^{-a}x_D}{e^{a} + e^{-a}}$$

$$y_P = \frac{e^{b}y_S + e^{-b}y_N}{e^{b} + e^{-b}} \tag{23}$$

where

$$a = -\frac{J^2 F1}{2\alpha_{11}}, \quad b = -\frac{J^2 F2}{2\alpha_{22}}$$

The subscript P, D, U, N and S denote the node at center, downstream, upstream, north and south of the numerical cell as shown in figure 10. Therefore for every nodal location there is one equation (23) to govern the transformation. For the computational domain as shown in figure 9 there is a set of simultaneous algebraic equations of Eq. (23) which can be solved easily by the tridiagonal algorithm if the appropriate boundary conditions for the computational domain and the flat plate are provided.

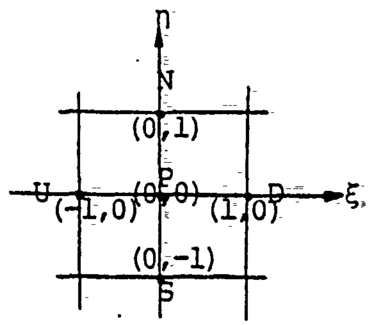


Figure 10. The Numerical Cell

In this study the distribution of a and b used to generate the grid nodes along the ξ and η directions was the one suggested by Chen and Patel [22]. They are

$$a = \begin{cases} -A1 & 0 \leqslant z_1 \leqslant \frac{1}{2} \\ A1\ \mathrm{Sin}(\pi z_1) & \frac{1}{2} < z_1 \leqslant 2 \\ A2\ \mathrm{Sin}(\pi z_2) & 0 \leqslant z_2 \leqslant \frac{3}{2} \\ -A2 & z_2 > \frac{3}{2} \end{cases}$$

where

$$z_1 = \frac{\xi - 1}{\xi_1 - 1}, \qquad z_2 = \frac{\xi - 2\xi_1 + 1}{\xi_2 - 2\xi_1 + 1}$$

ξ_1 and ξ_2 correspond to the leading and trailing edges respectively, and A1, A2 and A3 are positive constants which can be selected to yield the desired grid concentration around x=0, 1 and y=0.

4.2 Laminar Flow Without Angle Of Attack

To compare the results of this study with those shown by Chen and Patel [22], same values of A1, A2 and A3 are given. They are A1=0.3, A2=0.2, A3=0.4 and ξ_1=19 at x=0, ξ_2=49 at x=1, η=1 at y=0 and η=10 at y=0.2196. Thus, a 65x10 mesh was used to cover the physical region that extends from a distance 1.385L upstream of the leading edge to 3.488L downstream of the trailing edge, and 0.2196L normal to the plate, with the grid concentrated in the neighborhood of the leading and trailing edges and the plate. Figure 9 shows the numerical grid in the whole computational domain.

In order to compare with some previous studies [22,51,52], the Reynolds number Re=10^5 is chosen for the calculation of laminar flow over the flat plate without angle of attack. In this study the regular grid system discussed in section 3.5 is used and the incompressible laminar Navier-Stokes equations Eqs. (1) and (2) with $\overline{u_i u_j}=0$ are solved. The uniform velocity with zero pressure was specified at the upstream station x=-1.385L. Symmetric boundary condition at y=0 and free stream boundary condition outside the computational domain at y=0.2196L are prescribed as discussed in chapter II. The FANS-3DEF is then used to solve this problem in which Eqs. (1) and (2) are expressed in the algebraic form on the body-fitted coordinates as Eqs. (13) and (17). The solutions for Eqs. (13) and (17) are obtained with the time marching procedure. Since this is a steady state flow problem a large time increment can be used, here $\Delta t=1$ is used.

Figure 11 shows the history of the dimensionless pressure distribution $(p=(p-p_{ambient})/\rho U_0^2)$ before the plate, on the plate and along the wake centerline calculated at different time steps. The flow starts with an initially zero pressure throughout and uniform velocity u=1, v=0. It is seen that the solution reaches the steady state first around the leading edge in 10 time steps, while the flow in the wake area is still in the transient change and becomes steady after 30 time steps. Figure 12 compares the pressure distribution around the trailing edge predicted by the FANS-3DEF with other previous studies. It is seen that the present analysis predicts a pressure distribution between that calculated by Chen and Patel [22] also that of Saint-Victor and Cousteix [51], and that calculated by Rubin and Reddy [52]. It should be remarked that the present solution is obtained from the elliptic solution by specifying the upstream condition at x=-1.385L with uniform free stream, and the downstream condition at x=4.488L with vanishing second derivatives. The previous studies [22,51,52] were based on partially parabolic solution

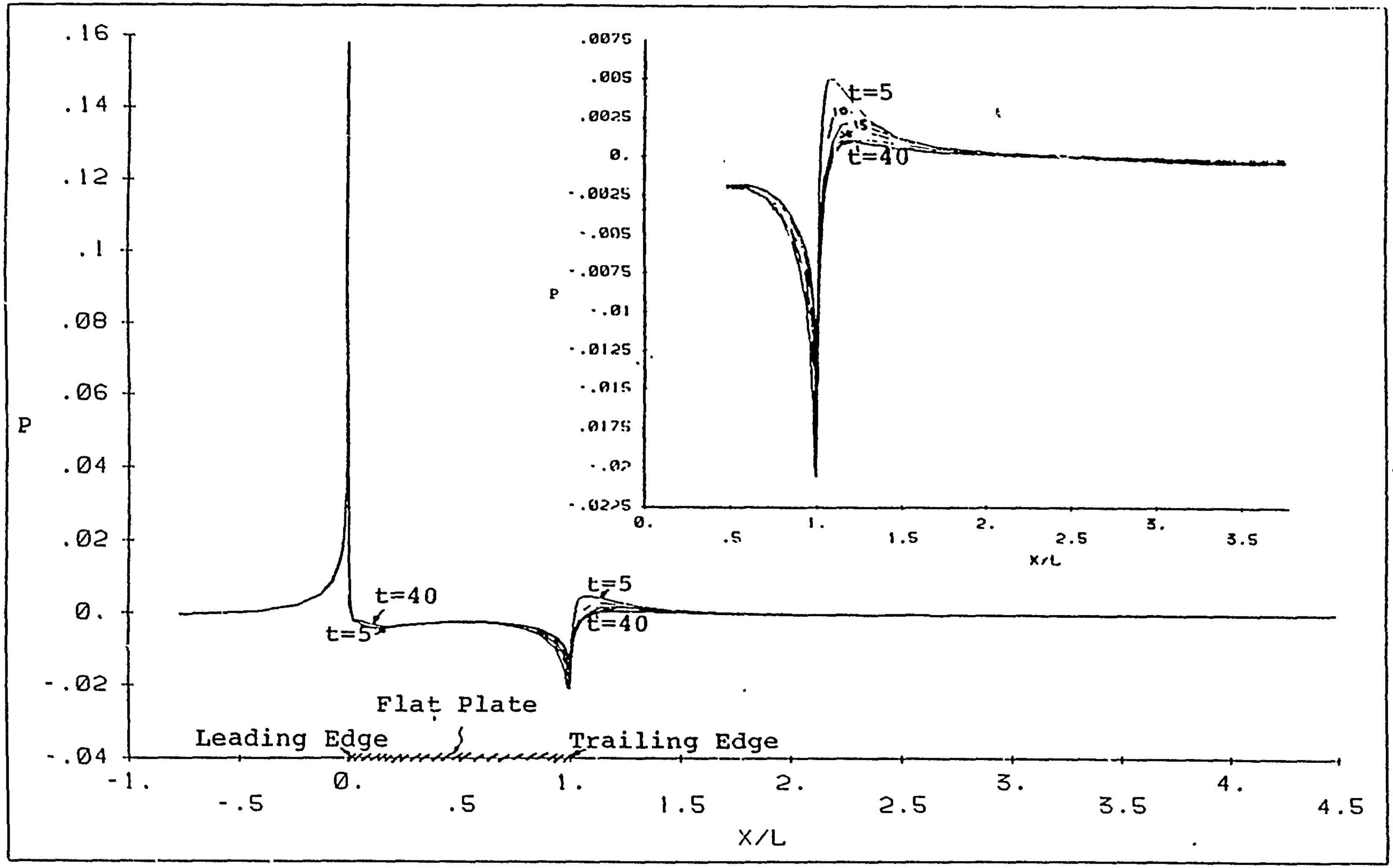


Figure 11. Convergence History Of The Pressure Distribution

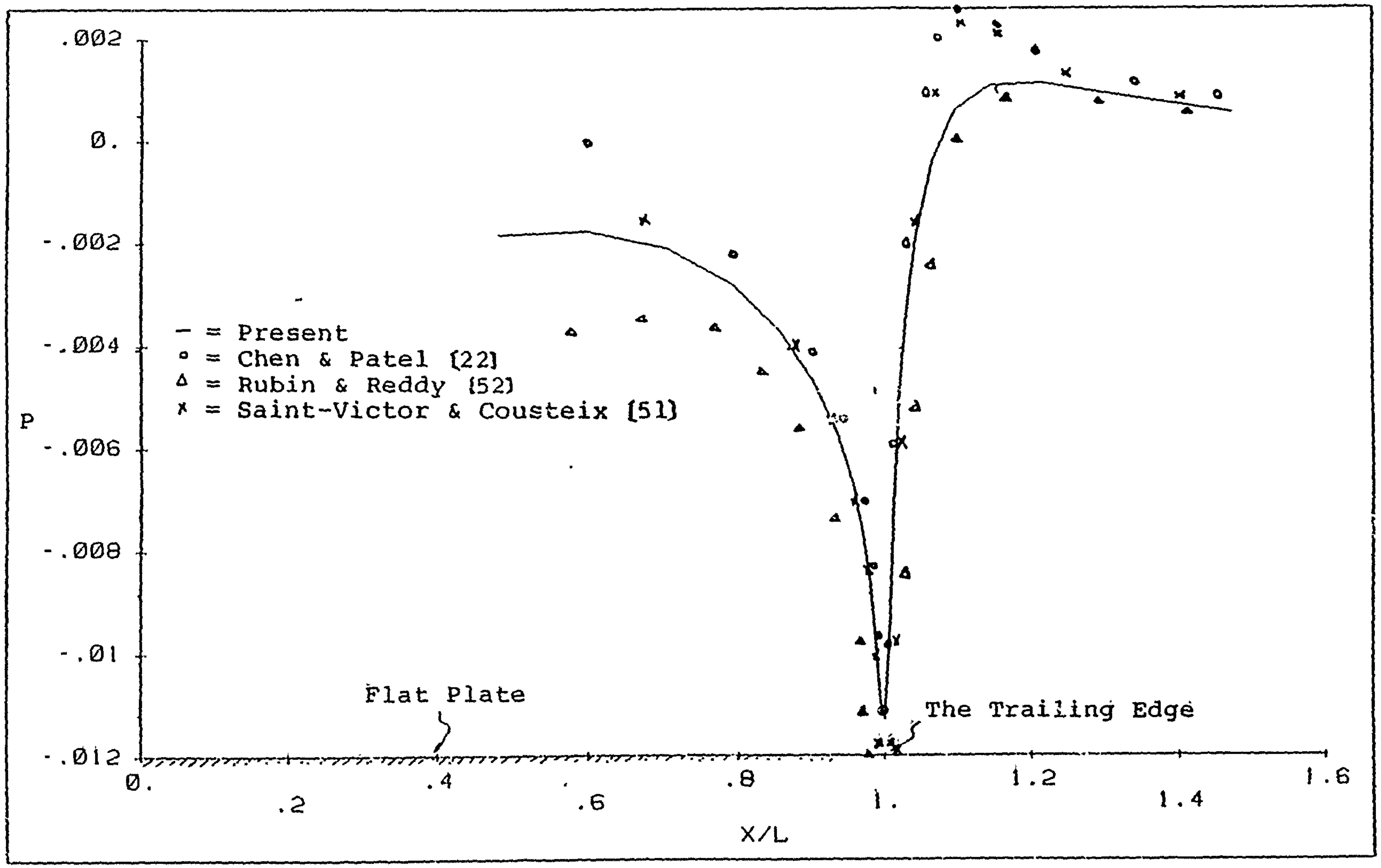


Figure 12. Pressure Distribution Around The Trailing Edge

specifying the upstream condition on the flat plate behind the leading edge. Rubin and Reddy used the triple-deck solutions and obtained the solution with the Blasius profile imposed at a large distance upstream of the trailing edge on the plate (x=0.1L), whereas the other two methods [22] and [51] specified the initial conditions on the plate at x=0.5L with Blasius solution. Chen and Patel [22] had investigated the solution with the initial conditions specified closer to the leading edge (typically x=0.18L) and found that the pressure distribution along the flat plate will be lower and hence closer to the present solution. They also commented on the influence of the location of the outer boundary of the solution domain on the upstream pressure distribution. In the present study the Navier-Stokes equations are solved by FANS-3DEF. The pressure distribution around the leading edge is predicted and shown that the peak value of the pressure at the leading edge is about 0.156 which is not a negligible value. This is in contrary to the boundary layer approximation that assumes the pressure is uniform even at the leading edge. The prediction of non zero pressure at the leading edge is physically sound, since the flow is decelerated from a uniform velocity distribution before reaching the plate to zero velocity on the plate surface. The pressure at the leading edge is thus expected to increase from this velocity deceleration. Once the flow past the leading edge, the magnitude of velocity deceleration is then gradually reduced. Consequently the pressure drops to almost that of the free stream value again.

The wall skin coefficient ($C_\tau=\sqrt{Re}\tau_w/\rho U_0^2$) shown in figure 13 indicates that there is a feed back from the downstream since the skin friction is increased near the trailing edge. Physically this is due to flow near the trailing edge and is influenced by the acceleration of the flow in the wake because the flow is no longer hold to the no-slip zero velocity at the center line. When the flow at the center line is accelerated, the velocity gradient normal to the plate near the trailing edge is increased and hence the skin friction. At $Re=10^5$ it seems that the flow from $x=0.8L$ to the trailing edge are affected by the wake flow. The prediction of skin friction by the FANS-3DEF in the trailing edge region is in good agreement with that predicted by other methods. From the present result it also shows that the Blasius solution may be specified in the region $0.4<x/L<0.8$ of a flat plate to predict the wall skin coefficient in the trailing edge. Figure 13 also shows that in the leading edge region the present analysis of the Navier-Stokes equations shows that the Blasius solution based on the boundary layer equation predicts higher value of wall skin friction. Figure 14 shows the velocity

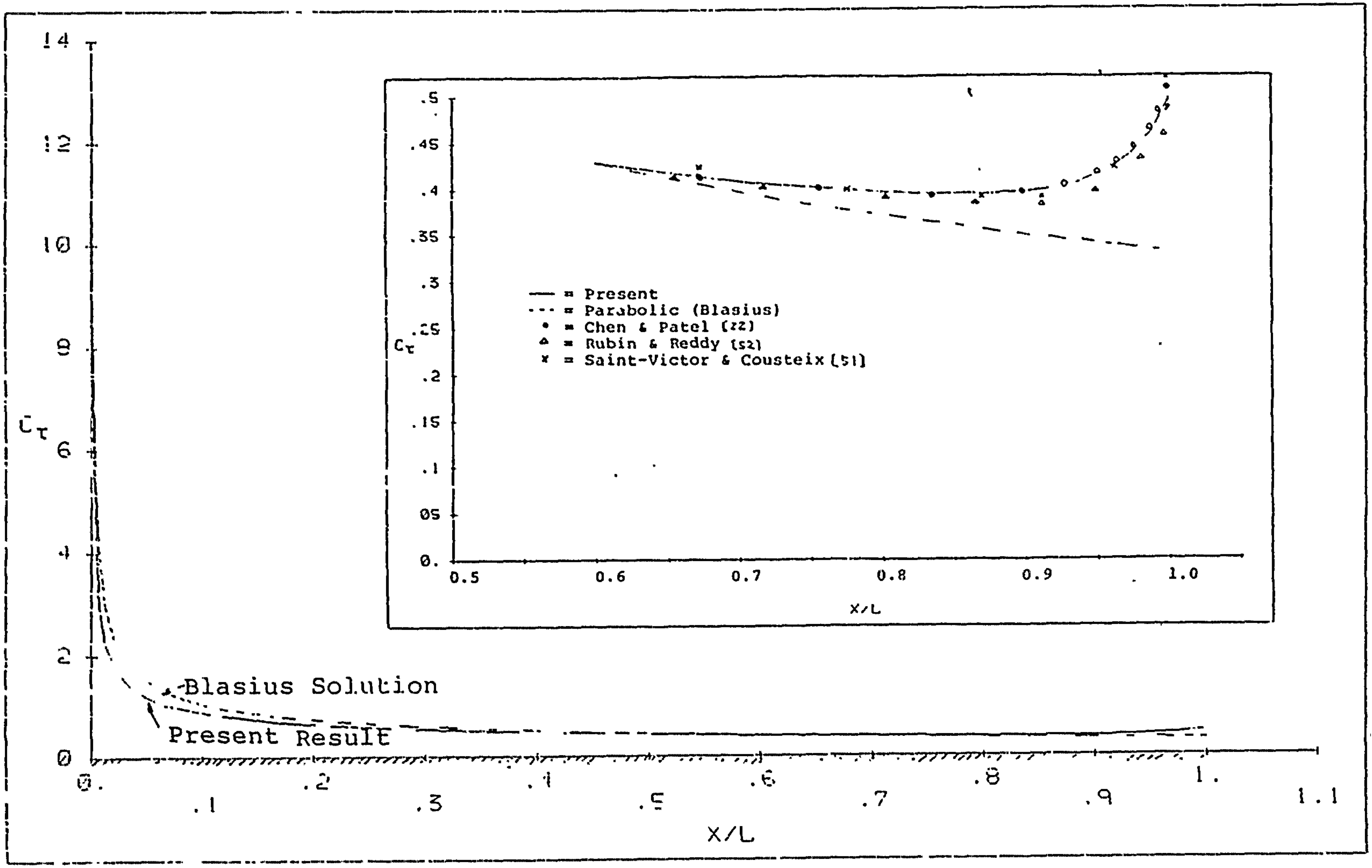


Figure 13. Skin Friction On The Flat Plate

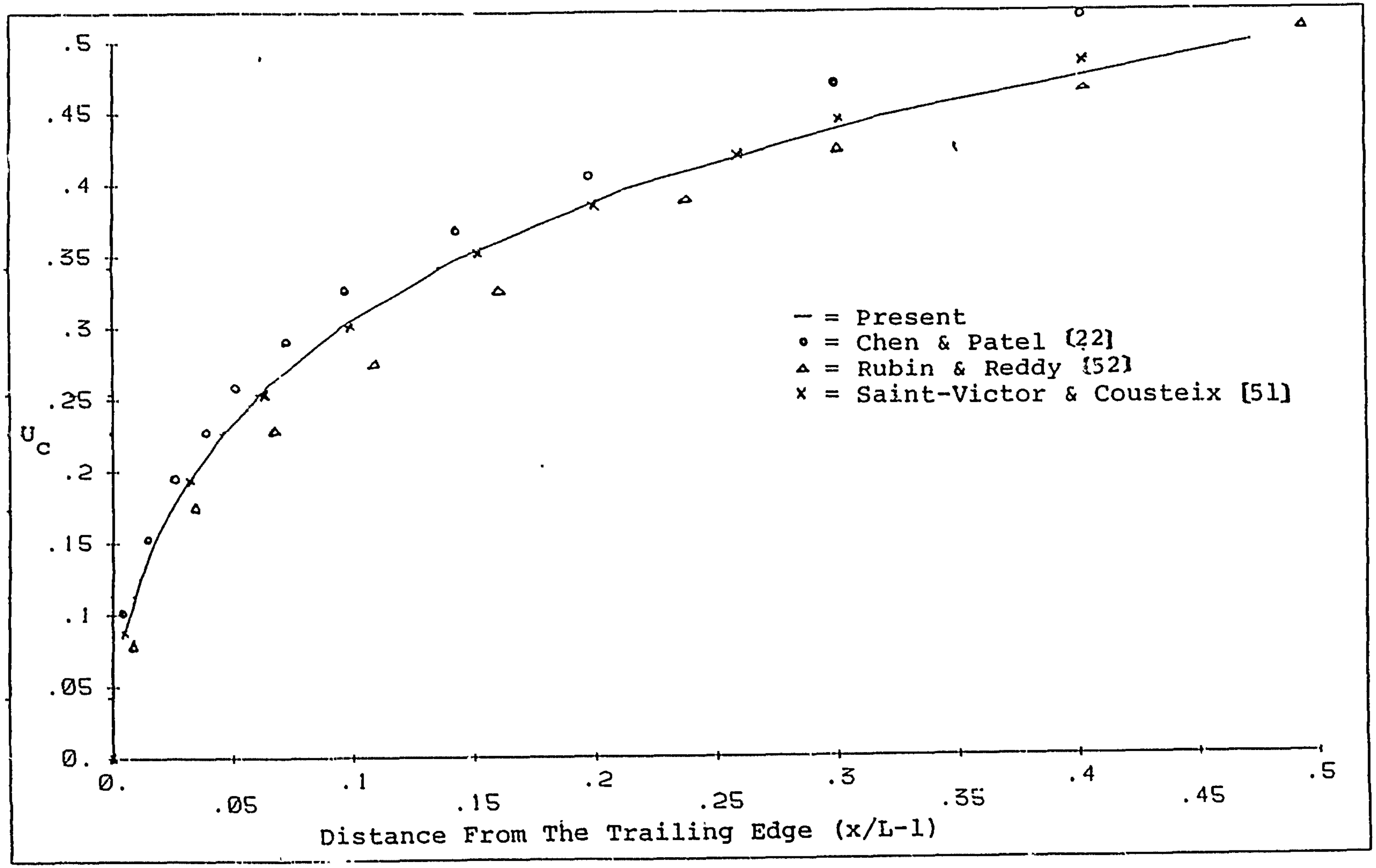


Figure 14. Wake Centerline Velocity

variation along the wake centerline. Here the present solutions agree well with that predicted by Saint-Victor and Cousteix [51]. When compared to the present solution one finds that the analysis of Chen and Patel [22] predicted a higher velocities and that of Rubin and Reddy [52] predicted smaller velocities.

From the above comparisons one observes that with different numerical approaches, initial conditions and computational domain the solution to the same problem may be different although all solutions are qualitatively similar. The FANS-3DEF program shows here that it is capable of solving the complete laminar flow past a finite flat plate from the upstream of the plate to the wake region.

4.3 Laminar Flow With Angle Of Attack

Although the solutions for the flow past a flat plate without angle of attack are available the solution for the flow with an angle of attack is scarce if not available. This is primarily because when there is an angle of attack the flow may be separated and shedded and the problem becomes unsteady and is governed by the Navier-Stokes equations and not the parobolized equation or boundary layer equation. In the present study since the FANS-3DEF solves the Navier-Stokes equation the flow over a flat plate with angle of attack may be solved. Since the symmetric condition is no longer applied for laminar flow over a flat plate with an angle of attack the computational domain are redefined with extended boundaries both in the x and y directions as shown in figure 15. The same numerical grid generation technique used in the previous section was again employed but with A1=0.3, A2=0.2, A3=0.2 and ε_1=19 at x=0, ε_2=49 at x=1, η=1 at y=0 and η=19 at y=+1.5 for both upper and lower domains. Thus, a 67x37 mesh was used to cover the physical region that extends from a distance 1.385L upstream of the leading edge to 8.762L downstream of the trailing edge, and 1.5L normal to the plate on both upper and lower boundaries.

In this study the regular grid system with Navier-Stokes equations are solved again by the FANS-3DEF for two different angles of attack, namely α= 5 and 10 at Reynolds number $Re=10^4$. The inclined uniform velocities u, v ($u=U_0\cos(\alpha)$, $v=U_0 \sin(\alpha)$) and zero pressure were specified at upstream and both upper and lower free stream boundaries. Since the outlet plane is located at 7.762L downstream of the trailing edge which is far downstream from the plate, the second derivatives of all the variables at this plane are approximately set equal to zero. The problem then is solved on the FANS-3DEF program with the time marching procedure, since the separation and unsteady flow phenomena is expected for the flow at incidence, a smaller time

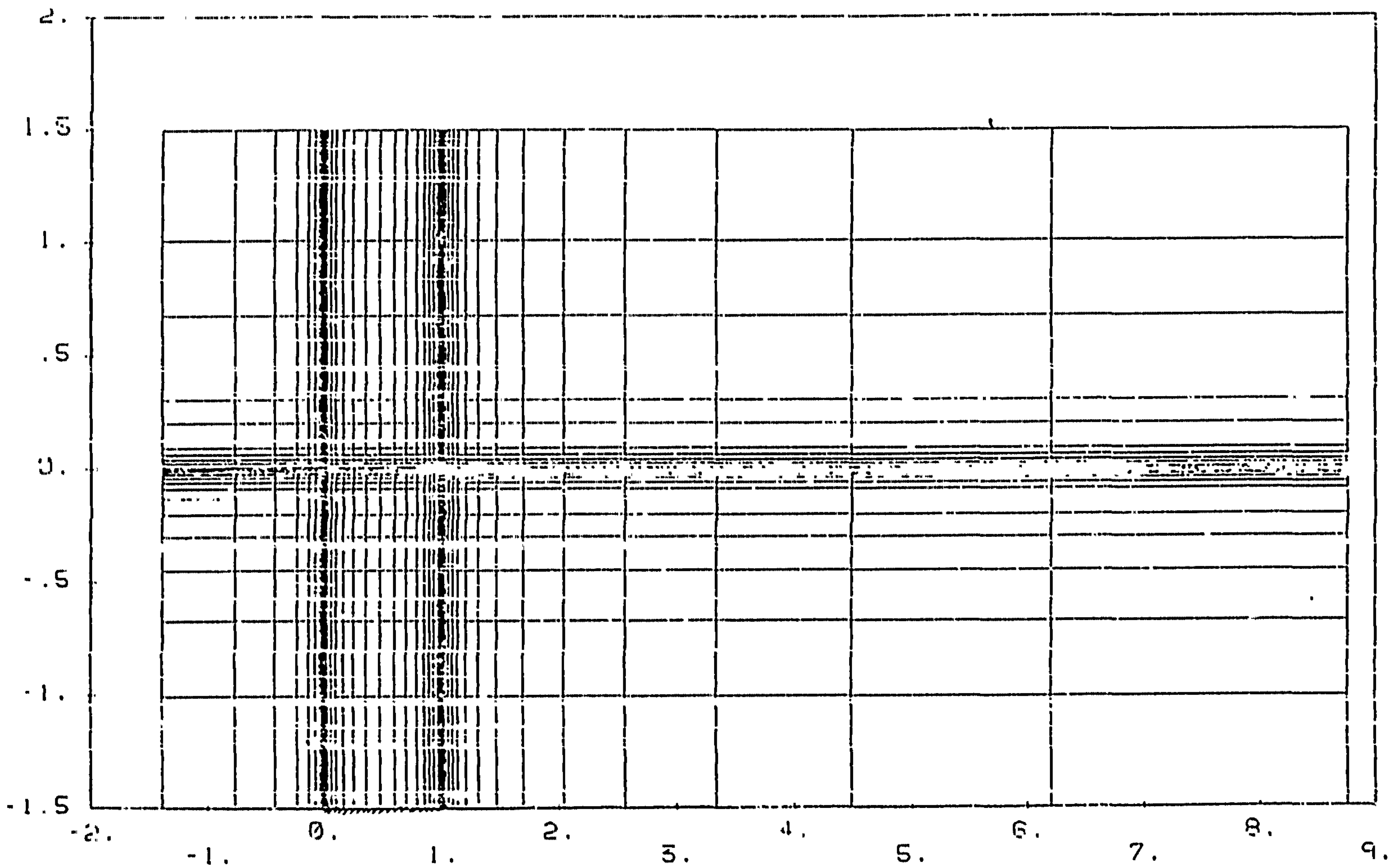


Figure 15. The Numerical Grid For Laminar Flow Past An Inclined Plate

increment $\Delta t=0.1$ is used here. The initial guess of the velocities in the whole computational domain are the uniformly inclined velocity. A total of 30 time steps were calculated. Calculation for each time step approximately consumes 60 seconds CPU time on PRIME 750.

4.3.1 5 Degree Angle Of Attack

Figure 16(a) shows the instantaneous streamline distribution around the flat plate at 5 degree angle of attack at time t=3 after the calculation. At this instance a small separation at the leading edge on the upside plate can be seen in figure 16(b), where the y coordinate normal to the plate is greatly stretched in order to visualize the separation zone. It should be remarked that the flow at a 5 degree angle of attack does not show shedding. In other words the separation at the leading edge is a stationary separation zone. Werle [53] experimentally showed the existence of the stationary separation at a small angle of attack and the shedding of separation flow at a large angle of attack. Figure 17(a) shows the experimental study of werle [53] who used a very thin flat plate (t=0.02L) at Reynolds number 10 for 2.5 degree angle of attack. In this figure a much larger separation zone than that predicted by the present study for $\alpha=5$ was seen at the leading edge over the upper surface of the plate. It should be kept in mind that in the present study a zero thickness is assumed while in the expermental study a two percent thickness of flat plate with sharpened leading edge is used. The sharpened edge tends to promote separation and make the separation zone bigger.

Figure 18 shows the convergence history of the pressure distribution on both the upside and downside of the plate at a 5 degree angle of attack. Since at this angle of attack the separation zone is still small and no shedding phenomena is seen the pressure distribution is stationary on both sides of the plate. The solution converges on the downside of the plate in 10 time steps and on the upside of the plate in 15 time steps. The pressue value on the windward or downside of the plate is positive while it is negative on the leeward or upperside of the plate. The maximum and minimum pressure distribution occurs at the leading edge of the plate. The maximum of p=0.48 on the windward and the minimum of p=-0.74 on the leeward. The absolute value of pressure on both sides continues to decrease from the leading edge to the trailing edge where the same pressure value p=-0.02 is found. No experimental data of pressure for the flow past a very thin flat plate at angle of attack is available. Figure 19 [54] shows the pressure distribution on the NACA 0012 airfoil at 4 degree of angle attack. NACA 0012 airfoil is a symmetric airfoil but has a

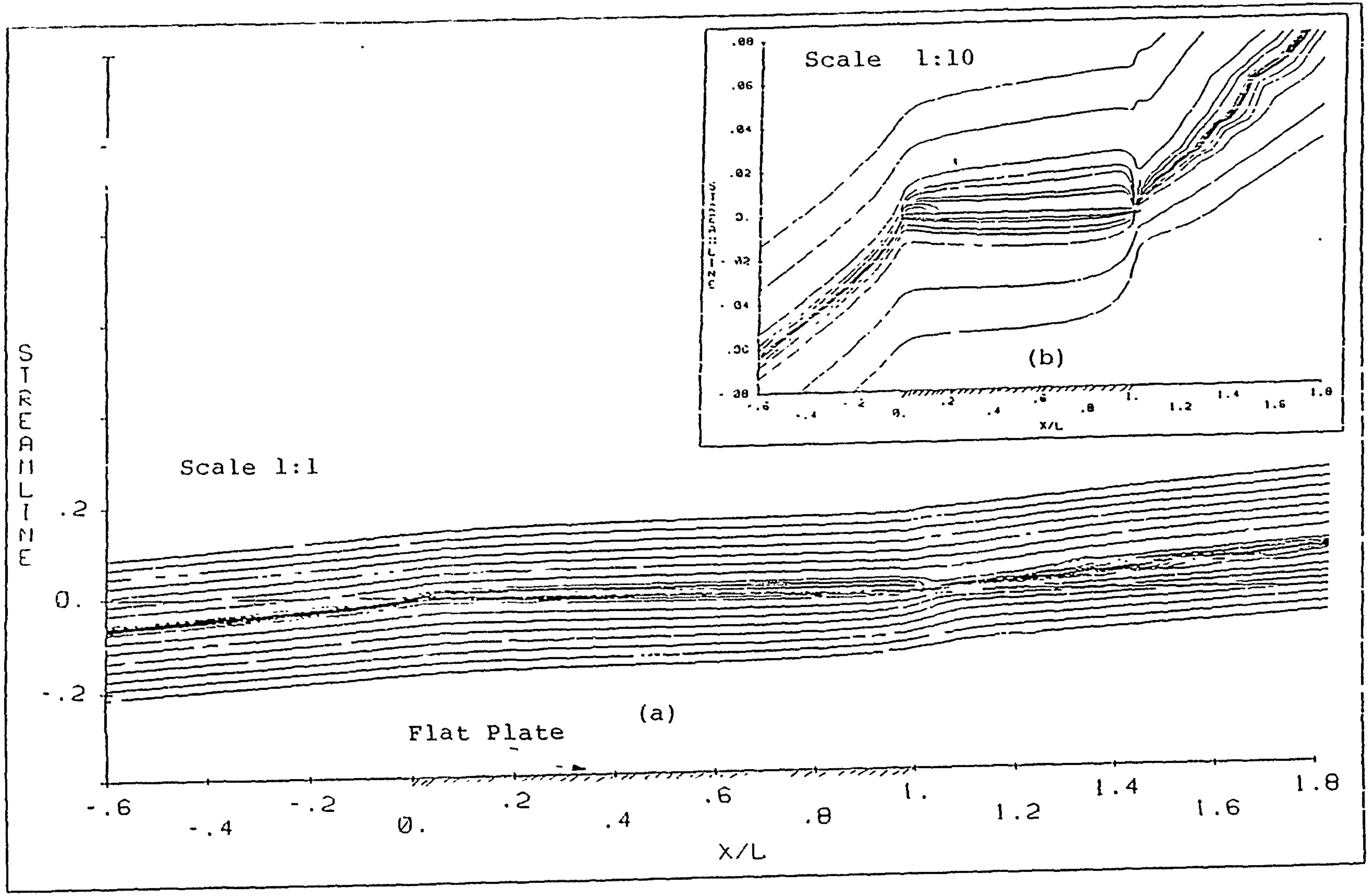
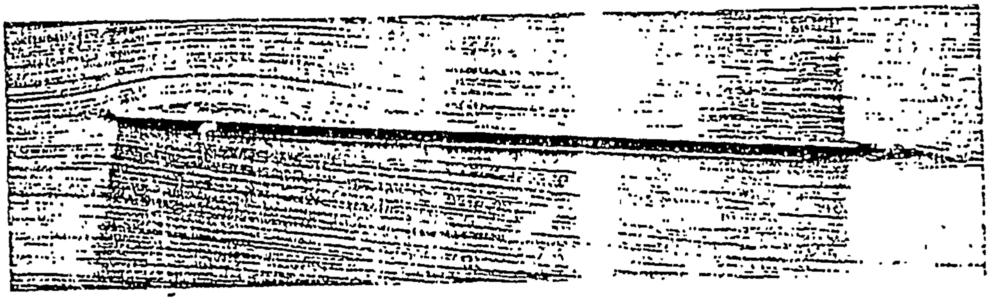


Figure 16. Streamline Distribution At $Re=10^4$ With 5 Degree Angle Of Attack



(a) A Flat Plate (2 Per Cent Thick) At $Re=10^4$ With 2.5 Degree Angle Of Attack

(b) A Flat Plate (2 Per Cent Thick) At $Re=10^4$ With 20 Degree Angle Of Attack

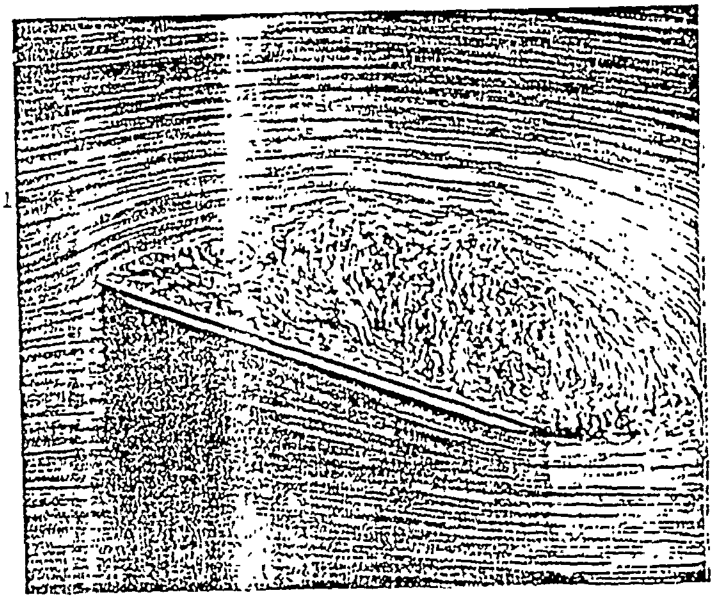


Figure 17. The Visualization of an Inclined Flat Plate at Different Angle of Attack

maximum thickness of 12 percent of the cord. Comparing the predicted result and experiment data of NACA 0012 airfoil, one sees that the present analysis predicted a similar solution for pressure distribution on both sides of the flat plate to that on the NACA 0012 airfoil.

4.3.2 10 Degree Angle Of Attack

In order to investigate the flow past a flat plate with a larger angle of attack so that the flow is shedded from the separation, the angle of attack is increased from 5 to 10 degree. At this angle of attack the same computational domain, grid space, time increatment and initial and boundary conditions used in the previous section for 5 degree angle of attack are adopted here. Figures 20 and 21 show a series of changes of streamline distribution and velocity vectors around the flat plate from time t=2 to t=6. The dimensionless t is defined by $t=TU_0/L$. Where T is the dimensional time, U_0, the free stream velocity and L the length of the plate. From figure 20(a) and 21(a) one sees that at time t=2 a large separation bubble which covers 0.8L of the upper surface is formed. From t=3 to t=5 these figures reveal that while the separation bubble is being pushed down toward the trailing edge of the plate a new separation bubble is created at the leading edge and grows in size. At time t=6 the first separation bubble is

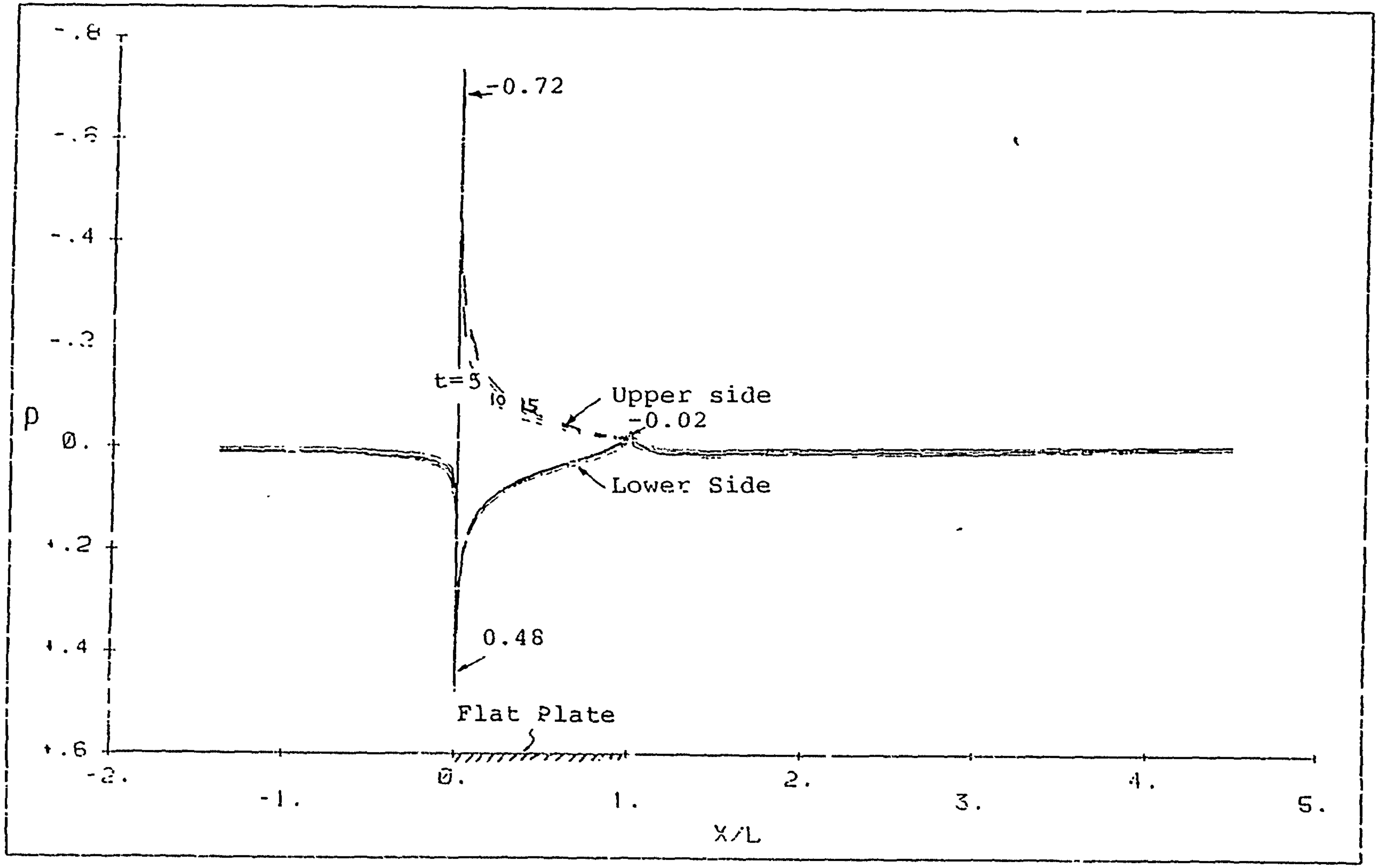


Figure 18. Convergence History Of The Pressure Distribution On Both Upper And Lower Side Of The Plate

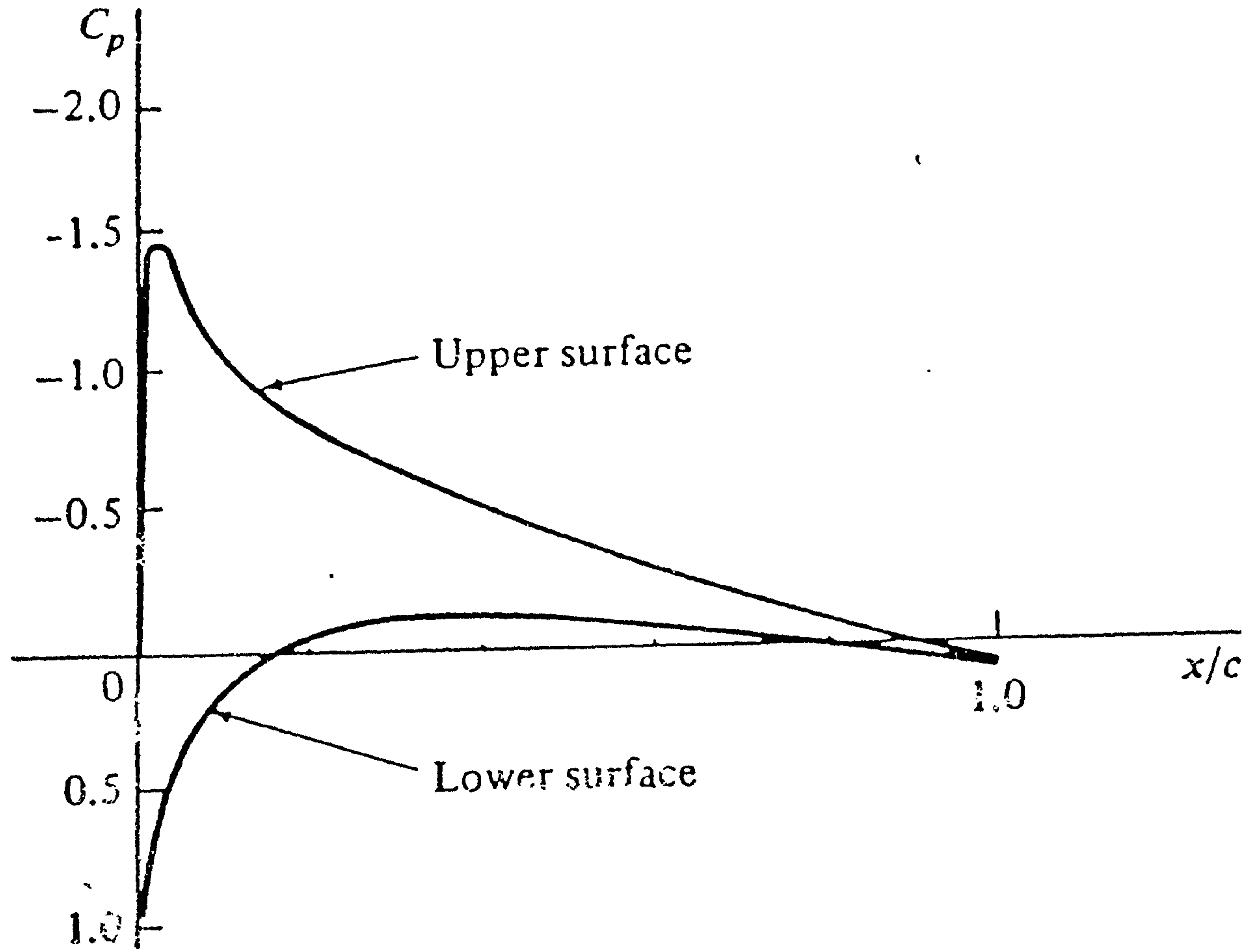


Figure 19. Distribution Of Pressure Coefficient Over The Top And Bottom Surfaces Of A NACA 0012 Airfoil At 3.93 Angle Of Attack, $Re=3.245 \times 10^6$.

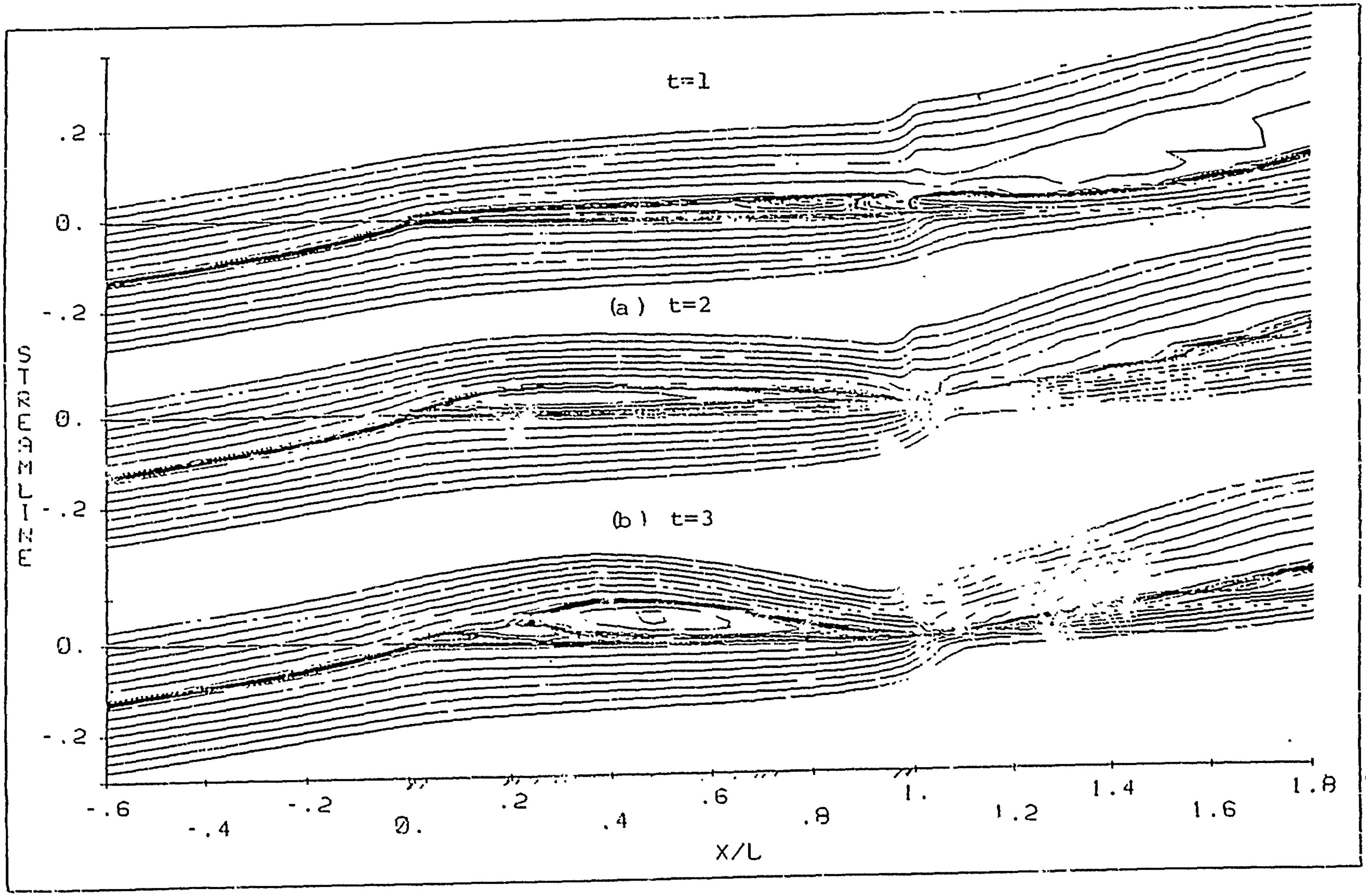


Figure 20. Streamline Distribution On A 10° Inclined Flat Plate

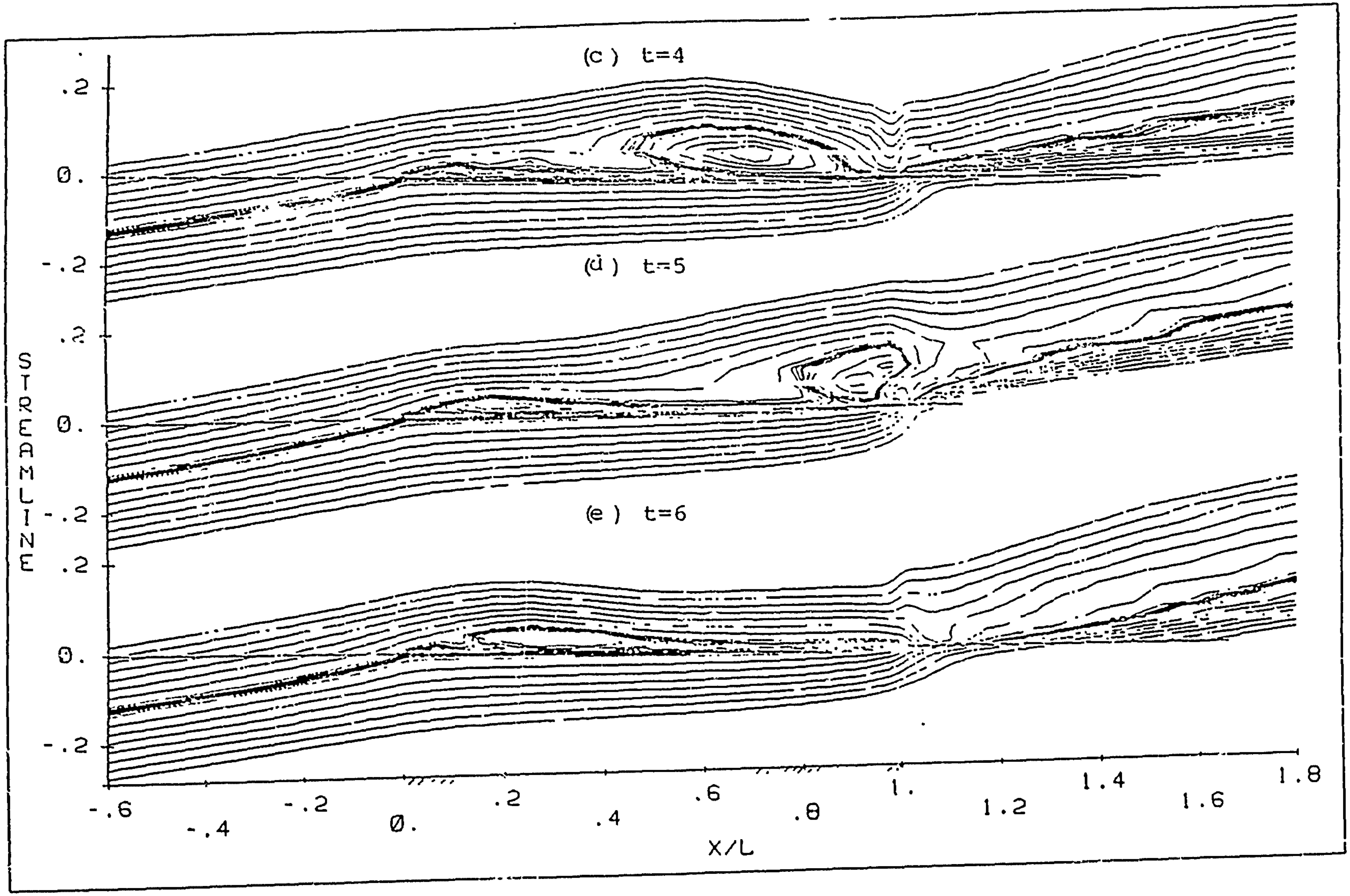


Figure 20. Continued

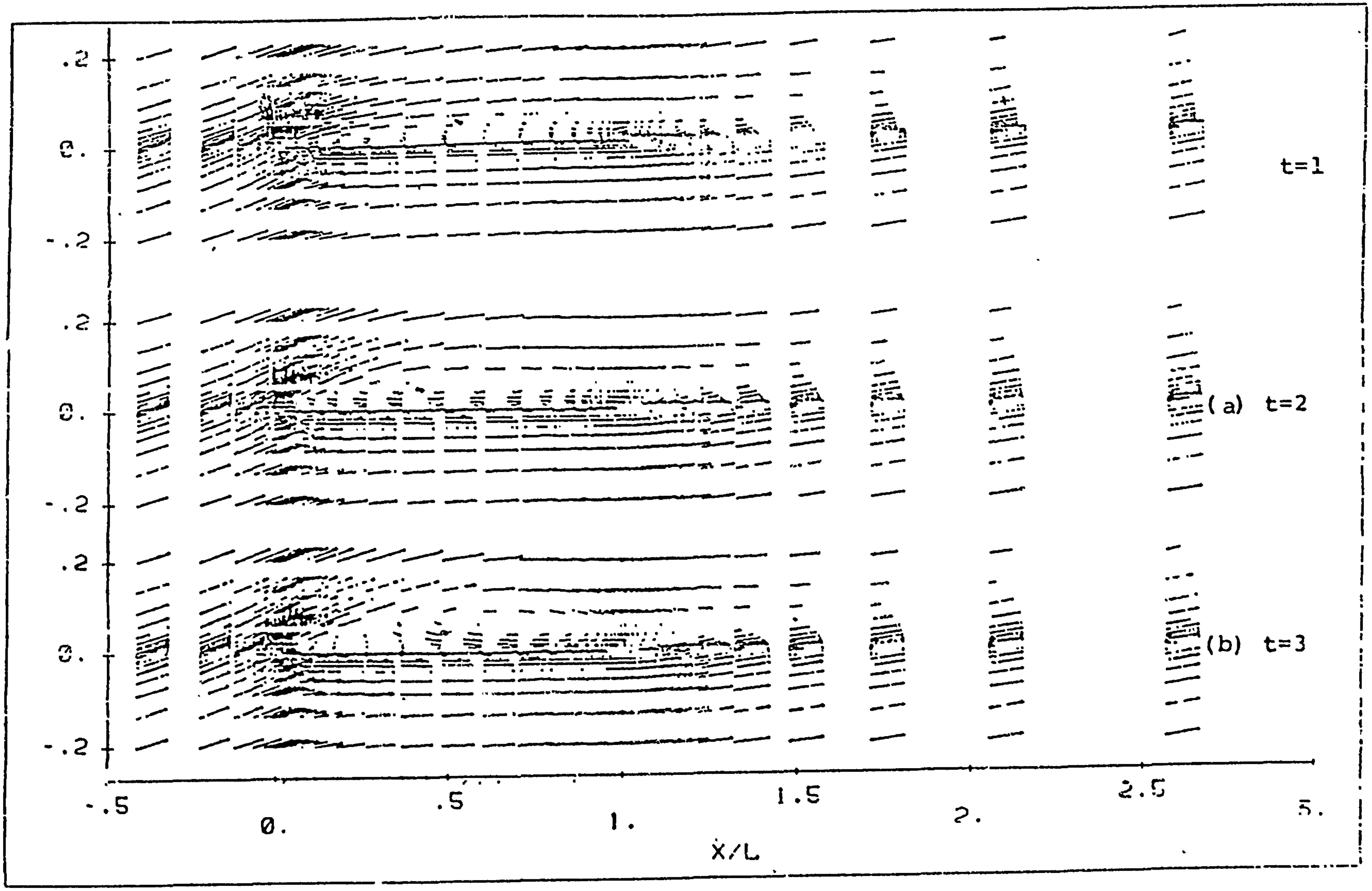


Figure 21. Velocity Vector On A 10^{0} Inclined Flat Plate

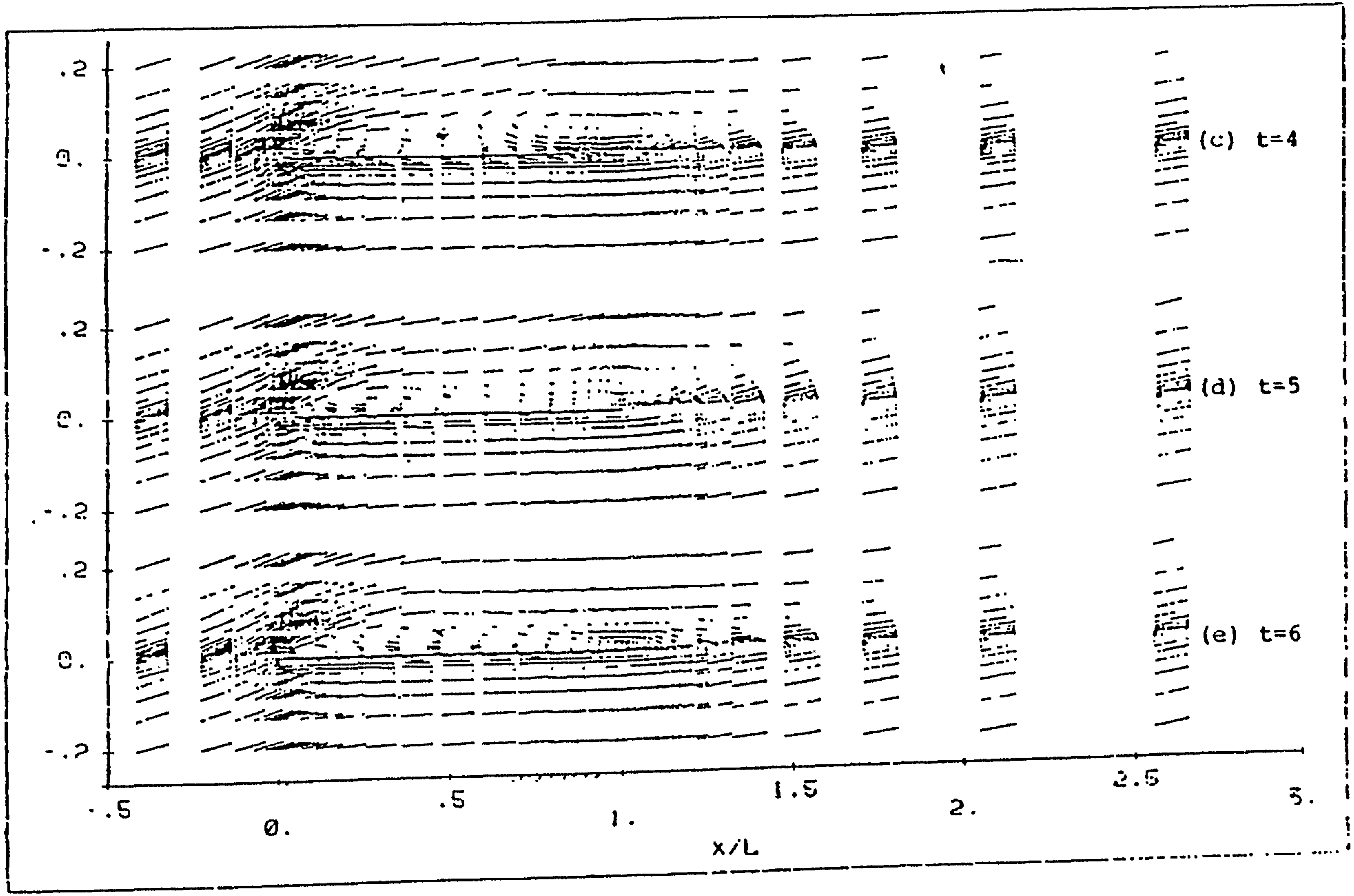


Figure 21. Continued

completely washed away behind the plate and the second new separation bubble has grown to a size such that the pressure in front of the bubble is larger than that behind and consequently the bubble begins to move and the shedding process repeats. A complete period is then achieved. The shedding Strouhal number, S, from this shedding is found to approximately 0.2. The Strouhal number is defined as $S=nL/U_0$ where n is frequency, U_0 and L are reference velocity and length.

In this chapter the calculation of flow past a flat plate is used to test the capability of the FANS-3DEF numerical algorithm and numerical method. It is found that the FANS-3DEF can predict laminar flow with or without angle of attack with reasonable accuracy. If a more accurate result of the flow phenomena is desired more fine grids and smaller time step should be used.

Figure 22 shows the corresponding pressure distribution on the both upper and lower sides of the plate at different times. One sees that the pressure distribution on the downside of the plate is almost constant at each different time step but the pressure on the upside of the plate varies rapidly even at two close time steps revealing the occurence of vortex shedding. In table 7 the value of pressure on each different station at different time step is shown. It shows that the pressure in front of the separation zone is

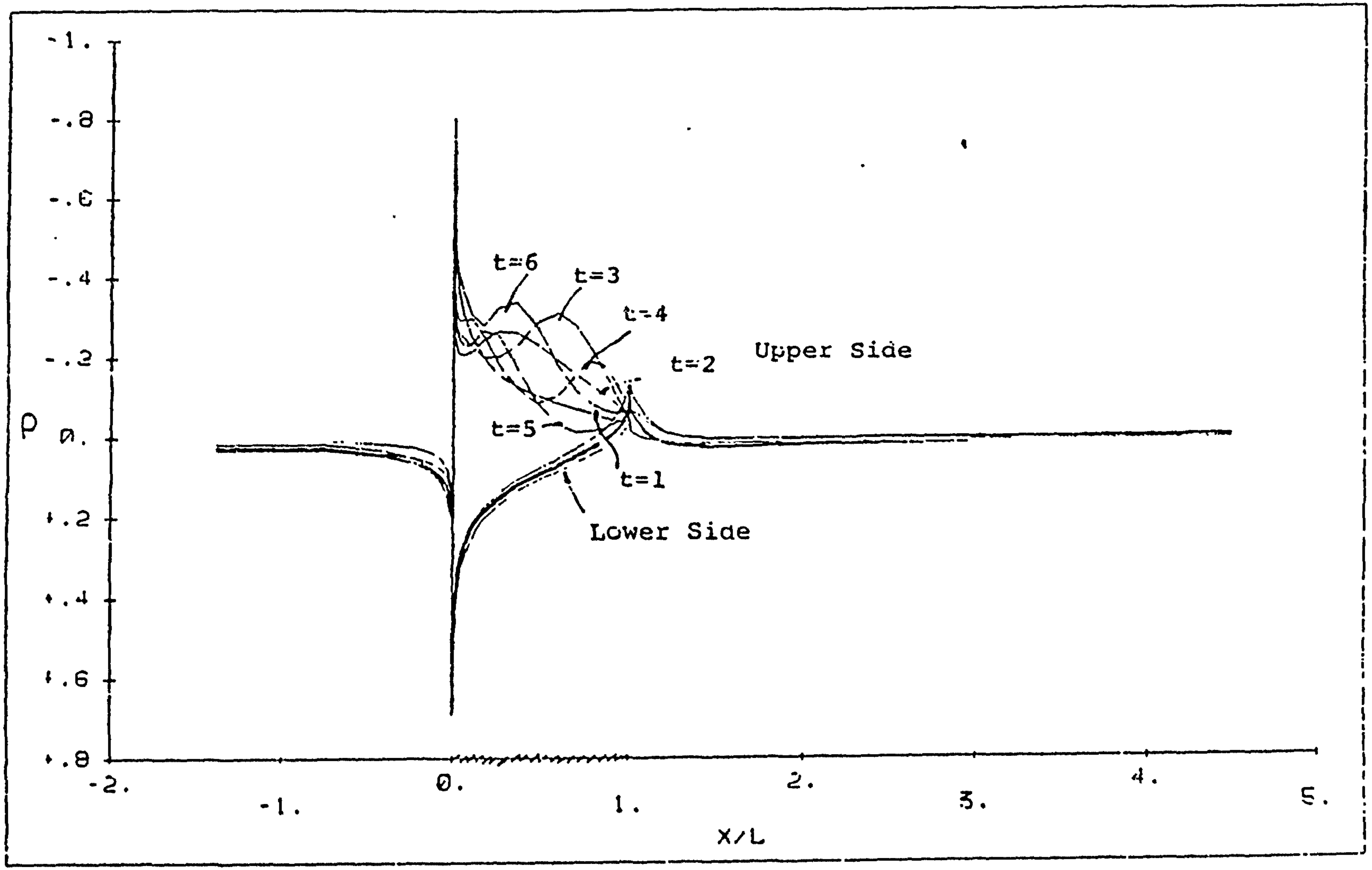


Figure 22. Pressure Distribution At Different Time Step

(A) Lower Side of the Plate

t \ x/L	0.00050	0.02133	0.10840	0.17230	0.25850	0.34440	0.48159	0.59930	0.70330	0.79200	0.857400	0.90360	0.95610	0.98040	0.99440
t = 1	0.48710	0.38480	0.24430	0.20240	0.16310	0.13390	0.10500	0.00091	0.05996	0.04140	0.02363	0.01149	-0.00978	-0.02341	0.03336
t = 2	0.41110	0.35530	0.22730	0.18850	0.15330	0.12190	0.09380	0.06860	0.06504	0.02313	0.00623	-0.01090	-0.03908	-0.05044	-0.07374
t = 3	0.45750	0.35490	0.22440	0.18620	0.15150	0.12150	0.09447	0.07034	0.06049	0.03090	0.01125	-0.00449	-0.03949	-0.06394	-0.05701
t = 4	0.44180	0.36010	0.22400	0.18490	0.14990	0.11900	0.09140	0.06441	0.04422	0.02370	0.00505	-0.01203	-0.04029	-0.06039	-0.07557
t = 5	0.43930	0.36370	0.21090	0.17210	0.13710	0.10540	0.07480	0.05020	0.02330	0.00170	-0.03048	-0.04159	-0.07709	-0.10300	-0.13390
t = 6	0.43530	0.35860	0.22000	0.18060	0.14970	0.11490	0.08733	0.06308	0.04113	0.02147	0.00387	-0.01164	-0.03611	-0.05359	-0.06438

(b) Upper Side of the Plate

t \ x/L	0.00050	0.02133	0.10840	0.17230	0.25850	0.34440	0.40150	0.59930	0.70330	0.79300	0.85740	0.90160	0.95610	0.98030	0.99440
t = 1	-0.17510	-0.11420	-0.25100	-0.20130	-0.16030	-0.12940	-0.10510	-0.08449	-0.07234	-0.06059	-0.05193	-0.04417	-0.04319	-0.04170	-0.04429
t = 2	-0.31190	-0.22610	-0.22113	-0.21610	-0.24770	[illegible]	-0.33360	-0.19570	-0.15940	-0.13190	-0.11040	-0.09313	-0.07164	-0.05490	-0.04673
t = 3	-0.14940	-0.27110	[illegible]	[illegible]	-0.23360	[illegible]	-0.23930	-0.31340	-0.20340	-0.12430	-0.16920	-0.12039	-0.08313	-0.06346	-0.05982
t = 4	-0.11120	[illegible]	-0.10013	[illegible]	[illegible]	[illegible]	-0.08485	-0.10728	-0.13970	-0.10070	-0.13090	-0.13540	[illegible]	-0.09930	-0.10230
t = 5	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-0.33451	-0.03500	-0.31744	-0.01965	-0.02710	-0.03543	-0.01833	-0.35407	-0.30737
t = 6	[illegible]	[illegible]	[illegible]	[illegible]	-0.22390	[illegible]	-0.27333	-0.18310	-0.11930	-0.08531	-0.07032	-0.36431	-0.06240	-0.30430	-0.06824

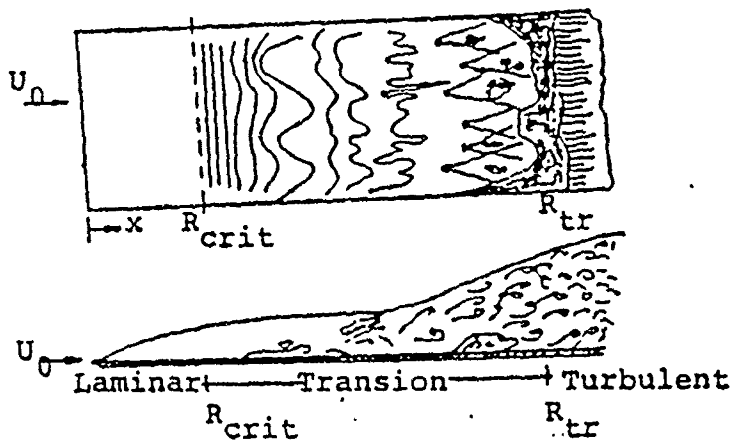
Table 4. The Pressure Values At Different Time Step And Different Location

large and only next to the pressure at the leading edge. The pressure difference created arround the separation zone is then responsible for moving the separation zone downstream and producing shedding. The feed back of shedding phenomena on the upper plate also promotes the change of maximum and minimum values on the leading edge at different time steps.

