

DTIC FILE COPY

4

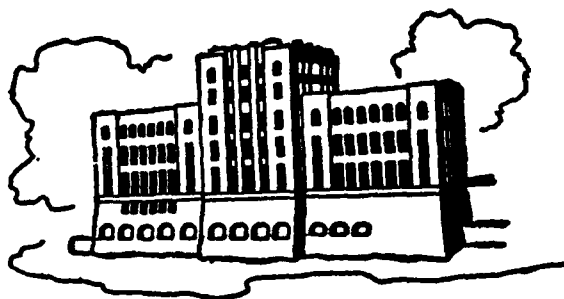
Volume I

AD-A222 787

# The Finite Analytic Method And Its Applications

## Laminar and Turbulent Flows Past Two Dimensional and Axisymmetric Bodies

Ching-Jen Chen  
Seok Ki Choi



DTIC  
ELECTE  
JUN 14 1990  
S E D  
ep

IIHR Report No. 334-I

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

March 1990

DISTRIBUTION STATEMENT A  
Approved for public release:  
Distribution Unlimited

90 06 14 185

Volume I

# **The Finite Analytic Method And Its Applications**

**Laminar and Turbulent Flows Past Two  
Dimensional and Axisymmetric Bodies**

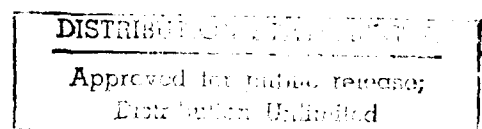
by

**Ching-Jen Chen  
Seok Ki Choi**

**IIHR Report No. 334-I**

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


**March 1990**



REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER <b>IIHR Report No. 334-II</b>	2. GOVT ACCESSION NO.	3. REPORT'S CATALOG NUMBER
4. TITLE (and Subtitle)  <b>Laminar and Turbulent Flows Past Two Dimensional and Axisymmetric Bodies</b>	5. TYPE OF REPORT & PERIOD COVERED <b>Final Report January</b>	
	6. PERFORMING ORG. REPORT NUMBER <b>IIHR Report No. 334-II</b>	
7. AUTHOR(s)  <b>Ching-Jen Chen and Seok Ki Choi</b>	8. CONTRACT OR GRANT NUMBER(s)  <b>N00167-86-K-0019</b>	
9. PERFORMING ORGANIZATION NAME AND ADDRESS <b>Department of Mechanical Engineering Iowa Institute of Hydraulic Research The University of Iowa, Iowa City, IA 52242</b>	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS <b>Office of Naval Research 800 North Quincy Street Arlington, Virginia 22217</b>	12. REPORT DATE <b>2-1-1990</b>	
	13. NUMBER OF PAGES	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report)  <b>Unclassified</b>	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report)  <b>Approval for public release; Distribution unlimited</b>		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  <b>Computational Fluid Dynamics, Numerical method, Laminar and turbulent flow, ship bodies, wakes, Finite Analytic Method, Navier-Stokes equations, Boundary layer</b>		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  <b>* see next page</b>		

## Laminar and Turbulent Flows Past Two Dimensional and Axisymmetric Bodies

### ABSTRACT



A numerical study of laminar and turbulent flows past two dimensional bodies and axisymmetric bodies is presented. Numerical methods are developed to solve Navier-Stokes equations for two dimensional and axisymmetric flows in the arbitrary geometries. The complex physical geometry is resolved by use of numerically generated, body-fitted coordinates. The governing equations are written in the transformed domain using the orthogonal velocity components as dependent variables for momentum equations. The governing equations are discretized using both the finite analytic method and the finite volume method. Both one velocity staggered grid method and two velocities staggered method are employed for grid arrangements. The velocity and pressure coupling techniques in these grid arrangements are presented. The solution procedure of the SIMPLER numerical algorithm is used with a parabolic marching technique and a global pressure calculation method. For turbulent flow calculations, both the  $k-\epsilon$  turbulence model and the two-layer turbulence model are used.  $(kR) \leftarrow \epsilon$

Calculations are performed for laminar and turbulent flows past a finite flat plate and turbulent flow past axisymmetric bodies with different solution domains, numerical methods and turbulence models. Calculations include the development of a wake function method for the prediction of turbulent wake of a flat plate, predictions of laminar and turbulent flows past a finite flat plate, predictions of turbulent flow past axisymmetric bodies by the wall function method and by the two-layer turbulence model and predictions of turbulent flow past finite axisymmetric bodies. Comparisons of predictions by finite analytic method with those by finite volume method are made for some calculations. All the predicted results were compared with available experiment measurements. Good agreements between the predicted results and measured data were obtained.



## PREFACE

This monograph entitled "The Finite Analytic Method and Its Applications" contains the continuing developments of the numerical method called the "finite analytic" method. The finite analytic method was developed in early 1977 when I and Dr. Peter Li, then a graduate student, had difficulty in obtaining a converged solution from a system of finite difference algebraic equations derived for the Navier-Stokes equations for two-dimensional turbulent flow with a second order turbulence model. I conceived the finite analytic method one night and solved the simple two-dimensional Laplace equation. Dr. Li then carried the finite analytic method to the unsteady diffusion equation and nonlinear ordinary differential equation. The basic idea of the finite analytic method is to derive the approximate algebraic representation of a governing linear or nonlinear differential equation from the local analytic solution. The local analytic solution is obtained for a small element of the total solution domain in which the governing equation, if nonlinear, is linearized. The local analytic solution is then expressed in an algebraic form. The system of local analytic algebraic equations is solved to provide the numerical solution of the problem.

Subsequently, in a span of ten years, many students took interest in the development of the finite analytic method and its applications to fluid mechanics and heat transfer problems. They are Hamid Naseri-Neshat, Hamn-Ching Chen, Kuo-San Ho, Young Hwan Yoon, Che Hsi Yu, Sen-Ming Chang, Wenchung Chen, Zahed Mohammad Sheikholeslami, Tzong-Shyan Wung, Wu Sun Chen, Kemakolam Obasih, Seok Ki Choi, Vahid Talaie, Ramiro Humberto Bravo, and Hakan Aksoy. They all contributed to the further development of the finite analytic method. In 1983 a monograph entitled "The Finite Analytic Method" was published as part of the Iowa Institute of Hydraulic Research Report. There are a total of seven volumes in the monograph entitled "The Finite Analytic Method." The special feature of the finite analytic method is that when it is applied to solve the Navier-Stokes equations, it provides relatively accurate representations of the convection term. The finite analytic method not only provides accurate simulation of the upwinding effect because of its analytic nature of the solutions, but it also provides automatic, gradual shifts of the upwinding. Consequently, the finite analytic numerical solutions of convection diffusion equations minimize the false numerical diffusion that would occur in the upwinding difference used in other numerical methods and provide stable and fast convergency of the solutions.

Six years have passed since the publication of the monograph entitled "Finite Analytic Method." During these years the emphasis has been to apply the finite analytic method to various problems and to develop a more friendly code for users. This monograph thus is developed to document the efforts made in the applications of the finite analytic method along with its further improvements.

I would like to thank my colleagues, V.C. Patel, K.B. Chandran, T.F. Smith, A. T. Chwang, F. Stern, and K. Atkinson for their interest in the development and application of the finite analytic method. I would like to acknowledge the support of Drs. William D. McNally, Peter M Sockol, Gary Johnson, and J.J. Adamczyk of NASA Lewis Center for their support in the early development of the finite analytic method. My thanks also go to Dr. Oscar P. Manley of the U.S. Department of Energy for his support of the application of the finite analytic method in energy related problems. Recent applications of the finite analytic method to ship hydrodynamics problems have been supported by the Navy GHR Grant No. N0014-84-0068, by the Naval Sea System Command GHR Grant No. N00168-86-J-0019 administered by the David Taylor Naval Ship Research and Development Center, and by a recent grant from the Office of Naval Research under the Applied Hydrodynamics Research Grant No. N00167-86-K-0019.

In Volume I of the Finite Analytic Method and Its Application, laminar and turbulent flows past a two-dimensional and axisymmetric body are studied. I would like to thank Dr. Seok Ki Choi for applying the finite analytic method to complex flow problems. Without his participation, the finite analytic method would not have been developed and understood as it is today. I shall be happy to receive any discussion or criticism on the finite analytic method. I attempt to present the finite analytic method and its applications in a systematic documentation. Whenever possible, all the listings of the codes developed associated with the finite analytic method are enclosed so that readers may apply the finite analytic method in their problems.

Ching Jen Chen  
 Professor and Chairman,  
 Department of Mechanical Engineering  
 Senior Research Scientist,  
 Iowa Institute of Hydraulic Research  
 The University of Iowa  
 Iowa City, Iowa 52242  
 Telephone: (319) 335-5673  
 FAX: (319) 335-5669

November 1989

<b>Accession For</b>		
NTIS	GRA&I	<input checked="" type="checkbox"/>
DTIC	TAB	<input type="checkbox"/>
Unannounced		<input type="checkbox"/>
Justification		
By _____		
Distribution/		
<b>Availability Codes</b>		
	Avail and/or	
Dist	Special	
A-1		



## ABSTRACT

A numerical study of laminar and turbulent flows past two dimensional bodies and axisymmetric bodies is presented. Numerical methods are developed to solve Navier-Stokes equations for two dimensional and axisymmetric flows in the arbitrary geometries. The complex physical geometry is resolved by use of numerically generated, body-fitted coordinates. The governing equations are written in the transformed domain using the orthogonal velocity components as dependent variables for momentum equations. The governing equations are discretized using both the finite analytic method and the finite volume method. Both one velocity staggered grid method and two velocities staggered method are employed for grid arrangements. The velocity and pressure coupling techniques in these grid arrangements are presented. The solution procedure of the SIMPLER numerical algorithm is used with a parabolic marching technique and a global pressure calculation method. For turbulent flow calculations, both the  $k - \epsilon$  turbulence model and the two-layer turbulence model are used.

Calculations are performed for laminar and turbulent flows past a finite flat plate and turbulent flow past

axisymmetric bodies with different solution domains, numerical methods and turbulence models. Calculations include the development of a wake function method for the prediction of turbulent wake of a flat plate, predictions of laminar and turbulent flows past a finite flat plate, predictions of turbulent flow past axisymmetric bodies by the wall function method and by the two-layer turbulence model and predictions of turbulent flow past finite axisymmetric bodies. Comparisons of predictions by finite analytic method with those by finite volume method are made for some calculations. All the predicted results were compared with available experimental measurements. Good agreements between the predicted results and measured data were obtained.

## TABLE OF CONTENTS

	Page
LIST OF TABLES .....	vii
LIST OF FIGURES .....	viii
NOMENCLATURE .....	xiv
 CHAPTER	
I. INTRODUCTION .....	1
I.1 Purpose of Study .....	1
I.2 Theoretical Background .....	2
I.3 Generation of Boundary-Fitted Coordinate .	8
I.4 Calculation Methods for Incompressible Flow in Complex Geometries .....	11
I.5 Turbulence Models .....	20
I.6 Outline of Study .....	27
II. FORMULATION OF PROBLEM .....	29
II.1 Introduction .....	29
II.2 Governing Equations .....	30
II.3 Solution Domains and Boundary Conditions .	31
II.4 Numerical Grid Generation .....	34
II.5 Finite Analytic Method .....	41
II.6 Finite Volume Method .....	51
III. SOLUTION PROCEDURE .....	56
III.1 Introduction .....	56
III.2 Pressure and Pressure Correction Equation in Cartesian Coordinate System .....	57
III.3 Pressure and Pressure Correction Equation in Boundary-Fitted Coordinate System .....	60
III.4 Overall Solution Procedure .....	72
IV. CALCULATION OF TURBULENT WAKE PAST A FLAT PLATE BY WAKE FUNCTION METHOD .....	76
IV.1 Introduction .....	76
IV.2 Numerical Grids and Solution Domains .....	79

IV.3	Boundary Conditions .....	80
IV.4	Wake Function Method .....	82
IV.5	Results and Discussions .....	92
IV.6	Conclusion .....	98
V.	LAMINAR AND TURBULENT FLOW PAST A FINITE FLAT PLATE .....	99
V.1	Introduction .....	99
V.2	Numerical Grids and Solution Domains .....	103
V.3	Boundary Conditions .....	103
V.4	Numerical Methods .....	104
V.5	Results and Discussions .....	105
VI.	TURBULENT FLOW PAST AXISYMMETRIC BODIES ..	114
VI.1	Introduction .....	114
VI.2	Two-Layer Model .....	120
VI.3	Numerical Grids and Solution Domains .....	122
VI.4	Boundary Conditions .....	122
VI.5	Results and Discussions .....	126
VII.	TURBULENT FLOW PAST FINITE AXISYMMETRIC BODIES .....	136
VII.1	Introduction .....	136
VII.2	Numerical Grids and Solution Domains .....	140
VII.3	Boundary Conditions .....	142
VII.4	Results and Discussions .....	145
VIII.	CONCLUSIONS AND RECOMMENDATIONS .....	151
REFERENCES	.....	235
APPENDIX	.....	247

LIST OF TABLES

Table	Page
1. Definition of Variables .....	156
2. Solution Domains for Calculations in Chapter IV ....	157
3. Solution Domains for Calculations in Chapter V .....	157
4. Solution Domains for Calculations in Chapter VI ( Wall Function Method ) .....	158
5. Solution Domains for Calculations in Chapter VI ( Two Layer Model ) .....	158
6. Solution Domains for Calculations in Chapter VII ...	159

## LIST OF FIGURES

Figures	Page
I-1. Grid Configurations.....	160
II-1. Solution Domains.....	161
II-2. Notations for Grid Generation.....	162
II-3. Numerical Grids for A Flat Plate.....	163
II-4. Numerical Grids for Axisymmetric Bodies ( Half Body Calculation ).....	164
II-5. Numerical Grids for Axisymmetric Bodies ( Full Body Calculation ).....	166
II-6. A Local Element for Finite Analytic Method in Catesian Coordinate System.....	168
II-7. A Local Element for Finite Analytic Method in Boundary-Fitted Coordinate System.....	169
II-8. A Local Element for Finite Volume Method in Boundary-Fitted Coordinate System.....	170
III-1. Control Volume for Pressure Variable in Cartesian Coordinate System.....	171
III-2. One Velocity Staggered Grid System.....	172
III-3. Two Velocities Staggered Grid System.....	173
IV-1. Computational Domain.....	174
IV-2. Notation for Wake Function Method.....	175
IV-3. Pressure Distribution on the Plate and along the Wake Centerline ( $y^+ \sim 150$ )....	176
IV.4. Pressure Distribution Given in Ref. [70].....	176
IV-5. Skin Friction Coefficient.....	177
IV-6. Wake Centerline Velocity Distribution.....	178



Figures	Page
IV-7. Wake Centerline Velocity Distribution Given in Ref.[70].....	178
IV-8. Velocity Profiles in the Near wake.....	179
IV-9. Reynolds Shear Stress Distributions in the Near Wake.....	180
IV-10. Turbulent Kinetic Energy Distributions in the Near Wake.....	181
IV-11. Near-Wake Normal Velocity Distributions.....	177
IV-12. Far-Wake Velocity Defect Profiles.....	182
IV-13. Far-Wake Reynolds Shear Stress Profiles.....	182
V-1. Pressure Distribution ( Laminar Flow ).....	183
V-2. Pressure Distribution near the Trailing Edge ( Laminar Flow ).....	183
V-3. Skin Friction Coefficient ( Laminar Flow )....	184
V-4. Wake Centerline Velocity Distribution ( Laminar Flow ).....	184
V-5. Pressure Distribution ( Turbulent Flow ).....	185
V-6. Pressure Distribution near the Trailing Edge ( Turbulent Flow ).....	185
V-7. Friction Velocity Distribution ( Turbulent Flow ).....	186
V-8. Wake Centerline Velocity Distribution ( Turbulent Flow ).....	186
V-9. Velocity Profiles ( Turbulent Flow ).....	187
V-10. Turbulent Kinetic Energy Profiles.....	188
VI-1. Notations and Solution Domains.....	189
VI-2. Convergence History.....	190

Figures	Page
VI-3. Pressure Convergence History ( Finite Analytic Method ).....	191
VI-4. Pressure Convergence History ( Finite Volume Method ).....	191
VI-5. Friction Velocity Convergence History ( Finite Analytic Method ).....	192
VI-6. Friction Velocity Convergence History ( Finite Volume Method ).....	192
VI-7. Wake Centerline Velocity Convergence History ( Finite Analytic Method ).....	193
VI-8. Wake Centerline Velocity Convergence History ( Finite Volume Method ).....	193
VI-9. Surface Pressure Distribution ( Afterbody-1 ).....	194
VI-10. Surface Pressure Distribution ( Afterbody-2 ).....	194
VI-11. Surface Pressure Distribution ( Afterbody-5 ).....	195
VI-12. Friction Velocity Distribution ( Afterbody-1 ).....	195
VI-13. Friction Velocity Distribution ( Afterbody-2 ).....	196
VI-14. Friction Velocity Distribution ( Afterbody-5 ).....	196
VI-15. Pressure Distributions ( Afterbody-1 ).....	197
VI-16. Pressure Distributions ( Afterbody-2 ).....	198
VI-17. Pressure Distributions ( Afterbody-5 ).....	199
VI-18. Velocity Profiles ( Afterbody-1 ).....	200
VI-19. Velocity Profiles ( Afterbody-2 ).....	201
VI-20. Velocity Profiles ( Afterbody-5 ).....	202

Figures	Page
VI-21. Turbulent Kinetic Energy Profiles ( Afterbody-1 ).....	203
VI-22. Turbulent Kinetic Energy Profiles ( Afterbody-2 ).....	204
VI-23. Turbulent Kinetic Energy Profiles ( Afterbody-5 ).....	205
VI-24. Surface Pressure Distribution ( Afterbody-1 ).....	206
VI-25. Surface Pressure Distribution ( Afterbody-2 ).....	206
VI-26. Surface Pressure Distribution ( Afterbody-3 ).....	207
VI-27. Surface Pressure Distribution ( Afterbody-5 ).....	207
VI-28. Friction Velocity Distribution ( Afterbody-1 ).....	208
VI-29. Friction Velocity Distribution ( Afterbody-2 ).....	208
VI-30. Friction Velocity Distribution ( Afterbody-3 ).....	209
VI-31. Friction Velocity Distribution ( Afterbody-5 ).....	209
VI-32. Friction Velocity Distribution for Afterbody-3 Given in Ref. [56].....	210
VI-33. Friction Velocity Distribution for Afterbody-5 Given in Ref. [56].....	210
VI-34. Velocity Profiles ( Afterbody-1 ).....	211
VI-35. Velocity Profiles ( Afterbody-2 ).....	212
VI-36. Velocity Profiles ( Afterbody-3 ).....	213
VI-37. Velocity Profiles ( Afterbody-5 ).....	214

Figures	Page
VI-38. Turbulent Kinetic Energy Profiles ( Afterbody-1 ) .....	215
VI-39. Turbulent Kinetic Energy Profiles ( Afterbody-2 ) .....	216
VI-40. Turbulent Kinetic Energy Profiles ( Afterbody-5 ) .....	217
VII-1. Convergence History.....	218
VII-2. Grid Dependence Test for Pressure.....	219
VII-3. Grid Dependence Test for Friction Velocity....	219
VII-4. Surface Pressure Distribution ( Afterbody-1 ) .....	220
VII-5. Surface Pressure Distribution ( Afterbody-2 ) .....	221
VII-6. Surface Pressure Distribution ( Afterbody-3 ) .....	222
VII-7. Surface Pressure Distribution ( Afterbody-5 ) .....	223
VII-8. Friction Velocity Distribution ( Afterbody-1 ) .....	224
VII-9. Friction Velocity Distribution ( Afterbody-2 ) .....	225
VII-10. Friction Velocity Distribution ( Afterbody-3 ) .....	226
VII-11. Friction Velocity Distribution ( Afterbody-5 ) .....	227
VII-12. Velocity Profiles ( Afterbody-1 ) .....	228
VII-13. Velocity Profiles ( Afterbody-2 ) .....	229
VII-14. Velocity Profiles ( Afterbody-3 ) .....	230
VII-15. Velocity Profiles ( Afterbody-5 ) .....	231

Figures	Page
VII-16. Turbulent Kinetic Energy Profiles ( Afterbody-1 ).....	232
VII-17. Turbulent Kinetic Energy Profiles ( Afterbody-2 ).....	233
VII-18. Turbulent Kinetic Energy Profiles ( Afterbody-5 ).....	234

## NOMENCLATURE

- $A, B$  : convection coefficients in the linerized transport equation in Eq. (II-63).
- $A_\phi, B_\phi, D_\phi, E_\phi$  : coefficients of the transport equation for variable  $\phi$  in Eq. (II-60).
- $A_E, A_W, A_N$  : coefficients of algebraic equations.
- $A_S, A_P$  :
- $A_W$  : wall surface area.
- $A_\epsilon, A_\mu$  : constants in turbulence model.
- $B$  : constant in the law of the wall ( = 5.5 ).
- $b$  : half width of wake.
- $b_\phi$  : total source term for variable  $\phi$ .
- $b_i^j$  : geometric coefficients.
- $C_f$  : friction coefficient ( =  $\frac{2\tau_w}{\rho U_\infty^2}$  ).
- $C_{nb}$  : finite analytic coefficients.
- $C_p$  : pressure coefficient ( =  $\frac{2(p-p_\infty)}{\rho U_\infty^2}$  ).
- $C_\mu, C_D, C_1$  : constants in turbulence model.
- $C_{\epsilon 1}, C_{\epsilon 2}$  :
- $D_N$  : mass source arised from non-orthogonality of grid in Eq. (III-33).

- $\hat{D}$  : mass source in pressure equation in Eq. (III-40).
- $D^*$  : mass source in pressure correction equation in Eq. (III-44).
- $D_i^j$  : geometric coefficients.
- $D_1^u, D_2^u, D_1^v, D_2^v$  : coefficients of pressure term in momentum equation in body-fitted coordinates.
- $D_e, D_w, D_n, D_s$  : diffusion terms in finite volume method.
- $d_e, d_w, d_n, d_s$  : coefficients of pressure term in momentum equation in cartesian coordinates.
- $d(\xi)$  : distance from the reference station to the body surface at  $\xi$ -station.
- $E_2$  : series summation term in Eq. (II-80).
- $F$  : source function in Eq. (II-35).
- $F_e, F_w, F_n, F_s$  : convection terms in finite volume method.
- $f^i$  : grid control functions.
- $F^2, F^A, F^B, F^C$  : grid control functions in  $\eta$ -direction.
- $G$  : generation term of turbulent kinetic energy.
- $g, g^{ij}, g_{ij}$  : geometric coefficients.
- $h$  : grid size in local element.
- $J$  : Jacobian.

$k$  : 1) turbulent kinetic energy.  
 2) grid size in local element.

$k^+$  : dimensionless turbulent kinetic energy.

$k_{in}$  : turbulent kinetic energy at inlet station.

$k_w$  : turbulent kinetic energy at the wake function region.

$L$  : body length.

$l_{in}$  : turbulent length scale at inlet station.

$l_m, l_t, l_\varepsilon, l_\mu$  : turbulent length scales.

$n$  : normal distance from the wall.

$n^+$  : dimensionless normal distance from the wall.

$n_1, n_2$  : components of unit normal vector at wall.

$P$  : pressure.

$P'$  : pressure correction.

$q$  : magnitude of velocity vector.

$q_\xi$  : velocity component in  $\xi$ -direction.

$R_o$  : body radius.

$R_{max}$  : maximum body radius.

$Re$  : Reynolds number ( =  $\frac{U_\infty L}{\nu}$  ).

$R_k$  : turbulent Reynolds number ( =  $Re n k^{1/2}$  ).

$R_\phi$  : effective Reynolds number for variable  $\phi$ .

$S_\phi, S_\phi^p, S_\phi^c, S_\phi^b$  : source functions for variable  $\phi$ .

$Tu$  : turbulent intensity.



$T_{wx}, T_{wy}$	:	wall shear force in x,y directions.
$u, v$	:	velocity components.
$u^*, v^*$	:	star velocity components.
$\hat{u}, \hat{v}$	:	pseudovelocity components.
$U, V$	:	contravariant velocity components.
$U^*, V^*$	:	contravariant star velocity components.
$\hat{U}, \hat{V}$	:	contravariant pseudovelocity components.
$U_\infty$	:	free stream velocity.
$u_c$	:	wake centerline velocity.
$u_\tau$	:	friction velocity ( = $(\frac{\tau_w}{\rho})^{1/2}$ ).
$u_{\tau 0}$	:	trailing edge friction velocity.
$-\overline{u_i u_j}$	:	Reynolds shear stresses.
$v_w$	:	normal velocity component in the wake function region.
$w$	:	velocity defect in the wake.
$w_0$	:	centerline velocity defect in the wake.
$x, y$	:	Cartesian coordinates.
$x^+, y^+$	:	dimensionless distance in x,y directions.
$x, r$	:	cylindrical coordinates.

#### Greek Symbols

$\alpha_u, \alpha_v, \alpha_p,$	:	under-relaxation factors.
$\alpha_k, \alpha_\epsilon$		

$\Delta_p$	:	dimensionless pressure gradient.
$\Delta t$	:	time step.
$\Delta_\tau$	:	dimensionless stress gradient.
$\Delta x, \Delta y$	:	size of control volume in x,y direction.
$\Delta V$	:	volume of control volume.
$\delta$	:	boundary layer thickness.
$\delta_{ij}$	:	Kronecker delta.
$\delta n$	:	normal distance from the wall.
$\delta x, \delta y$	:	distance between the pressure calculation points.
$\varepsilon$	:	rate of turbulent kinetic energy dissipation.
$\varepsilon^+$	:	dimensionless dissipation rate.
$\varepsilon_{in}$	:	$\varepsilon$ at inlet station.
$\varepsilon_w$	:	$\varepsilon$ at wake function region.
$\Gamma_\phi$	:	effective diffusion constant for $\phi$ .
$\kappa$	:	von-Karman constant.
$\lambda_m$	:	eigenvalues.
$\lambda_w$	:	coefficient in wall function.
$\mu$	:	dynamic viscosity.
$\nu$	:	kinematic viscosity.
$\nu_t$	:	turbulent eddy viscosity.
$\xi, \eta$	:	transformed coordinates.
$\xi^*, \eta^*$	:	grid stretching functions.
$\eta_{ref}$	:	reference station.

$\theta$  : momentum thickness.  
 $\rho$  : density.  
 $\sigma_k, \sigma_\epsilon$  : constants in turbulence model.  
 $\tau_w$  : wall shear stress.  
 $\phi$  : transport quantities.

### Subscripts

$E, W, N, S, P$  : calculation points.  
 $e, w, n, s$  : control volume surface.  
 $nb$  : neighbouring nodal points.  
 $x, y, t$  : first derivative with respect to  $x, y, t$ .  
 $xx, yy$  : second derivative with respect to  $x, y$ .  
 $\xi, \eta$  : first derivative with respect to  $\xi, \eta$ .  
 $\xi\xi, \eta\eta$  : second derivative with respect to  $\xi, \eta$ .  
 $\phi$  : variables.

### Superscripts

$e, w, n, s$  : control volume surface.  
 $P$  : calculation point.  
 $n-1$  : (n-1) th time step.  
 $\phi$  : variables.

## CHAPTER I

### INTRODUCTION

#### I.1 Purpose of Study

Recent demands in advanced ship design have prompted many researchers to study flow past ship propulsion systems. Since most ship propellers are located at the downstream end of ship body, the study of flow past ship bodies is important for the design of optimal ship shape and propellers. However, the present state-of-the-art of prediction methods of flow past complex geometries is not yet capable of calculating the entire flow field past ship bodies. The investigations of flows past axisymmetric bodies can provide a fundamental understanding of flow field past more complex ship bodies. Studies of fluid flows past two dimensional bodies such as hydrofoils and appendages are also important for ship design since the understanding of flow field past appendages will provide more realistic simulation of inlet flow conditions for the propeller.

The purpose of the present study is to develop numerical calculation methods for predicting laminar and

turbulent flows past two dimensional and axisymmetric bodies. The present study starts with the prediction of laminar and turbulent flows past a thin flat plate and then turbulent flows past axisymmetric bodies of DTNSRDC (David Taylor Naval Ship Research and Development Center) are investigated in detail. These geometries are chosen not only because their geometries approximately resemble the appendages or ship body but also because of the availability of extensive experimental data to compare with the predicted results. The shapes of these bodies are presented with generated numerical grid in the following chapter.

In the following sections, the theoretical background of present investigation is presented. It includes the review of numerical methods, grid generation techniques, calculation methods for incompressible flow in arbitrary geometry, and turbulence model.

## 1.2 Theoretical Background

Fluid flows past two dimensional or axisymmetric bodies are also encountered in a variety of practical engineering applications, for example, aerodynamics and turbomachineries. The most reliable information may be obtained by direct measurements. However, experimental

investigations are in most case expensive and time consuming. There also exist difficulties of measurement in many situations, and measuring instruments are not free from errors.

On the other hand, with the rapid development of computer technology in computational speed and size of storage, the computational methods became practical to simulate various flow phenomena encountered in many engineering problems. During the past twenty years, the computational methods for incompressible flow have been rapidly developed accompanied with the advances of practical turbulence model. Various computational methods have been proposed, tested and refined to a stage which may significantly impact the design of many engineering problems. Engineers and researchers begin to realize that the CFD (Computational Fluid Dynamics) is perhaps a cost effective and convenient way of analyzing the complex engineering problems.

Two numerical methodologies are the most commonly employed and developed, namely finite difference method and finite element method. Most flows in the engineering problems have rather arbitrary geometries. When the practical usage of calculation methods in arbitrary geometry is considered, the finite element method appears

to be the natural choice. However, the finite element method still has unresolved difficulties in the simulation of fluid flow phenomena although this method is widely used in other areas such as solid mechanics. By its rather simplicity and efficiency, the finite difference method has been the most widely used in the computational fluid dynamics. Among the various finite difference methods, the method developed by Patankar and Spalding [1,2] seems to be the most popular in the calculation of incompressible flows.

In addition to the finite difference method and the finite element method, a finite analytic method has been developed by Chen and Chen [3]. This method has been tested for the several practical engineering problems [4,5,6] and proved to be accurate and stable. In the finite analytic method, the algebraic representation of a certain nodal value with neighbouring nodal values are obtained by a local analytic solution of the governing equation. The novel nature of this method is that the numerical false diffusion problem in solving the Navier-Stokes equations, which is considered as the most serious problem for nearly all the numerical schemes of finite difference method, is satisfactorily minimized by using all the surrounding nodal points. The well developed calculation procedure for the finite difference method in

solving the Naviers-Stokes equations can be readily applied to finite analytic method without any difficulties. Both the finite analytic method and finite difference method are used for the several calculations presented in this study. Comparisons of predictions by these two methods are given for certain problems.

Recently, several calculation methods have been developed that employ boundary-fitted, curvilinear coordinate to appropriately handle fluid problems in complex geometries. The technique of generating an appropriate boundary fitted curvilinear coordinate and the development of related calculation procedure associated with using the curvilinear coordinate system have been subjects of numerous studies. Although the calculation methods for incompressible flow in the orthogonal grid system have been well developed and refined, those for non-orthogonal grid system seem to be not yet fully established. Detailed discussion of currently available calculation methods for incompressible flow associated with using the curvilinear coordinate system will be given in the following section.

Due to its importance in the engineering application, several numerical and experimental studies of laminar and turbulent flows past two dimensional or axisymmetric



bodies have been made during the last decade. Since this problem includes the leading edge interaction, boundary layer development on the body, the trailing edge interaction and wake development, it has been subject of testing of several calculation methods and turbulence models. At present, several reliable experimental informations are available which can be used to verify the accuracy of the calculation schemes or turbulence models. Detailed discussion of previous numerical and experimental studies is given in the chapter where each specific problem is mentioned.

Previous numerical and experimental studies indicate that although the first order boundary layer theory adequately describes flow fields over the middle of body, it fails to describe flow fields near the leading and trailing edge of a body and near the tail region of axisymmetric bodies. Two calculation methods have been widely used in order to appropriately handle these flow situations. The first method is the viscous-inviscid interaction method in which boundary layer solutions are matched with the external inviscid flows by interactive means. The second approach is the global numerical solutions of elliptic or partially parabolic form of Navier-Stokes equations. The latter approach is more attractive since it does not require another numerical

solution of inviscid flow and is adopted in the present study.

In many previous numerical studies of flow past ship bodies, calculations are started either at the middle of the body or at the trailing edge of the body with upstream conditions specified by well documented profiles from boundary layer theory or experimental data. Since these calculations do not include the leading edge interaction and initial development of boundary layer over the body, the resulting solutions are dependent on the upstream conditions or the location of upstream solution boundary. A more reliable numerical solution may be obtained by including the leading edge within the solution domain with prescribing uniform flow conditions at the inlet and with the solution domain large enough to capture the whole viscous-inviscid interaction. However, inclusion of leading edge in the solution domain may cause the difficulties of turbulence modelling in the transition region for turbulent flow calculation. In the present study, the upstream solution boundary is made to locate either at the uniform flow region far upstream of body or at the middle of the body according to the objective of the problem considered.

Since the problems considered in the present study have rather arbitrary geometries which require the use of boundary-fitted curvilinear coordinate system, a brief review of methods for the generation of boundary-fitted coordinates and available calculation methods for incompressible flow in arbitrary geometry is presented. A brief discussion of available turbulence models will be also presented.

### 1.3 Generation of Boundary-Fitted Coordinates

A boundary-fitted coordinate system is defined as a curvilinear coordinate system in which the boundaries of the physical domain coincide with a curvilinear coordinate line or surface. The generation of an appropriate computational grid is very important in order to achieve the accuracy and better convergence of numerical solution. Boundary-fitted curvilinear coordinates can be generated orthogonally or non-orthogonally. The orthogonal grid has the advantages that the transformed partial differential equations are simpler as the non-orthogonal terms do not appear. This advantage results in fewer computing time, fast convergence as well as better stability and accuracy of the solution. However, orthogonal coordinates are often difficult to generate and it is sometimes impossible

to generate the orthogonal coordinate for the three dimensional situations. An additional disadvantage for the generation of orthogonal coordinate is the fact that there is no control on the location of the grid points along the boundary which may deteriorate the grid spacing near the solid boundary where the appropriate grid spacing is required. Non-orthogonal coordinates are much easier to generate and can be produced also for three dimensional situations. Further, concentration of grid lines in the regions where good resolution is required is much easier to achieve. However, non-orthogonal coordinate introduce additional terms such as cross derivatives which may reduce both convergence and stability of numerical solution and, thereby, increase the computing time.

There exist several ways of the generating boundary-fitted coordinates. The available grid generation techniques can be broadly classified into two categories, algebraic methods and partial differential equation methods. The multi-surface technique of Eiseman [7] is a typical method for algebraic method in which simple functions are combined to generate grids for complex geometries. In this approach, the grid is generated by joining corresponding points on the inner and outer boundaries by polynomial curves. In general, the algebraic grid generation techniques are very attractive

by the reasons that they do not require a numerical solution of partial differential equations and a precise control over the grid spacing is possible. However, when this method is applied to a complex geometry, non-smoothing grids may be produced.

Boundary-fitted coordinates can also be generated by the solution of partial differential equations in which the dependent variables are the grid coordinates in the physical domain. Conformal mapping is a special case of grid generation technique using the Laplace equation. Much of the works have been done by Thompson and his coworkers [8]. The spacing between the grid lines is strongly dependent on the equations being solved and weakly dependent on the boundary point distributions. The equations can be easily changed by varying the values of the source term or the grid control functions in the Poisson equations. These grid control functions can be selected to appropriately attract the coordinate line toward particular grid points or grid lines or repel the coordinate lines. Thompson et al. [8] have suggested a series of exponential functions as source terms. However, the selection of grid control function is quite dependent on the problem being solved.

In the present study, all the numerical grids are generated by the solution of the Poisson type of partial differential equations with grid control functions specified properly depending on the physical situations being solved. Details of grid generation techniques adopted in the present study are given in chapter-II.

#### I.4 Calculation Methods for Incompressible Flow in Complex Geometries

##### I.4.1 Calculation Methods for Incompressible Flow in Orthogonal Coordinate System

The computational method for the prediction of fluid flow in the orthogonal grid system is well developed and have been widely used for a variety of problems. The orthogonal grid based calculation method is not much depart from that for Cartesian counterparts. The well developed calculation procedure for Cartesian coordinate system can be easily modified with a proper specification of geometric coefficients and source terms. Most of previous calculations use the staggered grid arrangement and the velocity components normal to the control volume surfaces (contravariant velocity components) are selected as the dependent variables in the momentum equations.

Pope [9] used such a orthogonal based calculation procedure to compute turbulent recirculating flows in diffusers, Gosman and Rapley [10] applied a similiar method to the calculation of fully developed flow in the ducts of arbitrary cross section. Recently, Raithby et al. [11] have presented a new calculation procedure in which the stresses are retained in the discretization equation instead of the usual procedure of substituting them in terms of strain rates.

However, the orthogonal coordinate based methods have limitations in the application due to the difficulty of generating appropriate numerical grid. For the sake of general application, the calculation methods for non-orthogonal coordinate system should be investigated.

#### I.4.2 Calculation Methods for Incompressible Flow in Non-orthogonal Coordinate System

Unlike the orthogonal coordinate based methods, several calculation methods for the non-orthogonal coordinate system have been proposed during the late 1970s and during the 1980s. The differences among these methods are the choice of dependent variables in the momentum equations, the grid arrangements and the treatment of pressure and velocity coupling. One interesting fact is that none of these methods have superiority over other

methods to justify its general usage. Each of the methods has its own merits and demerits which should be judged in terms of the problem being considered. As compared with the method based on the orthogonal coordinate system, these methods are still in a developing stage. Among the several calculation methods based on non-orthogonal coordinate system, five typical methods which use different grid arrangements or dependent variables as shown in Fig.I-1 are discussed below.

#### Non-Staggered Grid Method

In the first grid arrangement, which is shown in the Fig.I-1-a, all the dependent variables are computed and stored at the same location and the Cartesian velocity components are used as dependent variables in the momentum equation. This method has been named differently according to the author's choice. For example,  $P - \Omega$  scheme by Hsu [12]; Momentum Interpolation Method by Majumdar [13]; Pressure Weighted Interpolation Method by Miller and Schmidt [14]; Collocation Method by Peric et al. [15]; and, Non-Staggered Grid Method by Rhie and Chow [16]. Since this method uses Cartesian velocity components as dependent variables for the momentum equation, the governing equation is simple and the curvature terms which are extremely grid dependent can be



avoided. Moreover, a strongly conservative form of transport equation is possible which is a desired characteristics for finite volume method. As is discussed in Patankar [2], this grid arrangement may give rise to a checkerboard pressure oscillation. To overcome this checkerboard splitting of pressure field, a special kind of interpolation method for evaluating velocity components at the control volume cell surface has been devised. All the transport equations are solved at the center point of control volume cell and the cell surface velocities for the derivation of pressure or pressure correction equation are obtained by a linear interpolation using neighbouring nodal values except the pressure gradient term in the momentum equation is evaluated by the method employed in the staggered grid approach to prevent the splitting of the pressure field. Thus, this method indirectly uses the staggering idea and it was questioned that this method can be named as non-staggered method [17]. Since all the variables are calculated at the same point, the coefficients of algebraic equations are the same for all the variables which may reduce the computer storage and computing time, especially for the three dimensional situations. This feature will lead to more cost effective calculation when the multigrid method is employed. The earlier method by Rhie and Chow [16] or Peric [18] has

been blamed for the fact that the resulting converged solution is quite relaxation factor or time step dependent. Recently, Majumdar [13,19] and Miller and Schmidt [14] removed this problem by a rigorous derivation. This method has been successfully applied to a variety of engineering problems by many researchers. However, Miller and Schmidt [14] find that this method may result in a physically unrealistic solution in regions where there is a strong pressure gradient. This method also suffers from the lack of diagonal dominance of pressure correction equation when it is applied to strongly non-orthogonal grid. This method has been applied to a variety of problems; Rhie and Chow [16], Rhie and coworkers [20,21], Peric [18], Han [22], Majumdar [13,19], Majumdar et al. [17], Rodi et al. [23], Miller and Schmidt [14] and Peric et al. [15]. An excellent discussion of this method is given in Majumdar et al. [17].

#### One Velocity Staggered Grid Method

Shyy et al. [24] developed a calculation method employing the conventional staggered grid usually adopted in the Cartesian coordinate system. As shown in the Fig.I-1-b, the dependent variables in the momentum equations are the Cartesian velocity components and only

one velocity component is stored at the each cell surface. Since only one velocity component is available at each cell surface, the other unavailable velocity component which is necessary to compute the mass flow rate is usually obtained through the linear interpolation of neighbouring velocity components which are obtained in the previous iteration. In addition to this defficiency, further approximations have to be made, for example neglection of non-orthogonal pressure gradient terms, to provide the diagonal dominant pressure correction equation in strongly non-othorgonal grid. A more serious drawback of this method is that if the flow direction is normal to the direction of velocity component stored at the cell surface, zero convective mass fluxes across the control volume faces may occur and the pressure gradient driving the velocity components will be completely wrong, which may cause undesired pressure splitting. However, use of this method is justified when the flow direction is not much deviated from the direction of velocity components stored at the cell surface. Due to the simplicity of this method, many researchers used this method for the problem to which this method can be applied. For example, Braaten and Shyy [25], Nakayama [26], Ha [27], Chen and Patel [4], Cheng and Chen [5], Obashi and Chen [6], Patel et al. [28] and Choi and Chen [29].

### Two Velocities Staggered Grid Method

Discussions of the previous method suggest that in order to calculate flow fields around arbitrary geometries, both of the Cartesian velocity components should be stored at each cell surface as shown in Fig.I-1-c, while pressure and turbulent quantities are stored at the center point of control volume cell. Maliska and Raithby [30] adopted this configuration in the development of three dimensional parabolic calculation procedure for the fluid flow in the arbitrary cross sectioned ducts. This method is also not free from difficulties. In order to obtain the diagonal dominant pressure correction equation in the strongly non-orthogonal grid, the non-orthogonal pressure gradient term should be neglected. Storing the two velocity components at each control volume surface requires a complex programming and more computer storage and more computing time. These disadvantages became more grave in three dimensional situations. This method is not commonly used due to forementioned disadvantages.

### Algebraic Manipulation Method

An alternative to the Cartesian velocity components is to use the curvilinear velocity components as the

dependent variables in the momentum equations. The contravariant or the covariant velocity components will be the natural choices due to their relationship with the grid lines. Karki and Patankar [31] developed a calculation method using covariant velocity components as the dependent variables in the momentum equations. The discretization equations and source terms for covariant velocity components were obtained by the algebraic manipulation of discretization equation and source terms based on the Cartesian velocity components. As shown in the Fig.I-1-e, each covariant velocity component is stored at the control volume cell surface using staggering idea. The usage of covariant velocity components as the dependent variable in the momentum equation enables the strongly diagonal dominant pressure correction equation even in the strongly non-orthogonal grid situations. The complicated tensor algebra is avoided through a novel algebraic manipulation of discretization equation and the resulting algebraic equations seem to be cost effective and less grid dependent. The success of this method is largely dependent on the accuracy and grid dependency of the algebraic manipulation of source term of momentum equation which is considered as the most serious problem when the contravariant or covariant velocity components are adopted as a dependent variables. This method should

be tested for complicated three dimensional situations to justify its general usage. This method has been successfully applied to calculation of compressible and incompressible in two dimensional complex geometries [32,33]. Detailed study of this method is provided by Karki [34].

It is worthy to mention the calculation method based on the full transformation of dependent variables for momentum equations using a complicated tensor algebra. Demirdzic et al. [35] developed such a calculation method using contravariant velocity components as dependent variables for momentum equations. Fig.I-1-d shows that the only one contravariant velocity component is stored at the each control volume cell surface and the staggering idea is adopted. This method requires a complicated tensor algebra and more storages for the geometric coefficients. Moreover, the resulting governing equation is extremely complicate and the numerical solution is quite grid dependent. This fact restricts the usage of this method in certain situations. Like all the methods discussed previously, this method also gives a nine point pressure correction equation which lacks diagonal dominance in strongly non-orthogonal grid situations if the non-orthogonal pressure gradient terms are not neglected. The advantage of smaller numerical diffusion

in this approach must be judged against the serious disadvantages mentioned above. This method has been used by few people. For example, Demirdzic et al. [35], Demirdzic [36], Gal-Chen et al. [37], Stern et al. [38] and Richmond [39]. An excellent study of this method is given by Demirdzic [36].

Methods of Vanka et al. [40] and Faghri et al. [41] deserve to receive attention. However, these methods are also not free from many difficulties mentioned in the previous methods and are not commonly used by many researchers.

The one-velocity staggered grid method is adopted for the most of calculations of the present study since this method is very easy to program and the flow direction is not much deviated from the direction of velocity component stored at cell surface with an appropriate grid generation. The two-velocities staggered grid method is also used for some calculations in the present study with some modifications made from the original method of Maliska and Raithby [30].

### 1.5 Turbulence Models

During the last decade, many turbulence models ranging from the Prandtl mixing-length model [42] to the k

-  $\epsilon$  turbulence model [43] were used for a variety of practical calculations and tested in order to find out their range of applicability. Some progress has been made, but to date no model has been found to be both accurate and general. The models widely used in applications have been based on the Boussinesq (1877) assumption which relates the turbulent stresses to mean rates of strain by

$$-\overline{u_i u_j} = \nu_t \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} k \quad (I-1)$$

where  $\nu_t$  is turbulent eddy viscosity and  $k$  is turbulent kinetic energy.

For a suitable characteristic length scale  $l_t$  and velocity scale  $v_t$ , dimensional analysis suggests that turbulent eddy viscosity  $\nu_t$  can be evaluated as

$$\nu_t = v_t l_t \quad (I-2)$$

The difference among the existing models based on the Boussinesq assumption is how velocity scale  $v_t$  and length scale  $l_t$  are specified. Experimental evidence indicates that the Boussinesq assumption is reasonably valid in many flow circumstances with exception of some flow situations. Models based on the Boussinesq assumption are frequently called eddy viscosity model.



Turbulence models are often classified according to the number of supplementary partial differential equations that must be solved in order to supply the modelling parameters. Since details of these models are available in the various literatures, for example Chen [44], only a brief discussion of current available turbulence models will be given.