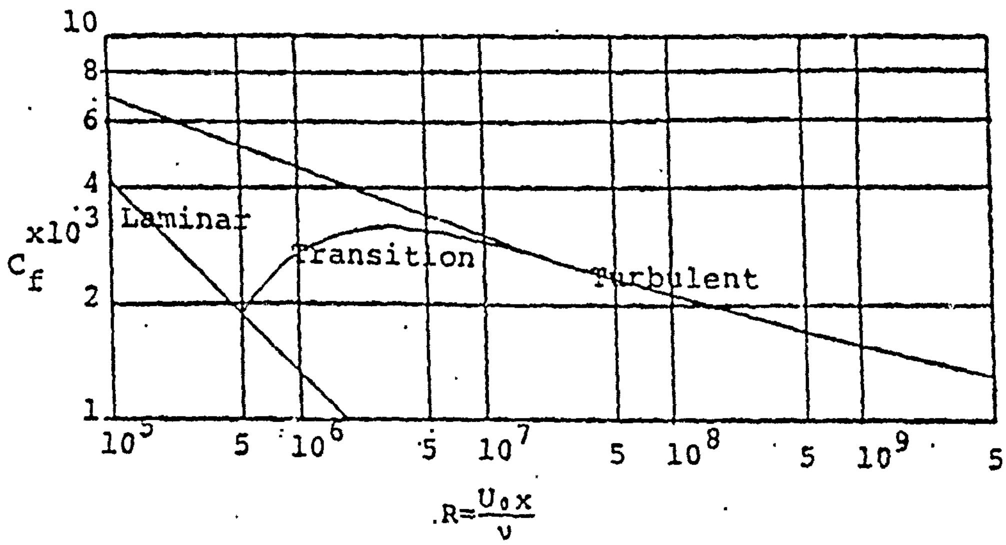
4.4 Modelling Of Laminar-Turbulent Transition

Before using the numerical method to solve a complete solution of high Reynolds number flow over a finite flat plate, a brief review of how the flow changes from the laminar to turbulent is needed. Figure 23(a) is a sketch of flow evolution from the leading edge of a flat plate at zero angle of incidence. It shows that between laminar and turbulent flow, there exists a small region called the the transition zone. Figure 23(b) shows the corresponding coefficient of skin friction around the transition zone. It can be seen that in the transition zone there is a sudden increase of skin friction and increase in the boundary layer thickness from laminar flow to turbulent flow. At present only a small and initial portion of the transition zone is amenable to a theoretical analysis. The analysis and theoretical treatment of the complete transition flow are still unavailable. Numerically Cebeci and Smith [55] and Granville [56] had proposed some correlation functions for predicting transition flow but they are all based on the boundary layer assumption. Since there exists this kind of difficulty and inability in predicting the transition zone, then many previous numerical studies [19,20,22] for turbulent flow past a plate or bodies were made only for the region where the flow is turbulent. In the present study an attempt is made to create a simple numerical model for predicting the transition.

In devising a numerical model for the transition zone the question is how the numerical treatment can be done to connect the laminar flow and turbulent flow so that the location of transition can be approximatel predicted and the overall behavior of the skin friction $C_f(2\tau_w/\rho U_o^2)$ on the plate can also be predicted. In other words since the actual length of the transition zone is not clearly defined no attempt is made to numerically predict the transition length. As mentioned in section 2.3, once the flow becomes turbulent we shall, instead of applying no-slip conditions on the surface, use the two-node log-law equation to approximate the near wall solution up to the first node from the wall. While in the laminar flow the computational domain is numerically extended to the wall. The numerical model for transition then requires a criteria to indicate



(a) Idealized Sketch Of Transition Zone In The Boundary Layer On A Flat Plate At Zero Incidence

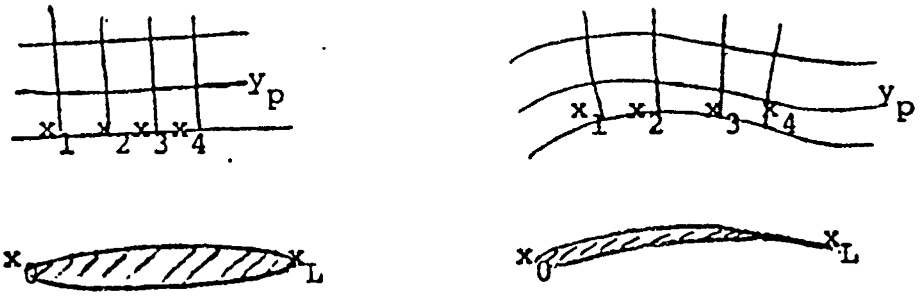


(b) Skin Coefficient For Smooth Flat Plate At Zero Incidence

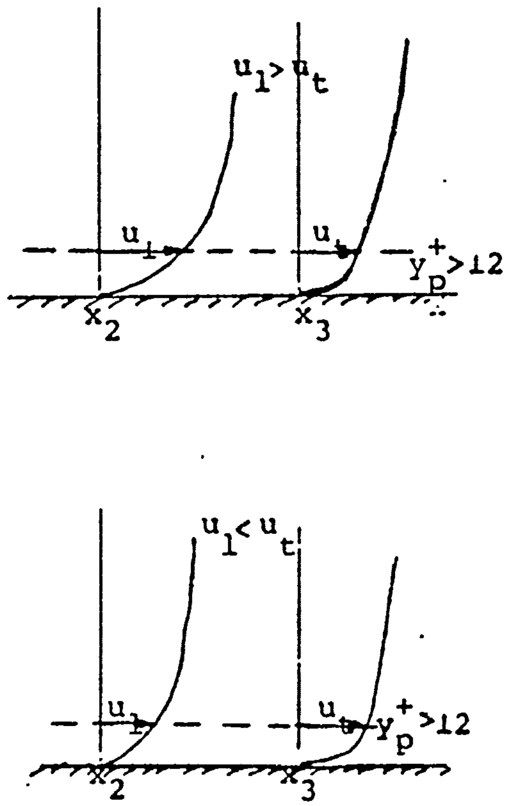
Figure 23. Transition On A Flat Plate At Zero Incidence

when the flow is turbulent so that turbulent wall function is used. The following is the process to identify the location when the flow chang from laminar to turbulent.

As shown in figure 24(a) the flow over a surface which can be a flat plate or a curved surface without an abrupt change in the curvatures is considered. When the local Reynolds number is sufficiently large on the surface the flow may go through transition from laminar to turbulent in the flow domain between x_0 and x_L. Here let's assume the transition from laminar motion to turbulent motion occures at two close computational nodes denoted by x_l and x_t and the tangential velocities at these two locations are u_l and u_t. The subscript l means the laminar flow while the subscript t means the turbulent flow. In reality the transition will normally take a larger distance before the laminar motion becomes a completely three dimensional, irregular unsteady and rotational flow of turbulent motion. A more realistic model of transition will be discussed later. Before we continue, some assumptions about the flow around the transition zone are made as (1) the velocity near the wall along the surface continues to decrease whether the flow is laminar or turbulent (2) the flow starts with laminar flow at the leading edge and continues to be laminar until the point of turbulent flow is defined (3) after this point the fully turbulent flow is considered. Under these



(a) Grid Nodes Along The Surface



(b) Tangential Velocity Between Two Computational Nodes

Figure 24. Criteria Of Transition Zone

assumptions a comparison between the two close tangential velocities at same normal distance y_p to the wall is made. Since in this study the log-law formulation is used for the turbulent calculation, therefore y must the value between 12 and 200. In this study we choose y ° 0.0006 and find that it satisfies the requirement for $Re=2.48x10^6$ on the whole plate. As shown in figure 24(b) if $u_l(x_2,y_p)$ is larger than $u_t(x_3,y_p)$ then the turbulent velocity at (x_3,y_p) is replaced by the laminar velocity and the comparison moves downstream by one node, or between $u_l(x_3,y_p)$ and $u_t(x_4,y_p)$. If $u_l(x_2,y_p)$ is less than $u_t(x_3,y_p)$ then the turbulent is assumed to occure at (x_3,y_p) and the comparison moves upstream by one node, or between $u_l(x_1,y_p)$ and $u_t(x_2,y_p)$. Repeat the same process until $u_l(x_2,y_p)$ is less than $u_t(x_3,y_p)$ and $u_t(x_2,y_p)$ is less than $u_l(x_1,y_p)$ then the location (x_2,y_p) is the starting point of transition.

As mentioned before, in reality the transition occurs in a larger spacing than between two computational nodes. To remedy the drastic transition of the solution from a laminar to turbulent flow in the present study the solution in the laminar region from the leading edge to the location of the transition is not solved by the laminar Navier-Stokes equations but by turbulent Navier-Stokes equation with a reduced eddy viscosity. The reduced eddy viscosity at a given location or node in this region is set equal to a 80%

of the eddy viscosity of the downstream node. In other words the value of the eddy viscosity from the starting point of turbulent flow to the leading edge is set equal to 80% of the downstream value or $\nu_t(x_{t-1},y_p)=0.8\ \nu_t(x_t,y_p)$, $\nu_t(x_{t-2},y_p)=0.8\ \nu_t(x_{t-1},y_p)$ etc., where x_t is the location of transition to turbulent flow.

4.5 Turbulent Flow Without Angle Of Attack

The grid distribution for the calculations of the turbulent flow over the flat plate without angle of attack was again generated by the body-fitted coordinate transformation given in the previous section but with A1=0.3, A2=0.12, A3=0.25, and $\xi=1$ at $x=-1.0619$, $\xi=19$ at $x=0$, $\xi=55$ at $x=1$, $\xi=82$ at $x=8.1406$, $\eta=1$ at $y=0$ and $\eta=15$ at $y=1.0$. The grid distribution for turbulent flow calculation in the y direction is different from that for laminar flow calculation. This is because the turbulent flow near the plate differs from the laminar flow and the implimentation of the wall function for the numerical calculation requires that the first two nodes from the wall must be within $12< y^+ <200$. Thus a total of 82x15 grid nodes is used for solving high Reynolds number flow over the flat plate without angle of attack. A partial view of grid distribution is shown in figure 25. Ramaprian, Patel and Sastry [57] measured the turbulent flow over a streamline body at Reynolds number

$Re=2.48x10^6$. In order to compare the present calculation with the above experiment data, the Reynolds number $Re=2.48x10^6$ is chosen in this study. The same boundary conditions as described in the laminar flow calculation in the previous section, section 5.2, are used here again. The FANS-3DEF program with one-scale k-ε turbulence model on the staggered grid system was solved by using time step $\Delta t=1$. The total marching steps are 100.

Figure 26 shows the convergence history of the wall skin coefficient $C_f(2\tau_\omega/\rho U_0^2)$ on the plate. It can be seen that after 30 time steps the wall skin coefficient hardly changes any more. A jump from $C_f=0.0012$ to $C_f=0.00455$ occurs around x/L=0.108 which is equal to a local Reynolds number about $2.6x10^5$. In other words the transition was predicted to take place at x/L=0.108 or $Rex=2.6x10^5$ while H. Schlitting [58] had predicted it was $5x10^5$ in his theoretical study. This indicates that the proposed model for numerical prediction of transition from laminar to turbulent motion is applicable to the flow over the flat plate. The convergence history of centerline velocity along the wake is shown in figure 27.

A comparison with the experimental data published by Ramparian, Patel and Sastry [57], shows that the present result has a slower velocity recovery in the wake centerline. The convergence history of the dimensionless

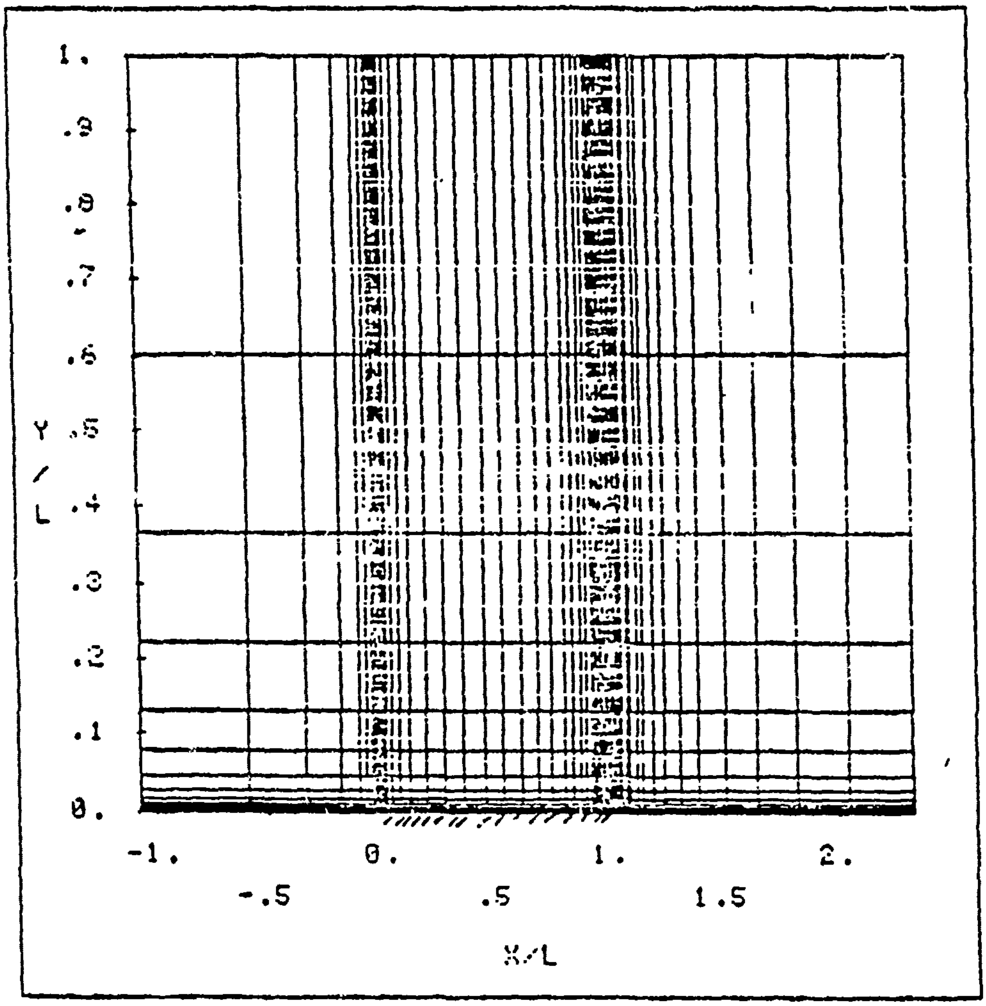


Figure 25. Partial View of Grid Distributuion with 15 Nodes in Y Direction

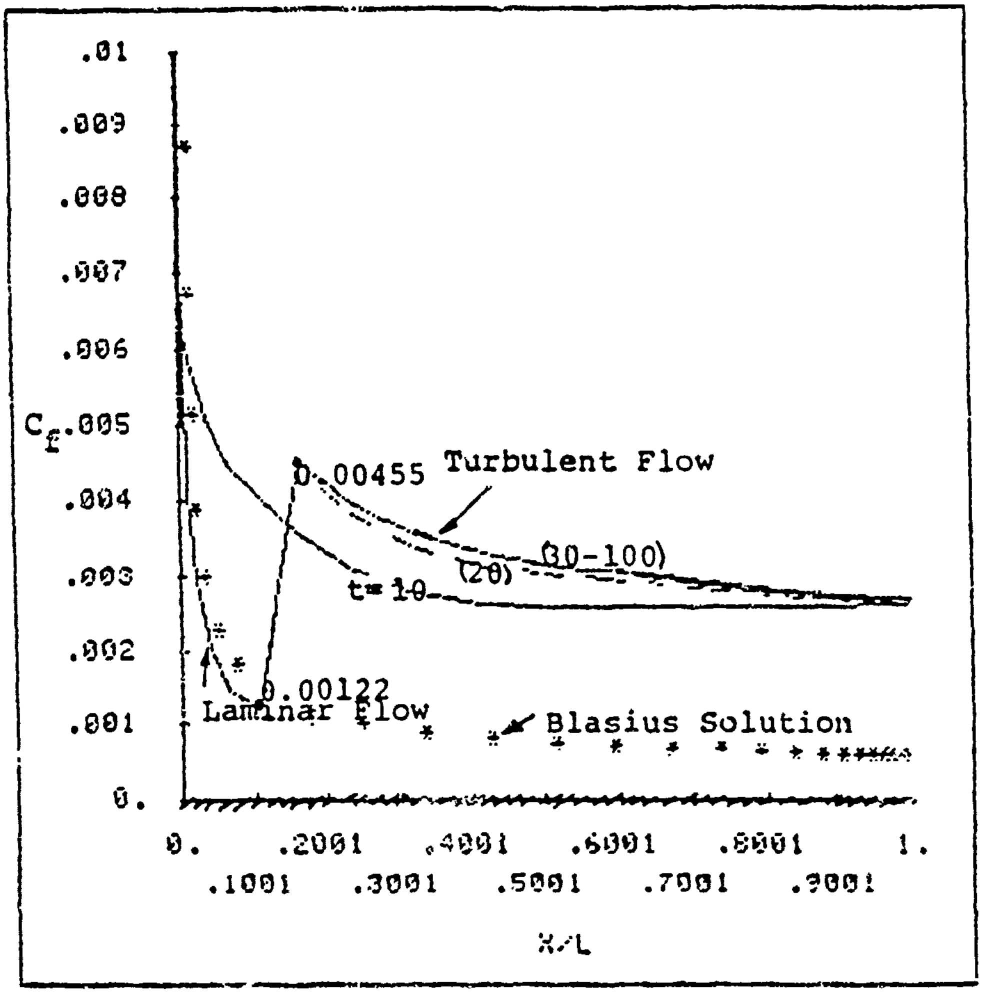


Figure 26. Convergence History Of Skin Coefficient

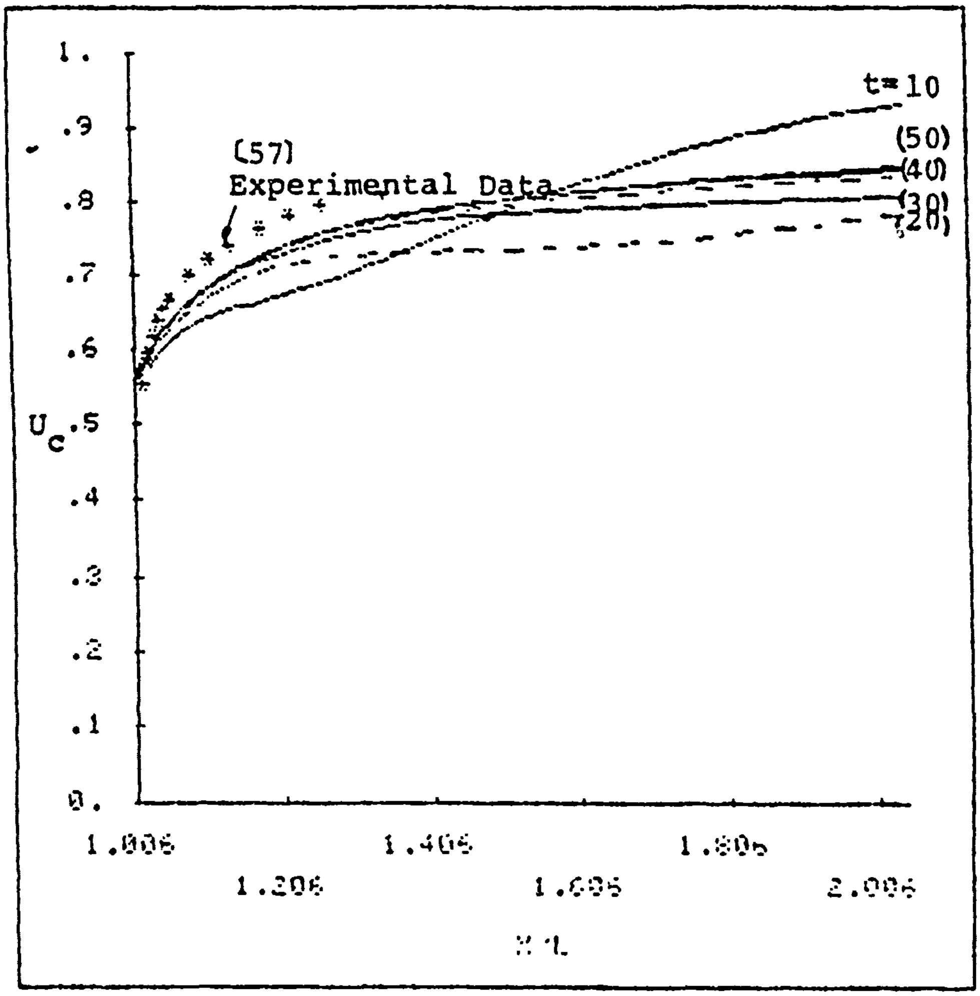


Figure 27. Convergence History of
Centerline Velocity in the Wake

pressure distribution for the entire region including upstream of the plate and wake region is shown in figure 28. A small rise of pressure or p=0.00165 is predicted at the leading edge. A better result of pressure distribution at the leading edge can be obtained by increasing the grid nodes near the plate. The slight underprediction of centerline velocity in the wakes region may be due to the use of two-node wall function on the wall and coarse grids. To improve the prediction more grids are needed especially at the centerline both before the plate and after the plate. In order to achieve a dense grid distribution at the region very close to the wall the concentration factor A3 is changed from 0.25 to 0.2835 and a total of 19 grid nodes along the y direction are used. The partial view of the new grid distribution is shown in figure 29. The computation of the flow is repeated on the FANS-3DEF program.

Comparing the grid distribution between 15 grid nodes and 19 grid nodes, one finds that the four additive grid nodes are created inside the original first node of 15 grid nodes distribution. In other words, the first node near the wall in 15 grid nodes distribution becomes the fifth node in the 19 grid nodes distribution. Since the log-law wall function is still used for the turbulent flow calculation and must be applied between $12 < y^{+} < 200$, the computational domain for the 19 grid nodes is rearranged as shown in the following figure.

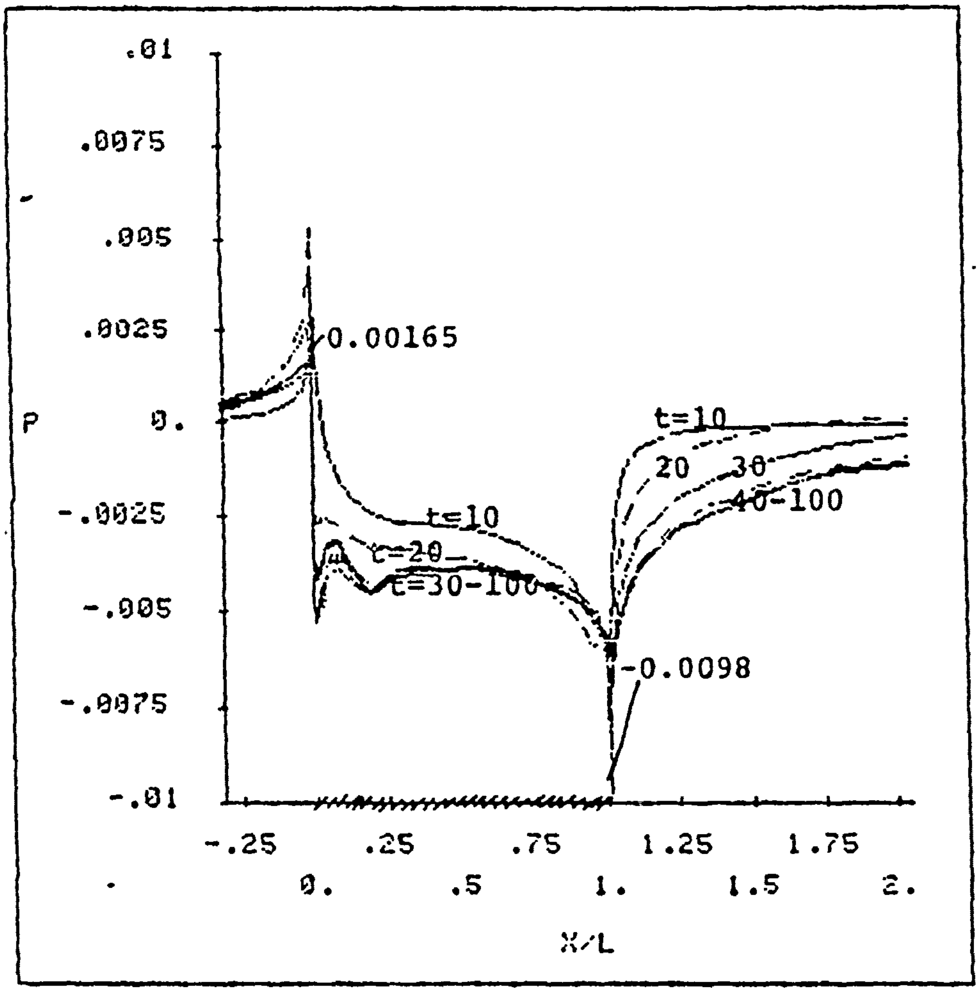


Figure 28. Convergence History of Pressure Distribution

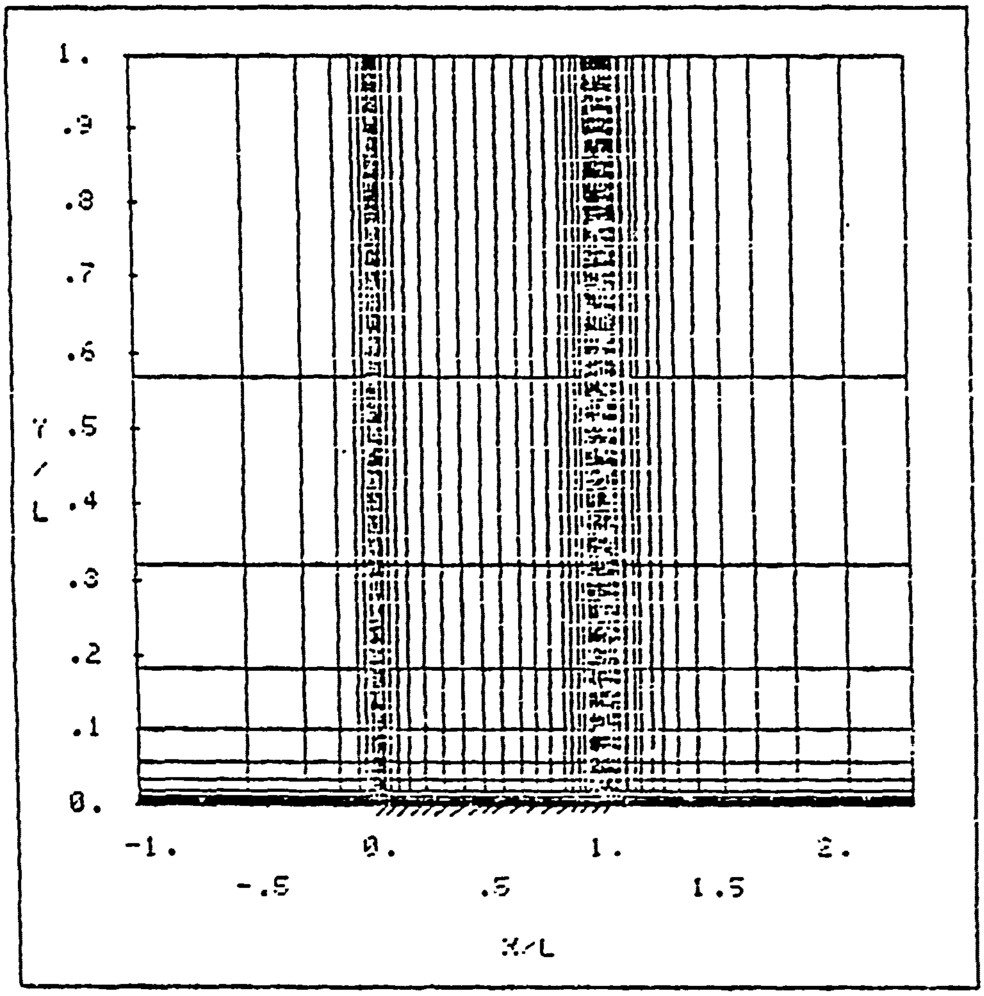


Figure 29. Partial View Of Grid Distribution
With 19 Nodes In Y Direction

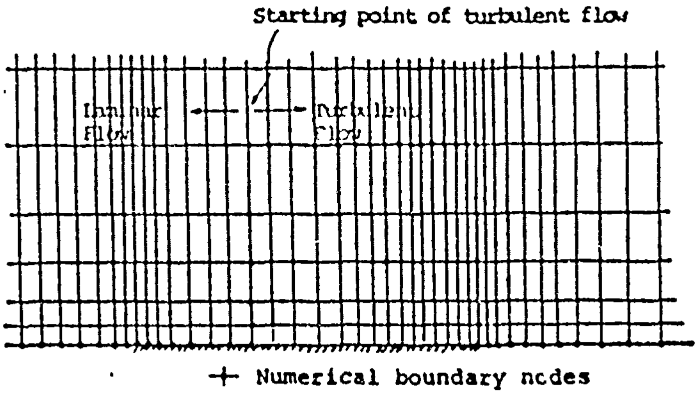


Figure 30. Computational Domain for 19 Nodes

The velocity component u, kinetic energy k and its dissipation rate ε inside the fifth node which involving the use the wall function are evaluated based on the log-law formulation or

$$\frac{u}{U_\tau} = \frac{1}{\kappa} \ln(E\, y^+) \qquad 12 < y^+ < 200$$

$$\frac{u}{U_\tau} = y^+ \qquad y^+ < 12$$

$$k = \frac{U_\tau^2 Y_n}{\sqrt{C}\; Y_5}, \qquad \varepsilon = \frac{U_\tau^3}{\kappa Y_n}, \qquad y+ = \frac{U_\tau Y_n}{\nu}$$

Here Y_n means the normal distance at the nth node, so that Y_5 is the normal distance at the fifth node.

Figure 31 shows the convergence history of the dimensionless pressure distribution before the plate, on the plate and along the wake centerline. One sees that the pressure distribution at the leading edge is now p=0.055 in the 19 nodes grid distribution, while it is only 0.00165

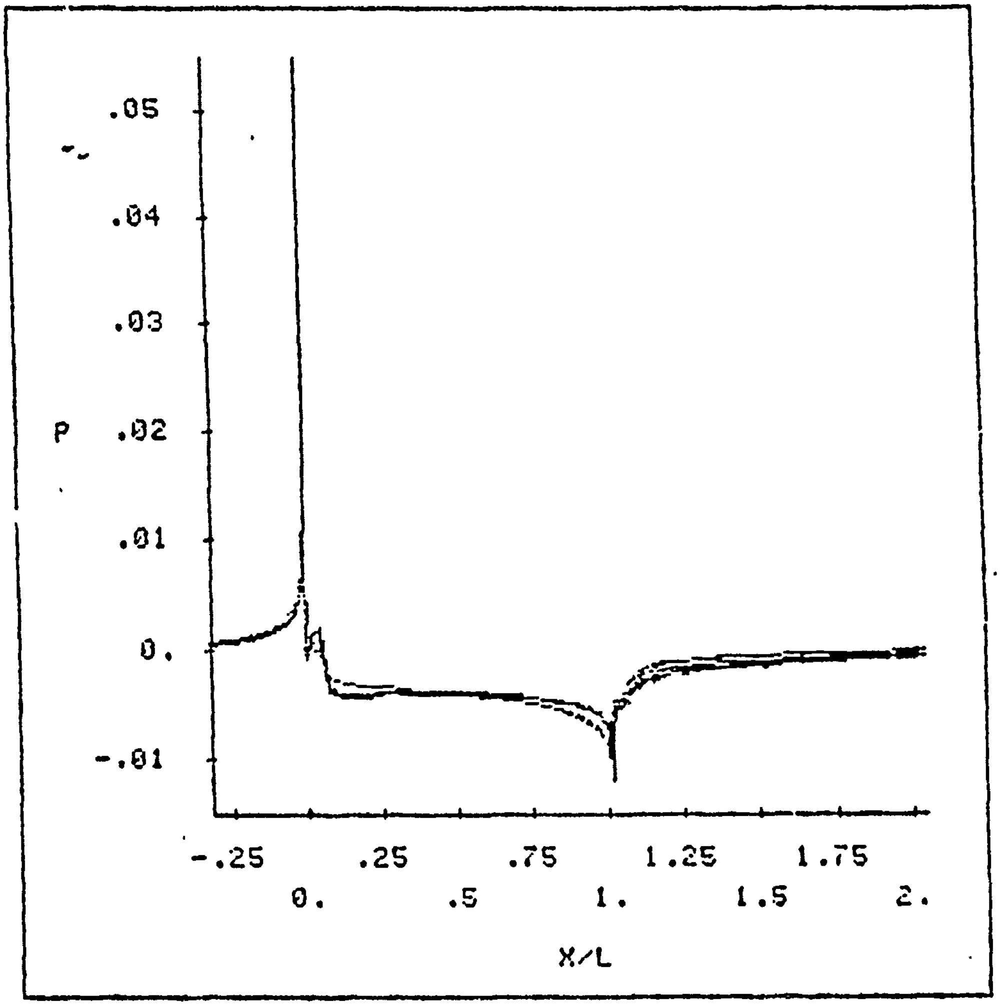


Figure 31. Pressure Distribution on the Entire Plate

when 15 nodes were used in the computation. Also a much smooth pressure distribution is obtained in figure 31 when compared to that in figure 29. Figure 32 gives an exergarate pressure distribution along the centerline of the plate. It shows a slight fluctuation close to the leading edge. This is perhaps affected by the velocity change from the laminar flow to the turbulent flow. The starting point of turbulent is predicted at x/L=0.067 or local Reynolds number $Rex=1.68x10^5$ as shown in figure 33. During the transition the skin friction C_f jumps from 0.0015 to 0.0047. Examining one point of the skin friction C_f measured by Ramaprine, Patel and Sastry [57] around the trailing edge, it shows a little difference between the experimental data and the present result. The convergence history of centerline velocity along the wake is shown in figure 34. Again it is compared with the experimental data published by Ramaprine, Patel and Sastry [57]. It shows that the prediction based on 19 grid nodes along the y direction now gives good agreement result at far wake. This also can be checked from figure 35(a) to 35(g) at different cross section. Figure 36(a) to 36(f) show the kinetic energy profile at different cross section. The comparison between the experimental data and the predicted result shows that the distribution profiles are very similar to each other. Slightly lower values under the experimental data are predicted by the present result.

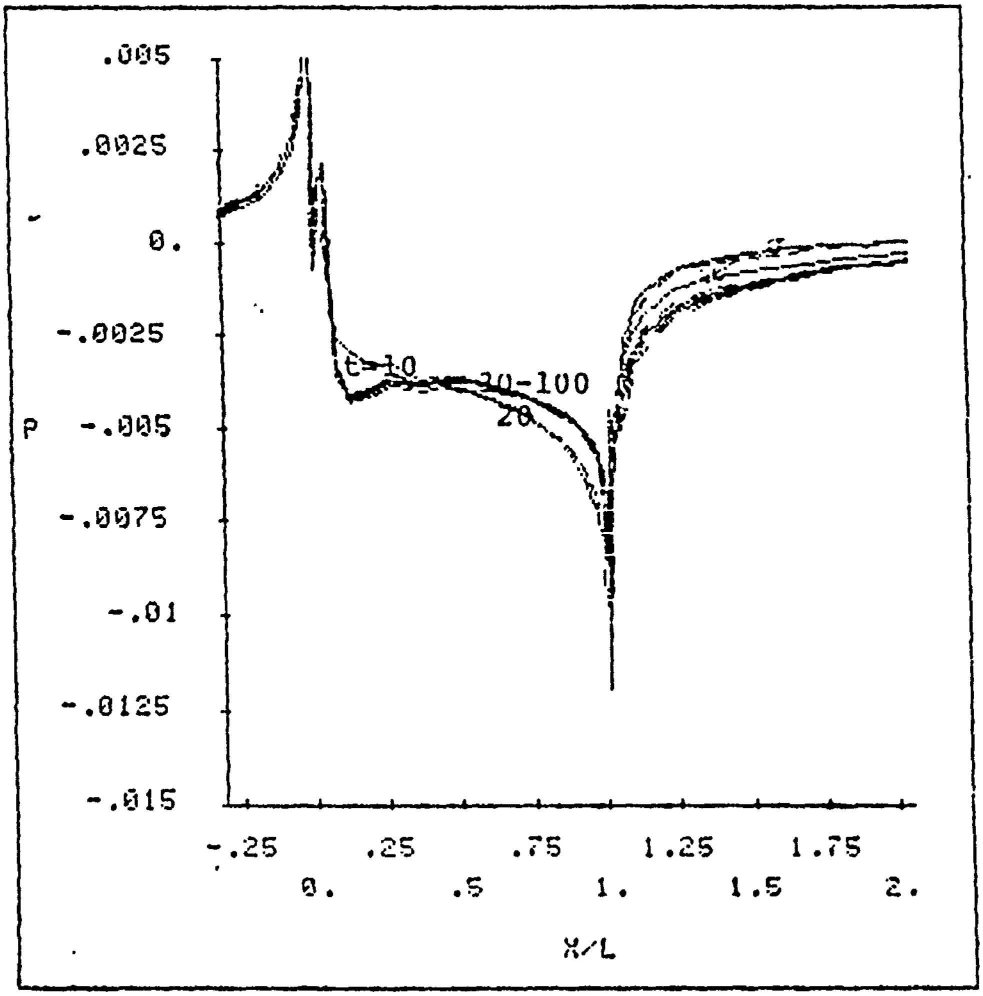


Figure 32. Exergarate Pressure Distribution Along The Centerline Of The Plate

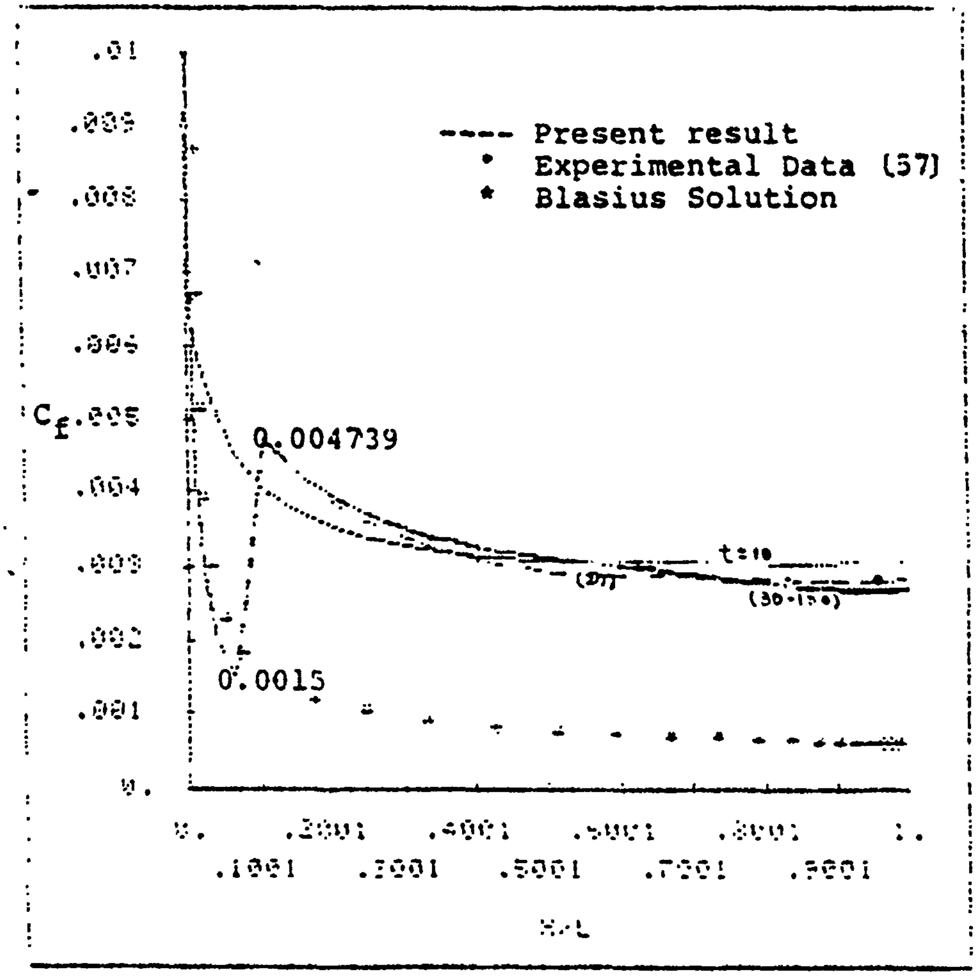


Figure 33. Convergence History of
Skin Coefficient on the Plate

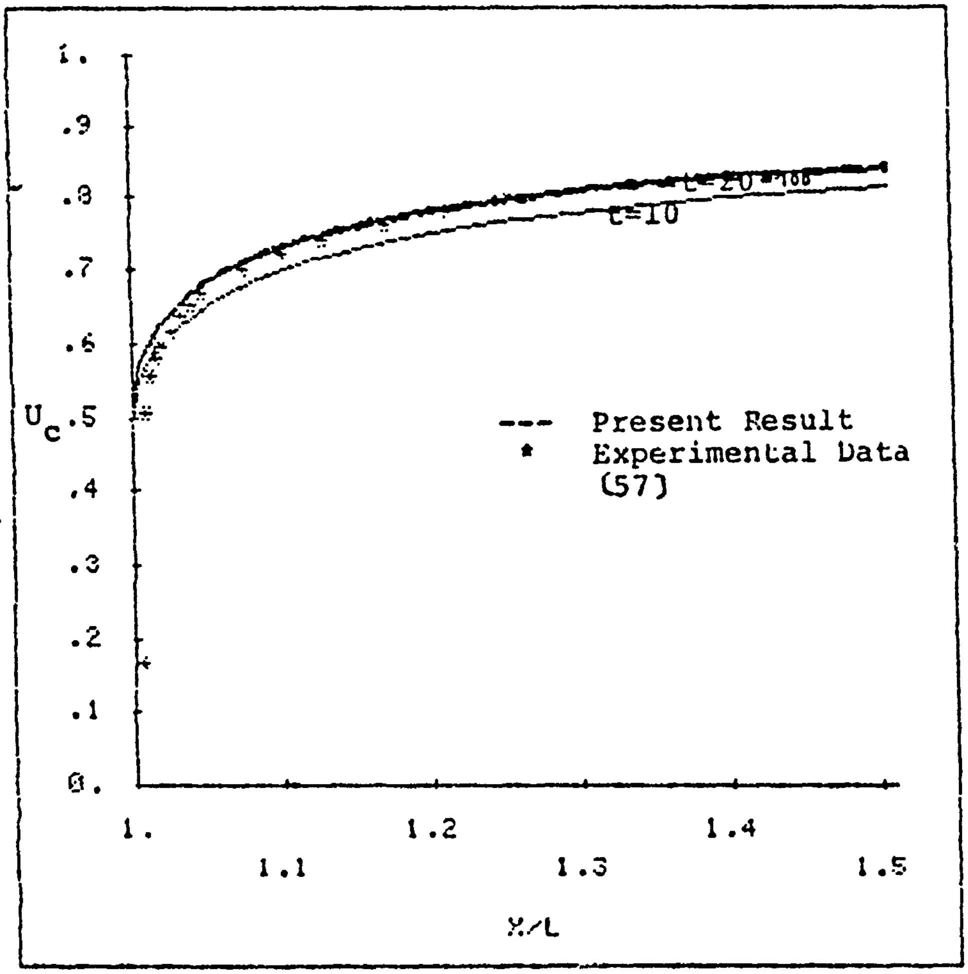


Figure 34. Convergence History of Centerline Velocity in the Wake

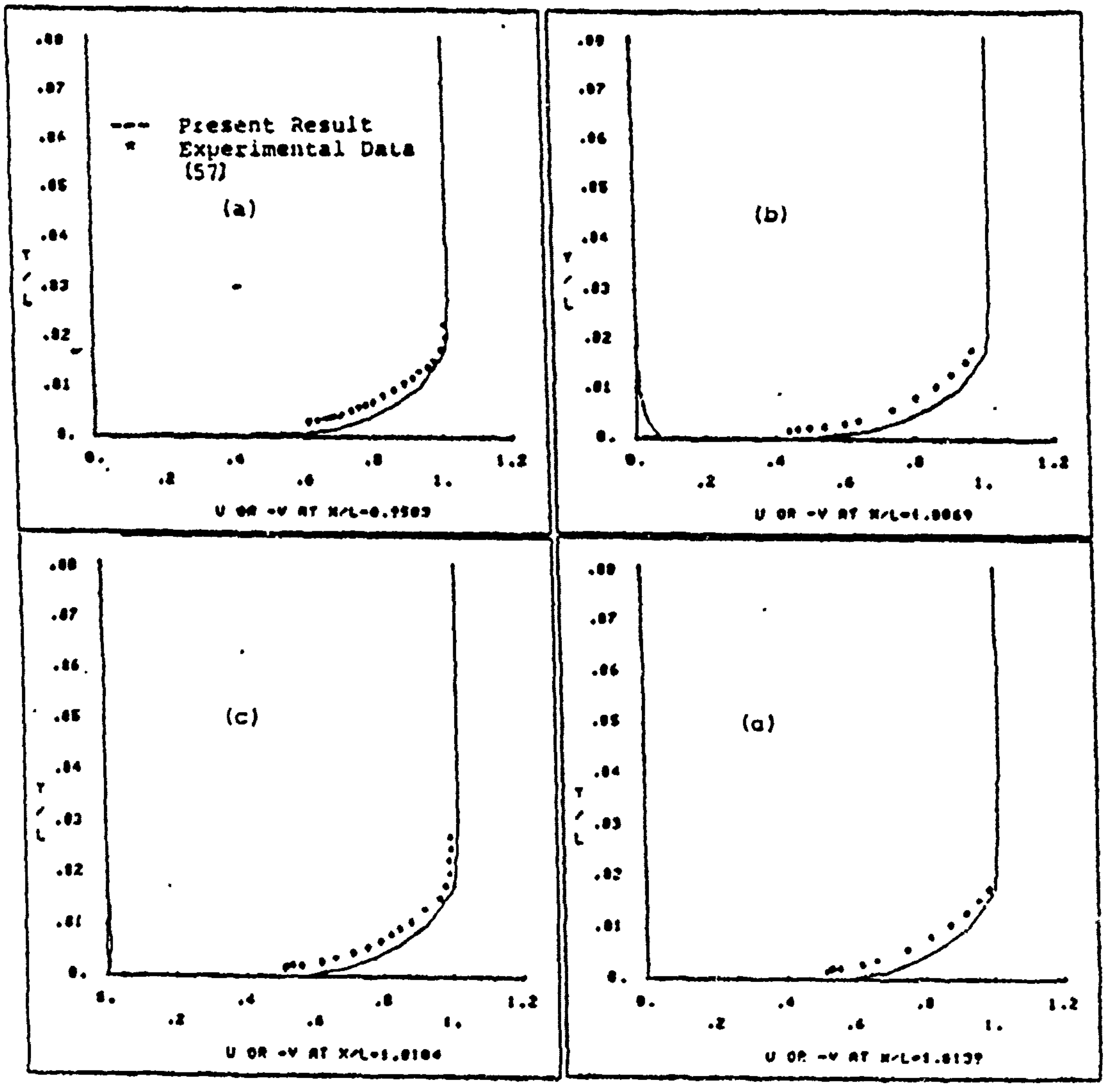


Figure 35. The Velocity Profile At Different Station

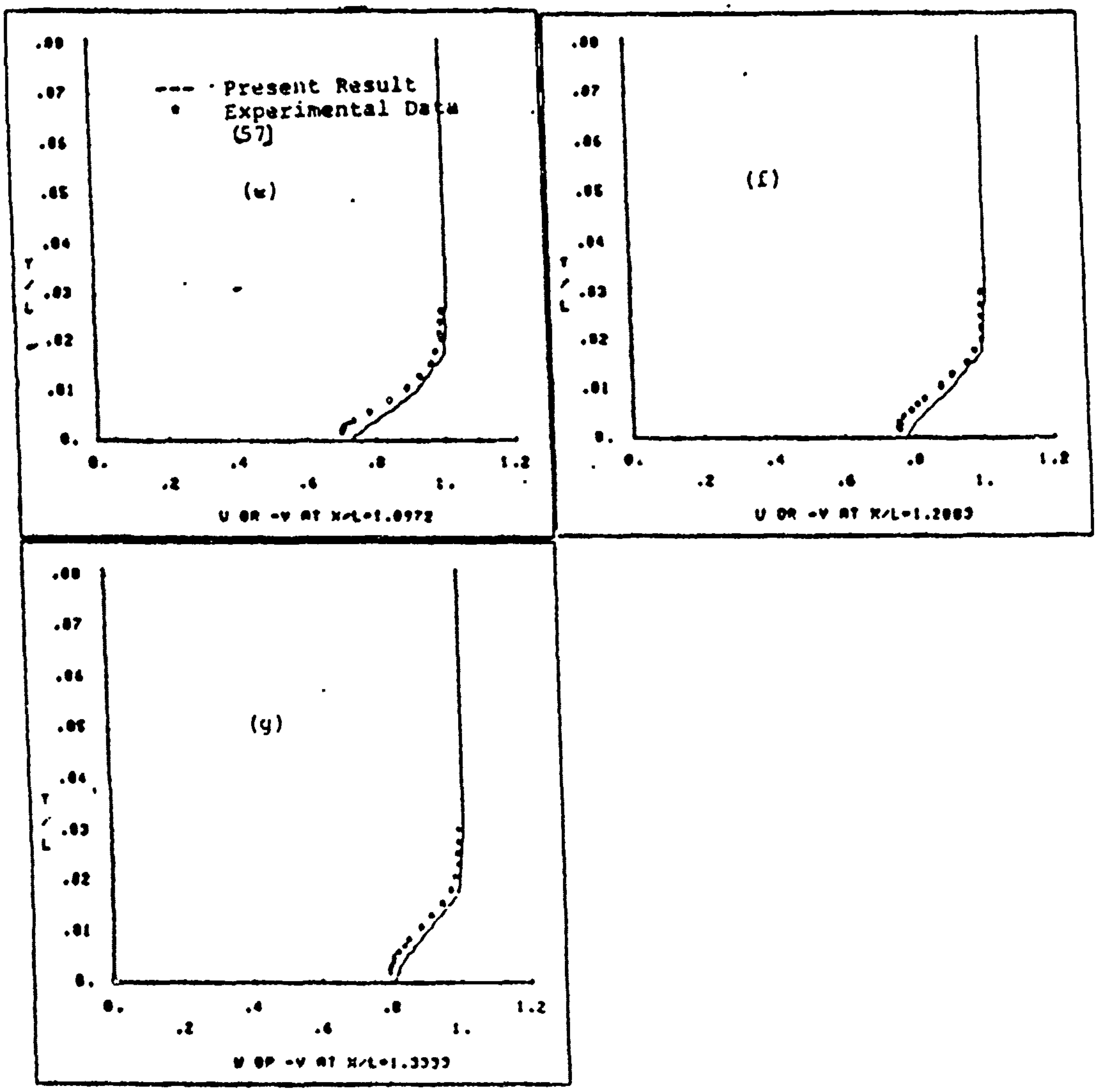


Figure 35. Continued

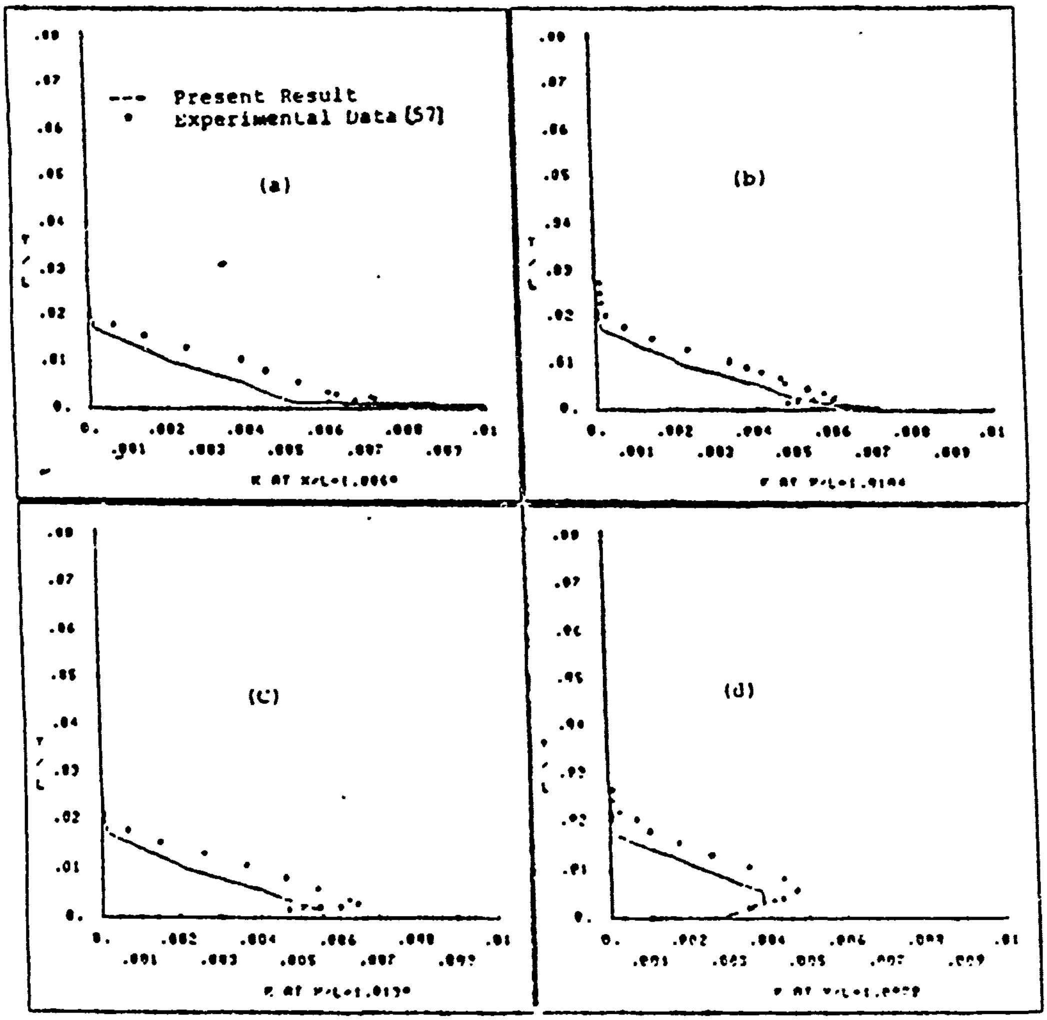


Figure 36. Kinetic Energy Profile at Different Station

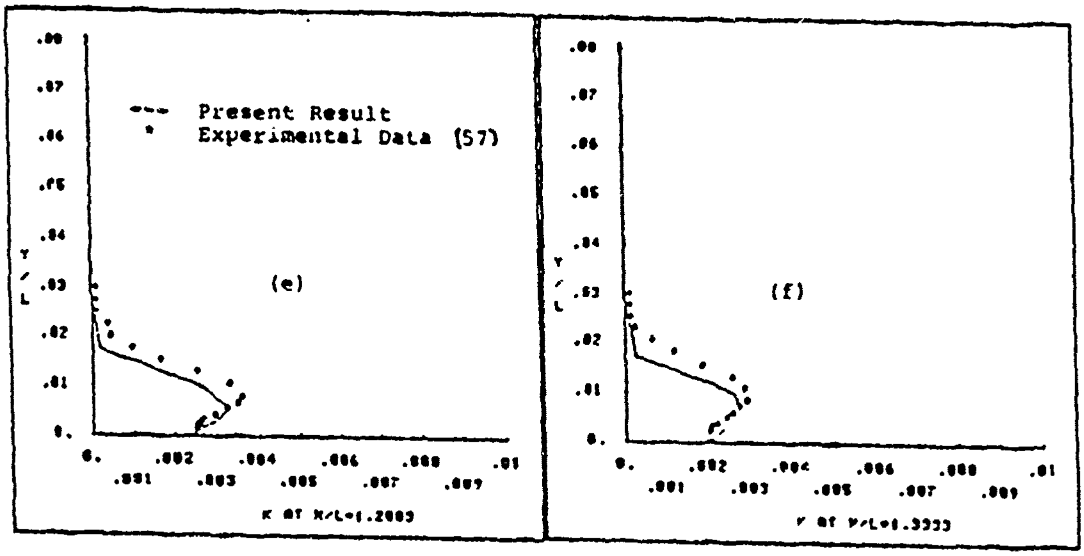


Figure 36. Continued

From the above predictions, one can see that the numerical model of transition for the turbulent flow over the flat plate without angle of attack is able to predict a good result when compared to the experimental data. Therefore, this transition model is used for the flow with angle of attack in the next section.

4.6 Turbulent Flow With Angle Of Attack

In this section the turbulent flow past a finite flat plate with two angles of attack, namely, α=5 and 10 both at Reynolds number Re=2.48×10^6 are solved. Since the symmetric condition is no longer applied for this case, the computational domain needs to be redefined with extended boundaries in the y direction as shown in figure 37. The same numerical grid generation constants A1, A2 and A3 used in the last section for 19 grid nodes are used here again but with the outer boundary in the y direction extended to y=+3 at η=21. Thus a 82x41 mesh was used to cover the physical region that extends from a distance 1.0619L upstream of the leading edge to 8.1406L downstream of the trailing edge and 3L distance normal to the plate on both upper and lower boundaries. The same boundary conditions used in section 5.3 for the laminar flow with angle of attack over the flat plate are used here again. The same numerical modeling used in the last section for the determination of transition from laminar to turbulent flow is used for the present calculation on both upper and lower sides. The FANS-3DEF program with time step $\Delta t=0.1$ and total 100 time steps for both 5 and 10 degree angles of attack are solved.

4.6.1 5 Degree Angle Of Attack

Figure 38 shows the streamline distribution around the flat plate at 5 degree angle of attack. At this high Reynolds number flow $Re=2.48x10^6$ no separation zone at the leading edge on the upper side of the plate is found. It should be remarked that the same problem at $Re=10^4$ solved in section 4.3.1, shows a small separation zone at the leading edge on the upper side of the plate. This can be explained because when the Reynolds number is increased at this small angle of attack the length of the separation zone is decreased until it disappears completely.

The convergence history of the pressure distribution on both the upper and lower sides of the plate is shown in figure 39. It shows that the pressure distribution is monotonically convergent about 30 time steps on the lower side and 40 time steps on the upper side of the plate. The pressure value on the lower side of the plate starts from a maximum pressure $p=0.24$ at the leading edge drops to $p=-0.16$ at the trailing edge, while the pressure value on the upper

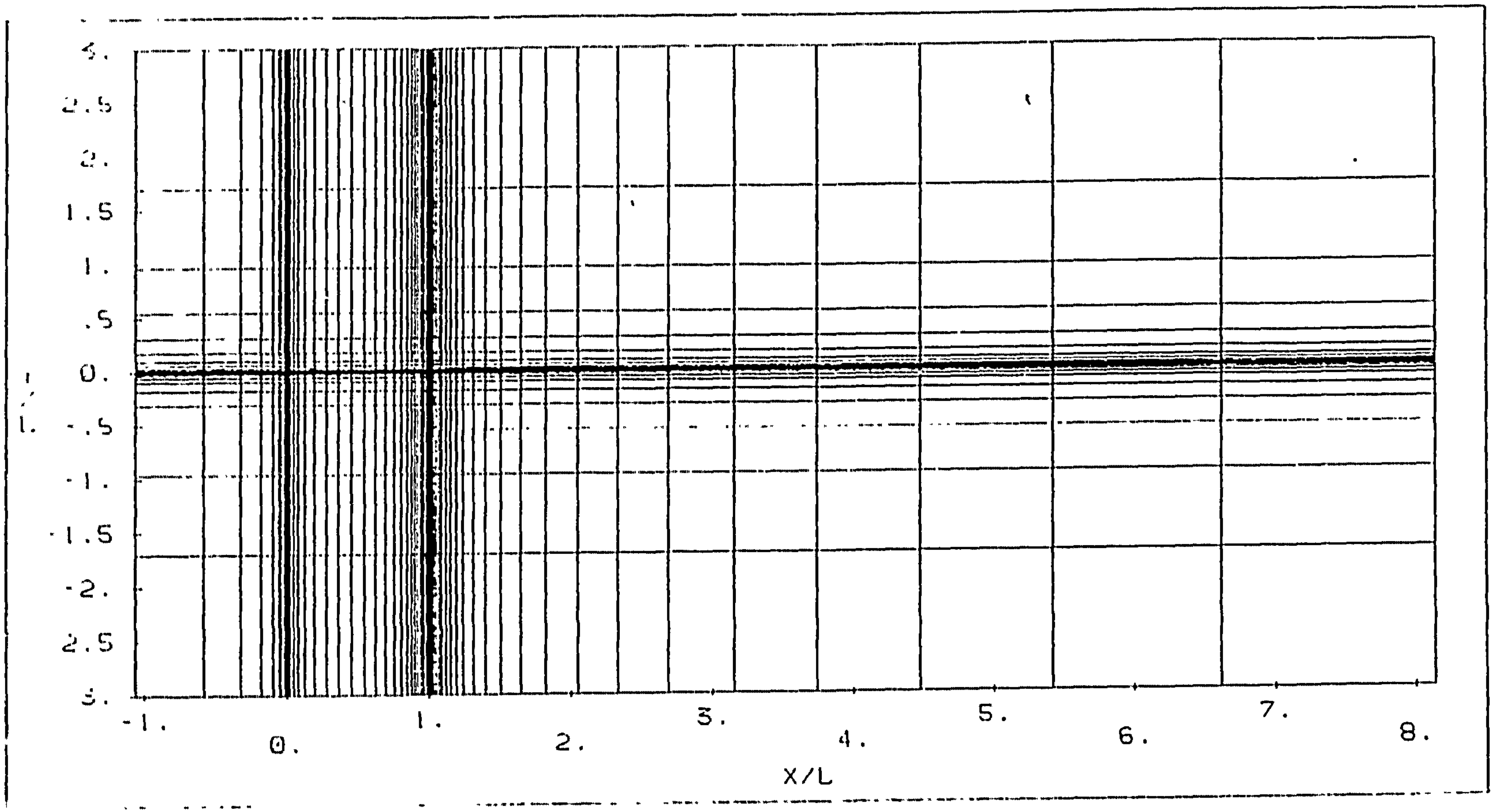


Figure 37. The Numerical Grid For Turbulent Flow With Angle Of Attack

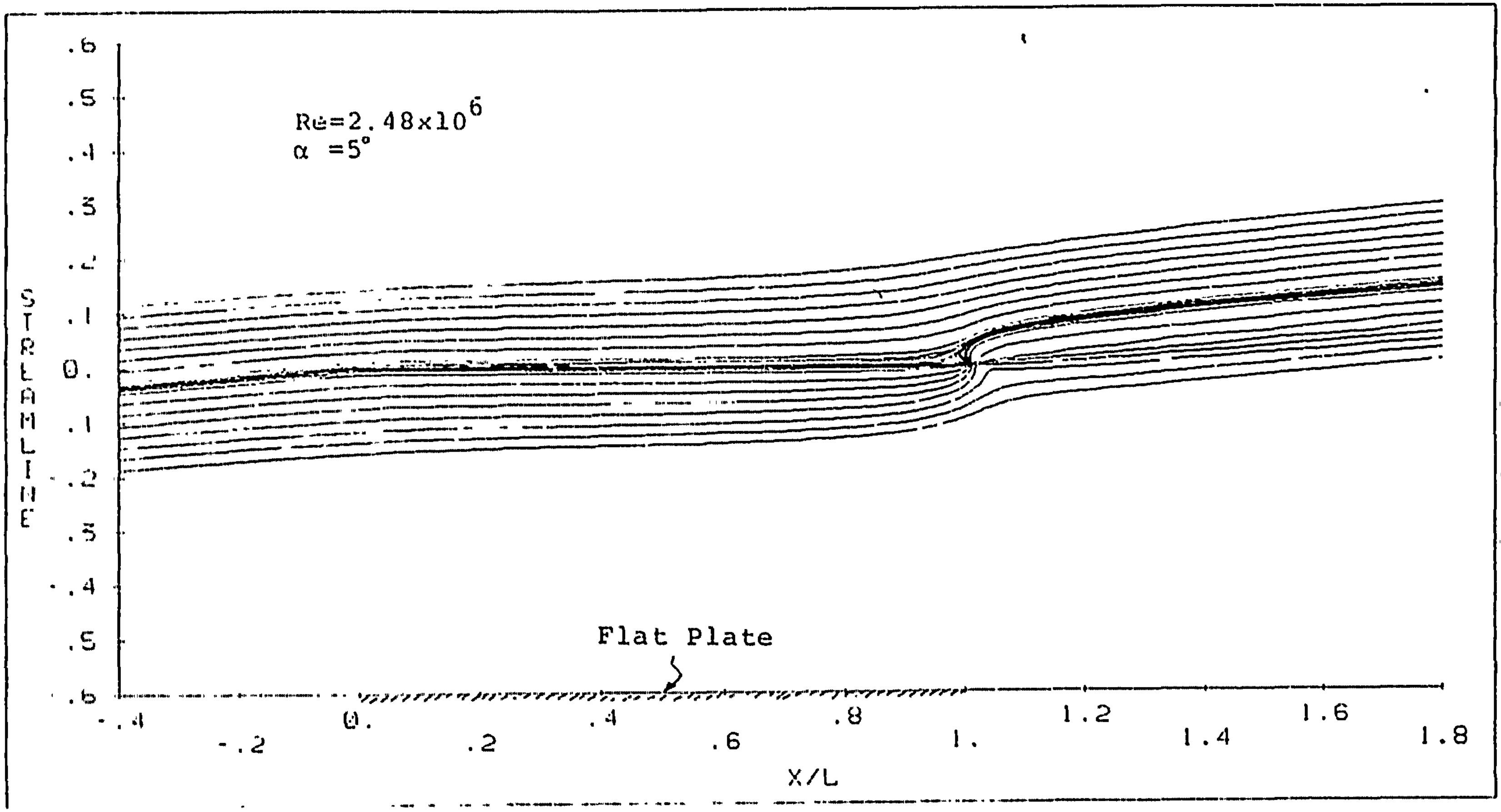


Figure 38. The Streamline Distribution On The Whole Plate

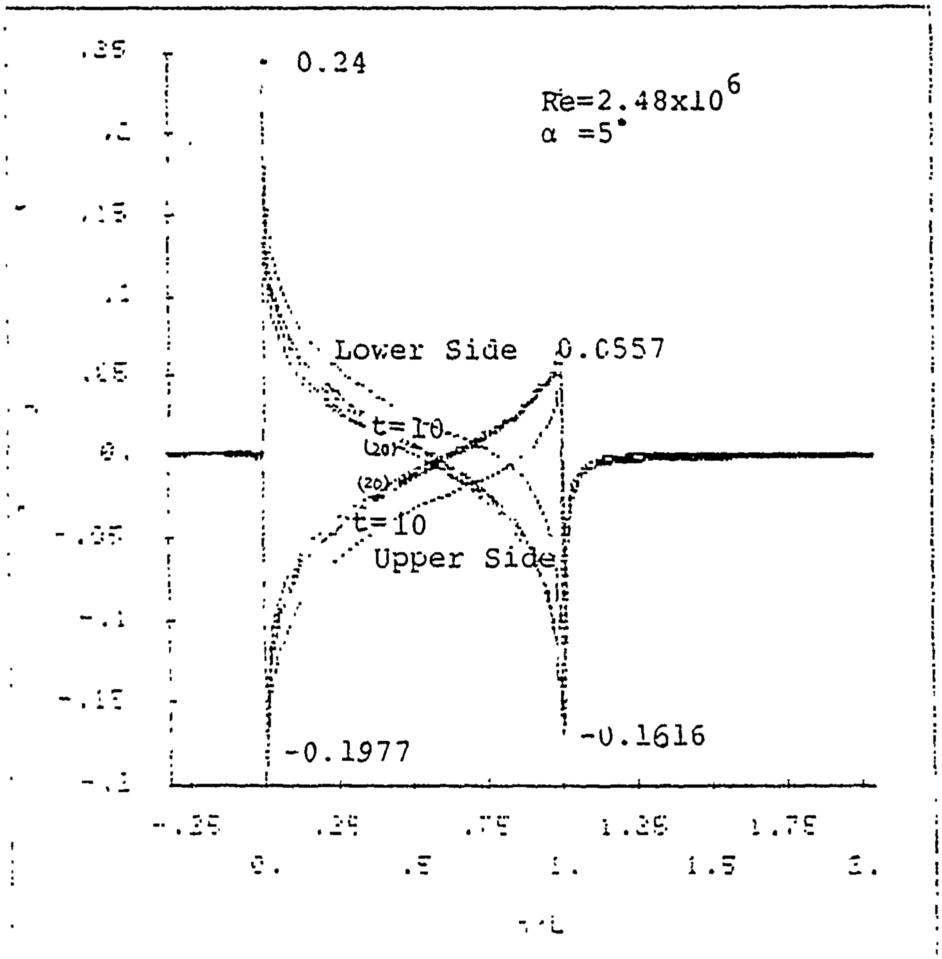


Figure 39. The Pressure Distribution on Both Upper and Lower Side Plate

side of the plate starts from a minimum pressure p=-0.198 increases to p=0.0557 at the trailing edge. A comparison of the pressure distribution given in figure 39 for turbulent flow $Re=2.48 \times 10^6$ and that shown in figure 18 for the laminar flow $Re=10^4$, shows that there is a mark difference between these two flows in the pressure drop at the trailing edge. A larger pressure difference between the lower side and upper side is observed at the trailing edge of the plate for turbulent flow at $Re=2.48 \times 10^6$, or p=0.0557 on the upper side and p=-0.16 on the lower side, while it is almost the same value p=-0.02 for the laminar flow $Re=10^4$. The difference in turbulent and laminar flows can also be seen in the streamline distributions given in figures 16 and 38. Figure 16 for laminar flow $Re=10^4$ shows that the zero streamline has a 5 degree angle of attack to the plate and leaves the plate at 5 degree too, while figure 40 for turbulent flow $Re=2.48 \times 10^6$ shows that the zero streamline has a 5 degree angle of attack to the plate and leaves the plate almost at 90 degrees then decreases sharply and becomes 5 degrees again in the far wake. With the carefully examination there is a small separation around the trailing edge at $Re=2.48 \times 10^6$ with a 5 degree angle of attack. Figure 40 shows the convergence history of the skin coefficient $C_f(2\tau_\omega/\rho U_o^2)$, while using the same numerical modeling of the transition zone for both the upper and lower sides of the plate. The starting point of turbulent flow is predicted at x/L=0.067 on the lower side of the plate and at x/L=0.165 on the upper side of the plate. The skin coefficient C_f on the lower side starts a sharp drop at the leading edge and then jumps from 0.00155 at x/L=0.067 to 0.004341 at x/L=0.108 where the flow becomes fully turbulent and C_f gradually decreases till close to the trailing edge. Then there is a sudden increase in C_f at the trailing edge to a value 0.004316. The skin coefficient C_f on the upper side of the plate also drops sharply from the leading edge to 0.001 at x/L=0.067 then decreases slowly to 0.00922 at x/L=0.165. From x/L=0.238 where the flow becomes fully turbulent to the trailing edge the friction coefficient C_f gradually decreases, with no sudden increase around the trailing edge is found. The different behavior of the skin coefficient at the trailing edge on both the upper and lower sides of the plate can also be explained from the behavior of the streamline pattern shown in figure 38. In figure 38 one observed that the upper zero streamline is almost 90 degrees when it leaves the upper trailing plate into the wake while the zero streamline on the lower side converges to the trailing edge parallel to the plate.

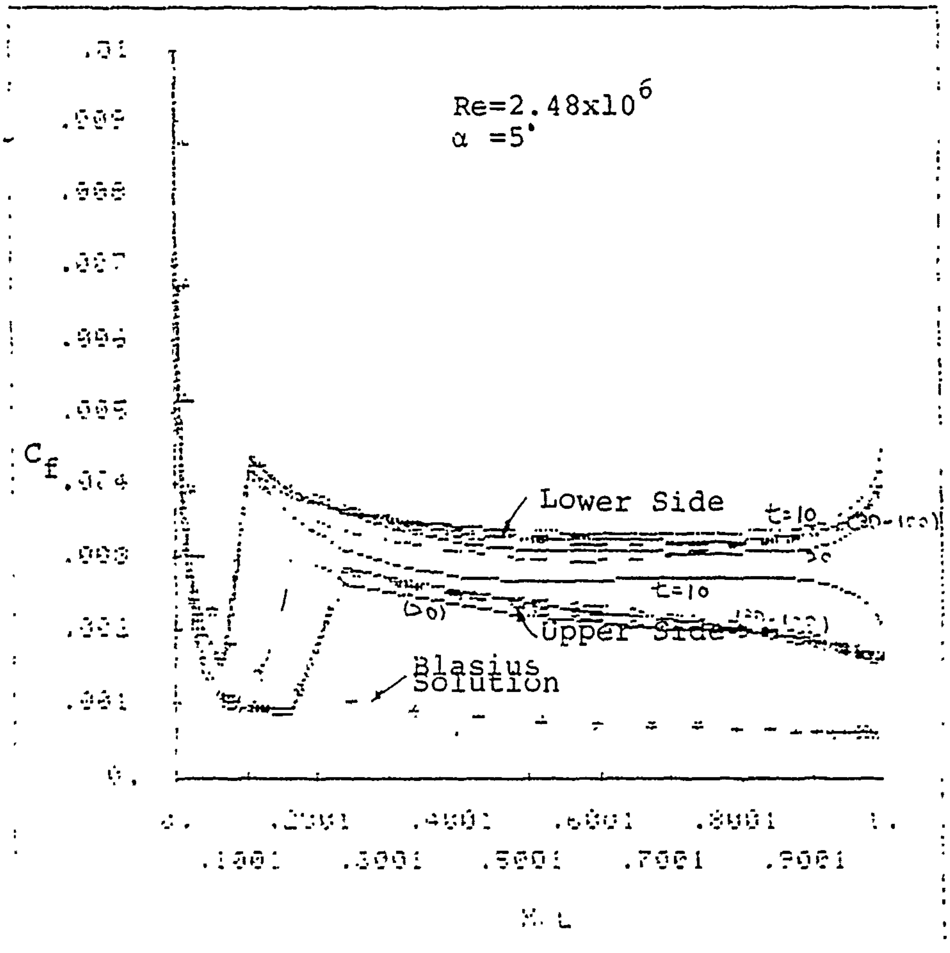


Figure 40. Convergence History of Skin Coefficient on Both Upper and Lower Side of Plate

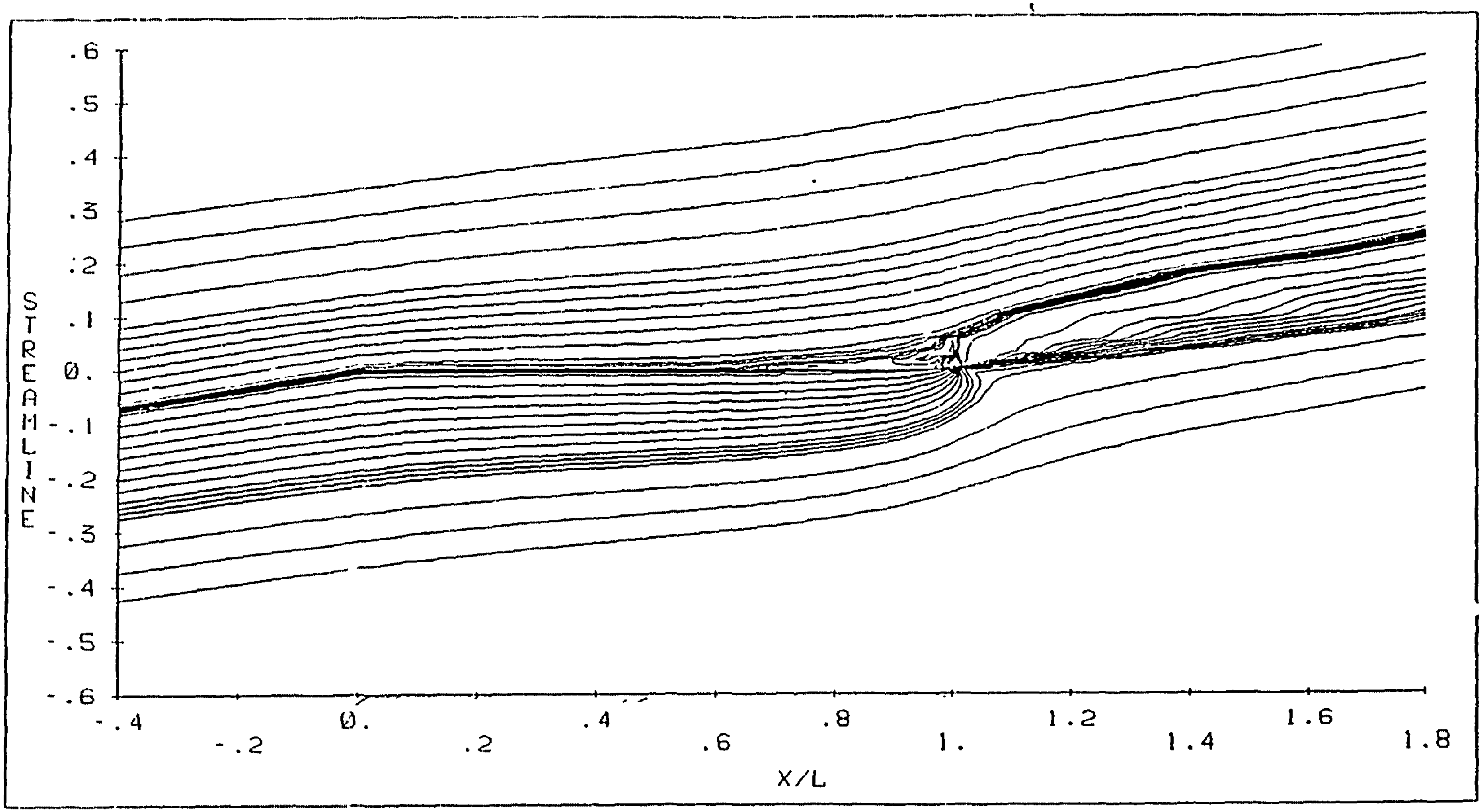


Figure 41. Streamline Distribution At 10 Degree Angle Of Attack

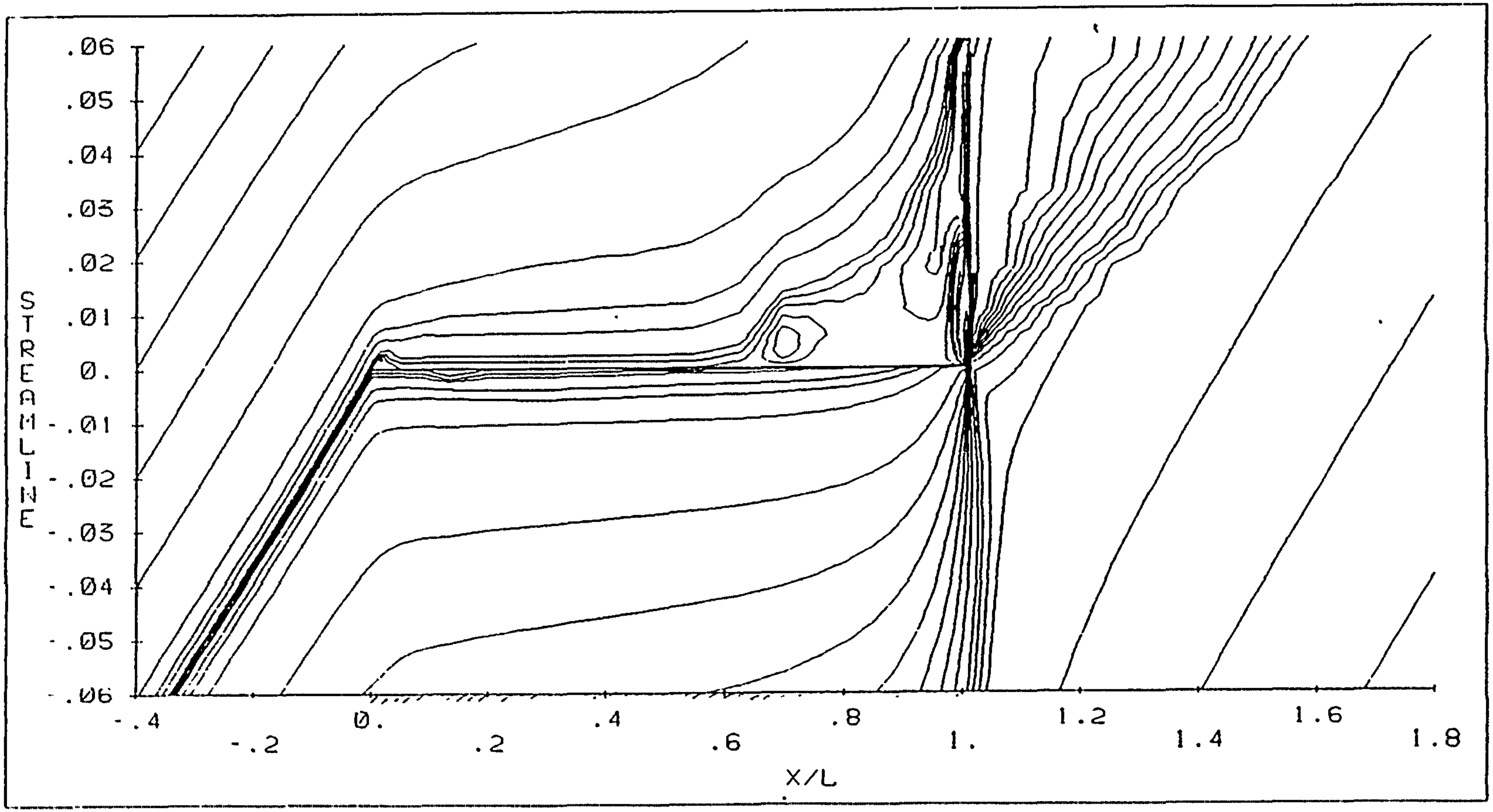


Figure 41. Continue

4.6.2 10 Degree Angle Of Attack

Figures 41 to 45 give the solution of a 10 degree angle of attack for Re=2.48×10^6. The flow patterns shown in figure 41 to 45 have many similarities with the flow patterns for 5 degree angle of attack shown in figures 37 to 40. However there are some differences. The following are some different points which require explanation. At this high angle of attack $\alpha=10$ one finds that a very small separation exists at the leading edge, also a strong separation around the trailing edge on the upper side of the plate is found as shown in figure 41. Like the flow with a 5 degree angle of attack, figure 42 shows the pressure distribution at the trailing edge of the plate has a large pressure difference between the upper and lower sides of the plate. The pressure distribution on both the upper and lower sides of the plate is similar to that of the 5 degree angle of attack except that the magnitude is higher for the 10 degree angle of attack. The convergence for a 10 degree angle of attack is slower as shown in figure 43. The convergence history of the skin coefficient given in figure 44, shows that the solution is convergent after 80 time steps. Figure 45 also shows a sudden decrease of skin coefficient at $x/L=0.68$ in the fully turbulent flow on the upper side of the plate. This sudden decrease may correspond to the strong separation on the upper side of the plate as shown in figure 41.

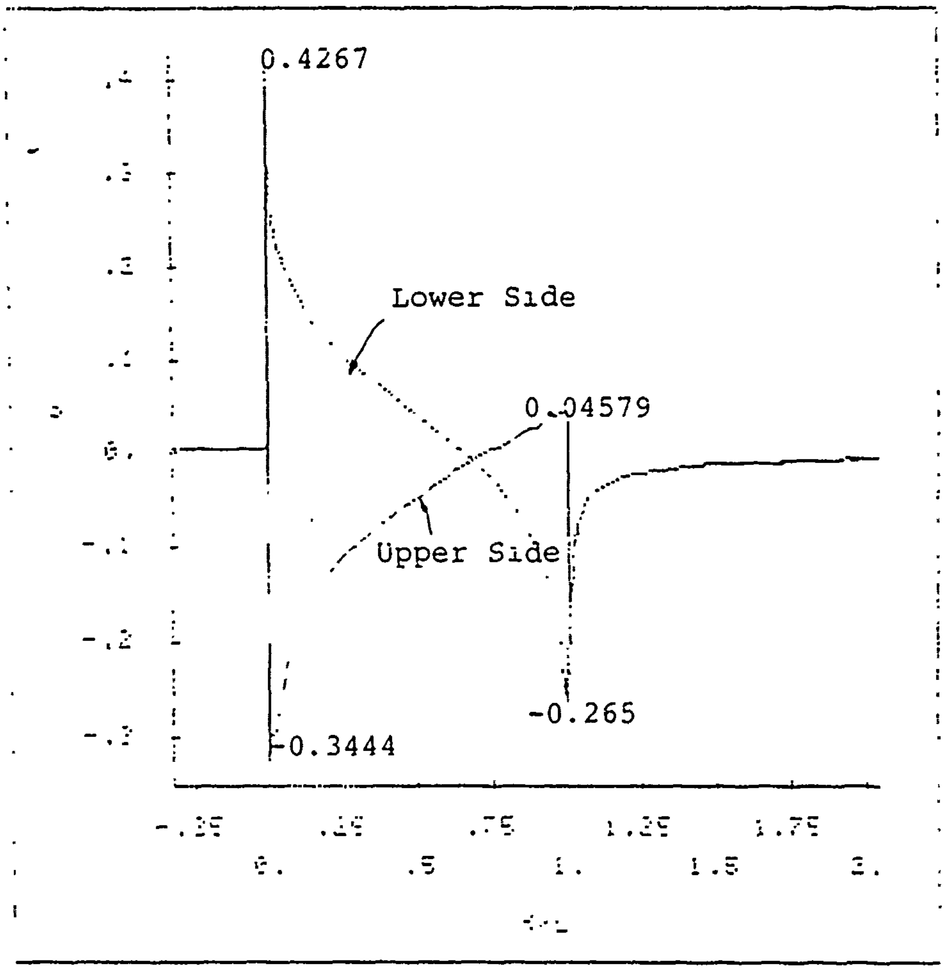


Figure 42. Pressure Distribution at 10 Degree Angle of Attack

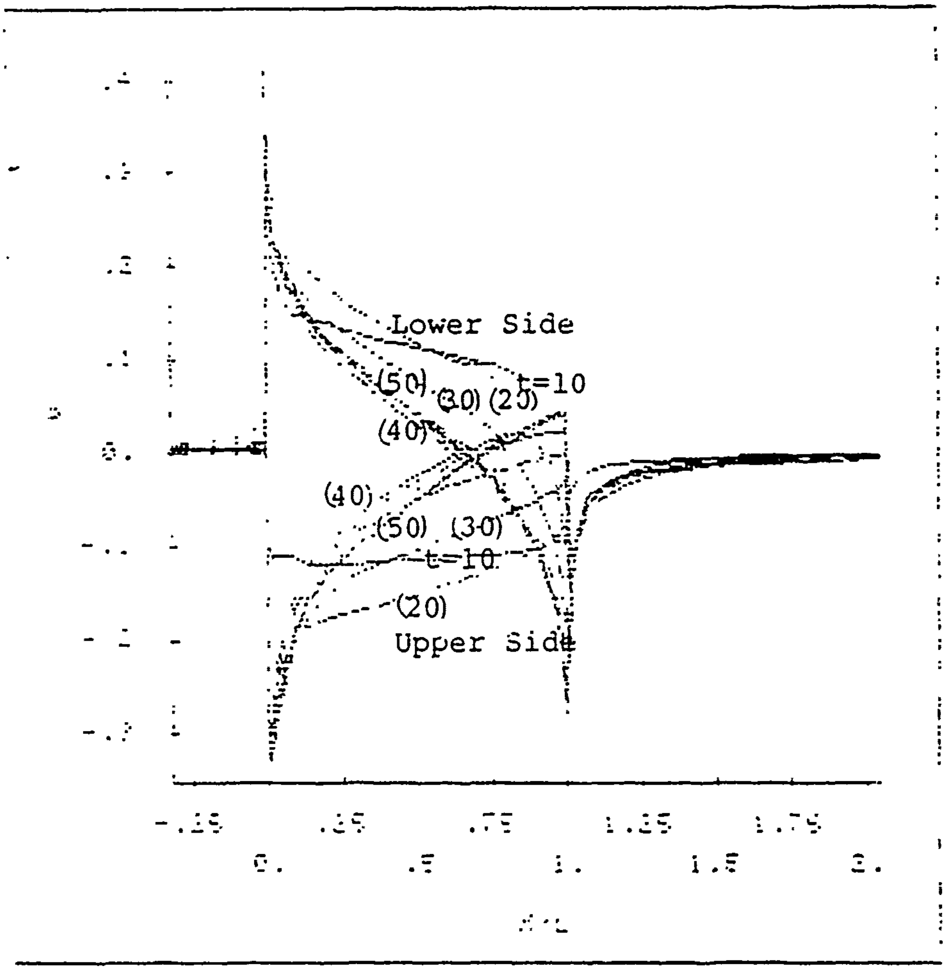


Figure 43. Convergence History of Pressure Distribution on Both Upper and Lower Side

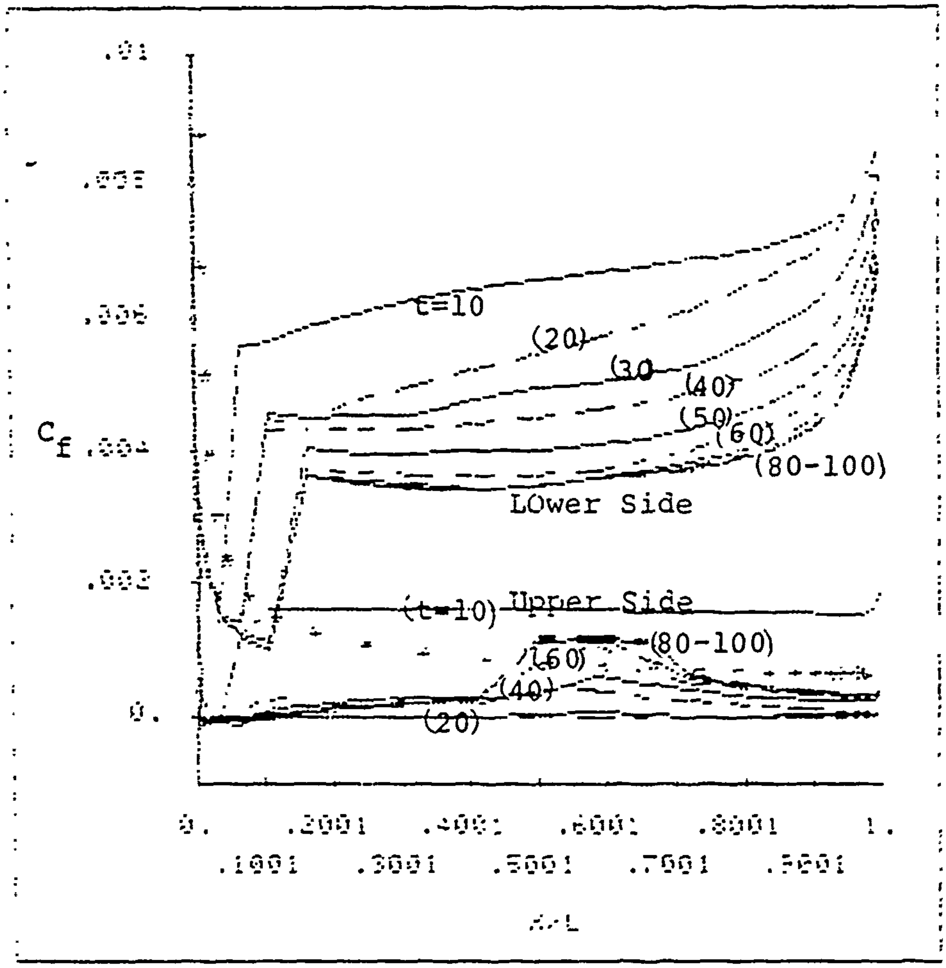
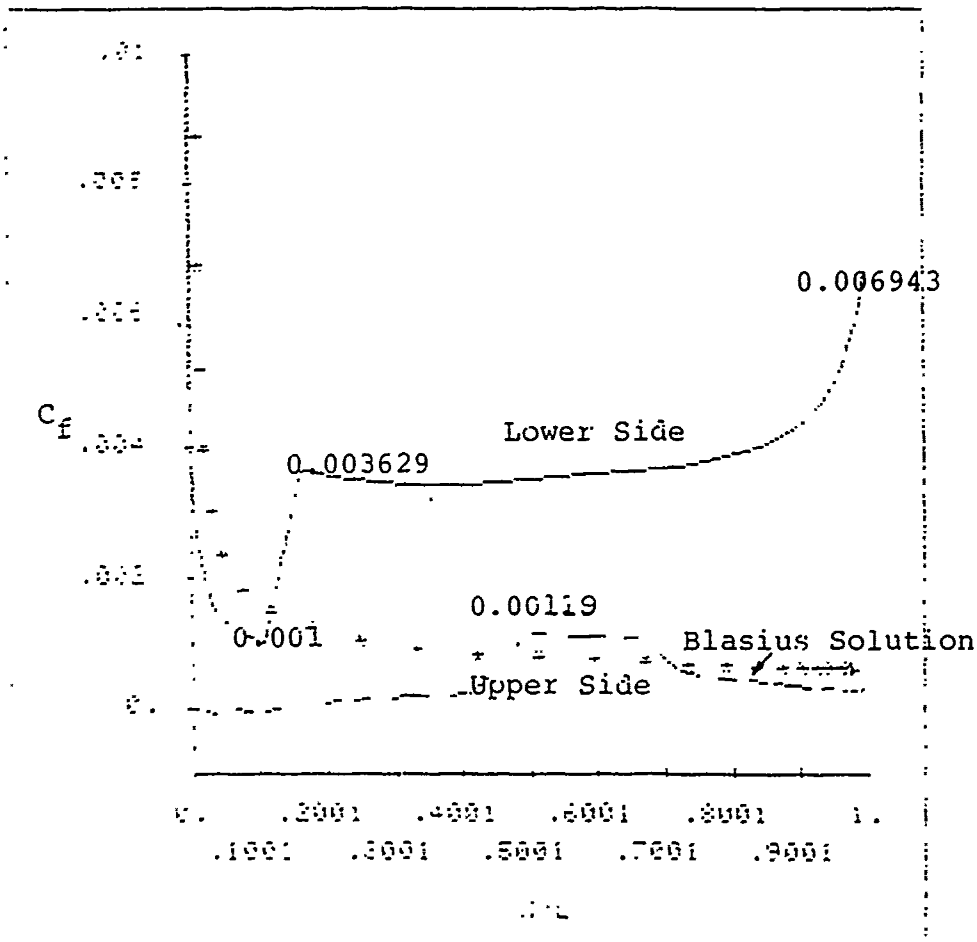


Figure 44. Convergence History of Skin Coefficient on Both Upper and Lower Side



Figur 45. Skin Coefficient At 10 Degree Angle Of Attack

From the solutions presented in the last section and in this section, one may conclude that the proposed criteria for predicting the transition provide a reasonable and accurate solution. Also the FANS-3DEF is a stable program and can predict good results for the case of zero incidence and reasonable sol tions for the flows with angle of attack. This application is further extended to the flow over an inclined axisymmetric body in chapter 6.

CHAPTER V

FLOW PAST AXISYMMETRIC BODY WITHOUT ANGLE OF ATTACK

In the last chapter the FA numerical solution for laminar and turbulent flows over a finite flat plate with and without an angle of attack had been solved by the FANS-3DEF program. Therefore the FANS-3DEF code is verified at least for prediction of flows past a finite flat plate at varied angles of attack using the body-fitted coordinate transformation and FA method in solving turbulent Navier-Stokes equations with the k-ε turbulence model. In this chapter the turbulent flows past more complicated axisymmetric bodies for which detailed experimental data are available are predicted by the FANS-3DEF. Two bodies were chosen because of their importance in ship hydrodynamic study and availability of experimental data. The first geometry is known as "Afterbody 1" used by Huang et al. [10] who provide detailed measurements of velocity, pressure and turbulent quantities arround the rear part of the body. The second geometry is "F-57 body" used by Lee [9] in his study of turbulent flow past the body. The geometries of these two bodies are shown in figure 46. These body shapes are described by an analytic equation and detailed measurements. In this chapter the k-ε turbulence model is used to obtain the numerical results and a comparison is made with the experimental data.

5.1 Numerical Grid System

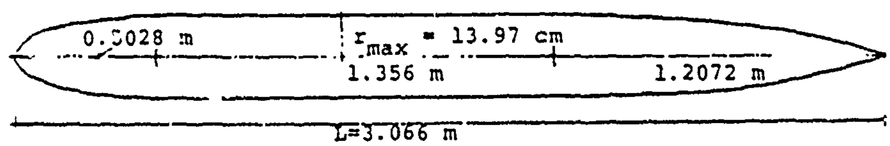
In the FANS-3DEF program, the body fitted coordinate system is again used to generate the grid nodes for axisymmetric bodies. To minimize the possible approximation error in the pressure equation Eq. (17) as described in section 3.3, the staggered grid system where the constant ξ stations is a sole function of the x coordinate or $\xi = \xi(X)$ is used in this study. Under this arrangement, Eq. (9) can then be rewritten in the cylindrical polar coordinates as

$$\begin{aligned} &\alpha_{11}x_{\xi\xi} + J^2(F1X_\xi) = 0 \\ &\alpha_{11}r_{\xi\xi} + \alpha_{22}r_{\eta\eta} + 2\alpha_{12}r_{\xi\eta} + J^2(F1r_\xi + F2r_\eta) = \frac{J^2}{r} \end{aligned} \tag{24}$$

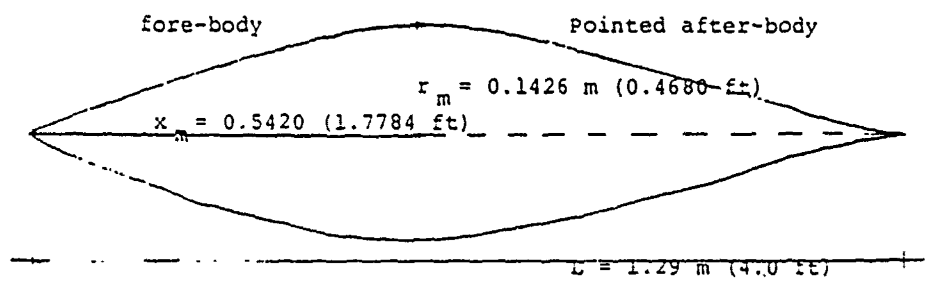
where

$$\alpha_{11} = r^2(x_\eta^2 + r_\eta^2), \quad \alpha_{22} = r^2(x_\xi^2 + r_\xi^2),$$
$$\alpha_{12} = -r^2(x_\xi x_\eta + r_\xi r_\eta), \quad J = r(x_\xi r_\eta - x_\eta r_\xi)$$

The control function F1 is therefore determined by the desired distribution of the axial station or $F1 = -(\alpha_{11}x_{\xi\xi})/(J^2 x_\xi)$. With F1 specified, equation (24) yields the distribution of points in the radial direction, $r(\xi,\eta)$. To obtain the desired grid distribution in the r direction, the control function F2 must be prescribed. If the control function F2 is set equal to



(a) Afterbody 1



(b) F-57 Body

Figure 46. The Geomery of Axisymmetric Bodies

$$F2 = \frac{1}{rr_\eta} + f2(\xi,\eta)$$

then Eq. (23) can be rewritten as

$$\alpha_{11}r_{\xi\xi} + \alpha_{22}r_{\eta\eta} + 2\alpha_{12}r_{\xi\eta} + J^2(F1r_\xi+f2r_\eta) = 0$$

which is equivalent to the two-dimensional body fitted coordinates for the Cartesian coordinates (x,y) as given in Eq. (22) with the control functions F1 and f2. In other words, the same grid distribution can be generated in both the cylindrical and Cartesian coordinates if the control function F2 is replaced by f2 in the cylindrical formulation.

In this chapter the FANS-3DEF progrm is used to solve the flow over a more complicated axisymmetric body with the k-ε turbulence model. Since the experimental measurements of Huang et al. [10] on "Afterbody 1" and Lee [9] on the "F-57 body" provide data only at the rear part of body to the wake, the computational domain is chosen from the half part of the body to the far wake. As the body shape and computational domain are different from the flat plate problem the distribution of control factor a ($a = -(J^2F_1)/(2\alpha_{11})$) as shown in Eq. (23) for generation of the grid nodes along the ξ direction is chosen as

$$a = \begin{cases} -A1 & 0.25 \leq z_1 < 0.5 \\ A1\ \mathrm{Sin}(\pi z_1) & 0.5 < z_1 \leq 1 \\ A2\ \mathrm{Sin}(\pi z_1) & 1 \leq z_1 \leq b \\ -A2 & z_1 > b \end{cases} \qquad (25)$$

where $z_1 = \dfrac{\xi}{\xi_2}$

ξ_2 corresponds to the trailing edge, the constant b (>1) is the grid number to be affected by the concentration at the near wake region and A1 and A2 are positive constants for condensing the grid nodes to the trailing edge. For "Afterbody 1" A1=0.05, A2=0.2, b=1.2 and ξ_2=40 at x=1.0 are used. For "F-57 body" A1=0.01, A2=0.2, b=1.1 and ξ_2=40 at x=1.0 are used. Here the grid nodes along the body near the trailing edge and in the near wake region are assigned. To concentrate the grid nodes at the inlet plane, the same concentration values obtained around the trailing edge are used and assigned them to the nodes at the inlet plane. In this study it is set as:

$$a(I) = -a(N_2+I-5) \qquad 1 \leqslant I \leqslant 5$$

$$a(I) = -a(10-I) \qquad 6 \leqslant I \leqslant 9$$

where the number shown in the bracket is the grid number along the ξ direction, I=1 is inlet plane and N_2=40 is the trailing edge of the body. In this study f2 is also defined as

$$f_2(\xi,\eta) = \begin{cases} Fa(\eta) & \xi < \xi_a \\ Fc(\xi,\eta) & \xi_a < \xi \leqslant \xi_b \\ Fb(\eta) & \xi > \xi_b \end{cases}$$

where Fa and Fb are given by the user or deterimed by the node distribution at the initial, $\xi=1$, and final stations, $\xi=n$, as

$$Fa = - \left.\frac{\alpha_{22} r_{\eta\eta}}{J^2 r_\eta}\right|_{\xi=1}$$

$$Fb = - \left.\frac{\alpha_{22} r_{\eta\eta}}{J^2 r_\eta}\right|_{\xi=n}$$

and Fc is obtained by a linear combination of Fa and Fb or

$$Fc(\xi,\eta) = |(\xi_b-\xi)Fa(\eta)+(\xi-\xi_a)F_b(\eta)| \;/\; (\xi_b-\xi_a)$$

In this study $\xi_a=15$, $\xi_b=42$, Fa=0.2 and Fb=0.15 are given for the prediction of flows past the two bodies.

5.2 Afterbody 1

As shown in figure 46(a) the total length of the Afterbody 1, L, is 3.066m and the maximum diameter of the parallel middle body is 27.94cm. The experimental investigation was conducted by Huang et al. [10] in the wind tunnel of the DTNSRDC anechoic flow facility. The common forebody and a portion of the parallel middle body were constructed with wood. The afterbody and the remaining portions of the parallel middle body were constructed with molded fiberglass. The wind tunnel was a 2.44m by 2.44m closed jet test section, followed by a 7.16m by 7.16m open jet test section. In this experiment the velocity of the wind tunnel was held constant at 30.48 m/sec therefore the Reynolds number based on the maximum diameter or $Re=6.6x10^6$ was obtained.

Since in this experiment the velocity profile and turbulent shear stress are measured from x/L=0.706 to x/L=1.182 (in the open jet test section), where x is measured along the axis of the body from the body nose and L is the body length, the prediction of flow was made for the latter half of the body. The calculations for Afterbody 1 were performed with 56 stations in the domain 0.364<x/L<6.58. A partial view of the body-fitted coordinates is shown in figure 47. 19 grid nodes were used between the body surface and the external boundary which varies from r/L=0.68 to 0.72. Here r is the radial distance from the body axis. The use of coordinate-stretching functions F1 in the longitudinal direction and F2 in the radial direction ensure that the grid points are closely spaced inside the region of large velocity gradient and near the stern.

The numerical calculation is confined to the domain from x=0.364L at the middle part of the body to the wake region x=6.58L. Since the FANS-3DEF program solves elliptic partial differential equations Eq. (12) the boundary conditions at the boundary of the computational domain must

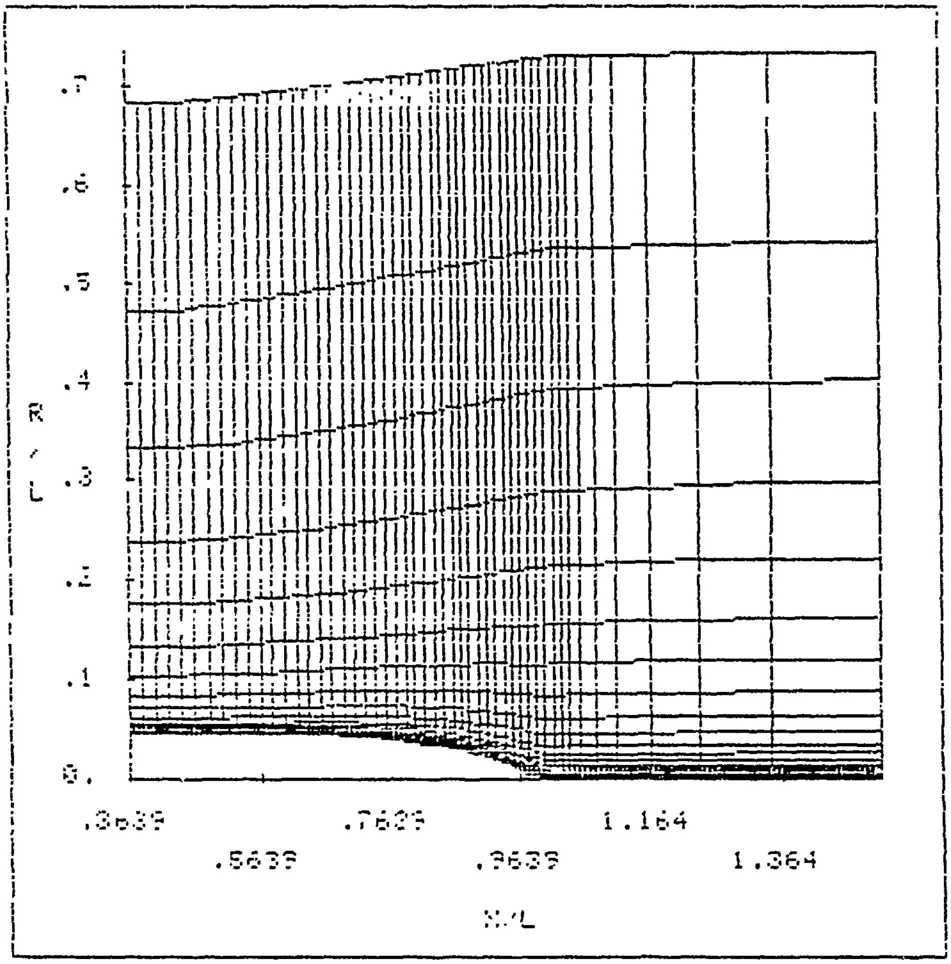


Figure 47. The Partial View of Numerical Grid for Afterbody 1

be given. Afterbody 1 as shown in figure 46 (a) has a long section of middle body (from x/L=1.64 to x/L=0.6064) which is a slender circular cylinder with constant radius r/L=0.0456, therefore it seems reasonable to assume constant ambient pressure, i.e. p=0 as the pressure profile at the upstream boundary condition at x/L=0.364 and use one-seventh power formulation for turbulent velocity. The turbulent profiles for k and at this upstream condition (x=0.364L) are specified too.

$$u(y)=(y/\delta)^{.143} \qquad \text{for } y < \delta$$

$$u(y)=1 \qquad \text{for } y > \delta$$

$$k=.002(1-y/\delta)$$

$$\varepsilon=\frac{(\sqrt{C_\mu}k)^{1.5}}{\kappa y}, \quad C_\mu=.09, \quad \kappa=.42$$

Here δ is the dimensionless boundary layer thickness and is assumed as 0.004 in this study.

The FANS-3DEF program with the k-ε turbulence model was solved with t=1. The total marching steps are 40. Figure 48 shows the convergence history of the dimensionless pressure field, defined by $(P-P_{ambient})/\rho U_0^2$, on the body surface and along the wake centerline.

It is seen that the solution converges monotonically and the converged solution is obtained after 10 time steps. The

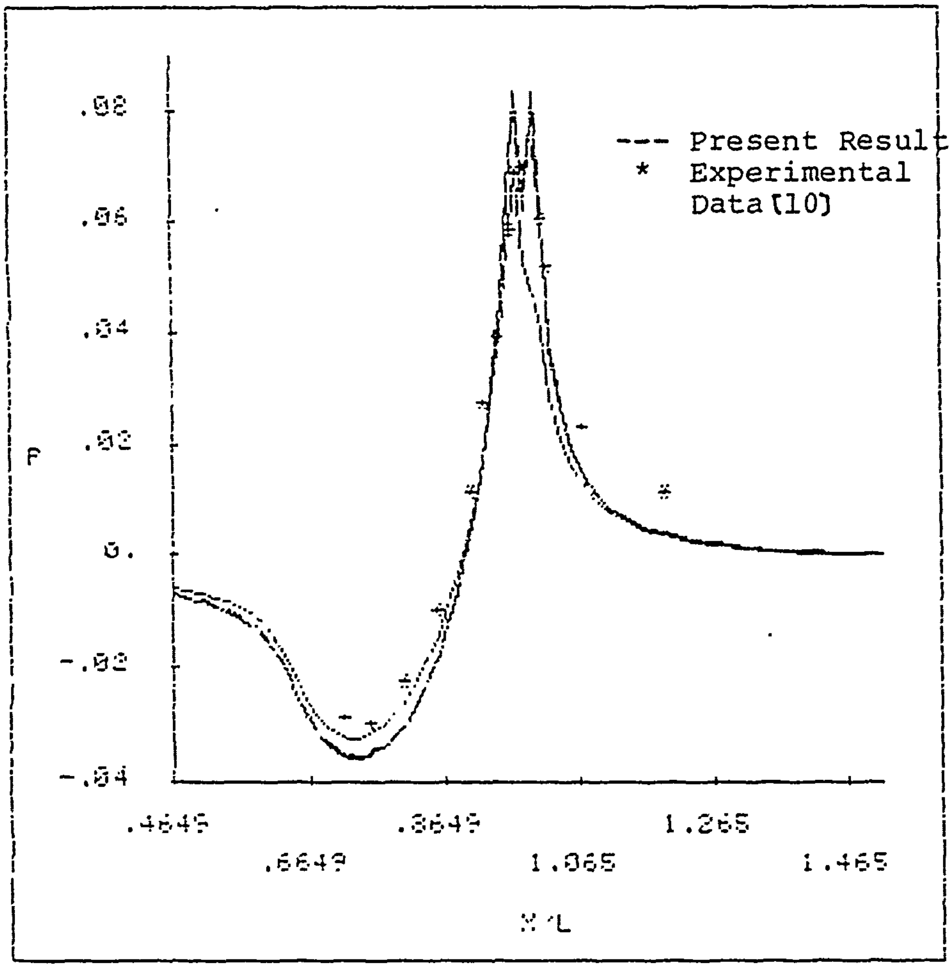


Figure 48. Convergence History of Pressure Distribution on Afterbody 1

The typical computational time for each time step on the PRIME 750 is 16 cpu seconds. Since Afterbody 1 has a long section of constant radius (from x/L=0.164 to x/L=0.6064) before a gradual reduction of radius from r/L=0.0456 to zero radius at the trailing edge, the pressure begins to change from the location x/L=0.6064 and the behavior like the flow over the trailing edge of the flat plate. As the radius of "Afterbody 1" along the axis decreases gradually from r/L=0.0456 at x/L=0.6064 to zero at x/L=1, the pressure is gradually increased due to the deceleration of the flow. Then the pressure along the center line of the wake has to recover the ambient pressure in the far wake. The predicted solution for pressure in figure 48 is in fairly good agreement with the data of Huang et al. [10]. In the wake the pressure along the wake centerline (x/L>1.0) decays somewhat faster than the experimental data in the near wake and becomes slightly negative before gradually recovering to the zero ambient pressure in the far wake. The detailed pressure variations in the radial direction is shown in figure 49, with the pressure as a function of the normal distance from surface ($r-r_0$), where r_0 is the local radius of the body.

Here again the agreement with the available experimental data is quite good, considering the difficulties in measuring pressure in such an enviroment. It

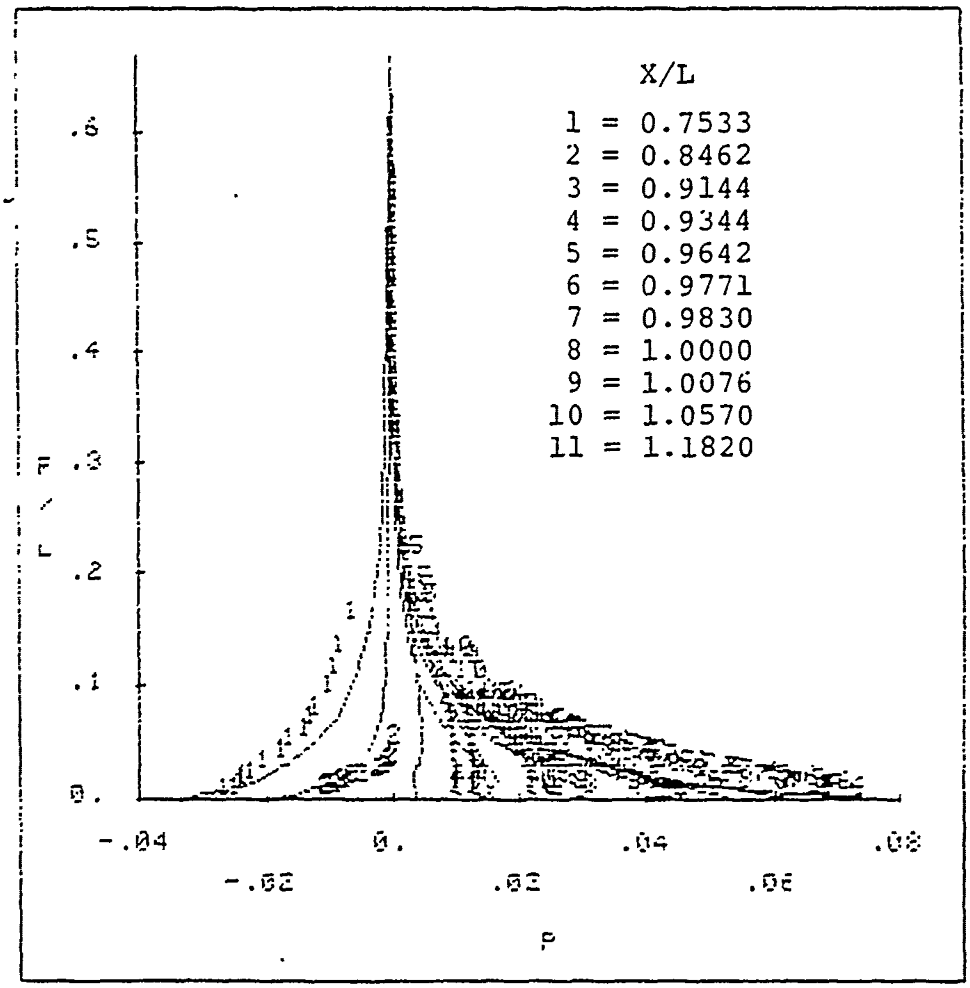


Figure 49. Pressure Profile at
Different Station for Afterbody 1

is seen that zero ambient pressure is recovered when the radius distance is beyond r=0.35L from the body surface $r=r_0$. Here r_0=0.0456L between x/L=0.164 to 0.6064 and r_0<0.0456 between x/L=0.6064 to 1.

Figures 50 and 51 show the detailed comparisons between calculated and experimental profile of the axial velocity (u), radial velocity (v) and kinetic energy k at different stations. Here u and v are dimensionless x and r velocity components normalized by U_0 and k the dimensionless kinetic energy normalized by U_0. It is seen that the boundary layer thickness and half-width of the wake are correctly predicted. The axial (u) and radial (v) components of velocity in the rear end of the body and near wake region are also in good agreement with the corresponding data. The predicted turbulent kinetic-energy k shown in figure 51 gives a somewhat larger value in the wall region near the tail (x/L>0.96), where the boundary layer becomes thick. The larger values are predicted for the mean velocity, hence the velocity gradient in the wall region of the thick boundary layer are presumably related to the over-estimation of the eddy-viscosity by the k-ε turbulence model.

Figure 52 shows that the predicted wall-shear velocity U_τ or (τ_ω/ρ) is slightly larger than the data especially the last 5 percent of the body length. All these differences maybe due to the use of the simple wall functions, Eqs (6)

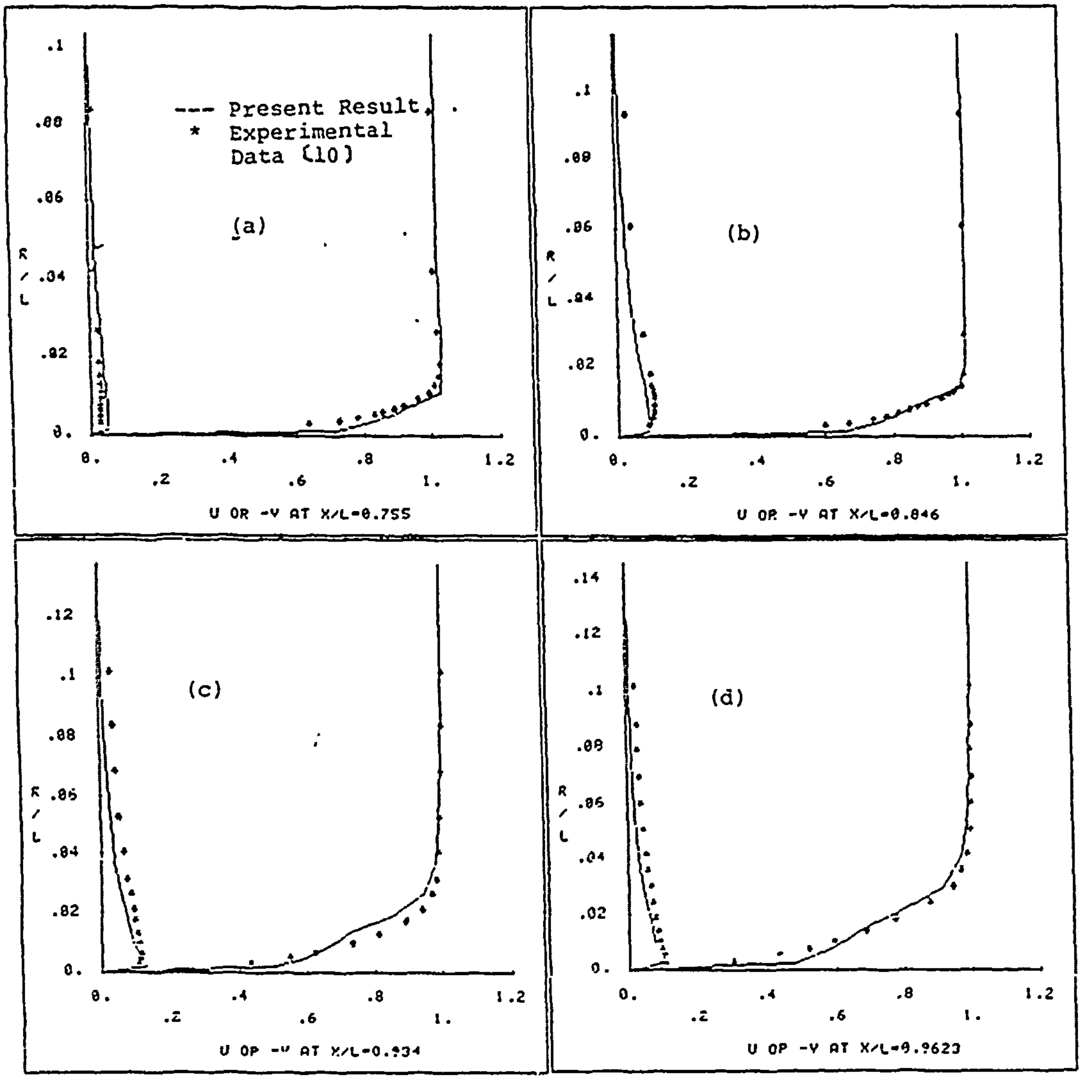


Figure 50. Velocity Profile at Different Stations

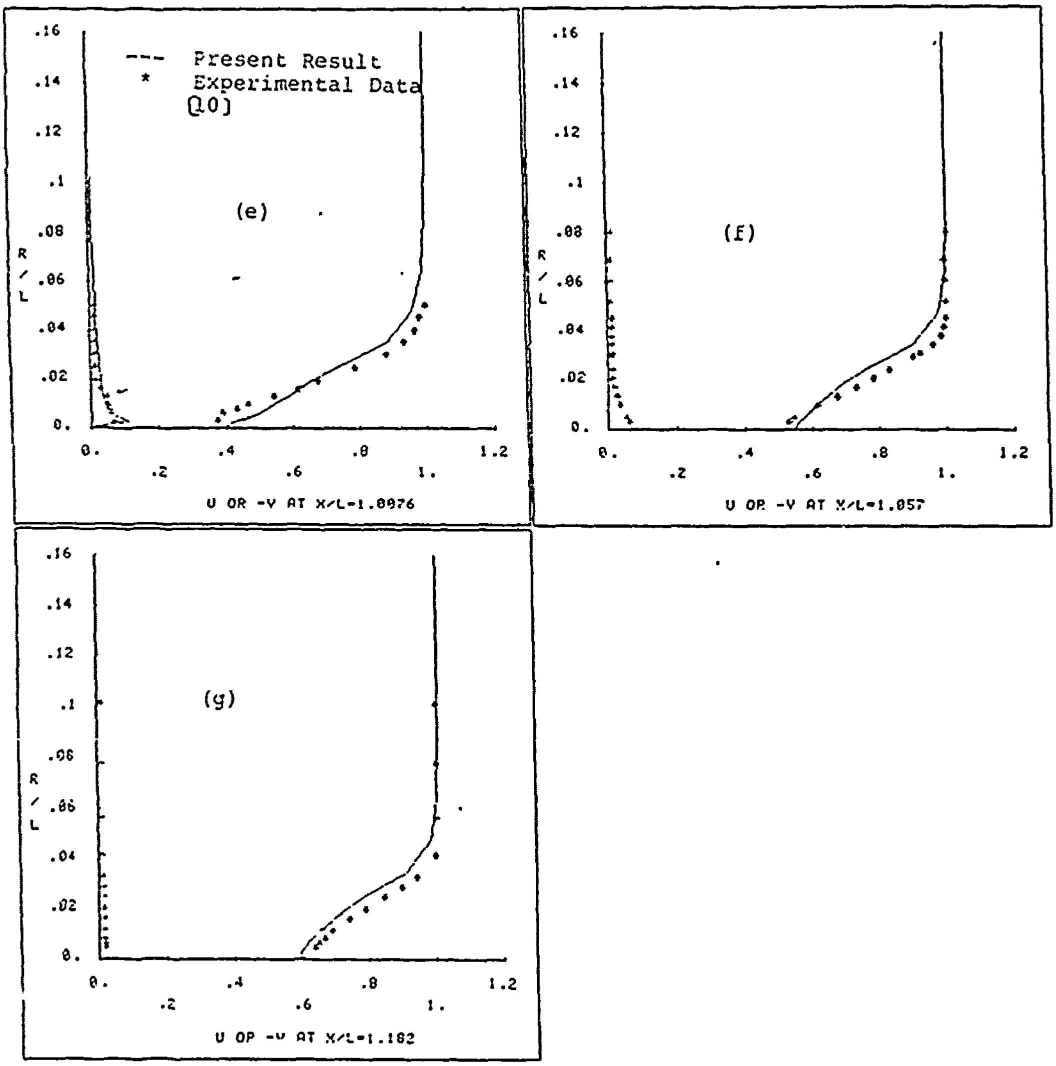


Figure 50. Continue

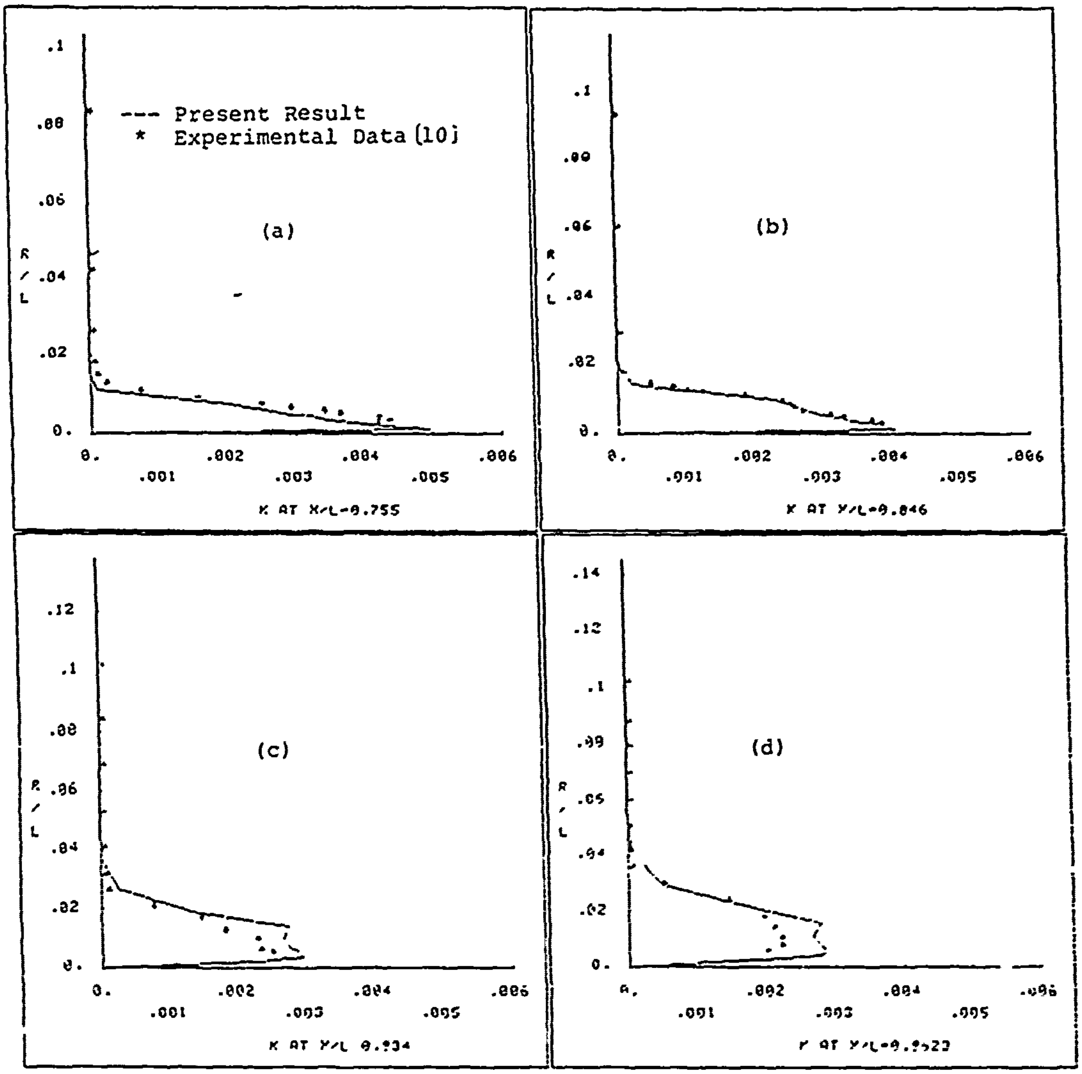


Figure 51. Kinetic Engergy
Profile at Different Station

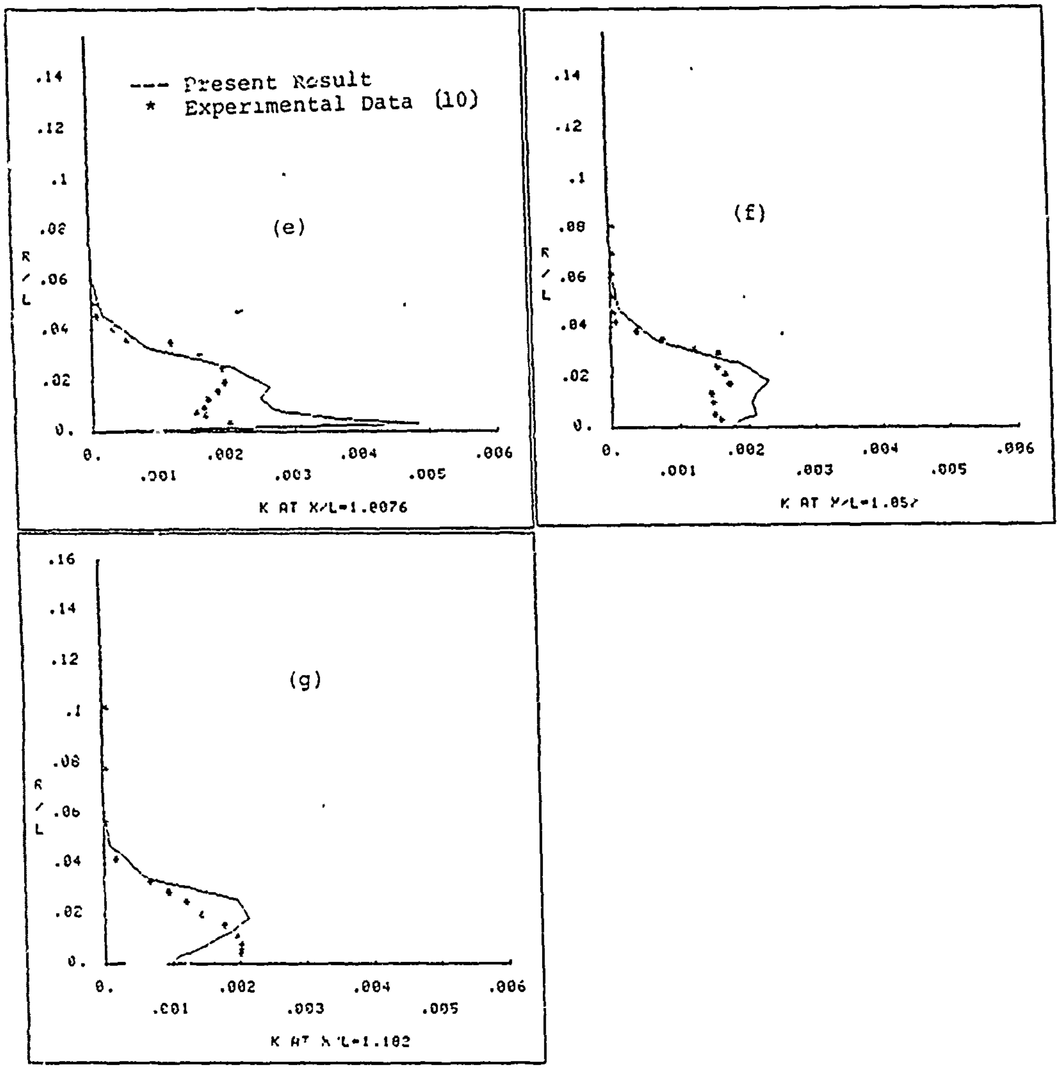


Figure 51. Continue

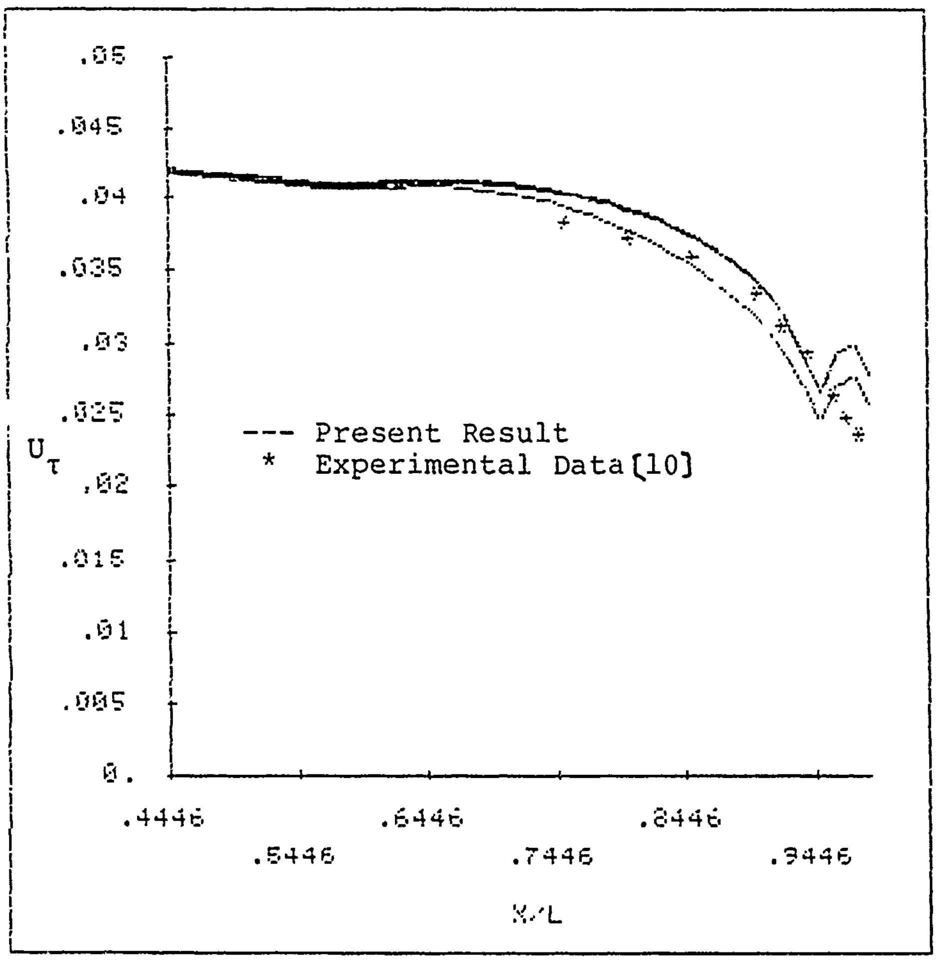


Figure 52. The Distribution Of Wall-Shear Velocity U_τ On The Body Surface

and (7), at the tail of the body where the curvature changes sharply. In the future investigation the simple two-node log-law wall function used in FANS-3DEF may require modification in order to provide a real similation of flow over a surface where the large curvature occurs.

5.3 F-57 Body

As shown in figure 46 (b) the total length, L, of F-57 body is 1.219m (4ft). The coordinates of this body are given by

For $0<x<x_m$ (fore-body)

$$\frac{r_0}{r_m} = (-1.1723a + 0.7088a + 1.0993a + 0.3642a)^{0.5} \qquad 26(a)$$

For $x_m<x<L$ (pointed aft-body)

$$\frac{r_0}{r_m} = (-0.11996b - 2.58278b + 3.52544b + 0.1773b)^{0.5} \qquad 26(b)$$

where $a=x/x_m$, $b=L-x/L-x_m$, x is the axial distance measured from the nose, r_0 is the local radius, x_m (=0.4446L) is the axial location of maximum radius r_m (=0.117L), and L is the total length of the body. In the experiments the main body of the model was made of seasoned wood but metal nose- and tail-pieces, 5.08cm and 12.70cm in length, respectively, were used to provide accuracy and durability. The experiments were performed by Lee [9] in the large wind tunnel of the Iowa Institute of Hydraulic Research. The working section of the tunnel is 7.3m long with a cross-section in the form of a 1.5m octagon provided by throating a 3.7m square approach section. In this experiment the velocity of wind tunnel was held constant at 15.24 m/s (50 fps), where a Reynolds number of $Re=1\ 2x10^6$ was obtained. The model was mounted in the wind tunnel by means of eight 0.84mm diameter steel wires in tension at x/L=0.475, and the major measurements were conducted only from x/L=0.601 to x/L=2.472.

Like Afterbody 1, the staggered grid system with the $k-\varepsilon$ model is used in the FANS-3DEF program for the calculations of flow past F-57 body. There are 56 stations in the axial direction between $0.364<x/L<6.580$ and 19 grid points between the body surface and the external boundary r/L=1.35. The partial viw of grid distributions is shown in figure 53. The same coordinate-stretching functions and upstream condition as for Afterbody 1 were used again in this case. The principal results of the calculations for F-57 body are shown in figures 54 through 58.