#### I.5.1 Zero Equation Turbulence Model

In the zero equation model or algebraic model, such as proposed by Cebeci and Smith [45] or Baldwin and Lomax [46], the eddy viscosity is prescribed from length scale and velocity gradient.

$$v_t = (l_t)^2 \left| \frac{\partial U}{\partial y} \right| \quad (I-3)$$

The length scale  $l_t$  is usually obtained from the empirical correlations developed for the boundary layers. The algebraic models have proven to be accurate and reliable for relatively simple flows. The disadvantage of this model is the lack of generality that the length scale which is needed for specification of eddy viscosity is function of various parameters and it is not easy to specify an appropriate length scale over the whole flow field, especially for the separated flow region or the wake region. However, these models can be employed all

the way to the wall where the viscosity effect is important and the numerical solutions of partial differential equations are not needed for the specification of length scale.

### I.5.2 One Equation Turbulence Model

This model is based on the suggestion of Prandtl and Kolomogorov that the turbulent velocity scale is proportional to the square root of turbulent kinetic energy  $k$ , implying;

$$v_t = C_\mu k^{1/2} l_\mu \quad (\text{I-4})$$

This model requires the numerical solution of turbulent kinetic energy equation which can be derived from the Navier-Stokes equations. The turbulent kinetic energy equation contains a diffusion term, a generation term and a dissipation term which must be modelled through additional assumptions such as;

$$\epsilon = C_D \frac{k^{3/2}}{l_\epsilon} \quad (\text{I-5})$$

which requires specification of another length scale  $l_\epsilon$ .

There exist several one equation models [47,48,49] and the differences among the these models are mainly how the length scales  $l_\mu$ ,  $l_\epsilon$  are specified. In general, the

predictions of the one equation model have been marginally better than those obtained from zero equation models.

### I.5.3 $k - \epsilon$ Turbulence Model

The  $k - \epsilon$  turbulence model [43] is usually called two equation model. In this model, the eddy viscosity is related to the turbulent kinetic energy  $k$  and its dissipation rate  $\epsilon$  ;

$$v_t = C_\mu \frac{k^2}{\epsilon} \quad (\text{I-6})$$

The turbulent dissipation rate  $\epsilon$  is assumed to be related to the length scale through

$$l_t = C_D \frac{k^{3/2}}{\epsilon} \quad (\text{I-7})$$

The turbulent kinetic energy  $k$  and its dissipation rate  $\epsilon$  are determined from two modelled equations. Thus the eddy viscosity and the length scale are obtained from the numerical solutions of partial differential equations. Since this model does not require algebraic specification of length scale, it can be applied to several flow situations.

The  $k - \epsilon$  turbulence model can only be applied to the fully turbulent region and calculations of the viscosity-affected near wall region is avoided by the use of wall functions. The wall function method [43] has been found

to be quite successful for attached flow, it is not suitable for some flow situations. The wall functions relating the velocity and turbulent quantities at the second grid point to friction velocity heavily lean on the assumptions of a logarithmic velocity distribution and the validity of local equilibrium of turbulence. These assumptions are not valid in the separated flow region. Furthermore, the extension of wall function method to unsteady flow and three dimensional flow requires additional assumptions which are not substantiated by experiment or theory [50]. An alternative to the use of wall function method will be to employ the turbulence model which are valid all the way to the wall.

#### I.5.4 Low Reynolds Number Turbulence Model

Several low Reynolds number extensions of the  $k - \epsilon$  turbulence model [51,52,53] which can be applicable to the viscous affected layer have been proposed. These models introduce damping functions in order to achieve the observed reduction of turbulent transport very near the wall. However, these models have an undesirable feature of requiring very high numerical resolution near the wall region because of the steep gradient of the dissipation rate  $\epsilon$ . It was found that all these models perform rather poorly in adverse pressure gradient boundary layers [54]

and in separated flows [55]. Patel et al. [50] reviewed the several available low Reynolds number  $k - \epsilon$  turbulence models and concluded that none of these models can be used confidently even in the boundary layer calculations.

#### I.5.5 Two-Layer Turbulence Model

In order to reduce the computational effort and to preserve numerical stability, the well established length scale distribution near the wall region can be introduced to the calculation of the viscosity-affected near wall region instead of using the wall function method while the outer, fully turbulent region is resolved by use of the  $k - \epsilon$  turbulence model. This model is named as two-layer model and it is usually used with the combination of the  $k - \epsilon$  turbulence model and zero equation or one equation turbulence model. Some successes in the prediction of complex turbulent flows by use of this model is reported [56,57,58,59,60].

In the present study, the  $k - \epsilon$  turbulence model with wall function method will be employed for most of turbulent flow calculations while the two-layer model is applied to some flow situations involving the pressure gradient and body curvature effects to improve the accuracy of the calculations.

### I.6 Outline of Study

In chapter II, mathematical and numerical formulations used in the present study are presented. The governing equations, boundary conditions, methods of generation of boundary-fitted coordinates and discretization schemes are discussed.

In chapter III, numerical aspects are discussed. These include the coupling between the continuity equation and momentum equations, numerical algorithm and overall solution procedures.

Chapters IV to VII are devoted to the numerical calculations which include development of the wake function method for the prediction of flat plate wake, prediction of laminar and turbulent flows past a finite flat plate, numerical solutions of turbulent flow past axisymmetric bodies by wall function method and by two layer model, prediction of turbulent flow past finite axisymmetric bodies. Some comparisons of the predictions by finite analytic method with those by finite volume method are presented. All the calculated results are compared with available experimental data to demonstrate the applicability and accuracy of the numerical scheme.

In chapter VIII, a brief summary of the present study is given and some suggestions for future work are discussed.

## CHAPTER II

### FORMULATION OF PROBLEM

#### II.1 Introduction

In this chapter, the mathematical and numerical formulation of the present study is presented. This chapter starts with the mathematical formulation of the problem which includes governing equations, boundary conditions and solution domains. Then the grid generation techniques and discretization methods employed in the present study are presented.

The solution domains are made large enough to capture the whole important feature of flow field. The numerical grids are generated through the numerical solutions of Poisson equations in which the dependent variables are the physical coordinate of solution domain. The governing equations are written in the transformed domain using the partial transformation where the original Cartesian or cylindrical velocity components are left as dependent variables for momentum equations. These transformed governing equations are discretized using both finite analytic method and finite volume method.



## II.2 Governing Equations

The continuity equation and general form of transport equations for the incompressible flow in Cartesian or in cylindrical coordinate system with the  $k - \epsilon$  turbulence model can be written as follow;

$$\frac{\partial}{\partial x}(\rho r u) + \frac{\partial}{\partial y}(\rho r v) = 0 \quad (\text{II-1})$$

$$\begin{aligned} & \frac{\partial}{\partial t}(\rho r \phi) + \frac{\partial}{\partial x}(\rho r u \phi) + \frac{\partial}{\partial y}(\rho r v \phi) \\ & = \frac{\partial}{\partial x}\left(r \Gamma_{\phi} \frac{\partial \phi}{\partial x}\right) + \frac{\partial}{\partial y}\left(r \Gamma_{\phi} \frac{\partial \phi}{\partial y}\right) + r S_{\phi} \end{aligned} \quad (\text{II-2})$$

In these equations,  $\phi$  represents the general dependent variables ( $u, v, k, \epsilon$ ),  $\rho$  is the density of fluid,  $\Gamma_{\phi}$  is the effective diffusion coefficient, ( $u, v$ ) are the velocity components in ( $x, y$ ) directions and  $S_{\phi}$  denotes the source term for the variable  $\phi$ . For two dimensional flow,  $r = 1$  and for axisymmetric flow,  $y = r$ . Details of the definition of all the variables are given in Table 1.

## II.3 Solution Domains and Boundary Conditions

### II.3.1 Solution Domains

All the calculations presented in the present study can be broadly classified into two categories; half body calculations and full body calculations. The difference between these two calculations is the location of the upstream solution boundary as shown in Fig.II-1. In the half body calculations, calculations start at the middle of body with inlet conditions specified by the well documented profiles from the boundary layer theory. In the full body calculations, calculations start far upstream of body with inlet conditions specified by uniform flow conditions. The solution domain in the normal direction and in the downstream direction is made large enough to capture the whole viscous-inviscid interaction and wake development. The location of upstream inlet solution boundary in the full body calculations is also made far away from the body to capture the whole leading edge interaction. The details of the solution domains used in the present calculations are given in Tables 2 to 6.

### II.3.2 Boundary Conditions

The boundary conditions for all the problems considered in the present study can be stated as follows;

Inlet or Upstream Boundary (  $x = x_i$  )

The upstream inlet solution boundary is located either at the middle of the body or at the uniform flow region far upstream of the body. The u-velocity component and turbulent quantities  $k$ ,  $\epsilon$  are prescribed from the simple flat-plate correlations for the half body calculations and from uniform flow conditions for the full body calculations. For v-velocity component, the condition of  $v_x = 0$  is specified for all the calculations.

Outlet or Downstream Boundary (  $x = x_d$  )

The downstream outlet condition is placed in the wake region far downstream of body. The zero pressure-gradient condition  $p_x = 0$  is specified at the outlet boundary. For the exit boundary conditions for transport quantities (  $u$ ,  $v$ ,  $k$ ,  $\epsilon$  ),  $u_x = v_x = k_x = \epsilon_x = 0$  conditions are specified.

Body Surface

The no-slip boundary condition (  $u = v = 0$  ) is specified for all the laminar flow calculations and for

the turbulent flow calculations when the two layer model is employed to calculate the flow field all the way to the wall. For all the other turbulent flow calculations, the wall function method is employed in order to avoid the use of large number of grids to resolve the steep gradient in the near wall region and uncertainties of the turbulence model in the viscous-affected region.

Upper Boundary (  $y = y_U$  )

The upper boundary is placed at a large distance from the body where the flow is assumed to be uniform. The uniform boundary condition is ;

$$u = 1, p = 0, k_y = \epsilon_y = 0. \quad (\text{II-3})$$

The v-velocity component in this boundary is obtained from the law of conservation of mass within the computational control volume cell during the solution process.

Wake Centerline (  $y = 0$  )

The boundary conditions along the wake centerline are the symmetry conditions and are specified as

$$v = 0, u_y = k_y = \epsilon_y = 0. \quad (\text{II-4})$$

#### II.4. Numerical Grid Generation

The boundary-fitted coordinate generation technique proposed by Thompson et al. [8] is employed for the generation of numerical grid in the present study. The complex solution domain in physical coordinates ( x, y ) is transformed to a simple rectangular domain in non-orthogonal, body-fitted coordinates (  $\xi$ ,  $\eta$  ) by the solutions of the Poisson equations;

$$\frac{\partial^2 \xi}{\partial x^2} + \frac{a_c}{r} \frac{\partial \xi}{\partial y} + \frac{\partial^2 \xi}{\partial y^2} = f^1(\xi, \eta) \quad (\text{II-5})$$

$$\frac{\partial^2 \eta}{\partial x^2} + \frac{a_c}{r} \frac{\partial \eta}{\partial y} + \frac{\partial^2 \eta}{\partial y^2} = f^2(\xi, \eta) \quad (\text{II-6})$$

where  $f^1$  and  $f^2$  are the grid control functions which are chosen priori to obtain the desired distribution of grid points in the physical ( x, y ) domain. In these equations,  $r = 1$ ,  $a_c = 0$  for a Cartesian coordinate system and  $r = y$  and  $a_c = 1$  for a cylindrical coordinate system. The origin of coordinate system is located at the leading edge of body.

The numerical grid is generated by solution of inverted forms of these equations where the dependent variables are the physical coordinates of solution domain.

$$g^{11} \frac{\partial^2 x}{\partial \xi^2} + g^{22} \frac{\partial^2 x}{\partial \eta^2} + 2g^{12} \frac{\partial^2 x}{\partial \xi \partial \eta} + f^1 \frac{\partial x}{\partial \xi} + f^2 \frac{\partial x}{\partial \eta} = 0 \quad (\text{II-7})$$

$$g^{11} \frac{\partial^2 y}{\partial \xi^2} + g^{22} \frac{\partial^2 y}{\partial \eta^2} + 2g^{12} \frac{\partial^2 y}{\partial \xi \partial \eta} + f^1 \frac{\partial y}{\partial \xi} + f^2 \frac{\partial y}{\partial \eta} = \frac{a_c}{r} \quad (\text{II-8})$$

where

$$g^{11} = r^2 (x_\eta^2 + y_\eta^2) \quad (\text{II-9})$$

$$g^{22} = r^2 (x_\xi^2 + y_\xi^2) \quad (\text{II-10})$$

$$g^{12} = -r^2 (x_\xi x_\eta + y_\xi y_\eta) \quad (\text{II-11})$$

$$g = r^2 (x_\xi y_\eta - x_\eta y_\xi)^2 \quad (\text{II-12})$$

The grid generation techniques outlined below are essentially the same as those reported in Chen and Patel [4] or Cheng and Chen [5]. However, some modifications are made for certain problems.

#### II.4.1 Generation of Numerical Grids for Flat Plate

Non-uniform rectangular numerical grids are generated for the calculation of laminar and turbulent flow past a flat plate with a proper grid concentration made near the wall and near the leading and trailing edges. The numerical grid in the normal direction is generated algebraically instead of solving the differential equation. The first few grids near the plate are generated uniformly and next the other grids are generated

non-uniformly with the grid expansion ratio about 1.2 to 1.3. The size of the first grid in the normal direction from the plate is adjusted according to the physical situations being considered. The numerical grid in x-direction is generated by the numerical solution of the following differential equation which is obtained from Eq. (II-7).

$$g^{11} \frac{\partial^2 x}{\partial \xi^2} + f^1 \frac{\partial x}{\partial \xi} = 0 \quad (\text{II-13})$$

The grid control function  $f^1$  is selected by following equations which is previously used by Chen and Patel [4] or Cheng and Chen [5];

$$\begin{aligned} f^1/2g^{11} &= A_1 & 0.0 \leq z_1 \leq 0.5 \\ & A_1 \sin(\pi z_1) & 0.5 \leq z_1 \leq 2 \\ & A_2 \sin(\pi z_2) & 0.0 \leq z_2 \leq 1.5 \\ & - A_2 & 1.5 \leq z_2 \end{aligned} \quad (\text{II-14})$$

where

$$z_1 = \frac{\xi-1}{\xi_1-1}, \quad z_2 = \frac{\xi-2\xi_1+1}{\xi_2-2\xi_1+1} \quad (\text{II-15})$$

The terms  $\xi_1$  and  $\xi_2$  correspond to the leading and trailing edge stations as shown in Fig. II-2 and  $A_1$  and  $A_2$

are positive constants which can be adjusted to yield the desired grid concentration near the leading and trailing edges.

The partial view of generated numerical grids used for laminar and turbulent flow calculations are shown in Fig.II-3. As shown in the figures, an appropriate grid concentration was made near the wall region and near the leading and trailing edge regions and the desired grid expansion in the wake region is also achieved.

#### II.4.2 Generation of Numerical Grids for Axisymmetric Bodies

Body fitted coordinates for axisymmetric bodies are generated by the numerical solutions of Poisson equations, Eq.(II-7) and Eq.(II-8). In the present study, the constant-x lines are chosen as constant- $\xi$  lines to facilitate the comparison of numerical results with experimental data. Then, Eq.(II-7) and Eq.(II-8) can be rewritten as following simplified form.

$$g^{11} \frac{\partial^2 x}{\partial \xi^2} + f^{10} \frac{\partial x}{\partial \xi} = 0 \quad (\text{II-16})$$

$$g^{11} \frac{\partial^2 r}{\partial \xi^2} + g^{22} \frac{\partial^2 r}{\partial \eta^2} + 2g^{12} \frac{\partial^2 r}{\partial \xi \partial \eta} + f^{10} \frac{\partial r}{\partial \xi} + F^{20} \frac{\partial r}{\partial \eta} = 0 \quad (\text{II-17})$$

$$\text{where } F^2 = f^2 - \frac{1}{r r_\eta} \quad (\text{II-18})$$



The x-coordinates are algebraically generated instead of solving Eq.(II-16). The x-coordinate over the body and in the wake region near the trailing edge of body (  $x/L < 1.1$  ) is generated uniformly while x-coordinate in the wake region away from the body (  $x/L > 1.1$  ) is generated nonuniformly with grid expansion ratio about 1.2 to 1.3. The grid control function  $f^1$  is then determined from following equation using the x-coordinate already generated.

$$f^1 = -g^{11} \frac{x_{\xi\xi}}{x_{\xi}} \quad (\text{II-19})$$

With  $f^1$  specified by Eq.(II-19), Eq.(II-17) can be solved with a proper specification of grid control function  $F^2$ . The resulting numerical solution gives the grid distribution in the radial direction.

Along the inlet and outlet boundary where the grid is orthogonal, Eq.(II-17) can be written as following.

$$g^{22} \frac{\partial^2 r}{\partial \eta^2} + F^2 \frac{\partial r}{\partial \eta} = 0 \quad \text{at} \quad \xi = 1 \text{ and } \xi = \xi_{\max} \quad (\text{II-20})$$

Thus the grid control function  $F^2$  along these boundaries can be determined from Eq.(II-20) if the r-coordinate at these boundary is provided.

$$F_A(\eta) = F^2(1, \eta) = -g^{22} \frac{r_{\eta\eta}}{r_{\eta}} \Big|_{\xi=1} \quad (\text{II-21})$$

$$F_B(\eta) = F^2(\xi_{\max}, \eta) = -g^{22} \frac{r_{\eta\eta}}{r_\eta} \Big|_{\xi = \xi_{\max}} \quad (\text{II-22})$$

where  $\xi = 1$  is the upstream inlet station and  $\xi = \xi_{\max}$  is the downstream outlet station.

The value of  $F^2$  inside the solution domain can be obtained by a linear combination of  $F_A$  and  $F_B$ .

$$F^2(\xi, \eta) = F_A(\eta) \quad 1 < \xi < \xi_A \quad (\text{II-23})$$

$$F_C(\xi, \eta) \quad \xi_A < \xi < \xi_B$$

$$F_B(\eta) \quad \xi_B < \xi < \xi_{\max}$$

where

$$F_C(\xi, \eta) = \frac{[x(\xi_B) - x(\xi)]F_A(\eta) + [x(\xi) - x(\xi_A)]F_B(\eta)}{[x(\xi_B) - x(\xi_A)]} \quad (\text{II-24})$$

and  $\xi_A$  and  $\xi_B$  are determined from the body shape. In the case of half body calculation,  $\xi_A$  is placed at the station where radius of body is initially changed and  $\xi_B$  is placed at the trailing edge of body as shown in Fig.II-2. However, the specification of  $F^2(\xi, \eta)$  is quite problem dependent and can be adjusted to obtain the desired grid distribution depending on the shape of geometry.

With the specification of proper grid control functions  $f^1$  and  $F^2$ , Eq.(II-17) is solved by finite

difference method. Eq. (II-17) is discretized as following using the exponential scheme of Spalding [4].

$$\begin{aligned}
 & ( 2a g^{11} \coth a + 2b g^{22} \coth b )_{\xi,\eta} r_{\xi,\eta} \\
 & = ( 2a g^{11} \operatorname{csch} a )_{\xi,\eta} ( e^{-a} r_{\xi+1,\eta} + e^a r_{\xi-1,\eta} ) \\
 & + ( 2b g^{22} \operatorname{csch} b )_{\xi,\eta} ( e^{-b} r_{\xi,\eta+1} + e^b r_{\xi,\eta-1} ) \\
 & + 2 (g^{12})_{\xi,\eta} ( r_{\xi+1,\eta+1} + r_{\xi-1,\eta-1} - r_{\xi-1,\eta+1} - r_{\xi+1,\eta-1} )
 \end{aligned} \tag{II-25}$$

where

$$2a = - \frac{f_{11}^1}{g^{11}} \Big|_{\xi,\eta} \quad \text{and} \quad 2b = - \frac{F_{22}^2}{g^{22}} \Big|_{\xi,\eta} \tag{II-26}$$

The above system of algebraic equation is solved by the tridiagonal-matrix algorithm.

The body fitted coordinate system for full body calculation was obtained with a slightly different specification of grid control function  $F^2(\xi, \eta)$ . In order to attract grid lines toward body surface in the leading edge region of body, the following grid control function  $F^2(\xi, \eta)$  was devised.

$$\begin{aligned}
 F^2(\xi, \eta) & = [ (d(\xi_B) - d(\xi)) F_A(\eta) + (d(\xi) - d(\xi_A)) F_B(\eta) ] \\
 & \quad / [d(\xi_B) - d(\xi_A)]
 \end{aligned} \tag{II-27}$$

$$\text{where } d(\xi) = r(\xi, \eta_{\text{ref}}) - r(\xi, 1) \quad (\text{II-28})$$

In Eq.(II-27), the station  $\xi = \xi_A$  is the station where  $d(\xi)$  is minimum and the station  $\xi = \xi_B$  is the station where  $d(\xi)$  is maximum as shown in Fig.II-2. The grid control functions at these stations,  $F_A(\eta)$ ,  $F_B(\eta)$  are evaluated using Eq.(II-21) and Eq.(II-22) as the same way as done in half body calculation. The normal reference station  $\eta = \eta_{\text{ref}}$  which is shown in Fig.II-2 was adjusted to obtain the desired grid concentration toward body and  $\eta_{\text{ref}} = \eta_{\text{max}} - 5$  is used for most calculations.

As seen from the Fig.II-4 and Fig.II-5, the boundary-fitted coordinates generated by above method evolve smoothly from the body into the wake and the desired grid expansion in the tail region and in the wake region is obtained.

## II.5 Finite Analytic Method

### II.5.1 Partially Parabolic Form of Finite Analytic Method in the Cartesian Coordinate system

The non-dimensionized form of general transport equation in the Cartesian coordinate system can be written as follows;

$$\frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y} = \frac{1}{R_\phi} \left( \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \right) + S^\phi \quad (\text{II-29})$$

where  $S^\phi$  is source term for variable  $\phi$  and  $R_\phi$  is effective Reynolds number for variable  $\phi$  defined as

$$R_u = R_v = \frac{1}{(1/Re + \nu_t)} \quad (\text{II-30})$$

$$R_k = \frac{\sigma_k}{(\sigma_k/Re + \nu_t)} \quad (\text{II-31})$$

$$R_\epsilon = \frac{\sigma_\epsilon}{(\sigma_\epsilon/Re + \nu_t)} \quad (\text{II-32})$$

If we locally linearize the convective velocity in the governing equation by the value at calculation point P as shown in Fig.II-6 and neglect the diffusion of transport quantities in the flow direction, the governing equation can be approximately written in a following simplified form:

$$\frac{\partial^2 \phi}{\partial y^2} = 2B \frac{\partial \phi}{\partial y} + F \quad (\text{II-33})$$

where

$$2B = (R_\phi v)_P \quad (\text{II-34})$$

$$F = (D_\phi)_P (\phi_P - \phi_W) + \frac{(E_\phi)_P}{\Delta t} (\phi_P - \phi_P^{n-1}) - (S_\phi)_P \quad (\text{II-35})$$

and

$$D_\phi = \frac{(R_\phi u)_P}{\Delta x} \quad (\text{II-36})$$

$$E_{\phi} = (R_{\phi})_P \quad (\text{II-37})$$

$$S_{\phi} = (S^{\phi} R_{\phi})_P \quad (\text{II-38})$$

In these equations, the subscript P denotes the calculation point and the superscript n-1 denotes the previous time step value. The unsteady term and the x-derivative of  $\phi$  are approximated by backward and upwind finite difference respectively. Eq.(II-30) is solved analytically with boundary conditions specified by following equations.

$$\phi(h) = \phi_N, \quad \phi(-k) = \phi_S \quad (\text{II-39})$$

Evaluation of  $\phi$  at calculation point P gives following algebraic equations.

$$\phi_P = A_N \phi_N + A_S \phi_S - C_P F \quad (\text{II-40})$$

where

$$A_S = \frac{\exp(2Bh) - 1}{\exp(2Bh) - \exp(-2Bk)} \quad (\text{II-41})$$

$$A_N = \frac{1 - \exp(-2Bk)}{\exp(2Bh) - \exp(-2Bk)} \quad (\text{II-42})$$

$$C_P = \frac{hk}{2} \left( \frac{A_N}{Bk} - \frac{A_S}{Bh} \right) \quad (\text{II-43})$$

If the node P is placed on the wake centerline where the normal convective velocity component v is equal to zero, then

$$A_N = \frac{k}{h + k} \quad (\text{II-44})$$

$$A_S = \frac{h}{h + k} \quad (\text{II-45})$$

$$C_P = \frac{hk}{2} \quad (\text{II-46})$$

With the source term linearized as

$$(S_\phi)_P = S_\phi^P \phi_P + S_\phi^C \quad (\text{II-47})$$

the following four point partially parabolic discretization formula can be obtained by manipulating Eq. (II-40), Eq. (II-35) and Eq. (II-47).

$$A_P \phi_P = A_N \phi_N + A_S \phi_S + A_W \phi_W + b_\phi \quad (\text{II-48})$$

where

$$A_W = C_P (D_\phi)_P \quad (\text{II-49})$$

$$A_P = 1 + C_P \left( (D_\phi)_P + \frac{(E_\phi)_P}{\Delta t} - S_\phi^P \right) \quad (\text{II-50})$$

$$b_\phi = C_P \left( \frac{(E_\phi)_P}{\Delta t} \phi_P^{n-1} + S_\phi^C \right) \quad (\text{II-51})$$

and  $A_N$ ,  $A_S$  and  $C_P$  are given by Eq. (II-41) through Eq. (II-46).

This partially parabolic form of finite analytic formulation may be interpreted as a combination of the upwind scheme in the flow direction and the exponential scheme in the normal direction.

### II.5.2 Elliptic Form of Finite Analytic Method in the Boundary-Fitted Coordinate System

The non-dimensionized governing equations based on the cylindrical coordinate system can be written in the transformed coordinate system as following using the partial transformation where the original orthogonal velocity components are left as dependent variables;

$$[r(b_1^1 u + b_2^1 v)]_\xi + [r(b_1^2 u + b_2^2 v)]_\eta = 0 \quad (\text{II-52})$$

$$g^{11} \phi_{\xi\xi} + g^{22} \phi_{\eta\eta} = 2A_\phi \phi_\xi + 2B_\phi \phi_\eta + E_\phi \phi_t + S_\phi \quad (\text{II-53})$$

where

$$2A_\phi = \frac{R_\phi}{J} r ( b_1^1 u + b_2^1 v ) - f^1 \quad (\text{II-54})$$

$$2B_\phi = \frac{R_\phi}{J} r ( b_1^2 u + b_2^2 v ) - f^2 \quad (\text{II-55})$$

$$E_\phi = R_\phi \quad (\text{II-56})$$

$$S_\phi = R_\phi S^\phi - 2g^{12} \phi_{\xi\eta} \quad (\text{II-57})$$

and

$$b_1^1 = y_\eta, \quad b_2^1 = -x_\eta, \quad b_1^2 = -y_\xi, \quad b_2^2 = x_\xi \quad (\text{II-58})$$

$$J = r ( x_\xi y_\eta - x_\eta y_\xi ) \quad (\text{II-59})$$



If we locally linearize Eq.(II-53) in each numerical element,  $\Delta\xi = \Delta\eta = 1$ , by the value at calculation point P as shown in Fig.II-7 and linearize the source term by Eq.(II-47), one obtains following equation.

$$g_P^{11} \phi_{\xi\xi} + g_P^{22} \phi_{\eta\eta} = 2(A_\phi)_P \phi_\xi + 2(B_\phi)_P \phi_\eta + G \quad (\text{II-60})$$

where

$$G = \frac{(E_\phi)_P}{\Delta t} (\phi_P - \phi_P^{n-1}) + S_\phi^P \phi_P + S_\phi^C \quad (\text{II-61})$$

If we introduce the grid stretching functions such that

$$\xi^* = \frac{\xi}{\sqrt{g_P^{11}}}, \quad \eta^* = \frac{\eta}{\sqrt{g_P^{22}}} \quad (\text{II-62})$$

then Eq.(II-60) can be written in a standard finite analytic form.

$$\phi_{\xi^*\xi^*} + \phi_{\eta^*\eta^*} = 2A \phi_{\xi^*} + 2B \phi_{\eta^*} + G \quad (\text{II-63})$$

where

$$A = \frac{(A_\phi)_P}{\sqrt{g_P^{11}}}, \quad B = \frac{(B_\phi)_P}{\sqrt{g_P^{22}}} \quad (\text{II-64})$$

and the grid size is stretched to

$$\Delta\xi^* = h = \frac{1}{\sqrt{g_P^{11}}}, \quad \Delta\eta^* = k = \frac{1}{\sqrt{g_P^{22}}} \quad (\text{II-65})$$

Equation (II-63) is solved analytically using the method of separation of variable with boundary condition expressed as the combinations of constant, linear and exponential functions involving the nodal values. Evaluation of the analytic solution at the center node then provides a nine point algebraic discretization formula [3].

$$A_P \phi_P = \sum_{n=1}^8 A_{nb} \phi_{nb} + b_\phi \quad (\text{II-66})$$

where

$$A_E = \frac{\exp(-Ah)}{2\cosh(Ah)} * P_B \quad (\text{II-67})$$

$$A_W = \frac{\exp(Ah)}{2\cosh(Ah)} * P_B \quad (\text{II-68})$$

$$A_N = \frac{\exp(-Bk)}{2\cosh(Bk)} * P_A \quad (\text{II-69})$$

$$A_S = \frac{\exp(Bk)}{2\cosh(Bk)} * P_A \quad (\text{II-70})$$

$$A_{SW} = \frac{\exp(Ah+Bk)}{4\cosh(Ah)\cosh(Bk)} * (1-P_A-P_B) \quad (\text{II-71})$$

$$A_{SE} = \frac{\exp(-Ah+Bk)}{4\cosh(Ah)\cosh(Bk)} * (1-P_A-P_B) \quad (\text{II-72})$$

$$A_{NW} = \frac{\exp(Ah-Bk)}{4\cosh(Ah)\cosh(Bk)} * (1-P_A-P_B) \quad (\text{II-73})$$

$$A_{NE} = \frac{\exp(-Ah-Bk)}{4\cosh(Ah)\cosh(Bk)} * (1-P_A-P_B) \quad (\text{II-74})$$

$$A_P = 1 + C_P \left( \frac{(E_\phi)_P}{\Delta t} - S_\phi^P \right) \quad (\text{II-75})$$

$$b_{\phi} = C_P \left( \frac{(E_{\phi})_P}{\Delta t} \phi_P^{n-1} + S_{\phi}^C \right) \quad (\text{II-76})$$

$$C_P = \frac{h \tanh(Ah)}{2A} * (1-P_A) = \frac{k \tanh(Bk)}{2B} * (1-P_B) \quad (\text{II-77})$$

and

$$P_A = 4Ah \cosh(Ah) \cosh(Bk) \coth(Ah) E_2 \quad (\text{II-78})$$

$$P_B = 1 + \frac{Bh \coth(Bk)}{Ak \coth(Ah)} * (P_A - 1) \quad (\text{II-79})$$

with

$$E_2 = \sum_{m=1}^{\infty} \frac{-(-1)^m \lambda_m h}{[(Ah)^2 + (\lambda_m h)^2]^2 \cosh \sqrt{(A^2 + B^2 + \lambda_m^2) k^2}} \quad (\text{II-80})$$

$$\lambda_m h = \left( m - \frac{1}{2} \right) \pi \quad (\text{II-81})$$

For large cell Reynolds numbers, the series summation in  $E_2$  can be avoided by the following asymptotic expressions of  $P_A$  and  $P_B$  based on the theory of characteristics. [4]

$$Ak \coth Ah > Bh \coth Bk;$$

$$P_A = 0, P_B = 1 - (Bh \coth Bk) / (Ak \coth Ah).$$

$$Ak \coth Ah < Bh \coth Bk;$$

$$P_B = 0, P_A = 1 - (Ak \coth Ah) / (Bh \coth Bk) \quad (\text{II-82})$$

### II.5.3 Partially Parabolic Form of Finite Analytic Method in the Boundary-Fitted Coordinate System

The problems considered in the present study are mostly free from separation and have a predominant flow direction. In these problems, diffusion in the flow direction is much smaller than diffusion in the normal direction. Thus the diffusion term in the flow direction can be neglected without loss of accuracy. This partially parabolic form has advantages over the fully elliptic form in the fact that the computing time for the calculation of finite analytic coefficients can be reduced. In the situations when the grid ratio and cell pecelet number become too large, the nine point finite analytic coefficients requires more computational efforts to be evaluated accurately. The partially parabolic form of finite analytic method can be used without any difficulties. The partially parabolic formulation can also be used for the simulation of small separated flow with FLARE approximation [61].

Eq. (II-60) can be approximated using the partially parabolic assumption ( $\phi_{\xi\xi} = 0$ ) and the upwind difference for the convection term in the  $\xi$ -direction as the following form;

$$g_P^{22} \phi_{\eta\eta} = 2(B_\phi)_P \phi_\eta + (D_\phi)_P (\phi_P - \phi_W) + G \quad (\text{II-83})$$

where

$$(D_\phi)_P = 2(A_\phi)_P \quad (\text{II-84})$$

Introduction of the grid stretching functions  $\eta^* = \eta / \sqrt{g_P^{22}}$  leads Eq.(II-83) to the standard finite analytic form.

$$\phi_{\eta^*\eta^*} = 2B \phi_{\eta^*} + F \quad (\text{II-85})$$

where

$$B = \frac{(B_\phi)_P}{\sqrt{g_P^{22}}} \quad (\text{II-86})$$

$$F = (D_\phi)_P (\phi_P - \phi_W) + G \quad (\text{II-87})$$

and the grid size is stretched to

$$\Delta\eta^* = k = \frac{1}{\sqrt{g_P^{22}}} \quad (\text{II-88})$$

Equation (II-85) can be solved analytically using the north and south nodal values as boundary conditions. Evaluation of the analytic solution at the center node then provides a four point algebraic discretization formula.

$$A_P \phi_P = A_N \phi_N + A_S \phi_S + A_W \phi_W + b_\phi \quad (\text{II-89})$$

where

$$A_N = \frac{\exp(-Bk)}{2\cosh(Bk)} \quad (\text{II-90})$$

$$A_S = \frac{\exp(Bk)}{2\cosh(Bk)} \quad (\text{II-91})$$

$$A_w = C_P (D_\phi)_P \quad (\text{II-92})$$

$$A_P = 1 + C_P \left( (D_\phi)_P + \frac{(E_\phi)_P}{\Delta t} - S_\phi^P \right) \quad (\text{II-93})$$

$$b_\phi = C_P \left( \frac{(E_\phi)_P}{\Delta t} \phi_P^{n-1} + S_\phi^C \right) \quad (\text{II-94})$$

$$C_P = \frac{k \tanh(Bk)}{2B} \quad (\text{II-95})$$

## II.6 Finite Volume Method

The strong conservative form of governing equations in the transformed domain can be written as follows;

$$(\rho r U)_\xi + (\rho r V)_\eta = 0 \quad (\text{II-96})$$

$$\begin{aligned} (\rho r U \phi)_\xi + (\rho r V \phi)_\eta &= \left[ \frac{\Gamma_\phi}{J} (D_1^1 \phi_\xi + D_2^1 \phi_\eta) \right]_\xi \\ &+ \left[ \frac{\Gamma_\phi}{J} (D_1^2 \phi_\xi + D_2^2 \phi_\eta) \right]_\eta + J S_\phi \end{aligned} \quad (\text{II-97})$$

where

$$U = (b_1^1 u + b_2^1 v) \quad (\text{II-98})$$

$$V = (b_1^2 u + b_2^2 v) \quad (\text{II-99})$$

and

$$D_1^1 = r^2 (x_\eta^2 + y_\eta^2) \quad (\text{II-100})$$

$$D_2^2 = r^2 (x_\xi^2 + y_\xi^2) \quad (\text{II-101})$$

$$D_2^1 = D_1^2 = -r^2 (x_\xi x_\eta + y_\xi y_\eta) \quad (\text{II-102})$$

In the finite volume method [2], the governing differential equations are integrated over a finite number of control volumes with size,  $\Delta\xi = \Delta\eta = 1$ , as shown in Fig.II-8. Using the Gauss theorem, the volume integral of terms under differential operator can be converted into surface integrals over the four faces of a two dimensional control volume. The resulting balance equations for each control volume and variable  $\phi$  can be expressed as follows.

$$I_e - I_w + I_n - I_s = \int_{\Delta V} J S_\phi d\xi d\eta \quad (\text{II-103})$$

where

$$I_e = (\rho r U \phi)_e - \left[ \frac{\Gamma_\phi}{J} (D_1^1 \phi_\xi + D_2^1 \phi_\eta) \right]_e \quad (\text{II-104})$$

$$I_n = (\rho r V \phi)_n - \left[ \frac{\Gamma_\phi}{J} (D_1^2 \phi_\xi + D_2^2 \phi_\eta) \right]_n \quad (\text{II-105})$$

The volume integral of the source term may be evaluated as follows;

$$\int_{\Delta V} J S_\phi d\xi d\eta = (S_\phi^P \phi_P + S_\phi^C) \Delta V \quad (\text{II-106})$$

By putting the non-orthogonal terms into source term, the Eq.(II-103) can be written as as follows.

$$\begin{aligned}
 & \left[ \rho r U \phi - \frac{\Gamma_{\phi}}{J} D_1^1 \phi_{\xi} \right]_e - \left[ \rho r U \phi - \frac{\Gamma_{\phi}}{J} D_1^1 \phi_{\xi} \right]_w \\
 & + \left[ \rho r V \phi - \frac{\Gamma_{\phi}}{J} D_2^2 \phi_{\eta} \right]_n - \left[ \rho r V \phi - \frac{\Gamma_{\phi}}{J} D_2^2 \phi_{\eta} \right]_s \\
 & = ( S_{\phi}^P \phi_P + S_{\phi}^C ) \Delta V + S_{\phi}^b \quad (II-107)
 \end{aligned}$$

where  $S_{\phi}^b$  is the source term arised from the non-orthogonality of numerical grid and defined as;

$$S_{\phi}^b = \left( \frac{\Gamma_{\phi}}{J} D_2^1 \phi_{\eta} \right)_e - \left( \frac{\Gamma_{\phi}}{J} D_2^1 \phi_{\eta} \right)_w + \left( \frac{\Gamma_{\phi}}{J} D_1^2 \phi_{\xi} \right)_n - \left( \frac{\Gamma_{\phi}}{J} D_1^2 \phi_{\xi} \right)_s \quad (II-108)$$

The Eq.(II-107) is the standard finite volume formulation and discretization of this equation is well explained in Patankar [2]. Using the hybrid numerical scheme, the discretization equation can be obtained as follows.

$$A_P \phi_P = A_E \phi_E + A_W \phi_W + A_N \phi_N + A_S \phi_S + b_{\phi} \quad (II-109)$$

where

$$A_E = \text{MAX} [ 0.5 * | F_e | , D_e ] - 0.5 * F_e \quad (II-110)$$

$$A_W = \text{MAX} [ 0.5 * | F_w | , D_w ] + 0.5 * F_w \quad (II-111)$$

$$A_N = \text{MAX} [ 0.5 * | F_n | , D_n ] - 0.5 * F_n \quad (II-112)$$



$$A_S = \text{MAX} [ 0.5 * | F_S | , D_S ] + 0.5 * F_S \quad (\text{II-113})$$

$$A_P = A_E + A_W + A_N + A_S - S_\phi^P \Delta V \quad (\text{II-114})$$

$$b_\phi = S_\phi^C \Delta V + S_\phi^b \quad (\text{II-115})$$

and

$$F_e = ( \rho r U )_e \quad (\text{II-116})$$

$$F_w = ( \rho r U )_w \quad (\text{II-117})$$

$$F_n = ( \rho r V )_n \quad (\text{II-118})$$

$$F_s = ( \rho r V )_s \quad (\text{II-119})$$

with

$$D_e = \left( \frac{\Gamma_\phi}{J} D_1^1 \right)_e \quad (\text{II-120})$$

$$D_w = \left( \frac{\Gamma_\phi}{J} D_1^1 \right)_w \quad (\text{II-121})$$

$$D_n = \left( \frac{\Gamma_\phi}{J} D_2^2 \right)_n \quad (\text{II-122})$$

$$D_s = \left( \frac{\Gamma_\phi}{J} D_2^2 \right)_s \quad (\text{II-123})$$

In these equations, the notation  $\text{MAX} [ A , B ]$  means the greater of A and B and  $| A |$  means the absolute value of A.

The problems considered in the present study are mostly free of separation. Thus the numerical false

diffusion problem has a negligible effect on the accuracy of solution. In these situations, the choice of numerical scheme is not important. However, the use of different numerical method requires different usage of wall function method in the turbulent flow calculation. The finite analytic method usually uses the two-point wall function method while the wall function method of Launder and Spalding [43] is employed for most of finite volume calculations. The primary numerical scheme for the present study is the finite analytic method. However, the finite volume method is employed when the use of the two-point wall function method is restricted by a certain problem or when the comparisons of predictions by two different methods are thought to be important. It is noted that the finite analytic method usually employs non-dimensionalized form of governing equations while finite volume method employes dimensional form. All the derivations follow this common practice. The use of each numerical scheme for a certain problem is explained when a specific problem is mentioned.

## CHAPTER III

### SOLUTION PROCEDURE

#### III.1 Introduction

A difficulty encountered in solving the Navier-Stokes equations for incompressible flow concern the handling of pressure terms. The difficulty arises from the fact that pressure does not have its own equation in the sets of Navier-Stokes equations. The pressure term only appears in the momentum equations which must be considered as the governing equations for velocity components. Thus the continuity equation should be used to solve the pressure variable. However, there is no pressure term in the continuity equation. This difficulty is usually circumvented by deriving the pressure equation by manipulating the continuity equation and momentum equations. It is noted that the algebraic relations obtained in the previous chapter can be used only for the velocity components and turbulent quantities. The way of resolving pressure variable and satisfying mass conservation should be sought. In the present study, the SIMPLER ( Semi-Implicit Method for Pressure Linked Equation - Revised ) numerical algorithm of Patankar [2]

is employed for this purpose. In this algorithm, a Poisson like equation which is derived from the manipulation of continuity equation and momentum equations is solved to obtain the pressure field and in addition, the pressure correction equation is again solved to correct velocity components to satisfy continuity equation. Following sections are devoted to the derivation of pressure and pressure correction equation in the Cartesian coordinate system and in the body-fitted coordinate system.

### III.2. Pressure and Pressure Correction Equation in Cartesian Coordinate System

From the discretization equations for momentum equations such as Eq.(II-40) or Eq.(II-66), the velocity field  $(u, v)$  can be decomposed into pseudovelocity components  $(\hat{u}, \hat{v})$  and the pressure gradient terms contained in the source term as;

$$u_e = \hat{u}_e - d_e (p_E - p_p) \quad (\text{III-1})$$

$$v_n = \hat{v}_n - d_n (p_N - p_p) \quad (\text{III-2})$$

where

$$\hat{u}_e = \frac{1}{A_e} \left( \sum_{nb} A_{nb} u_{nb} + b_u \right)_e \quad (\text{III-3})$$

$$\hat{v}_n = \frac{1}{A_n} \left( \sum_{nb} A_{nb} v_{nb} + b_v \right)_n \quad (\text{III-4})$$

and

$$d_e = [ C_e (R_u)_e ] / [ A_e (\delta x)_e ] \quad (\text{III-5})$$

$$d_n = [ C_n (R_v)_n ] / [ A_n (\delta y)_n ] \quad (\text{III-6})$$

In these equations, subscripts e and n denote the value evaluated at staggered velocity nodes e and n as shown in Fig.III-1. For example  $A_e$ ,  $C_e$  are algebraic coefficients  $A_p$ ,  $C_p$  evaluated at node e and  $(\delta x)_e$ ,  $(\delta y)_n$  are the distances between pressure calculation points. The variables  $R_u$ ,  $R_v$  are the effective Reynolds number for variable u and v. All the coefficients are based on the finite analytic method in non-dimensional form given in Eq.(II-41) through Eq.(II-43).

The continuity equation within the control volume cell can be written as following;

$$(\Delta y)_e u_e - (\Delta y)_w u_w + (\Delta x)_n v_n - (\Delta x)_s v_s = 0 \quad (\text{III-7})$$

where  $\Delta x$ ,  $\Delta y$  are the sizes of control volume cell as shown in Fig.III-1.

The pressure equation can be derived by substituting momentum equation, Eq.(III-1) and Eq.(III-2) into continuity equation, Eq.(III-7).

$$A_P p_P = A_E p_E + A_W p_W + A_N p_N + A_S p_S - \hat{D} \quad (\text{III-8})$$

where

$$A_E = (\Delta y)_e d_e \quad (\text{III-9})$$

$$A_W = (\Delta y)_w d_w \quad (\text{III-10})$$

$$A_N = (\Delta x)_n d_n \quad (\text{III-11})$$

$$A_S = (\Delta x)_s d_s \quad (\text{III-12})$$

$$A_P = A_E + A_W + A_N + A_S \quad (\text{III-13})$$

and

$$\hat{D} = (\Delta y)_e \hat{u}_e - (\Delta y)_w \hat{u}_w + (\Delta x)_n \hat{v}_n - (\Delta x)_s \hat{v}_s \quad (\text{III-14})$$

The velocity components ( u , v ) obtained from the solution of the momentum equations will generally not satisfy the continuity equation unless the pressure field is correct. We denote these imperfect velocity components as starred velocity components ( u\* , v\* ). These starred velocity components must be corrected to satisfy continuity equation.

$$u_e = u_e^* - d_e ( P'_E - P'_P ) \quad (\text{III-15})$$

$$v_n = v_n^* - d_n ( P'_N - P'_P ) \quad (\text{III-16})$$

where P' is the pressure correction.

The pressure correction equation is obtained by substituting these velocity correction equations into continuity equation, Eq. (III-7).

$$A_P P_P' = A_E P_E' + A_W P_W' + A_N P_N' + A_S P_S' - D^* \quad (\text{III-17})$$

where

$$D^* = (\Delta y)_e u_e^* - (\Delta y)_w u_w^* + (\Delta x)_n v_n^* - (\Delta x)_s v_s^* \quad (\text{III-18})$$

and the coefficients  $A_E$ ,  $A_W$ ,  $A_N$ ,  $A_S$  and  $A_P$  are same as those given in pressure equation and the mass source  $D^*$  is based on starred velocity components which are obtained from the solution of momentum equations.