Figure 54 shows the convergence history of the pressure on the body surface and along the centerline of the wake. Unlike the Afterbody 1 the F-57 body does not have a constant radius at the middle part of the body, instead the F-57 body continuously increases radius from the leading

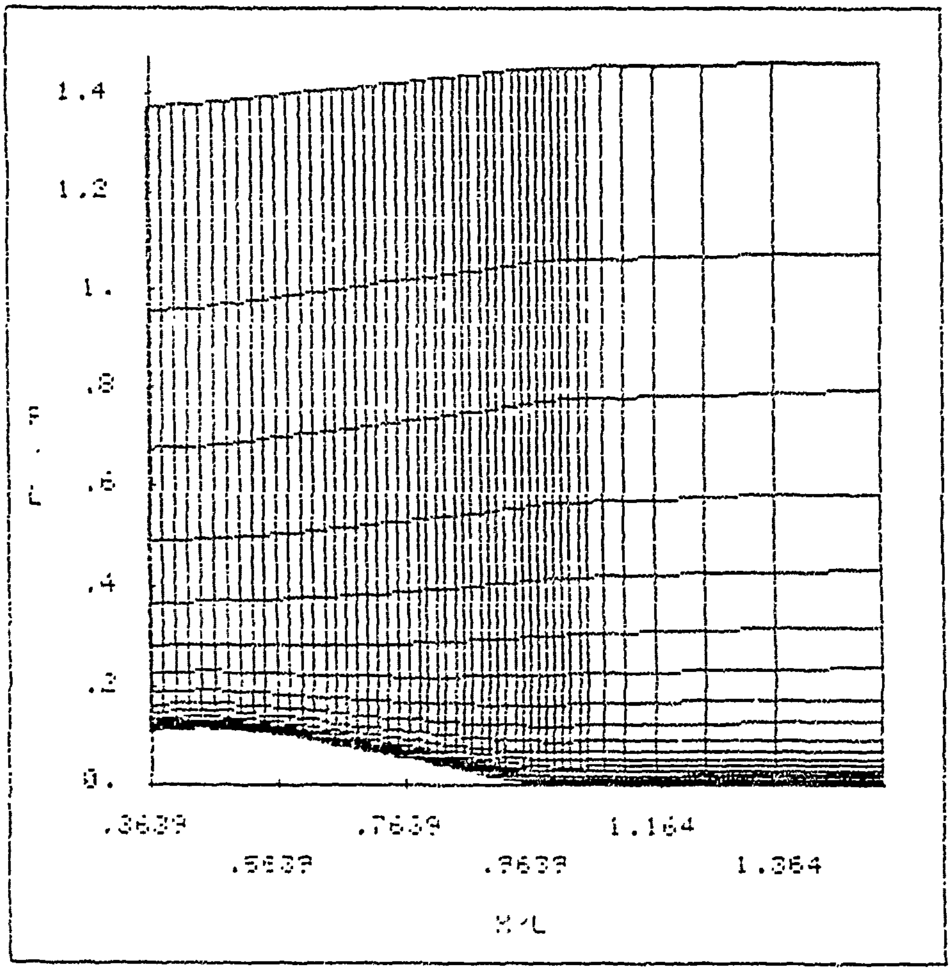


Figure 53. The Partial View of
Numerical Grid for f-57 Body

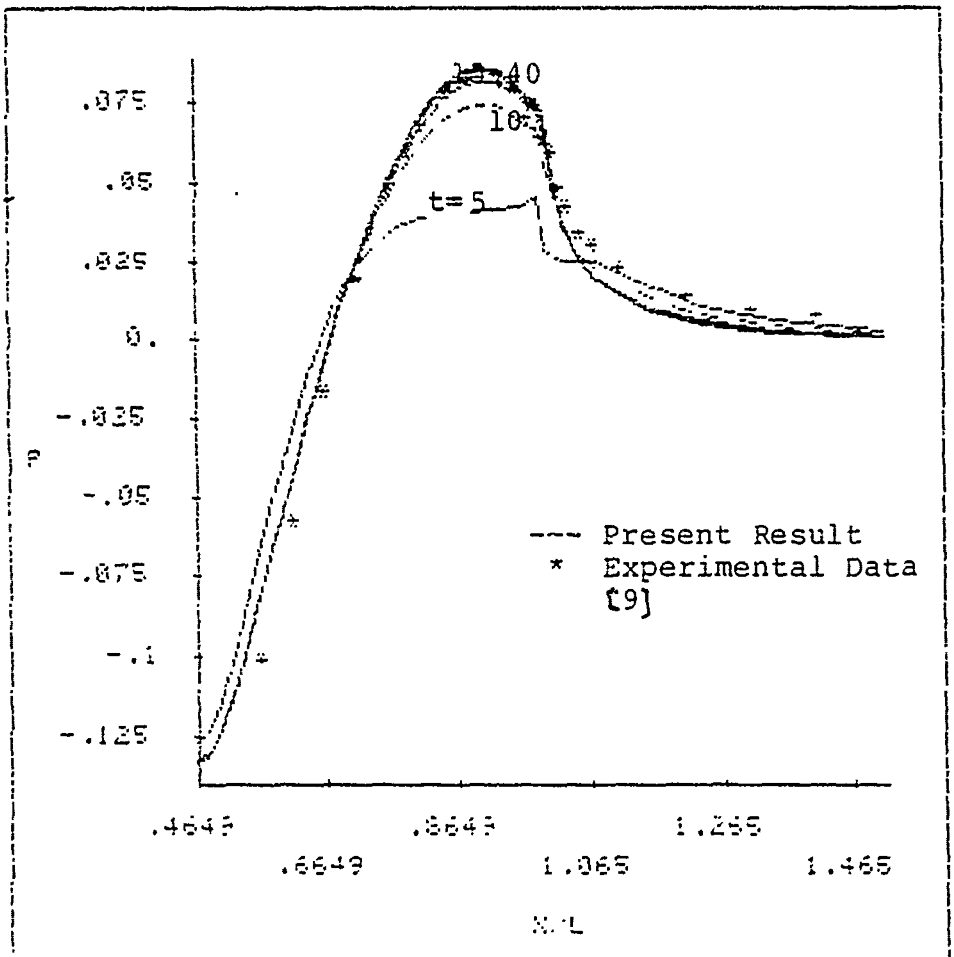


Figure 54. Convergence History of Pressure Distributin on F-57 Body

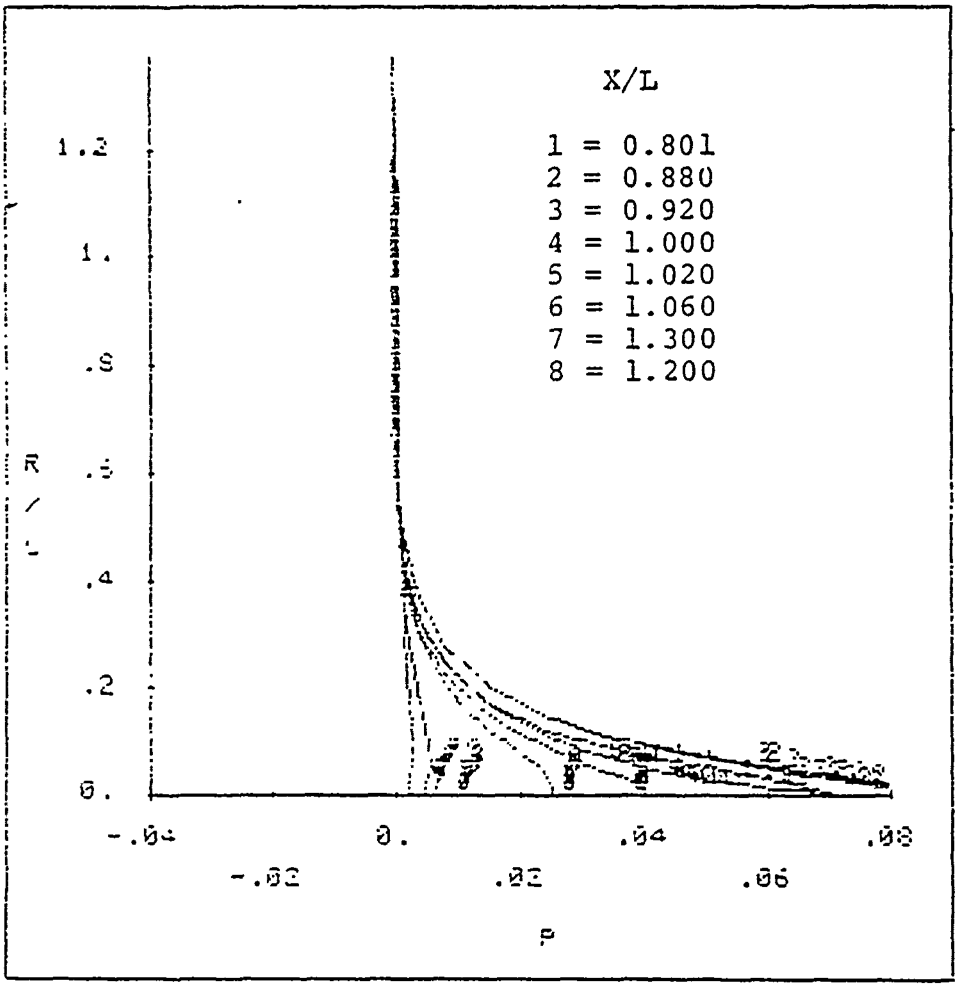


Figure 55. Pressure Profile at different Station for F-57 Body

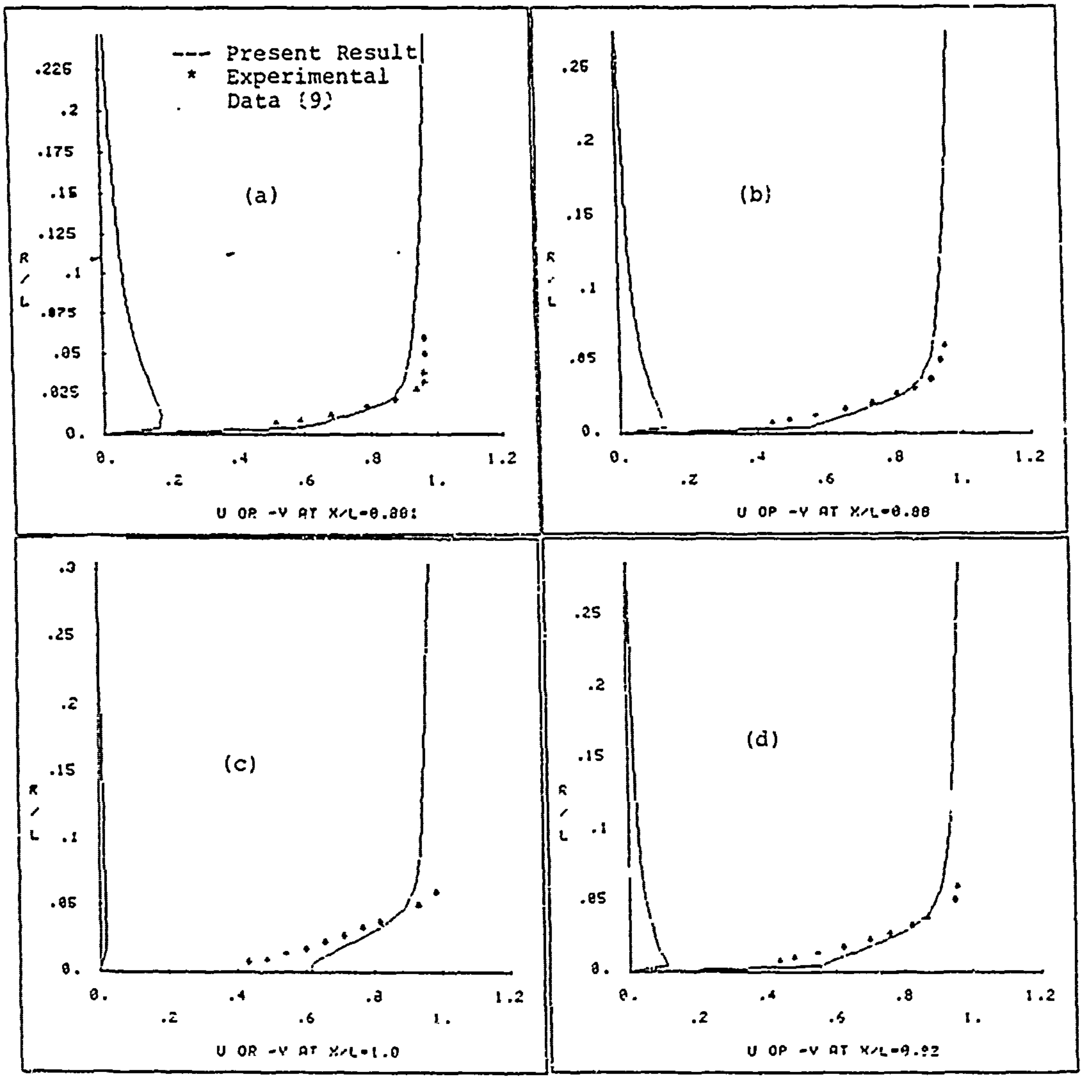


Figure 56. Velocity Profile
at Different Stations

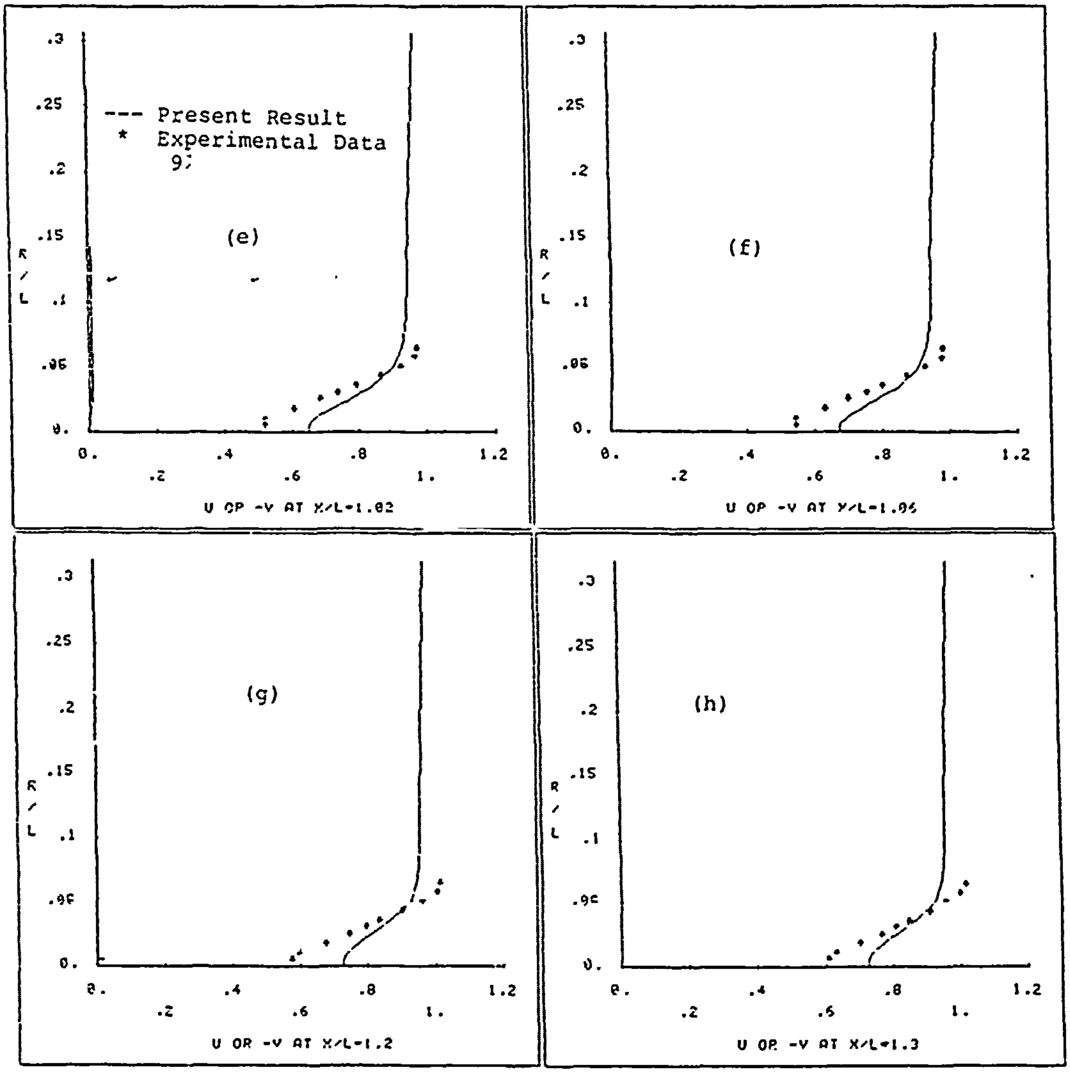


Figure 56. Continue

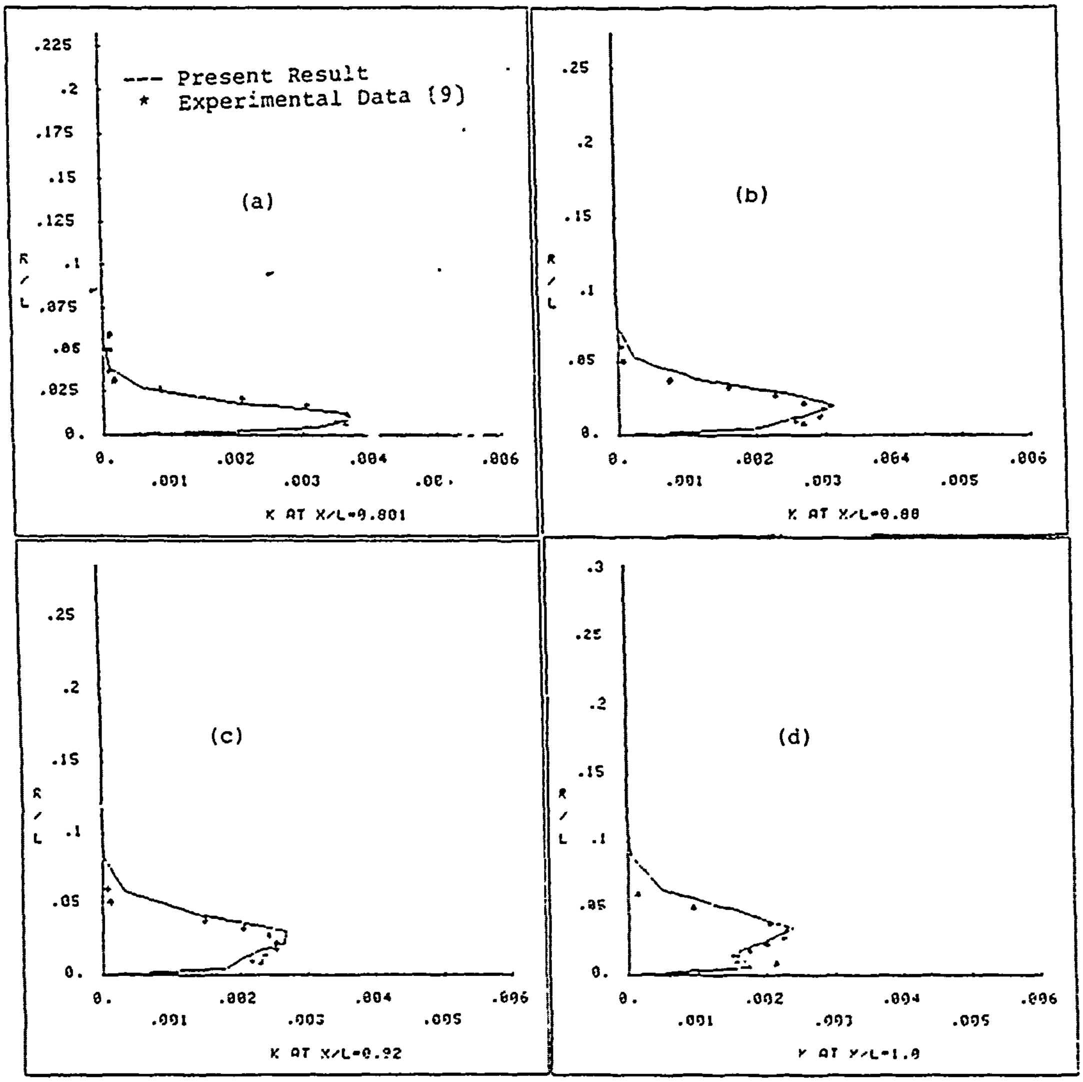


Figure 57. Kinetic Engergy Profile at different Stations

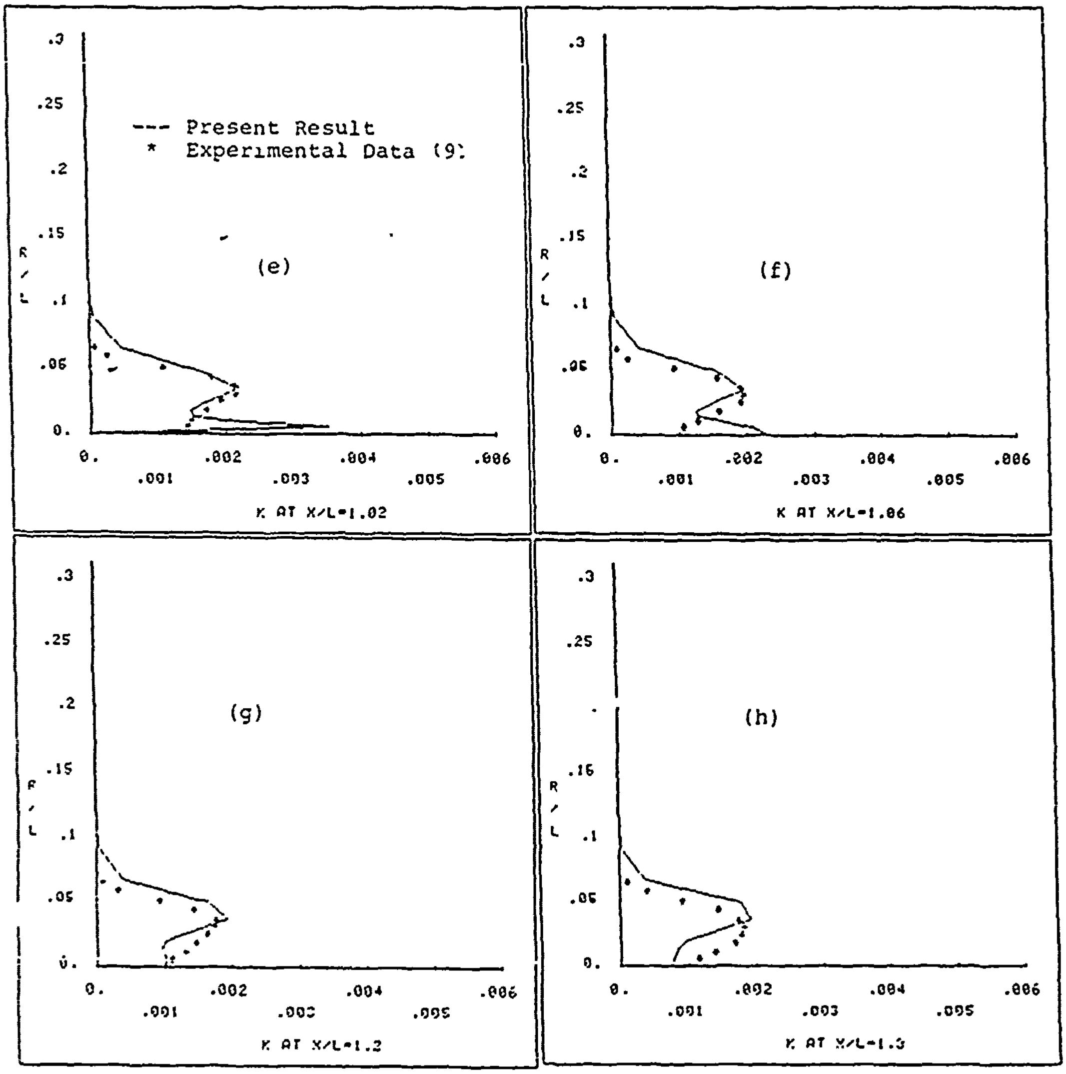


Figure 57. continue

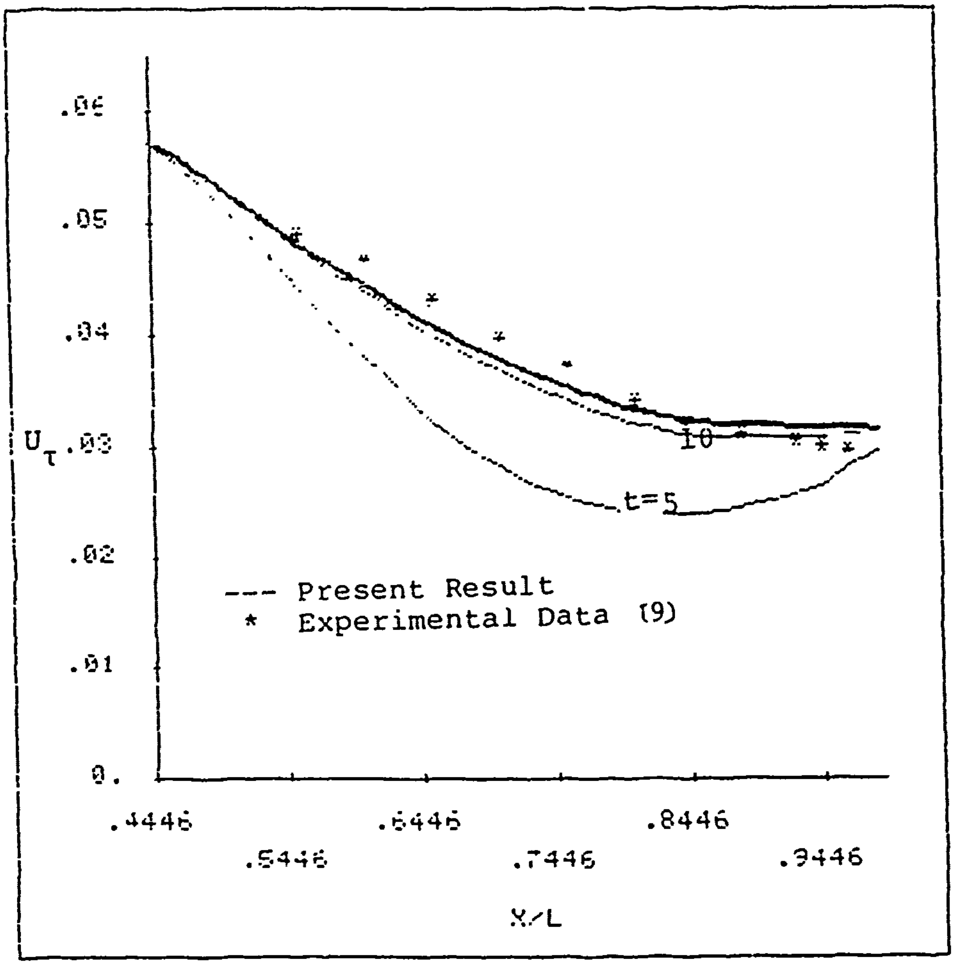


Figure 58. Convergence History Of Wall-Shear Velocity U_τ On F-57 Body

edge with Eq. 26(a) to the maximum radius at x/L=0.4446 then continuously decreases with Eq. 26(b) to the trailing edge. Thus the minimum pressure occurs at the location where the radius is maximum (r/L=0.117,x/L=0.4446). The pressure smoothly recovers at the trailing edge and drops again to the ambient pressure in the wake region. The converged solution as shown in figure 54 is obtained in less than 20 time steps and is in excellent agreement with the experimental data except at the tail of the body and near wake where the predicted values are slightly lower than the data. Before a comparison is made between the calculated and experimental profile in the radial direction at different stations, it should be remarked that the F-57 experimental data were measured along the direction normal to the body surface while the numerical calculations were solved along the direction normal to the axis of the body. Since it is not easy to transfer results in either way without avoiding any error and since the curvature of body surface does not change sharply except at the region very close to the tail. in this study both the experimental data and numerical solution are kept in their orginal directions. Figures 55, 56 and 57 show the detailed comparisons between the calculated and experimental profile of pressure, axial velocity (u), radial velocity (v) and kinetic energy k at each different station. Overall the predictions are in good agreement with experimental data except that the axial velocities along the center line of the wake are higher. This is again due to use of simple wall function along the body surface, and simple initial condition at the upstream station. Unlike Afterbody 1 the surface of F-57 body continuously changes its shape, therefore it is more difficult in specifying the initial condition for the computational domain. In the next chapter the prediction of flow past the whole axisymmetric body will be considered. In this situation the specification of the initial condition at upstream of the body may become simple and accurate.

CHAPTER VI

FLOW PAST AXISYMMETRIC BODY WITH ANGLES OF ATTACK

In this chapter the FANS-3DEF program that includes all numerical methods described before is used to predict flow past an inclined axisymmetric body. The prediction of turbulent flow past an axisymmetric body is conducted for the whole axisymmetric body including (1) the approaching flow, (2) the flow past the body from the leading to trailing edge and (3) the wake region. The calculation was first made for the flow corresponding to the experiments of Yasuhara [5]. The experiment was conducted on a 20 mm diameter brass pipe that was 1750 mm long with a 100 mm long ogive-nose as shown in figure 59(a).

This model was placed at zero degree angle of attack in a wind tunnel that has a velocity range from about 8 m/s to 35 m/s. The ogive-nose extends 100 mm from the base of the cylinder. The brass pipe was clamped by a supporting device at the rear end and the model was supported by a cantilever beam. The pressure distribution was measured by a Pito tube with 0.2mm x 1.0mm hole. For wind velocities up to 20.3 m/s or Reynolds number based on the length of cylinder,

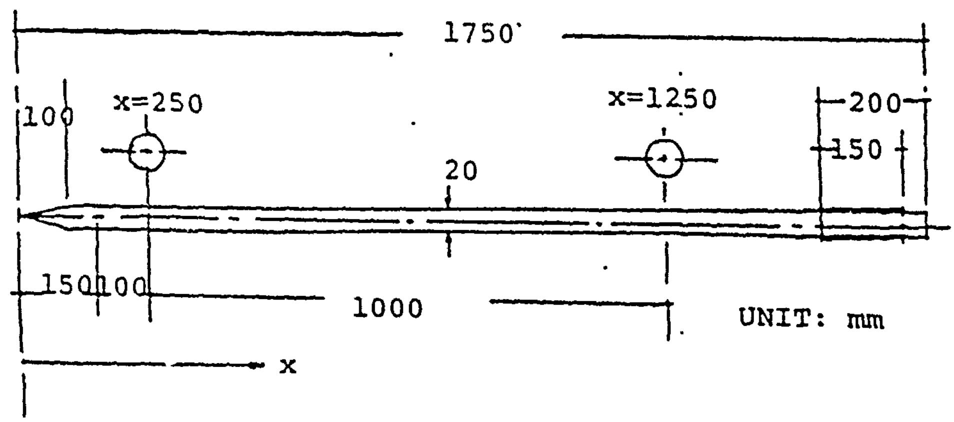


Figure 59(a). The Geometry Of Ogive-Nose Cylinder Used In The Experiment

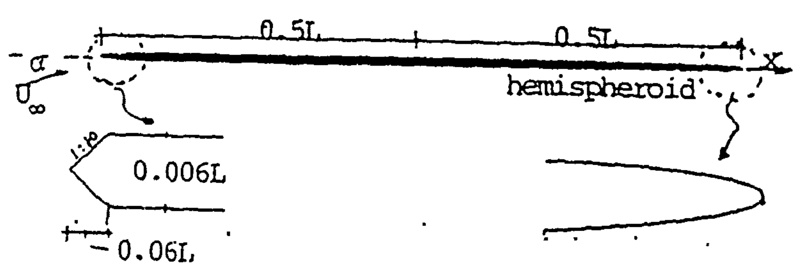


Figure 59(b). The Geometry Of Ogive-Nose Cylinder Used In This Study

THIS
PAGE
IS
MISSING
IN
ORIGINAL
DOCUMENT

Therefore, the computational domain is $-1.02<x/L<8.146$, $0<(r-r_0)/L<0.85$ where r_0 is the radius of cylinder changes along the axis of the cylinder body. A nonuniform grid, with 82 points in the x direction and 19 points in the r direction was used. A partial view of the grid distribution is shown in figure 60. The computation was done by marching in time with a time step $\Delta t=0.1$. The initial guess of uniform velocity and zero pressure were used. The total time steps of 100 were used when a steady solution was predicted.

The same numerical model of transition used for the flat plate is used here again. Figure 61 shows the convergence history of the skin coefficient C_f on the cylinder body. The predicted transition is near $x/L=0.06$ or local Reynolds number $Re_x=2.22\times10^5$. It is obvious that the numerical modelling of transition predicts that the transition to occur at $x/L=0.06$, the intersection point of the ogive-nose curve and the straight cylinder. Physically the flow on the ogive-nose cone is constantly accelerated between $0<x/L<0.06$ because of the increase in body radius from the leading edge to the straight cylinder. Therefore the u velocity inside the boundary layer was predicted to increase until the flow reaches the intersection point of the ogive-nose and the straight cylinder. The velocity on the straight cylinder then begins to decelerate due to the

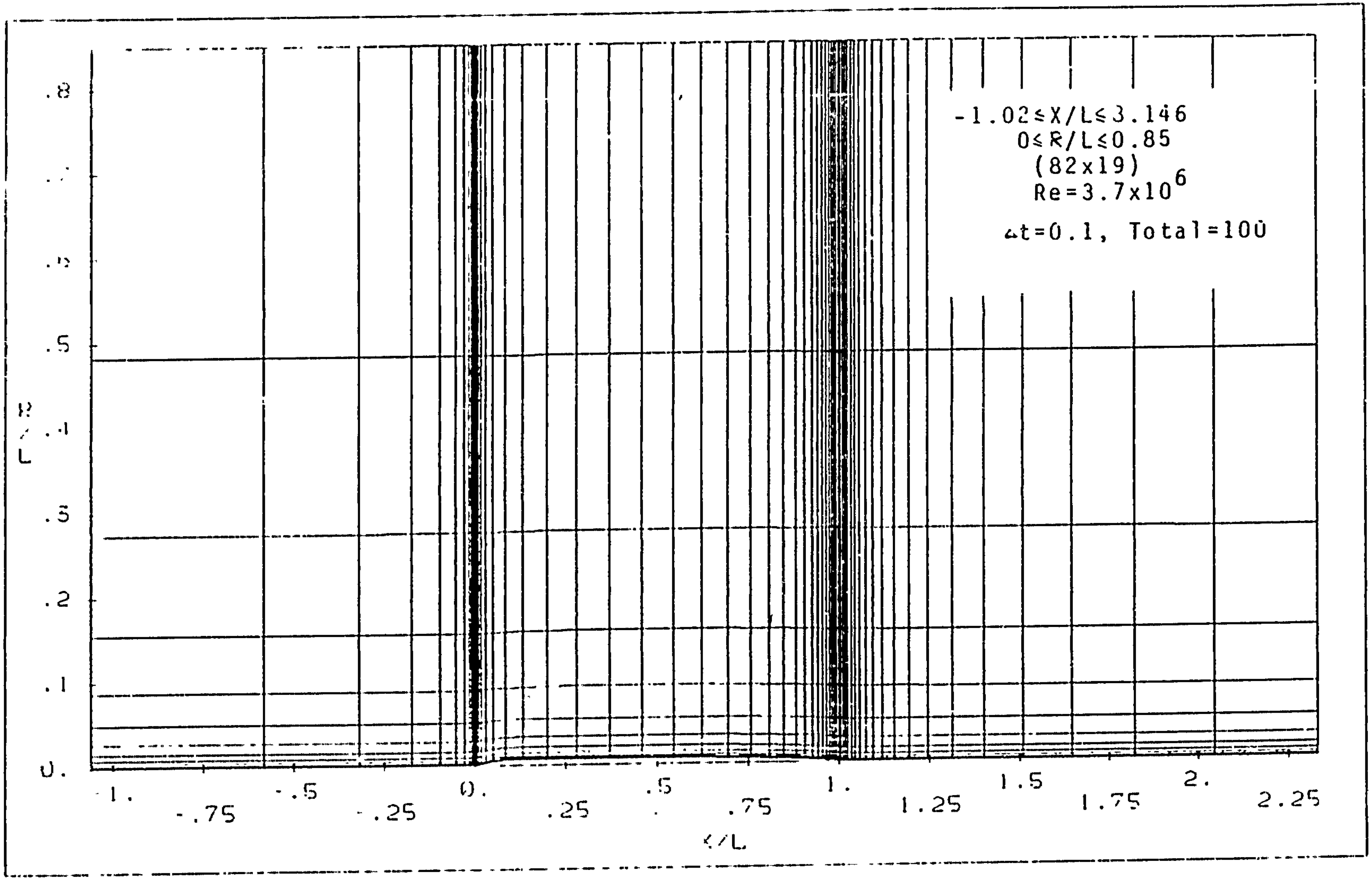


Figure 60. The Partial View Of Numerical Grid (82x19) For Flow Past A Ogive-Nose Cylinder Without Angle Of Attack

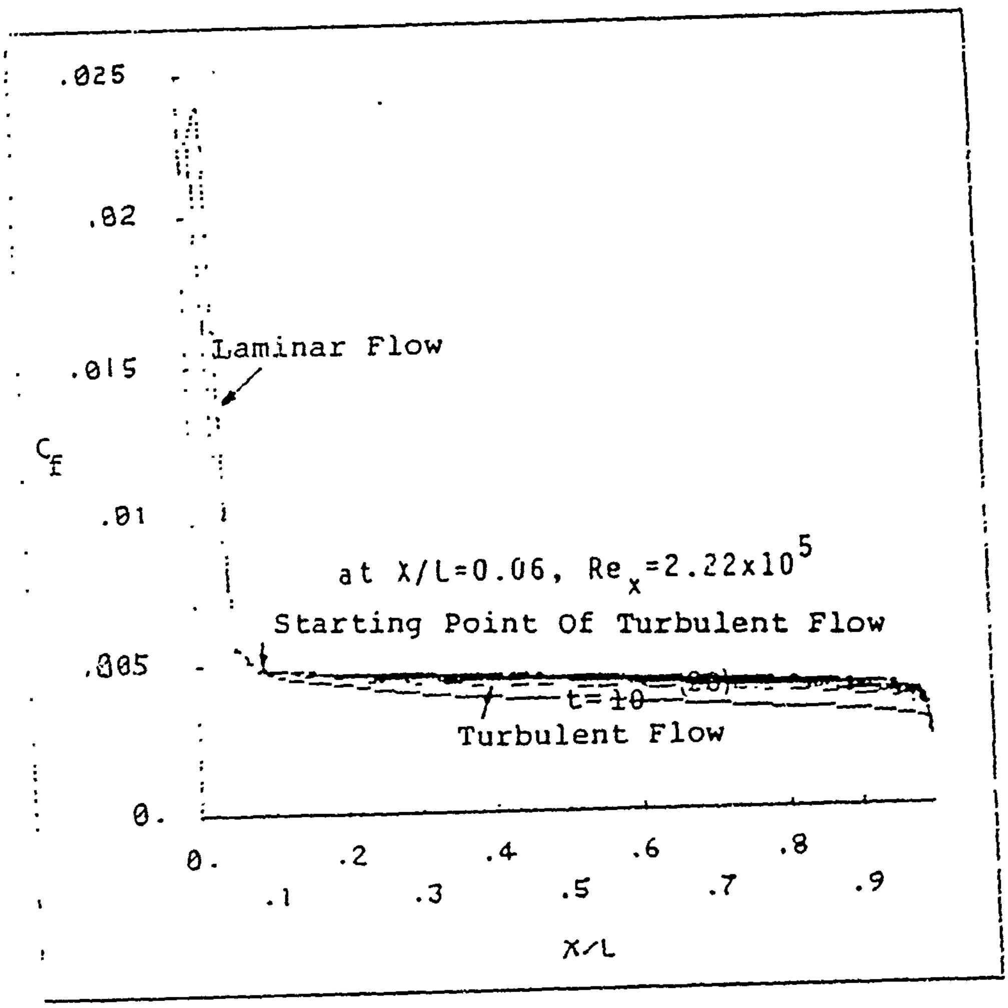


Figure 61. Convergence History of Skin Coefficient on the Ogive-Nose Cylinder With Transition Model

disappearence of the pressure gradient on the straight cylinder and the constant resistance of the viscous flow. The prediction of transtion based on the criteria discussed in section 5.4 to occur at $Re_x=2.22x10^5$ seems to predict much earlier transition than that indicated by Yasuhara [5] of $Re_x=1.2\text{-}1.8x10^6$. Yasuhara [5] determined the transition by examining the measured velocity profiles at four stations namely x/L=0.143, 0.286, 0.429, 0.572 and reported that the transition may start between $Re_x=1.2\text{-}1.8x10^6$. The numerical modelling of transition proposed for the flat plate thus required further modification.

The above computation is repeated with the exception that the transition is set at x/L=0.37 or $Re_x=1.37x10^6$ as given by Yasuhara [5]. The predicted skin coefficient for this case with experimentally determined transition and that predicted with built-in transition criteria are given in figure 62 for comparison. It shows that the skin coefficient on the ogive-nose cylinder based on transition model did not have dip in the distribution when the flow changes from laminar to turbulent flow. The predicted skin coefficient with the transition point fixed at x/L=0.37 shows a dip at the skin coefficient arcund the end of the ogive-nose or x/L=0.06 even the flow is laminar at this region. As shown in figure 62 both calculations give approximately the same maximum skin friction coefficient of about 0.023 and the same value in the turbulent flow from $x/L=0.5$ to 1. The comparison of the predicted pressure distribution on the cylinder body with the experimental data [5] was shown in figure 63.

It shows that the predicted surface pressure based on the transition model under prediction but gives the same trend with the experiment. The predicted surface pressure with experimentally determined transition point seems to match closely to the experimental data. In general both calculations predict that the pressure rises as the flow approaches the nose and then drops to a minimum before recovering to a constant value before it reaches the tail edge where the pressure increases before the flow past the tail. The physical explanation is that as the flow approaches the ogive-nose it decelerates and the pressure begins to rise. Once the fluid is on the ogive-nose it begins to accelerate as the pressure starts to drop sharply and reaches the minimum pressure around the end of nose (or the start of the straight cylinder). The flow begins to decelerate after it reaches the end of the ogive-nose. Once the fluid is on the straight cylinder surface the pressure quickly recovers to a level which is almost that of the free stream pressure. This is because the fluid is no longer accelerated along the straight surface of the cylinder and the pressure variation across the boundary layer on the

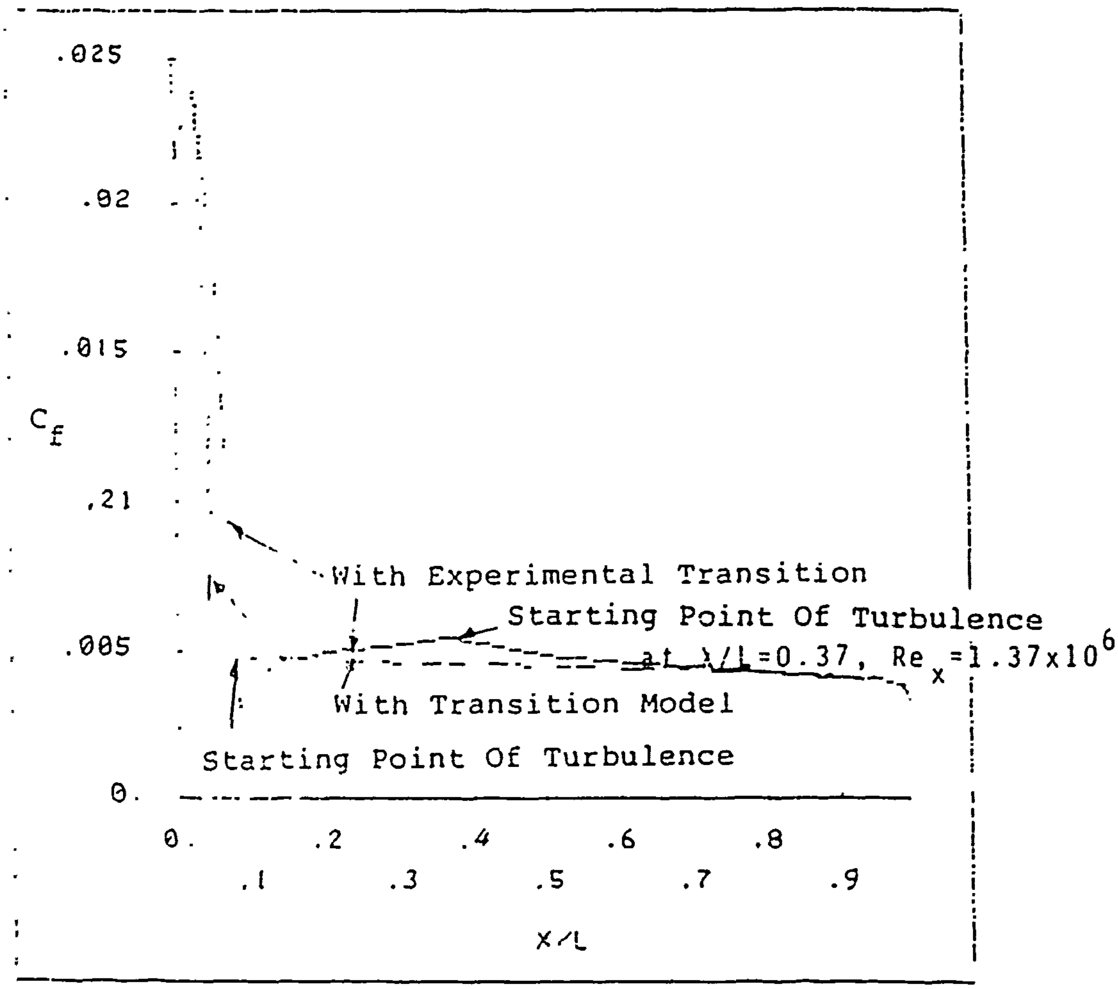


Figure 62. The Skin Coefficient on the Ogive-Nose Cylinder

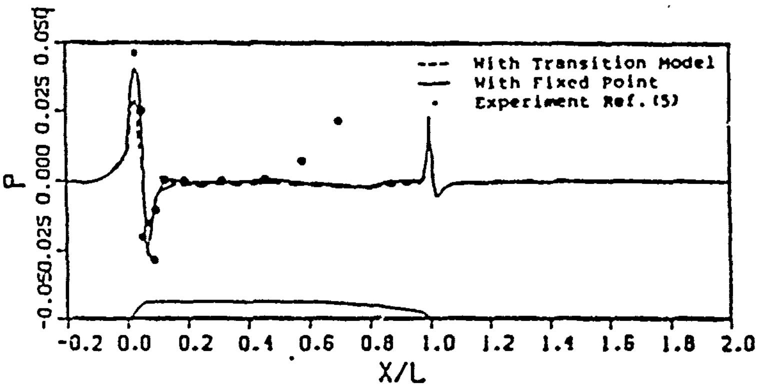


Figure 63. Pressure Distribution on the Cylinder and Along Wake Centerline

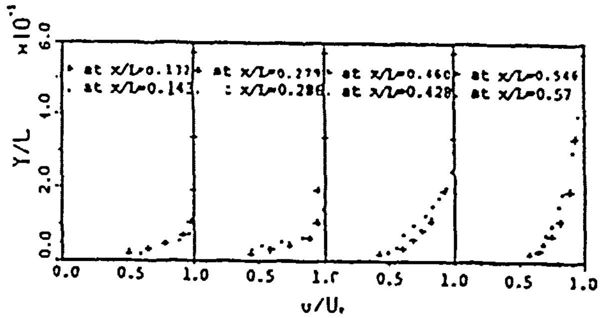


Figure 64. Velocity Profiles in Boundary Layer

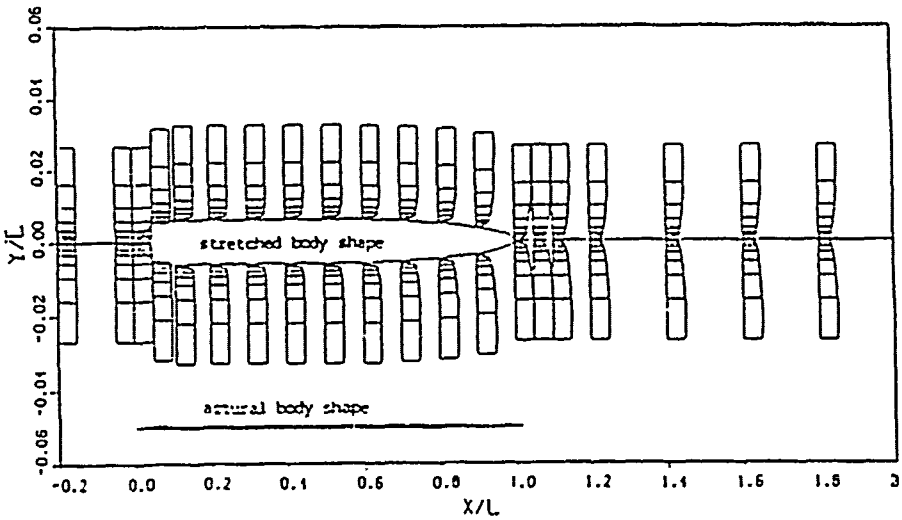


Figure 65. Velocity Profiles (u) at Zero Angle Attack

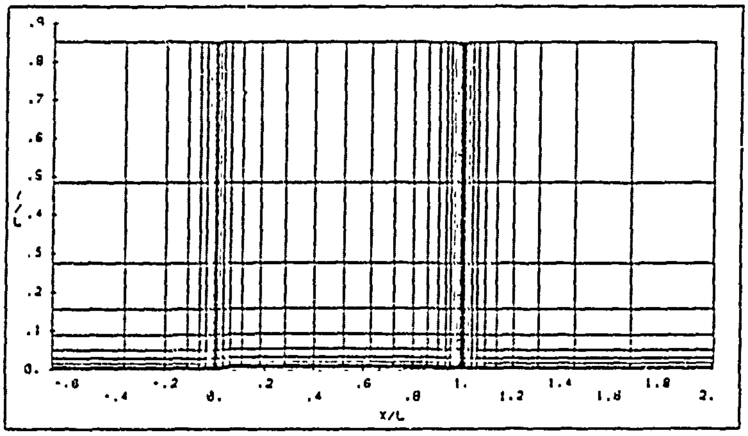
straight cylinder is negligible. When the flow enters the rear region x/L>0.5 where the flow is decelerated due to decrease in the diameter of the cylinder. The pressure on the surface rises again so that the sum of velocity head and pressure head is approximately conserved. When the flow leaves the body to beco... wake, the flow along the axial direction is then accelerated from zero velocity at the nonslip surface to some velocity. This acceleration causes the pressure momenterly to drop but recoves to the free stream pressure soon after the acceleration is reduced. Yasuhara [5] remarked that his experimental data are not accurate after x/L=0.6 because his model was clamped by a supporting device at its rear end, and the model was hanged from above by a cantilever beam. Therefore pressure variation near the trailing edge which was predicted by the present method can not be compared with Yasuhara's data. However the predicted pressure distribution shown in figure 63 is qualitatively similar to those predicted by inviscid theory or experimrntaly obtained by Ramaprian, Patel and Choi [12] for flow past a body with hemispheroid at the rear end.

The predicted longitudial velocity based on the experimentally determined transition at x/L=0.37 shown in figure 64 is quite the same as that measured by Yasuhara [5]. Cebeci [59] solved the same flow past the slender cylinder by the boundary layer equations using two-layer mixing length model, and also found that there is a good agreement between the prediction and measured velocity profiles of Yasuhara. It should be remarked that the use of boundary layer equations can not predict the pressure distribution since the boundary layer approximation assumes that the pressure is given by the free stream flow. Therefore, in Cebeci [59] calculation the experimental data of pressure distribution was used as inputs. However in the present FANS-3DEF calculation the pressure distribution and velocity components are predicted simultaneously and no experimental data of pressure distribution or assumed potential flow solution are required as a priori. Figure 65 shows the development of the x-component velocity u from upstream to the wake region. It should be mentioned that the y coordinate in figure 65 is streched about nine times over the axial scale in order to visualize the velocity distribution near the body. It is seen that the boundary layers grew symmetrically along the axial direction and merged at the rear end to form wake.

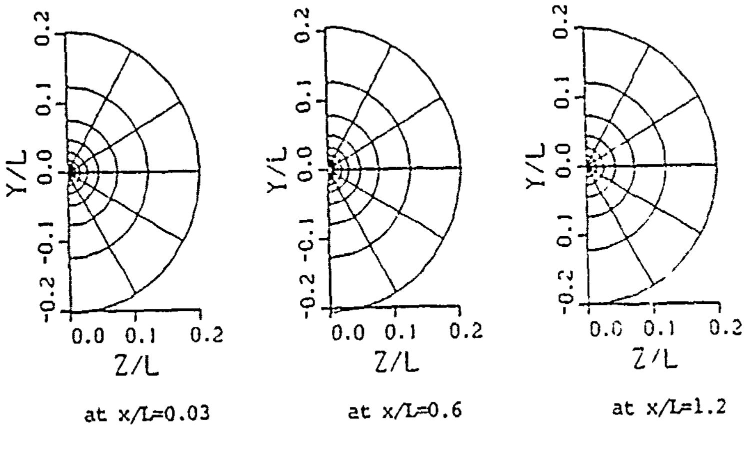
6.2 Flow Past Inclined Ogive Cylinder

Once the FANS-3DEF program was verified with the experiment for the flow past the axisymmetric body without angle of attack, the flow past the ogival cylinder for angles of attack at 5, 10 and 15 degrees are predicted. This is a complex three dimensional flow calculation since the flow is no longer a symmetric one and three dimensional variables and grids in the x, r and θ directions are required. Since the computational space for each user is limited at the University of Iowa, relative coarse grid spaces are used here. The whole compuational domain are $-0.65<x/L<8.55$, $0<(r-r_0)/L<0.850$. Figure 66 is the partial view of the whole computatonal domain. There are 62 points in the x direction (axial direction), 19 points in the r direction (radial direction) and 9 points in the θ direction (azimuthal direction). It should be mentioned that the relative coarse grid spaces are used only to illustrate the capabilities and stabilities of the FANS-3DEF program. More accurate solutions can be achieved when the grid spaces are allowed to be refined. The upstream and boundary conditions for the x and r component velocity u, v were reset as $u=U_0COS(\alpha)$ and $v=U_0SIN(\alpha)$. The angle of attack α was varied from 5, 10 to 15 degrees. The Reynolds number $Re=3.7x10^5$ is used.

The transition model for the flat plate is not quite adequate for the flow past an ogive-nose cylinder as solved in the last section. One can not pre-predict the real transition point at different position around the azimuthal direction when the ogive-nose cylinder is subjected to an



(a) Partial View of Computational Domain



(b) Cross Sections at Different Location

Figure 66. The Numberical Grid (62x19x9) for Flow Past Ogive-Nose Cylinder with Angle of Attack

angle of attck. Therefore in this study, we approximately assume that the turbulent flow starts at X/L = 0.4 or local Reynold number is 14.8x10.

Figure 67 - 69 show the predicted skin coefficent μ_τ at three generators (θ = 0°, 90°, and 180°) with respect to the angle of attack α = 5, 10 and 15 degrees. All these three figures show the following common features. First the values of skin coefficient at θ = 0° are increased sharply at the front part of ogive-nose; then droo to a very small value at the end of the ogive-nose or X/L = 0.06. After X/L = 0.06, the skin coefficients increases sharply again till X/L = 0.1. Beyond it, the skin coefficients varies slower till the end of the cylinder. For θ = 90° the skin coefficient μ_τ increase slowly and then a a big drop occurs at X/L=0.06. After that, there is no too much change till X/L=0.28. At X/L=0.4, the flow is assumed to be turbulent flow and the skin coefficients consequently have an obvious jump. The skin coefficient then decrease slightly downstream. For θ = 180°, which is the rearward of the cylinder, the trend of skin coefficent is almost the same as that at θ = 90°, except that the variation is smaller and smoother.

Figures. 70 to 72 show the corresponding pressure distribution at three generators, namely θ=0° (windward side; solid line), 90° (dotted line) and 180° (leeward side,

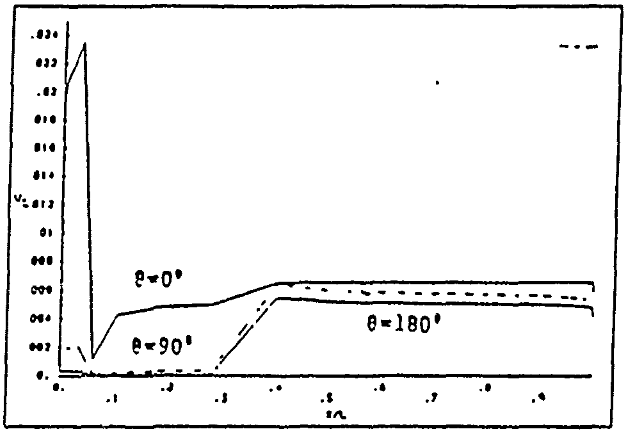


Figure 67. The Predicted Skin Coeffient at 5 Degree Angle Attack.

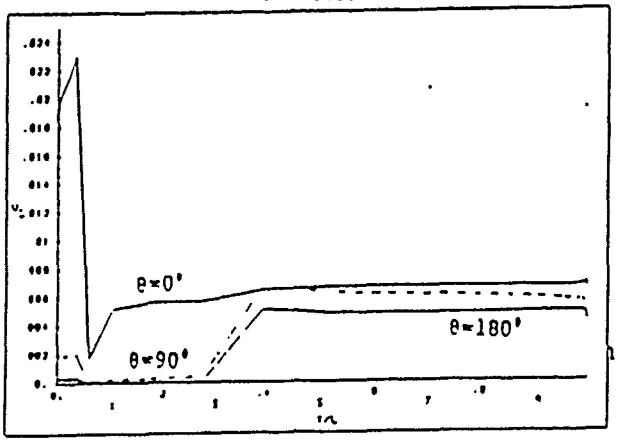


Figure 68. The Predicted Skin Coeffient at 10 Degree Angle Attack.

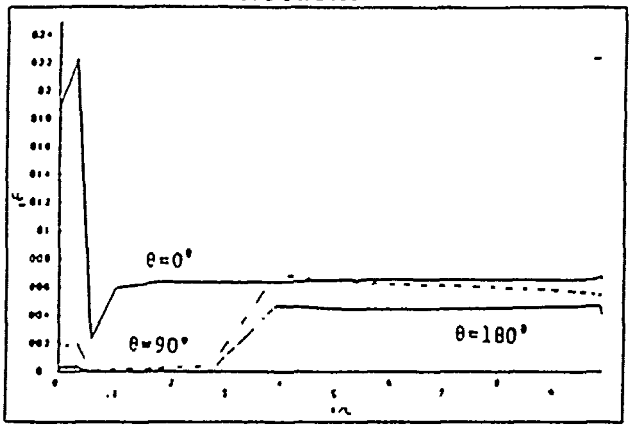


Figure 69. The Predicted Skin Coeffient at 15 Degree Angle Attack.

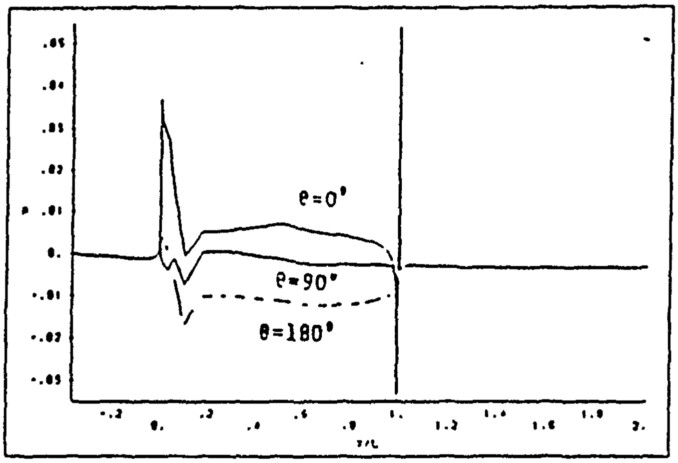


Figure 70. Pressure Distribution at 5 Degree Angle Attack.

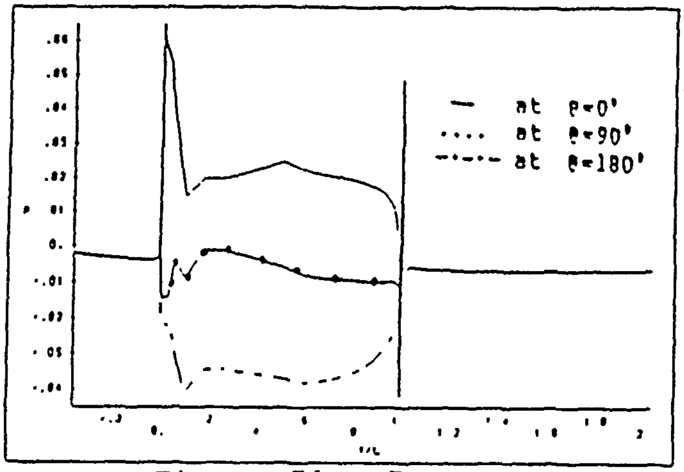


Figure 71. Pressure distribution at 10 Degree Angle Attack.

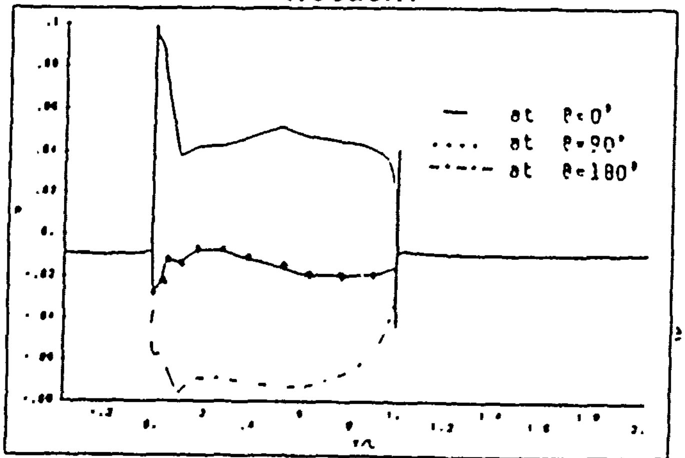


Figure 72. Pressure Distribution at 15 Degree Angle Attack

dashed-dot line), with respect to the angle of attack α=5, 10 and 15 degrees. All these three figures show the following common features. First, the pressure at the upstream location of the ogive-nose unlike the case of zero angle attack, first decreases along the axis before it reaches the nose. This is due to the fact that the flow is accelerated along the axial line when there is an angle attack so that the flow on the axial no longer like that of the case of zero angle attack where the flow slows down as it approaches the stagnation point at the nose tip. In other words when there is an angle of attack the stagnation point is no longer on the axial line and the flow along the axial line never needs to decelerate and instead it accelerates. Consequently the pressure decreases. Second, the increase of pressure in the nose region on the windward side ($\theta=0^\circ$) is the largest because of the existence of stagnation region on its plane and the increase of pressure in the leeward side ($\theta=180^\circ$) is the smallest with the tangential side ($\theta=90^\circ$) in the middle. Third, the greater the angle attack the larger is the spread in pressure difference from windward side to the leewardside.

Figures. 73 to 75 depict the variation of the x component velocity ,u, on the plane of θ=0 and 180 at different angles of attack. It is found as expected that the boundary thickness is thinner on the windward side ($\theta=0^\circ$)

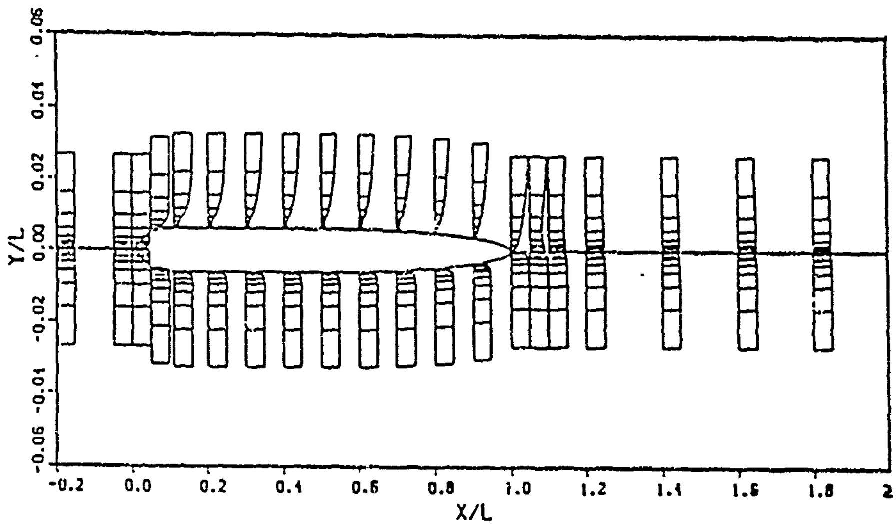


Figure 73. Velocity Profiles
(u) at 5 Degree angle Attack

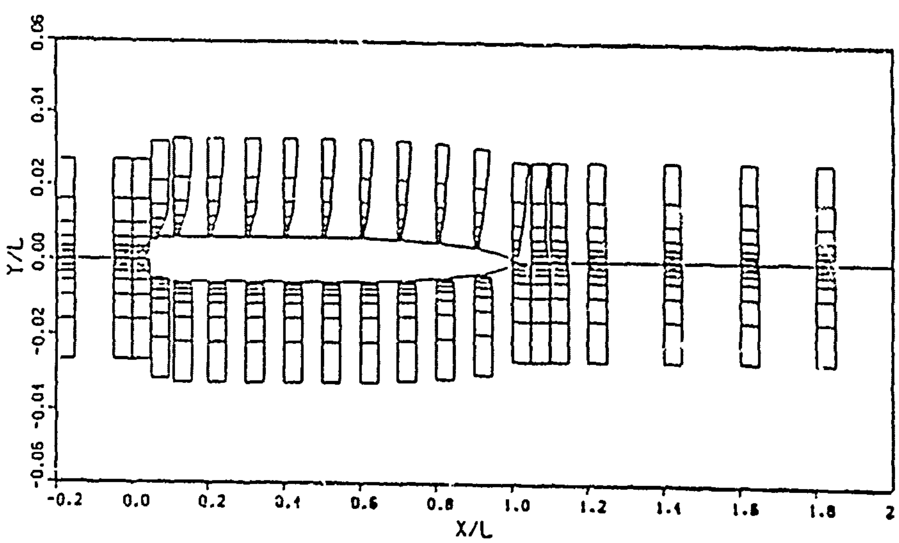


Figure 74. Velocity Profiles
(u) at 10 Degree angle Attack

than that on the leeward side (θ=180°). No boundary separations are predicted for 5, 10 and 15 degrees of angle attack. This is partly due to a small angle of attack and partly due to moderate curvature of the ogive-nose shape at the front end and hemispheroid body at the rear end. It is seen that the wake flow is unsymmetrical when there is an angle of attack and it shows the location of the maximum defect in the u velocity in the wake region is not on the axial line. As the degree of angle of attack increases, the location of the maximum defect moves more to the leeward side. Figures. 76 to 78 show the variation of the r component velocitr ,v, at different angles of attack. Here the positive value denotes that the flow in the positive r direction or the radial direction. It is seen there the v component velocity on the leeward side in general is small except near the body where the fluid merged after it passes the body from the windward side.

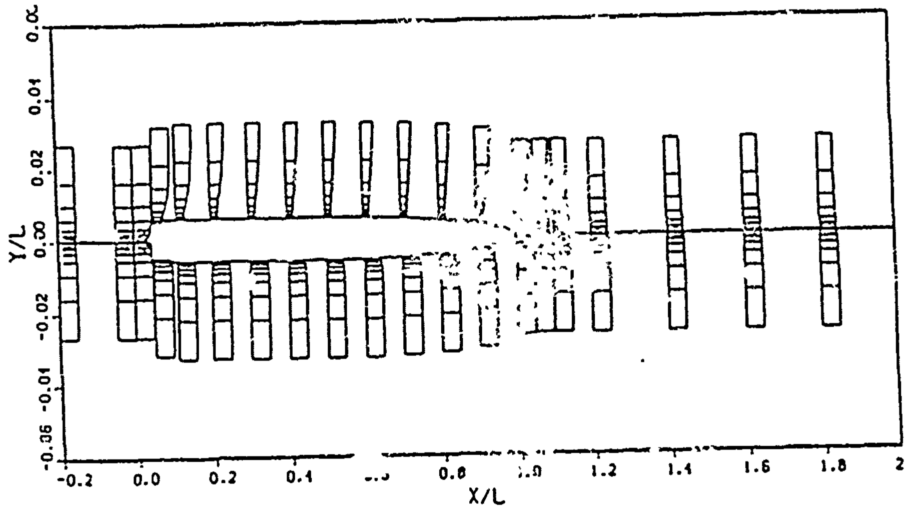


Figure 75. Velocity Profiles
(u) at 15 Degree angle Attack

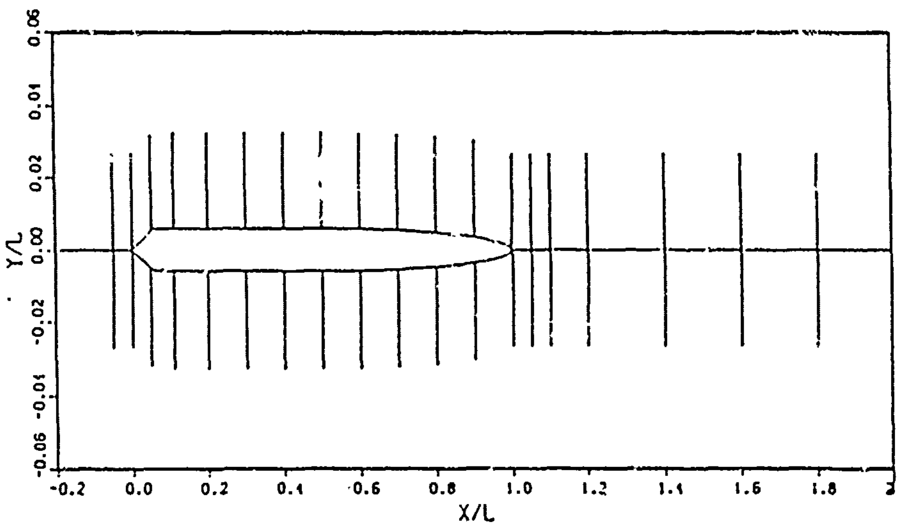


Figure 76. Velocity Profiles
(v) at 5 Degree angle Attack

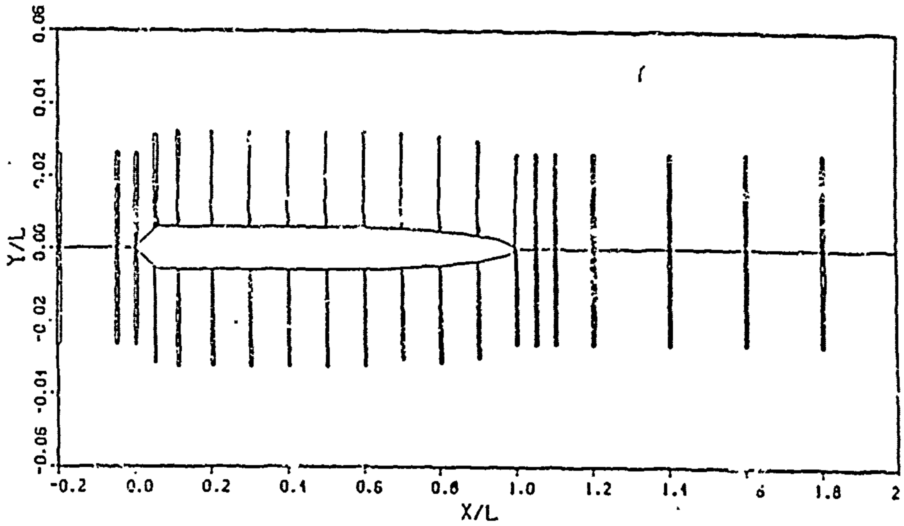


Figure 77. Velocity Profiles
(v) at 10 Degree angle Attack

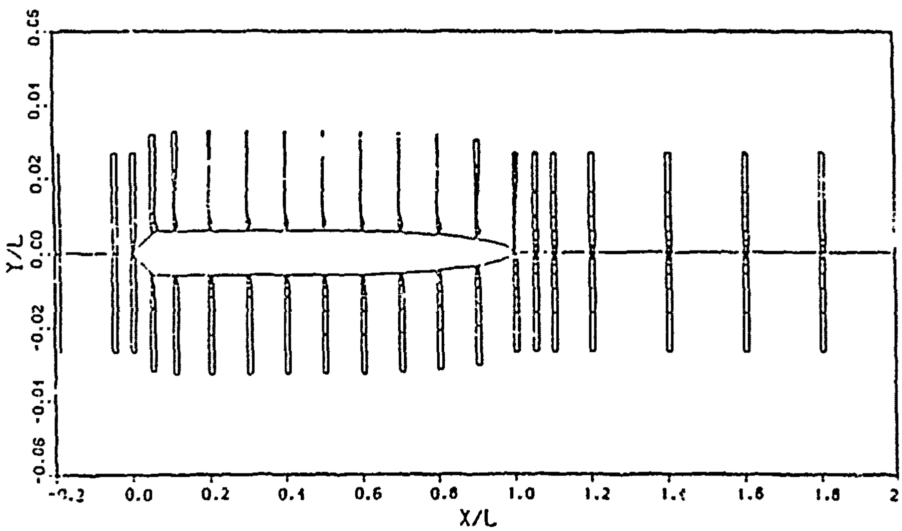


Figure 78. Velocity Profiles
(v) at 15 Degree angle Attack

CHAPTER VII

CONCLUSION AND SUGGESTION

In this study a user interactive numerical program called FANS-3DEF (Finite Analytic Numerical Solution of Three Dimensional External Flow) is developed. This program which is based on the finite analytic method on the body-fitted coordinate system with modified SIMPLER algorithm was used to predict incompressible laminar and turbulent flows past the finite flat plate and axisymmetric bodies with or without angles of attck. Some examples of flow prediction where the experimental data are available are presented to demonstrate the accuracy and validity of the FANS-3DEF program.

The major contributions of the present work are:

1. Derivation of Finite Analytic solution for unsteady three dimensional laminar and turbulent flows on the body-fitted coordinate system.
2. Calculation of a computational domain includes the entire geometry from the approaching flow to the wake region.
3. Development of FANS-3DEF program and its applications.

4. Investigation of complex vortex shedding behind a flat plate and complex flow past axisymmetric bodies.
5. A simple numerical model for transition zone is developed and tested on the flat plate so that the prediction of a flow may be calculated for the entire plate from the approaching flow to the wake region.

All calculations presented here were performed on a Prime 750 minicomputer at the CAELAB of the University of Iowa with computing times of less than half hour for the two dimensional and axisymmetric cases, and of the order of two hours for the flat plate with vortex shedding and flow past axisymmetric body with angles of attack. It should, therefore, be rasonable to use the FANS-3DEF program for practical applications.

While the overall predicted results are shown to be in good agreement with experimental data or reasonable when the experimental data are unavailable there are still several aspects about the numerical methods and turbulence and transition models in the FANS-3DEF that can be further developed and improved. The following suggestions are submitted for further study.

1. The application of the numerical model of transition zone: the numerical model of transition zone presented here is developed based on the simple physical phenomena on the flat plate, therefore it

needs more study and tests before it can be completely applied to other geometries of bodies or the flow problem involving the strong curvature.

2. The sensitivity of the solution to the turbulence model: the validity of the one-scale $k-\varepsilon$ turbulence model for more complex flow problems has not been verified. The two-scale $k-\varepsilon$ turbulence model which has strong physical support can be considered in the further study. The two-node wall function based on the fully developed flow assumption in general is not applicable to flow with separation. Thus a wall function that is valid for the turbulent flow with separation should be developed if complex separation flows are to be predicted.

3. The use of a grid system for the pressure equation: the regular grid system which has some advantages in saving computer time and storage requires further study to become competiable with the staggered grid system in accuracy and stability.

4. The programing of FANS-3DEF: the FANS-3DEF program is a research code, it needs more testing and modification to become a general program.

APPENDIX A

THE TWO DIMENSIONAL FA COEFFICIENTS

In this appendix a general finite analytic algebraic representation of two dimensional convective transport equations is briefly outlined. We consid:r a two dimensional convective transport equation in a given finite analytic element shown in Fig. A-1. The equation is

$$D\phi_t + 2A\phi_x + 2B\phi_y = \phi_{xx} + \phi_{yy} + f \qquad \text{(A -1)}$$

Where D, A, B and f are constants in a given FA element such as those shown in Eq. (12). In order to solve Eq. (A-1) in a given element, one must specify boundary and initial conditions for the element. Among the possible solution forms of Eq. (A-1) are a constant, an exponential and a linear function. A conatant, an exponential and a linear function then are used to describe the boundary function of the local element. For example the northern boundary function of a element as shown in figure A-1 can be approximated by

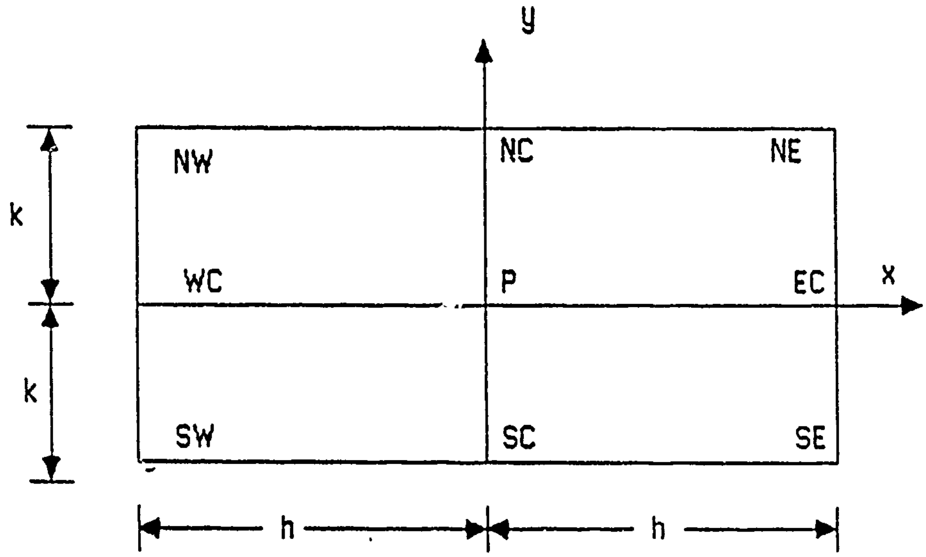
$$\phi_N(x) = a_N(e^{2Ax} - 1) + b_N x + C_N \qquad \text{(A-2)}$$

In term of the three nodal values on the northern boundary the coefficients a_N, b_N, and c_N are

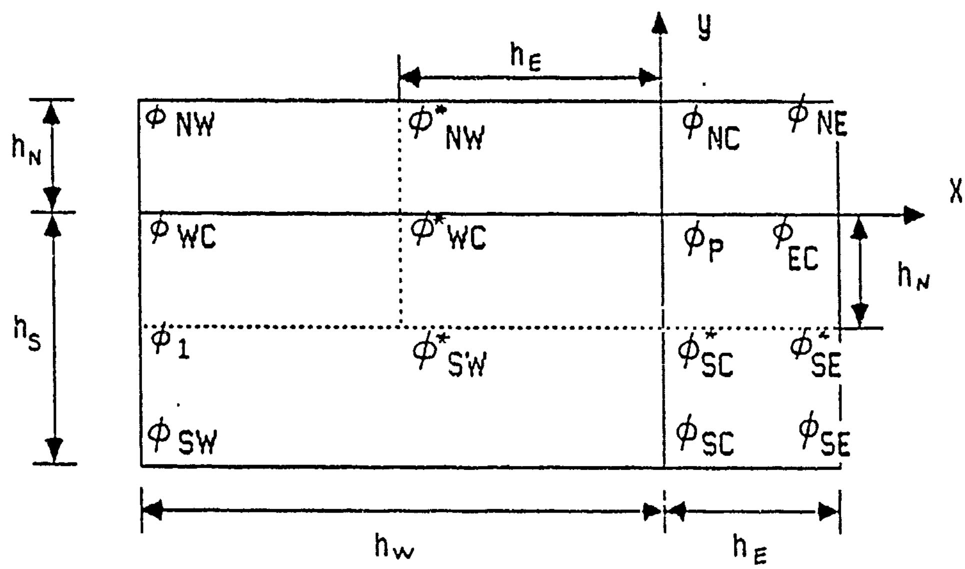
$$a_N = \frac{\phi_{NE} + \phi_{NW} - 2\phi_{NC}}{4 \sinh^2 Ah}$$

$$b_N = \frac{1}{2h}\{\phi_{NE} - \phi_{NW} - \coth Ah \, (\phi_{NE} + \phi_{NW} - 2\phi_{NC})\}$$

$$C_N = \phi_{NC}$$



(a) Local element of uniform grid sapcing



(b) Local element of nonuniform grid spacing

Figure 79 (A-1) Uniform and Nonuniform
Finite Analytic Element

similary, the boundary conditions for south, east and west side can be approximated as follow:

$$\phi_S(x) = a_S(e^{2Ax} - 1) + b_S x + c_S$$

$$\phi_E(y) = a_E(e^{2By} - 1) + b_E y + c_E$$

$$\phi_W(y) = a_W(e^{2By} - 1) + b_W y + c_W$$

where a_S, b_S c_S ...... etc. are expressed in terms of nodal values on each boundary in a way similary to that for a_N, b_N and c_N. The FA solution of Eq. (A-1) can be derived directly from uniform grid mesh as shown in figure A-1. Details of the derivation and related discussion can be found in Ref. [31]. When the FA solution is evaluated at node p the FA formulation or uniform grid mesh can be written as:

$$\phi_p = \frac{1}{1 + \frac{DC_p}{\Delta t}} \left(\sum_{nb=1}^{8} C_{nb}\phi_{nb} + C_p f_p^{\,n-1} + \frac{DC_p}{\Delta t}\phi^{n-1} \right) \tag{A-3}$$

where ϕ and ϕ^{n-1} means the value evalued at the nth and (n-1)th time step respectively. For uniform grid mesh, the FA coefficients are:

$$C_{SC} = \left(\frac{e^{Bk}}{2\,\mathrm{Cosh}(Bk)} \right) P_A, \quad C_{NC} = e^{-2Bk} C_{SC}$$

$$C_{WC} = \left(\frac{e^{Ah}}{2\,\mathrm{Cosh}(Ah)} \right) P_B, \quad C_{EC} = e^{-2Ah} C_{WC}$$

$$C_{SW} = \left(\frac{e^{Ah+Bk}}{4\,\mathrm{Cosh}(Ah)\,\mathrm{Cosh}(Bk)} \right) (1 - P_A - P_B)$$

$$C_{SE} = e^{-2Ah} C_{SW}, \quad C_{NW} = e^{-2Bk} C_{SW}, \quad C_{NE} = e^{-2(Ah+Bk)} C_{SW}$$

$$C_p = \frac{h\,\mathrm{Tanh}(Ah)}{2A}(1 - P_A) + \frac{k\,\mathrm{Tanh}(Bk)}{2B}(1 - P_B)$$

One may use one of the following series to evaluate P_A and P_B in the above expression. They are:

(A)
$$E_2 = \sum_{m=1}^{\infty} \frac{-(-1)^m \lambda_m h}{\{(Ah)^2 + (\lambda_m h)^2\}^2 \cosh(\mu_m k)}$$

$$P_A = 4E_2 \; Ah \; Cosh(Ah) \; Cosh(Bk) \; Coth(Ah)$$

$$P_B = 1 + \frac{Bh \; Coth(Bk)}{Ak \; Coth(Ah)} (P_A - 1)$$

$$\mu_m' = (A^2 + B^2 + \lambda_m'^2)^{1/2}$$

$$\mu_m = (A^2 + B^2 + \lambda_m^2)^{1/2}$$

(B)
$$E_2' = \sum_{m=1}^{\infty} \frac{-(-1)^m \lambda_m' k}{\{(Bk)^2 + (\lambda_m k)^2\}^2 \cosh(\mu'_m h)}$$

$$P_B = 4E_2' \; Bk \; Cosh(Ah) \; Cosh(Bk) \; Coth(Bk)$$

$$P_A = 1 + \frac{Ak \; Coth(Ah)}{Bh \; Coth(Bk)} (P_B - 1)$$

$$\lambda_m = \frac{(2m - 1)\pi}{2h}$$

$$\lambda_m' = \frac{(2m - 1)\pi}{2k}$$

Although both series should provided same P_A and P_B values, it is however more convenient to use E2 over E2' series if the first term of E2 series is less than that of E2' series and vice versa.

In the present study the problem is solved on the transformed domain, and the general two demensional FA equation on the transformed domain can be written as

$$D\phi_t + 2A\phi_\xi + 2B\phi_\eta = E\phi_{\xi\xi} + F\phi_{\eta\eta} + f \qquad (A\text{-}4)$$

Here if E and F are equal to one then Eq. (A-4) is reduced to Eq. (A-1). However, in general E and F are positive values and not equal to one. Therefore in order to cast Eq. (A-4) into Eq. (A-1), one can introduce the coordinate-stretching functions.

$$\xi^* = \frac{\xi}{\sqrt{E}}, \qquad \eta^* = \frac{\eta}{\sqrt{F}}$$

Then Eq. (A-4) can be reduced to the same form as Eq. (A-1) as

$$D\phi_t + 2A^*\phi_{\xi^*} + 2B^*\phi_{\eta^*} = \phi_{\xi^*\xi^*} + \phi_{\eta^*\eta^*} + f$$

with

$$A^* = \frac{A}{\sqrt{E}}, \qquad B^* = \frac{B}{\sqrt{F}}$$

and a local element with dimensions

$$\Delta\xi^* = h = \frac{1}{\sqrt{E}}, \qquad \Delta\eta^* = k = \frac{1}{\sqrt{F}}$$

Thus one will obtain the same FA formular as shown in Eq. (A-3). For non-uniform grid, the FA solution becomes:

$$\phi_P = \frac{1}{G + \frac{C_P}{\Delta t}} \left(\sum_{nb=1}^{8} B_{nb}\phi_{nb} + B_P f_P^{n-1} + \frac{DB_P}{\Delta t}\phi^{n-1} \right) \qquad (A\text{-}5)$$

where

$$G = 1-(2-s-\bar{s})C_{WC} - (2-t-\bar{t})C_{SC} - (2-s-\bar{s})(2-t-\bar{t})C_{SW}$$

$$B_{NE} = C_{NE} + (s-1)C_{NW} + (t-1)C_{SE} + (s-1)(t-1)C_{SW}$$

$$B_{NW} = \bar{s}\, C_{NW} + \bar{s}(t-1)C_{SW}$$

$$B_{SE} = \bar{t}\, C_{SE} + \bar{t}(s-1)C_{SW}$$

$$B_{SW} = \bar{s}\,\bar{t}\, C_{SW}$$

$$B_{EC} = C_{EC} + (s-1)C_{WC} + (2-t-\bar{t})C_{SE} + (s-1)(2-t-\bar{t})C_{SW}$$

$$B_{WC} = \bar{s}\, C_{WC} + \bar{s}(2-t-\bar{t})C_{SW}$$

$$B_{NC} = C_{NC} + (t-1)C_{SC} + (2-s-\bar{s})C_{NW} + (t-1)(2-s-\bar{s})C_{SW}$$

$$B_{SC} = \bar{t}\, C_{SC} + \bar{t}(2-s-\bar{s})C_{SW}$$

$$B_P = C_P$$

where C's are FA coefficients of uniform grid space and

$$s = \frac{h_W(e^{2Ah_E} + e^{-2Ah_E} - 2)}{h_W(e^{2Ah_E} - 1) + h_E(e^{-2Ah_W} - 1)} \quad , \quad \bar{s} = s\frac{h_E}{h_W}$$

$$t = \frac{h_S(e^{2Bh_N} + e^{-2Bh_N} - 2)}{h_S(e^{2Bh_N} - 1) + h_N(e^{-2Bh_S} - 1)} \quad , \quad \bar{t} = t\frac{h_N}{h_W}$$

The above relations between B's and C's coefficients are derived from interpolation of nodal values for uniform grid between nodal value of non-uniform grid with interpolating function of a.exp(x)+bx+c.

APPENDIX B

THE FANS-3DEF PROGRAM

B.1 Main Program

The main program of FANS-3DEF is essentially written to solve the unsteady three dimensional turbulent incompressible flow governed by continuity equation, Eq. (1), momentum equation, Eq. (2), and turbulent transport equation, Eqs. (3) to (5), based on one or two scale k-ε turbulence model. These equations, Eqs.(1) to (5), are transformed to the body-fitted coordinate system based on Poisson equation Eq. (8). Thus Eq. (1) is expressed in Eq. (10) and Eq. (2) combines with Eqs. (3), (4) and (5) are expressed in a general form given in Eq. (11). Numerically the finite analytic formulation converts Eqs. (10) and (11) into algebraic equations. They are Eq. (11) into Eq. (13) and Eq. (10) into the pressure equation (17). In summary the main program of FANS-3DEF is written to obtain solution of Eq. (1) to (5) based on their algebraic equations given by Eqs. (13) and (17).

The numerical procedure for solving Eqs. (13) and (17) is programmed based on the modified SIMPLER algorithm introduced in the section 3.7 of the last chapter. The computer programs are written such that there are many independent subroutines which can be called to the main program to execute some specified function. In this way these independent subroutines can be modified by the user and some new subroutines can be added by user as desired.

In table 5 and table 6 the computational procedure and structure the main program of FANS-3DEF are illustrated. Table 5 shows the flow chart of the main program and table 6 shows the relationship between each subroutine. In the following the functions of each subroutine in the main program are described in the alphabetical order.

---

(1) CHECK(N)

This is a check and change subroutine. It check if it is required to update boundary conditions either along with symmetric line or at the center line of wake. The N in the bracket denotes as 1 for velocity component u, 2 for velocity component v, 3 for velocity component w, 4 for pressure, 5 for kinetic energy and 6 for dissipation rate.

---

(2) COEF

COEF solves FA coefficients, based on Eq. (12).

---

(3) EQCOE

EQCOE calculates the coefficients of governing equation Eq. (12). The coefficient a^{ϕ}, b^{ϕ},...

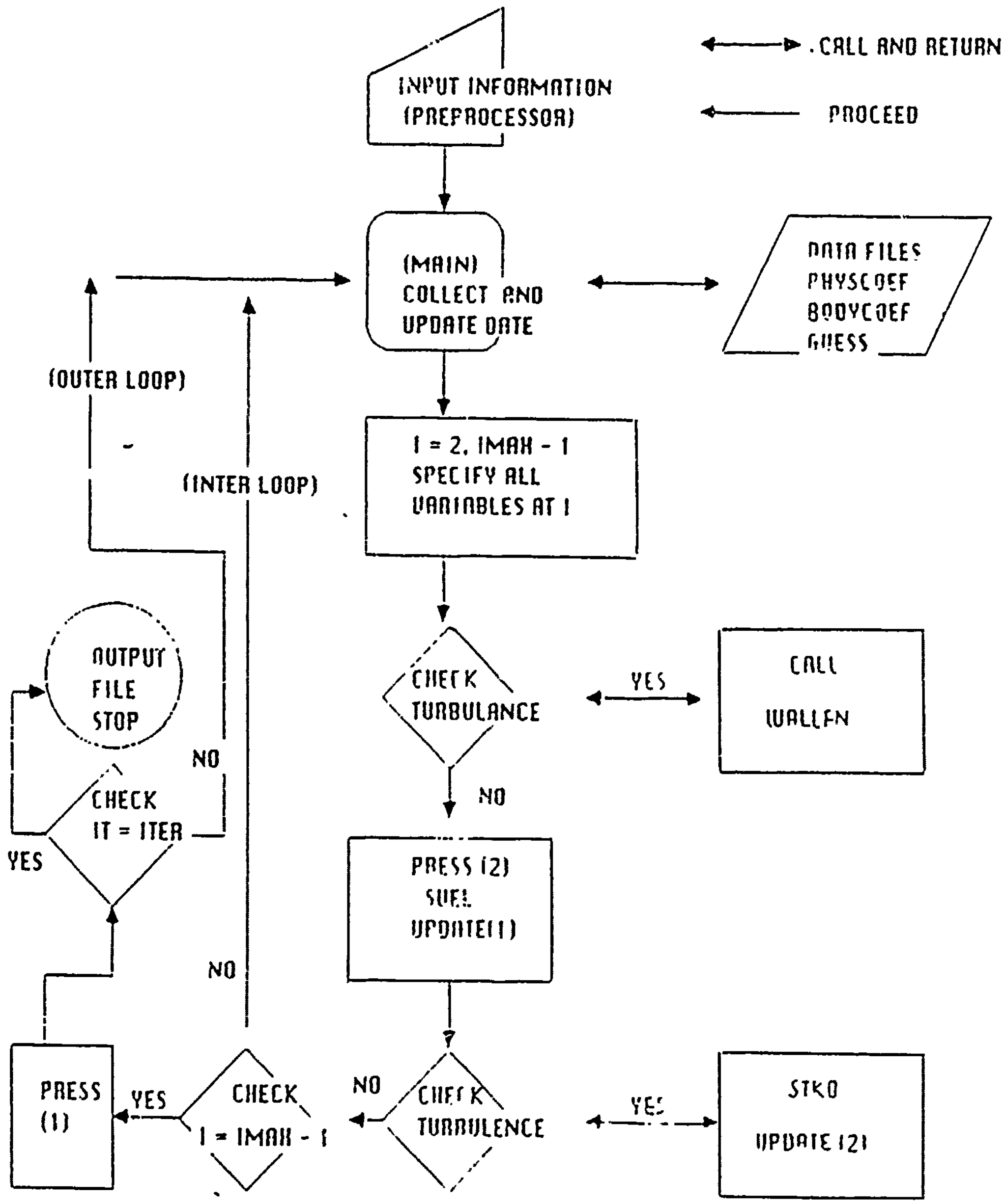


Table 5 (B-1) : The Flow Chart of Main Program of FANS-3DEF

```
MAIN ─┬─ WALLFN
      ├─ PRESS(2) ──── TRIDAG
      │
      │           ┌─ EQCOE
      │           ├─ COEF
      │           │
      │           │            ┌─ TRIDAG
      ├─ SVEL ────┼─ SOLVE ────┼─ CHECK
      │           │            └─ MSI ──── CHECK
      │           ├─ VELCOR
      │           │                        ┌─ TRIDAG
      │           └─ HVEL ── SOLVE ────────┼─ CHECK
      │                                    └─ MSI ── CHECK
      │
      │           ┌─ EQCOE
      ├─ STKD ────┼─ COEF
      │           │            ┌─ TRIDAG
      │           └─ SOLVE ────┼─ CHECK
      │                        └─ MSI ──── CHECK
      │
      └─ PRESS(1) ──── TRIDAG
```

Table 6 (B-2) : The Structure of Main Program of FANS-3DEF are listed in table 3.

---

(4) HVEL

Hvel evaluates the pseudovelocity (u_i, in Eq. (14))

---

(5) MSI

MSI stands for Modified Strongly Implicit method (see Ref. 50). The subroutine solves the system of 9-point FA algebraic equations.

---

(6) PRESS(N)

PRESS(N) solves pressure equations given in Eqs. (17) and (21). N=1 refers to pressure variable in the whole domain and N=2 for pressure correction at each cross scetion given in Eq. (21).

---

(7) SOLVE

SOLVE is a solution subroutine using either MSI or tridiagonal method.

---

(8) STKD

STKD solves the turbulent transport equations for kinetic energy k and its dissipation rate ε.

---

(9) SVEL

SVEL solves the starred velocity (u^*).

---

(10) TRIDAG

TRIDAG stands for tridiagonal metric solver. It solves a set of algebraic equation have a tridiagonal matrix. The subroutine can be execute either in the row or column direction in the flow.

---

(11) UPDATE(N)

UPDATE(N) updates the outlet boundary conditions. N=1 to 3 for velocity components, 4 for pressure and 5, 6 for k and .

---

(12) VELCOR

VELCOR solves the correction velocity (u'). which is defined in the section 'Pressure Equation' Eq. (20).

---

(13) WALLFN

WALLFN is a subroutine to specify the values for u, v, w, k and ε for the first computational node from the wall if the flow is turbulent. The wall functions are specified in Eqs. (7) and (8).

B.2 I/O System

FANS-3DEF has a very flexible I/O system. The input operations can be read from a data file or interactively from the terminal. If the user uses interactive session, all input data once installed will be saved in a data file named FANS_INPUT automatically. This is useful because, if the computed result is not satisfied or the user wants to revise a portion of data, he/she may do so in the data file and then run the program without needing to type the whole input data again. The output result is always stored in the output files but the user has options to print the result after nth iteration and to choose the three types of output file. The "n" specified in the output result means that an output result is printed at the end of every n iterations.

A list of the commands, options and variables which control the I/O system are given below.

COMMANDS:

---

(1) RDFILE

This command executes FANS-3DEF from the data file named FANS_INPUT directly.

---

(2) RDINT

This command reads input data from the interactive terminal and stores them in the file named FANS_INPUT automatically.

---

(3) CHECK

This command displays the information selected by the user.

---

(4) RUN

This command executes the main program to solve the problem.

---

(5) STOP

This command is used to stop the computation and tc be out of the program FANS-3DEF.

In the command RDINT, there are some built options. Their functions and selections are:

Options		Selections
(1) GRID	--	STAggered or REGular grid system.
(2) DIMN	--	2D, AXIsymmetric or 3D dimension.
(3) LORT	--	LAMinar or TURbulent flow.
(5) TUMS	--	ONE or TWO scale k-ε turbulence model.
(6) INIT	--	UNIform or UPDate initial guess.
(7) FORM	--	TY1, TY2 or TY3 output files.
(8) END	--	To leave RDINT.

In FANS-3DEF the codes for the options and selections to be used are the ones shown in bold characters. For example, in selecting the grid system, the option is GRID. In this option, the selection of staggered grid system is

STA and the selection of regular grid system is REG. Thus one should type either GRID STA or GRID REG for staggered or regular grid system.

The size or length of variables must be given after choosing the option DIMN, INIT and FORM.

(a) The variables after the option DIMN are:

Variables	Meaning
(1) IMAX --	The maximun node number in the (ξ) direction.
(2) JMAX --	The maximun node number in the (η) direction.
(3) KMAX --	The maximun node number in the (ζ) direction.
(4) ITER --	The total iteration numbers allowed.
(5) RE --	Reynolds number.
(6) DT --	Time increatment or Δt in Eq. (13).

(b) The variables after the option INIT are:

Variables	Meaning
(1) UI --	The u velocity component of incoming flow.
(2) VI --	The v velocity component of incoming flow.

(c) The variables after the option FORM are:

Variables	Meaning
(1) IT --	The output result is printed at each IT iteration.

FANS-3DEF recognizes free format inputs. The user can type variables with real or integer number but no character. To distinguish many variables in the same line, the user should use a space or a comma between two variables. The structure of I/O system is shown in table 7.

The easiest way for the user to become familiar with the FANS-3DEF program is to illustrate I/O system with some examples. In the following sections both interactive

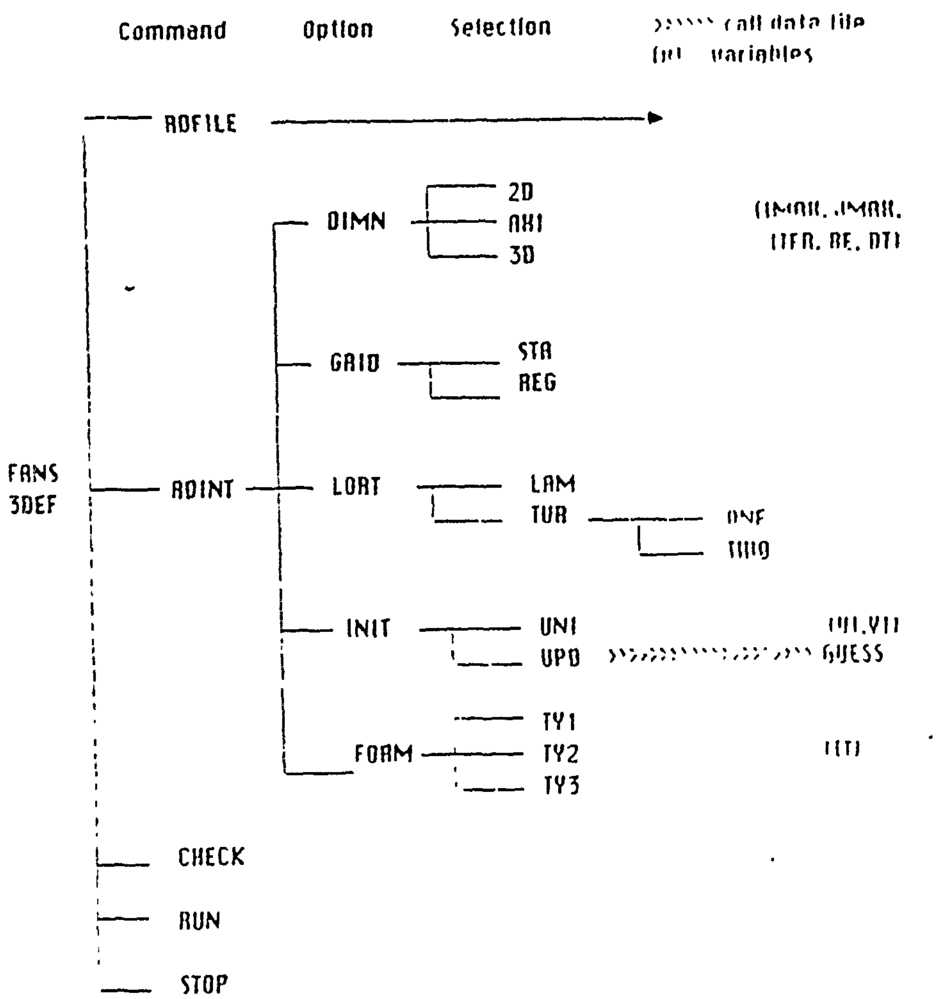


Table 7 (B-3) : The Structure of I/O system

session and data file reading will be introduced to explain how to input desired data into the FANS-3DEF program.

B.3 Interactive Session

After compiling FANS-3DEF program by FORTRAN 77 compiler, the user can then run the program. In PRIME sysytem SEG command was used to run a compiled program. Here we present the print-out which was actually shown on the screen between the two symbols --*--*--n. Where n is the number of 1, 2, 3 ..... used to distinguish the different print-out. The explanation of the print-out is given in the parentheses.

```
--*--*--1
SEG FANS-3DEF
--*--*--1
```

(A welcome message and commands will come out as)

```
--*--*--2


 ***********************************************
 *        WELCOME TO USE RESEARCH CODE         *
 *                                             *
 *             F A N S - 3 D E F               *
 *                                             *
 *              Version.1 1986                 *
 *                                             *
 *  -If you have any comments or suggestions   *
 *              please inform                  *
 *                                             *
 *          C.J. CHEN  (2216EB)                *
 *          UNIVERSITY OF IOWA                 *
 *          IOWA CITY, IOWA 52242              *
 ***********************************************


 Specify the following commands:
 (RDFILE), RDINT, CHECK, RUN, STOP
```

```
Command is >
--*--*--2
```

(If the interactive session is used, the user should type RDINT and the screen will show)

```
--*--*--3
Command is > RDINT
Specify the following options:
 -- DIMN  (2D, AXI, 3D)
 -- GRID  (STA, REG)
 -- LORT  (LAM, TUR)
 -- INIT  (UNI, UPD)
 -- FORM  (TY1, TY2, TY3)
 -- END   To leave RDINT

Option is >
--*--*--3
```

(There are six options to be chosen and no particular order is set. Thus the user can choose any option except the last option END, because if the option END is chosen the I/O system will return from the option level back to the orginal command level. The following are explanations of these options from DIMN to END. If the user chooses 2D or AXI for DIMN, then the user types DIMN 2D or DIMN AXI. The terminal will respond)

```
--*--*--4
Option is > DIMN 2D
TYPE VALUES FOR IMAX,JMAX,ITER,RE,DT
THEY ARE >

  (or)

Option is > DIMN AXI
TYPE VALUES FOR IMAX,JMAX,ITER,RE,DT
THEY ARE >
--*--*--4
```

(For 2D or axisymmetric flow, FANS-3DEF solves the fully elliptic equation on the XY cross section automatically. Thus one only has to specify variables such as IMAX, JMAX, ITER, RE and DT. For example the user may choose IMAX=20, JMAX=20, ITER=30, RE=100000 and DT=0.1, then the user can type as)

```
--*--*--5
THEY ARE > 20., 20  30. 100000,0.1
--*--*--5
```

(Since FANS-3DEF recognizes free format inputs, the user can type real or integer numbers for either real or integer variables. The user doesn't have to worry about whether variables are integer or real, the FANS-3DEF can recognize them. In order to distinguish one variable from the other, the user needs to use either a space or a comma between the variables. Now if the user chooses 3D for DIMN then the user should type DIMN 3D. The FANS-3DEF program will respond)

```
--*--*--6
Option is > DIMN 3D
TYPE VALUES FOR IMAX,JMAX,KMAX,ITER,RE,DT
THEY ARE >
--*--*--6
```

(Here the user needs one more variable KMAX if three dimensional problem is considered. After the user specifies all variables more messages will come on the terminal)

```
--*--*--7
THEY ARE > 20, 20, 20, 30, 100000., 0.1
```

```
--*--*--7
```

(After specifying the option DIMN, the user may go for the option GRID. In this option there are two selections STA (staggered grid system) and REG (regular grid system)).

```
--*--*--8
Option is > GRID STA

 (or)

Option is > GRID REG
--*--*--8
```

(Next option is LORT (laminar or turbulent). If one chooses LAM for laminar flow, then)

```
--*--*--9
Option is > LORT LAM

Option is >
--*--*--9
```

(Here no other selections or messages will be shown. However when the user chooses TUR for turbulent flow then)

```
--*--*--10
Option is > LORT TUR
Please type ONE for one-scale or TWO for two-scale k-ε
turbulence model
IT IS >
--*--*--10
```

(Here we can type ONE or TWO for one or two scale turbulence model. Although the two-scale k-ε turbulence model was not tested in this study, the option for this model is provided here for further expansion and study.)

The next option is initial guess INIT. In this option there are two selections. One is UNI for uniform distribution, the other is UPD which uses the previous result as initial guess. The message and variables of each selection are)

```
--*--*--11
Option is > INIT UPD
Please make sure you have a data file called GUESS.

 (or)

Option is > INIT UNI
Now the velocity components are uniform and
other variables are zero in the whole domain.
Please type the values for velocity components
of incoming flow, that is: UI, VI.
THEY ARE >
--*--*--11
```

(Here the user may need a hand calculator to find out the values of UI and VI. For example, if angle of attack =5 then UI=0.9961947, VI=0.0871557.)

The next option is to choose the type of output files. There are three types TY1, TY2 and TY3. The format of each output file will be listed and explained in the next section. If the user chooses TY3 then)

```
--*--*--12
Option is > FORM TY3
Type number of iterations per output
IT IS >
--*--*--12
```

(So the user needs to specify the number of iteration at which the output of computed result is made. After having specified all desired data, then user can use the option END to return to the orginal command level. They are)

```
--*--*--13
Option is > END

Specify the following commands:
 (RDFILE), RDINT, CHECK, RUN, STOP

Command is >
--*--*--13
```

(The user is returned to the command level. The user can use command CHECK to find out whether the desired data had been read correctly or not)

```
--*--*--14
Command is > CHECK

 ******************************************************
 *  The FANS-3DEF program will run under the following *
 *  conditions:                                         *
 *                                                      *
 *  -- DIMN is:3D                                       *
 *  -- GRID is:STA                                      *
 *  -- LORT is:TUR                                      *
 *  -- TUMS is:TWO                                      *
 *  -- INIT is:UNI                                      *
 *            ** UI=0.9961947  VI=0.0871557             *
 *  -- FORm is:TY3                                      *
 *            ** at     5     iteration                 *
 *    IMAX=20    JMAX=20    KMAX=20    ITER=30          *
 *    RE=  0.1000e+05   DT=  0.1000E+00                 *
 ******************************************************


 Specify the following commands:
  (RDFILE), RDINT, CHECK, RUN, STOP

 Command is >
 --*--*--14
```

(If the user finds some input data in which changes are needed, he/she may use command RDINT again and give the desired data. These new data will replace the original data. If all the desired data are correct and input data files (PHYSBODY, GUESS) are ready, then the user may use command RUN to call subroutine SOLVER to solve the problem.

So far, only the correct typing was used as an example. Suppose the user has made some typing errors. FANS-3DEF will return a warning message immediately and the user may retype it. For example)

(in command level)

```
--*--*--15
Command is > CHECC
  **INVALID COMMAND, CHECK MANUAL!

 Specify the following commands:
 (RDFILE), RDINT, CHECK, RUN, STOP

Command is >
```

(or in option level)

```
Option is > LOT
  **INVALID OPTION, CHECK MANUAL!

Option is >
--*--*--15
```

(If the user forgets typing selection after the option, then FANS-3DEF will give the message and the user can add those selections immediately. For example)

```
--*--*--16
Option is > LORT
NEED SELECTIONS  1 THROUGH  1
SELS:
--*--*--16
```

(Here "1 THROUGH 1" means need selections from selection 1 to selection 1. Since for LORT there is only one selection so we can just type LAM or TUR.

If user wants to leave the FANS-3DEF program, he/she can do so by typing command STOP then a good-bye message will come out. It is)

```
--*--*--17
Command is > STOP

 ************************************************
 *    BYE NOW !!                                *
 *                                              *
 *           -- Thank you for using FANS-3DEF   *
 ************************************************

--*--*--17
```

B.4 Reading from Data File

In this section the same data file FANS_INPUT which is created by the interactive session is used as an example. Before using this data file, the user must add command RDFILE at the first line of the file and corrects some errors that were made during the creation of the data fiel FANS_INPUT. The following is an example of the correct list of FANS_INPUT.

```
-------------------------------------
RDFILE
RDINT
DIMN 3D
 20 20 20 30  0.1000E+05  0.1000E+00
CORD BOD
```

```
LORT TUR
TWO
FORM TY3
  5
END
CHECK
RUN
STOP
-------------------------------------
```

Then, run the program FANS-3DEF as before.

```
--*--*--18
SEG FANS-3DEF
--*--*--18
```

(A welcome message and commands will come out as)

```
--*--*--19

 **********************************************
 *       WELCOME TO USE RESEARCH CODE         *
 *                                            *
 *             F A N S - 3 D E F              *
 *                                            *
 *              Version.1 1986                *
 *                                            *
 *  -If you have any comments or suggestions  *
 *              please inform                 *
 *                                            *
 *           C.J. CHEN  (2216EB)              *
 *           UNIVERSITY OF IOWA               *
 *           IOWA CITY, IOWA 52242            *
 **********************************************

 Specify the following commands:
 (RDFILE), RDINT, CHECK, RUN, STOP

Command is >
--*--*--19
```

(Now type command RDFILE and on a terminal it appears as)

```
--*--*--20
Option is > RDFILE
```

```
Do you want to see the procedure on the terminal(Y/N)
Answer is >
--*--*--20
```

(If the user already has the general idea of the whole procedure, he/she may use N (no) to save time. In this case the FANS-3DEF only provides information of the total input data and the good-bye message when the program is finished. They are)

```
--*--*--21
Answer is > N

*******************************************************
*  The FANS-3DEF program will run under the following *
*  conditions:                                         *
*                                                      *
*  -- DIMN is:3D                                       *
*  -- GRID is:STA                                      *
*  -- LORT is:TUR                                      *
*  -- TUMS is:TWO                                      *
*  -- INIT is:UNI                                      *
*            ** UI=0.9961947  VI=0.0871557             *
*  -- FORm is:TY3                                      *
*            ** at     5     iteration                 *
*   IMAX=20    JMAX=20    KMAX=20    ITER=30           *
*   RE=  0.1000e+05   DT=  0.1000E+00                  *
*******************************************************
```

(If the subroutine SOLVER is executed completely, then good-bye message will come out.)

```
*******************************************
*   BYE NOW !!                            *
*                                         *
*       -- Thank you for using FANS-3DEF  *
*******************************************
OK,
--*--*--21
```

If the user still desires to see the whole procedure on the screen he may choose Y (yes) instead of N (no) at the last question.

B.5 Format Of Input Data File

There are two kinds of input data files in the FANS-3DEF program. One is the data file named PHYSBODY for coordinate relationships, the other is the data file named GUESS for initial guess of the variables u_i, p, k and ε. Since these two input data files must be read immediately after the command RUN is executed, the user has to create these two data files before running the program and has to make sure that they have the same format as shown in the following. Otherwise, a I/O error message will be shown on the screen and the program FANS-3DEF will be forced out of the running mode by PRIME computer.

B.5.1 PHYSBODY

The format of data file PHYSBODY for 2D and axisymmetric flows are:

```
**************************************************
READ(,1) (NBOSE(I),I=1,2)
READ(,2)((X(I,J),I=1,IMAX),J=1,JMAX)
READ(,2)((Y(I,J),I=1,IMAX),J=1,JMAX)
READ(,2)((F1(I,J),I=1,IMAX),J=1,JMAX)
READ(,2)((F2(I,J),I=1,IMAX),J=1,JMAX)
   1 FORMAT(1X,6I10)
   2 FORMAT(1X,6E12.4)
**************************************************
```

Here the first statement NBOSE is the integer number between 1 to IMAX. The user needs to specify two different positions under NBOSE, one for the leading edge and the other for the trailing edge. NBOSE(1) is the nodal point denoting the leading edge of the body, NBOSE(2) is the nodal point denoting the trailing edge of the body. The second and third statement X and Y are the cartesian coordinates for 2D body shape or the cylindrical coordinates for axisymmetric body shape. F1 and F2 are the control functions given in Eq. (8).

Then the format of PHYSBODY for 3D body shape are:

```
*****************************************************
READ(,1) (NBOSE(I),I=1,2)
READ(,2)(((X(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
READ(,2)(((Y(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
READ(,2)(((Z(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
READ(,2)(((F1(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
READ(,2)(((F2(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
READ(,2)(((F3(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
   1 FORMAT(1X,6I10)
   2 FORMAT(1X,6E12.4)
*****************************************************
```

For 3D body the meaning of each variable is the same as 2D flow and two more READ statements for the third coordinate Z and control function F3 are added.

B.5.2 GUESS

The user has two options for INIT (initial guess), one is UNI (uniform initial guess) and the other is UPD (update initial guess). If the user chooses UNI then the FANS-3DEF program will assume that initial velocity componenets are uniform and other variables i.e. p, k and ε are zero in the whole domain. If UPD was chosen, then the user has to prepare a data file GUESS according to following format.

The format of GUESS are

```
***********************************************   *
READ(,1) UI, VI
READ(,2)(((U(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
READ(,2)(((V(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
READ(,2)(((W(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
READ(,2)(((PR(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
READ(,2)(((TK(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
READ(,2)(((TD(I,J,K),I=1,IMAX),J=1,JMAX),K=1,KMAX)
   1 FORMAT(1X,6I10)
   2 FORMAT(1X,6E12.4)
**************************************************
```

Here UI, VI are the velocity components of the incoming flow, and U, V, W, PR, TK, TD are the three velocity components, pressure, turbulent kinetic energy k and its dissipation rate ε. For laminar flow, the user may just specify the turbulent kinetic energy TK and its dissipation rate TD all are zero. For 2D or AXI flow, the user may just think KMAX=1 and W velocity component is zero everywhere.

B.6 Format Of Output Files

There are three options, TY1, TY2 and TY3 for the user to choose. TY1 creates a file called 'RESULT.1'. The format of RESULT.1 is the same as the input data file GUESS. The reason for creating the output file RESULT.1 to be the same format as the input file GUESS is so that later RESULT.1 can be used as the initial guess for other similar problems, or when the selected iteration number ITER is not large enough to obtain a converget solution. Since RESULT.1 is designed for computer reading, it may not be a good output format for users to read. The second option TY2 creates a file called 'RESULT.2' which has a readable format. They are

```
*******************************************************
DO 10 I=1,IMAX
WRITE(,3) IT, I, X(I)
WRITE(,4)
DO 9 K=1,KMAX
WRITE(,5) K
DO 8 J=1,JMAX
WRITE(,6) U(I,J,K),V(I,J,K),W(I,J,K),
          TK(I,J,K),TD(I,J,K),PR(I,J,K)
   8 CONTINUE
   9 CONTINUE
  10 CONTINUE
   3 FORMAT(//3X,'NO. OF ITERATION=',I3,4X,'STATION=',
           I3,4X,'X=',F7.4)
   4 FORMAT(//4X,'U VEL',7X,'V VEL',7X,'W VEL',6X,
           'TURB KE',5X,'TURB DISP',2X,'  PRESS    '//)
   5 FORMAT(5X,'AT K= ',I3)
   6 FORMAT(1X,6E12.4)
*******************************************************
```

The third choice of the output format is TY3 which provides both 'RESULT.1' and 'RESULT.2' for computer and user's reading.