After the pressure correction equation is solved, the velocity components are corrected through velocity correction equations to satisfy continuity equation.

### III.3 Pressure and Pressure Correction Equation in Boundary-Fitted Coordinate System

#### III.3.1 One Velocity Staggered Grid Method

In one velocity staggered grid method, only one velocity component is stored in each staggered node  $e$  and  $n$  as shown in Fig. III-2. Thus derivation of pressure and pressure correction equation in this grid arrangement is not much depart from those in Cartesian coordinate system.

The discretized momentum equations for  $u_e$  and  $v_n$  can be written as follows;

$$u_e = (H_u)_e - (D_1^u)_e (p_E - p_p) - (D_2^u)_e (p_{en} - p_{es}) \quad (\text{III-19})$$

$$v_n = (H_v)_n - (D_1^v)_n (p_{en} - p_{wn}) - (D_2^v)_n (p_N - p_p) \quad (\text{III-20})$$

where

$$(H_u)_e = \frac{1}{A_e} \left( \sum_{nb} A_{nb} u_{nb} + b_u \right)_e \quad (\text{III-21})$$

$$(H_v)_n = \frac{1}{A_n} \left( \sum_{nb} A_{nb} v_{nb} + b_v \right)_n \quad (\text{III-22})$$

and

$$(D_1^u)_e = [ C_e ( R_u r b_1^1 )_e ] / [ A_e J_e ] \quad (\text{III-23})$$

$$(D_2^u)_e = [ C_e ( R_u r b_1^2 )_e ] / [ A_e J_e ] \quad (\text{III-24})$$

$$(D_1^v)_n = [ C_n ( R_v r b_2^1 )_n ] / [ A_n J_n ] \quad (\text{III-25})$$

$$(D_2^v)_n = [ C_n ( R_v r b_2^2 )_n ] / [ A_n J_n ] \quad (\text{III-26})$$

In these equations, the algebraic coefficients are based on the finite analytic method with non-dimensional form given in chapter II-4.

The velocity components  $(u, v)$  are then decomposed into pseudovelocity components  $(\hat{u}, \hat{v})$  and the pressure gradient terms.



$$u_e = \hat{u}_e - (D_1^u)_e (P_E - P_P) \quad (\text{III-27})$$

$$v_n = \hat{v}_n - (D_2^v)_n (P_N - P_P) \quad (\text{III-28})$$

where

$$\hat{u}_e = (H_u)_e - (D_2^u)_e (p_{en} - p_{es}) \quad (\text{III-29})$$

$$\hat{v}_n = (H_v)_n - (D_1^v)_n (p_{en} - p_{wn}) \quad (\text{III-30})$$

The cross derivative terms of pressure are contained in the pseudovelocity components to avoid a complicated nine point pressure equation and are explicitly evaluated using the pressure obtained from the previous iteration.

The continuity equation within the control volume cell in non-orthogonal coordinate system can be written as follows.

$$\begin{aligned} & [r(b_1^1 u + b_2^1 v)]_e - [r(b_1^1 u + b_2^1 v)]_w \\ & + [r(b_1^2 u + b_2^2 v)]_n - [r(b_1^2 u + b_2^2 v)]_s = 0 \end{aligned} \quad (\text{III-31})$$

The Eq.(III-31) can be rearranged as following form.

$$(rb_1^1 u)_e - (rb_1^1 u)_w + (rb_2^2 v)_n - (rb_2^2 v)_s = -D_N \quad (\text{III-32})$$

where

$$D_N = (rb_2^1 v)_e - (rb_2^1 v)_w + (rb_1^2 u)_n - (rb_1^2 u)_s \quad (\text{III-33})$$

The pressure equation can be derived by substituting Eq. (III-27) and Eq. (III-28) to continuity equation, Eq. (III-32).

$$A_P p_P = A_E p_E + A_W p_W + A_N p_N + A_S p_S - \hat{D} \quad (\text{III-34})$$

$$A_E = (r b_1^1)_e (D_1^u)_e \quad (\text{III-35})$$

$$A_W = (r b_1^1)_w (D_1^u)_w \quad (\text{III-36})$$

$$A_N = (r b_2^2)_n (D_2^v)_n \quad (\text{III-37})$$

$$A_S = (r b_2^2)_s (D_2^v)_s \quad (\text{III-38})$$

$$A_P = A_E + A_W + A_N + A_S \quad (\text{III-39})$$

where

$$\hat{D} = (r b_1^1 \hat{u})_e - (r b_1^1 \hat{u})_w + (r b_2^2 \hat{v})_n - (r b_2^2 \hat{v})_s + D_N \quad (\text{III-40})$$

and the mass source  $\hat{D}$  is based on the pseudovelocity components. The mass source term  $D_N$  expressed in Eq. (III-33) should be evaluated from the continuity-satisfying velocity components from the previous iteration level. It is noted that the velocity components that are needed for evaluation of  $D_N$  are not stored at the staggered nodes and are obtained by averaging neighbouring nodal values.

The imperfect velocity components ( $u^*$ ,  $v^*$ ) which are obtained from the solution of momentum equations are corrected by the pressure correction in order to satisfy the continuity equation.

$$u_e = u_e^* - (D_1^u)_e (P'_E - P'_P) \quad (\text{III-41})$$

$$v_n = v_n^* - (D_2^v)_n (P'_N - P'_P) \quad (\text{III-42})$$

where  $P'$  is the pressure correction. The pressure correction terms arised from the cross derivatives are neglected to ensure the diagonal dominant pressure correction equation which is an important parameter to achieve the stability of solution procedure. The neglection of these terms do not effect the accuracy of final converged solution since the pressure correction should be zero at the final converged stage. However, this practice may slow down the convergence when the numerical grid is strongly non-orthogonal.

The pressure correction equation is obtained by substituting velocity correction equations into continuity equation, Eq.(III-32).

$$A_P P'_P = A_E P'_E + A_W P'_W + A_N P'_N + A_S P'_N - D^* \quad (\text{III-43})$$

where

$$D^* = (rb_1^1 u^*)_e - (rb_1^1 u^*)_w + (rb_2^2 v^*)_n - (rb_2^2 v^*)_s + D_N \quad (\text{III-44})$$

where the coefficients  $A_E$ ,  $A_W$ ,  $A_N$ ,  $A_S$  and  $A_P$  are the same as those given in pressure equation and  $D^*$  is based on the starred velocities  $(u^*, v^*)$  which only satisfy the momentum equation.

With the updated pressure correction, the velocities are improved through the velocity correction equations to satisfy continuity equation.

It is noted that pressure and velocity coupling technique used in the present study is slightly different from that used by Shyy et al. [24] although same one velocity staggered grid method is employed.

### III.3.2 Two Velocities Staggered Grid Method

In two velocities staggered method, both velocity components are stored at each staggered node  $e$  and  $n$  as shown in Fig.III-3. The discretized form of momentum equations for these velocity components can be written as follows with the pressure gradient term expressed explicitly.

$$u_e = (H_u)_e - (D_1^u)_e (p_E - p_P) - (D_2^u)_e (p_{en} - p_{es}) \quad (\text{III-45})$$

$$v_e = (H_v)_e - (D_1^v)_e (p_E - p_p) - (D_2^v)_e (p_{en} - p_{es}) \quad (\text{III-46})$$

and

$$u_n = (H_u)_n - (D_1^u)_n (p_{en} - p_{wn}) - (D_2^u)_n (p_N - p_p) \quad (\text{III-47})$$

$$v_n = (H_v)_n - (D_1^v)_n (p_{en} - p_{wn}) - (D_2^v)_n (p_N - p_p) \quad (\text{III-48})$$

where

$$H_u = \frac{1}{A_p} \left( \sum_{nb} A_{nb} u_{nb} + b_u \right) \quad (\text{III-49})$$

$$H_v = \frac{1}{A_p} \left( \sum_{nb} A_{nb} v_{nb} + b_v \right) \quad (\text{III-50})$$

and

$$D_1^u = (r b_1^1) / A_p \quad (\text{III-51})$$

$$D_2^u = (r b_1^2) / A_p \quad (\text{III-52})$$

$$D_1^v = (r b_2^1) / A_p \quad (\text{III-53})$$

$$D_2^v = (r b_2^2) / A_p \quad (\text{III-54})$$

In these equations, the algebraic coefficients are based on finite volume method.

The velocity field  $(u, v)$  can be decomposed into pseudovelocity components  $(\hat{u}, \hat{v})$  and the pressure gradient terms as follows.

$$u_e = \hat{u}_e - (D_1^u)_e (P_E - P_p) \quad (\text{III-55})$$

$$v_e = \hat{v}_e - (D_1^v)_e (P_E - P_P) \quad (\text{III-56})$$

and

$$u_n = \hat{u}_n - (D_2^u)_n (P_N - P_P) \quad (\text{III-57})$$

$$v_n = \hat{v}_n - (D_2^v)_n (P_N - P_P) \quad (\text{III-58})$$

where

$$\hat{u}_e = (H_u)_e - (D_2^u)_e (p_{en} - p_{es}) \quad (\text{III-59})$$

$$\hat{v}_e = (H_v)_e - (D_2^v)_e (p_{en} - p_{wn}) \quad (\text{III-60})$$

and

$$\hat{u}_n = (H_u)_n - (D_1^u)_n (p_{en} - p_{wn}) \quad (\text{III-61})$$

$$\hat{v}_n = (H_v)_n - (D_1^v)_n (p_{en} - p_{wn}) \quad (\text{III-62})$$

The cross derivative terms of pressure are again contained in pseudovelocity components to obtain diagonal dominant five point pressure equations.

The continuity equation for the control volume cell shown in Fig.III-3 can be written as follows;

$$U_e - U_w + V_n - V_s = 0 \quad (\text{III-63})$$

where  $U$  and  $V$  are the contravariant velocity components defined as;

$$U = r ( b_1^1 u + b_2^1 v ) \quad (\text{III-64})$$

$$V = r ( b_1^2 u + b_2^2 v ) \quad (\text{III-65})$$

The pressure equation is then obtained by substituting Eq.(III-55) through Eq.(III-58) into continuity equation, Eq.(III-63).

$$A_P P_P = A_E P_E + A_W P_W + A_N P_N + A_S P_S - \hat{D} \quad (\text{III-66})$$

where

$$A_E = r_e ( b_1^1 D_1^u + b_2^1 D_1^v )_e \quad (\text{III-67})$$

$$A_W = r_w ( b_1^1 D_1^u + b_2^1 D_1^v )_w \quad (\text{III-68})$$

$$A_N = r_n ( b_1^2 D_2^u + b_2^2 D_2^v )_n \quad (\text{III-69})$$

$$A_S = r_s ( b_1^2 D_2^u + b_2^2 D_2^v )_s \quad (\text{III-70})$$

$$A_P = A_E + A_W + A_N + A_S \quad (\text{III-71})$$

$$\hat{D} = \hat{U}_e - \hat{U}_w + \hat{V}_n - \hat{V}_s \quad (\text{III-72})$$

and  $\hat{U}$  and  $\hat{V}$  are based on the pseudovelocity components and defined as;

$$\hat{U} = r ( b_1^1 \hat{u} + b_2^1 \hat{v} ) \quad (\text{III-73})$$

$$\hat{V} = r ( b_1^2 \hat{u} + b_2^2 \hat{v} ) \quad (\text{III-74})$$

Since the velocity components obtained from the solution of momentum equations with imperfect pressure field generally do not satisfy continuity equation. These starred velocity components ( $u^*$ ,  $v^*$ ) are corrected through the pressure correction in order to satisfy mass conservation during the iteration process.

$$u_e = u_e^* - (D_1^u)_e (P'_E - P'_P) - (D_2^u)_e (P'_{en} - P'_{es}) \quad (\text{III-75})$$

$$v_e = v_e^* - (D_1^v)_e (P'_E - P'_P) - (D_2^v)_e (P'_{en} - P'_{es}) \quad (\text{III-76})$$

and

$$u_n = u_n^* - (D_1^u)_n (P'_{en} - P'_{wn}) - (D_2^u)_n (P'_N - P'_P) \quad (\text{III-77})$$

$$v_n = v_n^* - (D_1^v)_n (P'_{en} - P'_{wn}) - (D_2^v)_n (P'_N - P'_P) \quad (\text{III-78})$$

where  $P'$  is the pressure correction

In order to facilitate the velocity correction procedure, following contravariant velocity correction equations are derived using Eq.(III-75)-Eq.(III-78) and Eq.(III-64)-Eq.(III-65).

$$U_e = U_e^* - (D_U)_e (P'_E - P'_P) \quad (\text{III-79})$$

$$V_n = V_n^* - (D_V)_n (P'_N - P'_P) \quad (\text{III-80})$$

where



$$U_e^* = r_e ( b_1^1 u^* + b_2^1 v^* )_e \quad (\text{III-81})$$

$$V_n^* = r_n ( b_1^2 u^* + b_2^2 v^* )_n \quad (\text{III-82})$$

and

$$(D_U)_e = r_e ( b_1^1 D_1^u + b_2^1 D_1^v )_e \quad (\text{III-83})$$

$$(D_V)_n = r_n ( b_1^2 D_2^u + b_2^2 D_2^v )_n \quad (\text{III-84})$$

The pressure correction terms arised from the non-orthogonality of numerical grids are neglected to avoid the complicated nine point pressure correction equation and to ensure diagonal dominant pressure correction equation.

The pressure correction equation is obtained by substituting these contravariant velocity correction equations into continuity equation, Eq.(III-63).

$$A_P P_P' = A_E P_E' + A_W P_W' + A_N P_N' + A_S P_N' - D^* \quad (\text{III-85})$$

where

$$D^* = U_e^* - U_w^* + V_n^* - V_s^* \quad (\text{III-86})$$

where the coefficients  $A_E$ ,  $A_W$ ,  $A_N$ ,  $A_S$  and  $A_P$  are the same as those in pressure equation and  $D^*$  is based on the starred velocities which are obtained from the solutions of the momentum equations.

From the updated pressure correction, the contravariant velocity components are corrected to satisfy mass conservation. The other contravariant velocity components,  $U_n$  and  $V_e$  which lie parallel to the control volume surfaces are obtained from linear interpolation of newly computed neighbouring nodal values.

Then the corrected velocities are obtained from these continuity-satisfying, corrected contravariant velocity components.

$$u = ( b_2^2 U - b_2^1 V ) / J \quad (\text{III-87})$$

$$v = - ( b_1^2 U - b_1^1 V ) / J \quad (\text{III-88})$$

where

$$J = r ( b_1^1 b_2^2 - b_2^1 b_1^2 ) \quad (\text{III-89})$$

It is noted that the continuity-satisfying, corrected contravariant velocity components are stored and used for the evaluation of algebraic coefficients  $A_E$ ,  $A_W$ ,  $A_N$ ,  $A_S$  and  $A_P$  for transport quantities.

Maliska and Raithby [30] used the same procedure as outlined above in the calculation of three dimensional parabolic flow of arbitrary cross sectioned ducts. However, they used exact nine-point pressure and pressure

correction equations which may deteriorate the stability of solution process when the numerical grids are strongly non-orthogonal.

#### III.4 Overall Solution Procedure

The solution procedure of SIMPLER algorithm is adopted for all the calculations in the present study. The momentum equations and turbulent transport equations are first solved by parabolic marching technique in the flow direction using the pressure obtained from the previous iteration. After the marching of calculations for turbulent quantities is completed, the pressure field is solved elliptically with several global iterations. The advantage of this global, elliptic pressure calculation technique is a monotonic and rapid convergence of pressure due to a proper consideration of the elliptic nature of the pressure field, especially near the leading and trailing edges.

The detailed overall solution procedure can be outlined as following.

- 1) Generate the numerical grids using the methods outlined in chapter II-3 and save the geometric data, such as  $b_i^j$ ,  $J$ ,  $f^i$ ,  $g^{ij}$  etc.

2) Specify the inlet boundary conditions for transport quantities, the velocity components and turbulent quantities.

3) Set pressure and pressure correction equal to zero everywhere in the solution domain.

4) Calculate the algebraic coefficients and source terms for momentum equations, such as those given in Eq.(II-67) through Eq.(II-81), at downstream station using the transport quantities obtained from the previous iteration level. This step includes the calculation of coefficients of pressure and pressure correction equation such as those given in Eq.(III-35) through Eq.(III-39).

5) Solve the momentum equations, such as Eq.(III-19) and Eq.(III-20), using the pressure from the previous iteration level to obtain starred velocity field,  $(u^*, v^*)$ .

6) Using this starred velocity components, calculate the mass source for the pressure correction equation,  $D^*$ .

7) Solve the pressure correction equation, such as Eq.(III-43) and update the velocity field using Eq.(III-41) and Eq.(III-42) to satisfy continuity equation.

8) Calculate the pseudovelocity field,  $(\hat{u}, \hat{v})$  from Eq.(III-29) and Eq.(III-30) using the corrected velocity components.

9) Calculate the mass source for pressure equation,  $\hat{D}$ , and store for later use.

10) Calculate the algebraic coefficients and source terms for the turbulent transport equations using the updated velocity field and turbulent quantities from the previous iteration level.

11) Solve the turbulent transport equations.

12) March to the next downstream station and repeat the procedure 4) to 11).

13) After the marching is completed at the last downstream station, the pressure equation such as Eq.(III-34) is solved elliptically with several global iterations ( typically 20 sweeps ) from downstream to upstream to update new pressure field.

14) Return to step 4) for the next iteration level.

15) Steps 4) through 14) are repeated until convergence or steady state solution is reached.

All the algebraic equations are solved by line by line technique using tridiagonal matrix algorithm. It is noted that although parabolic marching technique is employed for the calculation of transport quantities, all the transport quantities are stored in two dimensional array. Thus the present solution procedure has potential for the calculation of separated flow.

CHAPTER IV  
CALCULATION OF TURBULENT WAKE PAST A FLAT PLATE BY WAKE  
FUNCTION METHOD

IV.1 Introduction

The turbulent flow past a finite flat plate has been studied by many researchers because it provides a fundamental understanding of wake development and the basic feature of viscous-inviscid interaction at the trailing edge of the body. According to Alber [62], the wake region can be divided into three fundamental regions; laminar wake region where the laminar sublayer on the plate is destroyed, turbulent inner near wake region where the logarithmic remnant of the trailing edge boundary layer is destroyed, the far wake region where the flow field loses the memory of the turbulent boundary layer on the flat plate and attains the self preserving form. The experimental data of Chevray and Kovasznay [63], Ramaprian et al. [64], Pot [65], Andreopoulous and Bradshaw [66] confirm such a wake behaviour behind a flat plate. Some comparisons and reviews of these data can be seen in Ramaprian et al. [64].

The earlier calculations by Rodi [67], Launder et al. [68] had placed more emphasis on the test of turbulence model. More extensive calculations were carried out by Patel and Scheuerer [69] for symmetric and asymmetric turbulent wake of a flat plate. The results were compared with several available experimental data. In these works, the authors used the boundary layer calculation method with zero pressure gradient and calculations were started at the trailing edge of the flat plate with initial conditions provided by experimental data. Thus, the boundary layer on the flat plate and viscous-inviscid interaction at the trailing edge of flat plate were not considered. A detailed large domain solution surrounding the trailing edge of the flat plate was obtained recently by Patel and Chen [70]. The two-layer model was adopted for the boundary layer calculation on the flat plate and the  $k - \epsilon$  turbulence model was adopted for the wake calculation. The boundary layer on the flat plate was accurately calculated to the laminar sublayer ( $y^+ \sim 0.4$ ) by the two-layer model in which the turbulent kinetic energy and turbulent kinetic energy dissipation rate in the laminar sublayer and the buffer layer and a part of logarithmic layer are specified by universal functions based on the experimental data curve-fitting and eddy viscosity in these regions is calculated by the Van Driest



formula. Comparisons of calculated results with experimental data and results from the previous calculations were made. However, a detailed destruction of laminar sublayer and buffer layer in the wake was not calculated since the  $k - \epsilon$  turbulence model was adopted in the wake calculation. For example, the destruction of turbulent kinetic energy peak at  $y^+ \sim 15$  on the trailing edge of turbulent boundary layer and the development of wake centerline velocity in the laminar wake region were not calculated. While the two layer model predicted a reasonably accurate result, it requires many computational nodes in the near wall region over the plate. It is quite questionable that the present low Reynolds number turbulence model can be used confidently for the prediction of rapid mixing in the laminar wake region although it has been successful in the various calculations of boundary layer type flow.

The purpose of the present study is to develop a calculation method in which the concept of a wall function method is adopted and extended to the wake calculation. In this approach, the laminar wake region is excluded from the calculation domain and wake functions are introduced to accurately account for the large streamwise velocity gradient along the wake centerline in the same spirit that wall functions are used in the near wall calculation. It

is shown that the application of the combination of wall and wake function may satisfactorily predict the turbulent boundary layer on the plate, the trailing edge interaction and the development of turbulent wake.

#### IV.2 Numerical Grids and Solution Domains

The solution domain shown in Fig.IV-1 consists of wall function region, wake function region, and calculation region. The wall function region is one computational control volume over the plate which extends from  $y^+ = 0$  to  $y^+ \sim 150$ . The wake function region is one computational control volume just downstream of the trailing edge of flat plate extending approximately from  $x^+ = 0$  to  $x^+ \sim 500$  in the wake region.

Non-uniform 57x31 grids are generated numerically by the grid generation technique outlined in chapter II-4-1 with exponential distribution along the y-direction and sinusoidal distribution along the x-direction so that an appropriate grid concentration can be made close to the wall and near the trailing edge of flat plate as shown in Fig.II-3. The details of informations on grid and sizes of solution domains are given in Table 2. The numerically generated coordinate lines are treated as control volume lines and grid lines are placed at the center of control

volume lines as suggested by Patankar [2]. Of the 31 grid points in the y-direction, 10 grid points are placed within the boundary layer on the body. The origin of coordinate system is placed at the trailing edge of plate.

### IV.3 Boundary Conditions

The upstream inlet condition for u-velocity is specified by the following simple flat-plate correlations [71].

$$u = \left( \frac{y}{\delta} \right)^{1/7} \quad (\text{IV-1})$$

$$\delta(x) = 0.37 x (Re x)^{-1/5} \quad (\text{IV-2})$$

where  $\delta(x)$  denotes the boundary layer thickness at distance  $x$  from the leading edge and  $Re$  is the Reynolds number based on the free stream velocity and plate length.

The upstream inlet condition for turbulent kinetic energy and the rate of turbulent kinetic energy dissipation are specified by following simple relations.

$$k = c_{\mu}^{-1/2} u_{\tau}^2 \left( 1 - \frac{y}{\delta} \right) \quad (\text{IV-3})$$

$$\varepsilon = c_{\mu}^{3/4} k^{3/2} / l_m \quad (\text{IV-4})$$

where  $u_\tau = (\tau_w/\rho)^{1/2}$  is the friction velocity,  $c_\mu = 0.09$  and the length scale  $l_m$  is calculated by Escudier formula [72].

$$y/\delta < c_\mu/\kappa, \quad l_m = \kappa y \quad (\text{IV-5})$$

$$y/\delta > c_\mu/\kappa, \quad l_m = c_\mu \delta \quad (\text{IV-6})$$

In the wall function region,  $u$ ,  $k$ , and  $\epsilon$  are calculated by the following conventional relations [43].

$$\frac{u}{u_\tau} = \frac{1}{\kappa} \ln y^+ + B \quad (\text{IV-7})$$

$$k = c_\mu^{-1/2} u_\tau^2 \quad (\text{IV-8})$$

$$\epsilon = u_\tau^3 / \kappa y \quad (\text{IV-9})$$

where  $y^+ = Re u_\tau y$ ,  $\kappa$  is von-Karman constant,  $\kappa = 0.42$  and  $B = 5.5$ .

The two point wall function method used in the present study is a slightly different from the wall function method generally used in the control volume method. It is assumed that at least two  $u$  calculation nodes are placed in the logarithmic region. The friction velocity  $u_\tau$  is first determined by iterative method using the  $u$  value in the third node ( $y^+ \sim 150$ ) and Eq.(IV-7). Then the  $u$ ,  $k$ ,  $\epsilon$  values in the second node ( $y^+ \sim 60$ ) are

evaluated using this friction velocity and Eqs. (IV-7)-(IV-9).

In the wake function region, the wake centerline velocity  $u_c$ , and other variables,  $v_w$ ,  $k_w$ , and  $\epsilon_w$  are specified at the locations as shown in the Fig. IV-2. Detailed specification and explanation of these wake functions will be given later.

The other boundary conditions are same as those reported in chapter II-3-2.

#### IV.4. Wake Function Method

##### IV.4.1. Significance of Wake Function Method.

One of the difficulties that arises in the prediction of wake flow is that when the turbulent boundary layer leaves the trailing edge of flat plate, the turbulent structure of flow changes rapidly because the no slip condition on the plate abruptly changes to the symmetry condition in the wake. There is a rapid mixing near the wake centerline and the turbulent boundary layer on the plate breaks down and develops into the wake. The flow development in the wake can be divided into three fundamental regions as mentioned before. The practical difficulty encountered in the calculation of turbulent

wake is that none of the present turbulence models can be used confidently for the calculation of the laminar wake and far wake. On the other hand, it is not easy to capture numerically the rapid variation of turbulent structure near the wake centerline unless large number of nodes are placed near the trailing edge. The same difficulties are also encountered in the calculation of flow near the wall. The wall function is introduced to circumvent these two difficulties. In the present study, we introduce wake functions which may account for the development of velocity along the wake centerline while laminar wake region is excluded from the calculation domain. The detailed calculation in the laminar wake region and the uncertainty of the turbulence model in the laminar wake calculation can be removed by this approach. If there is no separation near the trailing edge of the body like present problem, the downstream near wake calculation is strongly depends on the flow condition immediately downstream of the trailing edge. Thus a proper imposition of the wake function may reduce the error initiated by the wake inlet condition.

One of the difficulties in the calculation of a symmetric wake when the wall function method is used in the calculation of boundary layer of body is the imposition of symmetry condition for the  $u$ -velocity along

the wake centerline in the turbulent inner near wake region. In the initial phase of this study, an interpolation function which satisfies the symmetry condition along the wake centerline was used for the evaluation of quantities of the first control volume in the wake and along wake centerline while calculations were carried out from the second control volume. The square interpolation function,  $Ay^2 + B$ , was used for the evaluation of  $u$ ,  $k$ ,  $\epsilon$  and linear interpolation function,  $Ay$ , was used for the evaluation of  $v$ . However, these interpolation functions do not accurately account for the physical behaviour of the flow. The errors generated by these interpolation functions thus affect the solution in the normal direction by diffusion and in the downstream direction by convection. As a result, the downstream calculation, particularly in the turbulent near wake region, suffers from these errors.

According to the theory of Alber [62], the mean  $x$ -directional velocity component has the logarithmic behaviour of the turbulent boundary layer in the turbulent inner near wake region although it breaks down as the flow moves downstream. In the present study, the first  $x$ -directional computational grid in the wake region is placed at the starting point of turbulent inner wake region ( $x^+ \sim 250$ ) and the first three  $y$ -directional  $u$ -

velocity calculation nodes are placed at the logarithmic layer ( $y^+ \sim 60, 150, 250$ ). Thus, the computational nodes used in the interpolation function for the initial portion of turbulent inner near wake region are placed in the logarithmic layer. The discrepancies between the physical behaviour and interpolation functions result in a considerable error, especially in the evaluation of u-velocity. If the square interpolation function is used in the evaluation of u-velocity, it gives a higher u-velocity values for the second node and the wake centerline. Consequently, this higher u-velocity influences the generation term of kinetic energy in the third node. Thus, the turbulent kinetic energy is underpredicted. These errors can be reduced if many finer grids are placed near the wake centerline. However, there is a limit because the wall function used in the boundary layer calculation can only be applied to the region  $60 < y^+ < 400$ . On the other hand, only the laminar sublayer,  $y^+ < 10$ , from the boundary layer flow is destroyed in the starting portion of turbulent inner wake region. In other words, the wake symmetry condition satisfies only about a laminar sublayer thickness above the wake centerline. Thus, an accurate resolution of the symmetry condition along the wake centerline by a simple interpolation function is impossible, and more importantly, the solution



will be quite grid dependent. It is quite apparent that a direct imposition of the symmetry condition  $\phi(i,2) = \phi(i,1)$  over a distance of  $\Delta y^+$  in the order of 60 should cause severe error.

One possible way of avoiding these errors will be a direct calculation along the wake centerline with the inlet condition provided properly. If we consider the calculation of asymmetric wake, the whole domain should be calculated. Rhie and Chow [16] performed the whole domain calculation even in the symmetric wake calculation to avoid numerical error from interpolation along the line of symmetry. However, details of the inlet conditions of transport quantities for the wake calculation immediately downstream of the plate were not explained in their paper.

Since the  $k - \epsilon$  turbulence model is used in this study, the control volume which contains the laminar wake region should be excluded from the wake calculation. Since the laminar sublayer and buffer layer and a part of the logarithmic layer in the boundary layer on the body are not calculated in the wall function method, the detailed destruction of these layers in the wake can not be calculated. Thus a universal function which can resolve the large streamwise velocity gradient caused by the destruction of these layers should be used in the wake

function region. The idea of creating the wake function is the same as that the wall function is introduced in the calculation of boundary layer on the body.

#### IV.4.2. Proposed Wake Functions.

In order to adopt a wake function, we note that Alber [62] obtained the mean velocity profile in the turbulent inner near wake region using the boundary layer approximation and similarity transformation under the assumption of a linear distribution of eddy viscosity in the normal direction. The wake centerline velocity and velocity components near the wake centerline in the turbulent inner near wake region are

$$\frac{u_c}{u_{\tau 0}} = \frac{1}{\kappa} [ \ln g(x^+) - \gamma ] + B \quad (\text{IV-10})$$

$$\frac{u}{u_{\tau 0}} = \frac{1}{\kappa} [ \ln y^+ + E_1(\zeta) ] + B \quad (\text{IV-11})$$

$$\frac{v}{u_{\tau 0}} = \kappa [ 1 - \exp(-\zeta) ] / \ln g(x^+) \quad (\text{IV-12})$$

where

$$x^+ = \text{Re } u_{\tau 0} x, \quad y^+ = \text{Re } u_{\tau 0} y, \quad \zeta = y^+/g(x^+)$$

$$E_1(\zeta) = \int_{\zeta}^{\infty} \exp(-t)/t dt$$

and

$$g(\alpha) [ \ln g(\alpha) - 1 ] = \kappa^2 \alpha \quad (\text{IV-13})$$

and  $u_{\tau 0}$  is the trailing edge friction velocity and  $\gamma$  ( $= 0.5772157$ ) is the Euler constant. The constants  $\kappa$ ,  $B$  in these equations were taken as  $\kappa = 0.42$ ,  $B = 5.5$  for the consistency with the wall function. Eq.(IV-12) was obtained from Eq.(IV-11) and continuity equation.

One may alternatively use the centerline velocity formula given in Eq.(IV-10) with the correlation obtained from experiment by Andreopoulos and Bradshaw [66];

$$\frac{u_c}{u_{\tau 0}} = 2.02 \ln x^+ + 0.7 \quad (\text{IV-14})$$

Alber's solution, Eq.(IV-10), is used as the inlet condition for the downstream wake centerline u-velocity calculation in this study.

For the proper introduction of wake functions for the turbulent quantities in the wake function region, we may consider Chevray and Kovasnay's data [63]. They show that the turbulent structure in the logarithmic layer from the trailing edge of the flat plate to the starting point of turbulent inner near wake region remains the same although the structure of laminar sublayer and buffer layer evolve rapidly by means of mixing in the laminar wake region. Therefore, we may use the second y-directional nodal values of the  $k$  and  $\epsilon$  at the trailing edge of plate as the second y-directional nodal values of  $k$  and  $\epsilon$  in the first

control volume in the wake, as shown in Fig.IV-2, without loss of accuracy. Obviously, a direct calculation of these quantities is impossible since the calculation point for  $k_w$  and  $\epsilon_w$  is located immediately downstream of the trailing edge where the centerline u-velocity can not be calculated. We thus adopt

$$k_w = c_{\mu}^{-1/2} u_{\tau 0}^2 \quad (\text{IV-15})$$

$$\epsilon_w = u_{\tau 0}^3 / \kappa y \quad (\text{IV-16})$$

as the second nodal values of the turbulent kinetic energy and the rate of its dissipation in the first control volume in the wake. Here,  $u_{\tau 0}$  is the trailing edge friction velocity.

There is no detailed experimental observation made for the variation of  $k$  and  $\epsilon$  along the wake centerline within our knowledge. This lack of inlet conditions for  $k$  and  $\epsilon$  along the wake centerline lead us to use the interpolation function (  $\phi = Ay^2 + B$  ) in the subsequent calculation of  $k$ ,  $\epsilon$  along the wake centerline. However, this does not create error in the calculation of  $k$ ,  $\epsilon$  in the second node in the wake because the value of  $k$  and  $\epsilon$  in the second node in the wake are more dependent on the generation term of the turbulent kinetic energy than the  $k$ ,  $\epsilon$  values along the wake centerline. In other words,

accurate resolution of the wake centerline velocity is more important for the accurate prediction of second nodal turbulent kinetic energy than the specification of  $k$  and  $\epsilon$  along the wake centerline.

#### IV.4.3 Enforcing the Conservation of Mass

An important problem in the calculation of wake by the wall and wake function method is the calculation of the second nodal  $v$ -velocity component in the wake region. The difficulty arises from the fact that the detailed destruction ( mixing ) of laminar sublayer and buffer layer and logarithmic layer in the first  $y$ -directional control volume in the wake can not be properly described in the wall function method due to the lack of computational nodes near the wake centerline. Obviously, this is more severe in the calculation of the initial portion of turbulent inner near wake. To avoid these numerical difficulties, we adopt PSL ( Parabolic Sublayer ) like treatment [73]. In this study, the second nodal  $v$ -velocity in the wake which is denoted as  $v_2$  in Fig.IV-2 is calculated by the mass conservation of the first control volume using the following simple integration formula.

$$\Delta x_c v_2 = \int_0^{\Delta y_c} u_1 dy - \int_0^{\Delta y_c} u_2 dy \quad (\text{IV-17})$$

where

$$\int_0^{\Delta y_c} u \, dy = \left( \frac{1}{3} u(i,1) + \frac{13}{24} u(i,2) + \frac{1}{8} u(i,3) \right) \Delta y_c$$

(IV-18)

The accuracy of the integration formula is very significant in the wake calculation since a small error in mass conservation may lead to a large error in the prediction of the pressure field in the weak interaction like present problem. It is not easy to find a good integration formula which can accurately resolve the mass flux of laminar sublayer, buffer layer and a part of logarithmic layer at the trailing edge of body ( $x = 0$ ) so that  $v_w$  in Fig.IV-2 can be calculated by Eq.(IV-17). Thus the Eq.(IV-12) is used only for the specification of the v-velocity component in the wake function region ( $v_w$  in Fig.IV-2) for simplicity and accuracy. The pressure in the wake is calculated from the third node in the y-direction using these v-velocity components as velocity boundary condition. The u-velocity component in the second node is calculated using the pressure in the third node. The starting point of calculation in the y-direction in the wake region is shown in Fig.IV-2.

#### IV.5. Results and Discussions

Calculations were performed using the partially parabolic form of finite analytic method which is derived in chapter II-5-1 and SIMPLER numerical algorithm outlined in chapter III-4. The derivation of pressure and pressure correction equation is given in chapter III-2. The time marching technique is employed for the iteration process and satisfactory convergence were obtained after 100 time marching with time step  $\Delta t = 1$ .

The calculation is made for Reynolds number  $2.48 \times 10^6$  based on the plate length and free stream velocity. The predicted results are presented and compared with available experimental data of Ramaprian et al. [64] who measured the mean velocity and turbulent quantities in the turbulent inner near wake region.

The converged pressure distribution near the trailing edge of the flat plate along  $y^+ \sim 150$  over the plate and wake centerline is shown in the Fig.IV-3. Fig.IV-3 shows the pressure near the trailing edge, either upstream or downstream, is lower than the free stream pressure which is set equal to zero. The minimum pressure is located at the trailing edge where the fluid experiences the maximum acceleration and deceleration. According to Alber [62],

the pressure gradient effect is mainly confined to a small laminar wake region close to the trailing edge where  $x^+ < 100$ . Obviously, the present wall and wake function method together with the  $k - \epsilon$  turbulence model can not predict the detailed interaction in this laminar wake region. However, the sharp reduction followed by a rapid increase of pressure near the trailing edge and its slow recovery in the wake is predicted by the present method as shown in Fig.IV-3. The prediction of pressure distribution by Patel and Chen [70] by two-layer model is given in Fig.IV-4. There exists a small difference between the present prediction and the calculation by Patel and Chen [70] in the region of plate. However, both predictions give nearly the same magnitude of trailing edge interaction.

The calculated result of the skin friction coefficient is shown in Fig.IV-5 and compared with the conventional flat-plate correlation. In general, Fig.IV-5 shows a good agreement between the predicted value and simple flat plate correlation. Since the pressure on the plate drops rapidly near the trailing edge causing the boundary layer to accelerate, the skin friction coefficient near the trailing edge is expected to increase. This increase, even though it is small, is well predicted in the present calculation.



Fig.IV-6 shows the comparison of the predicted wake centerline velocity distribution with experimental data by Ramaprian et al.[64]. The first computational node along the centerline is located approximately at  $x^+ \sim 250$ . An excellent agreement is obtained by the application of wall and wake function method. As compared with the predicted wake centerline velocity distribution by Patel and Chen [70], which is shown in Fig.IV-7, the present wake function method accurately predicts the wake development without the detailed calculation of the laminar sublayer on the plate and the laminar wake.

Figs.IV-8, IV-9, and IV-10 show the detail of the predicted u-velocity component, Reynolds shear stress and turbulent kinetic energy. The comparison of the predicted results with experimental data is excellent. As mentioned before, the direct imposition of symmetry condition or the use of interpolation function as the symmetry condition for u-velocity along the centerline may lead to predictions of a higher u-velocity and lower turbulent kinetic energy due to the inadequate grid concentration near the wake centerline when the wall function method is adopted in the calculation of boundary layer on the body. On the other hand, the present method of combining the wall and wake function clearly removes these difficulties.

It should be mentioned that while the selection of appropriate wake functions is important, an accurate evaluation of the second nodal v-velocity component is also important in achieving the accurate prediction of the near wake turbulent structure. We note that the turbulent kinetic energy and turbulent shear stress measurements of Ramaprian et al. [64] give slightly higher values than those of Pot [65] in the initial portion of turbulent inner near wake region which can be seen at the locations  $x = 25.4$  mm and  $x = 127$  mm.

A proper prediction of the v-velocity component in the turbulent near wake region is very significant for the accurate prediction of the flow structure since the amount of mixing in this region is more or less related to the v-velocity component. The second nodal v-velocity in the wake region calculated from the present PSL (parabolic sublayer) like treatment can not be confirmed due to the lack of experimental data. A comparison with Alber theory [62] is made. Two calculations are performed with a different grid distribution near the wake centerline,  $\Delta y_c^+ \sim 120$  ( grid 1 ),  $\Delta y_c^+ \sim 200$  ( grid 2 ) where  $\Delta y_c$  is the size of the first y-directional control volume in the wake as shown in Fig.IV-2. Fig.IV-11 shows the comparison of the present prediction with Alber theory [62] in the turbulent inner near wake region. Good agreement is

obtained but the integral formula gives a little grid dependency.

Fig.IV-12 and Fig.IV-13 indicate the prediction of velocity defect and Reynolds shear stress in the far wake region where the velocity defect  $w$  ( $= 1 - u$ ) becomes small compared with  $u$ -velocity component. The assumption of similarity of velocity and shear stress profiles leads the half power law for the decay of the centerline velocity defect  $w_0$  ( $= 1 - u_c$ ) and the growth of the half width  $b$  ( $y_h = b/2$  where  $w/w_0 = 1/2$ ). According to asymptotic theory [71],

$$b \sim x^{1/2}, \quad w_0 \sim x^{-1/2} \quad (\text{IV-19})$$

A simple velocity defect and shear stress distribution can be obtained [69] when one introduces a constant eddy viscosity across the wake in the normal direction for the momentum equation.

$$w/w_0 = \exp(-4 \ln 2 \eta^2) \quad (\text{IV-20})$$

$$\frac{-\overline{uv}}{w_0^2} = 4 (\pi \ln 2)^{1/2} (v_t/u\theta) \eta \exp(-4 \ln 2 \eta^2) \quad (\text{IV-21})$$

where  $\eta = y/b$ , and  $v_t/u\theta = 0.032$  which has been confirmed by Rodi [74] from a survey of several sets of experimental data. Fig.IV-12 and Fig.IV-13 show the

comparison between present prediction and asymptotic theory, Eq.(IV-20) and Eq.(IV-21). The velocity defect, in general, is predicted well except near the edge of the wake where the computed profile underpredicted the asymptotic value. This may be due to the fact that the turbulence model can not properly simulate the high intermittance of turbulent and laminar flow occurring at the edge of the wake. On the other hand, the turbulent shear stress is underpredicted. This is a well known defect of the current  $k - \epsilon$  turbulence model which was already observed in the previous calculations [67], [69], and [70]. An improvement in the turbulence modelling is required in order to achieve a better agreement in the prediction of turbulent structure in the far wake region with the experiment. For example, Chen and Singh [75] showed a better prediction can be achieved with the  $k - \epsilon$  model based on the two turbulence scale concept where the first scale is based on the turbulent kinetic energy,  $k$ , and its dissipation rate,  $\epsilon$ , to characterize the large, energy-containing eddies while the second scale, the Kolmogorov scale, is based on the dissipation rate,  $\epsilon$ , and the kinematic viscosity,  $\nu$ , to characterize the small, energy-dissipating eddies.

A more sophisticated analysis of wake functions and related calculation method is needed for the application

of the present method to the calculation of wakes behind bodies of streamline curvature for which the pressure gradient effect are important.

#### IV.6. Conclusion

A new calculation method is presented in the present study for an accurate prediction of turbulent wake flow behind a flat plate. The method proposes a wake function that models the laminar and turbulent wake immediately downstream of the trailing edge of the plate from  $x^+ = 0$  to approximately  $x^+ = 250$ . The combination of the wake function for the near wake region and the wall function for the near wall region on the plate provides a means for accurate prediction of turbulent wake flow and eliminates excess computational grids required near the wall and the trailing edge and the uncertainty of the turbulence model.

## CHAPTER V

## LAMINAR AND TURBULENT FLOW PAST A FINITE FLAT PLATE

V.1 Introduction

The classical problem of incompressible laminar and turbulent flow past a finite flat plate has been studied by numerous investigators. It is a problem that not only provides much fundamental understanding of basic feature of leading and trailing edge interaction, boundary layer on the plate and wake development, but also of considerable importance in diverse engineering applications such as thin airfoil theory.

When the Reynolds number based on the plate length and the free stream velocity is large, the laminar boundary layer on the plate is described by the well-known Blasius solution. However, the Blasius solution, which is based on the boundary layer theory of Prandtl, does not properly describe the flow field near the leading and trailing edges of plate due to the breakdown of assumptions made in the boundary layer theory.

Due to its geometric simplicity, the laminar flow near the trailing edge of plate has been subject of

various numerical and theoretical investigations. The study of this problem provides us the basic understanding of inviscid-viscous interaction at the trailing edge and near wake development. According to the triple deck theory of Stewartson [76] or Messiter [77], for infinite Reynolds number flow, the laminar boundary layer on the plate develops into Goldstein's near-wake through a small transition region around the trailing edge which is known as the triple-deck region. This triple deck region which is embedded inside the boundary layer near the trailing edge arises in order to avoid the singularity of the boundary layer equations at the trailing edge. Melnik and Chow [78] and Veldman and van de Vooren [79] have obtained the numerical solutions of the triple deck equations.

The breakdown of boundary layer theory for describing the flow field near the trailing edge of plate leads many investigators to use alternative approaches. Two approaches are commonly used: the inviscid-viscous interaction method [80]; and the numerical solutions of fully elliptic or partially parabolic form of equations [81,82]. Recently, Chen and Patel [81] have obtained the numerical solutions of this problem using the elliptic form of finite analytic numerical scheme. The grid dependence and convergence tests, the influence of the size of solution domain on the solution were investigated

in detail. The predicted results were compared with previous results by interacting boundary layer theory and by partially parabolic numerical solution. One valuable observation made in this study is that the solution domain should be larger than that used in the previous studies in order to capture the whole inviscid-viscous interaction and far wake development and to obtain domain independent solution. The same observation is also reported in Rhie [83] in the calculation of laminar flow past a thin airfoil.

However, most of the previous studies are confined to the analysis of the flow field near the trailing edge. Thus the leading edge interaction and the initial development of boundary layer near the leading edge were not considered. In order to remove the dependency of the solution on the location of inlet solution boundary and to obtain whole flow field past a plate, the calculations should be started at the upstream of the plate with inlet conditions provided by uniform flow conditions. The earlier numerical solutions of the full Navier-Stokes equations obtained by Dennis and Dunwoody [84] are limited to relatively low Reynolds number flow. Xu and Chen [82] has recently obtained a numerical solution of complete laminar flow field past a finite flat plate using a novel nine-point finite analytic numerical scheme. The



predicted results of pressure distribution on the plate and along the wake centerline, the wake centerline velocity development and the friction coefficient distribution near the trailing edge were presented with comparison with previous studies.

In the present study, the classical problem of laminar and turbulent flow past a finite plate is reconsidered. The laminar flow past a finite flat plate is first solved using both partially parabolic form of finite analytic numerical method given in chapter II-5-1 and elliptic form of finite volume method with modified TEACH code. This practice will not only give a critical evaluation of suitability of the partially parabolic form of Navier-Stokes equations in the prediction of the strong leading edge interaction but also provide us the comparisons of results by both methods which has not been made before. Then the numerical solutions of turbulent flow past a finite flat plate is obtained by solving the Reynolds averaged Navier-Stokes equations with the  $k - \epsilon$  turbulence model and the wall function method. The finite volume method is used in this calculation. This practice will show how the introduction of turbulence model influences the overall solution. The predicted results are compared with available experimental data of Ramaprian et al. [64].