B.7 Program Listing

```
C
C***********************************************************
C
C.... INTERACTIVE PROGRAM (I/O SYSTEM) OF FANS-3DEF
C
C.... PROGRAMMED BY WU-SUN CHENG
C.... MAY, 1985
C
C***********************************************************
C
      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*10 COM, ARG(5), CMANDS(10)
      CHARACTER*10 OPTION(10), OPT
      CHARACTER*10 BLANKS
      CHARACTER*10 GRID, DIMN, LORT, INIT, TUMS, FORM, YON
      CHARACTER*80 LINE
      CHARACTER  BLANK
      INTEGER F, CRT
      LOGICAL HELP, ASK
      COMMON/COEF1/ IMAX,JMAX,KMAX
      COMMON/COEF2/ RE,DT,IPRINT,ITER,CNU,AK,E
      COMMON/COEF3/ NA23,NSR,LOT,NTS,INI,NTY
      COMMON/COEF4/ UI,VI,M1,M2,M3
      COMMON/UVW8/ C1,C2,CK,CD
      DATA BLANK/' '/, BLANKS/'          '/
      DATA CMANDS(1) /'RDFILE    '/, CMANDS(2) /'RDINT     '/,
     $     CMANDS(3) /'CHECK     '/,
     $     CMANDS(4) /'RUN       '/, CMANDS(5) /'STOP      '/
      DATA OPTION(1) /'DIMN      '/, OPTION(2) /'GRID      '/,
     $     OPTION(3) /'LORT      '/, OPTION(4) /'INIT      '/,
     $     OPTION(5) /'FORM      '/, OPTION(6) /'END       '/
C
C.... PRINT WELCOME MESSAGE
C
      CALL LEAD
C
      CRT=1
      RET=0.
       PN=0.
C
      OPEN(UNIT=8, FILE='FANS_INPUT')
C
    5 IF(PN .LT. 5) THEN
      PRINT *, ' '
      PRINT *, ' Specify the following commands:'
      PRINT *, ' (RDFILE), RDINT, CHECK, RUN, STOP'
      PRINT *, ' '
      PRINT *, 'Command is > '
      END IF
C
C.... INITIALIZE 'LINE' AND 'COM' TO ALL BLANKS
```

```
C
      F=1
      DO 6 I=1,80
    6 LINE(I:I)=BLANK
      COM=BLANKS
C
       READ(CRT,1000) LINE
       IF(RET .LT. 5.) WRITE(8,1000) LINE
C
C.... GET THE FIRST WORD FROM THE LINE (WHICH IS COMMAND)
C
      CALL PARSE(LINE, COM, F, LENGTH)
C
C.... IGNORE AN ALL BLANK LINE
C
      IF (LENGTH .EQ. 0) GO TO 5
C
C.... FIND OUT COMMAND AND EXECUTE IT
C
C.... COMMAND 1, READ FROM DATA FILE
C
      IF(COM(1:LENGTH) .EQ. CMANDS(1)(1:LENGTH)) THEN
        CRT=8
        RET=9.
C
      PRINT *, 'Do you want to see the procedure on the termin
    1 PRINT *, ' Answer is > '
       READ(1,1000) YON
      IF(YON(1:1) .EQ. 'Y') THEN
        PN=0.
      ELSE IF(YON(1:1) .EQ. 'N') THEN
        PN=9.
      ELSE
        PRINT *, ' Please use Y or N '
        GO TO 1
      END IF
C
        GO TO 5
C
C.... COMMAND 2, READ FROM INTERACTIVE TERMINAL
C
      ELSE IF(COM(1:LENGTH) .EQ. CMANDS(2)(1:LENGTH)) THEN
   20   IF(PN .LT. 5) THEN
        PRINT *, ' '
        PRINT *, 'Specify the following options:'
        PRINT *, ' -- DIMN  (2D, AXI, 3D)'
        PRINT *, ' -- GRID  (STA, REG)'
        PRINT *, ' -- LORT  (LAM, TUR)'
        PRINT *, ' -- INIT  (UNI, UPD)'
        PRINT *, ' -- FORM  (TY1, TY2, TY3)'
        PRINT *, ' -- END   To leave RDINT'
```

```
      END IF
C
C
      F=1
      IF(PN .LT. 5) THEN
      PRINT *, ' '
      PRINT *, ' Option is > '
      END IF
C
C.... INITIALIZE 'LINE' AND 'OPT' TO ALL BLANKS
C
      DO 16 I=1,80
   16 LINE(I:I)=BLANK
      DO 17 I=1,5
   17 ARG(I)=BLANKS
      OPT=BLANKS
C
      READ(CRT,1000) LINE
      IF(RET .LT. 5.) WRITE(8,1000) LINE
C
C.... GET THE FIRST WORD FROM THE LINE (WHICH IS OPTION)
C
      CALL PARSE(LINE, OPT, F, LENGTH)
C
C.... FIND OUT OPTION AND EXECUTE IT
C
C.... OPTION 1, SPECIFY DIMENSION
C
         IF(OPT(1:LENGTH) .EQ. OPTION(1)(1:LENGTH)) THEN
           CALL ARGCHK(LINE, ARG, 1, F, HELP)
           IF(HELP) GO TO 20
           DIMN=BLANKS
           DIMN=ARG(1)
           IF(DIMN(1:2) .EQ. '2D' .OR. DIMN(1:3) .EQ. 'AXI')
             NA23=1
             IF(DIMN(1:2) .EQ. '2D') NA23=2
             IF(PN .LT. 5) THEN
             PRINT *, ' TYPE VALUES FOR IMAX,JMAX,ITER,RE,DT'
             PRINT *, ' They are > '
             END IF
             READ(CRT,*) AIMAX,AJMAX,AITER,RE,DT
             IMAX=AIMAX
             JMAX=AJMAX
             KMAX=3
             ITER=AITER
             IF(RET.LT.5) WRITE(8,2500)IMAX,JMAX,ITER,RE,DT
           ELSE IF(DIMN(1.2) .EQ  '3D') THEN
             IF(PN .LT. 5) THEN
             PRINT *, 'TYPE VALUES FOR IMAX,JMAX,KMAX,ITER,RE
             PRINT *, ' They are > '
             END IF
```

```
          READ(CRT,*) AIMAX,AJMAX,AKMAX,AITER,RE,DT
          IMAX=AIMAX
          JMAX=AJMAX
          KMAX=AKMAX
          ITER=AITER
          IF(RET.LT.5) WRITE(8,2000)IMAX,JMAX,KMAX,ITER,RE
        ELSE
          WRITE(1,3000)
        END IF
C
C.... OPTION 2, SPECIFY GRID SYSTEM
C
      ELSE IF(OPT(1:LENGTH) .EQ. OPTION(2)(1:LENGTH)) THEN
        CALL ARGCHK(LINE, ARG, 1, F, HELP)
        IF(HELP) GO TO 20
        GRID=BLANKS
        GRID=ARG(1)
C
        IF(GRID(1:3) .EQ. 'STA') THEN
          IF(PN .LT. 5) THEN
          PRINT *, ' Using staggered grid system '
          END IF
        ELSE IF(GRID(1:3) .EQ. 'REG') THEN
          IF(PN .LT. 5) THEN
          PRINT *, ' Using regular grid system '
          END IF
        ELSE
          WRITE(1,3000)
        END IF
C
C.... OPTION 3, CHECK LAMINAR OR TURBULENCE
C
      ELSE IF(OPT(1:LENGTH) .EQ. OPTION(3)(1:LENGTH)) THEN
        CALL ARGCHK(LINE, ARG, 1, F, HELP)
        IF(HELP) GO TO 20
        LORT=BLANKS
        LORT=ARG(1)
C
        IF(LORT(1:3) .EQ. 'LAM') THEN
          LOT=1
          GO TO 20
        ELSE IF(LORT(1:3) .EQ. 'TUR') THEN
          LOT=2
          IF(PN .LT. 5) THEN
          PRINT *, ' Please type ONE for one-scale or TWO
          PRINT *, ' two-scale turbulence model'
          PRINT *, ' It is > '
          END IF
          READ(CRT,1000) TUMS
          IF(RET .LT. 5) WRITE(8,1000) TUMS
C
```

```
C.... NEAR WALL COEFFICIENTS
C.
      CNU=0.09D0
       AK=0.41D0
        E=9.D0
C
C.... TURBULENCE SCALE
C
      IF(TUMS(1:3) .EQ. 'ONE') THEN
        NTS=1
        CK=1.D0
        CD=1.3D0
        C1=1.44D0
        C2=1.92D0
      ELSE IF(TUMS(1:3) .EQ. 'TWO') THEN
        NTS=2
        CK=1.D0
        CD=0.045D0
        DRE=1.D0/DSQRT(RE)
        C1=17.5D0*DRE
        C2=18.9D0*DRE
      END IF
           ELSE
             WRITE(1,3000)
           END IF
C
C.... OPTION 4, SPECIFY INITIAL GUESS
C
          ELSE IF(OPT(1:LENGTH) .EQ. OPTION(4)(1:LENGTH)) THEN
          CALL ARGCHK(LINE, ARG, 1, F, HELP)
          IF(HELP) GO TO 20
          INIT=BLANKS
          INIT=ARG(1)
C
          IF(INIT(1:3) .EQ. 'UNI') THEN
            INI=1
            IF(PN .LT. 5) THEN
            PRINT *, ' Now the velocity components are uniform
            PRINT *, ' other variables are zero in the whole d
            PRINT *, ' Please type the values for velocity com
            PRINT *, ' of incoming flow, that is: UI, VI.'
            PRINT *, 'They are > '
            READ(CRT,*) UI, VI
            IF(RET.LT.5) WRITE(8,1500) UI, VI
            END IF
          ELSE IF(INIT(1:3) .EQ. 'UPD') THEN
            INI=2
            IF(PN .LT. 5) THEN
            PRINT *, ' Please make sure you have a data file c
            PRINT *, ' GUESS.'
            END IF
```

```
          ELSE
            WRITE(1,3000)
          END IF
C
C.... OPTION 5, SPECIFY OUTPUT FILES
C
          ELSE IF(OPT(1:LENGTH) .EQ. OPTION(5)(1:LENGTH)) THEN
            CALL ARGCHK(LINE, ARG, 1, F, HELP)
            IF(HELP) GO TO 20
            FORM=BLANKS
            FORM=ARG(1)
C
            IF(FORM(1:3) .EQ. 'TY1') THEN
              NTY=1
              GO TO 57
            ELSE IF(FORM(1:3) .EQ. 'TY2') THEN
              NTY=2
              GO TO 57
            ELSE IF(FORM(1:3) .EQ. 'TY3') THEN
              NTY=3
              GO TO 57
            ELSE
              WRITE(1,3000)
              GO TO 20
            END IF
   57         IF(PN .LT. 5) THEN
              PRINT *, ' Type number of iterations per output'
              PRINT *, ' It is > '
              END IF
              READ(CRT,*) AIT
              IT=AIT
              IPRINT=IT
              IF(RET .LT. 5) WRITE(8,2000) IT
C
C.... OPTION 6, LEAVE RDINT
C
          ELSE IF(OPT(1:LENGTH) .EQ. OPTION(6)(1:LENGTH)) THEN
            GO TO 5
          ELSE
            PRINT *, '  **INVALID OPTION, CHECK MANUAL!'
          END IF
C
          GO TO 20
C
C
C.... COMMAND 3, CHECK INFORMATION
C
      ELSE IF(COM(1:LENGTH) .EQ. CMANDS(3)(1:LENGTH))  THEN
        PRINT *, ' '
        PRINT *, ' FANS-3DEF will run under the following cond
        PRINT *, ' '
```

```
      PRINT *, ' -- DIMN is:', DIMN
      PRINT *, ' -- GRID is:', GRID
      PRINT *, ' -- LORT is:', LORT
      PRINT *, ' -- TUMS is:', TUMS
      PRINT *, ' -- INIT is:', INIT
      PRINT *, '          ** UI=', UI, '  VI=', VI
      PRINT *, ' -- FORM is:', FORM
      PRINT *, '           ** at',IT,' iteration'
      PRINT 4000, IMAX,JMAX,KMAX,ITER,RE,DT
C
C.... COMMAND 4, CALL MAIN PROGRAM
C
    ELSE IF(COM(1:LENGTH) .EQ. CMANDS(4)(1:LENGTH)) THEN
      IF(NA23 .EQ. 2) THEN
      CALL MAIN2D
      ELSE
      CALL MAIN3D
      END IF
C
C.... COMMAND 5, STOP THE PROGRAM
C
    ELSE IF(COM(1:LENGTH) .EQ. CMANDS(5)(1:LENGTH)) THEN
      PRINT *, ' '
      PRINT *, '  BYE NOW !! '
      PRINT *, ' '
      PRINT *, '          -- Thank you for using FANS-3DEF '
      PRINT *, ' '
      PRINT *, ' '
      CLOSE (8)
      CALL EXIT
C
    ELSE
      PRINT *, '  **INVALID COMMAND, CHECK MANUAL!'
C
    END IF
C
    GO TO 5
C
C
 1000 FORMAT(A)
 1500 FORMAT(6F8.4)
 2000 FORMAT(4I3,2E12.4)
 2500 FORMAT(3I3,2E12.4)
 3000 FORMAT(3X, ' **INVAID ARGUMENT, CHECK MANUAL!')
 4000 FORMAT(3X,'IMAX=',I3,3X,'JMAX=',I3,3X,'KMAX=',I3,3X,'ITE
     $      /3X,'RE=',E12.4,3X,'DT=',E12.4)
    END
C
C*************************************************************
C
C..   SUBROUTINE LEAD
```

```
C
C**************************************************************
C
      SUBROUTINE LEAD
      CHARACTER*60 B(17)
      INTEGER I
      B(1)='
      B(2)='
      B(3)='  ***********************************************
      B(4)='  *          WELCOME TO USE RESEARCH CODE        *
      B(5)='  *                                              *
      B(6)='  *               F A N S - 3 D E F              *
      B(7)='  *                                              *
      B(8)='  *                Version.1  1986               *
      B(9)='  *                                              *
      B(10)='  *    If you have any comments or suggestions  *
      B(11)='  *               please inform                 *
      B(12)='  *                                             *
      B(13)='  *          Prof.  C.J. CHEN  (2216EB)         *
      B(14)='  *            The University Of Iowa           *
      B(15)='  *            Iowa City, Iowa 52242            *
      B(16)='  *               (319)  353-4473               *
      B(17)='  ***********************************************
C
      DO 10 I=1,17
      PRINT 100, B(I)
   10 CONTINUE
C
  100 FORMAT(A)
      RETURN
      END
C
C************************************************************
C
C.... SUBROUTINE ARGCHK
C
C************************************************************
C
C.... CHECKS TO SEE IF NUNBER OF ARGUMENTS SPECIFIED IS EQUAL
C.... TO 'NUMARG'. IF NOT, THE USER IS PROMPTED FOR NECESSARY
C.... ARGUMENTS. IF ANY OF THE ARGUMENTS IS 'HELP', THE HELP
C.... FLAG IS RETURNED 'TRUE'. 'NXTCOL' IS THE LOCATION IN
C.... 'LINE' WHERE SEARCH FOR THE ARGUMENTS BEGINS.
C
C
      SUBROUTINE ARGCHK (LINE, ARG, NUMARG, NXTCOL, HELP)
      IMPLICIT REAL*8 (A-H, O-Z)
      INTEGER CRT, NXTCOL, START, NARG
      CHARACTER*(80) LINE
      CHARACTER*10 ARG(10), BLANKS
      LOGICAL HELP
```

```
      DATA CRT/1/
      DATA BLANKS/'          '/
C
      NARG=1
      HELP=.FALSE.
C
C
  100 START=NARG
C
C.... GET THE NEXT ARGUMENT
C
      DO 110 I=START, NUMARG
      CALL PARSE(LINE, ARG(I), NXTCOL, LENGTH)
      IF(LENGTH .EQ. 0) GO TO 120
      IF(ARG(I) (1:4) .EQ. 'HELP') HELP=.TRUE.
      NARG=NARG+1
  110 CONTINUE
C
C.... RETURN BECAUSE ALL ARGUMENTS ARE SPECIFIED
C
      RETURN
C
C.... REACH HERE IF SOME ARGUMENTS ARE MISSING
C
  120 IF (HELP) RETURN
C
C
      WRITE(CRT,2000) NARG, NUMARG
 2000 FORMAT(' NEED SELECTIONS ',I2,' THROUGH ',I2)
       PRINT *, ' Selections '
C
C
      READ(CRT,1000) LINE
 1000 FORMAT(A80)
      NXTCOL=1
      GO TO 100
C
      END
C
C*****************************************************
C
C.... SUBROUTINE PARSE
C
C*****************************************************
C
C.... PARSES THE 'LINE' AND RETURNS NEXT 'WORD' WHICH IS 'LENG
C.... LONG. IF THE WORD IS LONGER THAN 'MAXLEN' CHARACTERS THE
C.... EXTRA CHARACTERS ARE IGNORED.
C
      SUBROUTINE PARSE (LINE, WORD, NXTCOL, LENGTH)
      IMPLICIT REAL*8 (A-H,O-Z)
```

```
      PARAMETER (MAXLEN=10)
      CHARACTER *(*) LINE
      CHARACTER *(*) WORD
      CHARACTER BLANK, COMMA
      LOGICAL FIRST
      DATA BLANK/' '/, COMMA/','/
C
C
      LENGTH=0
      IST=NXTCO'
      FIRST=.TRUE.
C
      DO 90 I=1,MAXLEN
        WORD(I:I)=BLANK
   90 CONTINUE
C
C
      DO 100 I=IST,LEN(LINE)
        NXTCOL=I
C
        IF(LINE(I:I) .EQ. BLANK .OR. LINE(I:I) .EQ. COMMA) THE
          IF(.NOT. FIRST) RETURN
          GO TO 100
        ELSE
          FIRST=.FALSE.
          IF(LENGTH .LT. MAXLEN) THEN
            LENGTH=LENGTH+1
            WORD(LENGTH:LENGTH)=LINE(I:I)
          END IF
        END IF
  100 CONTINUE
C
      RETURN
      END
C
C*************************************************
C
C.... SUBROUTINE MAIN2D
C
C*************************************************
C.... MAIN2D IS USED TO SOLVE 2D FLAT PLAT PROBLEM
C.... WITH ANGLE OF ATTACK

      SUBROUTINE MAIN
      IMPLICIT REAL*8 (A-H,O-Z)

$INSERT BLOCK.MAIN
      COMMON/COEF1/ IE,JE,KE
      COMMON/COEF2/ ARE,ADT,IPRINT,ITER,CNUU,AKK,EE
      COMMON/COEF3/ NA23,NSR,LOT,NTS,INITIAL,NTY
      COMMON/COEF4/ UI,VI,M1,M2,M3
```

```
      DIMENSION UL(99), UTL(99), TAUW(99), NT2(2)
      COMMON/B01/ UBO(82,44)
      COMMON/B02/ VBO(82,44)
      COMMON/B03/ PRBO(82,44)
      COMMON/B04/ TKBO(82,44)
      COMMON/B05/ TDBO(82,44)

      OPEN(6,FILE='DATA.IN')
      NNX=IE-1
      NNY=JE-1
      RE=ARE
      DT=ADT
      NB1=M1
      NB3=M2
      NB2=M3
      II=49
      RET=RE/DT
      NT2(1)=NB2
      NT2(2)=NB2
      JE2=JE+2
      JE3=JE+3

C---  CALL BFC TO CALCULATE BODY-FITTED COORDINATE

       CALL BFC

      ABC=0.33206
      DO 35 I=NB1+1,NB3
      XX=X(I,2)
      REX=DSQRT(RE*XX)
      UTL(I)=DSQRT(ABC/REX)
   35 CONTINUE
      DO 33 I=19,29
   33 UL(I)=1.0*UI
      UL(30)=0.9994*UI
      UL(31)=0.9852*UI
      UL(32)=0.9250*UI
      UL(33)=0.8200*UI
      UL(34)=0.7050*UI
      UL(35)=0.6027*UI
      UL(36)=0.5275*UI
      UL(37)=0.4862*UI
C
      IF( INITIAL .EQ. 1 ) THEN
      DO 20 I=1,IE
      DO 20 J=1,44
        UBO(I,J)=UI
        VBO(I,J)=VI
        IF((I.GE.NB1.AND.I.LE.NB3).AND.
     $     (J.EQ.2.OR.J.EQ.JE2)) THEN
        UBO(I,J)=0.
```

```
        VBO(I,J)=0.
        END IF
        PRBO(I,J)=0.D0
        TKBO(I,J)=1.D-9
        TDBO(I,J)=1.D-9
 20   CONTINUE
      ELSE
        PRINT *, 'READ GUESS'
        OPEN(10,FILE='GUESSP',STATUS='OLD')
        OPEN(11,FILE='GUESSU',STATUS='OLD')
        OPEN(12,FILE='GUESSV',STATUS='OLD')
        OPEN(14,FILE='GUESSKD',STATUS='OLD')
        READ (10,2400)  (( PR(I,J),I=1,IE ), J=2,JE )
        READ (11,2400)  ( ( U(I,J),I=1,IE ), J=2,JE )
        READ (12,2400)  ( ( V(I,J),I=1,IE ), J=2,JE )
        READ(14,2400)  (( TK(I,J),I=1,IE), J=2,JE )
        READ(14,2400)  (( TD(I,J),I=1,IE), J=2,JE )
        CLOSE(10)
        CLOSE(11)
        CLOSE(12)
        CLOSE(14)

      END IF

      DO 60 I=1,IE
      DO 60 J=2,JE
      DU(I,J)=0.0
      DV(I,J)=0.0
      DS(I,J)=0.0
      DST(I,J)=0.0
      PP(I,J)=0.0
 60   CONTINUE

      PRINT *, 'BEGIN'
      OPEN(7,FILE='RESULT')
      OPEN(5,FILE='EPST')
      OPEN(15,FILE='DATAP')
      OPEN(16,FILE='DATAU')
      OPEN(17,FILE='DATAV')
      OPEN(18,FILE='DATAKD')
      OPEN(19,FILE='DATAEV')
      OPEN(20,FILE='DATA.ADD')
      ITC=0
      ITC1=0
      ID=0

C.... ITERATION LOOP

      JMM=8
      JMN=JMM-1
      DO 999 IT=1,ITER
```

```
      ITC=ITC+1
      ITC1=ITC1+1
      DO 888 IBO=1,2
      IF(IBO.EQ.1) THEN
      DO 235 I=1,IE
      DO 234 J=2,JE
      U(I,J)=UBO(I,J)
      V(I,J)=-VBO(I,J)
      PR(I,J)=PRBO(I,J)
      TK(I,J)=TKBO(I,J)
      TD(I,J)=TDBO(I,J)
  234 CONTINUE
      U(I,1)=UBO(I,JE3)
      V(I,1)=-VBO(I,JE3)
      PR(I,1)=PRBO(I,JE3)
      TK(I,1)=TKBO(I,JE3)
      TD(I,1)=TDBO(I,JE3)
  235 CONTINUE
      DO 236 I=1,IE
  236 V(I,JE)=-VI
      DO 237 J=1,JE
      V(NNX,J)=-VI
  237 V(IE,J)=-VI
      ELSE
      DO 345 I=1,IE
      DO 344 J=2,JE
      JJ=J+JE
      U(I,J)=UBO(I,JJ)
      V(I,J)=VBO(I,JJ)
      PR(I,J)=PRBO(I,JJ)
      TK(I,J)=TKBO(I,JJ)
      TD(I,J)=TDBO(I,JJ)
  344 CONTINUE
      U(I,1)=UBO(I,3)
      V(I,1)=VBO(I,3)
      PR(I,1)=PRBO(I,3)
      TK(I,1)=TKBO(I,3)
      TD(I,1)=TDBO(I,3)
  345 CONTINUE
      DO 346 I=1,IE
  346 V(I,JE)=VI
      DO 347 J=1,JE
      V(NNX,J)=VI
  347 V(IE,J)=VI
      END IF
      DO 456 I=1,IE
      DO 456 J=1,JE
      RU(I,J)=U(I,J)
      RV(I,J)=V(I,J)
      RPR(I,J)=PR(I,J)
      RTK(I,J)=TK(I,J)
```

```
 456 RTD(I,J)=TD(I,J)
     NB2=NT2(IBO)
     DO 10 I=1,IE
     JB(I)=JMM
     IF(I.LT.NB2.OR.I.GT.NB3+1) JB(I)=3
10   CONTINUE

     DO 30 I=NB2,IE
     DO 30 J=2,JE
     IF(TD(I,J).LT.1.D-9) TD(I,J)=1.D-9
     EV(I,J)=TK(I,J)*TK(I,J)/TD(I,J)*CNU
30   CONTINUE
     CALL CHECK(EV,6,IE,JE)
     DO 50 I=NB2,IE
     DO 50 J=5,JE
     IF(EV(I,J-1).LE.EV(I,J-2) .AND. EV(I,J-1).LT.EV(I,J))
    $ EV(I,J)=EV(I,J-1)
50   CONTINUE
     DO 90 I=NB2-1,1,-1
     DO 90 J=2,JE
     TK(I,J)=TK(I+1,J)*0.8
     TD(I,J)=TD(I+1,J)*0.8
     EV(I,J)=EV(I+1,J)*0.8
  90 CONTINUE

C    CALL BFC(IBO)

C   ********************************************
C   *   MODIFIED SIMPLER ALGORITHM
C   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

C---  CALCULATE THE MOMENNTUM EQUATIONS

      CALL SVEL

C---  CALCULATE THE PRESSURE CORRECTION

      CALL PRESS(2)

C---  COMPUTE THE PSEUDO-VELOCITY FIELD

      CALL HVEL

C---  CALCULATE THE PRESSURE FIELD

      CALL PRESS( 1 )

C---  CALCULATE TURBULENT VARIABLES

      CALL STKD
```

```
C---  END OF ONE SWEEP

      WRITE(1,2500) NB2
      WRITE(5,2500) NB2
      IF(IT .GT. 2) THEN
      IF(U(NB2,JMN) .LT. UL(NB2+1)) NB2=NB2+1
      IF(U(NB2,JMN) .GT. U(NB2-1,JMN)) NB2=NB2-1
      END IF

C    ****************************************************
C    * RESULTS
C    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

      WRITE(7,4111) IT
 4111 FORMAT(/5X,'NO.  OF  ITERATION =',I5,5X,'PRE. DIS.')
      WRITE(7,2400) (PR(I,2),I=1,IE)
      WRITE(7,4112)
 4112 FORMAT(/5X,'SKIN-FRICTION COEFFICIENT ')
      DO 4222 I=NB1+1,NB3
      IF(I.LT.NB2) THEN
      UT=UTL(I)*U(I,JMN)/UL(I)
      ELSE
      UT=UTA(I)
      END IF
      TAUW(I)=2.*UT*UT
 4222 CONTINUE
      WRITE(7,2400) (TAUW(I),I=NB1+1,NB3)
      WRITE(7,4113)
 4113 FORMAT(/5X,'CENTERLINE VELOCITY')
      WRITE(7,2400) (U(I,2),I=NB3+1,IE)

 550  IF(ITC1 .EQ. IPPINT) THEN

      PRINT *, 'WRITE RESULT'

      DO 123  I=15,70
C     IF(I.GT.30.AND.I.LT.45) GO TO 123
      WRITE(7,2000) IT, I, X(I,2)
      WRITE(7,2100)
      DO 110 J=2,JE
      WRITE(7,2400) U(I,J),V(I,J),PR(I,J),TK(I,J),TD(I,J),EV(I
 110  CONTINUE
 123  CONTINUE

C.... WRITE SHEAR STRESS, MOMENTUM THICKNESS, REYOLNDS STRESS

      DO 120 I=1,NNX
      UVOS(I,2)=0.0
      UVOS(I,JE)=0.0
      THAT(I)=0.0
      DO 120 J=3,NNY
```

```
      THAT(I)=THAT(I)+(U(I,J)-U(I,J)*U(I,J))*(Y(I,J)-Y(I,J-1))
      TD11=B11(I,J)/DSJ(I,J)
      TD22=B22(I,J)/DSJ(I,J)
      UVOS(I,J)=EV(I,J)*((U(I,J+1)-U(I,J))*TD22
     $          +(V(I,J)-V(I-1,J))*TD11)
  120 CONTINUE

      WRITE(20,2400) (UTA(I),I=1,NNX)
      WRITE(20,2400) (THAT(I), I=1,NNX)
      WRITE(20,2400) ((UVOS(I,J),I=2,NNX),J=2,JE)

      END IF

      IF(ITC.EQ.IPRINT) THEN
      WRITE(15,2300) IT
      WRITE(16,2300) IT
      WRITE(17,2300) IT
      WRITE(18,2300) IT
      WRITE(19,2300) IT
      WRITE(15,2400) ((PR(I,J),I=1,IE),J=2,JE)
      WRITE(16,2400) (( U(I,J),I=1,IE),J=2,JE)
      WRITE(17,2400) (( V(I,J),I=1,IE),J=2,JE)
      WRITE(18,2400) ((TK(I,J),I=1,IE),J=2,JE)
      WRITE(18,2400) ((TD(I,J),I=1,IE),J=2,JE)
      WRITE(19,2400) ((EV(I,J),I=1,IE),J=2,JE)

      IF (ID.EQ.1) GO TO 500

      END IF

      EPSU=0.0
      EPSV=0.0
      EPSP=0.0
      EPTK=0.
      EPTD=0.
      EPDS=0.0
      DO 150 I=1,IE
      DO 150 J=2,JE
      EPS2=RU(I,J)-U(I,J)
      IF(DABS(EPS2).GT.DABS(EPSU)) THEN
      EPSU=EPS2
      NU=I
      MU=J
      END IF
      EPS2=RV(I,J)-V(I,J)
      IF(DABS(EPS2).GT.DABS(EPSV)) THEN
      EPSV=EPS2
      NV=I
      MV=J
      END IF
      EPS2=RPR(I,J)-PR(I,J)
```

```
      IF(DABS(EPS2).GT.DABS(EPSP)) THEN
      EPSP=EPS2
      NP=I
      MP=J
      END IF
      IF(DABS(DST(I,J)).GT.EPDS) THEN
      EPDS=DABS(DST(I,J))
      NDT=I
      MDT=J
      END IF
      IF(I.LT.NB2) GO TO 150
      EPS2=RTK(I,J)-TK(I,J)
      IF(DABS(EPS2).GT.DABS(EPTK)) THEN
      EPTK=EPS2
      NK=I
      MK=J
      END IF
      EPS2=RTD(I,J)-TD(I,J)
      IF(DABS(EPS2).GT.DABS(EPTD)) THEN
      EPTD=EPS2
      ND=I
      MD=J
      END IF

150   CONTINUE

      WRITE(1,1100) EPSU,NU,MU, IT
      WRITE(5,1100) EPSU,NU,MU,IT
      WRITE(1,1200) EPSV, NV, MV
      WRITE(5,1200) EPSV,NV,MV
      WRITE(1,1300) EPSP,NP,MP
      WRITE(5,1300) EPSP,NP,MP
      WRITE(1,1400) EPTK,NK,MK
      WRITE(5,1400) EPTK,NK,MK
      WRITE(1,1500) EPTD,ND,MD
      WRITE(5,1500) EPTD,ND,MD
      WRITE(1,1600) EPDS, NDT, MDT
      WRITE(5,1600) EPDS, NDT, MDT


      IF(DABS(EPSU).LT.EPST.AND.DABS(EPSV).LT.EPST.AND.DABS(EP
     +.LT.EPST) THEN
      ITC=IPRINT
      ID=1
      GO TO 550
      END IF
      NT2(IBO)=NB2
      IF(IBO.EQ.1) THEN
      DO 788 I=1,IE
      DO 788 J=1,JE
      UBO(I,J)=U(I,J)
```

```
      VBO(I,J)=-V(I,J)
      PRBO(I,J)=PR(I,J)
      TKBO(I,J)=TK(I,J)
      TDBO(I,J)=TD(I,J)
  788 CONTINUE
      ELSE
      DO 799 I=1,IE
      DO 799 J=1,JE
      JJ=J+JE
      UBO(I,JJ)=U(I,J)
      VBO(I,JJ)=V(I,J)
      PRBO(I,JJ)=PR(I,J)
      TKBO(I,JJ)=TK(I,J)
      TDBO(I,JJ)=TD(I,J)
  799 CONTINUE
       END IF
  888 CONTINUE
      IF(ITC.EQ.IPRINT) ITC=0
      IF(ITC1.EQ.IPRINT) ITC1=0
      DO 899 I=1,IE
      IF(I.GE.NB1.AND.I.LE.NB3) GO TO 899
      UU=.5*(UBO(I,3)+UBO(I,JE3))
      VV=.5*(VBO(I,3)+VBO(I,JE3))
      PM=.5*(PRBO(I,3)+PRBO(I,JE3))
      TKA=.5*(TKBO(I,3)+TKBO(I,JE3))
      TDA=.5*(TDBO(I,3)+TDBO(I,JE3))
      UBO(I,2)=UU
      VBO(I,2)=VV
      PRBO(I,2)=PM
      TKBO(I,2)=TKA
      TDBO(I,2)=TDA
      UBO(I,JE2)=UU
      VBO(I,JE2)=VV
      PRBO(I,JE2)=PM
      TKBO(I,JE2)=TKA
      TDBO(I,JE2)=TDA
  899 CONTINUE

  999 CONTINUE

      CLOSE(6)
      CLOSE(7)
      CLOSE(5)
      CLOSE(15)
      CLOSE(16)
      CLOSE(17)
      CLOSE(18)
      CLOSE(19)
      CLOSE(20)

C---  END OF PROGRAM
```

```
C    ************************************************
C    *    FORMAT
C    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

1000  FORMAT(2X,'RE=',E10.5,' DT=',F10.5,' IE=',I3,' JE=',I3,
     $' ITER=',I3,' II1=',I3,' II=',I3,/,'NB1=',I3,' NB3=',I3,
     $' ISCALE=',I3,' INITIAL=',I3,' EPST=',F10.5)
1100   FORMAT(/3X,'EPSU=',F10.5,'  I=',I5,' J=',I5,5X,'IT=',I5
1200   FORMAT(3X,'EPSV=',F14.5,'  I=',I5,'J=',I5)
1300   FORMAT(3X,'EPSP=',F10.5,'  I=',I5,'J=',I5)
1400   FORMAT(3X,'EPTK=',F10.5,'  I=',I5,'J=',I5)
1500   FORMAT(3X,'EPTD=',F10.5,'  I=',I5,'J=',I5)
1600   FORMAT(3X,'DT=  ',F10.5,'  I=',I5,'J=',I5)
2000  FORMAT(10X,'NO. OF ITERATION=',I3,4X,'STATION=',I3,
     $        4X, 'X=', F10.5)
2100  FORMAT(5X,'U VEL', 8X,'V VEL',8X,'PRESSURE',5X,'TK',
     $     8X,'TD',14X,'EV')
2300  FORMAT(24X, I3)
2400  FORMAT(6E13.4)
2500  FORMAT('START POINT OF TURBULENT FLOW ---',I5)

 500  RETURN
      END
C************************************************
C
C..... BLOCK.MAIN
C
C************************************************
C.... BLOCK.MAIN IS THE COMMON BLOCK USED IN THE 2D
C.... FLAT PLATE PROBLEM
      COMMON/COR1/ X(82,22), Y(82,22)
      COMMON/COR2/ B11(82,22), B12(82,22)
      COMMON/COR3/ B21(82,22), B22(82,22)
      COMMON/COR4/ F1(82,22), F2(82,22), DSJ(82,22)

      COMMON/VEL1/ U(82,22), V(82,22)
      COMMON/VEL2/ US(82,22), VS(82,22)
      COMMON/VEL3/ UH(82,22), VH(82,22)

      COMMON/PRE1/ PR(82,22), PP(82,22)
      COMMON/PRE2/ AN(82,22), AS(82,22), AE(82,22), AW(82,22)
      COMMON/PRE3/ AP(82,22), DS(82,22), DST(82,22)

      COMMON/COE1/ EB(82,22), EC(82,22), EE(82,22), EF(82,22)
      COMMON/COE2/ EH(82,22)
      COMMON/COE3/ SU(82,22), SV(82,22), SK(82,22), SD(82,22)

      COMMON/FAE1/ ZS(82,22), ZN(82,22), ZW(82,22), ZE(82,22)
      COMMON/FAE2/ ZSW(82,22), ZSE(82,22), ZNW(82,22), ZNE(82,
      COMMON/FAE3  ZC(82,22), DU(82,22), DV(82,22)
```

```
      COMMON/CAL1/ RE, DT, RET
      COMMON/CAL2/ JB(82), NB1, NB2, NB3, NNX, NNY

      COMMON/TUB1/ TK(82,22), TD(82,22), EV(82,22)
      COMMON/TUB2/ CK, CD, C1, C2, CNU, AK, E, ISCALE

      COMMON/ADD1/ RU(82,22), RV(82,22), RPR(82,22)
      COMMON/ADD2/ RTK(82,22), RTD(82,22)
      COMMON/STEP/ IT, FT(82,22)
      COMMON/DATA/ THAT(82), UVOS(82,22), UTA(82)

C    ***********************************************************
C    * SUBROUTINE BFC IS TO GENERATE THE BODY-FITTED           *
C    * COORDINATE SYSTEM ON FLAT PLATE FOR TURBULENT FLOW      *
C    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

      SUBROUTINE BFC
      IMPLICIT REAL*8 (A-H,O-Z)

$INSERT BLOCK.MAIN
      COMMON/COEF4/ UI, VI, NX1,NX2,NX3
      REAL*8 AX(99), BY(99)
      REAL*8 AA(99),BB(99),CC(99),DD(99),A(99),
     $         DX(99),DY(99),XXI(99),YET(99)

      ABCD=3.0D0
      PRINT*,'*** BFC ***'
      IMAX=82
      JMAX=22
      NX1=19
      NX2=55
      NX3=22
      DX(NX1)=0.0
      DX(NX2)=1.0
      DX(NX3)=1.0
      DY(2)=0.D0
      DY(JMAX)=ABCD
      A1=-0.3
      A2=-0.12
      B=0.2835
      IMAM=IMAX-1
      JMAM=JMAX-1
      PI=3.141592653589793D0
      EPS=1.D-12

C.... Y-DIRECTION

      EBG=DEXP(B)
      EBR=1.D0/EBG
      PSN=EBG+EBR
```

```
      EB2=EBG*EBG
      EB2R=1.D0/EB2
      PPSN=2.D0*B/(EB2-EB2R)

      DO 10 J=3,JMAM
      AA(J)=-EBG
      BB(J)=PSN
      CC(J)=-EBR
   10 DD(J)=0.D0

      DD(3)=DD(3)-AA(3)*DY(2)
      DD(JMAM)=DD(JMAM)-CC(JMAM)*DY(JMAX)

      CALL TRIDAG(3,JMAM,AA,BB,CC,DD,DY)

      DY(1)=-DY(3)

C.... YET

      DO 40 J=2,JMAM
   40 YET(J)=PPSN*(DY(J+1)-DY(J-1))

      YET(2)=.5D0*(DY(3)-DY(1))
      YET(1)=YET(3)
      YET(JMAX)=YET(JMAM)*YET(JMAM)/YET(JMAM-1)

      DO 45 J=1,JMAX
      BY(J)=PPSN
      DO 45 I=1,IMAX
      Y(I,J)=DY(J)
   45 CONTINUE

C.... X-DIRECTION
C
      AX1=NX1-1.
      AX2=2.*NX1-1.
      AX3=NX2-AX2
C
      DO 50 I=1,IMAX
      Z1=(I-1)/AX1
      Z2=(I-AX2)/AX3
C
      IF(Z1 .LE. 0.5) THEN
      A(I)=A1
      ELSE IF(Z1 .GT. 0.5 .AND. Z1 .LE. 2.)THEN
      PIZ=PI*Z1
      A(I)=A1*DSIN(PIZ)
      ELSE IF(Z2 .LE. 1.5) THEN
      PIZ=PI*Z2
      A(I)=A2*DSIN(PIZ)
      ELSE IF(Z2 .GT. 1.5) THEN
```

```
      A(I)=-A2
      END IF
C
   50 CONTINUE
C
      DO 60 I=2,IMAM
      AA(I)=-DEXP(A(I))
      CC(I)=1./AA(I)
      BB(I)=-(AA(I)+CC(I))
   60 DD(I)=0.D0
C
      N1=NX1+1
      N2=NX2-1
C
      DD(N1)=DD(N1)-AA(N1)*DX(NX1)
      DD(N2)=DD(N2)-CC(N2)*DX(NX2)
C
      CALL TRIDAG(N1,N2.AA,BB,CC,DD,DX)
C
      DO 72 I=NX1,2,-1
      DX(I-1)=-(DX(I)*BB(I)+DX(I+1)*CC(I))/AA(I)
   72 CONTINUE
C
      DO 74 I=NX2,IMAM
      DX(I+1)=-(DX(I)*BB(I)+DX(I-1)*AA(I))/CC(I)
   74 CONTINUE
C
C.... XXI

      DO 80 I=2,IMAM
      IF(DABS(A(I)) .LT. EPS) THEN
      EA=.5D0
      ELSE
      EA2=AA(I)*AA(I)
      EA2R=1./EA2
      EA=2.*A(I)/(EA2-EA2R)
      END IF
      AX(I)=EA
      XXI(I)=EA*(DX(I+1)-DX(I-1))
   80 CONTINUE

      XXI(1)=XXI(2)*XXI(2)/XXI(3)
      XXI(IMAX)=XXI(IMAM)*XXI(IMAM)/XXI(IMAM-1)
      AX(1)=AX(2)
      AX(IMAX)=AX(IMAM)

      DO 85 I=1,IMAX
      II=I
      DO 85 J=1,JMAX
      X(I,J)=DX(II)
   85 CONTINUE
```

```
      DO 90 I=1,IMAX
      DO 90 J=2,JMAX
      DSJ(I,J)=XXI(I)*YET(J)
      B11(I,J)=YET(J)
      B12(I,J)=0.0
      B21(I,J)=0.0
      B22(I,J)=XXI(I)
      F1(I,J)=-2.*A(I)/XXI(I)/XXI(I)
      F2(I,J)=-2.*B/YET(J)/YET(J)
 90   CONTINUE
      DO 106 I=1,NNX+1
      DO 106 J=1,NNY+1
      K=I
      X(I,J)=X(K,J)
      Y(I,J)=Y(K,J)
      B11(I,J)=B11(K,J)
      B12(I,J)=B12(K,J)
      B21(I,J)=B21(K,J)
      B22(I,J)=B22(K,J)
      F1(I,J)=F1(K,J)
      F2(I,J)=F2(K,J)
      AX(I)=AX(K)
      DSJ(I,J)=DSJ(K,J)
106   CONTINUE

      OPEN(30,FILE='OUTFP')
      WRITE(30,2400) (X(I,2),I=1,NNX+1)
      WRITE(30,2400) (Y(1,J),J=2,NNY+1)
 2400 FORMAT(6E13.4)
      CLOSE(30)

      RETURN
      END

C    ***********************************************************
C    * SUBROUTINE EQCOE IS TO CALCULATE THE COEFFICIENTS
C    * OF GOVERNING EQUATIONS
C    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

      SUBROUTINE EQCOE(M)
      IMPLICIT REAL*8 (A-H,O-Z)

$INSERT BLOCK.MAIN

      IF(M .EQ. 2) THEN

C.... Y MOMENTUM EQUATION

      DO 100 I=2,NNX
      IP1=I+1
```

```
      IM1=I-1
      DO 100 J=JB(I), NNY
      JP1=J+1
      JM1=J-1

      V11 = (B11(I,J)/DSJ(I,J)+B11(IM1,J)/DSJ(IM1,J))*0.5
      V12 = (B12(I,J)/DSJ(I,J)+B12(IM1,J)/DSJ(IM1,J))*0.5
      V22 = (B22(I,J)/DSJ(I,J)+B22(IM1,J)/DSJ(IM1,J))*0.5
      V21 = (B21(I,J)/DSJ(I,J)+B21(IM1,J)/DSJ(IM1,J))*0.5
      EVV=0.5*(EV(I,J)+EV(IM1,J))
      REV= 1.0/(1.0/RE+EVV)
      EVDY=0.25*(EV(IM1,JP1)-EV(IM1,JM1)+EV(I,JP1)-EV(I,JM1))
      EVDX=EV(I,J)-EV(IM1,J)
      EVDX1=(EVDX*V11+EVDY*V21)
      EVDY1=(EVDX*V12+EVDY*V22)

      UV=0.25*(U(I,JP1)+U(I,J)+U(IM1,JP1)+U(IM1,J))
      EB(I,J)=REV*(V11*(UV-EVDX1)+V12*(V(I,J)-2.*EVDY1))
     $-(F1(I,J)+F1(IM1,J))*0.5
      EC(I,J)=REV*(V21*(UV-EVDX1)+V22*(V(I,J)-2.*EVDY1))
     $-(F2(I,J)+F2(IM1,J))*0.5
      EE(I,J)=(V11*V11+V12*V12)
      EF(I,J)=(V21*V21+V22*V22)
      EH(I,J)=REV/DT

      TKDY=0.25*(TK(IM1,JP1)-TK(IM1,JM1)+TK(I,JP1)-TK(I,JM1))
      TKDX=TK(I,J)-TK(IM1,J)
      TKDY1=2./3.*(TKDX*V12+TKDY*V22)
      UDX=0.5*(U(I,JP1)-U(IM1,JP1)+U(I,J)-U(IM1,J))
      UDY=0.5*(U(IM1,JP1)-U(IM1,J)+U(I,JP1)-U(I,J))
      UDY1=UDX*V12+UDY*V22
      VDXDY=0.25*(V(IP1,JP1)-V(IP1,JM1)+V(IM1,JM1)-V(IM1,JP1))
      SOR=2.*(V11*V21+V12*V22)*VDXDY
      PRI=0.25*(PR(IP1,JP1)-PR(IM1,JP1)+PR(IP1,J)-PR(IM1,J))
      PRJ=PR(I,JP1)-PR(I,J)
      SV(I,J)=SOR+REV*(-PRI*V12-PRJ*V22-TKDY1+EVDX1*UDY1)
     &+REV/DT*V(I,J)

 100  CONTINUE

      ELSE IF( M .EQ. 1 ) THEN

C...  X MOMENTUM EQUATION

      DO 200 I=2,NNX
      IP1=I+1
      IM1=I-1
      DO 200 J=JB(I), NNY
        JP1=J+1
        JM1=J-1
```

```
      U11=(B11(I,J)/DSJ(I,J)+B11(I,JM1)/DSJ(I,JM1))*0.5
      U12=(B12(I,J)/DSJ(I,J)+B12(I,JM1)/DSJ(I,JM1))*0.5
      U22=(B22(I,J)/DSJ(I,J)+B22(I,JM1)/DSJ(I,JM1))*0.5
      U21=(B21(I,J)/DSJ(I,J)+B21(I,JM1)/DSJ(I,JM1))*0.5

      EVV=0.5*(EV(I,J)+EV(I,JM1))
      REU=1.0/(1./RE+EVV)
      EVDY=EV(I,J)-EV(I,JM1)
      EVDX=0.25*(EV(IP1,J)-EV(IM1,J)+EV(IP1,JM1)-EV(IM1,JM1))
      EVDX1=(EVDX*U11+EVDY*U21)
      EVDY1=(EVDX*U12+EVDY*U22)

      VU=0.25*(V(I,J)+V(I,JM1)+V(IP1,J)+V(IP1,JM1))
      EB(I,J)=REU*(U11*(U(I,J)-2.*EVDX1)+U12*(VU-EVDY1))
     $-(F1(I,J)+F1(I,JM1))*0.5
      EC(I,J)=REU*(U21*(U(I,J)-2.*EVDX1)+U22*(VU-EVDY1))
     $-(F2(I,J)+F2(I,JM1))*0.5
      EE(I,J)=(U11*U11+U12*U12)
      EF(I,J)=(U21*U21+U22*U22)
      EH(I,J)=REU/DT

      TKDY=TK(I,J)-TK(I,JM1)
      TKDX=0.25*(TK(IP1,J)-TK(IM1,J)+TK(IP1,JM1)-TK(IM1,JM1))
      TKDX1=2./3.*(TKDX*U11+TKDY*U21)
      VDX=0.5*(V(IP1,J)-V(I,J)+V(IP1,JM1)-V(I,JM1))
      VDY=0.5*(V(I,J)-V(I,JM1)+V(IP1,J)-V(IP1,JM1))
      VDX1=VDX*U11+VDY*U21
      UDXDY=0.25*(U(IP1,JP1)-U(IP1,JM1)+U(IM1,JM1)-U(IM1,JP1))
      PRJ=0.25*(PR(IP1,JP1)-PR(IP1,JM1)+PR(I,JP1)-PR(I,JM1))
      IF(J.EQ.3) PRJ=0.5*(PR(IP1,4)+PR(I,4)-PR(IP1,3)-PR(I,3))
      PRI=PR(IP1,J)-PR(I,J)
      IF(J .EQ. 3) THEN
      UDXDY=0.25*(U(IP1,JP1)-U(IP1,J)+U(IM1,J` U(IM1,JP1))
      PRJ=  0.25*(PR(I,JP1)-PR(I,J)+PR(IP1,JP1)-PR(IP1,J))
      END IF
      SOR=2.*(U11*U21+U12*U22)*UDXDY
      SU(I,J)=SOR+REU*(-PRJ*U21-PRI*U11-TKDX1+EVDY1*VDX1)
     &+REU/DT*U(I,J)

  200 CONTINUE

      ELSE IF( M .EQ. 4 .OR. M .EQ. 5 ) THEN

C...  TK AND TD EQUATION

      DO 300 I=NB2,NNX
      IP1=I+1
      IM1=I-1
      DO 300 J=JB(I),NNY
      JP1=J+1
      JM1=J-1
```

```
T11=B11(I,J)/DSJ(I,J)
T12=B12(I,J)/DSJ(I,J)
T22=B22(I,J)/DSJ(I,J)
T21=B21(I,J)/DSJ(I,J)

EVDX=0.5*(EV(IP1,J)-EV(IM1,J))
EVDY=0.5*(EV(I,JP1)-EV(I,JM1))
EVDX1=(EVDX*T11+EVDY*T21)
EVDY1=(EVDX*T21+EVDY*T22)

UDY=U(I,JP1)-U(I,J)
UDX=0.25*(U(IP1,JP1)-U(IM1,JP1)+U(IP1,J)-U(IM1,J))
UDX1=UDX*T11+UDY*T21
UDY1=UDX*T12+UDY*T22
VDY=0.25*(V(I,JP1)-V(I,JM1)+V(IP1,JP1)-V(IP1,JM1))
VDX=V(IP1,J)-V(I,J)
VDX1=VDX*T11+VDY*T21
VDY1=VDX*T12+VDY*T22
GG=EV(I,J)*(2.*(UDX1*UDX1+VDY1*VDY1)+(UDY1+VDX1)*(UDY1+V

 IF(M .EQ. 5) THEN
   SG=CD
 ELSE IF(M .EQ. 4) THEN
   SG=CK
 END IF
   RED=1./(1./RE+EV(I,J)/SG)

 UC=0.5*(U(I,JP1)+U(I,J))
 VC=0.5*(V(I,J)+V(IP1,J))
 EB(I,J)=RED*((UC-EVDX1/SG)*T11+(VC-EVDY1/SG)*T12)-F1(I,J
 EC(I,J)=RED*((UC-EVDX1/SG)*T21+(VC-EVDY1/SG)*T22)-F2(I,J
 EE(I,J)=T11*T11+T12*T12
 EF(I,J)=T21*T21+T22*T22

 IF(M .EQ.4) THEN
 TKDXY=0.25*(TK(IP1,JP1)-TK(IM1,JP1)+TK(IM1,JM1)-TK(IP1,J
 SOR=2.*(T11*T21+T12*T22)*TKDXY
 SK(I,J)=SOR+RED*(GG)+RED/DT*TK(I,J)
 EH(I,J)=TD(I,J)*RED/TK(I,J)+RED/DT
 ELSE IF( M. EQ.5) THEN
 TDDXY=0.25*(TD(IP1,JP1)-TD(IM1,JP1)+TD(IM1,JM1)-TD(IP1,J
 SOR=2.*(T11*T21+T12*T22)*TDDXY
 IF (ISCALE.EQ.1) THEN
 TSCAL=TD(I,J)/TK(I,J)
 ELSE IF(ISCALE.EQ.2) THEN
 TSCAL=DSQRT(TD(I,J))
 ELSE
 PRINT*, 'ERROR IN SELECTION OF TURBULENT SCALE!!'
 END IF
 SD(I,J)=SOR+C1*RED*GG*TSCAL+RED/DT*TD(I,J)
```

```
      EH(I,J)=C2*RED*TSCAL+RED/DT
      END IF

 300  CONTINUE

      END IF

      RETURN
      END

C     ******************************************************************
C     * SUBROUTINE HVEL IS TO CALCULATE THE PSEUDOVELOCITY
C     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

      SUBROUTINE HVEL
      IMPLICIT REAL*8 (A-H,O-Z)

$INSERT BLOCK.MAIN

       PRINT*,'*** HVEL ***'

      IE  = NNX+1
      JE  = NNY+1

C.... CORRECT THE VELOCITY BY PRESSURE CORRECTION

      DO 10 I=2,NNX
      IP1=I+1
      IM1=I-1
      DO 10 J=JB(I),NNY
      JP1=J+1
      JM1=J-1
      V(I,J)=VS(I,J)-DV(I,J)*(PP(I,JP1)-PP(I,J))
      U(I,J)=US(I,J)-DU(I,J)*(PP(IP1,J)-PP(I,J))
  10  CONTINUE

      CALL WALLFN
      CALL CHECK(U,1,IE,JE)
      CALL CHECK(V,2,IE,JE)

C.... PSEUDOVELOCITY OF V

      CALL EQCOE (2)
      CALL COEF(2)
      DO 150 I=2,NNX
        IP1=I+1
        IM1=I-1
      DO 150 J=JB(I),NNY
        JP1=J+1
        JM1=J-1
```

```
      REV=EH(I,J)*DT
      V22=(B22(I,J)/DSJ(I,J)+B22(IM1,J)/DSJ(IM1,J))/2.
      VHS=SV(I,J)+REV*V22*(PR(I,JP1)-PR(I,J))
      DV(I,J) = ZC(I,J) * REV
      VH(I,J)=V(IP1,J)*ZE(I,J)+V(IP1,JP1)*ZNE(I,J)
     &   +V(IP1,JM1)*ZSE(I,J)+V(I,JM1)*ZS(I,J)
     &   +V(IM1,JM1)*ZSW(I,J)+V(IM1,J)*ZW(I,J)
     &   +V(IM1,JP1)*ZNW(I,J)+V(I,JP1)*ZN(I,J)+ZC(I,J)*VHS

150   CONTINUE

C..... PSEUDOVELOCITY OF U

      CALL EQCOE (1)
      CALL COEF(1)

      DO 100 I = 2, NNX
         IP1=I+1
         IM1=I-1
      DO 100 J = JB(I), NNY
         JM1=J-1
         JP1=J+1

      U11=(B11(I,J)/DSJ(I,J)+B11(I,JM1)/DSJ(I,JM1))/2.
      REU=EH(I,J)*DT
      UHS=SU(I,J)+REU*U11*(PR(IP1,J)-PR(I,J))
      DU(I,J) = ZC (I,J) * REU
      UH(I,J)=U(IP1,J)*ZE(I,J)+U(IP1,JP1)*ZNE(I,J)
     &   +U(IP1,JM1)*ZSE(I,J)+U(I,JM1)*ZS(I,J)
     &   +U(IM1,JM1)*ZSW(I,J)+U(IM1,J)*ZW(I,J)
     &   +U(IM1,JP1)*ZNW(I,J)+U(I,JP1)*ZN(I,J)+ZC(I,J)*UHS

 100  CONTINUE

C... SET THE BOUNDARY DATA

      DO 200 I = 1, IE
         VH(I,JE ) = V(I,JE)
         UH(I,JE ) = U(I,JE)
         J=JB(I)-1
         UH(I,J)=U(I,J)
         VH(I,J)=V(I,J)
 200  CONTINUE

C...  ALONG THE I=1 -- THE INLET LINE

      DO 300 J = 1, JE
         VH(1,J) = V(1,J)
         UH(1,J) = U(1,J)
         UH(IE ,J)=U(IE ,J)
         VH(IE ,J)=V(IE ,J)
```

```
300   CONTINUE

      RETURN
      END

C     ************************************************************
C     * SUBROUTINE SVEL IS TO SOLVE THE VELOCITY
C     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

      SUBROUTINE SVEL
      IMPLICIT REAL*8 (A-H, O-Z)

$INSERT BLOCK.MAIN

      PRINT*,'*** SVEL ***'

      IE=NNX+1
      JE=NNY+1
      DO 20 I=1,NNX+1
      DO 20 J=2,NNY+1
      US(I,J)=U(I,J)
      VS(I,J)=V(I,J)
 20   CONTINUE

C.... U VELOCITY

      CALL EQCOE( 1 )
      CALL COEF(1)

      DO 100 I =2 ,NNX
      DO 100 J = JB(I), NNY
      FT(I,J)=ZC(I,J)*SU(I,J)
      DU(I,J)=ZC(I,J)*EH(I,J)*DT
 100  CONTINUE

      CALL SOLVE(US,FT,1,2,IE,JE)
      CALL CHECK(US,1,IE,JE)

C.... V VELOCITY

      CALL EQCOE( 2 )
      CALL COEF(2)

      DO 150 I=2,NNX
      DO 150 J=JB(I),NNY
      FT(I,J)=ZC(I,J)*SV(I,J)
      DV(I,J) = ZC(I,J) * EH(I,J)*DT
 150  CONTINUE

      CALL SOLVE(VS,FT,2,2,IE,JE)
      CALL CHECK(VS,2,IE,JE)
```

```
      RETURN
      END

C    ***********************************************************
C    * SUBROUTINE PRESS IS USED TO SOLVE 1. PRESSURE
C    *                                   2. PRESSURE CORECTION
C    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

      SUBROUTINE PRESS (NC)
      IMPLICIT REAL*8 (A-H,O-Z)

$INSERT BLOCK.MAIN
      DIMENSION AA(99),BB(99),CC(99),DD(99),T(99)

      PRINT*, ' ::: PRESS :::  NC = ',NC

      IE = NNX+1
      JE = NNY+1
      JMN=JB(NB3+1)-1

      DO 20 J=2,JE
      DU(IE,J)=DU(NNX,J)
      DV(IE,J)=DV(NNX,J)
      DU(1,J)=DU(2,J)
      DV(1,J)=DV(2,J)
   20 CONTINUE

      DO 30 I=1,IE
      J=JB(I)-1
      JP1=J+1
      DU(I,J)=DU(I,JP1)
      DV(I,J)=DV(I,JP1)
      DU(I,JE)=DU(I,NNY)
      DV(I,JE)=DV(I,NNY)
   30 CONTINUE

      DO 210 I=2, NNX
      DO 210 J=JB(I), NNY
      IM1=I-1
      JM1=J-1
      P11=(B11(I,J)/DSJ(I,J)+B11(I,JM1)/DSJ(I,JM1))/2.
      DU(I,J)=P11*DU(I,J)
      P22=(B22(I,J)/DSJ(I,J)+B22(IM1,J)/DSJ(IM1,J))/2.
      DV(I,J)=P22*DV(I,J)
  210 CONTINUE

      DO 100 I=2,NNX
      IP1=I+1
      IM1=I-1
      DO 100 J=JB(I) ,NNY
```

```
      JP1=J+1
      JM1=J-1
      AE(I,J)=(Y(I,J)-Y(I,JM1))*DU(I,J)
      AW(I,J)=(Y(IM1,J)-Y(IM1,JM1))*DU(IM1,J)
      AN(I,J)=(X(I,J)-X(IM1,J))*DV(I,J)
      AS(I,J)=(X(I,JM1)-X(IM1,JM1))*DV(I,JM1)
      AP(I,J)=AE(I,J)+AW(I,J)+AN(I,J)+AS(I,J)
  100 CONTINUE

      DO 175 J=3,NNY
      AP(NNX,J)=AP(NNX,J)-AE(NNX,J)
      AP(2,J)=AP(2,J)-AW(2,J)
      AE(NNX,J)=0.
      AW(2,J)=0.
  175 CONTINUE

      DO 117 I=2,NNX
      J=JB(I)
      AP(I,J)=AP(I,J)-AS(I,J)
      AS(I,J)=0.0
  117 CONTINUE

      DO 234 J=3,JMN
      AP(NB3+2,J)=AP(NB3+2,J)-AW(NB3+2,J)
      AW(NB3+2,J)=0.0
  234 CONTINUE
C.... FORM THE SOURCE TERM OF PRESSURE CORRECTION EQUATION
      IF(NC .EQ. 2) THEN
      DO 343 J=3,JMN
      VS(NB3+1,J)=V(NB3+1,J)
      US(NB3+1,J)=U(NB3+1,J)
  343 CONTINUE

      DO 52 I=2,NNX
      IP1=I+1
      IM1=I '
      DO  ` J=JB(I), N
      JP1=J+1
      JM1=J-1

      DS(I,J)=(Y(I,J)-Y(I,JM1))*US(I,J)
     $    -(Y(IM1,J)-Y(IM1,JM1))*US(IM1,J)
     $    +(X(I,J)-X(IM1,J))*VS(I,J)
     $    -(X(I,JM1)-X(IM1,JM1))*VS(I,JM1)
      DST(I,J)=DS(I,J)
   52 CONTINUE

      DO 62 I=1,IE
      DO 62 J=1,JE
      FT(I,J)=0.0
   62 CONTINUE
```

```
      ELSE IF(NC .EQ. 1) THEN

      DO 456 J=3,JMN
      UH(NB3+1,J)=U(NB3+1,J)
      VH(NB3+1,J)=V(NB3+1,J)
  456 CONTINUE
      DO 150 I=2,NNX
      IP1=I+1
      IM1=I-1
      DO 150 J=JB(I), NNY
      JP1=J+1
      JM1=J-1

      DS(I,J)=(Y(I,J)-Y(I,JM1))*UH(I,J)
     $   -(Y(IM1,J)-Y(IM1,JM1))*UH(IM1,J)
     $   +(X(I,J)-X(IM1,J))*VH(I,J)
     $   -(X(I,JM1)-X(IM1,JM1))*VH(I,JM1)
  150 CONTINUE

      DO 270 J=1,JE
      DO 270 I=1,IE
      FT(I,J)=PR(I,J)
  270 CONTINUE

      END IF

C.... SOLVE THE EQUATION DOMAIN BY USING TRIDIAGONAL METHOD

      ITP=50
      FAC=0.1
      EPS=1.D-7

      DO 400 IP=1,ITP
      SOR=0.

      DO 300 I=2,NNX
      IF(I.EQ.NB2) GO TO 300
      IP1=I+1
      IM1=I-1
      JJ=JB(I)
      DO 320 J=JJ,NNY
      JP1=J+1
      JM1=J-1
      AA(J)=-AS(I,J)
      BB(J)=AP(I,J)
      CC(J)=-AN(I,J)
      DD(J)=AE(I,J)*FT(IP1,J)+AW(I,J)*FT(IM1,J)-DS(I,J)
  320 CONTINUE
```

```
      DD(JJ)=DD(JJ)-AA(JJ)*FT(I,JJ-1)
      DD(NNY)=DD(NNY)-CC(NNY)*FT(I,JE)
      CALL TRIDAG(JJ,NNY,AA,BB,CC,DD,T)

      DO 340 J=JJ,NNY
      ST =FT(I,J)-T(J)
      IF(DABS(SOR) .LT. DABS(ST)) SOR=ST
      FT(I,J)=T(J)
  340 CONTINUE

  300 CONTINUE
      DO 310 J=1,NNY
C     IF(NC.EQ.1)FT(NB3+2,J)=0.5*(FT(NB3+1,J)+FT(NB3+3,J))
  310 FT(NB2,J)=0.5*(FT(NB2-1,J)+FT(NB2+1,J))

      IF(DABS(SOR) .LT. EPS) GO TO 345

  400 CONTINUE

  345 WRITE(6,900) NC, IP, SOR
      CALL CHECK(FT,3,IE,JE)


      IF(NC .EQ. 1) THEN
      DO 500 I=1,IE
      DO 500 J=2,JE
      PR(I,J)=(1.-FAC) *PR(I,J)+ FAC*FT(I,J)
  500 CONTINUE

      ELSE IF(NC .EQ. 2) THEN
      DO 550 I=1,IE
      DO 550 J=2,JE
      PP(I,J)=FT(I,J)
  550 CONTINUE
      END IF

  900 FORMAT(2I10,E12.4)

      RETURN
      END

C     **********************************************************
C     * SUBROUTINE COEF IS USED TO CALCULATE THE FA COEFFICIENTS
C     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

      SUBROUTINE COEF( NC )
      IMPLICIT REAL*8(A-H,O-Z)

$INSERT BLOCK.MAIN
      DIMENSION CF(3,3)
```

```
      PI=3.141592653589793D0
      MAX=6
      EPE=1.D-5
      C1W=1.D0
      EMAX=20.D0

      IJ1=2
      IF(NC .GE. 4) IJ1=NB2
      DO 200 I=IJ1,NNX
      DO 200 J=JB(I),NNY

      AR=EB(I,J)/2.D0
      BR=EC(I,J)/2.D0
      ER=DSQRT(EE(I,J))
      FR=DSQRT(EF(I,J))
      IF ( FR .LT. 1.D-23) PRINT*, 'ERROR IN COEF, FR=0.  '

      HX=1./ER
      HY=1./FR
      AR=AR/ER
      BR=BR/FR

      IF(DABS(AR).LT.EPE)AR=DSIGN(EPE,AR)
      IF(DABS(BR).LT.EPE)BR=DSIGN(EPE,BR)

C     CHECK THE SIZES OF THE GRIDS IF IT AGREES WITH THE ASSUM
C     DIRECTIONS IN THE DERIVATION, AND IF IT DOES NOT CHANGE
C     SEE PAGE 53. OF DR. H.C CHEN DISSERTION.

      ER2=ER*ER
      FR2=FR*FR
      AB2=AR*AR+BR*BR
      AH=AR*HX
      AKW=AR*HY
      BH=BR*HX
      BK=BR*HY
      DAH=DABS(AH)
      DBK=DABS(BK)
      AH2=AH*AH
      BK2=BK*BK

      IM=0
      JM=0
      IF(DAH.GT.EMAX) IM=1
      IF(DBK.GT.EMAX) JM=2
      M=IM+JM+1
      GO TO (1,2,3,4), M

    1 EPAH=DEXP(AH)
      EPBK=DEXP(BK)
      EPAHI=1./EPAH
```

```
      EPBKI=1./EPBK
      COSHA=0.5*(EPAH+EPAHI)
      COSHB=0.5*(EPBK+EPBKI)
      COTHA=2.*COSHA/(EPAH-EPAHI)
      COTHB=2.*COSHB/(EPBK-EPBKI)
      AKCTHA=AKW*COTHA
      BHCTHB=BH*COTHB
      PWR=1.
      IF(HX .GT. HY) GO TO 11
      EX2=0.
      DO 10 II=1,MAX
      ZA=(II-0.5)*PI
      ZA2=ZA*ZA
      PWR=-PWR
      DABK=DSQRT(AB2+ZA2*ER2)*HY
      IF(DABK .GT. 100.) GO TO 9
      AB=DEXP(DABK)
   10 EX2=EX2-PWR*ZA/((AB+1./AB)*(AH2+ZA2)*(AH2+ZA2))
    9 PA=8.*AH*COTHA*COSHA*COSHB*EX2
      PB=1.+BHCTHB/AKCTHA*(PA-1.)
      CF(2,2)=0.5*HX/(AR*COTHA)*(1.-PA)
      GO TO 100

   11 EY2=0.
      DO 12 II=1,MAX
      ZA=(II-0.5)*PI
      ZA2=ZA*ZA
      PWR=-PWR
      DABH=DSQRT(AB2+ZA2*FR2)*HX
      IF(DABH.GT.100.) GO TO 19
      AB=DEXP(DABH)
   12 EY2=EY2-PWR*ZA/((AB+1./AB)*(BK2+ZA2)*(BK2+ZA2))
   19 PB=8.*BK*COTHB*COSHA*COSHB*EY2
      PA=1.+AKCTHA/BHCTHB*(PB-1.)
      CF(2,2)=0.5*HY/(BR*COTHB)*(1.-PB)
      GO TO 100

    2 EPBK=DEXP(BK)
      EPBKI=1./EPBK
      COSHB=0.5*(EPBK+EPBKI)
      COTHB=2.*COSHB/(EPBK-EPBKI)
      COTHA=DSIGN(C1W,AR)
      AKCTHA=AKW*COTHA
      BHCTHB=BH*COTHB
      PWR=1.
      IF(AKCTHA.LT.BHCTHB) GO TO 22
      EX2=0.
      FX2=0.
      DO 20 II=1,MAX
      ZA=(II-0.5)*PI
      ZA2=ZA*ZA
```

```
      PWR=-PWR
      PZ=PWR*ZA/((AH2+ZA2)*(AH2+ZA2))
      FX2=FX2-PZ
      DABK=DSQRT(AB2+ZA2*ER2)*HY
      AB=1.
      IF(DABK.GT.100.) GO TO 20
      EPABK=DEXP(DABK)
      AB=1.-COSHB/(EPABK+1./EPABK)
   20 EX2=EX2-PZ*AB
      PA=1.-EX2/FX2
      PB=1.+BHCTHB/AKCTHA*(PA-1.)
      CF(2,2)=0.5*HY/(BR*COTHB)*(1.-PB)
      GO TO 100
   22 EY2=0.
      DO 23 II=1,MAX
      ZA=(II-0.5)*PI
      ZA2=ZA*ZA
      PWR=-PWR
      DABH=DAH-DSQRT(AB2+ZA2*FR2)*HX
      IF(DABS(DABH).GT.100.) GO TO 29
      AB=DEXP(DABH)
   23 EY2=EY2-PWR*ZA*AB/((BK2+ZA2)*(BK2+ZA2))
   29 PB=4.*BK*COTHB*COSHB*EY2
      PA=1.+AKCTHA/BHCTHB*(PB-1.)
      CF(2,2)=0.5*HY/(BR*COTHB)*(1.-PB)
      GO TO 100

    3 EPAH=DEXP(AH)
      EPAHI=1./EPAH
      COSHA=0.5*(EPAH+EPAHI)
      COTHA=2.*COSHA/(EPAH-EPAHI)
      COTHB=DSIGN(C1W,BR)
      AKCTHA=AKW*COTHA
      BHCTHB=BH*COTHB
      PWR=1.
      IF(AKCTHA.GT.BHCTHB) GO TO 32
      EY2=0.
      FY2=0.
      DO 30 II=1,MAX
      ZA=(II-0.5)*PI
      ZA2=ZA*ZA
      PWR=-PWR
      PZ=PWR*ZA/((BK2+ZA2)*(BK2+ZA2))
      FY2=FY2-PZ
      DABH=DSQRT(AB2+ZA2*FR2)*HX
      AB=1.
      IF(DABH.GT.100.) GO TO 30
      EPABH=DEXP(DABH)
      AB=1.-COSHA/(EPABH+1./EPABH)
   30 EY2=EY2-PZ*AB
      PB=1.-EY2/FY2
```

```
      PA=1.+AKCTHA/BHCTHB*(PB-1.)
      CF(2,2)=0.5*HY/(BR*COTHB)*(1.-PB)
      GO TO 100
   32 EX2=0.
      DO 33 II=1,MAX
      ZA=(II-0.5)*PI
      ZA2=ZA*ZA
      PWR=-PWR
      DABK=DBK-DSQRT(AB2+ZA2*ER2)*HY
      IF(DABS(DABK).GT.100.) GO TO 39
      AB=DEXP(DABK)
   33 EX2=EX2-PWR*ZA*AB/((AH2+ZA2)*(AH2+ZA2))
   39 PA=4.*AH*COTHA*COSHA*EX2
      PB=1.+BHCTHB/AKCTHA*(PA-1.)
      CF(2,2)=0.5*HY/(BR*COTHB)*(1.-PB)
      GO TO 100

    4 DAK=DABS(AKW)
      DBH=DABS(BH)
      COTHA=DSIGN(C1W,AR)
      COTHB=DSIGN(C1W,BR)
      IF(DAK.LT.DBH) GO TO 41
      PA=0.
      PB=1.-DBH/DAK
      CF(2,2)=0.5*HX/(AR*COTHA)
      GO TO 100

   41 PB=0.
      PA=1.-DAK/DBH
      CF(2,2)=0.5*HY/(BR*COTHB)

  100 Q=1.-PA-PB
      TANHA=1./COTHA
      TANHB=1./CO[illegible]
      BE=0.5*(1.-TANHA)
      BW=0.5*(1.+TANHA)
      BN=0.5*(1.-TANHB)
      BS=0.5*(1.+TANHB)
      CF(1,1)=BW*BS*Q
      CF(3,1)=BE*BS*Q
      CF(1,3)=BW*BN*Q
      CF(3,3)=BE*BN*Q
      CF(2,1)=BS*PA
      CF(2,3)=BN*PA
      CF(1,2)=BW*PB
      CF(3,2)=BE*PB
C
      CFC=CF(2,2)

      CFP=1.+CFC*EH(I,J)
      ZC(I,J)=CFC/CFP
```

```
C.... FINAL FA COEFFICIENTS ON TRANSFORMED DOMAIN

      ZS(I,J)=CF(2,1)/CFP
      ZN(I,J)=CF(2,3)/CFP
      ZW(I,J)=CF(1,2)/CFP
      ZE(I,J)=CF(3,2)/CFP
      ZSW(I,J)=CF(1,1)/CFP
      ZSE(I,J)=CF(3,1)/CFP
      ZNW(I,J)=CF(1,3)/CFP
      ZNE(I,J)=CF(3,3)/CFP

 200  CONTINUE

      RETURN
      END

C   ***************************************************************
C   * SUBROUTINE SOLVE IS USED TO SOLVE  STARED VELOCITY
C   >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

      SUBROUTINE SOLVE(HT,FZ,NC,IS1,IX,IY)
      IMPLICIT REAL*8 (A-H, O-Z)

$INSERT BLOCK.MAIN
      DIMENSION FZ(IX,IY),HT(IX,IY)
      REAL*8 AA(99),BB(99),CC(99),DD(99),T(99)


      IE=NNX+1
      JE=NNY+1
      DO 900 IM=1,10
      EPSR=0.0

      DO 100 I=IS1,NNX
      IP1=I+1
      IM1=I-1
      JJ=JB(I)
      DO 200 J=JJ, NNY
      JP1=J+1
      JM1=J-1

        AA(J)=-ZS(I,J)
        BB(J)=1.
        CC(J)=-ZN(I,J)
        DD(J)=ZE(I,J)*HT(IP1,J)+ZW(I,J)*HT(IM1,J)
     $   +ZSE(I,J)*HT(IP1,JM1)+ZNW(I,J)*HT(IM1,JP1)+ZNE(I,J)
     $   *HT(IP1,JP1)+ZSW(I,J)*HT(IM1,JM1)+FZ(I,J)

  200 CONTINUE
```

```
      DD(JJ)=DD(JJ)-AA(JJ)*HT(I,JJ-1)
      DD(NNY)=DD(NNY)-CC(NNY)*HT(I,JE)
      CALL TRIDAG(JJ,NNY,AA,BB,CC,DD,T)
      IF(I.LT.NB1.OR.I.GT.NB3+1) CALL UPDATE(HT,I,NC,IX,IY)

      DO 50 J=JJ,NNY
      EPS2=DABS(HT(I,J)-T(J))
      IF(EPS2.GT.EPSR) EPSR=EPS2
 50   HT(I,J)=T(J)

 100  CONTINUE
      IF(EPSR.LT.1.0D-7) GO TO 20

 900  CONTINUE

 20   WRITE(6,1000) NC, IM,EPSR
 1000 FORMAT('SOLVE  NC=', I5, 'ITERAT.= ',I10,'  EPSR=',E12.4
      RETURN
      END

C    *************************************************************
C    *   SUBROUTINE CHECK IS TO UPDATE BOUNDARY VALUES
C    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

      SUBROUTINE CHECK(GG,NC,IX,IY)
      IMPLICIT REAL*8 (A-H, O-Z)

$INSERT BLOCK.MAIN
      DIMENSION GG(IX,IY)


      IE=NNX+1
      JE=NNY+1
      DO 100 I=2,NNX
      CALL UPDATE(GG,I,NC,IX,IY)
  100 CONTINUE

      IF(NC .EQ. 1) THEN

      DO 110 J=1,JE
      IF(GG(NNX,J) .LT. GG(NNX-1,J)) THEN
      GG(NNX,J)=GG(NNX-1,J)*1.001
      GG(IE,J) =GG(NNX-1,J)*1.002
      ELSE
      D1=X(IE,J)-X(NNX,J)
      D2=X(NNX,J)-X(NNX-1,J)
      DX1=(D1+D2)/D2
      DX2=D1/D2
      GG(IE,J)=GG(NNX,J)*DX1-GG(NNX-1,J)*DX2
      END IF
      IF(GG(IE,J) .GT.1.0) GG(IE,J)=1.0
```

```
110   CONTINUE

      ELSE IF(NC .EQ. 2) THEN

      DO 210 J=1,JE
      GG(IE,J)=GG(NNX,J)
 210  CONTINUE

      ELSE IF(NC .EQ. 3) THEN

      DO 360 J=1,JE
      GG(IE,J)=GG(NNX,J)
 360  CONTINUE

      ELSE IF(NC .GE. 4 ) THEN

      DO 410 J=1,JE
      GG(IE,J)=GG(NNX,J)
 410  CONTINUE

      END IF

      RETURN
      END

C    *******************************************************
C    *   SUBROUTINE UPDATE IS TO UPDATE BOUNDARY VALUES
C    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

      SUBROUTINE UPDATE(GG,I,NC,IX,IY)
      IMPLICIT REAL*8 (A-H, O-Z)

$INSERT BLOCK.MAIN
      DIMENSION GG(IX,IY)

      IE=NNX+1
      JE=NNY+1
      JJ=JB(I)-1
      JJ1=JJ+1
      JJ2=JJ+2
      IF(NC .EQ. 1) THEN

      IF(I .GE. NB1 .AND. I .LE. NB3) RETURN
      IF(I.EQ.NB3+1) THEN
      D5=(Y(I,JJ2)+Y(I,JJ1))*0.5
      D4=(Y(I,JJ1)+Y(I,JJ))*0.5
      D42=(D4-Y(I,2))**2
      D52=(D5-Y(I,2))**2
C     GG(I,2)=GG(I-1,3)
      DG=GG(I,JJ1)-GG(I,2)
      DO 50 J=3,JJ
```

```
      D3=(Y(I,J)+Y(I,J-1))*0.5
      D32=(D3-Y(I,2))**2
C     GG(I,J)=GG(I,JJ1)-(GG(I,JJ2)-GG(I,JJ1))*(D42-D32)/(D52-D
      GG(I,J)=GG(I,2)+DG*D32/D42
   50 CONTINUE
      END IF
C     D4=(Y(I,4)+Y(I,3))*0.5
C     D3=(Y(I,3)+Y(I,2))*0.5
C     D32=(D3-Y(I,2))**2
C     D42=(D4-Y(I,2))**2
C     GG(I,2)=(GG(I,3)*D42-GG(I,4)*D32)/(D42-D32)
C     IF(IT .LT. 5) THEN
C     DO 10 J=3,10
C     J1=J+1
C     DO 20 JM=J1,JE
C  20 IF(GG(I,J) .GT. GG(I,JM)) GG(I,J)=GG(I,JM)
C  10 CONTINUE
C     END IF

      ELSE IF(NC .EQ. 2) THEN

      GG(I,JE)=GG(I,NNY)
      IF(I.GE.NB2.AND.I.LE.NB3) THEN
      DO 60 J=3,JJ-1
   60 GG(I,J)=GG(I,JJ)*Y(I,J)/Y(I,JJ)
C     GG(I,JJ)=0.
      END IF
      IF(I.EQ.NB3+1) THEN
      DO 70 J=3,JJ
   70 GG(I,J)=GG(I,JJ1)*Y(I,J)/Y(I,JJ1)
      END IF

      ELSE IF(NC .EQ. 3) THEN

      GG(I,JE)=0.
      IM1=I-1
      D2=(Y(I,2)+Y(IM1,2))*0.5
      D4=(Y(I,JJ1)+Y(I,JJ)+Y(IM1,JJ1)+Y(IM1,JJ))*0.25
      D5=(Y(I,JJ2)+Y(I,JJ1)+Y(IM1,JJ2)+Y(IM1,JJ1))*0.25
      D52=D5-D2
      D42=D4-D2
      IF(I.GE.NB2.AND.I.LE.NB3) THEN
      DO 80 J=3,JJ
      D3=(Y(I,J)+Y(I,J-1)+Y(IM1,J)+Y(IM1,J-1))*0.25
      D53=D5-D3
      D43=D4-D3
      GG(I,J)=(D53*GG(I,JJ1)-D43*GG(I,JJ2))/(D53-D43)
   80 CONTINUE
      GG(I,2)=(D52*GG(I,JJ1)-D42*GG(I,JJ2))/(D52-D42)
      ELSE
      D52=D52*D52
```

```
      D42=D42*D42
      IF(I.EQ.NB3+1) THEN
      DO 90 J=3,JJ
C     D3=(Y(I,J)+Y(I,J-1)+Y(IM1,J)+Y(IM1,J-1))*0.25
C     D32=(D3-D2)**2
C     GG(I,J)=GG(I,JJ1)-(GG(I,JJ2)-GG(I,JJ1))*(D42-D32)/(D52-D
      GG(I,J)=0.5*(GG(I+1,J)+GG(I-1,J))
   90 CONTINUE
      END IF
      IF(I.GE.NB1.AND.I.LT.NB2) THEN
      D3=(Y(I,3)+Y(I,2)+Y(IM1,3)+Y(IM1,2))*0.25
      D4=(Y(I,4)+Y(I,3)+Y(IM1,4)+Y(IM1,3))*0.25
      D32=(D3-D2)**2
      D42=(D4-D2)**2
      GG(I,2)=(D42*GG(I,3)-D32*GG(I,4))/(D42-D32)
      END IF
      END IF

      ELSE IF(NC .GE. 4 ) THEN

      IF(I.LE.NB3) RETURN
      D42=Y(I,JJ1)*Y(I,JJ1)
      D52=Y(I,JJ2)*Y(I,JJ2)
      IF(I.EQ.NB3+1) THEN
      DO 100 J=3,JJ
      D32=Y(I,J)*Y(I,J)
      GG(I,J)=GG(I,JJ1)-(GG(I,JJ2)-GG(I,JJ1))*(D42-D32)/(D52-D
  100 CONTINUE
      END IF
C     D32=Y(I,3)*Y(I,3)
C     D42=Y(I,4)*Y(I,4)
C     GG(I,2)=(D42*GG(I,3)-D32*GG(I,4))/(D42-D32)

      END IF

      RETURN
      END

C    ****************************************************************
C    * SUBROUTINE IS USED TO SOLVE 1. TURBULENT KINETIC ENERGY
C    *                             2. DISSIPATION RATE
C    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>

      SUBROUTINE STKD
      IMPLICIT REAL*8(A-H,O-Z)

$INSERT BLOCK.MAIN

      IE=NNX+1
      JE=NNY+1
C.... SOLVE K EQUATION
```

```
      CALL EQCOE( 4 )
      CALL COEF (4)

      DO 10 I=NB2,NNX
      DO 10 J=JB(I),NNY
      FT(I,J)=ZC(I,J)*SK(I,J)
   10 CONTINUE

      CALL SOLVE(TK,FT,4,NB2,IE,JE)
      CALL CHECK(TK,4,IE,JE)

      DO 60 I=NB2,IE
      DO 60 J=2,JE
      IF(TK(I,J).LT.1.D-9) TK(I,J)=1.D-9
 60   CONTINUE

C.... SOLVE DISSIPATION EQUATION

      CALL EQCOE(5)
      CALL COEF (5)

      DO 20 I=NB2,NNX
      DO 20 J=JB(I),NNY
      FT(I,J)=ZC(I,J)*SD(I,J)
   20 CONTINUE

      CALL SOLVE(TD,FT,5,NB2,IE,JE)
      CALL CHECK(TD,5,IE,JE)

C     DO 30 I=NB2,IE
C     DO 30 J=2,JE
C     IF(TD(I,J).LT.1.D-9) TD(I,J)=1.D-9
C     EV(I,J)=TK(I,J)*TK(I,J)/TD(I,J)*CNU
C30   CONTINUE
C     CALL CHECK(EV,6,IE,JE)
C     DO 50 I=NB2,IE
C     DO 50 J=5,JE
C     IF(EV(I,J-1).LE.EV(I,J-2) .AND. EV(I,J-1).LT.EV(I,J))
C    $ EV(I,J)=EV(I,J-1)
C50   CONTINUE
C     DO 90 I=NB2-1,1,-1
C     DO 90 J=3,JE
C     TK(I,J)=TK(I+1,J)*0.8
C     TD(I,J)=TD(I+1,J)*0.8
C     EV(I,J)=EV(I+1,J)*0.8
C  90 CONTINUE

      RETURN
      END
```

```
C    ************************************************************
C    * SUBROUTINE WALLFN IS USED TO DEFINE THE BOUNDAUY CONDITI
C    >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
C      YD2: DISTANCE OF THE FIRST NODE
C      YD3: DISTANCE OF THE SECOND NODE
C      U3: VELOCITY OF SECOND NODE
C      U2:                FIRST  NODE
C      RKAR: RECIPROCAL OF KARMAN CONSTANT
C      E    : LOG LAW CONSTANT E
C      RE   : REYONDS NUMBER
C      SHSQR: SHEAR STRESS AT FIRST TWO

      SUBROUTINE WALLFN
      IMPLICIT REAL*8 (A-H,O-Z)

$INSERT BLOCK.MAIN

      DO 900 I=NB2,NB3
      JJ=JB(I)
      JM1=JJ-1
      U3=U(I,JJ)
      YD3=DABS((Y(I,JJ)+Y(I,JM1))*0.5-Y(I,2))
      YTD=DABS(Y(I,JM1))

      ARG=RE*E*YD3
      SHEAR=0.1
       AVEL=DABS(U3)
      RKAR=1./AK

      DO 10 IH=1, 100

        ARSH=ARG*SHEAR
        DENUM=RKAR*(1.+DLOG(ARSH))

        SHNEW=( RKAR*SHEAR+AVEL)/DENUM

        DIFF=DABS(SHNEW-SHEAR)
        SHEAR=SHNEW
        IF( DIFF.LE.1.D-7) GO TO 20

   10  CONTINUE
   20  SIGN=U3/AVEL
       SHSQR=SHEAR
       DO 30 J=JM1,3,-1
       YD2=DABS(0.5*(Y(I,J)+Y(I,J-1))-Y(I,2))
       YT2=DABS(Y(I,J))
       ARG=RE*E*YD2*SHSQR
       SHEAR=SHSQR*SHSQR

       TK(I,J)=SHEAR/DSQRT(CNU)*YT2/YTD
       TD(I,J)=RKAR*SHEAR*SHSQR/YT2
```

```
      YPLUS=ARG/E

      IF(YPLUS.GT.20.)  THEN
      U(I,J)=RKAR*DLOG(ARG)*SHSQR*SIGN
      ELSE
      YD4=DABS(0.5*(Y(I,J+1)+Y(I,J))-Y(I,2))
      U(I,J)=U(I,J+1)*YD2/YD4
      END IF

   30 CONTINUE

      UTA(I)=SHSQR
      V(I,JM1)=V(I,JJ)*U(I,JM1)/U3
  900 CONTINUE

      RETURN
      END
C
C*************************************************************
C     SUBROUTINE TRIDAG TO SOLVE ALGEBRAIC EQUATIONS
C     SIMULTANEOUSLY FOR EACH ROW OR COLOUM
C*************************************************************
C
C
      SUBROUTINE TRIDAG(IF,L,A,B,C,D,V)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(99),B(99),C(99),D(99),V(99),BETA(99),GAMMA(9
      BETA(IF)=B(IF)
      GAMMA(IF)=D(IF)/BETA(IF)
      IFP1=IF+1
      DO 1 I=IFP1,L
      BETA(I)=B(I)-A(I)*C(I-1)/BETA(I-1)
    1 GAMMA(I)=(D(I)-A(I)*GAMMA(I-1))/BETA(I)
      V(L)=GAMMA(L)
      LAST=L-IF
      DO 2 K=1,LAST
      I=L-K
    2 V(I)=GAMMA(I)-C(I)*V(I+1)/BETA(I)
      RETURN
      END
C*********************************************
C
C....  SUBROUTINE MAIN3D
C
C*********************************************
C.... MAIN3D IS USED TO ARRANGE THE COMMON SPACE
C.... FOR FLOWS PAST AN AXISYMMETRIC BODY WITH
C.... OR WITHOUT ANGLE OF ATTACK. THE MAXIMUM SPACE
C.... CAN BE INCREASED OR DECREASED DEPENDING ON
C.... THE PROBLEM.
C
```

```
      SUBROUTINE MAIN3D
      IMPLICIT REAL*8 (A-H, O-Z)
      COMMON M(2000000)
      COMMON/NUMBER/N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,
     $              N11,N12,N13,N14,N15,N16,N17,N18,N19,N20,
     $              N21,N22,N23,N24,N25,N26,N27,N28,N29,N30,
     $              N31,N32,N33,N34,N35,N36,N37,N38,N39,N40,
     $              N41,N42,N43,N44,N45,N46,N47,N48,N49,N50,
     $              N51,N52,N53,N54,N55,N56,N57,N58,N59,N60,
     $              N61,N62,N63,N64,N65,N66,N67,N68,N69,N70,
     $              N71,N72,N73,N74,N75,N76,N77,N78,N79,N80,
     $              N81,N82,N83,N84,N85,N86,N87,N88,N89,N90,
     $              N91,N92,N93,N94,N95,N96,N97,N98,N99,N100,
     $              N101,N102,N103,N104,N105,N106,N107,N108
      COMMON/COEF1/ IE,JE,KE
      DATA IPR/2/
C
      MAXI=2000000
      I1=IE*JE*KE*IPR
      I2=JE*KE*IPR
      I3=IE*KE*IPR
      I4=IE*IPR
      I5=KE*IPR
      N1=1
      N2=N1+I1
      N3=N2+I1
      N4=N3+I1
      N5=N4+I1
      N6=N5+I1
      N7=N6+I1
      N8=N7+I1
      N9=N8+I1
      N10=N9+I1
      N11=N10+I1
      N12=N11+I1
      N13=N12+I1
      N14=N13+I1
      N15=N14+I1
      N16=N15+I1
      N17=N16+I3
      N18=N17+I3
      N19=N18+I3
      N20=N19+I3
      N21=N20+I3
      N22=N21+I3
      N23=N22+I3
      N24=N23+I3
      N25=N24+I3
      N26=N25+I3
      N27=N26+I2
      N28=N27+I2
```

```
N29=N28+I2
N30=N29+I2
N31=N30+I2
N32=N31+I2
N33=N32+I2
N34=N33+I2
N35=N34+I2
N36=N35+I2
N37=N36+I2
N38=N37+I2
N39=N38+I2
N40=N39+I2
N41=N40+I2
N42=N41+I2
N43=N42+I2
N44=N43+I2
N45=N44+I2
N46=N45+I2
N47=N46+I2
N48=N47+I2
N49=N48+I2
N50=N49+I2
N51=N50+I2
N52=N51+I2
N53=N52+I2
N54=N53+I2
N55=N54+I2
N56=N55+I2
N57=N56+I2
N58=N57+I2
N59=N58+I2
N60=N59+I2
N61=N60+I2
N62=N61+I2
N63=N62+I2
N64=N63+I2
N65=N64+I2
N66=N65+I2
N67=N66+I2
N68=N67+I2
N69=N68+I2
N70=N69+I2
N71=N70+I2
N72=N71+I2
N73=N72+I2
N74=N73+I2
N75=N74+I2
N76=N75+I2
N77=N76+I2
N78=N77+I2
N79=N78+I2
```

```
      N80=N79+I2
      N81=N80+I2
      N82=N81+I2
      N83=N82+I2
      N84=N83+I2
      N85=N84+I2
      N86=N85+I2
      N87=N86+I2
      N88=N87+I2
      N89=N88+I2
      N90=N89+I2
      N91=N90+I2
      N92=N91+I2
      N93=N92+I2
      N94=N93+I2
      N95=N94+I2
      N96=N95+I2
      N97=N96+I2
      N98=N97+I2
      N99=N98+I2
      N100=N99+I2
      N101=N100+I2
      N102=N101+I4
      N103=N102+I4
      N104=N103+I4
      N105=N104+I4
      N106=N105+I5
      N107=N106+I5
      N108=N107+I5
      N109=N108+I5
C
      IF(N109 .GE. MAXI) THEN
      PRINT 10, MAXI, N109
      ELSE
      NMAX=MAXI/IPR
      NN=N109/IPR
      CALL ZERO(M,NMAX,NN)
      CALL MESH(M(N101),M(N13),M(N14),M(N102),M(N15),M(N108),
     $          IE,JE,KE)
      CALL STAG3D(M(N1),M(N2),M(N3),M(N4),M(N5),
     $     M(N6),M(N7),M(N8),M(N9),M(N10),
     $     M(N11),M(N12),M(N13),M(N14),M(N15),
     $     M(N16),M(N17),M(N18),M(N19),M(N20),
     $     M(N21),M(N22),M(N23),M(N24),M(N25),
     $     M(N26),M(N27),M(N28),M(N29),M(N30),
     $     M(N31),M(N32),M(N33),M(N34),M(N35),
     $     M(N36),M(N37),M(N38),M(N39),M(N40),
     $     M(N41),M(N42),M(N43),M(N44),M(N45),
     $     M(N46),M(N47),M(N48),M(N49),M(N50),
     $     M(N51),M(N52),M(N53),M(N54),M(N55),
     $     M(N56),M(N57),M(N58),M(N59),M(N60),
```

```
     $     M(N101),M(N102),M(N103),M(N104),M(N105),
     $     M(N106),M(N107),M(N108),IE,JE,KE)
      END IF
C
   10 FORMAT('  WARNING***  INSUFFICIENT STORAGE  ',/5X,
     $       '  MAXIMUM = ',I10,5X,'PRACTICAL = ', I10)
      RETURN
      END
C     ***********************************************
C     SUBROUTINE MESH IS USED TO READ COORDINATES
C     ***********************************************
C
      SUBROUTINE MESH(XP,YP,ZP,F1,F2,F3,IP1,JP1,KP1)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION YP(IP1,JP1,KP1)
      DIMENSION ZP(IP1,JP1,KP1)
      DIMENSION F2(IP1,JP1,KP1)
      DIMENSION XP(IP1),F1(IP1),F3(KP1)
      COMMON/COEF4/ UI, VI, M1, M2, M3
C
      OPEN(UNIT=11,FILE='PHYSBODY')
      READ(11,2005) M1,M2,M3
      READ(11,2006)(XP(I),I=1,IP1)
      READ(11,2006)(F1(I),I=1,IP1)
      READ(11,2006)(YP(1,J,1),J=1,JP1)
      READ(11,2006) F2(1,1,1)
      CLOSE(11)
 2005 FORMAT(6I10)
 2006 FORMAT(5E14.7)
C
      DO 422 K=1,KP1
      F3(K)=0.
      DO 421 J=1,JP1
      DO 421 I=1,IP1
      ZP(I,J,K)=K*1.
      YP(I,J,K)=YP(1,J,1)
      F2(I,J,K)=F2(1,1,1)
  421 CONTINUE
  422 CONTINUE
      RETURN
      END
      SUBROUTINE ZERO(V,NMAX,NN)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION V(NMAX)
      DO 10 I=1,NN
   10 V(I)=1.E-25
      RETURN
      END
```

```
C***********************  ********************************
C
C.... SUBROUTINE STAG3D
C
C*********************************************************
C.... STAG3D IS USED TO SOLVE 3D FLOWS PAST AN AXISYMMETRIC
C.... BODY WITH STAGGER GRID SYSTEM.
C
      SUBROUTINE STAG3D(UO,VO,WO,AKEO,ADSO,ZUT,PR,PP,DH,
     S                  BCU,BCV,BCW,YP,ZP,F2,DPDYU,DPDYW,
     S                  DPDZU,DPDZV,YP2,YP3,YP4,VSF,WSF,UTAUA,
     S                  BD,BU,BV,BW,DS,US,VS,WS,
     S                  UHP,UHF,VH,WH,UB,UP,UF,VB,VP,VF,WB,WP,
     S                  AKEB,AKEP,ADSP,FU,GW,DF,
     S                  CU,CV,CW,CUY,CWY,CUZ,CVZ,XP,F1,UL,UTL,
     S                  TAUW,UTAU,YPP,F3,IP1,JP1,KP1)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION  UO(IP1,JP1,KP1)
      DIMENSION  VO(IP1,JP1,KP1)
      DIMENSION  WO(IP1,JP1,KP1)
      DIMENSION  AKEO(IP1,JP1,KP1)
      DIMENSION  ADSO(IP1,JP1,KP1)
      DIMENSION  ZUT(IP1,JP1,KP1)
      DIMENSION  PR(IP1,JP1,KP1)
      DIMENSION  PP(IP1,JP1,KP1)
      DIMENSION  DH(IP1,JP1,KP1)
      DIMENSION  BCU(IP1,JP1,KP1)
      DIMENSION  BCV(IP1,JP1,KP1)
      DIMENSION  BCW(IP1,JP1,KP1)
      DIMENSION  YP(IP1,JP1,KP1)
      DIMENSION  ZP(IP1,JP1,KP1)
      DIMENSION  F2(IP1,JP1,KP1)
      DIMENSION  DPDYU(IP1,KP1),DPDYW(IP1,KP1),DPDZU(IP1,KP1),
     S           DPDZV(IP1,KP1),YP2(IP1,KP1),YP3(IP1,KP1),YP4(
     S           VSF(IP1,KP1),WSF(IP1,KP1),UTAUA(IP1,KP1)
      DIMENSION  BD(JP1,KP1),BU(JP1,KP1),BV(JP1,KP1),BW(JP1,KP
     S           DS(JP1,KP1),GG(JP1,KP1)
      DIMENSION  US(JP1,KP1),VS(JP1,KP1),WS(JP1,KP1),
     S           UHP(JP1,KP1),UHF(JP1,KP1),VH(JP1,KP1),WH(JP1,
     S           UB(JP1,KP1),UP(JP1,KP1),UF(JP1,KP1),
     S           VB(JP1,KP1),VP(JP1,KP1),VF(JP1,KP1),
```

```
     $             WB(JP1,KP1),WP(JP1,KP1),WF(JP1,KP1)
      DIMENSION  FU(JP1,KP1),GW(JP1,KP1),DF(JP1,KP1),
     $             AKEB(JP1,KP1),AKEP(JP1,KP1),ADSP(JP1,KP1)
      DIMENSION  CU(JP1,KP1),CV(JP1,KP1),CW(JP1,KP1),
     $             CUY(JP1,KP1),CWY(JP1,KP1),CUZ(JP1,KP1),CVZ(JP
      DIMENSION  XP(IP1),F1(IP1),UL(IP1),UTL(IP1),
     $             TAUW(KP1),UTAU(KP1),YPP(KP1),F3(KP1)
      DIMENSION  AA(99),BB(99),CC(99),DD(99),T(99)
      COMMON M(1)
      COMMON/NUMBER/N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,
     $                   N11,N12,N13,N14,N15,N16,N17,N18,N19,N20,
     $                   N21,N22,N23,N24,N25,N26,N27,N28,N29,N30,
     $                   N31,N32,N33,N34,N35,N36,N37,N38,N39,N40,
     $                   N41,N42,N43,N44,N45,N46,N47,N48,N49,N50,
     $                   N51,N52,N53,N54,N55,N56,N57,N58,N59,N60,
     $                   N61,N62,N63,N64,N65,N66,N67,N68,N69,N70,
     $                   N71,N72,N73,N74,N75,N76,N77,N78,N79,N80,
     $                   N81,N82,N83,N84,N85,N86,N87,N88,N89,N90,
     $                   N91,N92,N93,N94,N95,N96,N97,N98,N99,N100,
     $                   N101,N102,N103,N104,N105,N106,N107,N108
      COMMON/COEF2/RE,TAU,IPRINT,ITERT,CD,AK,E
      COMMON/COEF3/NA23,NSR,LOT,NTS,INI,NTY
      COMMON/COEF4/UI,VI,M1,M2,M3
      COMMON/UVW1/IMAX,JMAX,KMAX,JPP,KPP,JA,JAM1,I,KM1,KMM
C
      JPP=JP1
      KPP=KP1
      IMAX=IP1-1
      JMAX=JP1-1
      KMAX=KP1-1
      CD2=DSQRT(CD)
      CD4=DSQRT(CD2)
      CD3=CD4*CD4*CD4
      CD2I=1./CD2
      REI=1./RE
      TAUI=1./TAU
      OPEN(UNIT=6, FILE='OUPT')
      WRITE(6,1232)RE,TAU
 1232 FORMAT(//5X,'RE =',F10.1,5X,'TAU =',F6.3//)
      WRITE(6,2005)(XP(I),I=1,IP1)
C
      M21=M2-1
      M23=M2-3
      M31=M3-1
      KM1=2
      KMM=2
      IF(NA23.EQ.3) THEN
      KM1=KMAX-1
      KMM=3
      END IF
      ITERA=3
```

```
      ITPP=2
      EPE=0.0001
      ABCD=0.06
      JOPT=3
C---- INITIAL PROFILES AT INLET STATION
      IF(INI .EQ. 1) THEN
      DO 935 K=1,KP1
      DO 935 J=1,JP1
      VIV=ZP(1,J,K)
      VIW=.5*(ZP(1,J,K)+ZP(1,J,k+1))
      UO(1,J,K)=UI
      VO(1,J,K)=VI*DSIN(VIV)
      WO(1,J,K)=VI*DCOS(VIW)
  935 CONTINUE
      ELSE
      OPEN(UNIT=11,FILE='GUESS')
      END IF
      IPR=0
      CALL ULUTL(M(N101),M(N103),M(N104),RE,IP1,M1,M21,ABCD)
      DO 237 K=1,KP1
 237  UTAUA(M31,K)=ABCD
C---- RETURN POINT OF GLOBAL SWEEPS
      DO 4000 IT=1,ITERT
      PRINT 2500, M3
 2500 FORMAT('  STARTING POINT OF TURBULENT FLOW AT ---', I5)
      IPR=IPR+1
      DO 38 J=1,JP1
      DO 38 K=1,KP1
      UHP(J,K)=UI
   38 CONTINUE
C---- RETURN POINT OF MARCHING PROCESS FOR
C---- CONVECTIVE TRANSPORT EQUATIONS
      DO 3000 I=2,IMAX
      JA=2
      IF(I.GE.M3) JA=JOPT
      JAM1=JA-1
      IF(IT.EQ.1) THEN
      DO 103 I=1,IP1
      DO 103 K=1,KP1
      YP2(I,K)=YP(I,1,K)**2+YP(I,JAM1,K)**2-2.*YP(I,1,K)*YP(I,
     $          *DCOS(ZP(I,JAM1,K)-ZP(I,1,K))
      YP3(I,K)=YP(I,1,K)**2+YP(I,JA,K)**2-2.*YP(I,1,K)*YP(I,JA
     $          *DCOS(ZP(I,JA,K)-ZP(I,1,K))
      YP4(I,K)=YP(I,1,K)**2+YP(I,JA+1,K)**2-2.*YP(I,1,K)*YP(I,
     $          *DCOS(ZP(I,JA+1,K)-ZP(I,1,K))
  103 CONTINUE
      DO 1202 J=1,JP1
      DO 1202 K=1,KP1
      UO(I,J,K)=UO(I-1,J,K)
      VO(I,J,K)=VO(I-1,J,K)
      WO(I,J,K)=WO(I-1,J,K)
```

```
      AKEO(I,J,K)=AKEO(I-1,J,K)
      ADSO(I,J,K)=ADSO(I-1,J,K)
      UO(I+1,J,K)=UO(I,J,K)
      VO(I+1,J,K)=VO(I,J,K)
      WO(I+1,J,K)=WO(I,J,K)
      AKEO(I+1,J,K)=AKEO(I,J,K)
      ADSO(I+1,J,K)=ADSO(I,J,K)
 1202 CONTINUE
      IF(I .EQ. M1) THEN
      DO 1203 K=1,KP1
      UO(M1,1,K)=0.
      VO(M1,1,K)=0.
      WO(M1,1,K)=0.
 1203 CONTINUE
      END IF
      END IF
C---- DEFINE THE BODY VISCOSITY DISTRIBUTION
      IF(I .GE. M3) THEN
      DO 149 K=2,KMAX
      DO 149 J=1,JP1
  149 ZUT(I,J,K)=CD*AKEO(I,J,K)*AKEO(I,J,K)/ADSO(I,J,K)
      DO 150 K=2,KMAX
      DO 150 J=4,JP1
      IF(ZUT(I,J-1,K).LE.ZUT(I,J-2,K).AND.ZUT(I,J-1,K).LT.
     $ZUT(I,J,K)) ZUT(I,J,K)=ZUT(I,J-1,K)
  150 CONTINUE
      DO 143 J=1,JP1
      ZUT(I,J,1)=ZUT(I,J,KMM)
  143 ZUT(I,J,KP1)=ZUT(I,J,KM1)
      IF(IT.EQ.1) THEN
      DO 198 J=1,JP1
      DO 198 K=1,KP1
  198 ZUT(I+1,J,K)=ZUT(I,J,K)
      END IF
      END IF
      IF(IT.GT.1.AND.I.LT.M3) THEN
      DO 189 J=1,JP1
      DO 189 K=1,KP1
      AKEO(I,J,K)=AKEO(I+1,J,K)*0.8
      ADSO(I,J,K)=ADSO(I+1,J,K)*0.8
  189 ZUT(I,J,K)=ZUT(I+1,J,K)*0.3
      END IF
C---- RESET THE SECTION VARIABLES
      DO 190 K=1,KP1
      DO 190 J=1,JP1
      UB(J,K)=UO(I+1,J,K)
      UP(J,K)=UO(I,J,K)
      UF(J,K)=UO(I-1,J,K)
      VB(J,K)=VO(I+1,J,K)
      VP(J,K)=VO(I,J,K)
      VF(J,K)=VO(I-1,J,K)
```

```
      WB(J,K)=WO(I+1,J,K)
      WP(J,K)=WO(I,J,K)
      WF(J,K)=WO(I-1,J,K)
      AKEB(J,K)=AKEO(I+1,J,K)
      AKEP(J,K)=AKEO(I,J,K)
      ADSP(J,K)=ADSO(I,J,K)
  190 CONTINUE
C---- FA COEFFICIENTS OF MOMENTUM EQUATIONS
      CALL FAUVW(M(N101),M(N102),M(N51),M(N52),M(N53),M(N26),
     $           IP1,JP1,KP1,NA23,REI,TAUI)
      DO 900 ITA=1,ITERA
      IF(I.GE.M3.AND.I.LT.M2) THEN
C---- BOUNDARY CONDITIONS: WALL FUNCTION
      XXI=XP(I+1)-XP(I)
      DO 155 K=2,KMAX
      R=0.5*(YP(I,1,K)+YP(I+1,1,K))
      IF(NA23 .EQ. 2) R=1.
      YXI=YP(I+1,1,K)-YP(I,1,K)
      YET=0.5*(YP(I+1,2,K)-YP(I+1,1,K)
     $+YP(I,2,K)-YP(I,1,K))
      YZT=0.25*(YP(I+1,1,K+1)-YP(I+1,1,K-1)
     $+YP(I,1,K+1)-YP(I,1,K-1))
      RZXI=R*(ZP(I+1,1,K)-ZP(I,1,K))
      RZET=0.5*R*(ZP(I+1,2,K)-ZP(I+1,1,K)
     $+ZP(I,2,K)-ZP(I,1,K))
      RZZT=0.25*R*(ZP(I+1,1,K+1)-ZP(I+1,1,K-1)
     $+ZP(I,1,K+1)-ZP(I,1,K-1))
      B11=YET*RZZT-YZT*RZET
      B12=YZT*RZXI-YXI*RZZT
      B13=YXI*RZET-YET*RZXI
      B22=XXI*RZZT
      B23=-XXI*RZET
      B32=-XXI*YZT
      B33=XXI*YET
      G11=XXI*XXI+YXI*YXI+RZXI*RZXI
      G22=YET*YET+RZET*RZET
      G33=YZT*YZT+RZZT*RZZT
      G12=YXI*YET+RZXI*RZET
      G13=YXI*YZT+RZXI*RZZT
      G23=YET*YZT+RZET*RZZT
      G=G11*G22*G33+2.*G12*G13*G23-G23*G23*G11-
     $G13*G13*G22-G12*G12*G33
      GI=1./G
      A11=GI*(G22*G33-G23*G23)
      A22=GI*(G11*G33-G13*G13)
      A33=GI*(G11*G22-G12*G12)
      A12=GI*(G13*G23-G12*G33)
      A13=GI*(G12*G23-G13*G22)
      A23=GI*(G12*G13-G23*G11)
      AJI=DSQRT(GI)
      DG11=DSQRT(G11)
```

```
      DG33=DSQRT(G33)
      COSA=DABS(B22*YET+B32*RZET)/DSQRT(G22*G*A22)
      GRADP=(PR(I+1,1,K)-PR(I,1,K))/(XP(I+1)-XP(I))*COSA
      IF(I.GE.M23.OR.IT.LT.5) GRADP=0.
      U3=UP(JA,K)
      V3=0.25*(VP(JAM1,K)+VB(JAM1,K)
     $+VP(JA,K)+VB(JA,K))
      W3=0.25*(WP(JA,K-1)+WB(JA,K-1)
     $+WP(JA,K)+WB(JA,K))
      Q3=DSQRT(U3*U3+V3*V3+W3*W3)
      Q3XI=AJI*DG11*B11*U3
      Q3ZT=AJI*DG33*(B13*U3+B23*V3+B33*W3)
      UTAU(K)=UTAUA(I,K)
      IF(IT.EQ.1) UTAU(K)=UTAUA(I-1,K)
      DO 555 IJK=1,50
      DPR=GRADP/(RE*UTAU(K)*UTAU(K)*UTAU(K))
      DPR=DMAX1(DPR,EPE)
      DTAU=0.5*DPR
      SQRT3=DSQRT(1.+DTAU*RE*UTAU(K)*DSQRT(YP3(I,K))*COSA)
      UTAUN=Q3/((DLOG(4.*(SQRT3-1.)/(SQRT3+1.)/DTAU)
     $+2.*SQRT3-2.)/AK+5.45+3.7*DPR)
      IF(DABS(UTAUN-UTAU(K)).LT.1.0D-5) GO TO 556
      UTAU(K)=UTAUN
  555 CONTINUE
  556 TAUW(K)=2.*UTAU(K)*UTAU(K)
      YPP(K)=RE*UTAU(K)*DSQRT(YP2(I,K))*COSA
      SQRT2=DSQRT(1.+DTAU*YPP(K))
      Q2=UTAU(K)*((DLOG(4.*(SQRT2-1.)/(SQRT2+1.)/DTAU)
     $+2.*SQRT2-2.)/AK+5.45+3.7*DPR)
      UTAUK=0.5*(UTAUA(I-1,K)+UTAU(K))
      AKEP(JAM1,K)=UTAUK*UTAUK*CD2I
      ADSP(JAM1,K)=UTAUK*UTAUK*UTAUK/(AK*DSQRT(YP2(I,K))*COSA)
      Q2XI=Q3XI*Q2/Q3
      Q2ZT=Q3ZT*Q2/Q3
      UP(JAM1,K)=Q2XI*XXI/DG11
      VSF(I,K)=Q2XI*YXI/DG11+Q2ZT*YZT/DG33
      WSF(I,K)=Q2XI*RZXI/DG11+Q2ZT*RZZT/DG33
  155 CONTINUE
      DO 151 K=2,KMAX
      UTAUA(I,K)=UTAU(K)
      V2=0.5*(VSF(I,K)+VSF(I-1,K))
      HN=0.5*(YP4(I,K)-YP2(I,K))
      HS=0.5*(YP3(I,K)-YP2(I,K))
      VP(JAM1,K)=(HN*V2+HS*VO(I,3,K))/(HN+HS)
      IF(K .EQ. KMAX) GO TO 151
      WP(JAM1,K)=0.25*(WSF(I,K)+WSF(I-1,K)+WSF(I,K+1)+WSF(I,K+
  151 CONTINUE
      UP(JAM1,1)=UP(JAM1,KMM)
      UP(JAM1,KP1)=UP(JAM1,KM1)
      VP(JAM1,1)=VP(JAM1,KMM)
      VP(JAM1,KP1)=VP(JAM1,KM1)
```

```
      WP(JAM1,1)=-WP(JAM1,2)
      WP(JAM1,KMAX)=-WP(JAM1,KM1)
      AKEP(JAM1,1)=AKEP(JAM1,KMM)
      AKEP(JAM1,KP1)=AKEP(JAM1,KM1)
      ADSP(JAM1,1)=ADSP(JAM1,KMM)
      ADSP(JAM1,KP1)=ADSP(JAM1,KM1)
      END IF
      DO 304 K=1,KP1
      DO 304 J=1,JP1
      US(J,K)=UP(J,K)
      VS(J,K)=VP(J,K)
      WS(J,K)=WP(J,K)
  304 CONTINUE
C---- CALCULATE THE STAR VELOCITY FIELD
      CALL FASVEL(PR,M(N16),M(N17),M(N18),M(N19),M(N26),M(N54)
     $            M(N55),M(N56),M(N57),M(N58),M(N59),M(N60),
     $            M(N32),M(N33),M(N34),M(N41),M(N27),M(N28),
     $            M(N29),M(N30),M(N53),M(N31),
     $            IP1,JP1,KP1,NA23)
C---- CALCULATE THE PRESSURE-CORRECTION FIELD
  179 DO 797 J=1,JP1
      DO 797 K=1,KP1
      PP(I-1,J,K)=0.
      PP(I,J,K)=0.
  797 PP(I+1,J,K)=0.
      DO 798 ITER=1,ITPP
      CALL PRESU(PP,M(N31),M(N10),M(N11),M(N12),0,IP1,JP1,KP1)
      DO 796 J=1,JP1
      PP(I,J,1)=PP(I,J,KMM)
  796 PP(I,J,KP1)=PP(I,J,KM1)
  798 CONTINUE
C---- CORRECT VELOCITY FIELD BY PP
      CALL FAVELCOR(PP,M(N40),M(N43),M(N46),M(N32),M(N33),M(N3
     $               M(N54),M(N55),M(N56),M(N29),M(N20),M(N21),
     $               IP1,JP1,KP1,NA23)
  900 CONTINUE
C---- CALCULATE THE PSEUDO-VELOCITY FIELD
      CALL FAHVEL(M(N35),M(N36),M(N37),M(N38),M(N43),M(N26),M(
     $       M(N58),M(N59),M(N60),M(N16),M(N17),M(N18),M(N19),
     $            M(N27),M(N28),M(N29),M(N30),M(N53),
     $            IP1,JP1,KP1,NA23)
      IF(I .GE. M3) THEN
C---- CALCULATE THE TURBULENT QUANTITIES
      CALL FAUVW(M(N101),M(N102),M(N51),M(N52),M(N53),M(N26),
     $           IP1,JP1,KP1,0,REI,TAUI)
      END IF
C---- UPDATE TRANSPORT QUANTITIES AT UPSTREAM STATION
      DO 679 K=1,KP1
      DO 679 J=1,JP1
      UO(I,J,K)=UP(J,K)
      VO(I,J,K)=VP(J,K)
```

```
      WO(I,J,K)=WP(J,K)
      AKEO(I,J,K)=AKEP(J,K)
  679 ADSO(I,J,K)=ADSP(J,K)
      IF(I.EQ.IMAX) THEN
      DO 373 K=1,KP1
      DO 373 J=1,JP1
      UO(IP1,J,K)=UO(IMAX,J,K)
      VO(IP1,J,K)=VO(IMAX,J,K)
      WO(IP1,J,K)=WO(IMAX,J,K)
      AKEO(IP1,J,K)=AKEO(IMAX,J,K)
      ADSO(IP1,J,K)=ADSO(IMAX,J,K)
      ZUT(IP1,J,K)=ZUT(IMAX,J,K)
  373 CONTINUE
      END IF
      IF(IPR .EQ. IPRINT) THEN
      WRITE(6,2)IT,I,XP(I),TAU
    2 FORMAT(//5X,'NO. OF ITERATION =',I3,5X,'STATION',I3
     $,5X,'X ='F7.4,5X,'TAU =',F6.3)
      IF(I.LT.M3.OR.I.GE.M2) GO TO 2222
      DO 2098 K=2,KMAX
 2098 WRITE(6,2099) K,YPP(K),UTAU(K),TAUW(K)
 2099 FORMAT(5X,'K=',I5,5X,'YPLUS=',E12.4,5X,'UTAU=',E12.4,5X,
     $'TAUW=',E12.4)
 2222 WRITE(6,3001)
 3001 FORMAT(/5X,'VELOCITY U='//)
      DO 3002 K=2,KMAX
 3002 WRITE(6,2007) (UP(J,K),J=1,JP1)
      WRITE(6,3003)
 3003 FORMAT(/5X,'VELOCITY V='//)
      DO 3004 K=2,KMAX
 3004 WRITE(6,2007) (VP(J,K),J=1,JP1)
C      WRITE(6,3005)
C 3005 FORMAT(/5X,'VELOCITY W='//)
C      DO 3006 K=2,KMAX
C 3006 WRITE(6,2007) (WP(J,K),J=1,JP1)
      WRITE(6,3007)
 3007 FORMAT(/5X,'TURBULENT KINETIC ENERGY='//)
      DO 3008 K=2,KMAX
 3008 WRITE(6,2007) (AKEP(J,K),J=1,JP1)
      WRITE(6,3009)
 3009 FORMAT(/5X,'TURBULENT DISSIPATION='//)
      DO 3010 K=2,KMAX
 3010 WRITE(6,2007) (ADSP(J,K),J=1,JP1)
      WRITE(6,3011)
 3011 FORMAT(/5X,'PRESSURE ='//)
      DO 3012 K=2,KMAX
 3012 WRITE(6,2007) (PR(I,J,K),J=1,JP1)
C      WRITE(6,3013)
C 3013 FORMAT(/5X,'MASS SOURCE ='//)
C      DO 3014 K=2,KMAX
C 3014 WRITE(6,2007) (DS(J,K),J=1,JP1)
```

```
      WRITE(6,3015)
 3015 FORMAT(/5X,'EDDY VISCOSITY ='//)
      DO 3016 K=2,KMAX
 3016 WRITE(6,2007) (ZUT(I,J,K),J=1,JP1)
      END IF
 3000 CONTINUE
 2007 FORMAT(6E12.4)
      IF(IPR .EQ. IPRINT) IPR=0
C---- UPDATE THE PRESSURE FIELD
      CALL FAPRESS(PR,PP,M(N9),M(N31),M(N20),M(N21),M(N22),M(N
     $             IP1,JP1,KP1,JOPT)
      WRITE(6,4111) IT,M3
 4111 FORMAT(/5X,'NO.  OF  ITERATION =',I5,5X,'M3 =',I5)
      WRITE(6,4110)
 4110 FORMAT(/5X,'PRESSURE DISTRIBUTION ALONG THE WALL')
      DO 4009 K=2,KMAX
      WRITE(6,2005) (PR(I,1,K),I=1,IP1)
 4009 CONTINUE
      WRITE(6,4112)
 4112 FORMAT(/5X,'SKIN-FRICTION COEFFICIENT')
      DO 4221 K=2,KMAX
      DO 4222 I=M1,M21
      IF(I .LT. M3) THEN
      UT=UTL(I)*UO(I,2,K)/UL(I)
      ELSE
      UT=UTAUA(I,K)
      END IF
      TAUW(I)=2.*UT*UT
 4222 CONTINUE
      WRITE(6,2005) (TAUW(I),I=M1,M21)
 4221 CONTINUE
      WRITE(6,4113)
 4113 FORMAT(/5X,'CENTERLINE VELOCITY')
      DO 4140 K=2,KMAX
      WRITE(6,2005) (UO(I,1,K),I=M2,IMAX)
      IF(IT .GT. 2) THEN
      IF(UO(M3,2,K).LT.UL(M3+1)) M3=M3+1
      IF(UO(M3,2,K).GT.UO(M31,2,K)) M3=M3-1
      END IF
      VSF(M1-1,K)=VO(M1-1,2,K)
 4140 CONTINUE
 4000 CONTINUE
      IF(IT .LT. ITERT+9) GO TO 9999
      DO 5001 I=1,IP1
      DO 5001 J=1,JP1
      DO 5001 K=1,KP1
 5001 WRITE(6,2005) UO(I,J,K),VO(I,J,K),WO(I,J,K),AKEO(I,J,K)
     $,ADSO(I,J,K),PR(I,J,K)
 2005 FORMAT(6E12.4)
 9999 CLOSE (11)
      CLOSE (6)
```

```
      CALL EXIT
      END
C*************************************************
C
C.... SUBROUTINE ULUTL
C
C*************************************************
      SUBROUTINE ULUTL(XP,UL,UTL,RE,IP1,M1,M21,ABCD)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION XP(IP1),UL(IP1),UTL(IP1)
C
      ABC=0.33206
      DO 35 I=M1,M21
      XX=0.5*(XP(I)+XP(I+1))
      REX=DSQRT(RE*XX)
   35 UTL(I)=DSQRT(ABC/REX)
      DO 33 I=19,29
   33 UL(I)=1.0
      UL(30)=0.9994
      UL(31)=0.9852
      UL(32)=0.9250
      UL(33)=0.8200
      UL(34)=0.7050
      UL(35)=0.6027
      UL(36)=0.5275
      UL(37)=0.4862
      ABCD=0.06
      RETURN
      END
C*******************************************************
C
C.... SUBROUTINE FAUVW
C
C*******************************************************
      SUBROUTINE FAUVW(XP,F1,FU,GW.DF,GG,
     $                 IP1,JP1,KP1,NA23,REI,TAUI)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON M(1)
      COMMON/UVW1/IMAX,JMAX,KMAX,JPP,KPP,JA,JAM1,I,KM1,KMM
      COMMON/UVW2/A1,A2,A3,A4.UU,VV,WW
      COMMON/UVW3/REFF,ZUTP,ZUTXI,ZUTET.ZUTZT.FX1,FY2,FZ3
      COMMON/UVW7/R,XXI,YXI,YET,YZT.RZXI.RZET,RZZT
      COMMON/UVW8/C1,C2,CEFFK.CEFFD
      COMMON/NUMBER/N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,
     $              N11,N12,N13,N14,N15,N16,N17,N18,N19,N20,
     $              N21,N22,N23,N24,N25,N26,N27,N28,N29,N30,
     $              N31,N32,N33,N34,N35,N36,N37,N38,N39,N40,
     $              N41,N42,N43,N44,N45,N46,N47,N48,N49,N50,
     $              N51,N52,N53,N54,N55,N56,N57,N58,N59,N60,
     $              N61,N62,N63,N64,N65,N66,N67,N68,N69,N70,
     $              N71,N72,N73,N74,N75,N76,N77,N78,N79,N80,
```

```
     $              N81,N82,N83,N84,N85,N86,N87,N88,N89,N90,
     $              N91,N92,N93,N94,N95,N96,N97,N98,N99,N100,
     $              N101,N102,N103,N104,N105,N106,N107,N108
      DIMENSION FU(JP1,KP1),GW(JP1,KP1),GF(JP1,KP1),GG(JP1,KP1
     $          XP(IP1),F1(IP1)
C
C---- FA MOMENTUM EQUATION
C
      IF(NA23 .NE. 0) THEN
      MM=1
C---- X-MOMENTUM
      XXI=XP(I+1)-XP(I)
      A1=2.
      A2=1.
      A3=1.
      A4=1.
      FX1=0.5*(F1(I)+F1(I+1))
      CALL FACST(M(N1),M(N13),M(N14),M(N15),M(N108),M(N6),
     $          M(N39),M(N40),M(N41),M(N42),M(N43),M(N44),
     $          M(N45),M(N46),M(N47),M(N48),M(N49),M(N50),
     $          M(N100),M(N61),M(N62),M(N63),M(N64),M(N65),
     $          M(N66),M(N67),M(N68),M(N69),M(N70),M(N71),
     $          M(N72),M(N73),M(N28),M(N27),M(N54),M(N57),
     $          M(N59),M(N10),
     $          IP1,JP1,KP1,MM,NA23,REI,TAUI)
      MM=2
C---- Y-MOMENTUM
      XXI=0.5*(XP(I+1)-XP(I-1))
      A1=1.
      A2=2.
      FX1=F1(I)
      CALL FACST(M(N2),M(N13),M(N14),M(N15),M(N108),M(N6),
     $          M(N39),M(N40),M(N41),M(N42),M(N43),M(N44),
     $          M(N45),M(N46),M(N47),M(N48),M(N49),M(N50),
     $          M(N100),M(N74),M(N75),M(N76),M(N77),M(N78),
     $          M(N79),M(N80),M(N81),M(N82),M(N83),M(N84),
     $          M(N85),M(N86),FU,M(N29),M(N55),M(N60),
     $          M(N59),M(N11),
     $          IP1,JP1,KP1,MM,NA23,REI,TAUI)
      IF(NA23 .EQ. 3) THEN
      MM=3
C---- Z-MOMENTUM
      A2=1.
      A3=2.
      CALL FACST(M(N3),M(N13),M(N14),M(N15),M(N108),M(N6),
     $          M(N39),M(N40),M(N41),M(N42),M(N43),M(N44),
     $          M(N45),M(N46),M(N47),M(N48),M(N49),M(N50),
     $          M(N100),M(N87),M(N88),M(N89),M(N90),M(N91),
     $          M(N92),M(N93),M(N94),M(N95),M(N96),M(N97),
     $          M(N98),M(N99),GW,M(N30),M(N56),M(N58),
     $          M(N59),M(N12),
```

```
     $          IP1,JP1,KP1,MM,NA23,REI,TAUI)
      END IF
C
C---- CALCULATE THE MASS SOURCE AND PRESSURE GRADIENTS
C
      DO 500 J=JA,JMAX
      DO 500 K=2,KMAX
      DF(J,K)=FU(J,K)-FU(J-1,K)+GW(J,K)-GW(J,K-1)
  500 CONTINUE
C
      ELSE
C---- TURBULENT EQUATION
      ITUVW=2
      DO 697 K=1,KP1
      DO 697 J=1,JP1
  697 GG(J,K)=0.
      MM=4
C---- K-EQUATION
      XXI=0.5*(XP(I+1)-XP(I-1))
      FX1=F1(I)
      A1=1.
      A2=1.
      A3=1.
      A4=CEFFK
      CALL FACST(M(N4),M(N13),M(N14),M(N15),M(N108),M(N6),
     $          M(N39),M(N40),M(N41),M(N42),M(N43),M(N44),
     $          M(N45),M(N46),M(N47),M(N48),M(N49),M(N50),
     $          M(N100),M(N87),M(N88),M(N89),M(N90),M(N91),
     $          M(N92),M(N93),M(N94),M(N95),M(N96),M(N97),
     $          M(N98),M(N99),GW,M(N30),M(N56),M(N58),
     $          M(N5  ),M(N12),
     $          IP1,JP1,KP1,MM,NA23,REI,TAUI)
      CALL SVEL(M(N49),GG,M(N4),M(N87),M(N88),M(N89),M(N90),M(
     $         M(N92),M(N93),M(N94),M(N95),M(N96),M(N97),M(N9
     $         M(N99),M(N20),M(N21),M(N22),M(N29),
     $         IP1,JP1,KP1,ITUVW,1,3,1)
      MM=5
C---- D-EQUATION
      A4=CEFFD
      CALL FACST(M(N5),M(N13),M(N14),M(N15),M(N108),M(N6),
     $          M(N39),M(N40),M(N41),M(N42),M(N43),M(N44),
     $          M(N45),M(N46),M(N47),M(N48),M(N49),M(N50),
     $          M(N100),M(N87),M(N88),M(N89),M(N90),M(N91),
     $          M(N92),M(N93),M(N94),M(N95),M(N96),M(N97),
     $          M(N98),M(N99),GW,M(N30),M(N56),M(N58),
     $          M(N59),M(N12),
     $          IP1,JP1,KP1,MM,NA23,REI,TAUI)
      CALL SVEL(M(N50),GG,M(N5),M(N87),M(N88),M(N89),M(N90),M(
     $         M(N92),M(N93),M(N94),M(N95),M(N96),M(N97),M(N9
     $         M(N99),M(N20),M(N21),M(N22),M(N29),
     $         IP1,JP1,KP1,ITUVW,1,3,1)
```

```
      END IF
      RETURN
      END
C************************************************
C
C.... SUBROUTINE FASVEL
C
C************************************************
      SUBROUTINE FASVEL(PR,DPDYU,DPDYW,DPDZU,DPDZV,GG,CU,CV,CW
     $                  CUY,CWY,CUZ,CVZ,US,VS,WS,UF,BD,BU,BV,
     $                  BW,DF,DS,IP1,JP1,KP1,NA23)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON M(1)
      COMMON/UVW1/IMAX,JMAX,KMAX,JPP,KPP,JA,JAM1,I,KM1,KMH
      COMMON/NUMBER/N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,
     $              N11,N12,N13,N14,N15,N16,N17,N18,N19,N20,
     $              N21,N22,N23,N24,N25,N26,N27,N28,N29,N30,
     $              N31,N32,N33,N34,N35,N36,N37,N38,N39,N40,
     $              N41,N42,N43,N44,N45,N46,N47,N48,N49,N50,
     $              N51,N52,N53,N54,N55,N56,N57,N58,N59,N60,
     $              N61,N62,N63,N64,N65,N66,N67,N68,N69,N70,
     $              N71,N72,N73 N74,N75,N76,N77,N78,N79,N80,
     $              N81,N82,N83,N84,N85,N86,N87,N88,N89,N90,
     $              N91,N92,N93,N94,N95,N96,N97,N98,N99,N100,
     $              N101,N102,N103,N104,N105,N106,N107,N108
      DIMENSION PR(IP1,JP1,KP1)
      DIMENSION DPDYU(JP1,KP1),DPDYW(JP1,KP1),DPDZU(JP1,KP1)
      DIMENSION DPDZV(JP1,KP1),GG(JP1,KP1),CU(JP1,KP1),CV(JP1,
      DIMENSION CW(JP1,KP1),CUY(JP1,KP1),CUZ(JP1,KP1),CVZ(JP1,
      DIMENSION CWY(JP1,KP1),US(JP1,KP1),VS(JP1.KP1),WS(JP1,KP
      DIMENSION UF(JP1,KP1),DS(JP1,KP1),BD(JP1,KP1),DF(JP1,KP1
      DIMENSION BU(JP1,KP1),BV(JP1,KP1),BW(JP1,KP1)
C
      ITUVW=4
      DO 688 J=JA,JMAX
      DO 688 K=2,KMAX
      DPDYU(J,K)=0.25*(PR(I,J+1,K)+PR(I+1,J+1,K)
     $                -PR(I,J-1,K)-PR(I+1,J-1,K))
      DPDYW(J,K)=0.25*(PR(I,J+1,K)+PR(I,J+1,K+1)
     $                -PR(I,J-1,K)-PR(I,J-1,K+1))
      DPDZU(J,K)=0.25*(PR(I,J,K+1)+PR(I+1,J,K+1)
     $                -PR(I,J,K-1)-PR(I+1,J,K-1))
  688 DPDZV(J,K)=0.25*(PR(I,J+1,K+1)+PR(I,J,K+1)
     $                -PR(I,J+1,K-1)-PR(I,J,K-1))
C
C---- CALCULATE THE LONGITUDIAL VELOCITY FIELD
C
      DO 330 K=2,KMAX
      DO 330 J=JA,JMAX
  330 GG(J,K)=CU(J,K)*(PR(I+1,J,K)-PR(I,J,K))
     $-CUY(J,K)*DPDYU(J,K)+CUZ(J,K)*DPDZU(J,K)
```

```
      CALL SVEL(US,GG,M(N1),M(N61),M(N62),M(N63),M(N64),M(N65)
     $          M(N66),M(N67),M(N68),M(N69),M(N70),M(N71),M(N7
     $          M(N73),M(N20),M(N21),M(N22),M(N29),
     $          IP1,JP1,KP1,ITUVW,1,1,1)
C
C---- CALCULATE THE RADIAL VELOCITY FIELD
C
      DO 430 K=2,KMAX
      DO 430 J=JA,JMAX
  430 GG(J,K)=CV(J,K)*(PR(I,J+1,K)-PR(I,J,K))+
     $CVZ(J,K)*DPDZV(J,K)
      CALL SVEL(VS,GG,M(N2),M(N74),M(N75),M(N76),M(N77),M(N78)
     $          M(N79),M(N80),M(N81),M(N82),M(N83),M(N84),M(N8
     $          M(N86),M(N20),M(N21),M(N22),M(N29),
     $          IP1,JP1,KP1,ITUVW,1,2,3)
C
C---- CALCULATE THE CIRCUMFERENTIAL VELOCITY FIELD
C
      IF(NA23 .EQ. 3) THEN
      DO 530 J=JA,JMAX
      DO 530 K=2,KM1
  530 GG(J,K)=CW(J,K)*(PR(I,J,K+1)-PR(I,J,K))+
     $CWY(J,K)*DPDYW(J,K)
      CALL SVEL(WS,GG,M(N3),M(N87),M(N88),M(N89),M(N90),M(N91)
     $          M(N92),M(N93),M(N94),M(N95),M(N96),M(N97),M(N9
     $          M(N99),M(N20),M(N21),M(N22),M(N29),
     $          IP1,JP1,KP1,ITUVW,2,3,3)
C
      DO 580 J=2,JMAX
      DO 580 K=2,KMAX
  580 DS(J,K)=BD(J,K)*US(J,K)-BU(J,K)*UF(J,K)+
     $BV(J,K)*VS(J,K)-BV(J-1,K)*VS(J-1,K)+BW(J,K)*
     $WS(J,K)-BW(J,K-1)*WS(J,K-1)+DF(J,K)
      END IF
 1000 FORMAT(I10)
 2000 FORMAT(6E12.4)
      RETURN
      END
C**********************************************************
C
C.... SUBROUTINE FAVELCOR
C
C**********************************************************
      SUBROUTINE FAVELCOR(PP,UP,VP,WP,US,VS,WS,CU,CV,CW,BV,
     $                    Y2,Y3,Y4,IP1,JP1,KP1,NA23)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/COEF4/ UI,VI,M1,M2,M3
      COMMON/UVW1/ IMAX,JMAX,KMAX,JPP,KPP,JA,JAM1,I,KM1,KMM
      DIMENSION PP(IP1,JP1,KP1)
      DIMENSION UP(JP1,KP1),VP(JP1,KP1),WP(JP1,KP1)
      DIMENSION US(JP1,KP1),VS(JP1,KP1),WS(JP1,KP1)
```

```
      DIMENSION CU(JP1,KP1),CV(JP1,KP1),CW(JP1,KP1),BV(JP1,KP1
      DIMENSION Y2(IP1,KP1),Y3(IP1,KP1),Y4(IP1,KP1)
C
C---- CORRECT THE IMPERFECT VELOCITY FIELD TO SATISFY
C---- THE EQUATION OF CONTINUITY
C
      DO 879 K=2,KMAX
      DO 800 J=JA,JMAX
  800 UP(J,K)=US(J,K)-CU(J,K)*(PP(I+1,J,K)-PP(I,J,K))
      IF(I.LT.M1.OR.I.GE.M2) THEN
      UP(JAM1,K)=((Y4(I,K)-Y2(I,K))*UP(JA,K)-
     S(Y3(I,K)-Y2(I,K))*UP(JA+1,K))/(Y4(I,K)-Y3(I,K))
      IF(JAM1 .GE. 2) THEN
      JAMM=JAM1-1
      DO 700 J=1,JAMM
  700 UP(J,K)=UP(JAM1,K)
      END IF
      END IF
  879 CONTINUE
      DO 801 J=1,JP1
      UP(J,1)=UP(J,KMM)
  801 UP(J,KP1)=UP(J,KM1)
      DO 810 K=2,KMAX
      DO 811 J=JA,JMAX
  811 VP(J,K)=VS(J,K)-CV(J,K)*(PP(I,J+1,K)-PP(I,J,K))
  810 VP(JP1,K)=VP(JMAX,K)*BV(JMAX,K)/BV(JP1,K)
      DO 812 J=1,JP1
      VP(J,1)=VP(J,KMM)
  812 VP(J,KP1)=VP(J,KM1)
      IF(NA23 .EQ. 3) THEN
      DO 821 K=2,KM1
      DO 821 J=JA,JP1
  821 WP(J,K)=WS(J,K)-CW(J,K)*(PP(I,J,K+1)-PP(I,J,K))
      DO 822 J=1,JP1
      WP(J,1)=-WP(J,2)
  822 WP(J,KMAX)=-WP(J,KM1)
      END IF
      RETURN
      END
C**************************************************
C
C.... SUBROUTINE FAHVEL
C
C**************************************************
      SUBROUTINE FAHVEL(UHP,UHF,VH,WH,VP,GG,CUY,CWY,CUZ,CVZ,
     S                  DPDYU,DPDYW,DPDZU,DPDZV,DH,BD,BU,BV,BW
     S                  IP1,JP1,KP1,NA23)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON M(1)
      COMMON/NUMBER/N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,
     S              N11,N12,N13,N14,N15,N16,N17,N18,N19,N20,
```

```
     $               N21,N22,N23,N24,N25,N26,N27,N28,N29,N30,
     $               N31,N32,N33,N34,N35,N36,N37,N38,N39,N40,
     $               N41,N42,N43,N44,N45,N46,N47,N48,N49,N50,
     $               N51,N52,N53,N54,N55,N56,N57,N58,N59,N60,
     $               N61,N62,N63,N64,N65,N66,N67,N68,N69,N70,
     $               N71,N72,N73,N74,N75,N76,N77,N78,N79,N80,
     $               N81,N82,N83,N84,N85,N86,N87,N88,N89,N90,
     $               N91,N92,N93,N94,N95,N96,N97,N98,N99,N100,
     $               N101,N102,N103,N104,N105,N106,N107,N108
T
      COMMON/UVW1/IMAX,JMAX,KMAX,JPP,KPP,JA,JAM1,I,KM1,KMM
      DIMENSION DH(IP1,JP1,KP1)
      DIMENSION UHP(JP1,KP1),UHF(JP1,KP1),VH(JP1,KP1),WH(JP1,K
      DIMENSION GG(JP1,KP1),CUY(JP1,KP1),CWY(JP1,KP1),CUZ(JP1,
      DIMENSION CVZ(JP1,KP1),DPDYU(JP1,KP1),DPDYW(JP1,KP1)
      DIMENSION DPDZV(JP1,KP1),BD(JP1,KP1),DF(JP1,KP1),DPDZU(J
      DIMENSION BU(JP1,KP1),BV(JP1,KP1),BW(JP1,KP1),VP(JP1,KP1
C
C---- CALCULATE THE PSEUDO-VELOCITY FIELD AND THE
C---- ASSOCIATED MASS SOURCE
C
      DO 671 K=2,KMAX
      DO 671 J=2,JMAX
  671 UHF(J,K)=UHP(J,K)
      DO 672 K=2,KMAX
      DO 672 J=JA,JMAX
  672 GG(J,K)=CUY(J,K)*DPDYU(J,K)
     $+CUZ(J,K)*DPDZU(J,K)
      CALL HVEL(UHP,GG,M(N1),M(N40),M(N61),M(N62),M(N63),M(N64
     $          M(N65),M(N66),M(N67),M(N68),M(N69),M(N70),M(N7
     $          M(N72),M(N73),
     $          IP1,JP1,KP1,1)
      DO 673 K=2,KMAX
      DO 673 J=JA,JMAX
  673 GG(J,K)=CVZ(J,K)*DPDZV(J,K)
      CALL HVEL(VH,GG,M(N2),M(N43),M(N74),M(N75),M(N76),M(N77)
     $          M(N78),M(N79),M(N80),M(N81),M(N82),M(N83),M(N8
     $          M(N85),M(N86),
     $          IP1,JP1,KP1,1)
      DO 674 K=2,KMAX
  674 VH(1,K)=VP(1,K)
      IF(NA23 .NE. 3) GO TO 1234
      DO 676 K=2,KM1
      DO 676 J=JA,JMAX
  676 GG(J,K)=CWY(J,K)*DPDYW(J,K)
      CALL HVEL(WH,GG,M(N3),M(N46),M(N87),M(N88),M(N89),M(N90)
     $          M(N91),M(N92),M(N93),M(N94),M(N95),M(N96),M(N9
     $          M(N98),M(N99),
     $          IP1,JP1,KP1,2)
      DO 677 J=JA,JMAX
      WH(J,1)=-WH(J,2)
```

```
  677 WH(J,KMAX)=-WH(J,KM1)
 1234 DO 680 J=JA,JMAX
      DO 680 K=2,KMAX
  680 DH(I,J,K)=BD(J,K)*UHP(J,K)-BU(J,K)*UHF(J,K)
     $+BV(J,K)*VH(J,K)-BV(J-1,K)*VH(J-1,K)
     $+BW(J,K)*WH(J,K)-BW(J,K-1)*WH(J,K-1)+DF(J,K)
      RETURN
      END
C************************************************************
C
C.... SUBROUTINE FAPRESS
C
C************************************************************
      SUBROUTINE FAPRESS(PR,PP,DH,DS,Y2SQ,Y3SQ,Y4SQ,XP,IP1,JP1
     $                   ,JOPT)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON M(1)
      COMMON/NUMBER/N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,
     $              N11,N12,N13,N14,N15,N16,N17,N18,N19,N20,
     $              N21,N22,N23,N24,N25,N26,N27,N28,N29,N30,
     $              N31,N32,N33,N34,N35,N36,N37,N38,N39,N40,
     $              N41,N42,N43,N44,N45,N46,N47,N48,N49,N50,
     $              N51,N52,N53,N54,N55,N56,N57,N58,N59,N60,
     $              N61,N62,N63,N64,N65,N66,N67,N68,N69,N70,
     $              N71,N72,N73,N74,N75,N76,N77,N78,N79,N80,
     $              N81,N82,N83,N84,N85,N86,N87,N88,N89,N90,
     $              N91,N92,N93,N94,N95,N96,N97,N98,N99,N100,
     $              N101,N102,N103,N104,N105,N106,N107,N108
      COMMON/COEF4/UI,VI,M1,M2,M3
      COMMON/UVW1/IMAX,JMAX,KMAX,JPP,KPP,JA,JAM1,I,KM1,KMM
      DIMENSION PR(IP1,JP1,KP1)
      DIMENSION PP(IP1,JP1,KP1)
      DIMENSION DH(IP1,JP1,KP1)
      DIMENSION Y2SQ(IP1,KP1),Y3SQ(IP1,KP1),Y4SQ(IP1,KP1)
      DIMENSION DS(JP1,KP1),XP(IP1)
C
      RFP=0.3
      ITERP=15
      X1=XP(M3)-XP(M3-1)
      X2=XP(M3+1)-XP(M3)
      DO 1660 I=1,IP1
      DO 1660 J=1,JP1
      DO 1660 K=1,KP1
 1660 PP(I,J,K)=PR(I,J,K)
      DO 3999 ITERG=1,ITERP
      DO 661 II=2,IMAX
      I=IMAX+2-II
      JA=2
      IF(I .GE. M3) JA=JOPT
      JAM1=JA-1
      IF(I.EQ.M3 .OR. I.EQ.(M2-1)) THEN
```

```
      DO 662 K=1,KP1
      DO 662 J=1,JP1
  662 PR(I,J,K)=(X1*PR(I+1,J,K)+X2*PR(I-1,J,K))/(X1+X2)
      ELSE
      DO 666 K=1,KP1
      DO 666 J=1,JP1
  666 DS(J,K)=DH(I,J,K)
      CALL PRESU(PR,DS,M(N10),M(N11),M(N12),1,IP1,JP1,KP1)
      DO 660 J=1,JP1
      PR(I,J,1)=PR(I,J,KMM)
  660 PR(I,J,KP1)=PR(I,J,KM1)
      END IF
  661 CONTINUE
      DO 664 I=2,IMAX
      JA=2
      IF(I .GE. M3) JA=JOPT
      JAM1=JA-1
      DO 663 K=1,KP1
      PR(I,JAM1,K)=((Y4SQ(I,K)-Y2SQ(I,K))*PR(I,JA,K)-
     $(Y3SQ(I,K)-Y2SQ(I,K))*PR(I,JA+1,K))/(Y4SQ(I,K)-Y3SQ(I,K)
  663 CONTINUE
  664 CONTINUE
      DO 4001 J=1,JP1
      DO 4001 K=1,KP1
 4001 PR(IP1,J,K)=PR(IMAX,J,K)
 3999 CONTINUE
      DO 4003 I=M2,IP1
      DO 4004 K=1,KP1
      DO 4004 J=1,JP1
 4004 PR(I,J,K)=PP(I,J,K)+RFP*(PR(I,J,K)-PP(I,J,K))
 4003 CONTINUE
      END IF
      RETURN
      END
C*************************************************
C
C.... SUBROUTINE FACST
C
C*************************************************
C
      SUBROUTINE FACST(PHI,YP,ZP,F2,F3,ZUT,UB,UP,UF,
     $                 VB,VP,VF,WB,WP,WF,AKEB,AKEP,ADSP,
     $                 GE,D1,E1,H1,SU,
     $                 UMM,UMN,UMP,UNM,UNN,UNP,UPM,UPN,UPP,
     $                 XX,YY,AA,BB,CC,DD,II,JJ,KK,M,NA23,REI,T
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION YP(II,JJ,KK)
      DIMENSION ZP(II,JJ,KK)
      DIMENSION F2(II,JJ,KK),F3(KK)
      DIMENSION ZUT(II,JJ,KK)
      DIMENSION PHI(II,JJ,KK)
```

```
      DIMENSION UB(JJ,KK),UP(JJ,KK),UF(JJ,KK)
      DIMENSION VB(JJ,KK),VP(JJ,KK),VF(JJ,KK)
      DIMENSION WB(JJ,KK),WP(JJ,KK),WF(JJ,KK)
      DIMENSION AKEB(JJ,KK),AKEP(JJ,KK),ADSP(JJ,KK)
      DIMENSION D1(JJ,KK),E1(JJ,KK),SU(JJ,KK),H1(JJ,KK)
      DIMENSION UMM(JJ,KK),UMN(JJ,KK),UMP(JJ,KK)
      DIMENSION UNM(JJ,KK),UNN(JJ,KK),UNP(JJ,KK)
      DIMENSION UPM(JJ,KK),UPN(JJ,KK),UPP(JJ,KK)
      DIMENSION GE(JJ,KK),XX(JJ,KK),YY(JJ,KK)
      DIMENSION AA(JJ,KK),BB(JJ,KK),CC(JJ,KK)
      DIMENSION DD(II,JJ,KK)
      COMMON/UVW1/IMAX,JMAX,KMAX,JPP,KPP,JA,JAM1,I,KMP,KMM
      COMMON/UVW2/A1,A2,A3,A4,UU,VV,WW
      COMMON/UVW3/REFF,ZUTP,ZUTXI,ZUTET,ZUTZT,FX1,FY2,FZ3
      COMMON/UVW4/AR,BR,DR,ER,FR,GR
      COMMON/UVW5/G11,G22,G33,A11,A22,A33,G,AJI
      COMMON/UVW6/B11,B12.B13,B22,B23,B32,B33,SG
      COMMON/UVW7/R,XXI,YXI,YET,YZT,RZXI,RZET,RZZT
      COMMON/COEF7/ CF(3,3)
C
      FACT=1.
      IF(M .GE. 4) FACT=30.
      IP1=I+1
      IM1=I-1
      DO 200 K=2,KMAX
      IF(M.EQ.3.AND.K.EQ.KMAX) RETURN
      FZ3=F3(K)
      KP1=K+1
      KM1=K-1
      IF(M.EQ.3) FZ3=0.5*(F3(K)+F3(KP1))
      DO 190 J=JAM1,JMAX
      IF(J.EQ.JAM1 .AND. M.NE.2) GO TO 190
      JP1=J+1
      JM1=J-1
      IF(M.EQ.1) THEN
C---- CALCULATE THE FINITE-ANALYTIC COEFFICIENTS AND
C---- SOURCE FUNCTION FOR LONGITUDINAL MOMENTUM EQUATION
      FY2=0.5*(F2(I,J,K)+F2(IP1,J,K))
      R=0.5*(YP(I,J,K)+YP(IP1,J,K))
      YXI=YP(IP1,J,K)-YP(I,J,K)
      YET=0.25*(YP(IP1,JP1,K)-YP(IP1,JM1,K)
     $+YP(I,JP1,K)-YP(I,JM1,K))
      YZT=0.25*(YP(IP1,J,KP1)-YP(IP1,J,KM1)
     $+YP(I,J,KP1)-YP(I,J,KM1))
      ZXI=ZP(IP1,J,K)-ZP(I,J,K)
      ZET=0.25*(ZP(IP1,JP1,K)-ZP(IP1,JM1,K)
     $+ZP(I,JP1,K)-ZP(I,JM1,K))
      ZZT=0.25*(ZP(IP1,J,KP1)-ZP(IP1,J,KM1)
     $+ZP(I,J,KP1)-ZP(I,J,KM1))
      ZUTP=0.5*(ZUT(IP1,J,K)+ZUT(I,J,K))
      ZUTXI=ZUT(IP1,J,K)-ZUT(I,J,K)
```

```
      ZUTET=0.25*(ZUT(IP1,JP1,K)-ZUT(IP1,JM1,K)
     $+ZUT(I,JP1,K)-ZUT(I,JM1,K))
      ZUTZT=0.25*(ZUT(IP1,J,KP1)-ZUT(IP1,J,KM1)
     $+ZUT(I,J,KP1)-ZUT(I,J,KM1))
      UU=UP(J,K)
      VV=0.25*(VP(JM1,K)+VB(JM1,K)
     $+VP(J,K)+VB(J,K))
      WW=0.25*(WP(J,KM1)+WB(J,KM1)
     $+WP(J,K)+WB(J,K))
      ELSE IF(M.EQ.2) THEN
C---- CALCULATE THE FINITE-ANALYTIC COEFFICIENTS AND
C---- SOURCE FUNCTION FOR RADIAL MOMENTUM EQUATION
      FY2=0.5*(F2(I,JP1,K)+F2(I,J,K))
      R=0.5*(YP(I,JP1,K)+YP(I,J,K))
      YXI=0.25*(YP(IP1,JP1,K)-YP(IM1,JP1,K)
     $+YP(IP1,J,K)-YP(IM1,J,K))
      YET=YP(I,JP1,K)-YP(I,J,K)
      YZT=0.25*(YP(I,JP1,KP1)-YP(I,JP1,KM1)
     $+YP(I,J,KP1)-YP(I,J,KM1))
      ZXI=0.25*(ZP(IP1,JP1,K)-ZP(IM1,JP1,K)
     $+ZP(IP1,J,K)-ZP(IM1,J,K))
      ZET=ZP(I,JP1,K)-ZP(I,J,K)
      ZZT=0.25*(ZP(I,JP1,KP1)-ZP(I,JP1,KM1)
     $+ZP(I,J,KP1)-ZP(I,J,KM1))
      ZUTP=0.5*(ZUT(I,JP1,K)+ZUT(I,J,K))
      ZUTXI=0.25*(ZUT(IP1,JP1,K)-ZUT(IM1,JP1,K)
     $+ZUT(IP1,J,K)-ZUT(IM1,J,K))
      ZUTET=ZUT(I,JP1,K)-ZUT(I,J,K)
      ZUTZT=0.25*(ZUT(I,JP1,KP1)-ZUT(I,JP1,KM1)
     $+ZUT(I,J,KP1)-ZUT(I,J,KM1))
      UU=0.25*(UP(J,K)+UP(JP1,K)
     $+UP(J,K)+UP(JP1,K))
      VV=VP(J,K)
      WW=0.25*(WP(J,KM1)+WP(JP1,KM1)
     $+WP(J,K)+WP(JP1,K))
      ELSE IF(M.EQ.3) THEN
C---- CALCULATE THE FINITE-ANALYTIC COEFFICIENTS AND
C---- SOURCE FUNCTION FOR CIRCUMFERENTIAL MOMENTUM EQUATION
      FY2=0.5*(F2(I,J,K)+F2(I,J,KP1))
      R=0.5*(YP(I,J,KP1)+YP(I,J,K))
      YXI=0.25*(YP(IP1,J,KP1)-YP(IM1,J,KP1)
     $+YP(IP1,J,K)-YP(IM1,J,K))
      YET=0.25*(YP(I,JP1,KP1)-YP(I,JM1,KP1)
     $+YP(I,JP1,K)-YP(I,JM1,K))
      YZT=YP(I,J,KP1)-YP(I,J,K)
      ZXI=0.25*(ZP(IP1,J,KP1)-ZP(IM1,J,KP1)
     $+ZP(IP1,J,K)-ZP(IM1,J,K))
      ZET=0.25*(ZP(I,JP1,KP1)-ZP(I,JM1,KP1)
     $+ZP(I,JP1,K)-ZP(I,JM1,K))
      ZZT=ZP(I,J,KP1)-ZP(I,J,K)
      ZUTP=0.5*(ZUT(I,J,KP1)+ZUT(I,J,K))
```

```
      ZUTXI=0.25*(ZUT(IP1,J,KP1)-ZUT(IM1,J,KP1)
     $+ZUT(IP1,J,K)-ZUT(IM1,J,K))
      ZUTET=0.25*(ZUT(I,JP1,KP1)-ZUT(I,JM1,KP1)
     $+ZUT(I,JP1,K)-ZUT(I,JM1,K))
      ZUTZT=ZUT(I,J,KP1)-ZUT(I,J,K)
      UU=0.25*(UF(J,KP1)+UF(J,K)
     $+UP(J,KP1)+UP(J,K))
      VV=0.25*(VP(JM1,KP1)+VP(JM1,K)
     $+VP(J,KP1)+VP(J,K))
      WW=WP(J,K)
      ELSE IF(M.GE.4) THEN
C---- CALCULATE THE FINITE-ANALYTIC COEFFICIENTS AND
C---- SOURCE FUNCTIONS FOR TURBULENCE QUANTITIES
      FY2=F2(I,J,K)
      R=YP(I,J,K)
      YXI=0.5*(YP(IP1,J,K)-YP(IM1,J,K))
      YET=0.5*(YP(I,JP1,K)-YP(I,JM1,K))
      YZT=0.5*(YP(I,J,KP1)-YP(I,J,KM1))
      ZXI=0.5*(ZP(IP1,J,K)-ZP(IM1,J,K))
      ZET=0.5*(ZP(I,JP1,K)-ZP(I,JM1,K))
      ZZT=0.5*(ZP(I,J,KP1)-ZP(I,J,KM1))
      ZUTP=ZUT(I,J,K)
      ZUTXI=0.5*(ZUT(IP1,J,K)-ZUT(IM1,J,K))
      ZUTET=0.5*(ZUT(I,JP1,K)-ZUT(I,JM1,K))
      ZUTZT=0.5*(ZUT(I,J,KP1)-ZUT(I,J,KM1))
      UU=0.5*(UF(J,K)+UP(J,K))
      VV=0.5*(VP(JM1,K)+VP(J,K))
      WW=0.5*(WP(J,KM1)+WP(J,K))
      END IF
C
      IF(NA23 .EQ. 2) R=1.
      RZXI=R*ZXI
      RZET=R*ZET
      RZZT=R*ZZT
      REFF=1./(REI+ZUTP)
      CALL EQCOE(PHI,UB,UP,UF,VB,VP,VF,WB,WP,WF,AKEB,
     $           AKEP,ADSP,GE,II,JJ,KK,M,I,J,K,NA23)
      DDD=DABS(DR)
      D1(J,K)=DR
      E1(J,K)=A4*REFF*TAUI*FACT
      H1(J,K)=GR
      SU(J,K)=SG
      IF(J .EQ. JAM1) GO TO 180
      CALL COEF
      UMM(J,K)=CF(1,1)
      UMN(J,K)=CF(1,2)
      UMP(J,K)=CF(1,3)
      UNM(J,K)=CF(2,1)
      UNN(J,K)=CF(2,2)
      UNP(J,K)=CF(2,3)
      UPM(J,K)=CF(3,1)
```

```
      UPN(J,K)=CF(3,2)
      UPP(J,K)=CF(3,3)
      D=AJI*REFF*UNN(J,K)/(1.+(DDD+2*H1(J,K)+E1(J,K))*UNN(J,K)
  180 IF(M.EQ.1) THEN
      XX(J,K)=YY(J,K)
      YY(J,K)=B11
      AA(J,K)=D*B11
      BB(J,K)=D*B12
      CC(J,K)=D*B13
      ELSE IF(M.EQ.2) THEN
      XX(J,K)=B12*UU+B32*WW
      YY(J,K)=B22
      IF(J .EQ. JAMJAM1) GO TO 190
      AA(J,K)=D*B22
      BB(J,K)=D*B23
      ELSE IF(M.EQ.3) THEN
      XX(J,K)=B13*UU+B23*VV
      YY(J,K)=B33
      AA(J,K)=D*B33
      BB(J,K)=D*B32
      ELSE
      GO TO 190
      END IF
      DD(I,J,K)=YY(J,K)*AA(J,K)
  190 CONTINUE
      IF(M.EQ.2) YY(JPP,K)=(1.5*YP(I,JPP,K)-0.5*YP(I,JMAX,K))*
     $XXI*0.5*(ZP(I,JPP,KP1)-ZP(I,JPP,KM1))
  200 CONTINUE
      IF(I.EQ.2 .AND. M.EQ.1) THEN
      DO 201 J=1,JPP
      DO 201 K=1,KPP
      DD(1,J,K)=DD(2,J,K)
  201 XX(J,K)=YY(J,K)
      END IF
      IF(M.EQ.3) THEN
      DO 202 J=1,JPP
      YY(J,KMAX)=YY(J,KMP)
      YY(J,1)=YY(J,2)
      AA(J,1)=AA(J,2)
      AA(J,KMAX)=AA(J,KMP)
      DD(I,J,1)=DD(I,J,2)
      DD(I,J,KMAX)=DD(I,J,KMP)
      XX(J,1)=-XX(J,2)
      XX(J,KMAX)=-XX(J,KMP)
  202 CONTINUE
      END IF
      RETURN
      END
C     ****************************************************
C     SUBROUTINE EQCOE: THE COEFFICIENTS OF FA EQUATION
C     ****************************************************
```

```
      SUBROUTINE EQCOE(PHI,UB,UP,UF,VB,VP,VF,WB,WP,WF,
     $                  AKEB,AKEP,ADSP,GE,
     $                  II,JJ,KK,M,I,J,K,NA23)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION PHI(II,JJ,KK)
      DIMENSION UB(JJ,KK),UP(JJ,KK),UF(JJ,KK)
      DIMENSION VB(JJ,KK),VP(JJ,KK),VF(JJ,KK)
      DIMENSION WB(JJ,KK),WP(JJ,KK),WF(JJ,KK)
      DIMENSION AKEB(JJ,KK),AKEP(JJ,KK),ADSP(JJ,KK)
      DIMENSION GE(JJ,KK)
      COMMON/UVW2/A1,A2,A3,A4,UU,VV,WW
      COMMON/UVW3/REFF,ZUTP,ZUTXI,ZUTET,ZUTZT,FX1,FY2,FZ3
      COMMON/UVW4/AR,BR,DR,ER,FR,GR
      COMMON/UVW5/G11,G22,G33,A11,A22,A33,G,AJI
      COMMON/UVW6/B11,B12,B13,B22,B23,B32,B33,SG
      COMMON/UVW7/R,XXI,YXI,YET,YZT,RZXI,RZET,RZZT
      COMMON/UVW8/C1,C2,CEFFK,CEFFD
C
      B11=YET*RZZT-YZT*RZET
      B12=YZT*RZXI-YXI*RZZT
      B13=YXI*RZET-YET*RZXI
      B22=XXI*RZZT
      B23=-XXI*RZET
      B32=-XXI*YZT
      B33=XXI*YET
      G11=XXI*XXI+YXI*YXI+RZXI*RZXI
      G22=YET*YET+RZET*RZET
      G33=YZT*YZT+RZZT*RZZT
      G12=YXI*YET+RZXI*RZET
      G13=YXI*YZT+RZXI*RZZT
      G23=YET*YZT+RZET*RZZT
      G=G11*G22*G33+2.*G12*G13*G23-G23*G23*G11-
     $G13*G13*G22-G12*G12*G33
      GI=1./G
      A11=GI*(G22*G33-G23*G23)
      A22=GI*(G11*G33-G13*G13)
      A33=GI*(G11*G22-G12*G12)
      A12=GI*(G13*G23-G12*G33)
      A13=GI*(G12*G23-G13*G22)
      A23=GI*(G12*G13-G23*G11)
      AJI=DSQRT(GI)
      FX=-2.*A11*FX1
      FY=-2.*A22*FY2
      IF(NA23 .NE. 2) FY=FY+1./R/YET
      FZ=-2.*A33*FZ3
      ZUT1=AJI*(B11*ZUTXI+B12*ZUTET+B13*ZUTZT)
      ZUT2=AJI*(B22*ZUTET+B23*ZUTZT)
      ZUT3=AJI*(B32*ZUTET+B33*ZUTZT)
      AP1=A4*UU-A1*ZUT1
      AP2=A4*VV-A2*ZUT2
      AP3=A4*WW-A3*ZUT3
```

```
      AR=0.5*(REFF*AJI*(B13*AP1+B23*AP2+
     $B33*AP3)-FZ)
      BR=0.5*(REFF*AJI*(B12*AP1+B22*AP2+
     $B32*AP3)-FY)
      DR=REFF*AJI*B11*AP1-FX
      ER=DSQRT(A33)
      FR=DSQRT(A22)
      GR=A11
C
      SS=-0.5*A12*(PHI(I+1,J+1,K)+PHI(I-1,J-1,K)-PHI(I+1,J-1,K
     $-PHI(I-1,J+1,K))-0.5*A13*(PHI(I+1,J,K+1)+PHI(I-1,J,K-1)
     $-PHI(I+1,J,K-1)-PHI(I-1,J,K+1))-0.5*A23*(PHI(I,J+1,K+1)
     $+PHI(I,J-1,K-1)-PHI(I,J+1,K-1)-PHI(I,J-1,K+1))
      IF(M .EQ. 1) THEN
      DKXI=AKEB(J,K)-AKEP(J,K)
      DKET=0.25*(AKEP(J+1,K)+AKEB(J+1,K)-AKEP(J-1,K)
     $-AKEB(J-1,K))
      DKZT=0.25*(AKEP(J,K+1)+AKEB(J,K+1)-AKEP(J,K-1)
     $-AKEB(J,K-1))
      DVXI=0.5*(VB(J,K)+VB(J-1,K)-VP(J,K)-VP(J-1,K))
      DVET=0.5*(VB(J,K)+VP(J,K)-VB(J-1,K)-VP(J-1,K))
      DVZT=0.125*(VB(J,K+1)+VB(J-1,K+1)+VP(J,K+1)
     $+VP(J-1,K+1)-VB(J,K-1)-VB(J-1,K+1)-VP(J,K-1)
     $-VP(J-1,K+1))
      DWXI=0.5*(WB(J,K)+WB(J,K-1)-WP(J,K)-WP(J,K-1))
      DWET=0.125*(WB(J+1,K)+WB(J+1,K-1)+WP(J+1,K)
     $+WP(J+1,K-1)-WB(J-1,K)-WB(J-1,K-1)-WP(J-1,K)
     $-WP(J-1,K-1))
      DWZT=0.5*(WB(J,K)+WP(J,K)-WB(J,K-1)-WP(J,K-1))
      DKX=AJI*(B11*DKXI+B12*DKET+B13*DKZT)
      DVX=AJI*(B11*DVXI+B12*DVET+B13*DVZT)
      DWX=AJI*(B11*DWXI+B12*DWET+B13*DWZT)
      SG=SS+REFF*(2./3.*DKX-ZUT2*DVX-ZUT3*DWX)
      ELSE IF(M .EQ. 2) THEN
      DKET=AKEP(J+1,K)-AKEP(J,K)
      DKZT=0.25*(AKEP(J+1,K+1)+AKEP(J,K+1)-AKEP(J+1,K-1)
     $-AKEP(J,K-1))
      DUET=0.5*(UP(J+1,K)+UF(J+1,K)-UP(J,K)-UF(J,K))
      DUZT=0.125*(UP(J+1,K+1)+UP(J,K+1)+UF(J+1,K+1)
     $+UF(J,K+1)-UP(J+1,K-1)-UP(J,K-1)-UF(J+1,K-1)
     $-UF(J,K-1))
      DWET=0.5*(WP(J+1,K)-WP(J,K)+WP(J+1,K-1)-WP(J,K-1))
      DWZT=0.5*(WP(J,K)+WP(J+1,K)-WP(J,K-1)-WP(J+1,K-1))
      DKY=AJI*(B22*DKET+B23*DKZT)
      DUY=AJI*(B22*DUET+B23*DUZT)
      DWY=AJI*(B22*DWET+B23*DWZT)
      DWZ=AJI*(B23*DWET+B33*DWZT)
      SG=SS+REFF*(2./3.*DKY-ZUT1*DUY-ZUT3*DWY)
      IF(NA23 .NE. 2) SG=SG+REFF*(ZUT3*WW/R-
     $WW*WW/R)+2./R*DWZ+VV/R/R
      ELSE IF(M .EQ. 3) THEN
```

```
      DKET=0.25*(AKEP(J+1,K+1)-AKEP(J-1,K+1)+AKEP(J+1,K)
     $-AKEP(J-1,K))
      DKZT=AKEP(J,K+1)-AKEP(J,K)
      DUET=0.125*(UB(J+1,K)+UB(J+1,K-1)+UP(J+1,K)
     $+UP(J+1,K-1)-UB(J-1,K)-UB(J-1,K-1)-UP(J-1,K)
     $-UP(J-1,K-1))
      DUZT=0.5*(UP(J,K+1)+UF(J,K+1)-UP(J,K)-UF(J,K))
      DVET=0.5*(VP(J,K+1)-VP(J-1,K+1)+VP(J,K)-VP(J-1,K))
      DVZT=0.5*(VP(J-1,K+1)-VP(J-1,K)+VP(J,K+1)-VP(J,K))
      DKZ=AJI*(B32*DKET+B33*DKZT)
      DUZ=AJI*(B32*DUET+B33*DUZT)
      DVZ=AJI*(B32*DVET+B33*DVZT)
      SG=SS+REFF*(2./3.*DKZ+WW*VV/R-ZUT1*DUZ-ZUT2*(DVZ-
     $WW/R)-ZUT3*2.*VV/R)-2./R*DVZ+WW/R/R
      ELSE IF(M .EQ. 4) THEN
      DUXI=UP(J,K)-UF(J,K)
      DUET=0.25*(UP(J+1,K)+UF(J+1,K)-UP(J-1,K)-UF(J-1,K))
      DUZT=0.25*(UP(J,K+1)+UF(J,K+1)-UP(J,K-1)-UF(J,K-1))
      DVXI=0.5*(VP(J,K)+VP(J-1,K)-VF(J,K)-VF(J-1,K))
      DVET=VP(J,K)-VP(J-1,K)
      DVZT=0.25*(VP(J,K+1)+VP(J-1,K+1)-VP(J,K-1)-VP(J-1,K-1))
      DWXI=0.5*(WP(J,K)+WP(J,K-1)-WF(J,K)-WF(J,K-1))
      DWET=0.25*(WP(J+1,K)+WP(J+1,K-1)-WP(J-1,K)-WP(J-1,K-1))
      DWZT=WP(J,K)-WP(J,K-1)
      DUX=AJI*(B11*DUXI+B12*DUET+B13*DUZT)
      DVX=AJI*(B11*DVXI+B12*DVET+B13*DVZT)
      DWX=AJI*(B11*DWXI+B12*DWET+B13*DWZT)
      DUY=AJI*(B22*DUET+B23*DUZT)
      DVY=AJI*(B22*DVET+B23*DVZT)
      DWY=AJI*(B22*DWET+B23*DWZT)
      DUZ=AJI*(B32*DUET+B33*DUZT)
      DVZ=AJI*(B32*DVET+B33*DVZT)
      DWZ=AJI*(B32*DWET+B33*DWZT)
      GE(J,K)=ZUTP*(2.*DUX*DUX+2.*DVY*DVY+2.*DWZ*DWZ
     $+(DUZ+DWX)**2+(DVX+DUY)**2+(DVZ+DWY)**2)
      IF(NA23 .NE. 2) GE(J,K)=GE(J,K)+ZUTP*(2.*(DWZ+VV/R)**2-
     $2.*DWZ*DWZ+(DVZ+DWY-WW/R)**2-(DVZ+DWY)**2)
      SG=SS-CEFFK*REFF*(GE(J,K)-ADSP(J,K))
      ELSE IF(M .EQ. 5) THEN
      SG=SS-CEFFD*REFF*(C1*GE(J,K)*ADSP(J,K)-
     $C2*ADSP(J,K)*ADSP(J,K))/AKEP(J,K)
      END IF
      RETURN
      END
C     ****************************************************
C     SUBROUTINE SVEL IS USED TO SOLVE FA ALGEBRAIC EQUATION
C     ****************************************************
C
      SUBROUTINE SVEL(US,GG,UO,D1,E1,H1,SU,
     $                UMM,UMN,UMP,UNM,UNN,UNP,
     $                UPM,UPN,UPP,Y2,Y3,Y4,BV,
```

```
     $                  II,JJ,KK,ITUVW,M,L,N)
      IMPLICIT REAL*8(A-H,O-Z)
C
      DIMENSION UO(II,JJ,KK)
      DIMENSION US(JJ,KK),GG(JJ,KK)
      DIMENSION D1(JJ,KK),E1(JJ,KK),H1(JJ,KK),SU(JJ,KK)
      DIMENSION UMM(JJ,KK),UMN(JJ,KK),UMP(JJ,KK)
      DIMENSION UNM(JJ,KK),UNN(JJ,KK),UNP(JJ,KK)
      DIMENSION UPM(JJ,KK),UPN(JJ,KK),UPP(JJ,KK)
      DIMENSION BV(JJ,KK),Y2(II,KK),Y3(II,KK),Y4(II,KK)
      DIMENSION AA(99),BB(99),CC(99),DD(99),T(99)
      COMMON/UVW1/IMAX,JMAX,KMAX,JP1,KP1,JA,JAM1,I,KM1,KMM
      COMMON/COEF4/ UI,VI,M1,M2,M3
C
C.... CALCULATE THE LONGITUDIAL VELOCITY FIELD
C
      DO 305 ITER=1,ITUVW
      DO 330 K=2,KMAX
      IF(M.EQ.2 .AND. K.EQ.KMAX) GO TO 330
      DO 320 J=JA,JMAX
      DDD=DABS(D1(J,K))
      UU=UO(I-1,J,K)
      IF(D1(J,K).LT.0.) UU=UO(I+1,J,K)
      AA(J)=-UMN(J,K)
      BB(J)=1.+(DDD+2*H1(J,K)+E1(J,K))*UNN(J,K)
      CC(J)=-UPN(J,K)
  320 DD(J)=UNP(J,K)*US(J,K+1)+UNM(J,K)*US(J,K-1)
     S+UPP(J,K)*US(J+1,K+1)+UPM(J,K)*US(J+1,K-1)
     S+UMP(J,K)*US(J-1,K+1)+UMM(J,K)*US(J-1,K-1)
     S+UNN(J,K)*(DDD*UU+E1(J,K)*UO(I,J,K)-SU(J,K)
     S+H1(J,K)*(UO(I+1,J,K)+UO(I-1,J,K)))
     S-BB(J)*GG(J,K)
      DD(JA)=DD(JA)-AA(JA)*US(JAM1,K)
      DD(JMAX)=DD(JMAX)-CC(JMAX)*US(JP1,K)
      CALL TRIDAG(JA,JMAX,AA,BB,CC,DD,T)
      DO 310 J=JA,JMAX
  310 US(J,K)=T(J)
  330 CONTINUE
      DO 333 K=2,KMAX
      IF(L .EQ. 1) THEN
      US(JP1,K)=UI
      ELSE IF(L .EQ. 2) THEN
      US(JP1,K)=US(JMAX,K)*BV(JMAX,K)/BV(JP1,K)
      ELSE IF(L .EQ. 3) THEN
      US(JP1,K)=US(JMAX,K)
      END IF
  333 CONTINUE
      IF(I.LT.M1 .OR. I.GE.M2) THEN
      DO 334 K=2,KMAX
      IF(N .EQ. 1) THEN
      US(JAM1,K)=((Y4(I,K)-Y2(I,K))*US(JA,K)-(Y3(I,K)-Y2(I,K))
```

```
     $            *US(JA+1,K))/(Y4(I,K)-Y3(I,K))
      ELSE IF(N .EQ. 2) THEN
      US(JAM1,K)=US(JA,K)*Y2(I,K)/Y3(I,K)
      ELSE IF(N .EQ. 3) THEN
      US(JAM1,K)=0.
      END IF
  334 CONTINUE
      END IF
      IF(M .EQ. 1) THEN
      DO 331 J=1,JP1
      US(J,1)=US(J,KMM)
  331 US(J,KP1)=US(J,KM1)
      ELSE IF(M .EQ. 2) THEN
      DO 332 J=1,JP1
      US(J,1)=-US(J,2)
  332 US(J,KMAX)=-US(J,KM1)
      END IF
  305 CONTINUE
      RETURN
      END
C     *****************************************************
C     CALCULATE THE PSEUDO-VELOCITY FIELD
C     *****************************************************
C
      SUBROUTINE HVEL(UH,GG,UO,UP,D1,E1,H1,SU,
     $                 UMM,UMN,UMP,UNM,UNN,UNP,
     $                 UPM,UPN,UPP,II,JJ,KK,M)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION UO(II,JJ,KK)
      DIMENSION UP(JJ,KK),UH(JJ,KK),SU(JJ,KK)
      DIMENSION GG(JJ,KK),D1(JJ,KK),E1(JJ,KK),H1(JJ,KK)
      DIMENSION UMM(JJ,KK),UMN(JJ,KK),UMP(JJ,KK)
      DIMENSION UNM(JJ,KK),UNN(JJ,KK),UNP(JJ,KK)
      DIMENSION UPM(JJ,KK),UPN(JJ,KK),UPP(JJ,KK)
      COMMON/UVW1/IMAX,JMAX,KMAX,JP1,KP1,JA,JAM1,I,KM1,KMM
C
      DO 672 K=2,KMAX
      IF(M.EQ.2 .AND. K.EQ.KMAX) RETURN
      DO 672 J=JA,JMAX
      DDD=DABS(D1(J,K))
      UU=UO(I-1,J,K)
      IF(D1(J,K) .LT. 0.) UU=UO(I+1,J,K)
  672 UH(J,K)=(UMN(J,K)*UP(J-1,K)+UPN(J,K)*UP(J+1,K)
     $+UNP(J,K)*UP(J,K+1)+UNM(J,K)*UP(J,K-1)
     $+UPP(J,K)*UP(J+1,K+1)+UPM(J,K)*UP(J+1,K-1)
     $+UMP(J,K)*UP(J-1,K+1)+UMM(J,K)*UP(J-1,K-1)
     $+UNN(J,K)*(DDD*UU+H1(J,K)*(UO(I-1,J,K)+UO(I+1,J,K))
     $+E1(J,K)*UO(I,J,K)-SU(J,K)))/(1.+(DDD+2*H1(J,K)
     $+E1(J,K))*UNN(J,K))-GG(J,K)
      RETURN
      END
```

```
C
C     ***************************************************
C     SUBROUTINE PRESSURE EQUATION
C     ***************************************************
C
      SUBROUTINE PRESU(P,DHS,BCU,BCV,BCW,M,II,JJ,KK)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION P(II,JJ,KK)
      DIMENSION DHS(JJ,KK)
      DIMENSION BCU(II,JJ,KK)
      DIMENSION BCV(II,JJ,KK)
      DIMENSION BCW(II,JJ,KK)
      DIMENSION AA(99),BB(99),CC(99),DD(99),T(99)
      COMMON/UVW1/IMAX,JMAX,KMAX,JP1,KP1,JA,JAM1,I,KM1,KMM
      DO 655 K=2,KMAX
      DO 660 J=JA,JMAX
      AA(J)=-BCV(I,J-1,K)
      BB(J)=BCU(I-1,J,K)*M+BCU(I,J,K)+BCV(I,J-1,K)
     $+BCV(I,J,K)+BCW(I,J,K-1)+BCW(I,J,K)
      CC(J)=-BCV(I,J,K)
  660 DD(J)=BCU(I,J,K)*P(I+1,J,K)+BCU(I-1,J,K)*P(I-1,J,K)
     $+BCW(I,J,K)*P(I,J,K+1)+BCW(I,J,K-1)*P(I,J,K-1)-DHS(J,K)
      DD(JA)=DD(JA)-AA(JA)*P(I,JAM1,K)
      DD(JMAX)=DD(JMAX)-CC(JMAX)*P(I,JP1,K)
      CALL TRIDAG(JA,JMAX,AA,BB,CC,DD,T)
      DO 671 J=JA,JMAX
  671 P(I,J,K)=T(J)
  655 CONTINUE
      RETURN
      END
C
C*****************************************************************
C     SUBROUTINE COEF IS USED TO CALCULATE THE FA COEFFICIENTS
C*****************************************************************
C
      SUBROUTINE COEF
      IMPLICIT REAL*8(A-H,O-Z)
C
      COMMON/COEF7/CF(3,3)
      COMMON/UVW4/AR,BR,DR,ER,FR,GR
      DATA MAX,EMAX,C1,EPE,PI /12,20.,1.,1.D-5,3.1415926535897
C
C
      HX=1./ER
      HY=1./FR
      AR=AR/ER
      BR=BR/FR
      IF(DABS(AR).LT.EPE)AR=DSIGN(EPE,AR)
      IF(DABS(BR).LT.EPE)BR=DSIGN(EPE,BR)
      ER2=ER*ER
      FR2=FR*FR
```

```
      AB2=AR*AR+BR*BR
      AH=AR*HX
      AK=AR*HY
      BH=BR*HX
      BK=BR*HY
      DAH=DABS(AH)
      DBK=DABS(BK)
      AH2=AH*AH
      BK2=BK*BK
C
      IM=0
      JM=0
      IF(DAH.GT.EMAX) IM=1
      IF(DBK.GT.EMAX) JM=2
      MT=IM+JM+1
      GO TO (1,2,3,4), MT
C
    1 EPAH=DEXP(AH)
      EPBK=DEXP(BK)
      EPAHI=1./EPAH
      EPBKI=1./EPBK
      COSHA=0.5*(EPAH+EPAHI)
      COSHB=0.5*(EPBK+EPBKI)
      COTHA=2.*COSHA/(EPAH-EPAHI)
      COTHB=2.*COSHB/(EPBK-EPBKI)
      AKCTHA=AK*COTHA
      BHCTHB=BH*COTHB
      PWR=1.
      IF(HX .GT. HY) GO TO 11
      EX2=0.
      DO 10 II=1,MAX
      ZA=(II-0.5)*PI
      ZA2=ZA*ZA
      PWR=-PWR
      DABK=DSQRT(AB2+ZA2*ER2)*HY
      IF(DABK .GT. 100.) GO TO 9
      AB=DEXP(DABK)
   10 EX2=EX2-PWR*ZA/((AB+1./AB)*(AH2+ZA2)*(AH2+ZA2))
    9 PA=8.*AH*COTHA*COSHA*COSHB*EX2
      PB=1.+BHCTHB/AKCTHA*(PA-1.)
      CF(2,2)=0.5*HX/(AR*COTHA)*(1.-PA)
      GO TO 100
C
   11 EY2=0.
      DO 12 II=1,MAX
      ZA=(II-0.5)*PI
      ZA2=ZA*ZA
      PWR=-PWR
      DABH=DSQRT(AB2+ZA2*FR2)*HX
      IF(DABH.GT.100.) GO TO 19
      AB=DEXP(DABH)
```

```
   12 EY2=EY2-PWR*ZA/((AB+1./AB)*(BK2+ZA2)*(BK2+ZA2))
   19 PB=8.*BK*COTHB*COSHA*COSHB*EY2
      PA=1.+AKCTHA/BHCTHB*(PB-1.)
      CF(2,2)=0.5*HY/(BR*COTHB)*(1.-PB)
      GO TO 100
C
    2 EPBK=DEXP(BK)
      EPBKI=1./EPBK
      COSHB=0.5*(EPBK+EPBKI)
      COTHB=2.*COSHB/(EPBK-EPBKI)
      COTHA=DSIGN(C1,AR)
      AKCTHA=AK*COTHA
      BHCTHB=BH*COTHB
      PWR=1.
      IF(AKCTHA.LT.BHCTHB) GO TO 22
      EX2=0.
      FX2=0.
      DO 20 II=1,MAX
      ZA=(II-0.5)*PI
      ZA2=ZA*ZA
      PWR=-PWR
      PZ=PWR*ZA/((AH2+ZA2)*(AH2+ZA2))
      FX2=FX2-PZ
      DABK=DSQRT(AB2+ZA2*ER2)*HY
      AB=1.
      IF(DABK.GT.100.) GO TO 20
      EPABK=DEXP(DABK)
      AB=1.-COSHB/(EPABK+1./EPABK)
   20 EX2=EX2-PZ*AB
      PA=1.-EX2/FX2
      PB=1.+BHCTHB/AKCTHA*(PA-1.)
      CF(2,2)=0.5*HY/(BR*COTHB)*(1.-PB)
      GO TO 100
   22 EY2=0.
      DO 23 II=1,MAX
      ZA=(II-0.5)*PI
      ZA2=ZA*ZA
      PWR=-PWR
      DABH=DAH-DSQRT(AB2+ZA2*FR2)*HX
      IF(DABS(DABH).GT.100.) GO TO 29
      AB=DEXP(DABH)
   23 EY2=EY2-PWR*ZA*AB/((BK2+ZA2)*(BK2+ZA2))
   29 PB=4.*BK*COTHB*COSHB*EY2
      PA=1.+AKCTHA/BHCTHB*(PB-1.)
      CF(2,2)=0.5*HY/(BR*COTHB)*(1.-PB)
      GO TO 100
C
    3 EPAH=DEXP(AH)
      EPAHI=1./EPAH
      COSHA=0.5*(EPAH+EPAHI)
      COTHA=2.*COSHA/(EPAH-EPAHI)
```

```
      COTHB=DSIGN(C1,BR)
      AKCTHA=AK*COTHA
      BHCTHB=BH*COTHB
      PWR=1.
      IF(AKCTHA.GT.BHCTHB) GO TO 32
      EY2=0.
      FY2=0.
      DO 30 II=1,MAX
      ZA=(II-0.5)*PI
      ZA2=ZA*ZA
      PWR=-PWR
      PZ=PWR*ZA/((BK2+ZA2)*(BK2+ZA2))
      FY2=FY2-PZ
      DABH=DSQRT(AB2+ZA2*FR2)*HX
      AB=1.
      IF(DABH.GT.100.) GO TO 30
      EPABH=DEXP(DABH)
      AB=1.-COSHA/(EPABH+1./EPABH)
   30 EY2=EY2-PZ*AB
      PB=1.-EY2/FY2
      PA=1.+AKCTHA/BHCTHB*(PB-1.)
      CF(2,2)=0.5*HY/(BR*COTHB)*(1.-PB)
      GO TO 100
   32 EX2=0.
      DO 33 II=1,MAX
      ZA=(II-0.5)*PI
      ZA2=ZA*ZA
      PWR=-PWR
      DABK=DBK-DSQRT(AB2+ZA2*ER2)*HY
      IF(DABS(DABK).GT.100.) GO TO 39
      AB=DEXP(DABK)
   33 EX2=EX2-PWR*ZA*AB/((AH2+ZA2)*(AH2+ZA2))
   39 PA=4.*AH*COTHA*COSHA*EX2
      PB=1.+BHCTHB/AKCTHA*(PA-1.)
      CF(2,2)=0.5*HY/(BR*COTHB)*(1.-PB)
      GO TO 100
C
    4 DAK=DABS(AK)
      DBH=DABS(BH)
      COTHA=DSIGN(C1,AR)
      COTHB=DSIGN(C1,BR)
      IF(DAK.LT.DBH) GO TO 41
      PA=0.
      PB=1.-DBH/DAK
      CF(2,2)=0.5*HX/(AR*COTHA)
      GO TO 100
C
   41 PB=0.
      PA=1.-DAK/DBH
      CF(2,2)=0.5*HY/(BR*COTHB)
C
```

```
  100 Q=1.-PA-PB
      TANHA=1./COTHA
      TANHB=1./COTHB
      BE=0.5*(1.-TANHA)
      BW=0.5*(1.+TANHA)
      BN=0.5*(1.-TANHB)
      BS=0.5*(1.+TANHB)
      CF(1,1)=BW*BS*Q
      CF(1,3)=BE*BS*Q
      CF(3,1)=BW*BN*Q
      CF(3,3)=BE*BN*Q
      CF(1,2)=BS*PA
      CF(3,2)=BN*PA
      CF(2,1)=BW*PB
      CF(2,3)=BE*PB
C
      RETURN
      END
C
C**********************************************************
C     SUBROUTINE TRIDAG TO SOLVE ALGEBRAIC EQUATIONS
C     SIMULTANEOUSLY FOR EACH ROW OR COLOUM
C**********************************************************
C
C
      SUBROUTINE TRIDAG(IF,L,A,B,C,D,V)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(99),B(99),C(99),D(99),V(99),BETA(99),GAMMA(9
      BETA(IF)=B(IF)
      GAMMA(IF)=D(IF)/BETA(IF)
      IFP1=IF+1
      DO 1 I=IFP1,L
      BETA(I)=B(I)-A(I)*C(I-1)/BETA(I-1)
    1 GAMMA(I)=(D(I)-A(I)*GAMMA(I-1))/BETA(I)
      V(L)=GAMMA(L)
      LAST=L-IF
      DO 2 K=1,LAST
      I=L-K
    2 V(I)=GAMMA(I)-C(I)*V(I+1)/BETA(I)
      RETURN
      END

C**********************************************************
C
C.... THIS PROGRAM IS USED TO GENERATE THE BODY-FITTED
C.... COORDINATES ON THE AFTERBODY1
C
C**********************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/GEO1/ XP(62),YP(62,20,4)
      COMMON/GEO2/ ZP(62,20,4)
```

```
      COMMON/BODY2/ F1(62),F2(62,20),F3(4),GE(20,4)
      DIMENSION AA(90),BB(90),CC(90),DD(90),T(90)
      DIMENSION FA(20), FB(20)
C
      IMAX=62
      JMAX=20
      KMAX=1
      IMAM=IMAX-1
      JMAM=JMAX-1
      KMAM=KMAX-1
      EPE=1.D-5
      A1=-.05
      A2=0.2
      NA=15
      NB=42
      NX1=5
      NX2=40
      NX3=40
C
      PI=3.141592653589793D0
      EPS=1.D-12
C
C.... X-DIRECTION
C
      AX3=NX3
C
      DO 50 I=10,IMAX
      Z1=I/AX3
C
      IF(Z1 .LE. 0.5) F1(I)=A1
      IF(Z1 .GT. 0.5 .AND. Z1 .LE. 1.)GO TO 20
      IF(Z1 .GT. 1. .AND. Z1 .LE. 1.2) GO TO 30
      IF(Z1 .GT. 1.2) F1(I)=A2
      GO TO 50
   20 PIZ=PI*Z1
      F1(I)=A1*DSIN(PIZ)
      GO TO 50
   30 PIZ=PI*Z1
      F1(I)=-A2*DSIN(PIZ)
   50 CONTINUE
C
      DO 55 I=1,5
   55 F1(I)=-F1(NX2+I-5)
      DO 56 I=5,9
   56 F1(I)=-F1(10-I)
C
      DO 60 I=2,IMAM
      AA(I)=-DEXP(F1(I))
      CC(I)=1./AA(I)
      BB(I)=-(AA(I)+CC(I))
   60 DD(I)=0.D0
```

```
C
      XP(NX1)=0.4446D0
      XP(NX2)=1.D0
      N1=NX1+1
      N2=NX2-1
C
      DD(N1)=DD(N1)-AA(N1)*XP(NX1)
      DD(N2)=DD(N2)-CC(N2)*XP(NX2)
C
      CALL TRIDAG(N1,N2,AA,BB,CC,DD,T)
C
      DO 70 I=N1,N2
   70 XP(I)=T(I)
C
      DO 72 I=NX1,2,-1
      XP(I-1)=-(XP(I)*BB(I)+XP(I+1)*CC(I))/AA(I)
   72 CONTINUE
C
      DO 74 I=NX2,IMAM
      XP(I+1)=-(XP(I)*BB(I)+XP(I-1)*AA(I))/CC(I)
   74 CONTINUE
C
C.... Y-DIRECTION
C
C.... READ THE BODY SURFACE FROM THE MEASUREMENTS
C
      OPEN(UNIT=5,FILE='AFTERBODY1')
      READ(5,300)(YP(I,1,1),I=1,IMAX)
      CLOSE(5)
C
      DO 150 I=1,IMAX
      YP(I,JMAX,1)=1.0
  150 CONTINUE
C
      DO 160 J=3,JMAM
      FB(J)=0.15
      FA(J)=0.20
  160 CONTINUE
      FB(2)=-0.15
      FB(3)=0.
      FA(2)=-0.20
      FA(3)=0.
C
      DO 10 J=2,JMAM
      EB=DEXP(FA(J))
      EBR=1.D0/EB
      PSN=EB+EBR
      EB2=EB*EB
      EB2R=1./EB2
      PPSN=2.*B/(EB2-EB2R)
C
```

```
      AA(J)=-EB
      BB(J)=PSN
      CC(J)=-EBR
   10 DD(J)=0.D0
C
      DO 170 I=1,IMAX
      DD(2)=-AA(2)*YP(I,1,1)
      DD(JMAM)=-CC(JMAM)*YP(I,JMAX,1)
C
      CALL TRIDAG(2,JMAM,AA,BB,CC,DD,T)
C
      DO 379 J=2,JMAM
  379 YP(I,J,1)=T(J)
  170 CONTINUE
C
      DO 1000 ITY=1,250
      SOS=0.
      DO 200 I=2,IMAM
      DO 210 J=2,JMAM
      XXIXI=XP(I+1)-2.*XP(I)+XP(I-1)
      YETET=YP(I,J+1,1)-2.*YP(I,J,1)+YP(I,J-1,1)
      XXI=.5*(XP(I+1)-XP(I-1))
      XET=0.
      YXI=.5*(YP(I+1,J,1)-YP(I-1,J,1))
      YET=.5*(YP(I,J+1,1)-YP(I,J-1,1))
      AJI=XXI*YET
      G11=(YET**2)/AJI/AJI
      G22=(XXI**2+YXI**2)/AJI/AJI
      G12=-YXI*YET/AJI/AJI
      IF(I .LT. NA)THEN
      F2(I,J)=FA(J)
      ELSE IF(I .GT. NB) THEN
      F2(I,J)=FB(J)
      ELSE
      F2(I,J)=((NB-I)*FA(J)+(I-NA)*FB(J))/(NB-NA)
      END IF
      A=F1(I)
      B=F2(I,J)
      IF(DABS(B) .LT. EPE) B=DSIGN(EPE,B)
      EPA=DEXP(A)
      EPB=DEXP(B)
      EPAI=1./EPA
      EPBI=1./EPB
      COSHA=.5*(EPA+EPAI)
      COSHB=.5*(EPB+EPBI)
      CSCHA=2./(EPA-EPAI)
      CSCHB=2./(EPB-EPBI)
      COTHA=COSHA*CSCHA
      COTHB=COSHB*CSCHB
      AB=G22*B*CSCHB
      AA(J)=-AB*EPB
```

```
      BB(J)=2.*(G11*A*COTHA+G22*B*COTHB)
      CC(J)=-AB*EPBI
      DD(J)=.5*G12*(YP(I+1,J+1,1)+YP(I-1,J-1,1)-YP(I-1,J+1,1)
     $       -YP(I+1,J-1,1))+G11*A*CSCHA*(EPA*YP(I-1,J,1)
     $       +EPAI*YP(I+1,J,1))
  210 CONTINUE
      DD(2)=DD(2)-AA(2)*YP(I,1,1)
      DD(JMAM)=DD(JMAM)-CC(JMAM)*YP(I,JMAX,1)
      CALL TRIDAG(2,JMAM,AA,BB,CC,DD,T)
      DO 220 J=2,JMAM
      YT=T(J)-YP(I,J,1)
      IF(DABS(SOS) .LT. DABS(YT)) SOS=YT
  220 YP(I,J,1)=1.8*T(J)-0.8*YP(I,J,1)
  200 CONTINUE
C
      IF(DABS(SOS) .LT. 0.00001) GO TO 999
      DO 666 J=2,JMAM
      YP(IMAX,J,1)=YP(IMAX-1,J,1)-YP(IMAX-1,1,1)+YP(IMAX,1,1)
  666 CONTINUE
C
      WRITE(1,222) ITY, SOS
 1000 CONTINUE
C
  999 DO 555 I=1,IMAX
      F2(I,1)=0.
      F2(I,JMAX)=F2(I,JMAM)
  555 CONTINUE
C
      DO 444 J=1,JMAX
      F2(1,J)=F2(2,J)
      F2(IMAX,J)=F2(IMAM,J)
  444 CONTINUE
C
C
      OPEN(UNIT=6,FILE='PHYSBODY')
      WRITE(6,300)(XP(I),I=1,IMAX)
      WRITE(6,300)(F1(I),I=1,IMAX)
      DO 550 J=1,19
      WRITE(6,300)(YP(I,J,1),I=1,IMAX)
      WRITE(6,300)(F2(I,J),I=1,IMAX)
  550 CONTINUE
      CLOSE(6)
  300 FORMAT(5E14.7)
  700 FORMAT(1X,6I10)
  222 FORMAT(I10,E12.4)
      CALL EXIT
      END
C
C*************************************************************
C     SUBROUTINE TRIDAG TO SOLVE ALGEBRAIC EQUATIONS
C     SIMULTANEOUSLY FOR EACH ROW OR COLOUM
```

```
C*********************************************************************
C
C
      SUBROUTINE TRIDAG(IF,L,A,B,C,D,V)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(90),B(90),C(90),D(90),V(90),BETA(90),GAMMA(9
      BETA(IF)=B(IF)
      GAMMA(IF)=D(IF)/BETA(IF)
      IFP1=IF+1
      DO 1 I=IFP1,L
      BETA(I)=B(I)-A(I)*C(I-1)/BETA(I-1)
    1 GAMMA(I)=(D(I)-A(I)*GAMMA(I-1))/BETA(I)
      V(L)=GAMMA(L)
      LAST=L-IF
      DO 2 K=1,LAST
      I=L-K
    2 V(I)=GAMMA(I)-C(I)*V(I+1)/BETA(I)
      RETURN
      END
C*********************************************************************
C
C.... THIS PROGRAM IS USED TO GENERATE THE BODY-FITTED
C.... COORDINATES ON THE F-57 BODY
C
C*********************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/GEO1/ XP(62),YP(62,20,4)
      COMMON/GEO2/ ZP(62,20,4)
      COMMON/BODY2/ F1(62),F2(62,20),F3(4),GE(20,4)
      DIMENSION AA(90),BB(90),CC(90),DD(90),T(90)
      DIMENSION FA(20), FB(20)
C
      IMAX=62
      JMAX=20
      KMAX=1
      IMAM=IMAX-1
      JMAM=JMAX-1
      KMAM=KMAX-1
      EPE=1.D-5
      A1=-.01
      A2=0.2
      NA=15
      NB=42
      NX1=5
      NX2=40
      NX3=40
C
      OPEN(UNIT=6, FILE='PHYSBODY')
      PI=3.141592653589793D0
      EPS=1.D-12
C
```

```
C.... X-DIRECTION
C
      AX3=NX3
C
      DO 50 I=10,IMAX
      Z1=I/AX3
C
      IF(Z1 .LE. 0.5) F1(I)=A1
      IF(Z1 .GT. 0.5 .AND. Z1 .LE. 1.)GO TO 20
      IF(Z1 .GT. 1. .AND. Z1 .LE. 1.1) GO TO 30
      IF(Z1 .GT. 1.1) F1(I)=A2
      GO TO 50
   20 PIZ=PI*Z1
      F1(I)=A1*DSIN(PIZ)
      GO TO 50
   30 PIZ=PI*Z1
      F1(I)=-A2*DSIN(PIZ)
   50 CONTINUE
C
      DO 55 I=1,5
   55 F1(I)=-F1(NX2+I-5)
      DO 56 I=5,9
   56 F1(I)=-F1(10-I)
C
      DO 60 I=2,IMAM
      AA(I)=-DEXP(F1(I))
      CC(I)=1./AA(I)
      BB(I)=-(AA(I)+CC(I))
   60 DD(I)=0.D0
C
      XP(NX1)=0.4446D0
      XP(NX2)=1.D0
      N1=NX1+1
      N2=NX2-1
C
      DD(N1)=DD(N1)-AA(N1)*XP(NX1)
      DD(N2)=DD(N2)-CC(N2)*XP(NX2)
C
      CALL TRIDAG(N1,N2,AA,BB,CC,DD,T)
C
      DO 70 I=N1,N2
   70 XP(I)=T(I)
C
      DO 72 I=NX1,2,-1
      XP(I-1)=-(XP(I)*BB(I)+XP(I+1)*CC(I))/AA(I)
   72 CONTINUE
C
      DO 74 I=NX2,IMAM
      XP(I+1)=-(XP(I)*BB(I)+XP(I-1)*AA(I))/CC(I)
   74 CONTINUE
C
```

```
C.... Y-DIRECTION
C
      A1=0.0112135D0
      A2=0.0761289D0
      A3=0.1104047D0
      A4=-0.4107083D0
      B2=0.007868094D0
      B3=0.281687965D0
      B4=-0.371566458D0
      B5=-0.031072748D0
C
C
      DO 100 I=1,IMAX
      IP1=I+1
      IM1=I-1
      IF(XP(I) .LT. 0.DO .OR. XP(I) .GT. 1.DO) GO TO 110
      IF(XP(I).GE.0. .AND. XP(I).LE.0.4446) GO TO 120
      IF(XP(I).GT.0.4446 .AND. XP(I).LE.1.) GO TO 130

  110 YP(I,1,1)=0.DO
      GO TO 100
  120 P=XP(I)
      YP(I,1,1)=DSQRT((((A4*P+A3)*P+A2)*P+A1)*P)
      GO TO 100
  130 P=1-XP(I)
      YP(I,1,1)=DSQRT((((B5*P+B4)*P+B3)*P+B2)*P*P)
  100 CONTINUE
C
      DO 150 I=1,IMAX
      YP(I,JMAX,1)=2.0
  150 CONTINUE
C
      DO 160 J=3,JMAM
      FB(J)=0.15
      FA(J)=0.20
  160 CONTINUE
      FB(2)=-0.15
      FB(3)=0.
      FA(2)=-0.20
      FA(3)=0.
C
      DO 10 J=2,JMAM
      EB=DEXP(FA(J))
      FBR=1.DO/EB
      PSN=EB+EBR
      EB2=EB*EB
      EB2R=1./EB2
      PPSN=2.*B/(EB2-EB2R)
C
      AA(J)=-EB
      BB(J)=PSN
```

```
      CC(J)=-EBR
   10 DD(J)=0.D0
C
      DO 170 I=1,IMAX
      DD(2)=-AA(2)*YP(I,1,1)
      DD(JMAM)=-CC(JMAM)*YP(I,JMAX,1)
C
      CALL TRIDAG(2,JMAM,AA,BB,CC,DD,T)
C
      DO 379 J=2,JMAM
  379 YP(I,J,1)=T(J)
  170 CONTINUE
C
      DO 1000 ITY=1,250
      SOS=0.
      DO 200 I=2,IMAM
      DO 210 J=2,JMAM
      XXIXI=XP(I+1)-2.*XP(I)+XP(I-1)
      YETET=YP(I,J+1,1)-2.*YP(I,J,1)+YP(I,J-1,1)
      XXI=.5*(XP(I+1)-XP(I-1))
      XET=0.
      YXI=.5*(YP(I+1,J,1)-YP(I-1,J,1))
      YET=.5*(YP(I,J+1,1)-YP(I,J-1,1))
      AJI=XXI*YET
      G11=(YET**2)/AJI/AJI
      G22=(XXI**2+YXI**2)/AJI/AJI
      G12=-YXI*YET/AJI/AJI
      IF(I .LT. NA)THEN
      F2(I,J)=FA(J)
      ELSE IF(I .GT. NB) THEN
      F2(I,J)=FB(J)
      ELSE
      F2(I,J)=((NB-I)*FA(J)+(I-NA)*FB(J))/(NB-NA)
      END IF
      A=F1(I)
      B=F2(I,J)
      IF(DABS(B) .LT. EPE) B=DSIGN(EPE,B)
      EPA=DEXP(A)
      EPB=DEXP(B)
      EPAI=1./EPA
      EPBI=1./EPB
      COSHA=.5*(EPA+EPAI)
      COSHB=.5*(EPB+EPBI)
      CSCHA=2./(EPA-EPAI)
      CSCHB=2./(EPB-EPBI)
      COTHA=COSHA*CSCHA
      COTHB=COSHB*CSCHB
      AB=G22*B*CSCHB
      AA(J)=-AB*EPB
      BB(J)=2.*(G11*A*COTHA+G22*B*COTHB)
      CC(J)=-AB*EPBI
```

```
      DD(J)=.5*G12*(YP(I+1,J+1,1)+YP(I-1,J-1,1)-YP(I-1,J+1,1)
     $      -YP(I+1,J-1,1))+G11*A*CSCHA*(EPA*YP(I-1,J,1)
     $      +EPAI*YP(I+1,J,1))
  210 CONTINUE
      DD(2)=DD(2)-AA(2)*YP(I,1,1)
      DD(JMAM)=DD(JMAM)-CC(JMAM)*YP(I,JMAX,1)
      CALL TRIDAG(2,JMAM,AA,BB,CC,DD,T)
      DO 220 J=2,JMAM
      YT=T(J)-YP(I,J,1)
      IF(DABS(SOS) .LT. DABS(YT)) SOS=YT
  220 YP(I,J,1)=1.8*T(J)-0.8*YP(I,J,1)
  200 CONTINUE
C
      IF(DABS(SOS) .LT. 0.00001) GO TO 999
      DO 666 J=2,JMAM
      YP(IMAX,J,1)=YP(IMAX-1,J,1)-YP(IMAX-1,1,1)+YP(IMAX,1,1)
  666 CONTINUE
C
      WRITE(1,222) ITY, SOS
 1000 CONTINUE
C
      DO 555 I=1,IMAX
      F2(I,1)=0.
      F2(I,JMAX)=F2(I,JMAM)
  555 CONTINUE
C
      DO 444 J=1,JMAX
      F2(1,J)=F2(2,J)
      F2(IMAX,J)=F2(IMAM,J)
  444 CONTINUE
C
C
  999 WRITE(6,700) IMAX, JMAX
      WRITE(6,300)(XP(I),I=1,IMAX)
      WRITE(6,300)(F1(I),I=1,IMAX)
      DO 550 J=1,19
      WRITE(6,300)(YP(I,J,1),I=1,IMAX)
      WRITE(6,300)(F2(I,J),I=1,IMAX)
  550 CONTINUE
      CLOSE(6)
  300 FORMAT(5E14.7)
  700 FORMAT(1X,6I10)
  222 FORMAT(I10,E12.4)
      CALL EXIT
      END
C
C*************************************************************
C     SUBROUTINE TRIDAG TO SOLVE ALGEBRAIC EQUATIONS
C     SIMULTANEOUSLY FOR EACH ROW OR COLOUM
C*************************************************************
C
```

```
C
      SUBROUTINE TRIDAG(IF,L,A,B,C,D,V)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(90),B(90),C(90),D(90),V(90),BETA(90),GAMMA(9
      BETA(IF)=B(IF)
      GAMMA(IF)=D(IF)/BETA(IF)
      IFP1=IF+1
      DO 1 I=IFP1,L
      BETA(I)=B(I)-A(I)*C(I-1)/BETA(I-1)
    1 GAMMA(I)=(D(I)-A(I)*GAMMA(I-1))/BETA(I)
      V(L)=GAMMA(L)
      LAST=L-IF
      DO 2 K=1,LAST
      I=L-K
    2 V(I)=GAMMA(I)-C(I)*V(I+1)/BETA(I)
      RETURN
      END

C*************************************************************
C
C.... THIS PROGRAM IS USED TO GENERATE THE BODY-FITTED
C.... COORDINATES ON THE OGIVE-NOSE BODY
C
C*************************************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/GEO1/ XP(67),YP(67,20,9)
      COMMON/GEO2/ ZP(67,20,9)
      COMMON/BODY2/ F1(67),F2(67,9),F3(9),GE(20,9)
      DIMENSION AA(90),BB(90),CC(90),DD(90),T(90)
      DIMENSION FA(20), FB(20)
C
C
      IMAX=62
      JMAX=20
      KMAX=9
      IMAM=IMAX-1
      JMAM=JMAX-1
      KMAM=KMAX-1
      EPE=1.D-5
      A1=-.3
      A2=-.2
      NA=10
      NB=42
      NX1=12
      NX2=40
      NX3=25
C
      OPEN(UNIT=6, FILE='PHYSBODY')
C
      PI=3.141592653589793D0
      EPS=1.D-12
```

```
C
C.... X-DIRECTION
C
      AX1=NX1-1.
      AX2=2.*NX1-1.
      AX3=NX2-AX2
C
      DO 50 I=1,IMAX
      Z1=(I-1)/AX1
      Z2=(I-AX2)/AX3
C
      IF(Z1 .LE. 0.5) THEN
      F1(I)=A1
      ELSE IF(Z1 .GT. 0.5 .AND. Z1 .LE. 2.)THEN
      PIZ=PI*Z1
      F1(I)=A1*DSIN(PIZ)
      ELSE IF(Z2 .LE. 1.5) THEN
      PIZ=PI*Z2
      F1(I)=A2*DSIN(PIZ)
      ELSE IF(Z2 .GT. 1.5) THEN
      F1(I)=-A2
      END IF
C
   50 CONTINUE
C
      DO 60 I=2,IMAM
      AA(I)=-DEXP(F1(I))
      CC(I)=1./AA(I)
      BB(I)=-(AA(I)+CC(I))
   60 DD(I)=0.D0
C
      XP(NX1)=0.D0
      XP(NX2)=1.D0
      N1=NX1+1
      N2=NX2-1
C
      DD(N1)=DD(N1)-AA(N1)*XP(NX1)
      DD(N2)=DD(N2)-CC(N2)*XP(NX2)
C
      CALL TRIDAG(N1,N2,AA,BB,CC,DD,T)
C
      DO 70 I=N1,N2
   70 XP(I)=T(I)
C
      DO 72 I=NX1,2,-1
      XP(I-1)=-(XP(I)*BB(I)+XP(I+1)*CC(I))/AA(I)
   72 CONTINUE
C
      DO 74 I=NX2,IMAM
      XP(I+1)=-(XP(I)*BB(I)+XP(I-1)*AA(I))/CC(I)
   74 CONTINUE
```

```
C
C.... Z-DIRECTION
C
      DO 421 K=1,KMAX
      F3(K)=0.
      DO 421 J=1,JMAX
      DO 421 I=1,IMAX
  421 ZP(I,J,K)=(K-5.)*PI/6.
C
C.... Y-DIRECTION
C
      AL=0.5D0
      BL=0.006D0
      XL=0.5D0

      DO 100 I=1,IMAX
      IP1=I+1
      IM1=I-1
      IF(XP(I) .LT. 0.D0 .OR. XP(I) .GT. 1.D0) THEN
      YP(I,1,1)=0.D0
      ELSE IF (XP(I) .GE. 0.D0 .AND. XP(I) .LE. 0.06D0) THEN
      YP(I,1,1)=0.1D0*XP(I)
      ELSE IF (XP(I) .GT. 0.06D0 .AND. XP(I) .LE. 0.5D0) THEN
      YP(I,1,1)=0.006D0
      ELSE IF (XP(I) .GT. 0.5D0 .AND. XP(I) .LE. 1.D0) THEN
      IF(XP(I) .GT. 0.9999D0) XP(I)=0.99999D0
      XXL=XP(I)-XL
      YP(I,1,1)=BL*DSQRT(1.D0-(XXL/AL)**2)
      IF(XP(I) .GE. 0.9999D0) XP(I)=1.D0
      END IF
  100 CONTINUE
C
      DO 150 K=1,KMAX
      DO 150 I=1,IMAX
      YP(I,1,K)=YP(I,1,1)
      YP(I,JMAX,K)=1.5
  150 CONTINUE
C
      DO 123 K=2,KMAM
C     FA(K)=0.26+(KMAM-K)**2/1000.
C     FB(K)=0.26+(KMAM-K)**2/1000.
      FA(K)=0.2835
      FB(K)=0.2835
  123 CONTINUE
      FA(1)=FA(3)
      FB(1)=FB(3)
      FA(KMAX)=FA(KMAM-1)
      FB(KMAX)=FB(KMAM-1)
C
      DO 10 K=1,KMAX
      EB=DEXP(FA(K))
```

```
      EBR=1.D0/EB
      PSN=EB+EBR
      EB2=EB*EB
      EB2R=1./EB2
      PPSN=2.*B/(EB2-EB2R)
C
      DO 40 J=2,JMAM
      AA(J)=-EB
      BB(J)=PSN
      CC(J)=-EBR
   40 DD(J)=0.D0
C
      DO 170 I=1,IMAX
      DD(2)=-AA(2)*YP(I,1,K)
      DD(JMAM)=-CC(JMAM)*YP(I,JMAX,K)
C
      CALL TRIDAG(2,JMAM,AA,BB,CC,DD,T)
C
      DO 379 J=2,JMAM
  379 YP(I,J,K)=T(J)
  170 CONTINUE
   10 CONTINUE
C
      DO 1000 ITY=1,500
      SOS=0.
      DO 190 K=2,KMAM
      C=F3(K)
      DO 200 I=2,IMAM
      IF(I .LT. NA)THEN
      F2(I,K)=FA(K)
      ELSE IF(I .GT. NB) THEN
      F2(I,K)=FB(K)
      ELSE
      F2(I,K)=((NB-I)*FA(K)+(I-NA)*FB(K))/(NB-NA)
      END IF
      A=F1(I)
      B=F2(I,K)
      IF(DABS(B) .LT. EPE) B=DSIGN(EPE,B)
      EPA=DEXP(A)
      EPB=DEXP(B)
      EPC=DEXP(C)
      EPAI=1./EPA
      EPBI=1./EPB
      EPCI=1./EPC
      COSHA=.5*(EPA+EPAI)
      COSHB=.5*(EPB+EPBI)
      COSHC=.5*(EPC+EPCI)
      CSCHA=2./(EPA-EPAI)
      CSCHB=2./(EPB-EPBI)
C     CSCHC=2./(EPC-EPCI)
      COTHA=COSHA*CSCHA
```

```
      COTHB=COSHB*CSCHB
C     COTHC=COSHC*CSCHC
      DO 210 J=2,JMAM
      XXI=.5*(XP(I+1)-XP(I-1))
      YXI=.5*(YP(I+1,J,K)-YP(I-1,J,K))
      YET=.5*(YP(I,J+1,K)-YP(I,J-1,K))
      YZT=.5*(YP(I,J,K+1)-YP(I,J,K-1))
      ZXI=0.
      ZET=0.
      ZZT=0.5*(ZP(I,J,K+1)-ZP(I,J,K-1))*YP(I,J,K)
      A11=XXI*XXI+YXI*YXI+ZXI*ZXI
      A22=YET*YET+ZET*ZET
      A33=YZT*YZT+ZZT*ZZT
      A12=YXI*YET+ZXI*ZET
      A13=YXI*YZT+ZXI*ZZT
      A23=YET*YZT+ZET*ZZT
      G=A11*A22*A33+2.*A12*A13*A23-A23*A23*A11-
     $A13*A13*A22-A12*A12*A33
      GI=1./G
      G11=GI*(A22*A33-A23*A23)
      G22=GI*(A11*A33-A13*A13)
      G33=GI*(A11*A22-A12*A12)
      G12=GI*(A13*A23-A12*A33)
      G13=GI*(A12*A23-A13*A22)
      G23=GI*(A12*A13-A23*A11)
      AB=G22*B*CSCHB
      AA(J)=-AB*EPB
      BB(J)=2.*(G11*A*COTHA+G22*B*COTHB+G33)
      CC(J)=-AB*EPBI
      DD(J)=.5*G12*(YP(I+1,J+1,K)+YP(I-1,J-1,K)-YP(I-1,J+1,K)
     $      -YP(I+1,J-1,K))+G11*A*CSCHA*(EPA*YP(I-1,J,K)
     $      +EPAI*YP(I+1,J,K))+G33*(EPC*YP(I,J,K-1)
     $      +EPCI*YP(I,J,K+1))+0.5*(G13*(YP(I+1,J,K+1)
     $      +YP(I-1,J,K-1)-YP(I+1,J,K-1)-YP(I-1,J,K+1))
     $      +G23*(YP(I,J+1,K+1)+YP(I,J-1,K-1)-YP(I,J+1,K-1)
     $      -YP(I,J-1,K+1)))
  210 CONTINUE
      DD(2)=DD(2)-AA(2)*YP(I,1,K)
      DD(JMAM)=DD(JMAM)-CC(JMAM)*YP(I,JMAX,K)
      CALL TRIDAG(2,JMAM,AA,BB,CC,DD,T)
      DO 220 J=2,JMAM
      YT=T(J)-YP(I,J,K)
      IF(DABS(SOS) .LT. DABS(YT)) SOS=YT
  220 YP(I,J,K)=1.8*T(J)-0.8*YP(I,J,K)
  200 CONTINUE
      Y1=YP(1,JMAX,K)-YP(1,1,K)
      Y2=YP(2,JMAX,K)-YP(2,1,K)
      DO 666 J=2,JMAM
      YP(1,J,K)=(YP(2,J,K)-YP(2,1,K))*Y1/Y2+YP(1,1,K)
      YP(IMAX,J,K)=YP(IMAM,J,K)-YP(IMAM,1,K)+YP(IMAX,1,K)
  666 CONTINUE
```

```
  190 CONTINUE
C
      DO 478 I=2,IMAX
      DO 478 J=2,JMAM
      YP(I,J,1)=YP(I,J,3)
  478 YP(I,J,KMAX)=YP(I,J,KMAM-1)
C
      IF(DABS(SOS) .LT. 0.00001) GO TO 999
C
      WRITE(1,222) ITY, SOS
 1000 CONTINUE
C
      DO 444 K=2,KMAM
      F2(1,K)=FA(K)
      F2(IMAX,K)=FB(K)
  444 CONTINUE
      DO 555 I=1,IMAX
      F2(I,1)=F2(I,3)
      F2(I,KMAX)=F2(I,KMAM-1)
  555 CONTINUE
C
  999 WRITE(6,300)(XP(I),I=1,IMAX)
      DO 540 K=1,KMAX
      DO 540 J=1,19
  540 WRITE(6,300) (YP(I,J,K),I=1,IMAX)
      WRITE(6,300) (F1(I),I=1,IMAX)
      DO 550 K=1,KMAX
  550 WRITE(6,300) (F2(I,K),I=1,IMAX)
      CLOSE(6)
  300 FORMAT(5E14.7)
  700 FORMAT(1X,6I10)
  222 FORMAT(I10,E12.4)
      CALL EXIT
      END
C
C**************************************************************
C     SUBROUTINE TRIDAG TO SOLVE ALGEBRAIC EQUATIONS
C     SIMULTANEOUSLY FOR EACH ROW OR COLOUM
C**************************************************************
C
C
      SUBROUTINE TRIDAG(IF,L,A,B,C,D,V)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(90),B(90),C(90),D(90),V(90),BETA(90),GAMMA(9
      BETA(IF)=B(IF)
      GAMMA(IF)=D(IF)/BETA(IF)
      IFP1=IF+1
      DO 1 I=IFP1,L
      BETA(I)=B(I)-A(I)*C(I-1)/BETA(I-1)
    1 GAMMA(I)=(D(I)-A(I)*GAMMA(I-1))/BETA(I)
      V(L)=GAMMA(L)
```

```
      LAST=L-IF
      DO 2 K=1,LAST
      I=L-K
    2 V(I)=GAMMA(I)-C(I)*V(I+1)/BETA(I)
      RETURN
      END
```