## V.2. Numerical Grids and Solution Domains

The numerical grids are generated using the method outlined in chapter II-4-1 and the partial view of generated grids are shown in Fig.II-3 for both laminar and turbulent calculations. Table-3 shows the details of solution domain and number of computational grids used in the present calculations. The solution domain in normal direction for laminar flow calculation is chosen to be  $y_u \sim 12L$  in which Chen and Patel [81] obtained the domain independent solution. The solution domain in the other directions is also made large enough to capture the whole leading and trailing edge interactions. The generated numerical grids are treated as control volume lines and the computational grid lines are placed at the center of control volume lines.

## V.3 Boundary Conditions

The upstream boundary conditions are specified by following uniform flow conditions;

$$u = 1, v_x = 0, k = k_{in}, \epsilon = \epsilon_{in} \quad (V-1)$$

where  $k_{in}$  and  $\epsilon_{in}$  is specified as follows

$$k_{in} = 1.5 (Tu)^2 \quad (V-2)$$

$$\epsilon_{in} = c_{\mu} k_{in}^{3/2} / l_{in} \quad (V-3)$$

and the turbulent intensity,  $Tu = 0.5\%$  and length scale,  $l_{in} = 0.03L$  are used in the present calculations.

For laminar flow calculations, the no-slip condition is specified on the plate. The wall function approach of Launder and Spalding [43] is used for turbulent flow calculation. The other boundary conditions are same as those reported in chapter II-3-2.

#### V.4 Numerical Method

The finite analytic computer code used in the previous chapter is modified for the laminar flow calculation. The well-known TEACH code [86] is used for finite volume calculations with substantial modification. The solution procedures outlined in chapter III-4 are implemented to TEACH code which includes the change of numerical algorithm, from SIMPLE to SIMPLER and the usage of parabolic marching technique with global pressure calculation procedure instead of fully elliptic calculation procedure. Details of TEACH code are well-documented in reference [86] and are not explained here.

## V.5 Results and Discussions

### V.5.1 Laminar Flow Past A Finite Flat Plate

Calculations are performed for  $Re = 10^5$  using both partially parabolic finite analytic method outlined in chapter II-5-1 and elliptic finite volume method given in reference [86]. The same numerical grids are employed for both calculations. Of 135 x-directional numerical grids, 35 grids are placed upstream of plate, 60 grids are placed on the plate and 40 grids are placed in the wake region. The first y-directional u-calculation point over the plate is made  $\Delta y = 0.35 \times 10^{-3}$  and the minimum grid size at the leading edge is made  $(\Delta x)_l = 0.897 \times 10^{-4}$  and the minimum grid size at the trailing edge is made  $(\Delta x)_t = 0.2 \times 10^{-2}$ . Time marching technique is employed for finite analytic calculation with time step  $\Delta t = 1$ . The relaxation factors used in the finite volume calculation are  $\alpha_u = \alpha_v = 0.7$  and  $\alpha_p = 0.5$ . Satisfactory convergence was obtained after 500 iterations for the calculation by finite volume method (  $RES = 0.6617 \times 10^{-3}$  ) and 1000 iterations for the calculation by partially parabolic finite analytic method (  $RES = 0.2891 \times 10^{-3}$  ) where RES denotes sum of mass residuals in pressure correction equation divided by inlet flow rate. It was observed that the velocity components

and pressure field are settled down within 200 iterations and the rest computational efforts are devoted to reduction of the mass residuals to a certain degree.

The predicted pressure distribution on the plate and along the wake centerline is shown in Fig.V-1. As shown in the figure, the magnitude of leading edge interaction is larger than the magnitude of trailing edge interaction. However, the interacting region near the leading edge is smaller than the interacting region near the trailing edge. The comparison of predicted pressure distributions by two different numerical schemes shows that both methods predicted nearly same pressure distribution except near the leading edge. The spurious pressure drop near the leading edge predicted by the present partially parabolic numerical scheme is not observed in the present elliptic calculation or in the nine-point elliptic calculation by Xu and Chen [82]. This fact shows that the partially parabolic assumption may not be appropriate for the simulation of strong leading edge interaction. However, it is also shown that the partially parabolic assumption is valid for the prediction of most of the boundary layer on the plate, the trailing edge interaction and wake development.

The enlarged view of pressure distribution near the trailing edge is shown in Fig.V-2. The magnitude of trailing interaction predicted by finite volume method (  $\sim -0.01563$  ) is slightly higher than that predicted by partially parabolic finite analytic method (  $\sim -0.01474$  ). The origin of this difference is not clearly understood. For reference, the trailing edge pressure predicted by Chen and Patel [81] using elliptic finite analytic method is reported as  $-0.0152$ .

Fig.V-3 shows the distribution of friction coefficient near the trailing edge by both methods. The difference in friction coefficient near the trailing edge by both methods is consistent with the predicted pressure distribution in this region. The stronger interaction cause stronger acceleration near the trailing edge and larger penetration to the upstream of the plate. This figure also clearly shows that the Blasius solution which is based on the boundary layer theory is not adequate for describing the flow field near the trailing edge.

The development of wake centerline velocity is also affected by the trailing edge interaction when the accelerated boundary layer at the trailing edge is destroyed in the wake with adverse pressure gradient. Fig.V-4 shows the predicted wake centerline velocity

profiles in the wake region by two different numerical methods. Little difference is evident, which is consistent with the pressure distribution in this region. There may exist some effects which are caused by using different numerical treatments near the wake centerline. In the finite volume method, the wake centerline velocity do not affect the second nodal velocity in the wake by placing the control volume surface along the wake centerline and applying the symmetry condition and zero normal velocity condition. In finite analytic method, the wake centerline velocity affects the second nodal velocity. These practice may result in smaller wake centerline velocity prediction by finite analytic method. However, this effect seems to be very small since the wake centerline velocity prediction by the present elliptic finite volume method more agree well with those reported in Chen and Patel [81] who used elliptic finite analytic method. It is also noted that these two elliptic calculations have nearly same magnitude of trailing edge interaction. Although there may exist some numerical errors, it may be concluded that the fully elliptic form of discretization equation should be used in order to accurately capture the leading and trailing edge interactions and thereby accurately predict the wake development.

However, the differences of predictions by two different methods are very small and are confined to a relatively small region. These effects are not significant in the practical calculations when the interaction is not weak as in the present problem.

#### V.5.2 Turbulent Flow Past A Finite Flat Plate

Calculations are performed for  $Re = 2.48 \times 10^6$ , corresponding to experiments of Ramaprian et al. [64], using finite volume method with the standard  $k - \epsilon$  turbulence model and the conventional wall function method. Of 120 x-directional numerical grids, 30 grids are placed upstream of plate, 55 grids are placed on the plate and next 35 grids are placed in the wake region. The first y-directional u-calculation point over the plate is made  $\Delta y = 0.164 \times 10^{-2}$  which corresponds to  $y^+ \sim 15$  at the trailing edge. The minimum grid size at the leading edge is made  $(\Delta x)_l = 0.323 \times 10^{-4}$  and the minimum grid size at the trailing edge is made  $(\Delta x)_t = 0.4 \times 10^{-2}$ . The relaxation factors used in this calculation are  $\alpha_u = \alpha_v = \alpha_k = \alpha_\epsilon = 0.8$  and  $\alpha_p = 0.5$ . Satisfactory convergence was obtained after 500 iterations. (  $RES = 0.4203 \times 10^{-3}$  )

Fig.V-5 and Fig.V-6 show the predicted pressure distribution on the plate and along the wake centerline.



There exists a sharp peak at the leading edge and a sharp drop at the trailing edge followed by slower recovery in the wake region. The predicted leading and trailing edge interactions in turbulent flow are much weaker than those in laminar flow by the introduction of turbulence model and insufficient grid refinement in these regions. The grid refinement in these regions is restricted since the  $k - \epsilon$  turbulence model can only be applied to fully turbulent region. Thus the detailed interactions near the leading edge and in the laminar wake region near the trailing edge can not be captured. Patel and Chen [70] explained that the slower recovery of wake centerline pressure to ambient value is related to the isotropic eddy viscosity assumption in the  $k - \epsilon$  turbulence model;

$$\frac{\partial p}{\partial x} \sim - \frac{2}{3} \frac{\partial k}{\partial x} \quad (v-4)$$

The predicted friction velocity distribution on the plate is shown in Fig.V-7 with a comparison of the simple flat plate correlation given in Reference [87]. The strange transition near the leading edge region is due to the improper modelling of transition by  $k - \epsilon$  turbulence model which is also observed in Rhie [83] in the calculation of turbulent flow past a thin airfoil. There exist a substantial difference between the prediction and correlation even in fully turbulent region. It was found

that the underprediction of friction velocity originates from use of the wall function method. The first u-velocity calculation point over the plate is placed around  $y^+ \sim 15$  at the trailing edge to provide proper calculation in the wake region. Thus all the first calculation points on the plate are placed in the buffer layer. However, the wall function region employed in the actual calculation is divided into only two regions, laminar flow region ( $y^+ < 11.63$ ) and fully turbulent region ( $y^+ > 11.63$ ). Thus the wall shear stress on the plate is calculated by way of logarithmic law even though the first u-calculation point is placed in the buffer layer. As a result, the friction velocity is underpredicted even though the velocity is overpredicted. The friction velocity in the turbulent region is corrected using the third nodal u-velocity component which lie in logarithmic layer and the law of the wall. The corrected skin friction velocity is consistent with the overprediction of velocity at the trailing edge which is shown in Fig.V-9. The amount of overprediction in friction velocity is not so serious as shown in the figure. It may be suggested that the predictions will be improved if the wall shear stress in the wall function method is evaluated using the third nodal velocity component in the practical calculation. It was also found that the prediction of friction velocity

was slightly improved when the first grid over the plate was slightly increased to  $y^+ \sim 30$ . However, this practice was abandoned due to the very poor prediction in the wake region which was explained in detail in the previous chapter. It was also observed that the prediction was extremely sensitive to the numerical grid distribution near the wall region which was not observed in the calculation of previous chapter in which two point wall function method was adopted. This sensitivity of prediction to the location of the first grid point is due to the Couette-flow type wall function method usually used in the finite volume calculations. It is noted that the two-point wall function method can not be used in the laminar flow region near the leading edge. The poor prediction of friction velocity can be improved with more advanced turbulence model which can properly describe the flow field near the viscosity-affected near wall region.

Figs.V-8, V-9 and V-10 show the predicted wake centerline velocity, mean velocity distribution and turbulent kinetic energy distribution in the near wake region. The predicted results are only fairly good due to the improper calculation of the boundary layer on the plate. The overprediction of velocity profile at the trailing edge results in higher velocity profiles and smaller kinetic energy distribution in the near wake

region. However, these predictions are not much deviated from the experimental data as shown in these figures. It is shown that as the flow approaches downstream, the flow becomes to lose the memory of boundary layer on the plate and the predictions agree better with the experimental data.

Although only fairly good predictions were achieved due to improper modelling of transition and near wall turbulence, the present study achieves the prediction of whole turbulent flow past a finite flat plate. The predictions can be improved if a suitable turbulence model is used. However, there does not exist a general turbulence model which can properly describe the transition of laminar to turbulent and the flow field in the viscosity-affected near wall region.

CHAPTER VI  
TURBULENT FLOW PAST AXISYMMETRIC BODIES

VI.1 Introduction

In order to design an optimal ship body and ship propelling system, it is very important to have a fundamental understanding and to be able to make an accurate prediction of fluid flow past various ship bodies. The accurate prediction of turbulent flow near the ship stern region is particularly important since many propellers and appendages are located inside the ship stern boundary layers. The flow evolution in this ship stern region is characterized by a rapid growth of boundary layer, a strong viscous-inviscid interaction and a general reduction in the level of turbulence by streamline curvature and pressure gradient [88] which is quite different from that in the hull region. These flow features show that the thin boundary layer approximation is not adequate for describing the flow field around the ship stern region although it may be suitable for describing the flow field around the ship hull region.

A review of the state-of-the-art of available experimental and numerical investigations of turbulent flow past axisymmetric bodies was recently made by Patel and Chen [89]. According to this detailed survey, many reliable experimental informations are available at present which can be used to verify the accuracy of numerical calculations. Chevry [90] made measurements of mean flow and turbulent quantities in the wake of 6 to 1 spheroid. These data provide quite detailed informations on the near wake evolved from the a separated flow just ahead of the tail region of spheroid. The experiment of Patel et al. [88] was conducted on the same model but the separation was eliminated by means of a short conical tail attachment. They measured the development of thick boundary layer on this modified spheroid. Patel and Lee [91] made extensive measurements on the boundary layer and near wake development of turbulent flow past a low-drag body ( F-57 body ). Since above two measurements are made normal to the body surface using boundary layer coordinates, it is difficult to use these data to compare with practical calculation if numerical grids are not generated in the same manner. The extensive experimental data measured by Huang et al. [92,93,94] for four different afterbodies, afterbody-1, afterbody-2, afterbody-3, afterbody-5, provide the detailed

informations on the thick boundary layer and the near wake development. These four bodies have the same streamlined forebody and parallel middle body, but have different afterbodies. All of these bodies have an inflection point and quite dramatic changes in surface curvature which induce a strong favorable and adverse pressure gradient over the stern region and propellor hub region. In the case of afterbody-3, experimental data indicate that there is a small separation bubble around the inflection point. These data have been widely used in the previous studies [89,95,96,97,98,99] for the evaluation of calculation methods and turbulence models. These data are also used in the present study to compare with the calculated results.

Many numerical efforts have been made in the past decade to compute this complex flows near the tail region of the axisymmetric bodies. These efforts have involved with different approximations made in the governing equations, with different turbulence models and with different calculation methods. According to the survey by Patel and Chen [89], the previous calculation methods can be broadly classified into three categories, the non-interactive numerical solution of thin layer or boundary layer equation with the pressure field given by experimental data [91], the viscous-inviscid interaction

method [100,101,102] and the global numerical solutions of partially parabolic or elliptic equations [89,98]. Among these methods, the global numerical solution method is the most attractive since this method does not require the experimental informations of pressure field or another numerical solution of inviscid flow. It is interesting to note that the recently developed calculation methods outlined in chapter-I have not been applied to this problem except the works of Patel and Chen [89]. The calculation methods used by Muraoka [97,103] or Markatos [98] are not general and can only be applicable when the numerical grids are generated in a special algebraic manner. It is also noted that most of previous calculations start at the middle of body to avoid the numerical difficulties associated with the calculation near the leading edge.

In the present study, numerical solutions of partially parabolic or fully elliptic form of Reynolds-averaged Navier-Stokes equations in numerically generated, nonorthogonal coordinates are obtained through the calculation procedure outlined in chapter-II and chapter-III. The complex, physical geometry is resolved by the use of nonorthogonal, body-fitted coordinates which are generated through the solutions of Poisson equations with appropriate grid control functions. The governing



equations are written in the transformed coordinates by the partial transformation where the original orthogonal velocity components are left as dependent variables. These transformed governing equations are discretized using the finite analytic method by Chen and Chen [3] which was already explained in chapter-II. The finite volume method is also applied to some problems in order to provide comparisons of two different numerical schemes in the prediction of turbulent boundary layer and wake of axisymmetric bodies. The one-velocity staggered grid technique is employed for all the calculations presented in this chapter. The derivation of pressure and pressure correction equations in this grid configuration is given in chapter III-3-1.

Since there are strong pressure gradient and streamline curvature effects on the flow near the stern region, some considerations should be made for the selection of turbulence model. The zero equation models of Cebeci-Smith model or Baldwin-Lomax model and the one equation model have been used in the earlier calculations [91,95,96,101] with some modifications. However, the  $k - \epsilon$  turbulence model with wall function method has been the most widely employed in the recent calculations [89,97,102,103]. The defect of the  $k - \epsilon$  turbulence model in the calculation of flow involving pressure gradient

effect is well explained by Rodi and Scheuerer [54]. The poor performance of the current  $k - \epsilon$  turbulence model is caused by inaccurate modelling of the  $\epsilon$  - equation. For example, the length scale determined using the  $\epsilon$  - equation rises steeper near the wall region in the adverse pressure gradient region while the experimental data shows that the length scale is virtually independent of the pressure gradient. By this reason, the turbulence models using the empirical length scale specification near the wall region yield much better predictions for the adverse pressure gradient boundary layers than does the current  $k - \epsilon$  turbulence model. Rodi and Scheuerer [54] suggest use of Hanjaliac and Launder's model [104]. On the other hand, Chen and Patel [56] developed a two-layer turbulence model to consider the pressure gradient and streamline curvature effects and flow separation. Their model has been successfully applied to the prediction of turbulent flow past axisymmetric bodies.

In the present study, some calculations are performed using the conventional  $k - \epsilon$  turbulence model and wall function method in which the pressure gradient effect is embodied in the wall function while some calculations are performed using the two-layer turbulence model of Chen and Patel [56].

## VI.2 Two-Layer Model

In the two-layer model, the calculation region is divided into two regions; a near wall region and an outer region, as shown in the Fig.VI-1. The boundary between the near wall region and the outer region is placed around  $y^+ \sim 100$ . The near wall region includes the laminar sublayer, buffer layer and a part of logarithmic layer adjacent to the body. In this region, calculations are carried all the way to the wall using the one equation turbulence model where the rate of turbulent kinetic energy dissipation  $\epsilon$  and the turbulent eddy viscosity  $\nu_t$  is specified by algebraic relations to consider the wall proximity viscous effects. The outer region is the upper region of the near wall region over the body and the whole wake region. The standard  $k - \epsilon$  turbulence model is used to simulate this fully turbulent region.

Three one equation turbulence models such as Hassid and Poreh's model [48], Norris and Reynolds's model [49], Wolfshtein's model [47] were tested in the course of present investigation. However, the Wolfshtein model is chosen in the present study by the simple reason that the other two models slightly overpredict the wall shear stress compared with the Wolfshtein model.

In the Wolfshtein model, the turbulent eddy viscosity  $v_t$  and the rate of turbulent kinetic energy dissipation  $\epsilon$  is specified by a simple algebraic form as following;

$$\epsilon = \frac{k^{3/2}}{l_\epsilon} \quad (\text{VI-1})$$

$$v_t = c_\mu k^{1/2} l_\mu \quad (\text{VI-2})$$

where the length scales  $l_\epsilon$ ,  $l_\mu$  contain the viscosity damping effect in the near wall region in terms of turbulent Reynolds number  $R_k$ .

$$l_\mu = c_1 n [ 1 - \exp( - R_k / A_\mu ) ] \quad (\text{VI-3})$$

$$l_\epsilon = c_1 n [ 1 - \exp( - R_k / A_\epsilon ) ] \quad (\text{VI-4})$$

where

$$R_k = Re n k^{1/2}, \quad c_\mu = 0.09, \quad c_1 = \kappa c_\mu^{-3/4},$$

$$A_\epsilon = 2 c_1, \quad A_\mu = 70.$$

and  $n$  is the normal distance from the wall.

These constants are evaluated by Chen and Patel [56] and somewhat different from those reported in Wolfshtein [47] or Yap [58].

### VI.3. Numerical grids and Solution Domains

The numerical grids are generated using the grid generation technique outlined in chapter II-4-2. The generated numerical grids for four different afterbodies are shown in Fig.II-4. The detailed informations on the solution domains are given in Table 4 and Table 5. These solution domains are large enough to capture the whole viscous-inviscid interactions and wake development.

### VI.4 Boundary Conditions

In the half body calculations using the wall function method, the upstream inlet conditions were specified by following simple correlations.

$$\frac{u}{u_{\tau}} = \frac{1}{\kappa} \ln y^{+} + B \quad (\text{VI-5})$$

$$k = c_{\mu}^{-1/2} u_{\tau}^2 \left( 1 - \frac{y}{\delta} \right) \quad (\text{VI-6})$$

$$\epsilon = c_{\mu}^{3/4} k^{3/2} / \kappa y \quad (\text{VI-7})$$

where  $y^{+} = Re u_{\tau} y$ ,  $\kappa$  is von-Karman constant,  $\kappa = 0.42$  and  $B = 5.5$ . The boundary layer thickness  $\delta$  and friction velocity  $u_{\tau} = (\tau_w / \rho)^{1/2}$  are provided by experimental data.

The wall function method employed in the present study is a little different from the conventional wall

function method usually used in the control volume method. In the present study, it is assumed that at least two  $u$ -calculation nodes are located in the logarithmic region. The friction velocity  $u_\tau$  is first determined by iterative method using the velocity components in the third node which is obtained from the previous iteration and following the generalized form of the law of the wall [4] in which the pressure gradient effects on the flow in the wall region are taken into consideration.

$$\frac{q}{u_\tau} = \frac{1}{\kappa} \left\{ \ln \left[ \frac{4}{\Delta_\tau} \frac{(1 + \Delta_\tau n^+)^{1/2} - 1}{(1 + \Delta_\tau n^+)^{1/2} + 1} \right] + 2 \left[ (1 + \Delta_\tau n^+)^{1/2} - 1 \right] \right\} + B + 3.7 \Delta_p \quad (\text{VI-8})$$

where  $n^+ = \text{Re } u_\tau n$  is dimensionless, normal distance from the surface,  $\Delta_p = -\Delta p / (\text{Re } u_\tau^3)$  is the dimensionless pressure gradient on the body surface,  $\Delta_\tau$  is the dimensionless shear stress gradient and is assumed to be  $\frac{1}{2} \Delta_p$ ,  $q$  is the magnitude of velocity or  $q = (u^2 + v^2)^{1/2}$ ,  $\kappa = 0.418$  is the von-Karman constant, and  $B = 5.45$ .

Then the magnitude of velocity in the second node  $q_2$  can be calculated using this friction velocity and Eq.(VI-8) while the velocity components parallel to  $\xi$  - grid line are calculated by the following relations.

$$q_3^\xi = \frac{\sqrt{g_{11}}}{J} r ( b_1^1 u_3 + b_2^1 v_3 ) \quad (\text{VI-9})$$

$$q_2^\xi = \frac{q_2}{q_3} q_3^\xi \quad (\text{VI-10})$$

where the geometric coefficients  $J$  and  $b_j^i$  are given in Eqs. (II-58) and (II-59) and  $g_{11} = x_\xi^2 + y_\xi^2$ .

The  $u$ ,  $v$ ,  $k$ ,  $\varepsilon$  values at the second node are evaluated using the friction velocity and following relations.

$$u_2 = \frac{x_\xi}{\sqrt{g_{11}}} q_2^\xi \quad (\text{VI-11})$$

$$v_2 = \frac{y_\xi}{\sqrt{g_{11}}} q_2^\xi \quad (\text{VI-12})$$

$$k_2 = c_\mu^{-1/2} u_\tau^2 \quad (\text{VI-13})$$

$$\varepsilon_2 = u_\tau^3 / \kappa y \quad (\text{VI-14})$$

The details of this two-point wall function method is given in Chen and Patel [4]. The advantage of this two-point wall function method is that the sensitivity of the solution to the location of the first grid point, which is found in other Couette-flow type wall function method usually used in the control volume method, is removed.

In the two-layer calculations, the upstream  $u$ -velocity boundary condition near the wall is specified by following Spalding's velocity law:

$$y^+ = u^+ + e^{-\kappa B} \left[ e^{-\kappa u^+} - 1 - \kappa u^+ - (\kappa u^+)^2/2 - (\kappa u^+)^3/6 \right] \quad (\text{VI-15})$$

The inlet boundary conditions for turbulent quantities in the inner layer are specified by following correlations given in Patel and Chen [70] while turbulent quantities in the outer layer is specified by Eq. (IV-6) and Eq. (IV-7).

$$\begin{aligned} k^+ &= 0.05 (y^+)^2 & y^+ < 5 & \quad (\text{VI-16}) \\ &= 1.25 + 0.325 (y^+ - 5) & 5 < y^+ < 15 \\ &= 4.5 + (y^+ - 15) / 37.5 & 15 < y^+ < 60 \\ &= 3.3 & 60 < y^+ < 120 \end{aligned}$$

and

$$\begin{aligned} \varepsilon^+ &= 0.1 y^+ / 120 & y^+ < 12 & \quad (\text{VII-17}) \\ &= 1 / \kappa y^+ & y^+ > 12 \end{aligned}$$

The dimensionless quantities in these equations are defined as  $k^+ = k / u_\tau^2$ ,  $\varepsilon^+ = \varepsilon / (\text{Re } u_\tau^4)$ ,  $y^+ = \text{Re } u_\tau y$  and  $u^+ = u / u_\tau$ .

The other boundary conditions are same as those reported in chapter II-3-2.



## VI.5 Results and Discussions

### VI.5.1 Prediction of Turbulent Flow Past Axisymmetric Bodies using Wall Function Method

Calculations are performed for three different axisymmetric bodies of DTNSRDC, afterbody-1, afterbody-2 and afterbody-5. Both finite analytic method and finite volume method are used to solve Reynolds-averaged Navier-Stokes equations with the  $k - \epsilon$  turbulence model and two-point wall function method. The same numerical grids and boundary conditions are employed for both calculations. The time marching technique is employed for finite analytic calculations with time step  $\Delta t = 1$  and pressure relaxation factor  $\alpha_p = 0.2$ . The relaxation factors used in the finite volume calculations are  $\alpha_u = \alpha_v = 0.7$ ,  $\alpha_k = \alpha_\epsilon = 0.8$  and  $\alpha_p = 0.5$ .

Figure VI-2 shows the convergence history of both calculations for the afterbody-1. The calculation by finite volume method converges faster than that by finite analytic method in the earlier stage. However, the calculation by finite volume method has a limit in convergence. This limited convergence originates from use of one velocity staggered grid method which is also reported in Shyy et al. [24]. As will be shown in the

next chapter, use of two velocities staggered grid method eliminates this deficiency. The calculation by finite analytic method monotonically converges without a limit although the same one velocity staggered method is used. Satisfactory convergence,  $RES < 2 \times 10^{-3}$ , was obtained within 100 iterations for both calculations where RES denotes the sum of mass residuals in the pressure correction equation divided by the inlet flow rate. However, the pressure and velocity fields are settled down within 50 iterations as shown in Figs.VI-3 to VI-8. These figures show that convergences of boundary layer and wake development by finite analytic method are a little faster than those by finite volume method. This is due to the fact that the relaxation factors in the finite volume method is same in the whole computational domain while relaxation factors in the finite analytic method are changed according to the size of numerical grid and the magnitude of convecting velocity components which is another novel nature of the finite analytic method.

Figs.VI-9 to VI-11 show the converged pressure distributions,  $C_p = 2(p - p_\infty) / (\rho U_\infty^2)$ , on the body surface and along the wake centerline for afterbody-1, afterbody-2, and afterbody-5. As shown in the figures, the agreement between the predictions and measured data is excellent. The differences in the prediction by two

different methods are small and are confined to only the region of inlet, propellor hub and trailing edge. The dramatic change of pressure in the stern and propeller hub region is well predicted. The pressure is recovered in the wake at a distance roughly  $0.3L$  from the trailing edge of the body. Since the solution domain of the present study extends far beyond this distance, the whole viscous-inviscid interaction is completely captured.

Figs.VI-12 to VI-14 show the comparison between predicted and measured friction velocity. Fairly good agreement between predictions and measurements was obtained in the stern region. The slight disagreement in the upstream region may be due to the simple imposition of inlet conditions. The results can be improved if the accurate inlet conditions of velocity and turbulent quantities are provided by experimental data. The finite volume method consistently overpredicts the friction velocity in all calculations. As shown in the figures, there is steep drop of friction velocity in the adverse pressure gradient region and the flow is very near to separate at  $x = 0.93L$  for afterbody-5,  $x = 0.97L$  for afterbody-2 but no flow separation is predicted.

Figs.VI-15 to VI-17 show the radial pressure distribution in the stern and near wake region.

Satisfactory agreements between the prediction and measurements are made although they are not as good as those of pressure distribution on the body surface. The small discrepancies may be partly due to the difficulties in the measurements. The differences in prediction between two methods are small and are again confined to the regions of propellor hub and trailing edge. We note that the viscous-inviscid interaction extends radially to  $0.35L$  from the surface of the body near the trailing edge. This fact shows the failure of thin boundary layer approximation for the present problem and some careful considerations should be made for the selection of radial direction solution domain. In the present study, the computational domain extends radially to  $y_u \sim 1.1L$  as shown in Table 4.

Comparisons of the predictions with the measured data for both axial and radial velocity components is seen in Figs. VI-18 to VI-20. The agreements between measured data and the predictions by finite analytic method is excellent. The finite volume method slightly overpredicts the axial velocity component, especially near the tail region and in the near wake region. One may notice the thickening of boundary layer in the tail and in the near wake region which is consistent with the pressure variation in these regions.

The turbulent kinetic energy in the tail region and in the near wake region is overpredicted as shown in the Figs.VI-21 to VI-23. The amount of overprediction is more severe in the prediction of afterbody-5 and afterbody-2 than that in the afterbody-1. The differences in prediction by two methods are small and consistent with the predicted velocity profiles. It is noted that the predictions of turbulent kinetic energy made by Huang and Chang [99] agrees quite well with experimental data. They used a different near wall treatment instead of the wall function method. As will be shown in the next section, the use of two-layer model slightly improve the prediction. From these observations, the overprediction of turbulent kinetic energy in the tail and near wake regions seems to be originated from the inadequacy of  $k - \epsilon$  turbulence model in the prediction of fluid flow involving pressure gradient and streamline curvature effects. The other predictions made by Patel and Chen [89], Muraoka [97,103] using the  $k - \epsilon$  turbulence model and the wall function method show the similiar trend as in the present prediction.

Overall agreements between measurements and predictions made using both methods are encouraging. For most of the stern region and near wake region, the

pressure and velocity components are accurately predicted. But some considerations must be made on the selection of turbulence model especially on the near wall treatment in order to accurately predict the turbulent quantities for the flow involving pressure gradient and streamline curvature effects. Although predictions by finite analytic method are in favor for all the calculations, the differences in the predictions by two different numerical schemes are small. It shows that the selection of numerical scheme is not important for calculating of boundary layer type flow where the numerical false diffusion effects are very small. The overall success of present calculations are mainly due to the use of two-point wall function method. However, the use of two-point wall function method is limited and can not be confidently applied to the flow involving separation. Calculations are also made using the partially parabolic form of finite analytic method although it is not presented here. The differences in predictions between partially parabolic calculations and elliptic calculations were found to be negligible. This fact shows that partially parabolic assumption is a good approximation if there is no flow separation.

### VI.5.2 Prediction of Turbulent Flow Past Axisymmetric Bodies by Two-Layer Model

Calculations are performed for four different axisymmetric bodies of DTNSRDC, afterbody-1, afterbody-2, afterbody-3 and afterbody-5. The partially parabolic form of finite analytic method explained in chapter II-5-3 is employed to solve Reynolds-averaged Navier-Stokes equations with two-layer model. The FLARE approximation where the  $\xi$ -directional convecting velocity ( $D_\phi$  in Eq. (II-84)) is zero if it is less than zero is employed in order to properly handle the flow separation occurring during the iteration process. The introduction of FLARE scheme is come from the observation made by Ramakrishnan and Rubin [105]. They found that the FLARE scheme was the most stable for the calculation of small separated, high Reynolds number flow. The time marching technique is employed for iteration process with time step  $\Delta t = 0.002$ . Satisfactory convergence was obtained after 1000 time marching.

Figs. IV-24 to IV-27 show the converged pressure distributions,  $C_p = 2(p - p_\infty) / (\rho U_\infty^2)$ , on the body surface and along the wake centerline. As shown in the figures, the agreements between the measured data and the predicted results are fairly good. The pressure is a little

sensitive to the shape of the body surface as compared with predictions using the wall function method. It is due to the fact that pressure is calculated all the way to the wall in the two-layer calculations while pressure on the body surface is extrapolated from the wall function region to the body surface using a simple linear function in the calculations using the wall function method. It is noted that the predictions for afterbody-3 are possible by use of the two-layer model.

Figs.IV-28 to VI-31 show the comparisons between the predicted and measured friction velocity. Although fairly good agreements are made in the hull region, systematic overpredictions are made in the tail region. The overprediction of friction velocity in the tail region may be partly due to the numerical difficulties in the generation of fine numerical grid near the wall in the tail region of the body and may be partly due to the fact that even the one equation turbulence model tends to overestimate the velocity in the near wall region indicating a reduction of turbulence length scale in the tail region. It is noted that flow separation was not predicted in the present calculation for afterbody-3 in which experimental data indicate there exist a small separation bubble around the inflection point. Figs.VI-32 to VI-33 show the predicted friction velocity for



afterbody-3 and afterbody-5 by Chen and Patel [56] using the same two-layer model and the nine-point finite analytic method. There exist substantial differences between two predictions. The origin of these differences can not be clearly explained since the details of pressure-velocity coupling technique are not reported in Chen and Patel [56]. However, the present predictions does not show the strange overshoots which is observed in the predictions by Chen and Patel [56].

Fairly good agreements between the predictions and the measured data were obtained for both the axial and radial velocity components which can be seen in Figs.VI-34 to VI-37 except the axial velocity component is slightly overpredicted in the tail region as explained before.

The overprediction of turbulent kinetic energy in the tail region and in the near wake region in the calculations using the wall function method is slightly improved by using the two-layer model as shown in the Figs.VI-38 to VI-40. This observation shows that the defect of the  $k - \epsilon$  turbulence model in the calculation of turbulent boundary layer and wake involving pressure gradient and curvature effects is a little relieved by the introduction of one equation turbulence model for the calculation of flow near the wall region.

Overall agreements between measurements and predictions made by the present finite analytic method with both wall function method and two-layer model are satisfactory. For most of the stern region and the near wake region, velocity components and turbulent quantities are accurately predicted except that the axial velocity component is overpredicted in the two-layer calculation in the tail region of body. The origin of this overprediction should be clearly investigated. It was found that the inlet condition severely influence the overall solution. Thus the proper ways of specifying the inlet condition should be sought. When it is considered the fact that the two-layer calculation not only provides the numerical solutions of laminar sublayer, buffer layer and a part of logarithmic layer without excessive numerical efforts but also the two-layer model can be applied to separated flow region to which the wall function method can not be confidently applied, the use of two-layer model is highly recommended.

CHAPTER VII  
TURBULENT FLOW PAST FINITE AXISYMMETRIC BODIES

VII.1 Introduction

Incompressible turbulent flow past axisymmetric bodies of finite length has been the subject of numerous investigations, as it is of major importance in aerodynamics and hydrodynamics. In most of previous studies, the calculations start at the middle of body with inlet conditions provided by curve fits of experimental data [97], simple flat plate correlations [56,89] or numerical solutions of boundary layer equation up to the inlet location [103]. However, these calculations do not provide the solution of leading edge interaction and initial development of boundary layer near the leading edge. The resulting solutions are dependent on the inlet conditions as shown in the previous chapter or the location of inlet boundary [81].

In order to avoid above deficiencies in the half body calculations and to obtain the solution of the whole flow field, the calculation should start far upstream of the body with inlet conditions provided by uniform flow

condition. However, numerical solutions of incompressible turbulent flow past axisymmetric bodies, including the leading edge interaction, boundary layer development on the body, the trailing edge interaction and the wake development, is rarely seen in the literature. For two dimensional flow situations, Rhie and Chow [16] obtained the numerical solutions of turbulent flow past airfoils with or without angle of attack using the  $k - \epsilon$  turbulence model and the wall function method. The C-type, body-fitted coordinates were introduced to handle the complex physical geometry and the SIMPLE numerical algorithm was employed with non-staggered grid arrangement. Detailed numerical solutions of laminar and turbulent flow past a circular cylinder were obtained by Majumdar and Rodi [55]. They used an orthogonal coordinate system with staggered grid arrangement. The  $k - \epsilon$  turbulence model was also employed in this study. An improvement of prediction by using the zero equation turbulence model in the near wall region instead of the wall function method is reported.

Since the physical solution domain has a complex geometry which requires the use of nonorthogonal, body-fitted coordinate system, some considerations should be made about the choice of calculation methods in this coordinate system. Among the several calculations methods outlined in chapter I, the two velocities staggered grid

method developed by Maliska and Raithby [30] keeps the novel nature of staggered grid arrangement without any numerical problems encountered in the use of one velocity staggered grid method although the use of this grid configuration requires a little more computer storage and computing time by storing both velocity components at each control volume surface. It is reported in Maliska and Raithby [30] that the computational efforts using this two velocities staggered grid method is not so grave as it looks by its relatively earlier convergence. This grid configuration allows to use H-type grid as well as C-type grid to which the one velocity staggered grid method may not be applicable. Therefore the two velocities staggered grid method is employed for all the calculation presented in this chapter with some modifications made from the original method by Maliska and Raithby [30].

In the initial phase of present study, some computational experiments were made using the finite analytic method and the two-point wall function method outlined in the previous chapter. However, this practice was abandoned since the two-point wall function method did not properly describe the laminar flow near the leading edge even in the simple flat plate calculation. It was also found that the general law of the wall, Eq.(VI-8), had singularity in the region of favorable pressure

gradient giving a negative argument in the logarithmic function. Although this two-point wall function method has a salient feature of removing the sensitivity of the first grid location on the solution, the application of this method to general flow situation is limited. After testing other possibilities, it was decided to use the same numerical scheme and the near wall treatment as those reported in Rhie and Chow [16] or Majumdar and Rodi [55]. These authors used the finite volume method and the Couette flow type wall function method by Launder and Spalding [43]. However, the numerical solutions using this wall function method usually suffer the sensitivity of the solution on the numerical grid and the overprediction of the axial velocity component in the adverse pressure gradient region due to the neglect of the pressure gradient effect in the wall function.

In the present study, numerical solutions of incompressible turbulent flow past axisymmetric bodies are obtained using the  $k - \epsilon$  turbulence model and the wall function method. The complex, physical solution domain is resolved by the use of numerically generated, nonorthogonal, body-fitted coordinates. The governing equations are written in the transformed domain with the cylindrical velocity components as the dependent variables. The transformed governing equations are

discretized using the finite volume method with hybrid numerical scheme as outlined in chapter II-6. As mentioned before, the two velocities staggered grid method is adopted and the derivation of pressure and pressure correction equations in this grid configuration is given in chapter III-3-2.

Calculations are performed for four axisymmetric bodies of DTNSRDC using the calculation procedure outlined in chapter III-4. The numerical results are compared with the experimental data.

#### VII.2. Numerical Grids and Solution Domains

In the initial phase of present study, use of C-type numerical grid was seriously considered due to the roundness of forebody shape. The introduction of two velocity staggered grid method should be understood within this context since the use of one velocity staggered grid method in C-type grid may cause numerical instability. However, the use of C-type grid was abandoned due to the difficulties in the specification of boundary conditions. The numerical difficulties related with the use of C-type grid are well reported in Rhie [83] and Chen and Patel [106]. For example, Chen and Patel [106] first performed calculation using H-type grid. Then the boundary

conditions for calculation by C-type grid are specified by these calculated results. In order to avoid these difficulties, it was decided to change the forebody shape up to  $x_i = 0.3L$  and to use H-type grid. The forebody shape was changed to orgive shape using following simple equations;

$$R = R_{\max} - R_i + ( R_i^2 - ( x - x_i )^2 )^{1/2} \quad (\text{VII-1})$$

where  $R_{\max}$  is the maximum radius of body and  $R_i$  is defined as following relation.

$$R_i = ( R_{\max}^2 + x_i^2 ) / ( 2 R_{\max} ) \quad (\text{VII-2})$$

As will be seen later, the change of forebody shape does not seriously influence the develoment of boundary layer in the region of middle body and tail since these DTNSRDC bodies have long parallel middle bodies.

The numerical grids are generated using the method outlined in chapter II-4-2. The partial view of generated numerical grids are shown in Fig.II-5. The x-directional numerical grids on the body are distributed uniformly while those in the upstream of body and in the wake region are stretched with expansion ratio 1.2 ~ 1.3. Table 6 shows the detailed information on the flow condition, grid and solution domains used in the present calculation. These solution domains are large enough to capture the



whole leading and trailing edge interactions. Of the 151 x-directional numerical nodes, 20 nodes are placed upstream of body, 101 nodes are placed on the body and the rest 30 nodes are placed in the wake.

### VII.3 Boundary conditions

The inlet boundary conditions are specified by following uniform flow conditions.

$$u = 1, v_x = 0, k = k_{in}, \epsilon = \epsilon_{in} \quad (\text{VII-3})$$

where  $k_{in}$  and  $\epsilon_{in}$  is specified as follows

$$k_{in} = 1.5 (Tu)^2 \quad (\text{VII-4})$$

$$\epsilon_{in} = c_{\mu} k_{in}^{3/2} / l_{in} \quad (\text{VII-5})$$

and the turbulent intensity,  $Tu = 0.5\%$  and length scale,  $l_{in} = 0.001L$  are used in the present calculations.

In the wall function region, the wall function approach of Launder and Spalding [43] is used. It is based on the one-dimensional Couette flow analysis. The wall shear stress,  $\tilde{\tau}_w$  is expressed as a function of the second nodal velocity component parallel to the wall,  $\tilde{U}_p$  as follows;

$$\tilde{\tau}_w = -\lambda_w \tilde{U}_p \quad (\text{VII-6})$$

where the coefficient,  $\lambda_w$  is determined from the two-part universal velocity profile.

$$\lambda_w = \mu / \delta n \quad \text{if } y_p^+ < 11.63 \quad (\text{VII-7})$$

$$\lambda_w = \rho c_\mu^{1/4} k_p^{1/2} \kappa / \ln(E y_p^+) \quad \text{if } y_p^+ > 11.63 \quad (\text{VII-8})$$

In these equations,  $\kappa = 0.418$  is the von-karman constant and  $E = 0.9793$  is the roughness parameter,  $\delta n$  is the normal distance from the wall and  $y_p^+$  is defined as follows;

$$y_p^+ = \rho c_\mu^{1/4} k_p^{1/2} \delta n / \mu \quad (\text{VII-9})$$

Since the dependent variables of momentum equations are cylindrical velocity components, the shear force,  $\tilde{T}_w = A_w \tilde{\tau}_w$  should be expressed in terms of its cylindrical components. The shear force,  $\tilde{T}_w$  acting along  $\tilde{U}_p$  direction is decomposed into two components  $T_{wx}$  and  $T_{wy}$  along x and y directions respectively as follows;

$$T_{wx} = - \lambda_w A_w [ (1 - n_1 n_1) u_p - n_1 n_2 v_p ] \quad (\text{VII-10})$$

$$T_{wy} = - \lambda_w A_w [ (1 - n_2 n_2) v_p - n_1 n_2 u_p ] \quad (\text{VII-11})$$

where  $n_1$  and  $n_2$  are the components of the unit vector normal to the wall along the x and y directions respectively.

$$n1 = - ( y_{\xi} / \sqrt{\gamma} )_w \quad (\text{VII-12})$$

$$n2 = ( x_{\xi} / \sqrt{\gamma} )_w \quad (\text{VII-13})$$

and the geometric coefficient  $\gamma = x_{\xi}^2 + y_{\xi}^2$ .

The equations for turbulent kinetic energy,  $k$  and its dissipation rate,  $\epsilon$  also need a special treatment for the near wall cell. In the equations for  $k$ , the diffusion flux through the wall surface is set equal to zero while the source terms are modified as follows.

The generation and dissipation term of turbulent kinetic energy equation is approximated as follows using the Couette flow assumption;

$$\int_{\Delta V} G \, dV \cong |\tilde{\tau}_w| |\tilde{U}_p| \Delta V / \delta n \quad (\text{VII-14})$$

$$\int_{\Delta V} \epsilon \, dV \cong c_{\mu}^{3/4} k_p^{3/2} U_p^+ \Delta V / \delta n \quad (\text{VII-15})$$

where  $U_p^+$  is given by

$$U_p^+ = y_p^+ \quad \text{if } y_p^+ < 11.63 \quad (\text{VII-16})$$

$$U_p^+ = \frac{1}{\kappa} \ln( E y_p^+ ) \quad \text{if } y_p^+ > 11.63 \quad (\text{VII-17})$$

The value of  $\epsilon$  in the near wall cell is obtained from the local value of  $k$  using the concept of local equilibrium between production and dissipation of turbulent kinetic energy as follows;

$$\epsilon_p = c_\mu^{3/4} k_p^{3/2} / (\kappa \delta n) \quad (\text{VII-18})$$

Since the details of this wall function method is well-documented in Launder and Spalding [43] or Peric [18], only a brief description of this method is given in the present study.

The other boundary conditions are same as those reported in chapter II-3 and not explained here.

#### VII.4 Results and Discussions

Calculations are performed for four axisymmetric bodies using the finite volume method outlined in chapter II-6. The relaxation factors used in the present calculations are  $\alpha_u = \alpha_v = 0.7$ ,  $\alpha_k = \alpha_\epsilon = 0.8$  and  $\alpha_p = 0.5$ .

Fig.VII-1 shows the convergence history of calculations for afterbody-1. The total mass residual decreases rapidly for the first fifty iterations. However, the rate of convergence slows down after fifty iterations. It is also noted that the calculation using the coarse grid gives a faster convergence. Satisfactory convergence,  $\text{RES} < 2 \times 10^{-4}$ , was obtained within 300 iterations although computations are continued to 500 iterations where RES denotes sum of mass residuals in the

pressure correction equation divided by inlet flow rate. In general, the velocity components and pressure field were settled down within 100 iterations and the rest computational efforts are devoted to the development of far wake.

Figs.VII-2 and VII-3 show the grid dependence tests for pressure and friction velocity. They show that present calculations are generally grid independent except for a small region near the leading and trailing edges. It should be noted that the first three y-directional computational grids near the wall were fixed for both calculations to avoid errors caused by the near wall treatment.

Figs.VII-4 to VII-7 show the converged pressure distributions,  $C_p = 2(p - p_\infty) / (\rho U_\infty^2)$ , on the body surface and along the wake centerline for four different afterbodies. The leading and trailing edge interactions as well as the dramatic change of pressure in the regions of stern and propellor hub are well predicted. The agreement between the predictions and measured data is also fairly good except pressure is overpredicted in the tail region of afterbody-3. This overprediction is the result of the inadequacy of the present  $k - \epsilon$  turbulence model with the present wall function method for describing the flow field

in this region. It is noted that although the magnitude of leading edge interaction is large, the interacting zone is confined to a small region. The pressure is recovered in the wake in a distance roughly  $0.3L$  from the trailing edge of the body. Since the solution domain of the present study extends far beyond this distance, the whole viscous-inviscid interaction is completely captured.

Figs.VII-8 to VII-11 show the predicted friction velocity distribution with comparison with measured data. There exist a strange transition near the leading edge which is originated from improper modelling of transition by present  $k - \epsilon$  turbulence model. The friction velocity is underpredicted in the region of middle body in spite of velocity components in this region are somewhat well predicted as will be shown in the next figures. As well explained in chapter V, the origin of this discrepancy is come from the usage of the wall function method in the fact that although the first normal calculation point is placed in the buffer layer ( $y^+ \sim 20$ ), the friction velocity is calculated by the logarithmic law. The increase of the size of the first grid would improve the prediction of friction velocity. However, this practice was abandoned due to the very poor overall predictions especially in tail region of the body. From these observations, it is suggested that the wall function

region should be at least divided into three regions, laminar sublayer, buffer layer and logarithmic layer as reported in Lee [107]. The use of more elaborated wall functions by Launder and Chieng [108] will also improve the predictions. However, these practices are not performed in the present investigation. A substantial overprediction of friction velocity is observed in the tail region of body, especially in the predictions for afterbody-3 and afterbody-5. These overprediction is due to the neglect of pressure gradient effect both in the turbulence model and in the wall functions. However, as mentioned before, the use of general law of the wall, Eq. (VI-8), leads to singularity in the region of favorable pressure gradient. It is noted that flow separation is not predicted for afterbody-3 although experimental data indicate that there exist a small separation bubble around the inflection point. It is believed that the use of more advanced turbulence model, for example the two-layer turbulence model, will remove all the deficiencies arised from using wall function method.

Comparisions of the predictions with the measured data for both axial and radial velocity components are shown in Figs. VII-12 to VII-15. The agreements between measured data and the predictions are fairly good except the velocity components are a little overpredicted in the

tail region and in the near wake region due to the neglect of pressure gradient effect in the turbulence model and in the wall functions. However, the thickening of boundary layer in the tail and in the near wake region is well predicted. It is noted that as shown in these figures, the change of forebody shape does not seriously influence the boundary layer development in the region of middle body, thereby, that in the stern and wake region.

As shown in Fig.VII-16 to VII-18, the predictions of turbulent kinetic energy agree fairly well with experimental measurements. However, when the overprediction of velocity components in the regions of tail and near wake is considered, the agreements of predictions with experimental data are not improved results over the predictions presented in the chapter VI. If the velocity components are accurately predicted, it is anticipated that the turbulent kinetic energy in the regions of tail and near wake will be overpredicted.

Overall agreements between measurements and predictions made by the present numerical method are encouraging. For the most of body and near wake region, the pressure and velocity components are fairly well predicted. However, some considerations must be made on the selection of turbulence model especially on the the



near wall treatment in order to accurately predict the turbulent quantities for the flow involving pressure gradient and streamline curvature effects. The use of more elaborated wall functions or introduction of more advanced turbulence model is indeed needed to properly simulate the flow field in these situations. These improvements on turbulence model are remained for future study and are not pursued further in the present study. However, the present calculation method provides the predictions of whole turbulent flow field past axisymmetric bodies which are not yet reported in the literature. The present calculations also gives an example of the successful use of two velocity staggered grid method in the calculation of fluid flow in complex geometry which has not been reported in the literature since the original works of Maliska and Raithby [30].

CHAPTER VIII  
CONCLUSIONS AND RECOMMENDATIONS

A numerical study of laminar and turbulent flow past finite two dimensional and axisymmetric bodies is presented. The available calculation methods for incompressible flow in a complex geometry, grid generation techniques and turbulence models are discussed. The numerical formulations and grid generation techniques and solution procedures employed in the present study are presented. Calculations are performed for laminar and turbulent flow past a thin flat plate, turbulent flow past axisymmetric bodies with changing body shapes, solution domains, turbulence models and numerical methods. Major contributions and findings of present study may be summarized as follows;

(1) Wake function method for the prediction of turbulent flow past a flat plate is developed. An accurate prediction of wake flow past a flat plate is made without the detailed calculations of the near wall region and the laminar wake region.

(2) Numerical solutions of a complete flow field past a finite flat plate and axisymmetric bodies which

include the leading edge interaction, the boundary layer development on the body, the trailing edge interaction and the wake development, are obtained. The influence of inlet boundary conditions and the location of inlet solution boundary on the solution which is observed in the half body calculations is removed in the full body calculation. It is observed that the magnitude of the leading edge interaction is large, but the interacting zone is confined to a relatively small region. It is also found that the partially parabolic form of Navier-Stokes equations is not appropriate for the simulation of the strong leading edge interaction although it is suitable for the simulation of the boundary layer on the body, the trailing edge interaction and the wake development.

(3) Comparisons of predictions by the finite analytic method and by the finite volume method are made. Although the finite analytic method gives better convergence histories and more accurate predictions, the differences are found to be quantitatively very small. Both methods well predict the important features of flow field. It is observed that the finite volume method always slightly overpredicts the velocity components on the boundary layer as well as in the wake region.

(4) It is found that use of one velocity staggered grid method with finite volume method gives a limit in convergence. The introduction of two velocities staggered method removes this deficiency. However, the two velocity staggered grid method requires more storage, more computing time and complicated programming.

(5) The defect of the  $k - \epsilon$  turbulence model with the Couette flow type wall function method for the prediction of transition of laminar flow to turbulent flow, turbulent boundary layer involving pressure gradient and streamline curvature effects is addressed. A strange transition near the leading edge was predicted and the velocity components are overpredicted in the adverse pressure gradient boundary layer region. As compared with the Couette flow type wall function method, the two-point wall function method well predicts the velocity components in the adverse pressure gradient boundary layer region by use of the general law of the wall in which the pressure gradient effect on the flow in the wall region are taken into consideration. However, the use of the general law of the wall is limited and can not be confidently applied to the laminar flow region near the leading edge and the favorable pressure gradient region. It is observed that the turbulent kinetic energy are overpredicted in the tail

region of axisymmetric bodies. This fact shows the the defect  $k - \epsilon$  turbulence model in the simulation of flow field involving pressure gradient and streamline curvature effects.

(6) Improvements of predictions by the introduction of two-layer turbulence model are observed. The predictions by the two-layer model not only provide the numerical solutions of viscous-affected near wall region without excessive computational efforts, but also the two-layer model can be applied to separated flow region where the wall function method can not be confidently applied. The use of two-layer model is highly recommended. However, it is found that the two-layer model slightly overpredicts the velocity components in the tail region of axisymmetric bodies.

Although present calculation methods are successfully applied to the simulation of various flow situations, there exist much rooms for further investigations which may be summarized as follows;

(1) The one velocity staggered grid method used in chapter VI has limitations in the general applications. The two velocities staggered grid method employed in the chapter VII removes the difficulties encountered in the use of one velocity staggered method. However, this

method requires a complicated programming, more storages and a more computing time. An introduction of newly developed calculation methods like algebraic manipulation method by Karki and Patankar [36] or the momentum interpolation method by Majumdar [19] will be alternative choices for removing these deficiencies. The introduction of these methods are straightfoward. The use of multi-grid method will improve the convergence.

(2) The origin of overprediction of velocity components in the tail region of axisymmetric bodies in the two-layer calculations should be clearly investigated in order to confidently apply this turbulence model to general flow situations.

(3) The introduction of more advanced turbulence model, or at least more elaborated wall functions is needed for the proper prediction of complete turbulent flows past two dimensional or axisymmetric bodies. The use of two-layer model in this purpose may be an appropriate choice although it is questionable whether this model may properly simulate the transition of laminar flow to turbulent flow.

(4) The present calculation methods can be easily applied to three dimensional situations with minor modifications.

Table 1. Definition of Variables

$\phi$	$\Gamma_\phi$	$S_\phi$
$u$	$\mu + \mu_t - \frac{\partial p}{\partial x} + \frac{\partial}{\partial x}(\mu_{ef} \frac{\partial u}{\partial x}) + \frac{1}{r} \frac{\partial}{\partial y}(r \mu_{ef} \frac{\partial v}{\partial y}) - \frac{2\partial k}{3\partial x}$	
$v$	$\mu + \mu_t - \frac{\partial p}{\partial y} + \frac{\partial}{\partial x}(\mu_{ef} \frac{\partial u}{\partial y}) + \frac{1}{r} \frac{\partial}{\partial y}(r \mu_{ef} \frac{\partial v}{\partial y}) - 2a_c \mu_{ef} \frac{v}{r^2} - \frac{2\partial k}{3\partial y}$	
$k$	$\mu + \frac{\mu_t}{\sigma_k}$	$G - \rho \epsilon$
$\epsilon$	$\mu + \frac{\mu_t}{\sigma_\epsilon}$	$c_{\epsilon 1} G \frac{\epsilon}{k} - \rho c_{\epsilon 2} \frac{\epsilon^2}{k}$

$$G = \mu_{ef} \left[ 2 \left( \frac{\partial u}{\partial x} \right)^2 + 2 \left( \frac{\partial v}{\partial y} \right)^2 + 2 a_c \left( \frac{v}{r} \right)^2 + \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 \right]$$

$$\mu_{ef} = \mu + \mu_t, \quad \mu_t = c_\mu \rho \frac{k^2}{\epsilon}$$

$$c_\mu = 0.09, \quad \sigma_k = 1.0, \quad \sigma_\epsilon = 1.3, \quad c_{\epsilon 1} = 1.44, \quad c_{\epsilon 2} = 1.92.$$

$a_c = 0, r = 1$  for Cartesian coordinate system.

$a_c = 1, y = r$  for cylindrical coordinate system.

Table 2. Solution Domains for Calculations in Chapter-IV

Type of Flow	Re	Grid	$X_i^a$	$X_d^a$	$Y_u^a$
Turbulent	$2.48 \times 10^6$	57x31	-0.6	8.57	1.0

Re = Reynolds number based on the free stream velocity and body length.

$X_i$  = location of upstream boundary.

$X_d$  = location of downstream boundary.

$Y_u$  = location of upper boundary.

<sup>a</sup>The origin of coordinate system is located at the trailing edge of plate.

Table 3. Solution Domains for Calculations in Chapter-V<sup>a</sup>

Type of Flow	Re	Grid	$X_i$	$X_d$	$Y_u$
Laminar	$10^5$	135x35	-1.25	14.6	12.7
Turbulent	$2.48 \times 10^6$	120x41	-1.15	12.3	3.9

<sup>a</sup>Variables defined in Table 2 and text.



Table 4. Solution Domains for Calculations in Chapter-VI<sup>a</sup>  
( Wall Function Method )

Type of Body	Re	Grid	X <sub>i</sub>	X <sub>d</sub>	Y <sub>u</sub>
Afterbody-1	6.6x10 <sup>6</sup>	96x35	0.5	10.9	1.60
Afterbody-2	6.8x10 <sup>6</sup>	96x35	0.6	10.9	1.55
Afterbody-5	9.3x10 <sup>6</sup>	96x35	0.6	10.9	1.15

<sup>a</sup>Variables defined in Table 2 and text.

Table 5. Solution Domains for Calculations in Chapter-VI<sup>a</sup>  
( Two Layer Model )

Type of Body	Re	Grid	X <sub>i</sub>	X <sub>d</sub>	Y <sub>u</sub>
Afterbody-1	6.6x10 <sup>6</sup>	85x53	0.5	6.75	3.64
Afterbody-2	6.8x10 <sup>6</sup>	85x53	0.6	5.60	3.71
Afterbody-3	6.0x10 <sup>6</sup>	90x53	0.6	12.4	4.20
Afterbody-5	9.3x10 <sup>6</sup>	85x53	0.6	5.60	2.73

<sup>a</sup>Variables defined in Table 2 and text.

Table 6. Solution Domains for Calculations in Chapter-VII<sup>a</sup>

Type of Body	Re	Grid	X <sub>i</sub>	X <sub>d</sub>	Y <sub>u</sub>
Afterbody-1	6.6x10 <sup>6</sup>	151x36	-2.23	9.29	1.77
Afterbody-2	6.8x10 <sup>6</sup>	151x36	-2.23	9.29	1.80
Afterbody-3	6.0x10 <sup>6</sup>	151x36	-2.23	9.29	2.04
Afterbody-5	9.3x10 <sup>6</sup>	151x36	-2.23	9.29	1.33

<sup>a</sup>Variables defined in Table 2 and text.

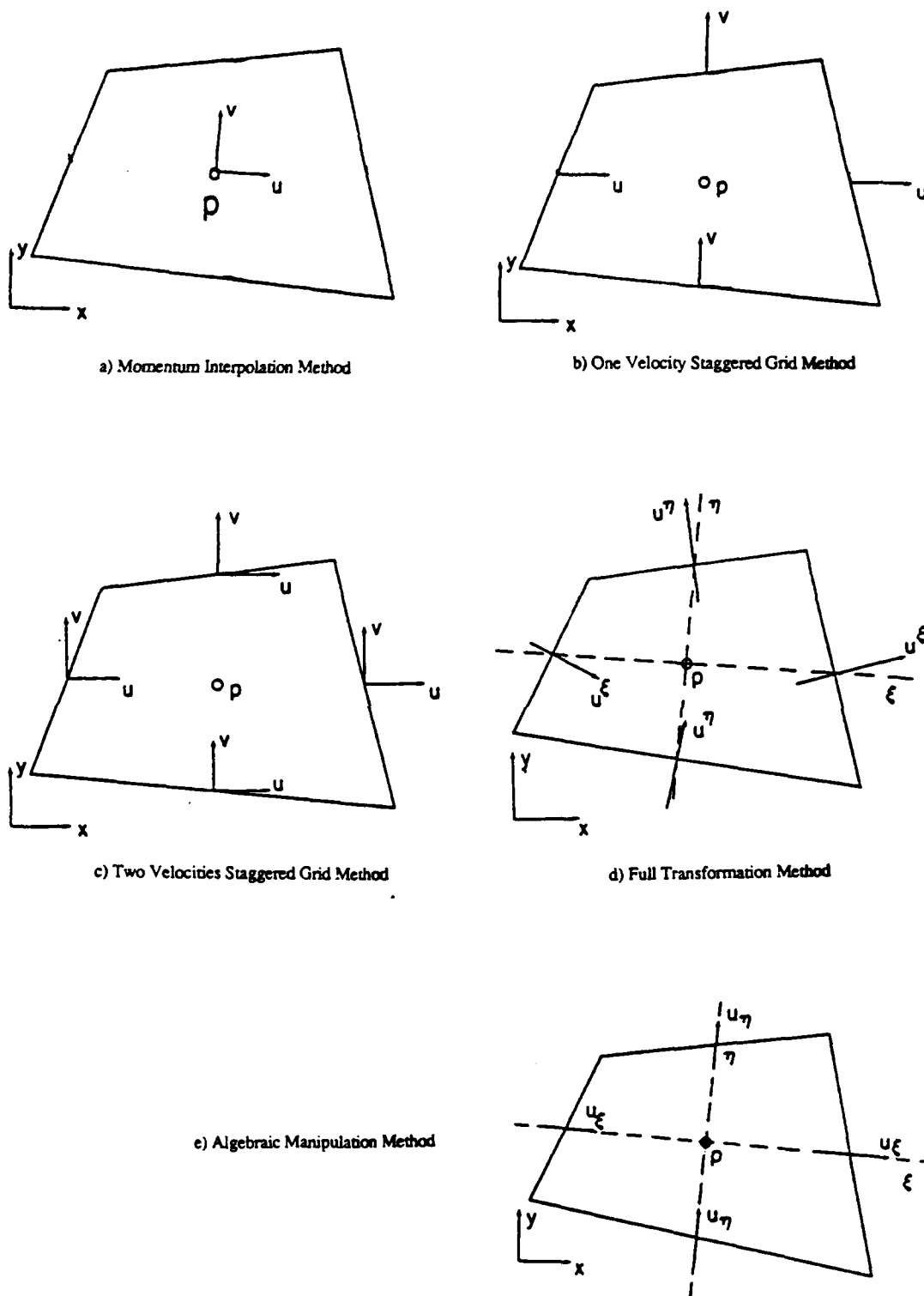
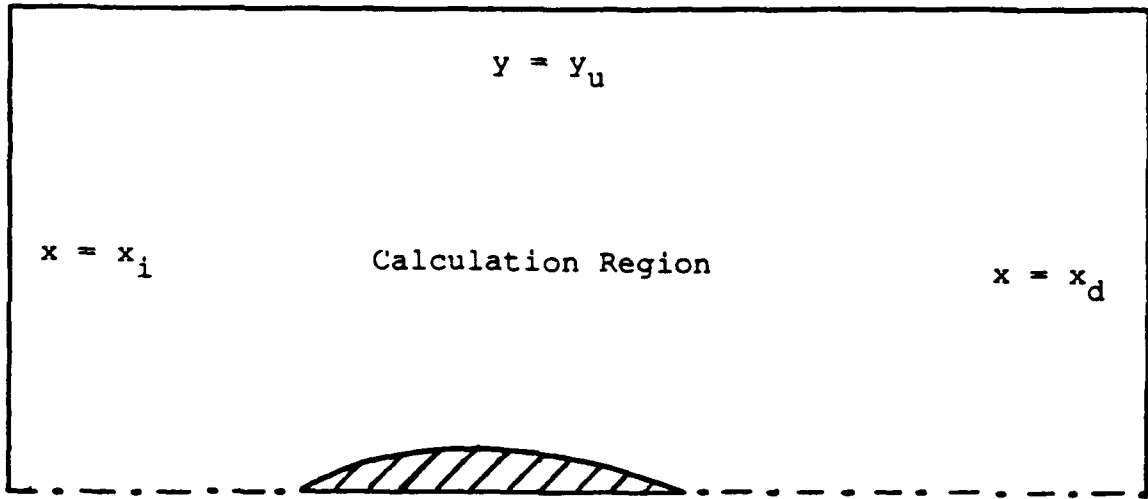
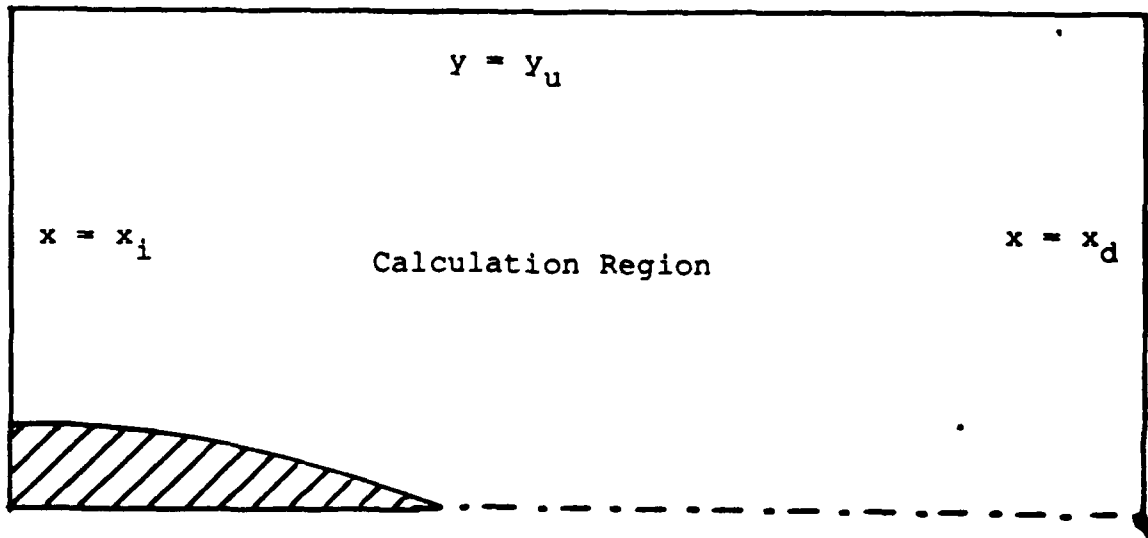


Fig.I-1. Grid Configurations.

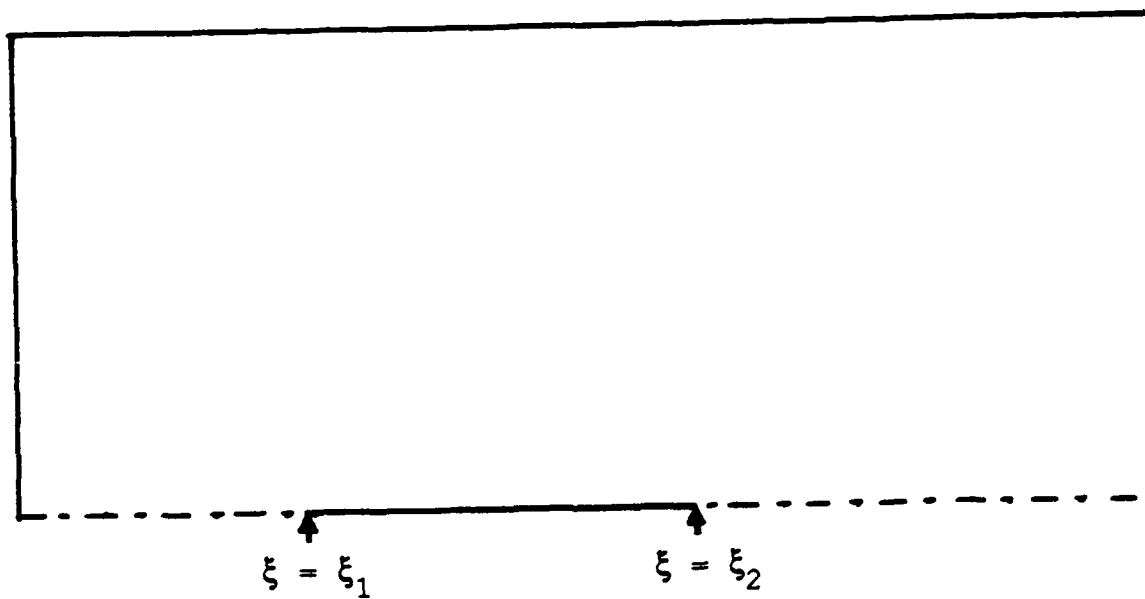


a) Full Body Calculation

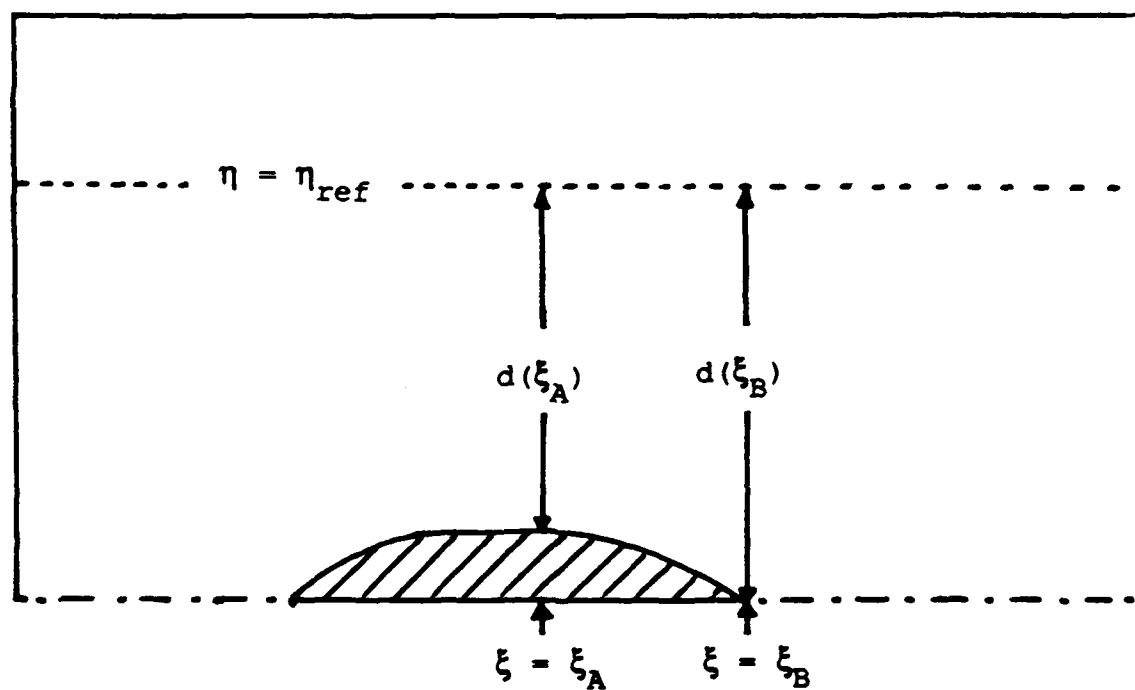


b) Half Body Calculation

Fig.II-1. Solution Domains.

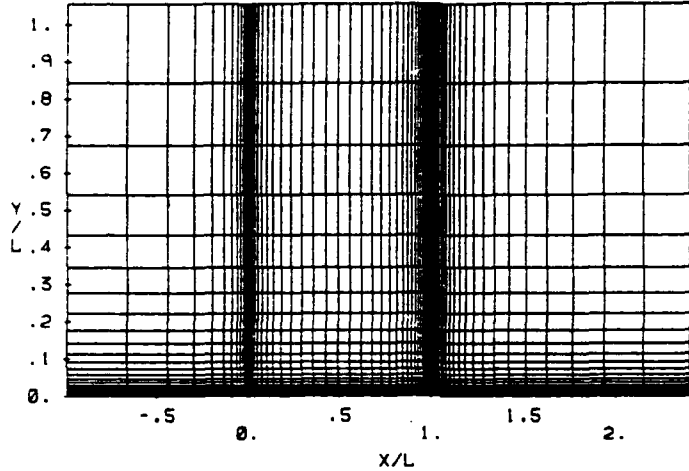


a) Flat Plate

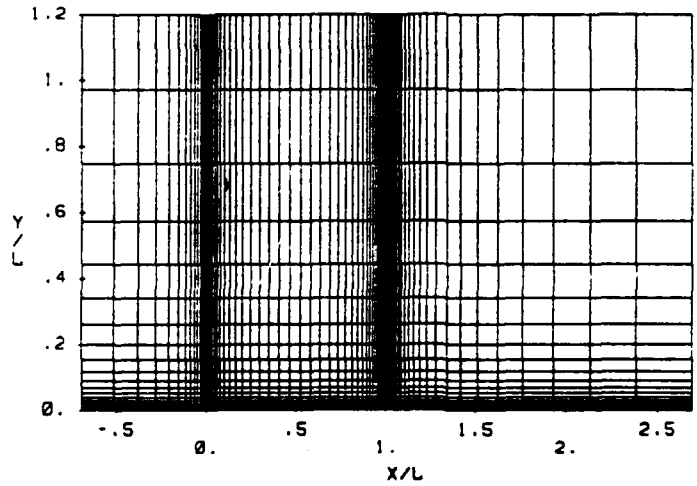


b) Axisymmetric Body

Fig.II-2. Notations for Grid Generation.

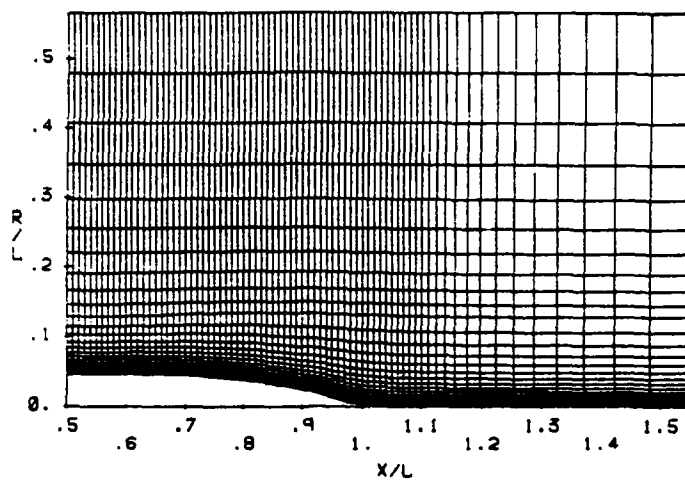


a) Laminar Flow.

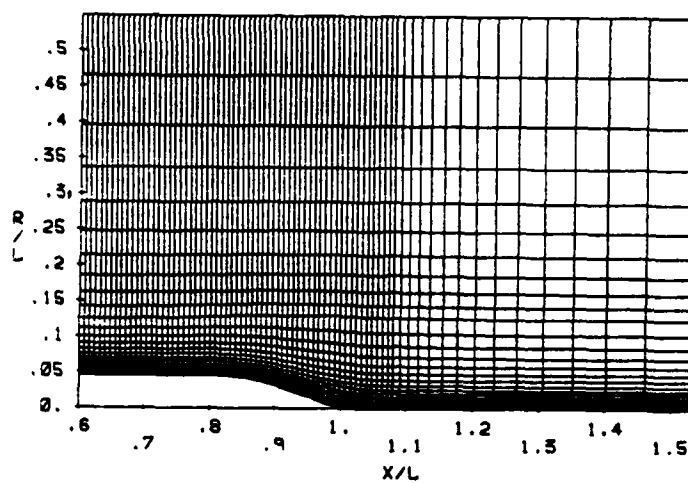


b) Turbulent Flow.

Fig.II-3. Numerical Grids for A Flat Plate.

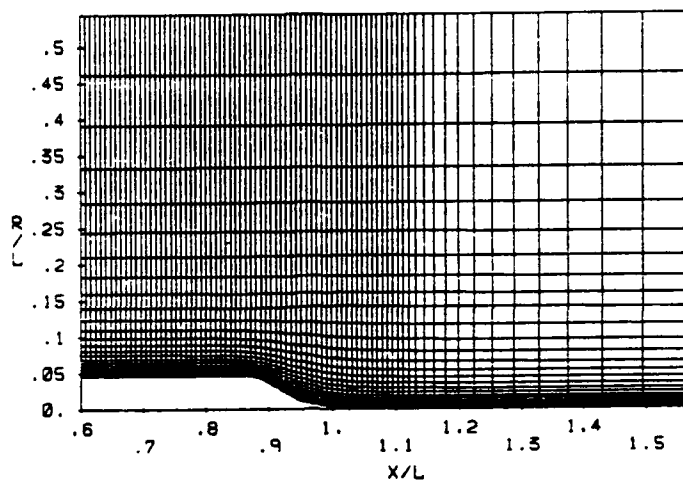


a) Afterbody-1

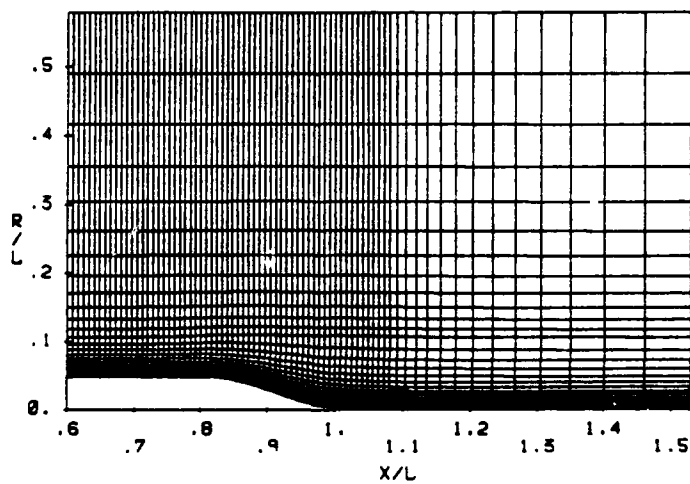


b) Afterbody-2

Fig.II-4. Numerical Grids for Axisymmetric Bodies  
( Half Body Calculation )



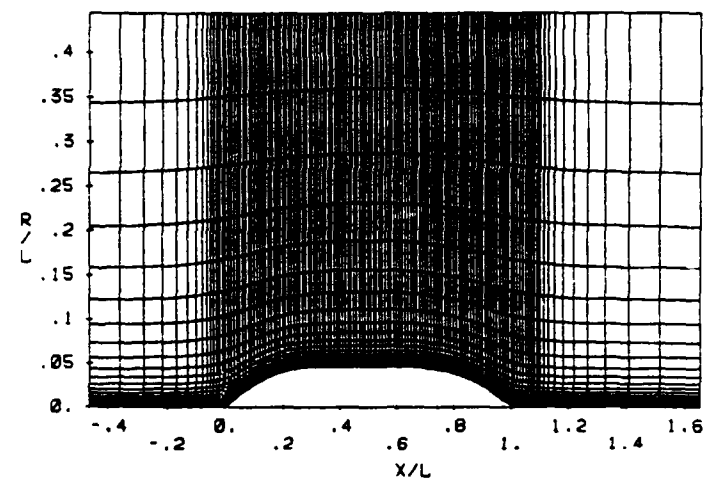
c) Afterbody-3



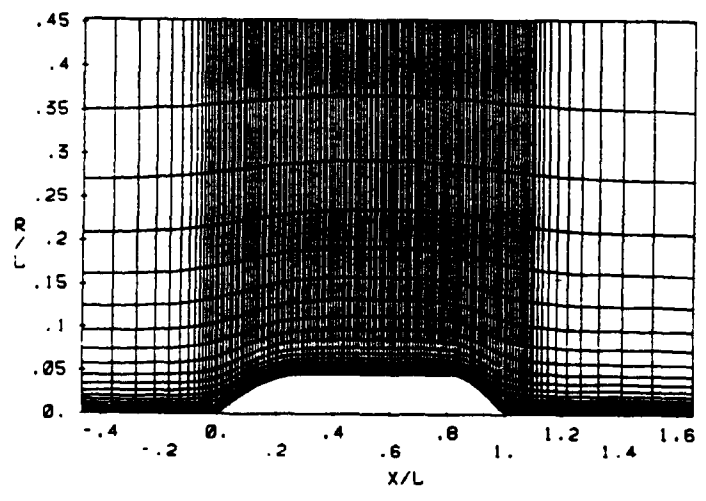
d) Afterbody-5

Fig.II-4. Numerical Grids for Axisymmetric Bodies  
( Half Body Calculation )



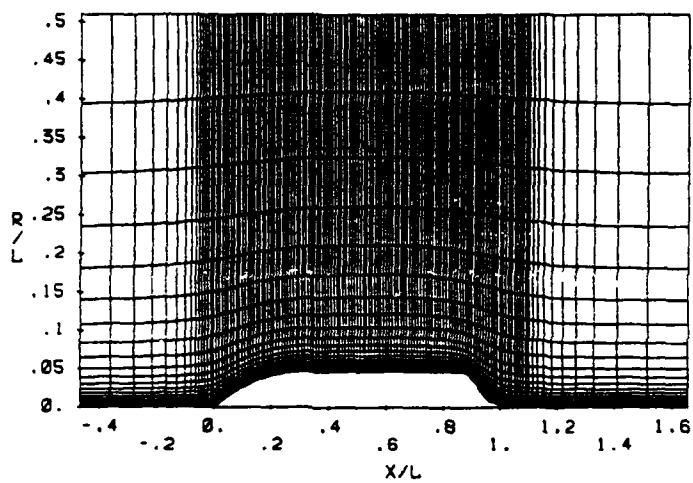


a) Afterbody-1

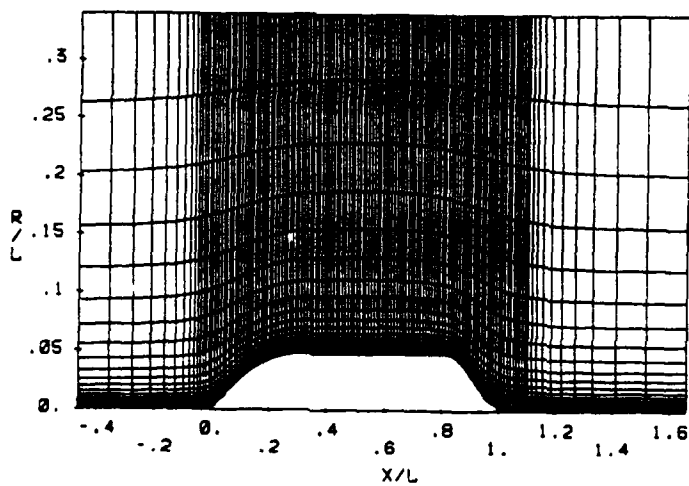


b) Afterbody-2

Fig.II-5. Numerical Grids for Axisymmetric Bodies  
( Full Body Calculation )



c) Afterbody-3



d) Afterbody-5

Fig.II-5. Numerical Grids for Axisymmetric Bodies  
( Full Body Calculation )

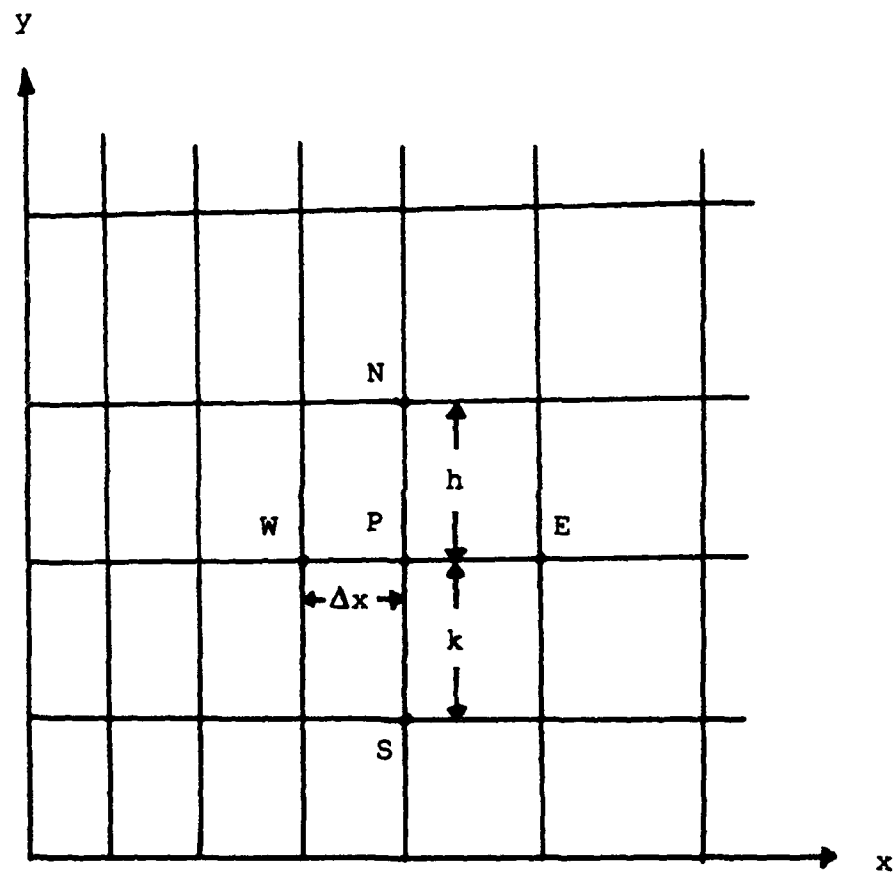


Fig.II-6. A Local Element for Finite Analytic Method in Cartesian Coordinate System

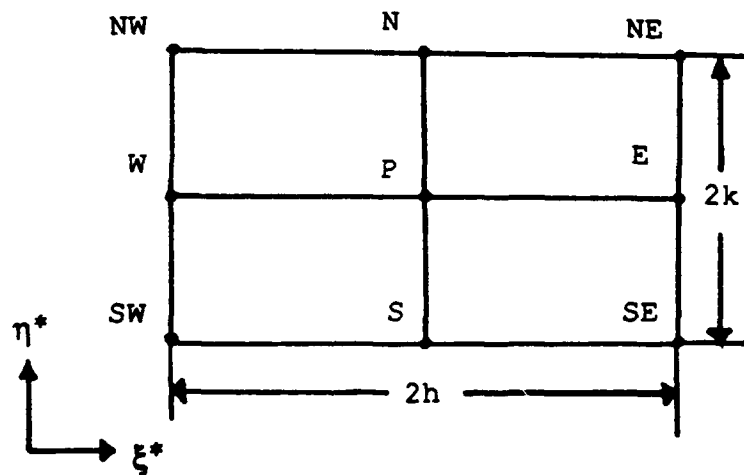
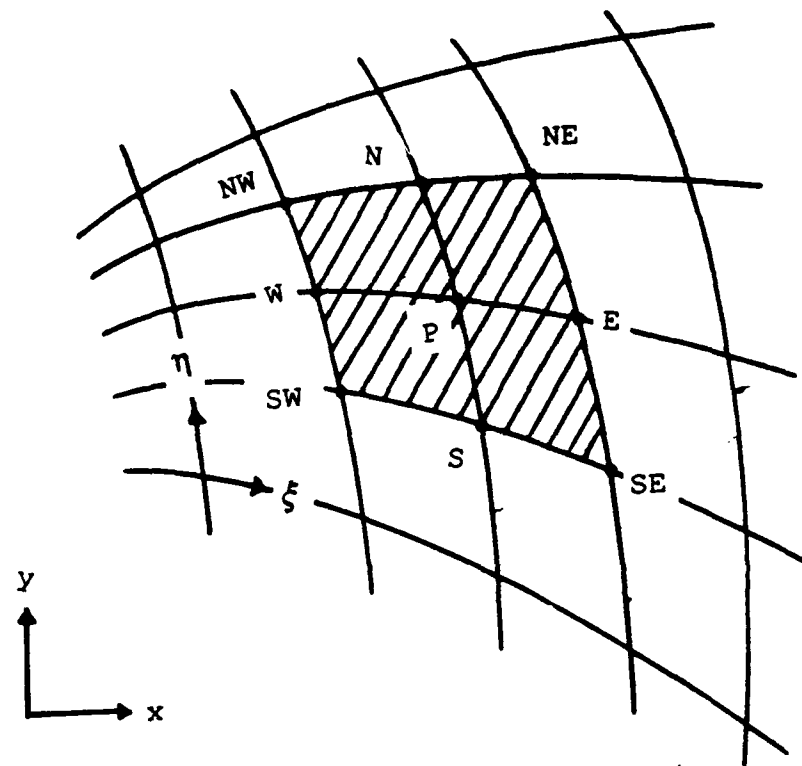


Fig.II-7. A Local Element for Finite Analytic Method in Boundary-Fitted Coordinate System

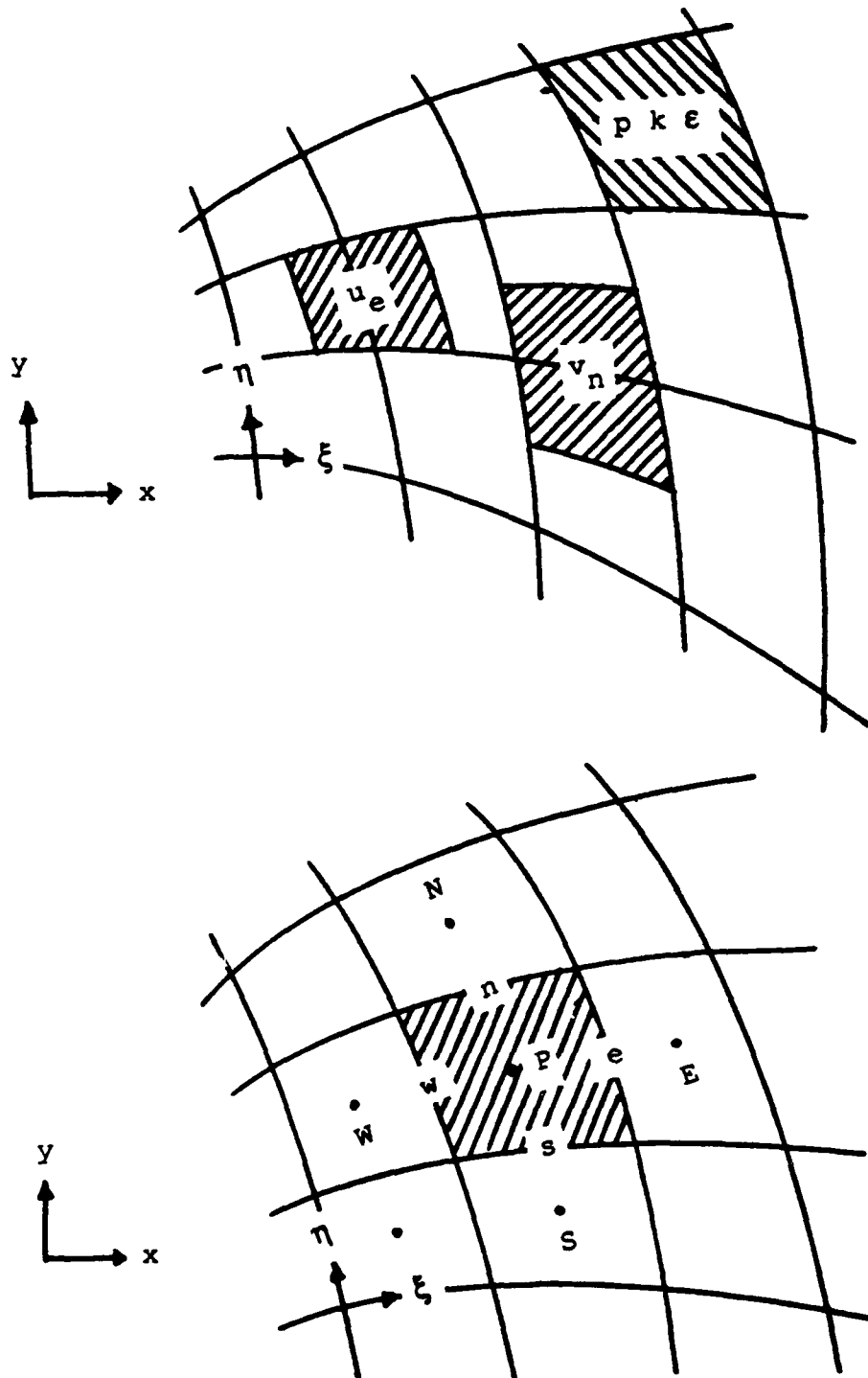


Fig.II-8. A Local Element for Finite Volume Method  
in Boundary-Fitted Coordinate System

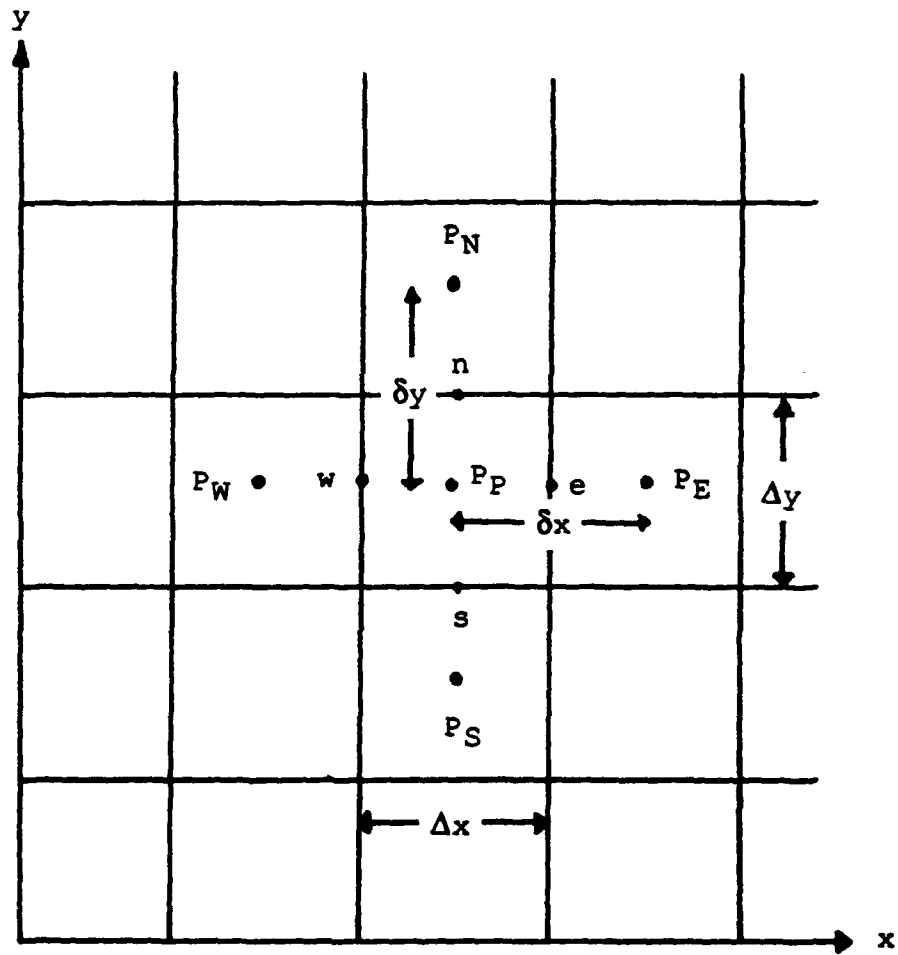


Fig.III-1. Control Volume for Pressure Variable  
in Cartesian Coordinate System

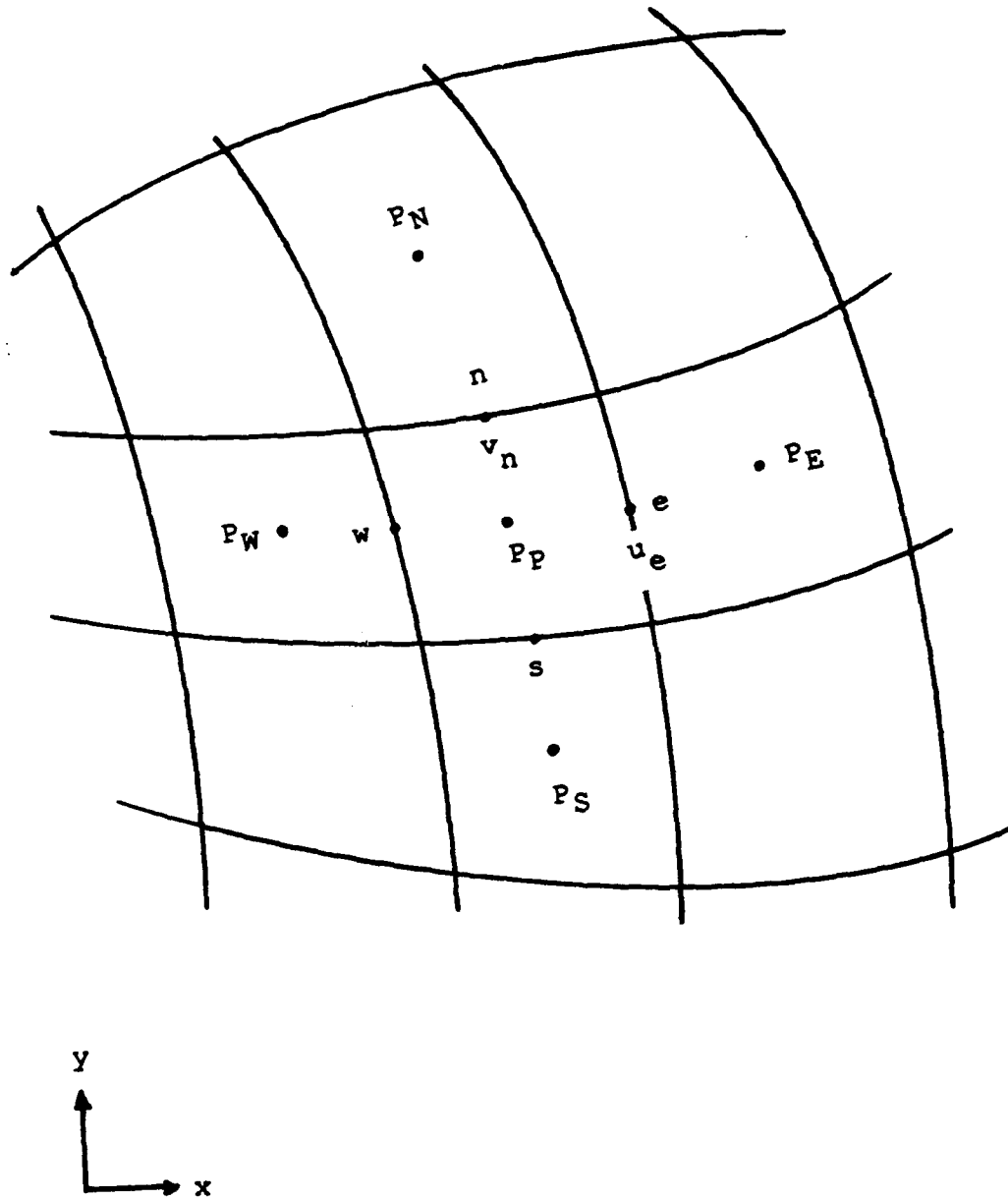


Fig.III-2. One Velocity Staggered Grid System.