REFERENCES

1. Prandtl, L., "Uber Flussigkeitsbewegung bei Sehr Kleiner Reibung", Proc. 3rd Int. Math. Congr. Heidelberg, pp. 484-491, 1904.

2. Gruschwitz, E., "Turbulente Reibungsschichten mit Secondarstromug", Ingenier-Archiv, Bd. VI, pp. 355-365, 1935.

3. Bradshaw, P. and M.G. Terrell, "The Response of a Turbulent Boundary Layer on an Infinite Swept-Wing to The Sudden Removal of Pressure Gradient", NPL Aero Rept. 1305, 1969.

4. Richmond, L., "Experimental Investigation of Thick Axially Symmetric Boundary Layers on Cylinder at Subsonic and Hypersonic Speeds", Ph.D. Thesis, California Institute of Technology, Pasadena, Ca., 1957.

5. Yasuhara, M., "Experiments of Axisymmetric Boundary Layers Along a Cylinder in Incompressible Flow", Transactions of the Japan Society of Aerospace Science, Vol. 2, pp. 72-76, 1959.

6. Willmarth, W.W. and C.S. Yang, "Wall-Pressure Fluctuations Beneath Turbulent Boundary Layers on a Flat Plate and a Cylinder", Journal of Fluid Mechanics, Vol. 41, pp. 47, 1970.

7. Chevray, R., "Turbulence in the Wakes of a Body of Revolution", Ph.D. Thesis, The University of Iowa, June, 1967.

8. Patel, V.C., A. Nakayama and R. Damian, "Measurements in the Thick Axisymmetric Turbulent Boundary Layer Near the Tail of a Body of Revolution", Journal of Fluid Mechanics, Vol. 63, pp. 345-362, 1974.

9. Patel, V.C. and Y.T. Lee, "Thick Axisymmetric Turbulent Boundary Layer and Near Wake of a Low-Drag Body of Revolution", IIHR Report No. 210, 1977.

10. Huang, T.T., N. Santelli and G.S. Belt, "Stern Boundary Layer Flow on Axisymmetric Bodies", Proc. 12th Sym. Naval Hydrodynamics, Nat. Acad. Sci., pp. 127-157, 1979.

11. Huang, T.T., N.C. Groves and G.S. Belt, "Boundary Layer Flow on an Axisymmetric Body with an Inflected Stern", DTNSRDC-80/064, 1980.

12. Ramaprian, B.R., V.C. Patel and D.H. Choi, "Mean Flow Measurements in the Three-Dimensional Boundary Layer Over a Body of Revolution at Incidence", Journal of Fluid Mechanics, Vol. 103, pp. 479-504, 1981.

13. Baek, J.H., "Three-Dimensional Turbulent Boundary Layers on the Bodies of Revolution at Incidence", Ph.D. Thesis, The University of Iowa, May, 1984.

14. Meier, H.U., H.P. Kreplin and H. Vollmers, "Development of Boundary Layers and Separation Patterns of a Body of Revolution at Incidence", 2nd Sym. on Num. and Phy. Aspects of Aerodynamic Flows, Long Beach, CA, Jan., 1984.

15. Smith, P.D., "The Numerical Computation of Three Dimensional Boundary Layers", Proc. IUTAM Sym., Berlin, Springer-Verlag, pp. 265-285, 1982.

16. Patankar, S.V. and D.B. Spalding, "A Calculation Procedure for Heat Mass and Momentum Transfer in Three-Dimensional Flows", Int. J. Heat Mass Transfer, Vol. 15, pp. 1787-1806, 1972

17. Pratap, V.S. and D.B. Spalding, "Fluid Flow and Heat Transfer in Three-Dimensional Duct Flows", International Journal Heat & Mass Transfer, Vol. 19, pp. 1183-1188, 1976.

18. Abdelmeguid, A.M., N.C.G. Markatos and D.B. Spalding, "A Method of Predicting Three-Dimensional Turbulent Flows Around Sihps' Hulls", Int. Sym. on Ship Viscous Resistance, SSPA, Goteborg, 1978.

19. Muraoka, K., "Calculation of Thick Boundary Layer and Wake of Ships by a Partially Parabolic Method", Proc. 13th ONR Sym. Naval Hydro., Tokyo, pp. 601-616, 1980.

20. Huang, T.T. and M.S. Chang, "Computation of Velocity and Pressure Variation Across Thick Turbulent Stern Flows", 3rd Sym. on Num. and Phy. Aspects of Aerodynamic Flows, Long Beach, CA. Jan., 1985.

21. Chen, H.C. and V.C. Patel, "Calculation of Stern Flows by a Time-Marching Solution of the Partially-Parabolic Equations", Proc. 15th ONR Symp. on Naval Hydrodynamics, Hamburg, FRG, 1984.

22. Chen, H.C. and V.C. Patel, "Calculation of Trailing-Edge, Stern and Wake Flows By a Time-Marching Solution of the Partially-Parabolic Equations", IIHR Report No. 285, The University of Iowa, 1985.

23. Patankar, S.V., "Numerical Heat Transfer and Fluid Flow", McGraw-Hill, N.Y., 1980.

24. Miloh, T. and V. C. Patel, "Orthogonal Coordinate Systems for Three-Dimensional Boundary Layers, With Particular Reference to Ship Forms", Journal of Ship Research, Vol. 17, No. 1, pp. 50-58, 1973.

25. Thompson, J.F., "Symposium on the Numerical Generation of Curvilinear Coordinate Systems and use in the Numerical Solution of Partial differential Equations", Nashville, Tennessee, 1982.

26. Thompson, J.F., Z.U.A. Warsi, and C.W. Mastin, "Boundary-Fitted Coordinate Systems for Numerical Solution of Differential Equations -- A Review", J. Comp. Physics, Vol. 47, pp. 1-108, 1982.

27. Roach, P.J., "Computational Fluid Mechanics", Hermosa Publishers, 1972.

28. Glowinski, R., "A Review on Finite-Element Methods for Fluid Flow Problems", Proc. of Int. Sym. on Refined Modelling of Flows, Paris, 1982.

29. Chen, C.J. and P. Li, "The Finite Analytic Methods for Steady and Unsteady Heat Transfer Problems", ASME Paper No. 80-HT-80, Nineteenth ASME National Heat Transfer Conference, Orlando Florida, July 29-30, 1980.

30. Chen, C.J., H.N. Hamid and K.S. HO, "Finite Analytic Numerical Solution of Heat Transfer in Two-Dimensional Cavity Flow", Numerical Heat Transfer, Vol. 4, pp. 179-197, 1981.

31. Chen, C.J. and H.C. Chen, "Development of Finite Analytic Numerical Method for Unsteady Two Dimensional Convective Transport Equation", Symp. on Refind Modelling Flows, Paris, sept, 1982.

32. Chen, C.J. and H.C. Chen, "Finite Analytic Numerical for Unsteady Two-Dimensional Navier-Stokes Equations", Journal of Computational Physics, Vol. 53, No. 2, Feb., 1984.

33. Chen, C.J. and Y.H. Yoon, "Finite Analytic Numerical Solution Axisymmetric Navier-Stokes and Energy Equations", Journal of Heat Transfer, Vol. 105, pp. 639-645, 1983.

34. Chen, C.J. and H.C. Chen, "The Finite Analytic Method", Vol. 4, IIHR Report No. 232-IV, The University of Iowa, 1982.

35. Chen, C.J., K.S. Ho and W.S. Cheng, "The Finite Analytic Method", Vol. 5, IIHR Report No. 232-V, The University of Iowa, 1982.

36. Vahid T. and C.J. Chen, "Finite Analytic Solutions of Steady and Transient Natural Convection in Two-Dimensional Rectangular Enclosures", ASME paper No. 85-WAHT-68, ASME Winter Annual Meeting, Florida, Nov. 17-21, 1985.

37. Chen, C.J., "Finite Analytic Method", Chapter 17, Handbook of Numerical Heat Transfer, Edited by W.J. Minkowycz, E.M. Sparrow, R.H. Plectcher and G.E. Schneider, John Willey & Sons, Jan., 1986.

38. Warming, R.F. and R.M. Beam, "Upwind Second-Order Difference Schemes and Applications in Unsteady Aerodynamic Flows", Proc. AIAA 2nd. Computational Fluid Dynamics Conference, Hartford, Connecticut, pp. 17-28, 1975.

39. Reynolds, O., "On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Criterion", Phil. Trans. R. Soc., Vol. 186, pp. 123-164, 1895.

40. Phillips, O.M., "The Dynamics of the Upper Ocean", Cambridge University Press, Cambridge, England, 1969.

41. Prandtl, L., "Bericht uber Untersuchungen zue ausgebildeten Turbulenz", Z. Angew. Math. Mech. , vol. 5, pp. 136-139, 1925.

42. Rodi, W., "The Prediction of Free Turbulent Boundary Layers by Use of a Two-Equation Model of Turbulence", Ph.D. thesis, University of London, 1972.

43. Jones, W.P. and B.E. Launder, "The Calculation of Low Reynolds Number Phenomena with a Two Equation Model of Turbulence", Int. J. of Heat and Mass Transfer, Vol. 16, pp. 119-130, 1973.

44. Chen, C.J. and K. Singh, "Prediction of Buoyant Free Shear Flows by K- Model Based on Two Turbulence Scale Concept", Proc. of Int. Sym. on Buoyant Flows, Athens, Greece, September, pp. 26-36, 1986.

45. Hinze, J.O., "Turbulence", Chapter 3, Mcgraw-Hill, 1975.

46. Launder, B.E., G.E. Reece and W. Rodi, "Progress in the Development of a Reynolds-stress Turbulent Closure", J. F. Mech., Vol. 68, part3, pp. 537-566, 1975.

47. Chen, C.J., "Prediction of Turbulent Flows", Central Research Institute of Electric Power Industry, Abiko City, Chibu, Japan, 1983.

48. Raithby, G.D. and G.E. Schneider, "Numerical Solution of Problem in Incompressible Fluid Flow: Treatment of the Velocity-Pressure Coupling", Numerical Heat Transfer, Vol. 2, pp. 417-440, 1979.

49. Harlow, F.H. and J.E. Welch, "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface", Phy. Fluids, Vol. 8, No. 12, pp. 2182-2189, 1965.

50. Schneider, G.E. and M. Zedan, "A modified Strongly Implicit Procedure for the Numerical Solution of Field Problems", Numer. Heat Transfer, Vol. 4, pp. 1-19, 1981.

51. Saint-Victor X. and J. Cousteix, "Methods Iteratives de Marche Appliquees au Calcul du Melange Tridimensionnel d'un Sillage et d'une Couche Limite", ONERA Rept., 61/2259 AND. 1984.

52. Rubin, S.G. and D.R. Reddy, "Global Solution Procedures for Incompressible Laminar Flow with Strong Pressure Interaction and Separation", Proc. 2nd Sym. Numerical & Physical Aspects of Aerodynamic Flows, Long Beach, CA. 1983.

53. Werle, H., "Le Tunnel Hydrodynamique au Service de la Recherche Aerospatiale", ONERA, Publ. no. 156, 1974.

54. Freuler and Gregorek, "Experimental data from Ohio State University in NACA Conference", Part 1, Advanced Technology Airfoil Research, Vol. 1, pp. 1590.

55. Cebeci, T. and A.M.O. Smith, "Analysis of Turbulent Boundary Layers", Academic Press., New York, 1974.

56. Granville, P.S., "The Prediction of Transition from Laminar to Turbulent Flow in Boundary Layers on Bodies of Revolution", in R.D. Cooper and S.W. Doroff, (eds.), "Tenth Symposium Naval Hydrodynamics", U.S. Government Printing Office, Washington D.C. pp. 705-729, 1974.

57. Ramaprian, B.R., V.C. Patel and M.S. Sastry, "Turbulent Wake Development Behind Streamline Bodies", AIAA journal, Vol. 20, pp. 1228-1235, 1982.

58. Schlichting, H., "Experimentelle Untersuchungen zum Rauhigkeitsproblem", NACA TM823, 1937.

59. Cebeci, T., "Laminar and Turbulent Incompressible Boundary Layers on Slender Bodies of Revolution in Axial Flow", J. of Basic Engineering, pp. 545-554, Sept., 1970.