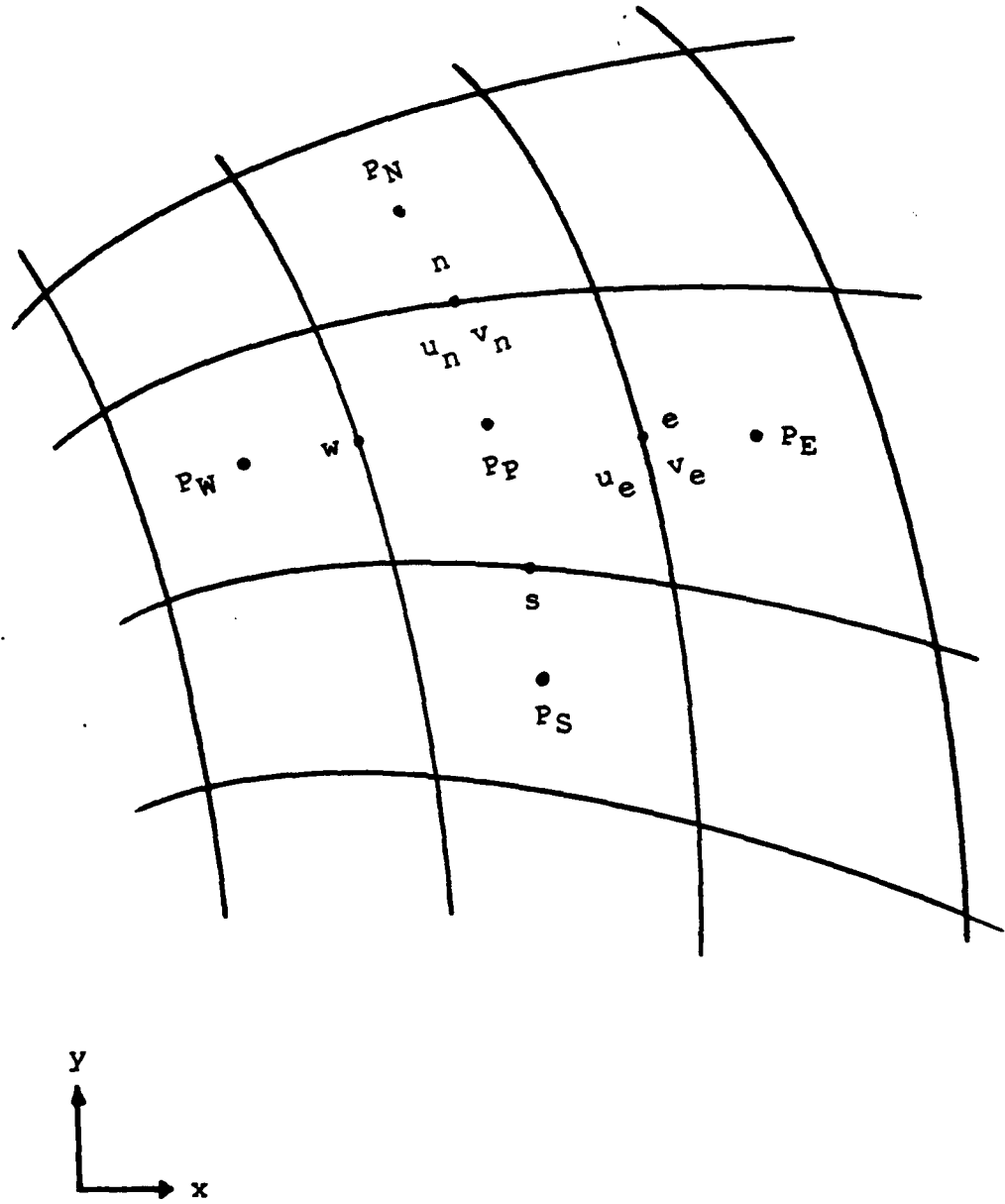


Fig.III-3. Two Velocities Staggered Grid System.



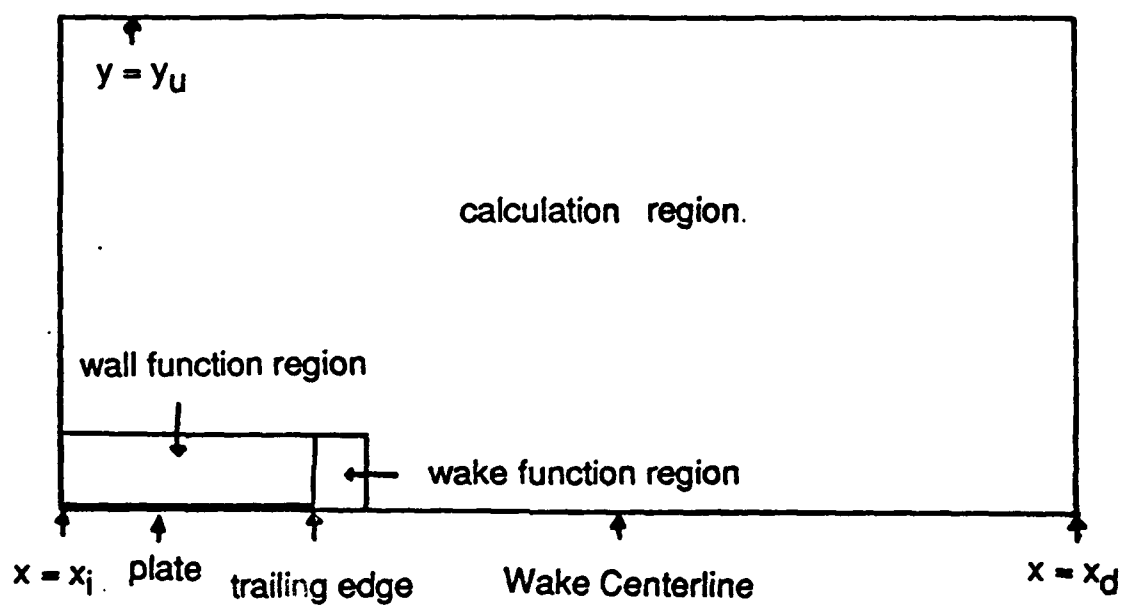


Fig.IV-1. Computational Domain.

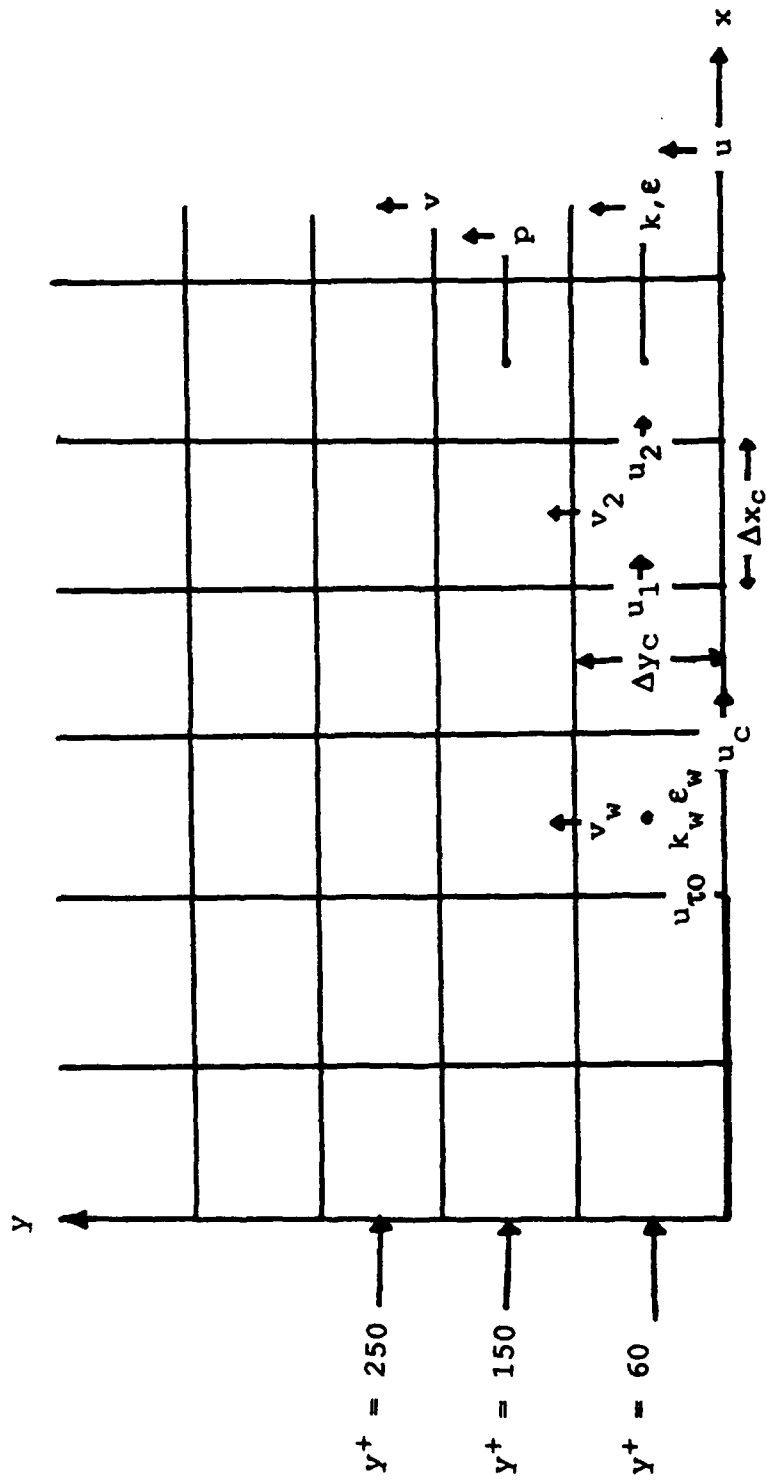


Fig. IV-2. Notation for Wake Function Method.

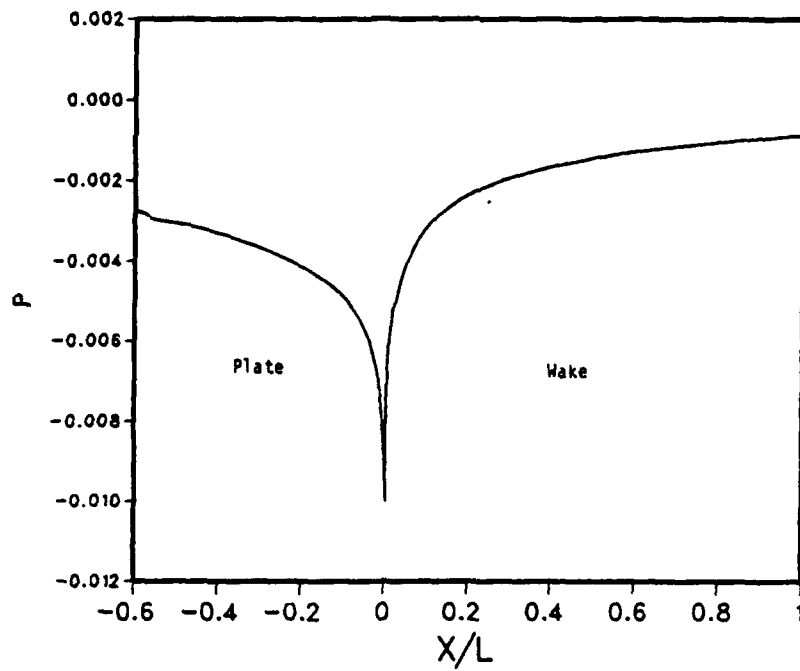


Fig.IV-3. Pressure Distribution on the Plate and along the Wake Centerline (  $y^+ \sim 150$  )

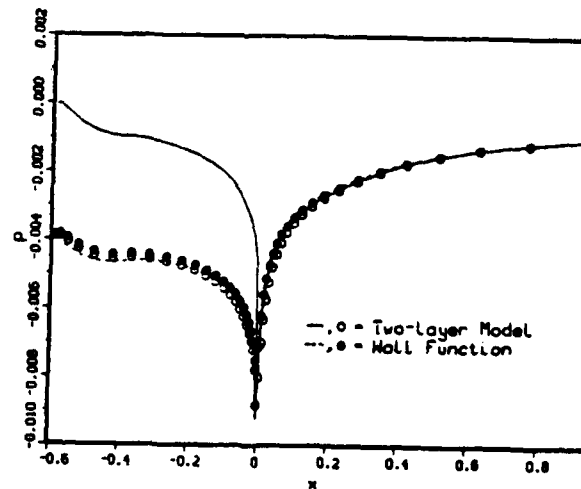


Fig.IV-4. Pressure Distribution Given in Ref. [70]

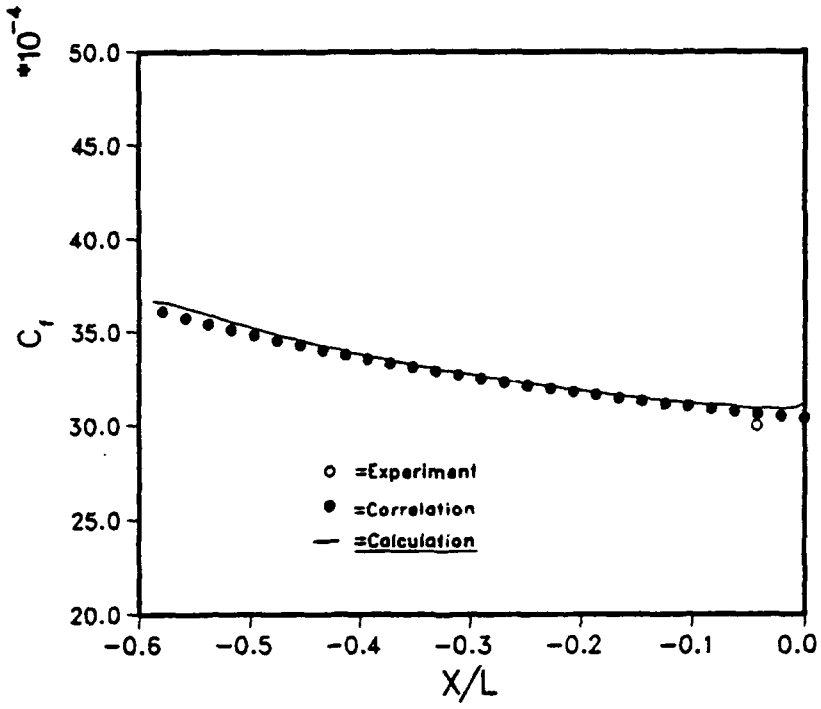


Fig.IV-5. Skin Friction Coefficient

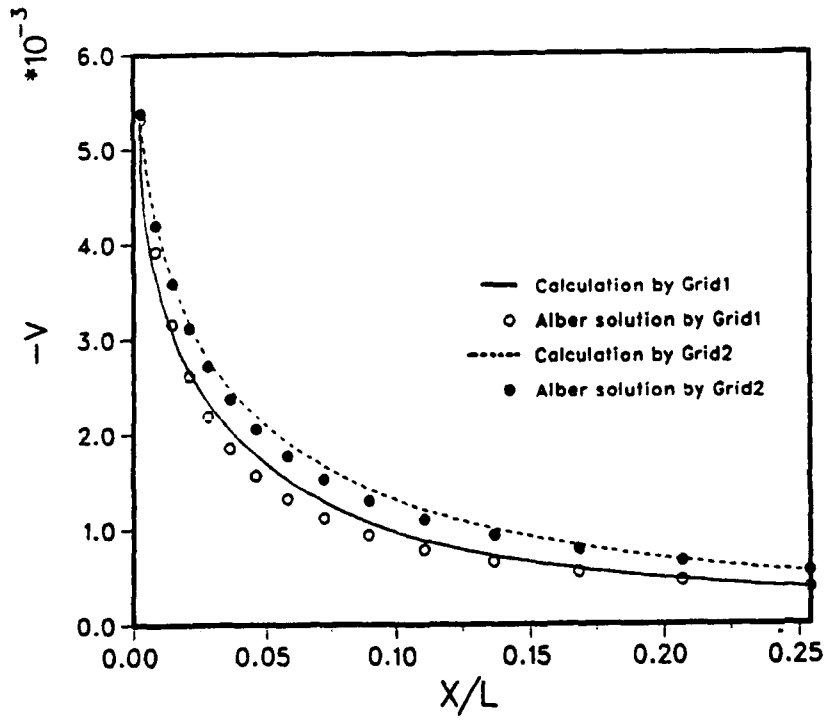


Fig.IV-11. Near-Wake Normal Velocity Distributions

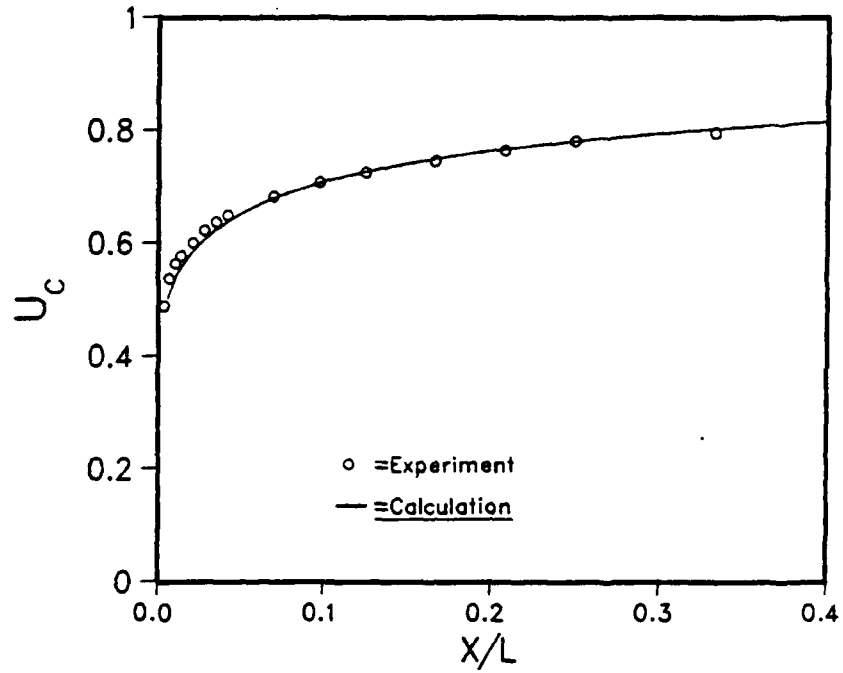


Fig. IV-6. Wake Centerline Velocity Distribution

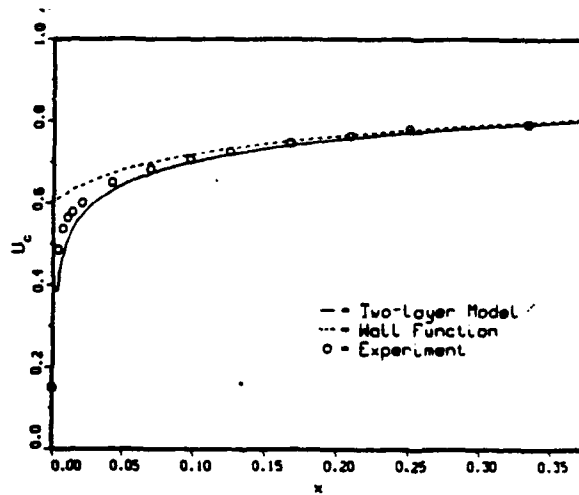


Fig. IV-7. Wake Centerline Velocity Distribution  
Given in Ref. [70]

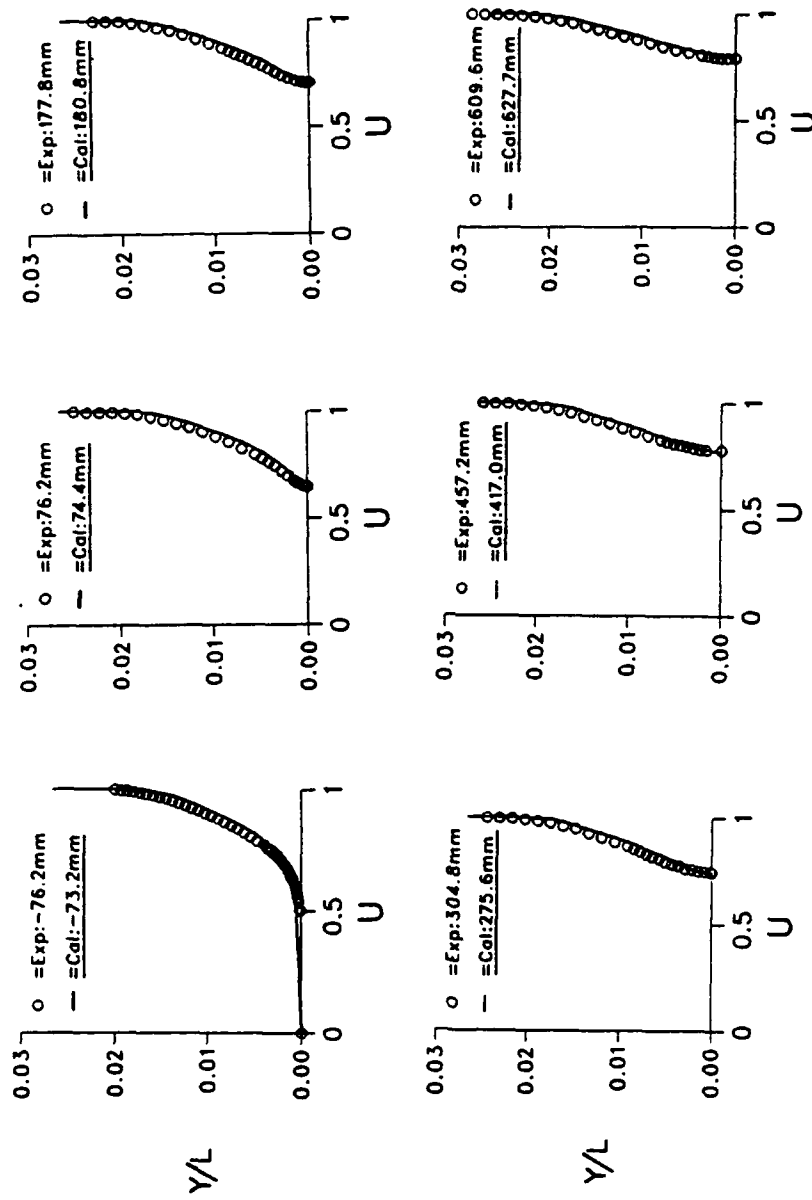


Fig.IV-8. Velocity Profiles in the Near wake

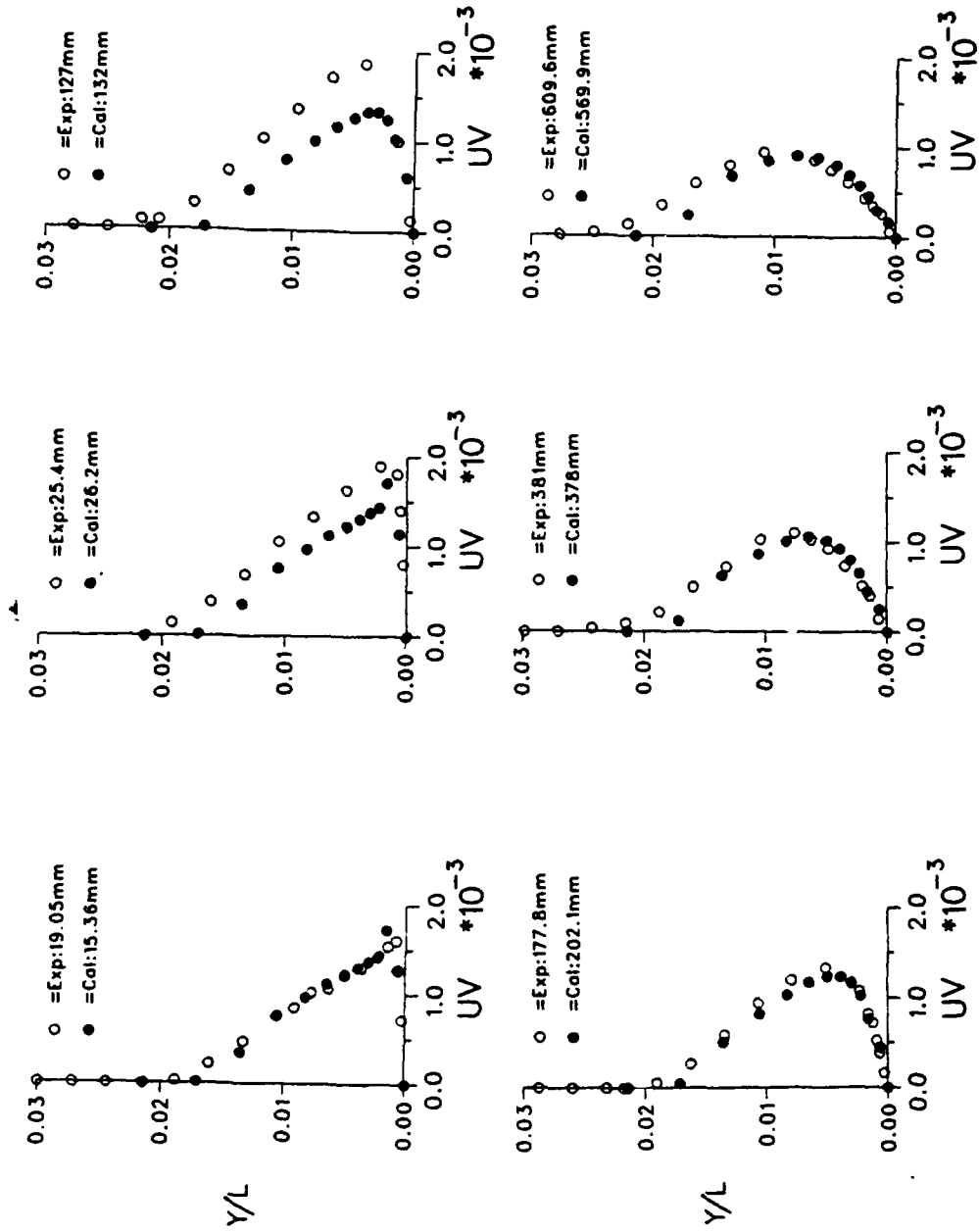


Fig. IV-9. Reynolds Shear Stress Distributions in the Near Wake

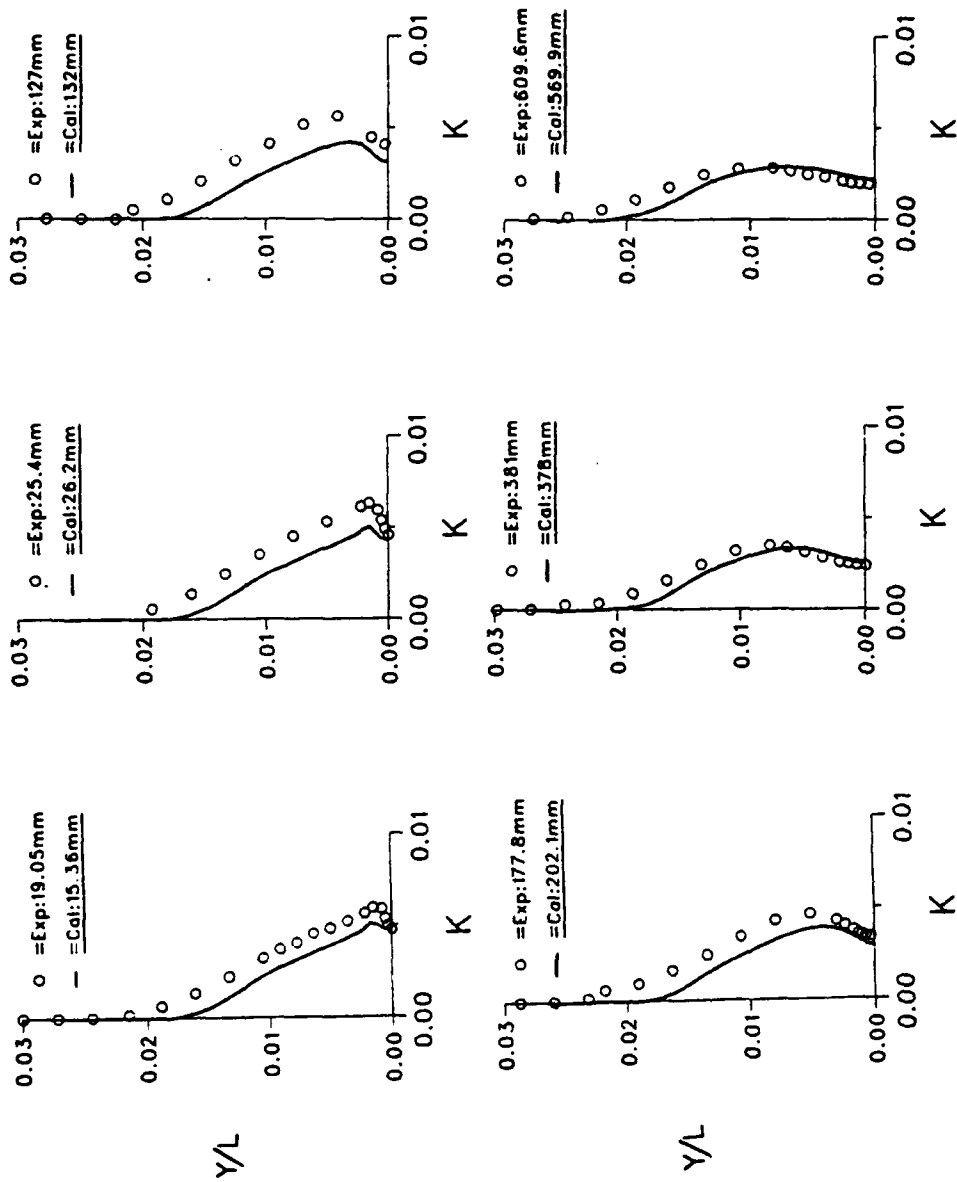


Fig. IV-10. Turbulent Kinetic Energy Distributions in the Near Wake



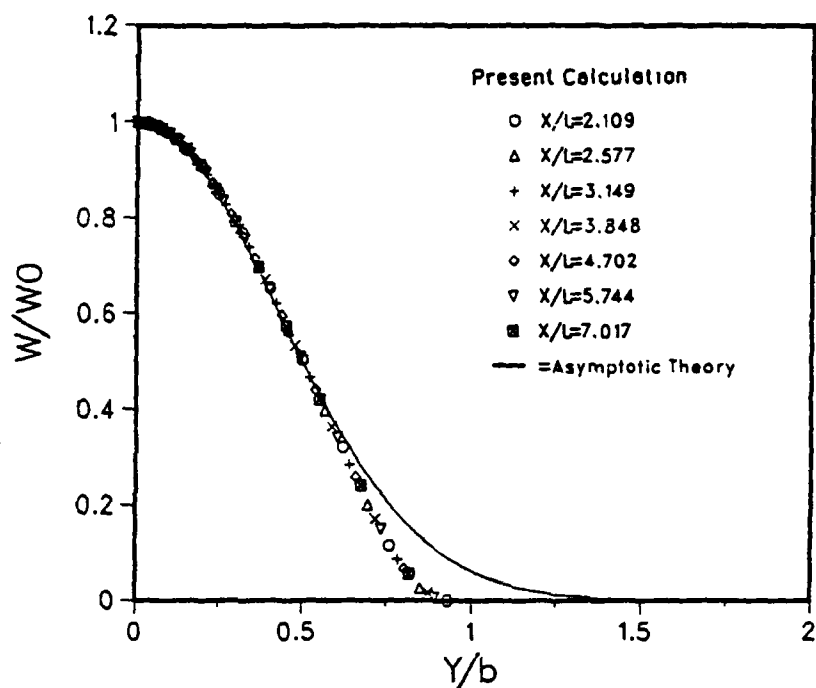


Fig.IV-12. Far-Wake Velocity Defect Profiles

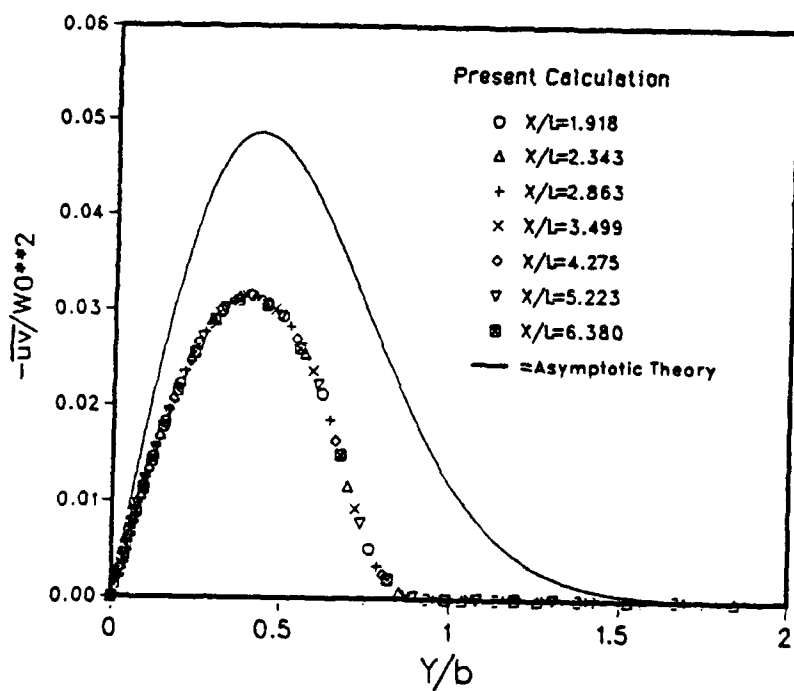


Fig.IV-13. Far-Wake Reynolds Shear Stress Profiles

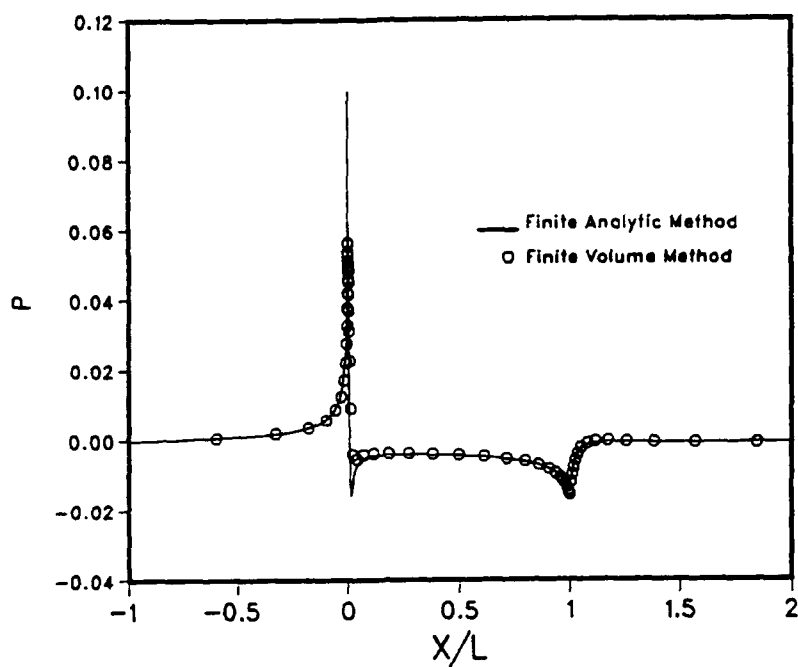


Fig.V-1. Pressure Distribution ( Laminar Flow )

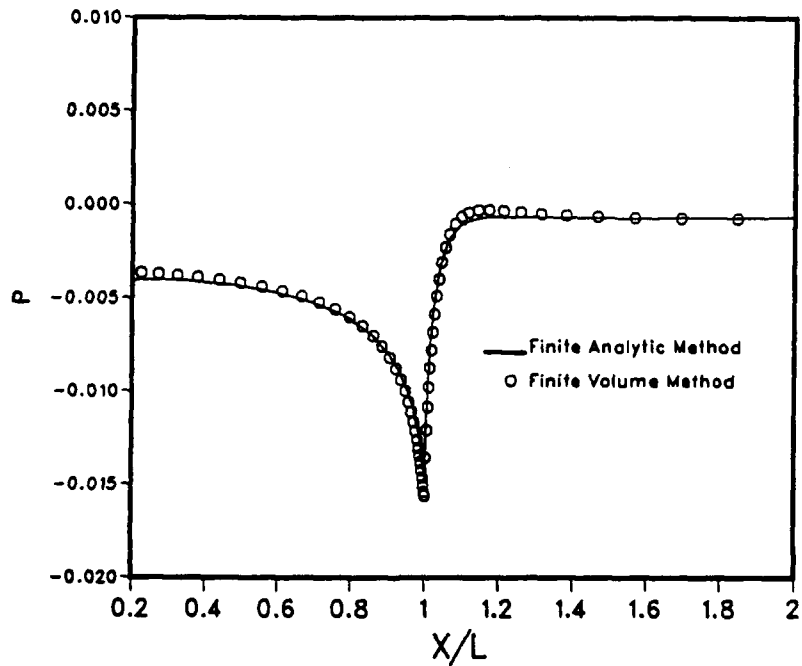


Fig.V-2. Pressure Distribution near the Trailing Edge ( Laminar Flow )

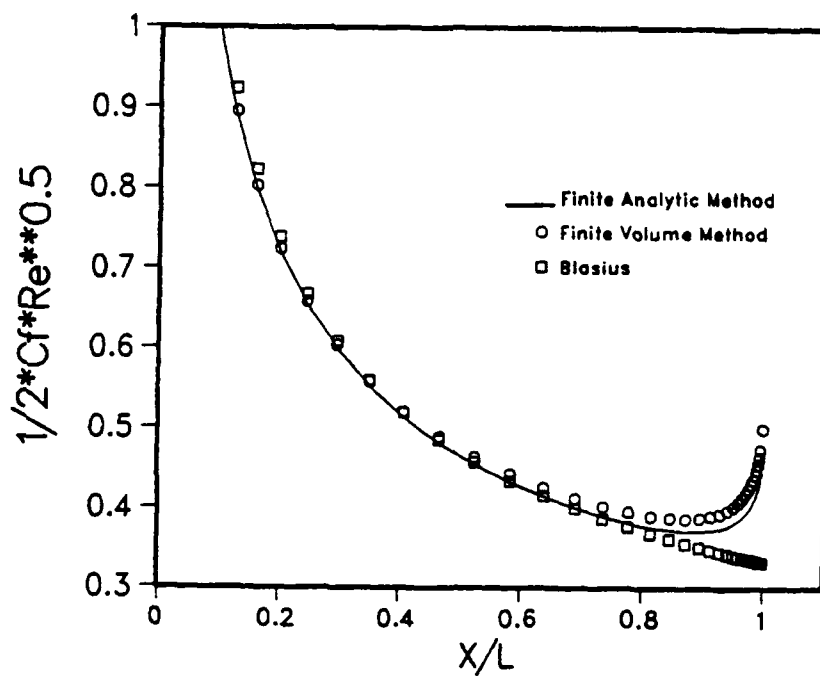


Fig.V-3. Skin Friction Coefficient  
( Laminar Flow )

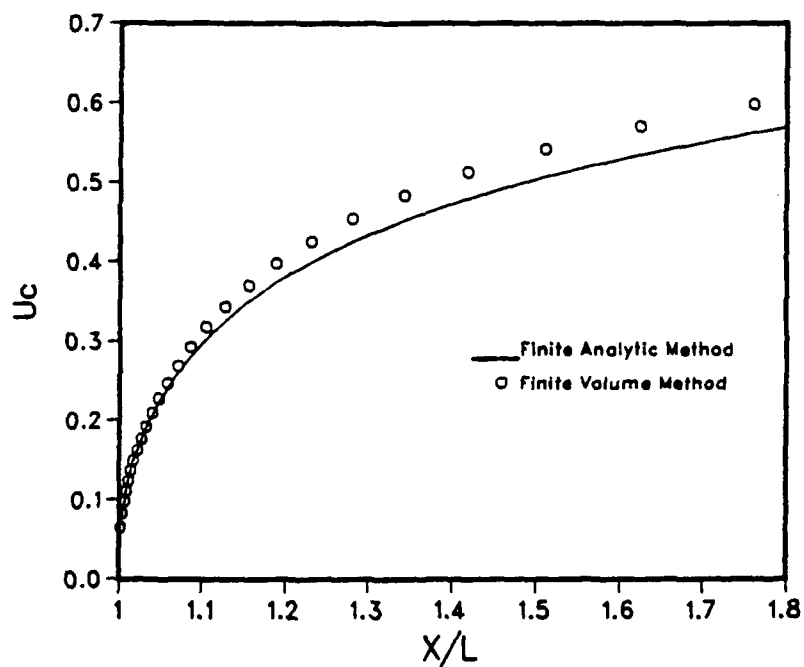


Fig.V-4. Wake Centerline Velocity Distribution  
( Laminar Flow )

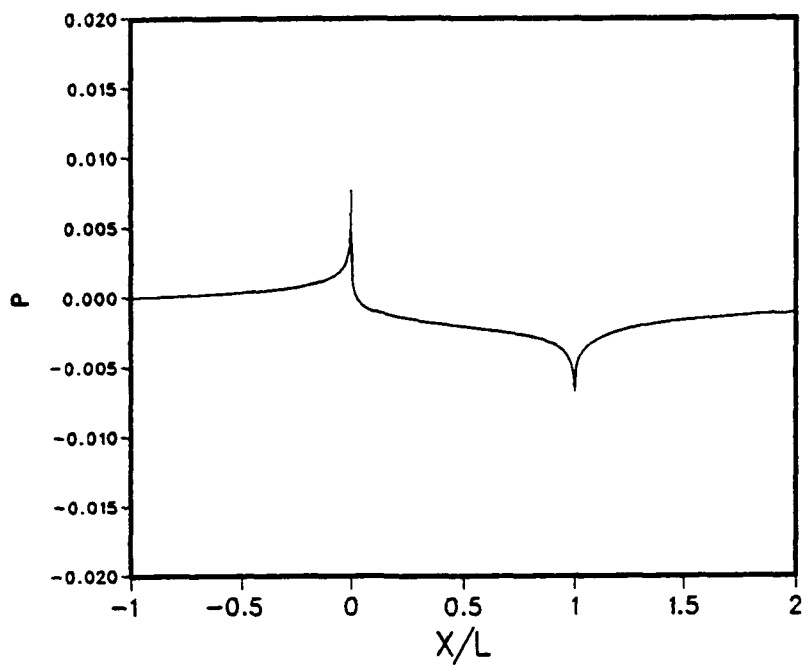


Fig.V-5. Pressure Distribution ( Turbulent Flow )

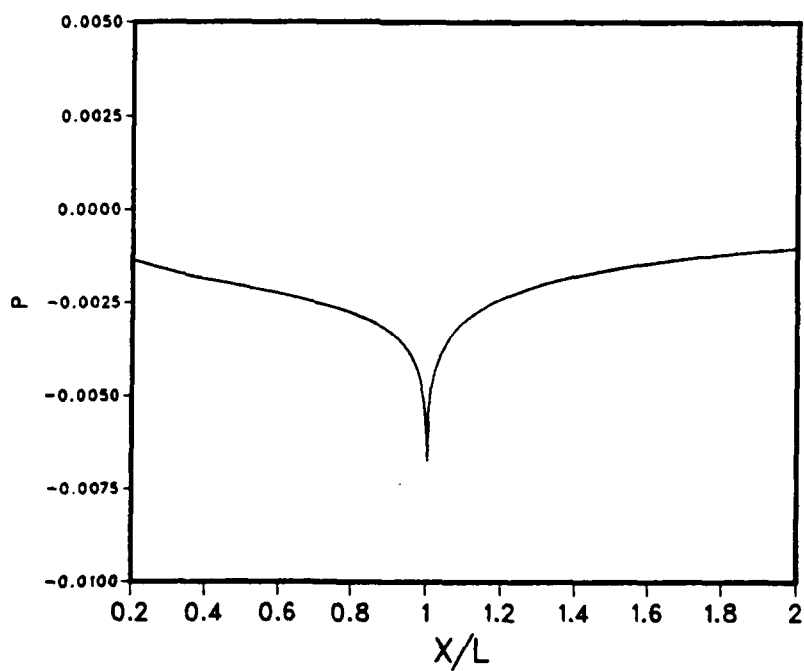


Fig.V-6. Pressure Distribution near the Trailing Edge ( Turbulent Flow )

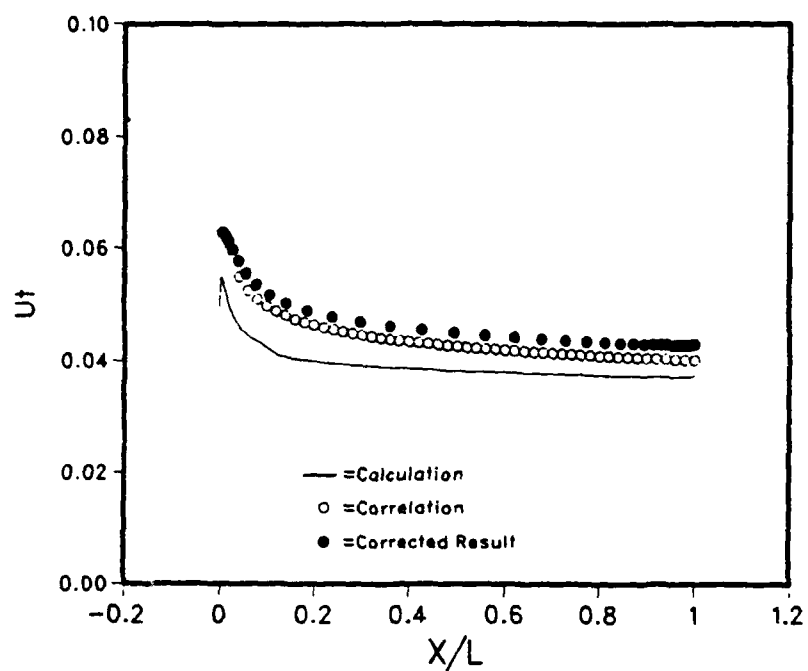


Fig.V-7. Friction Velocity Distribution  
( Turbulent Flow )

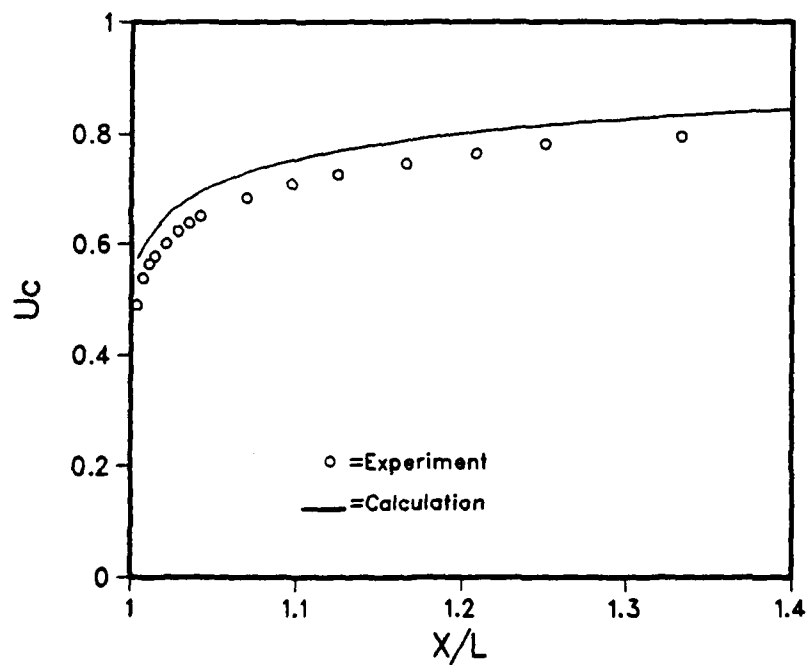


Fig.V-8. Wake Centerline Velocity Distribution  
( Turbulent Flow )

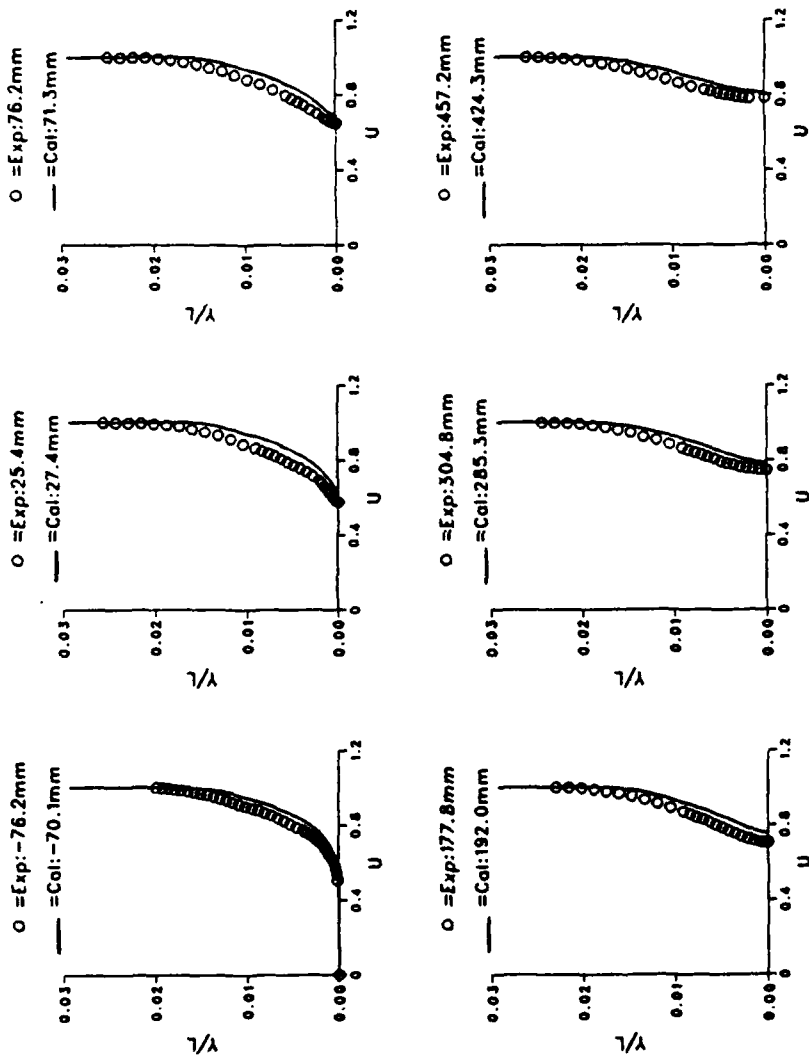


Fig.V-9. Velocity Profiles ( Turbulent Flow )

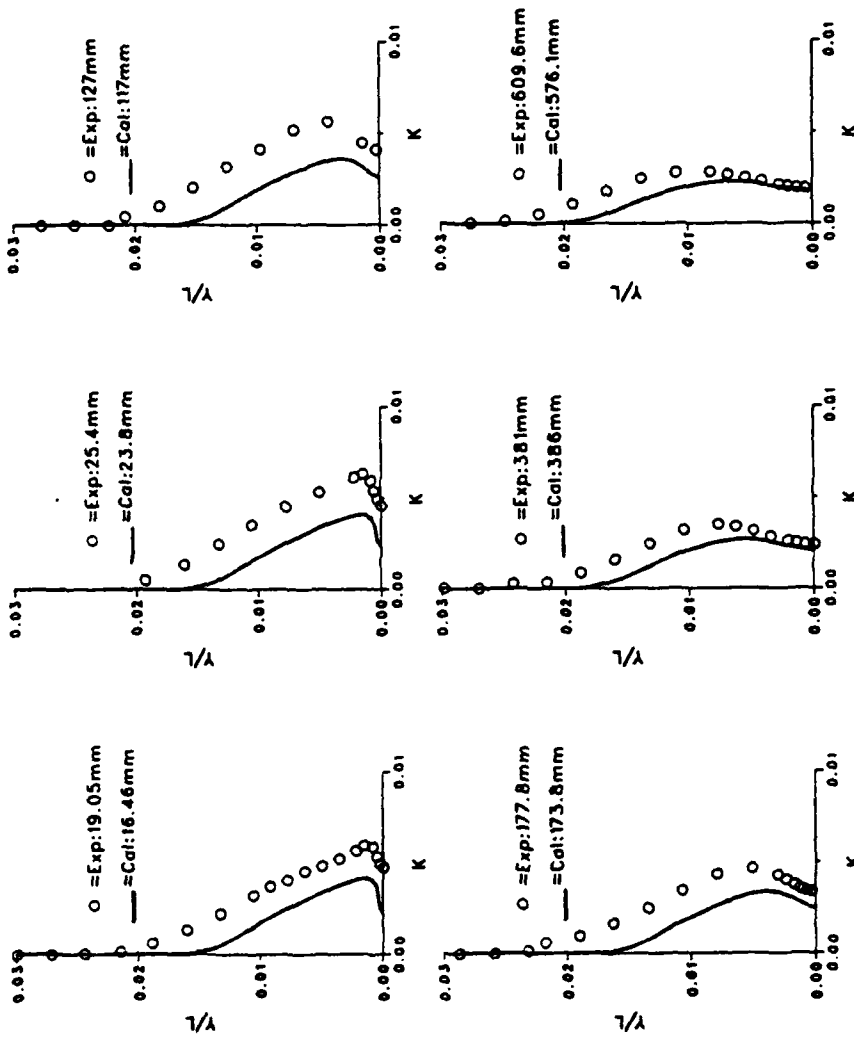


Fig.V-10. Turbulent Kinetic Energy Profiles

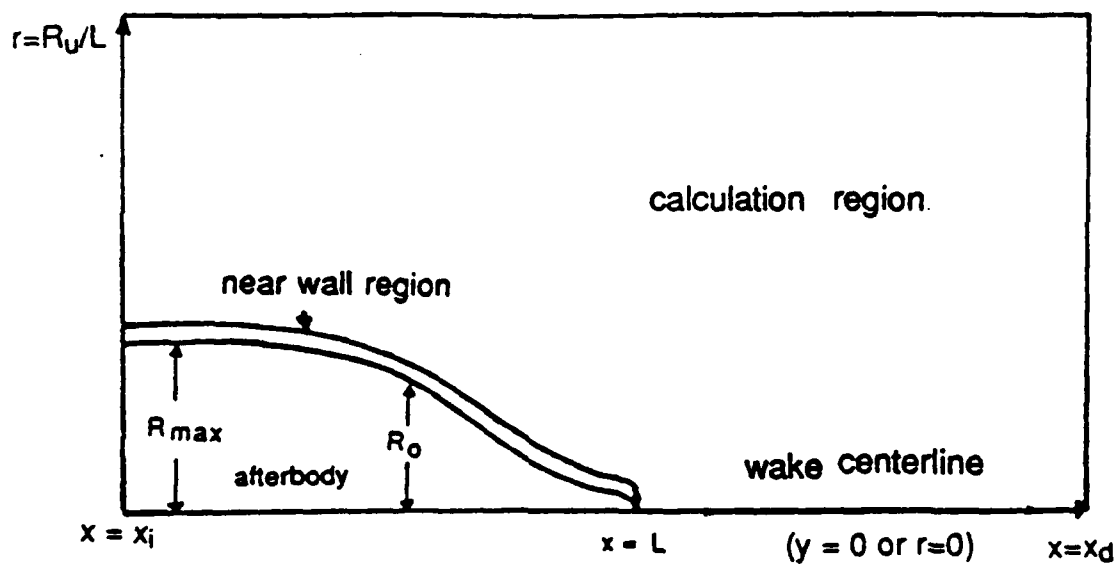


Fig.VI-1. Notations and Solution Domains



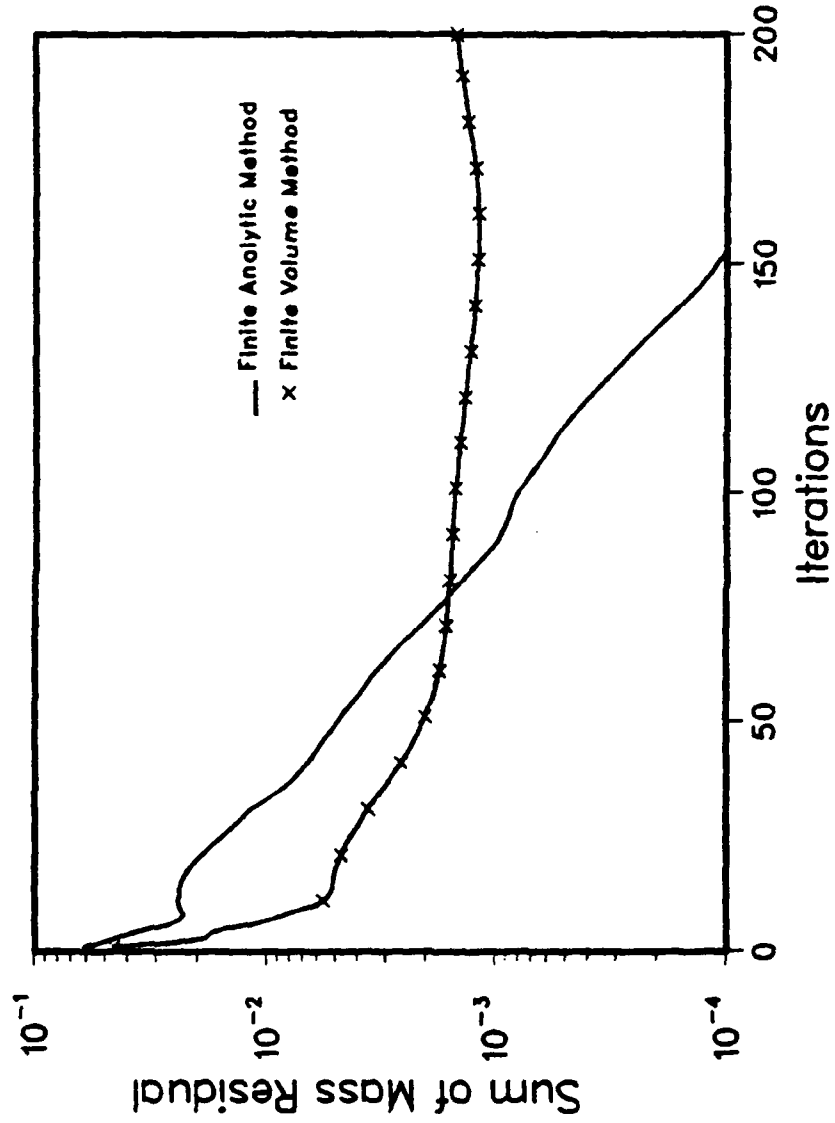


Fig. VI-2. Convergence History

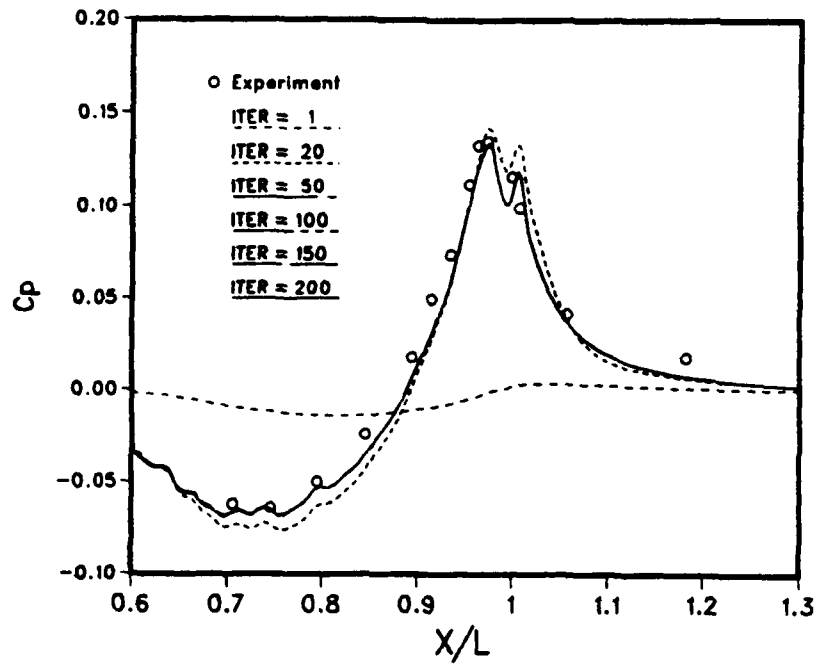


Fig. VI-3. Pressure Convergence History  
( Finite Analytic Method )

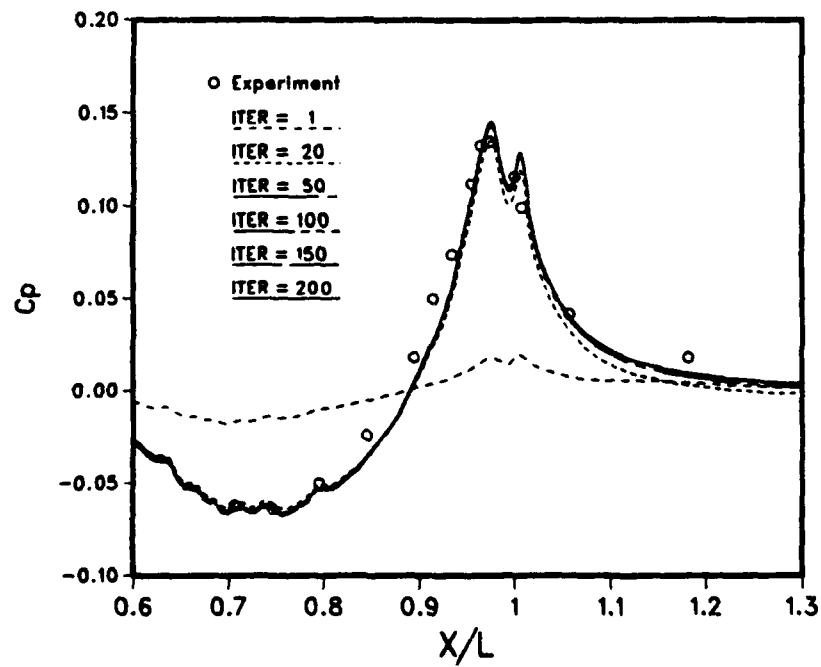


Fig. VI-4. Pressure Convergence History  
( Finite Volume Method )

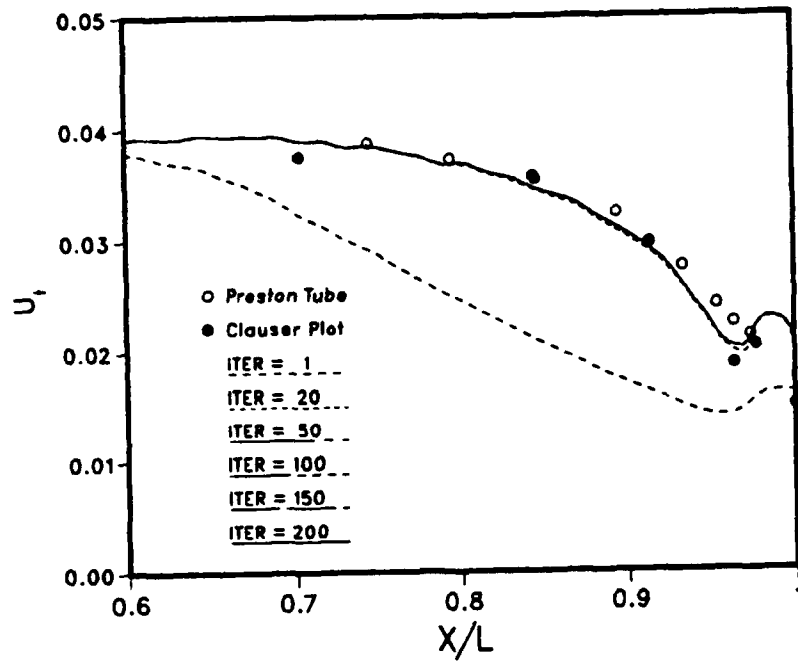


Fig.VI-5. Friction Velocity Convergence History  
( Finite Analytic Method )

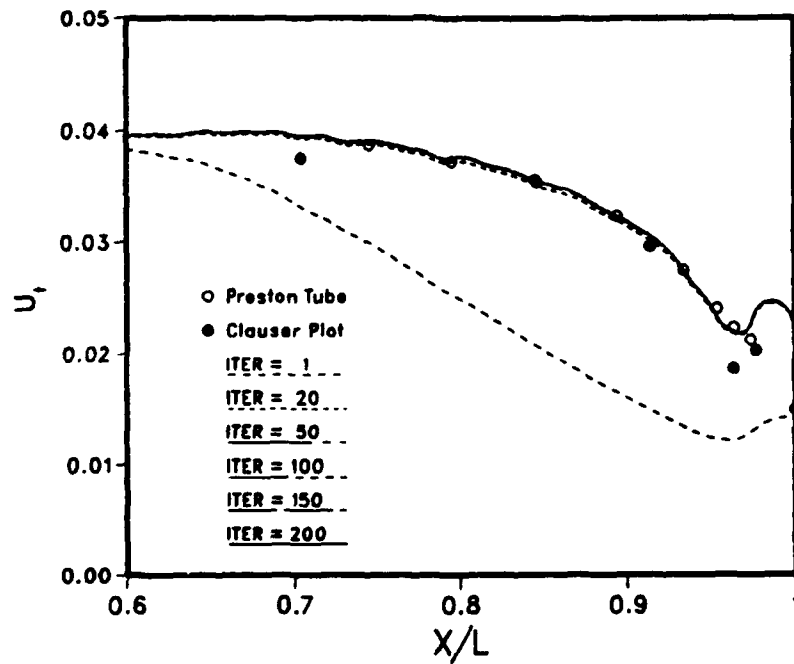


Fig.VI-6. Friction Velocity Convergence History  
( Finite Volume Method )

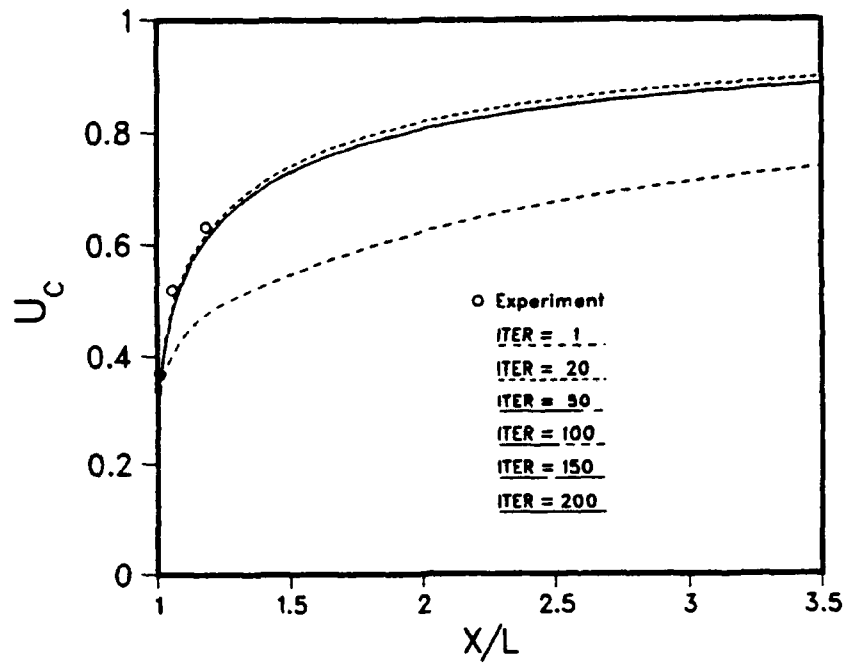


Fig.VI-7. Wake Centerline Velocity Convergence History ( Finite Analytic Method )

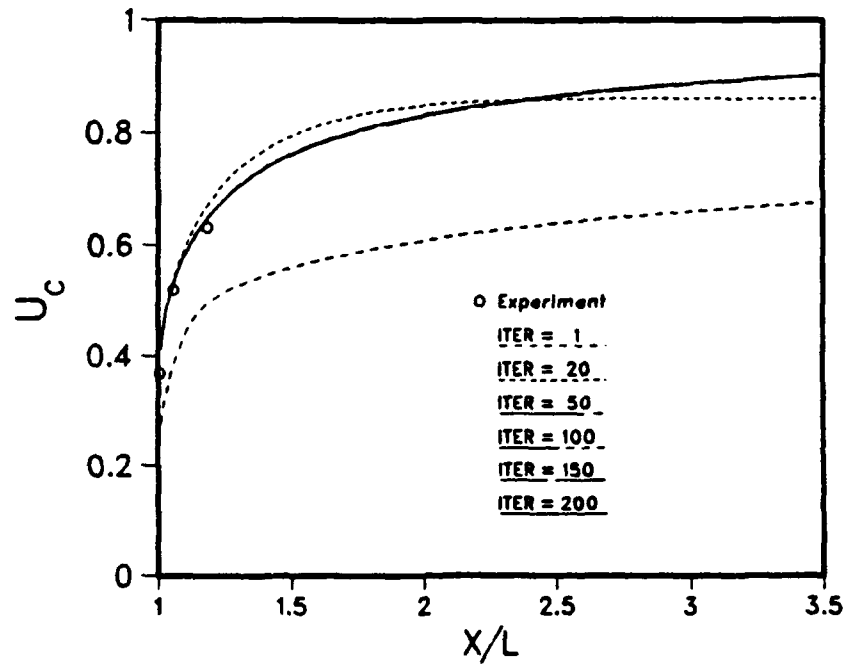


Fig.VI-8. Wake Centerline Velocity Convergence History ( Finite Volume Method )

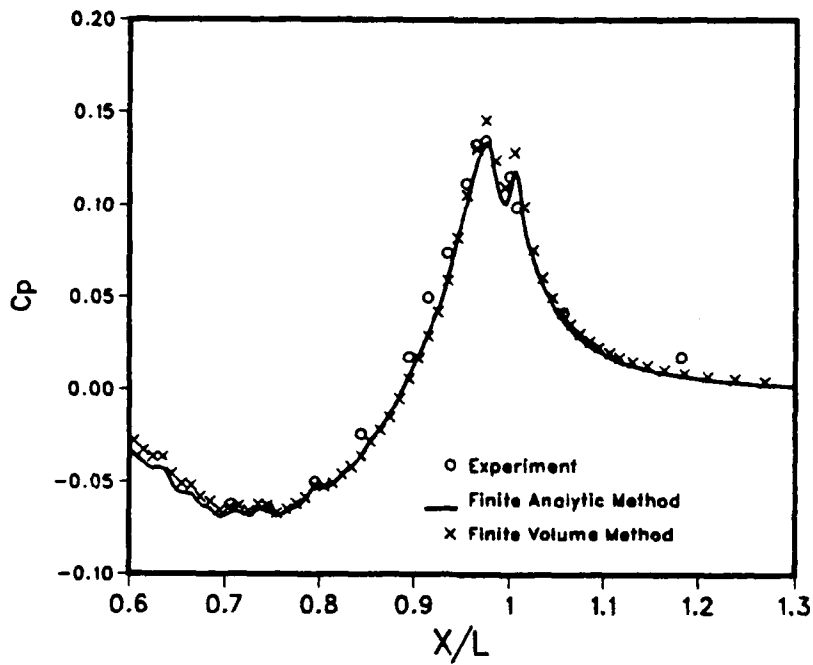


Fig.VI-9. Surface Pressure Distribution  
( Afterbody-1 )

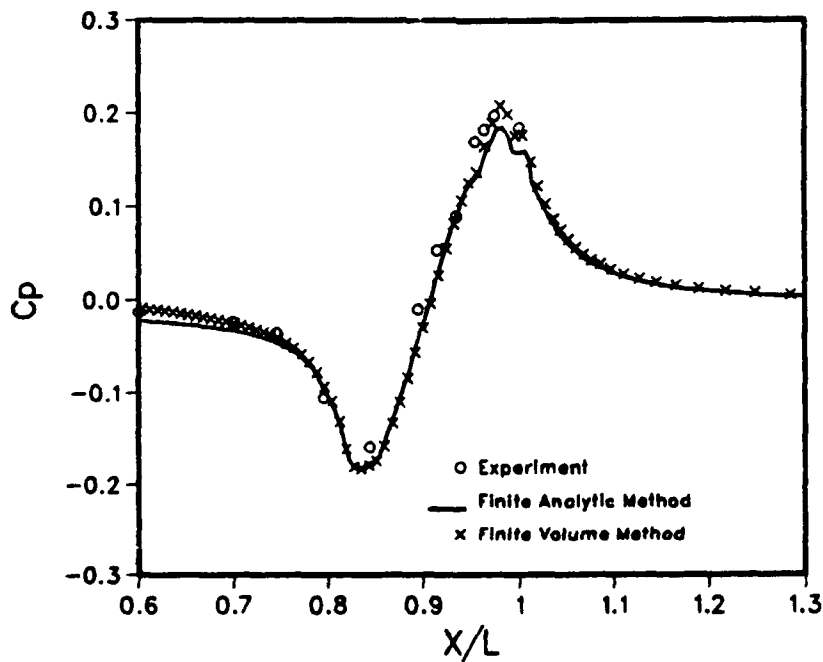


Fig.VI-10. Surface Pressure Distribution  
( Afterbody-2 )

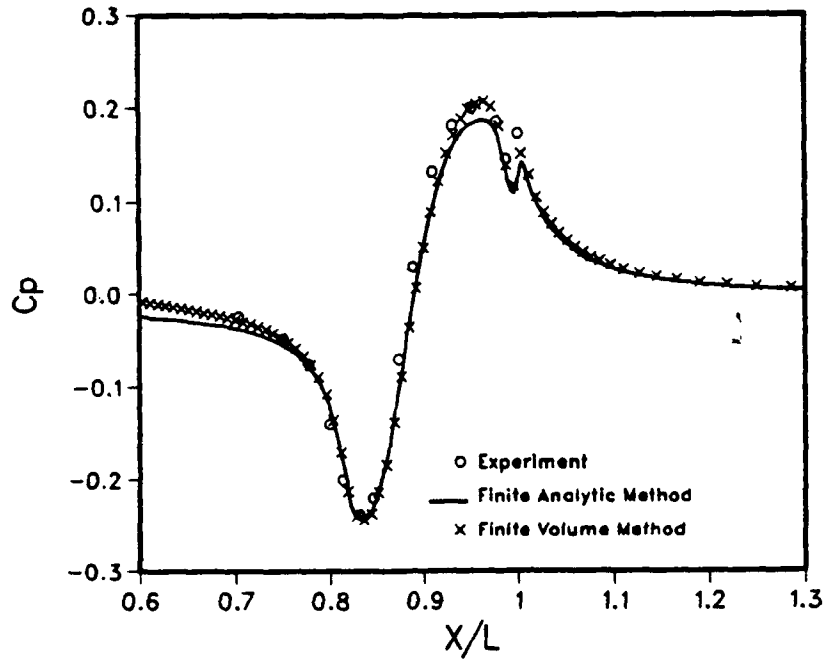


Fig.VI-11. Surface Pressure Distribution  
( Afterbody-5 )

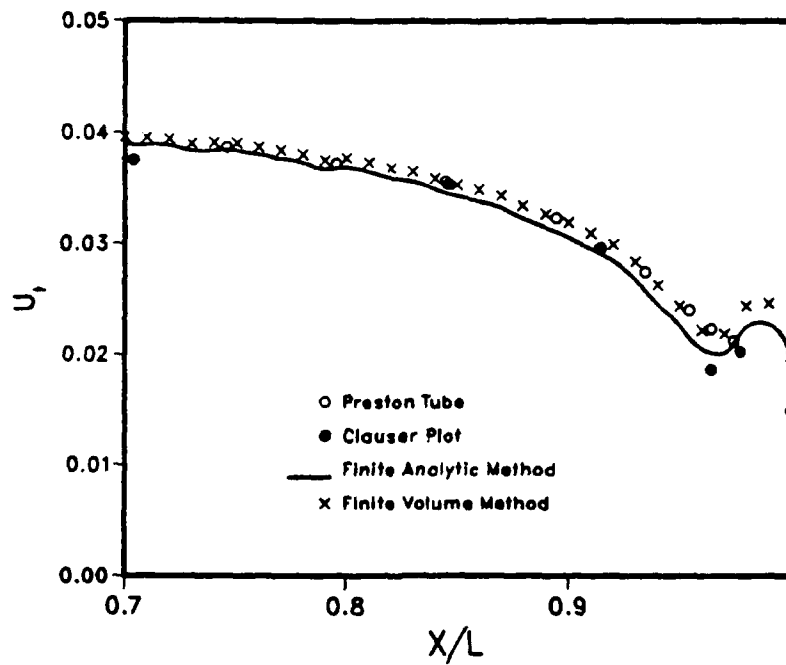


Fig.VI-12. Friction Velocity Distribution  
( Afterbody-1 )

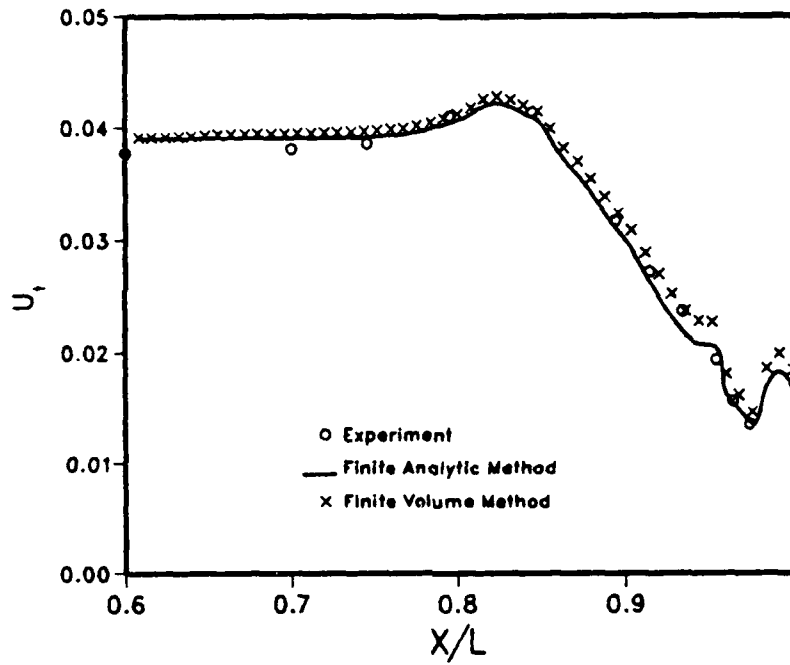


Fig.VI-13. Friction Velocity Distribution  
( Afterbody-2 )

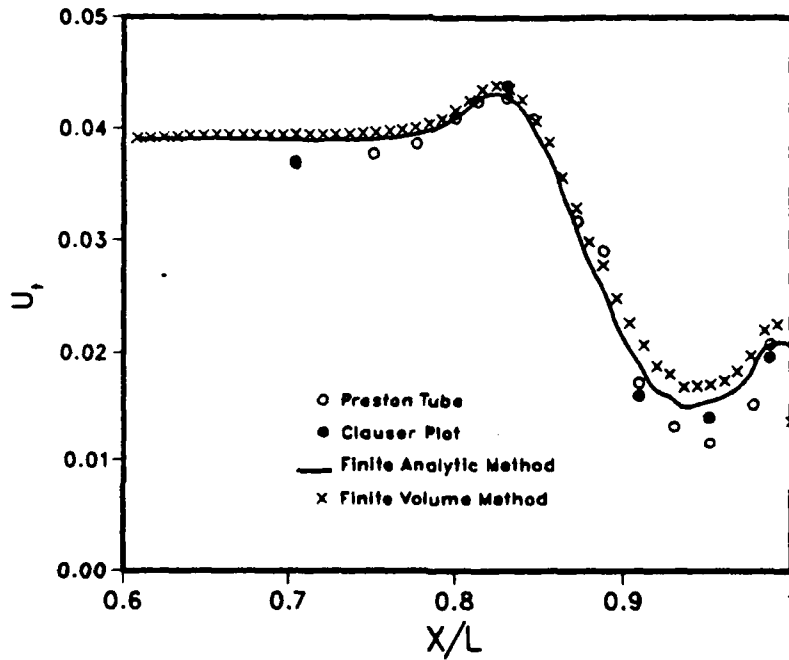


Fig.VI-14. Friction Velocity Distribution  
( Afterbody-5 )

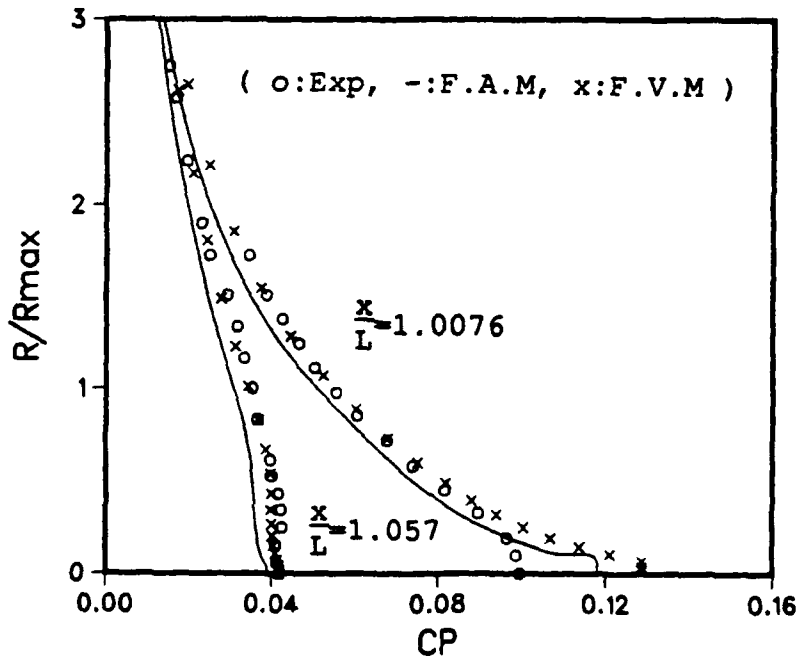
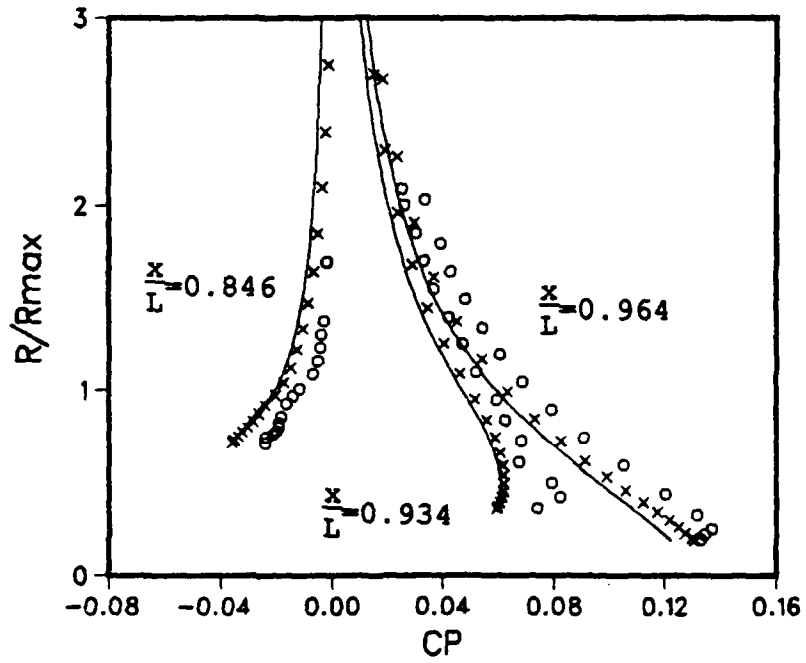


Fig.VI-15. Pressure Distributions ( Afterbody-1 )



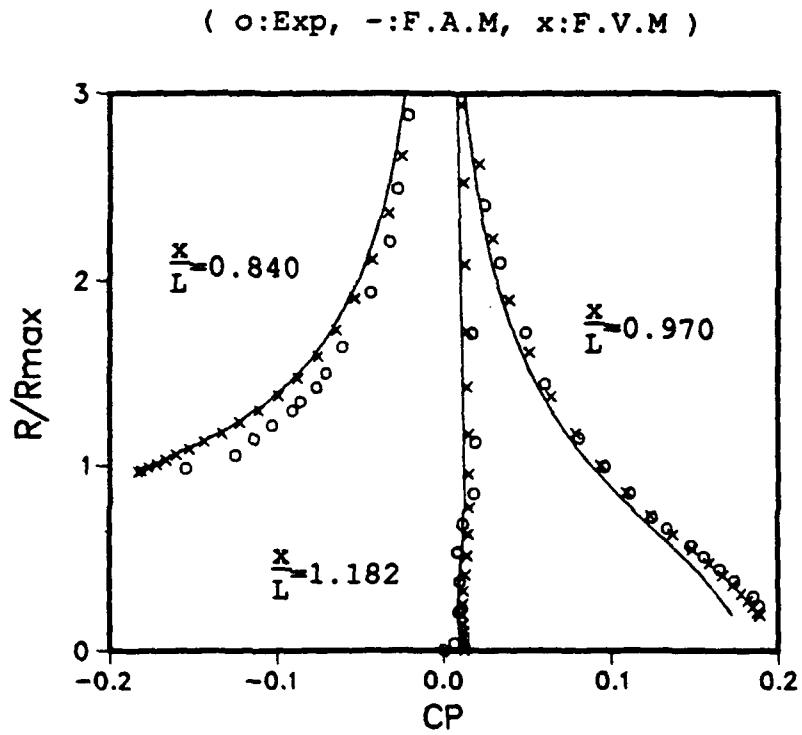


Fig.VI-16. Pressure Distributions ( Afterbody-2 )

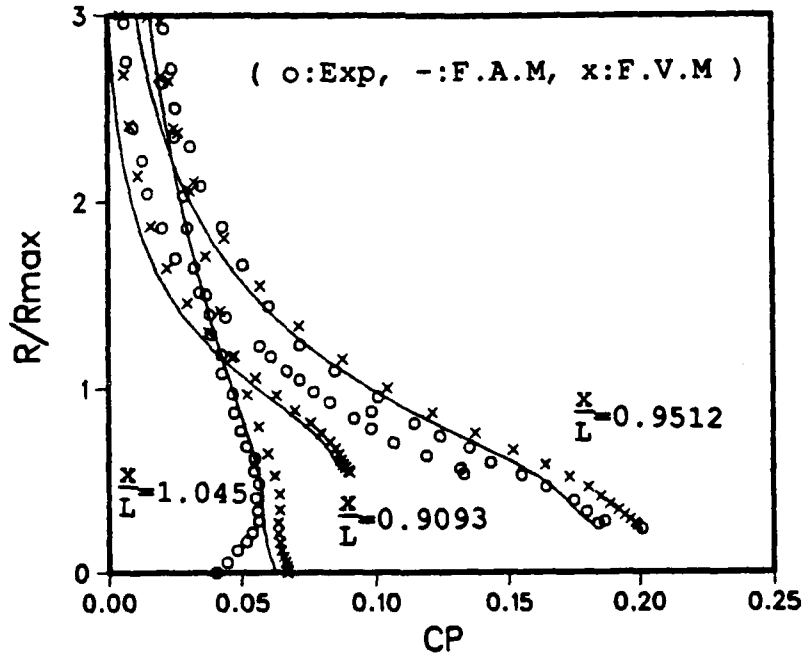
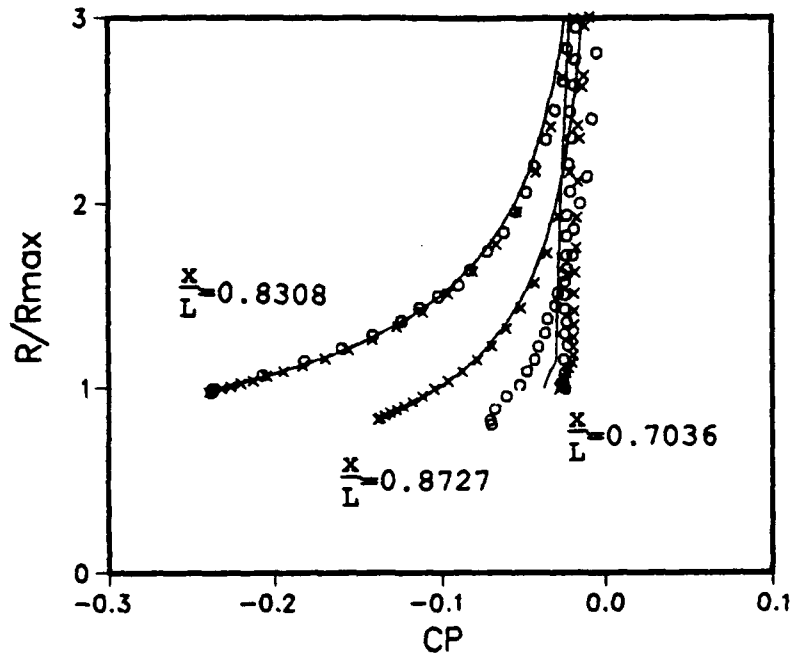
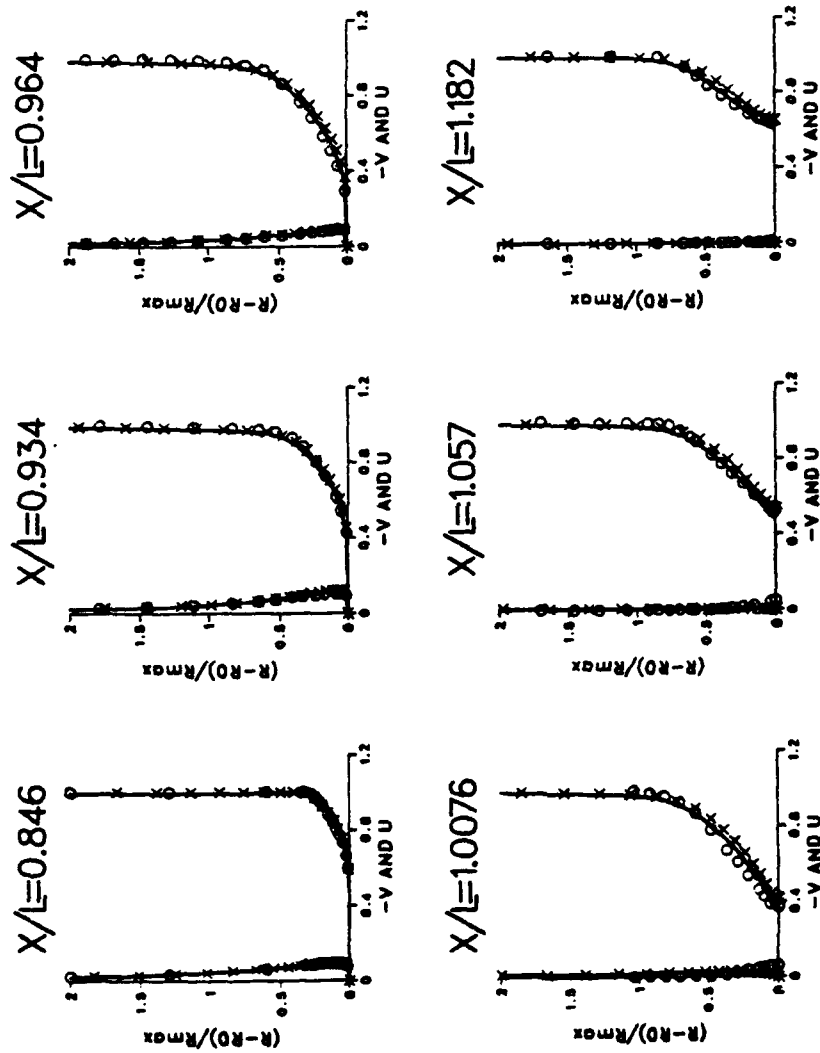
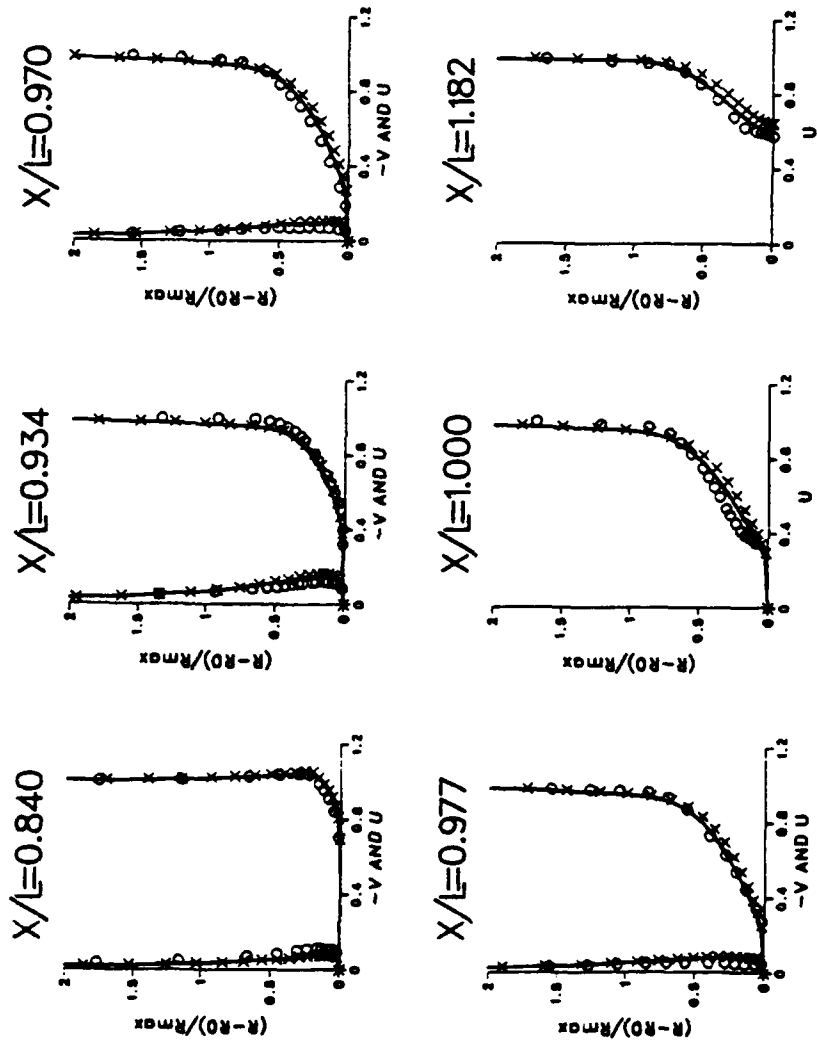


Fig.VI-17. Pressure Distributions ( Afterbody-5 )



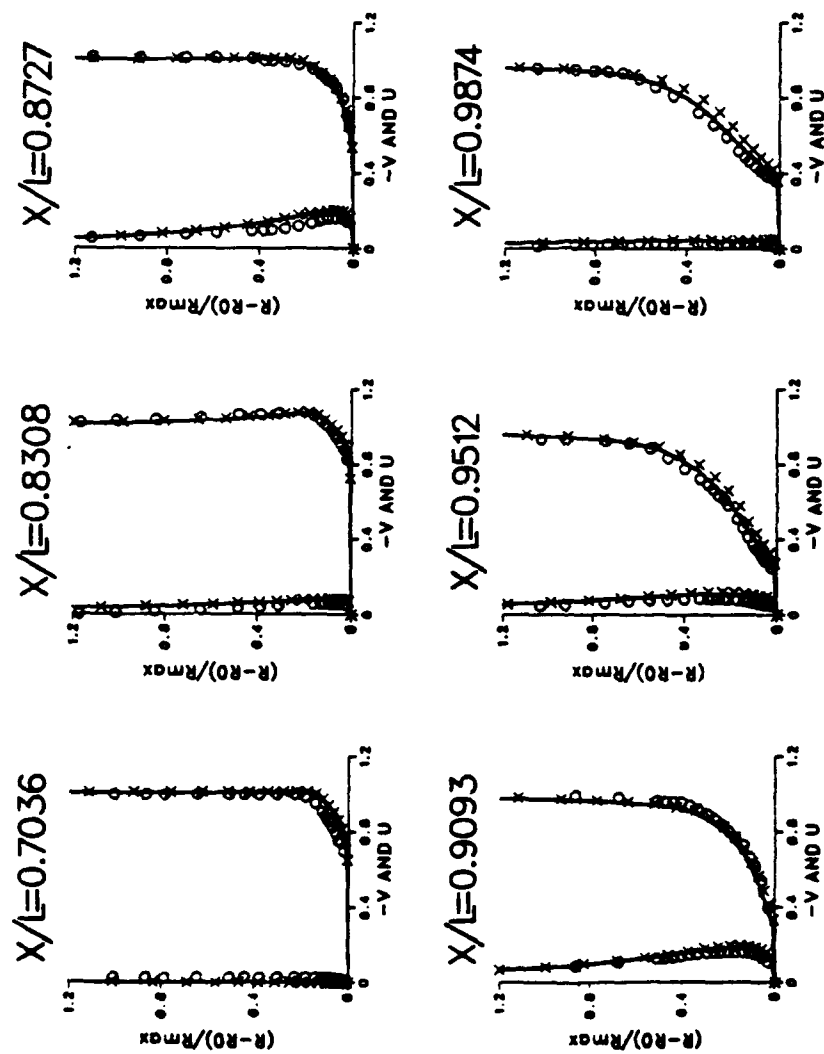
( o:Exp, -:F.A.M, x:F.V.M )

Fig.VI-18. Velocity Profiles ( Afterbody-1 )



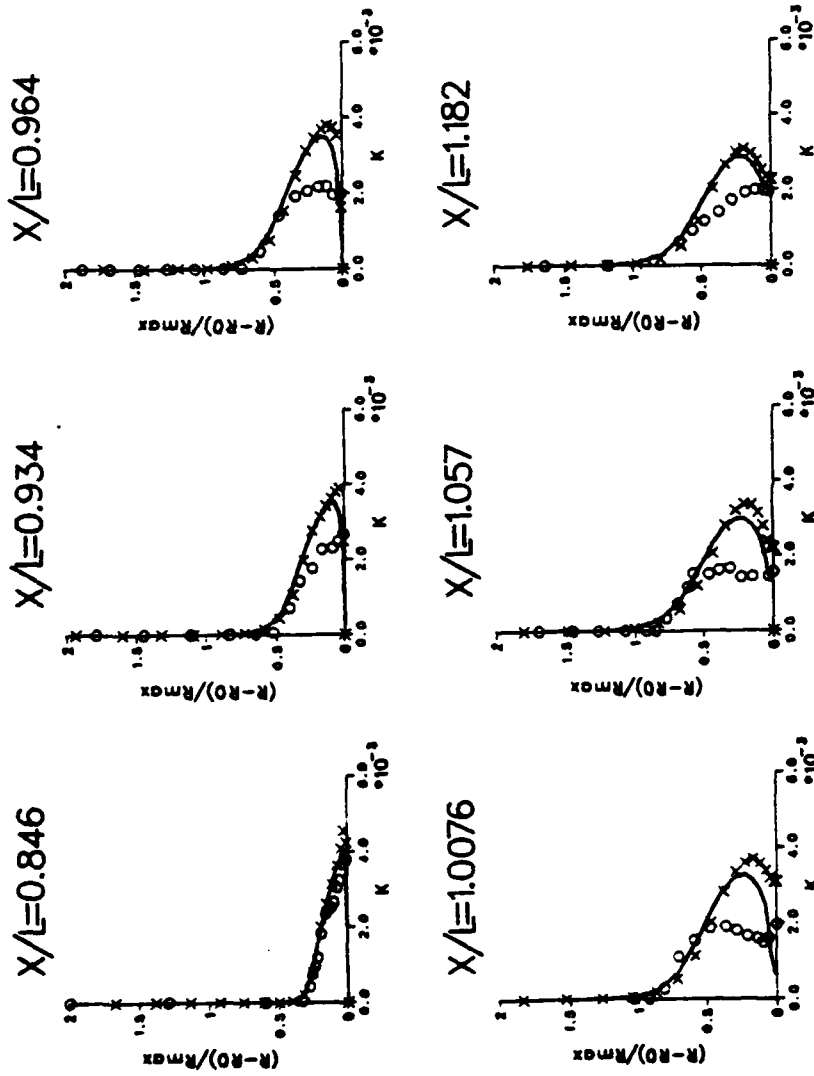
( o:Exp, -:F.A.M, x:F.V.M )

Fig.VI-19. Velocity Profiles ( Afterbody-2 )



( o:Exp, -:F.A.M, x:F.V.M )

Fig.VI-20. Velocity Profiles ( Afterbody-5 )



( o:Exp, -:F.A.M, x:F.V.M )

Fig.VI-21. Turbulent Kinetic Energy Profiles  
( Afterbody-1 )

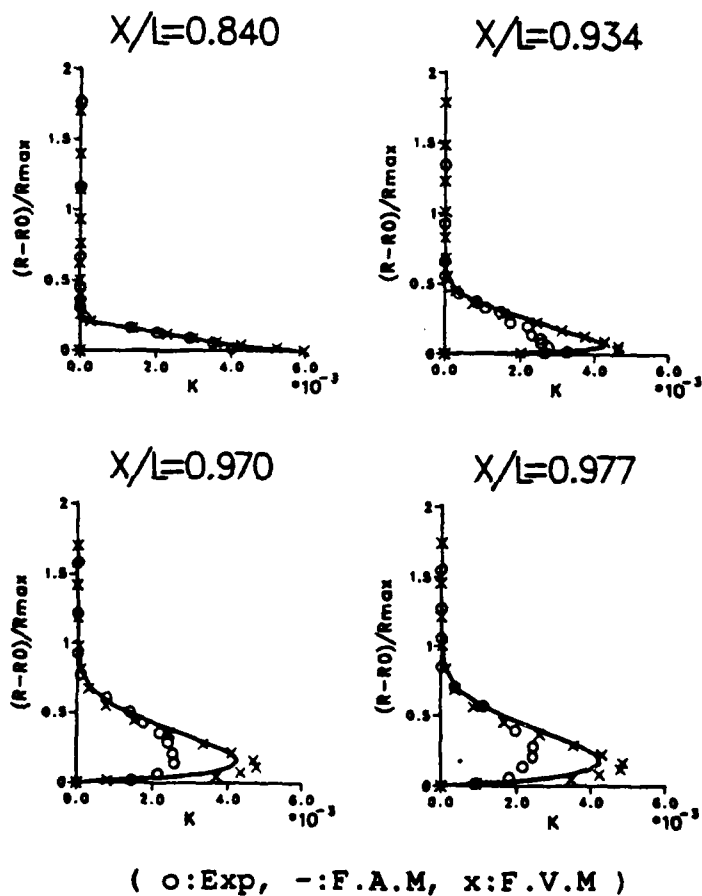
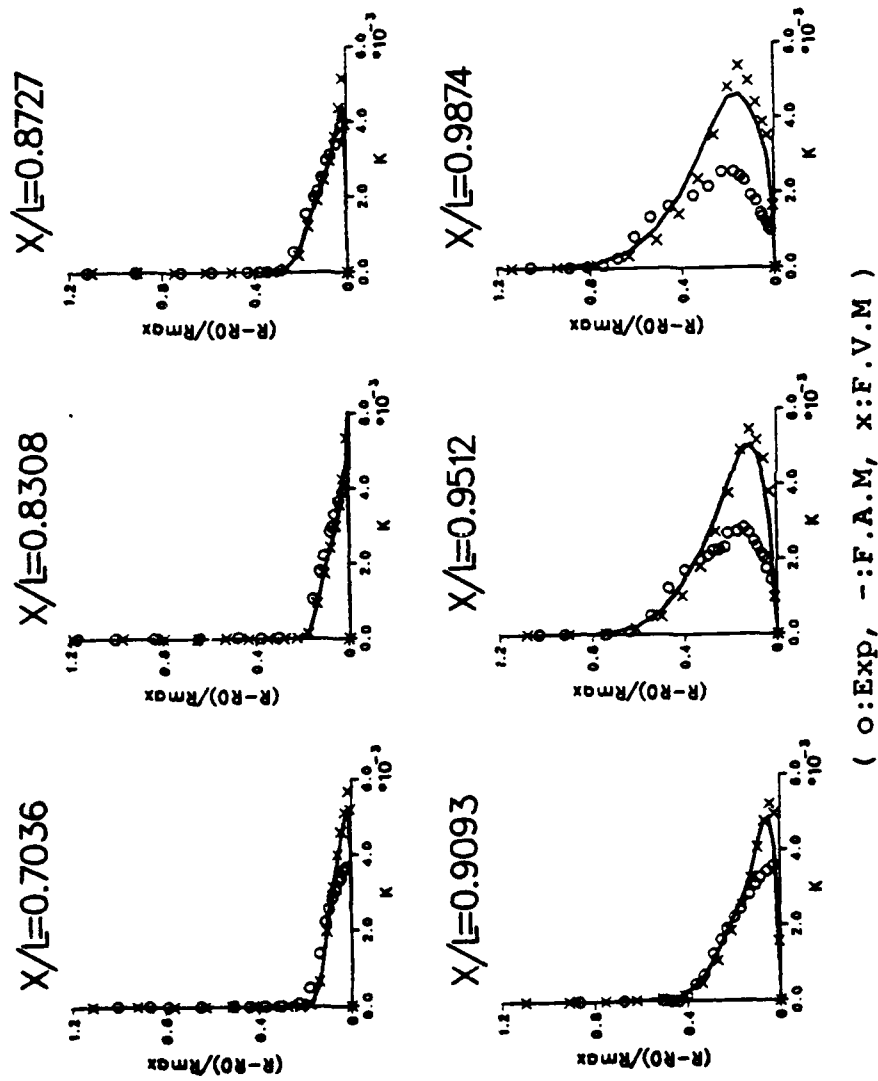


Fig.VI-22. Turbulent Kinetic Energy Profiles  
( Afterbody-2 )



( o:Exp, -:F.A.M, x:F.V.M )

Fig.VI-23. Turbulent Kinetic Energy Profiles  
( Afterbody-5 )



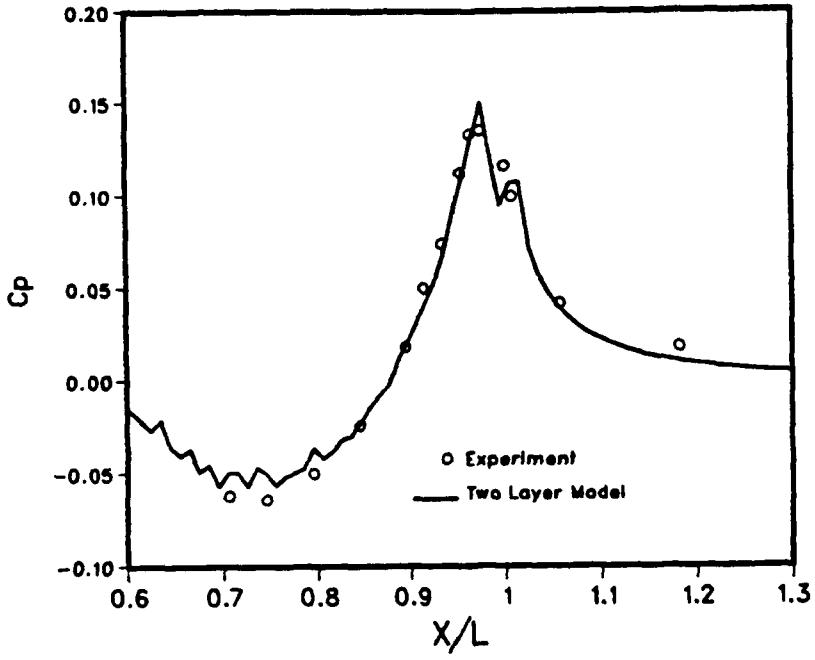


Fig.VI-24. Surface Pressure Distribution ( Afterbody-1 )

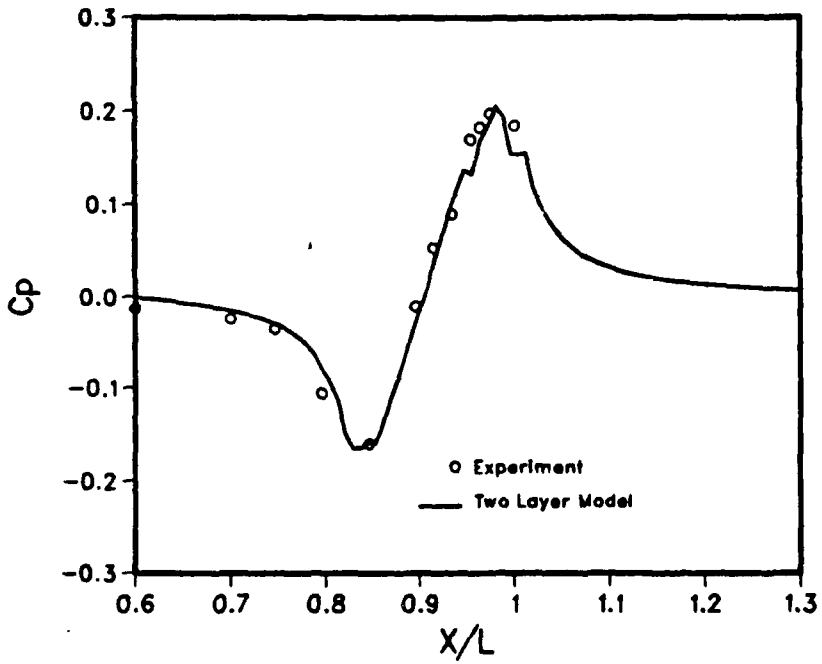


Fig.VI-25. Surface Pressure Distribution ( Afterbody-2 )

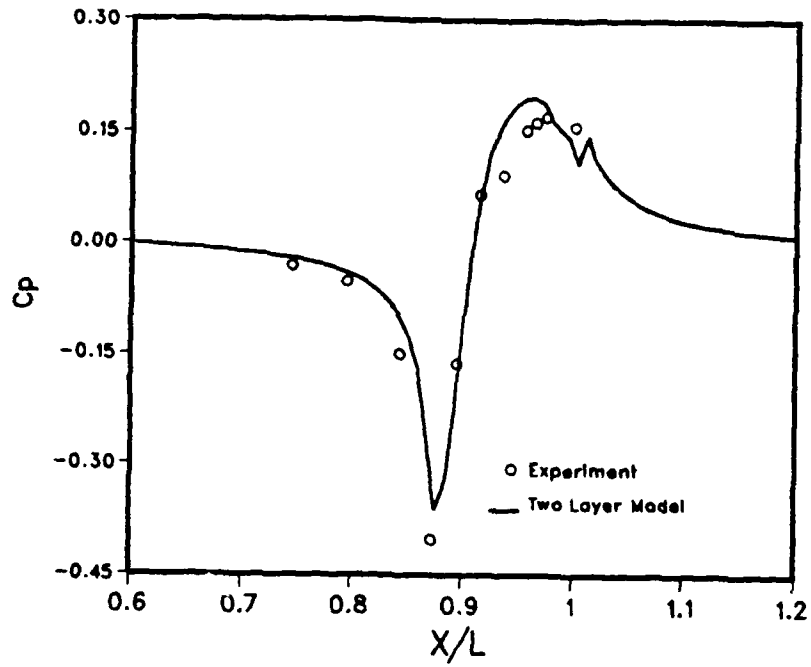


Fig.VI-26. Surface Pressure Distribution ( Afterbody-3 )

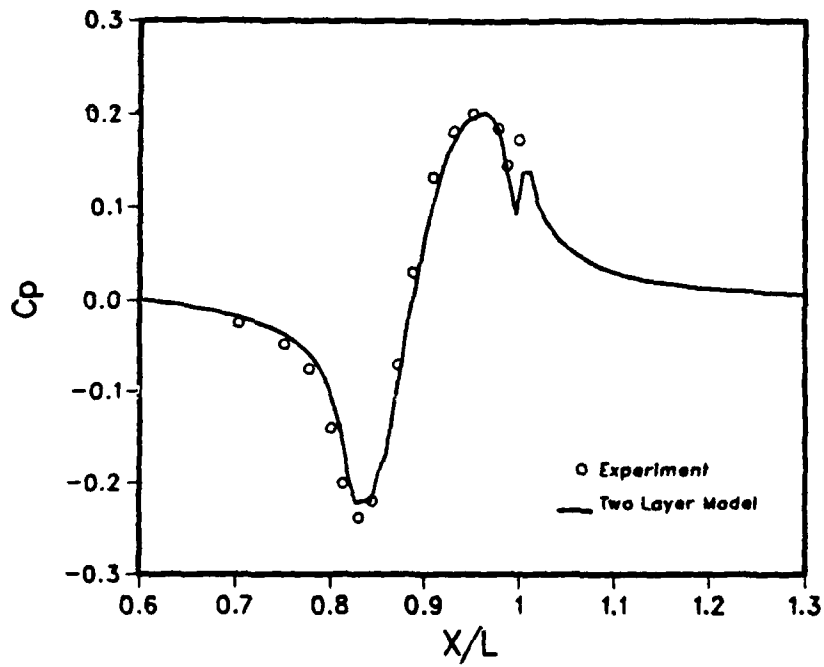


Fig.VI-27. Surface Pressure Distribution ( Afterbody-5 )

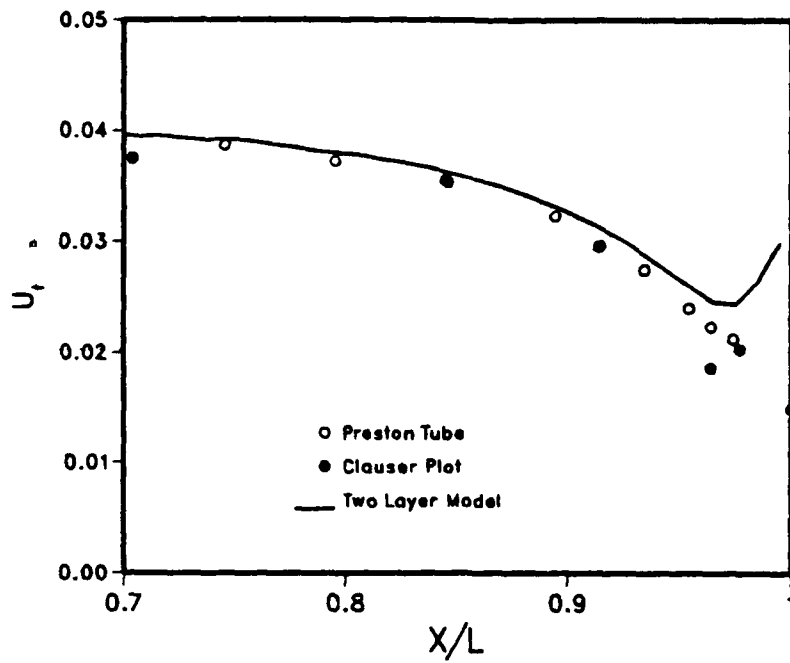


Fig.VI-28. Friction Velocity Distribution ( Afterbody-1 )

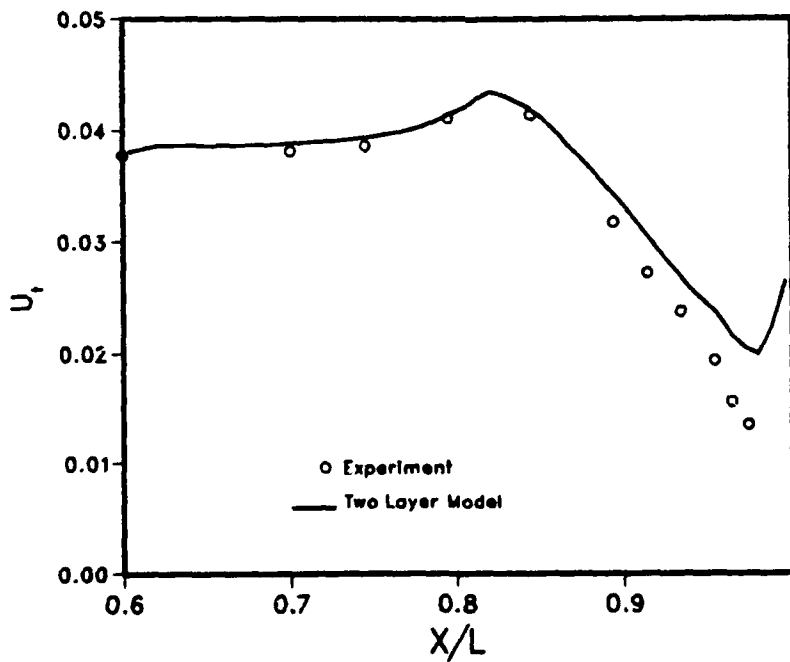


Fig.VI-29. Friction Velocity Distribution ( Afterbody-2 )

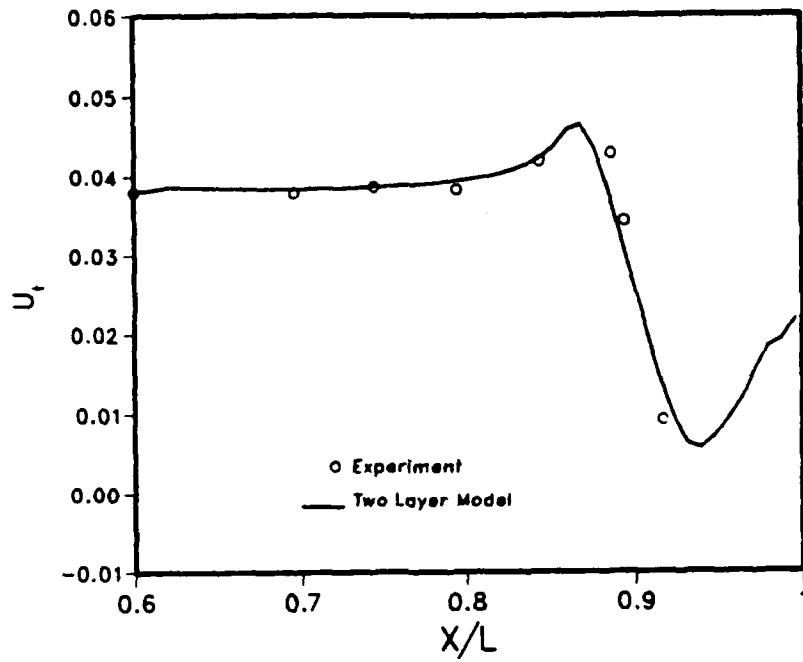


Fig.VI-30. Friction Velocity Distribution  
( Afterbody-3 )

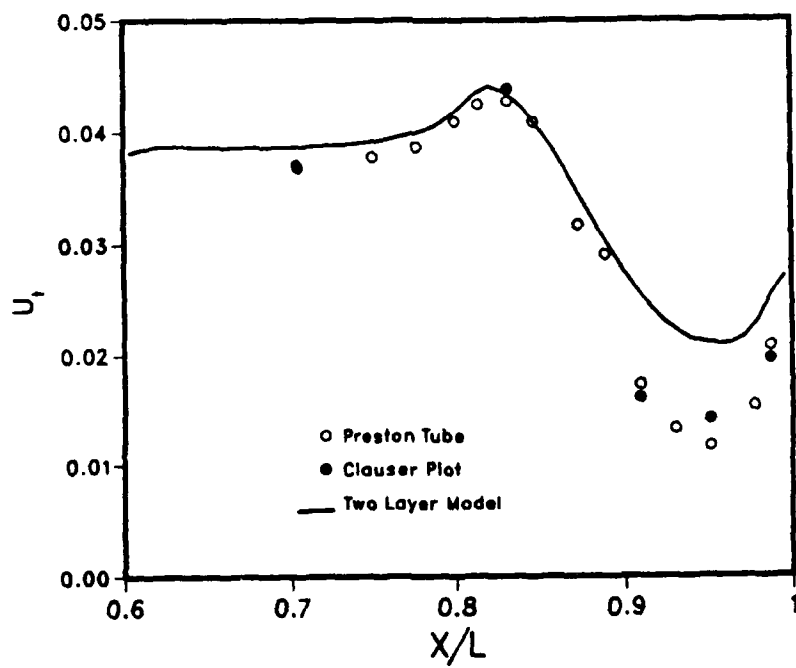


Fig.VI-31. Friction Velocity Distribution  
( Afterbody-5 )

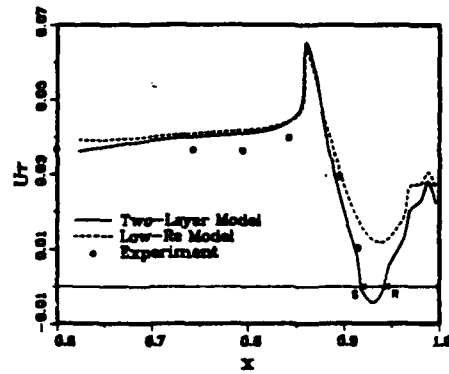


Fig.VI-32. Friction Velocity Distribution for Afterbody-3 Given in Ref.[56]

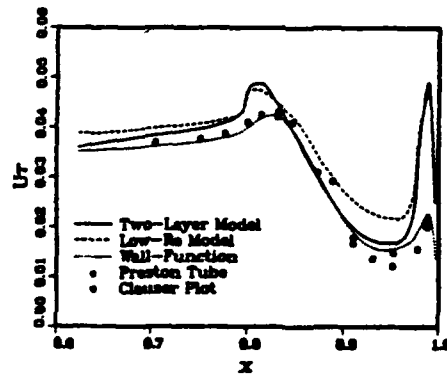


Fig.VI-33. Friction Velocity Distribution for Afterbody-5 Given in Ref.[56]

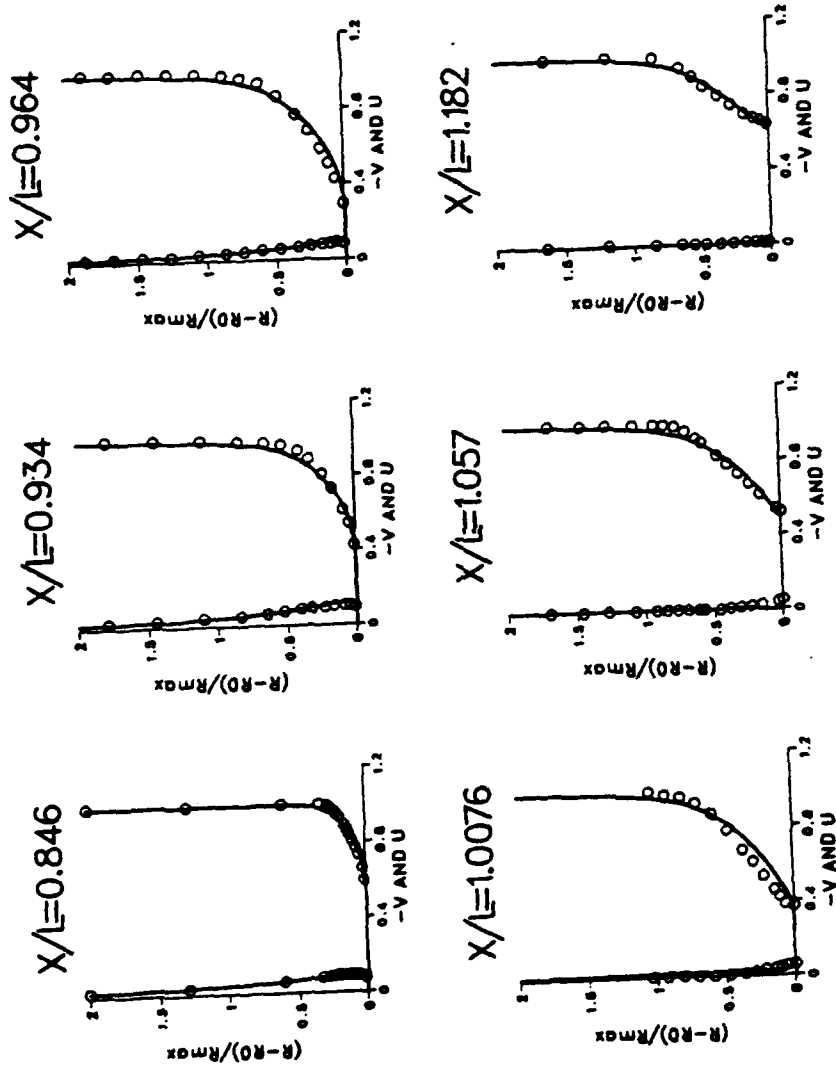


Fig.VI-34. Velocity Profiles ( Afterbody-1 )

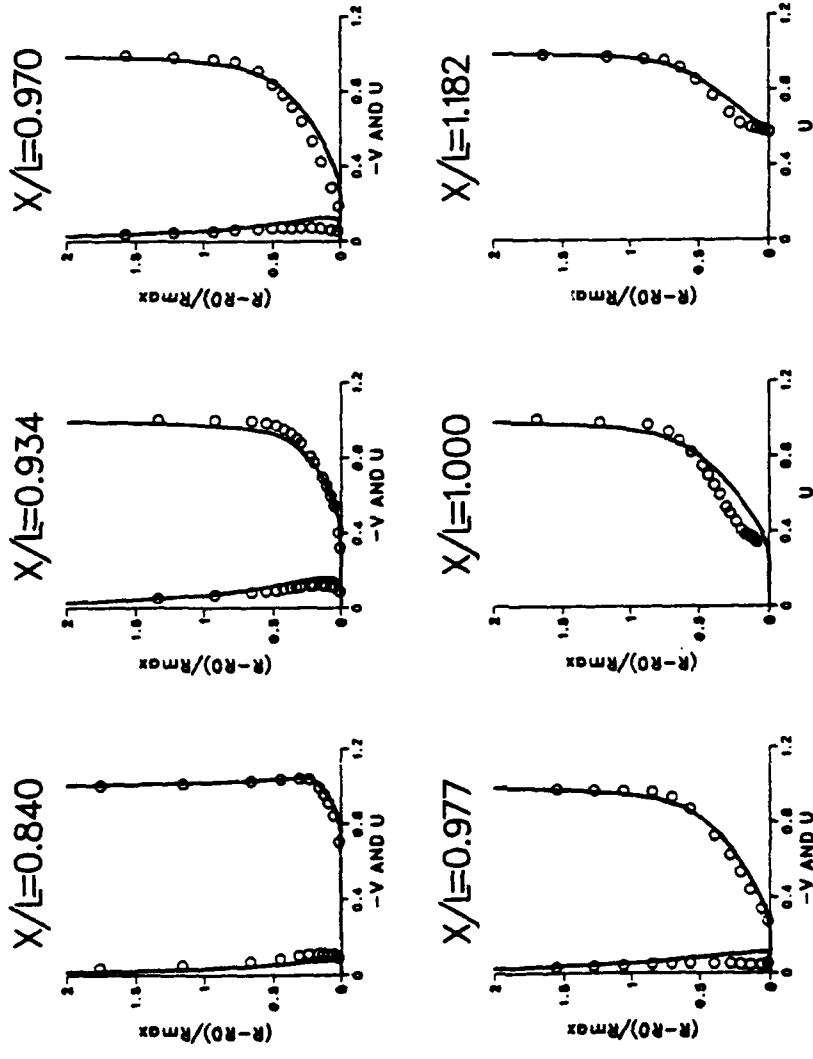


Fig. VI-35. Velocity Profiles ( Afterbody-2 )

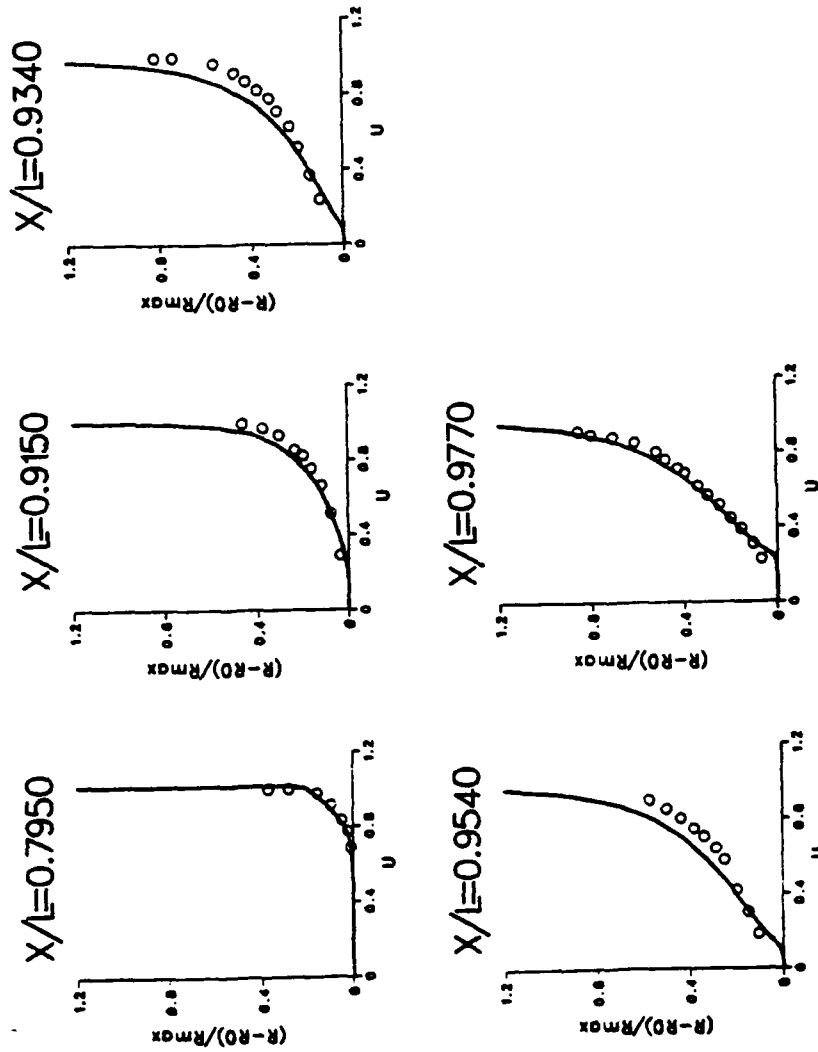


Fig.VI-36. Velocity Profiles ( Afterbody-3 )



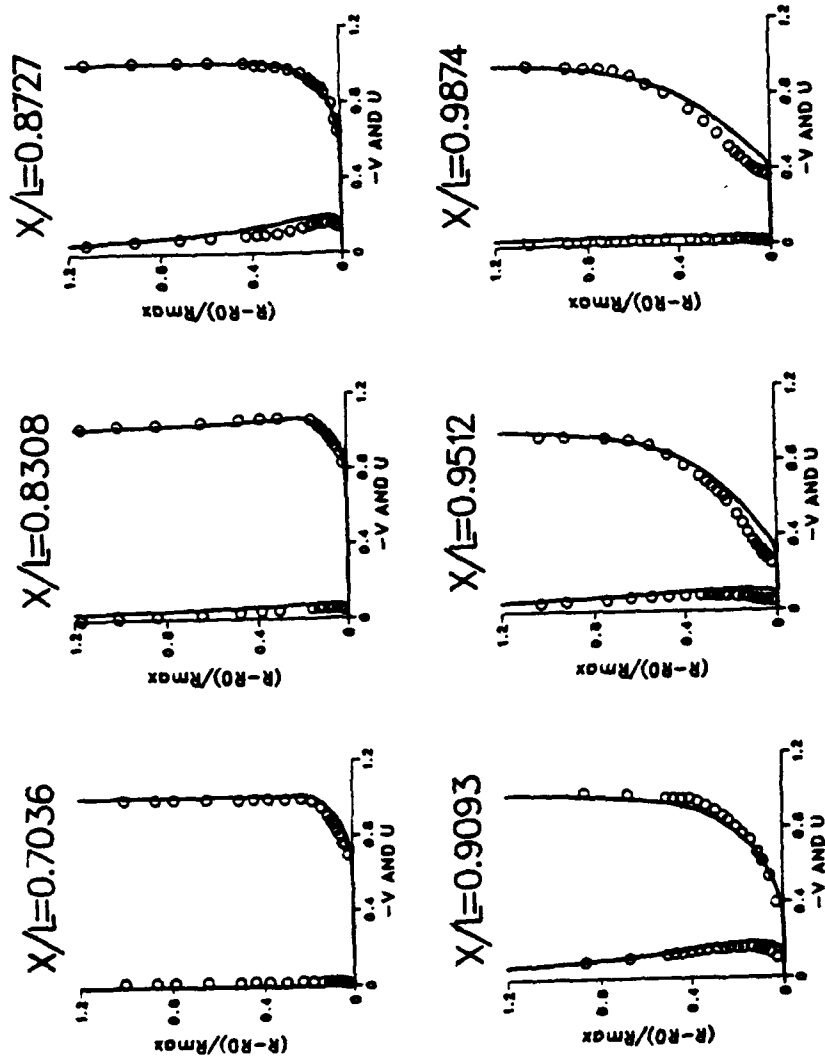


Fig.VI-37. Velocity Profiles ( Afterbody-5 )

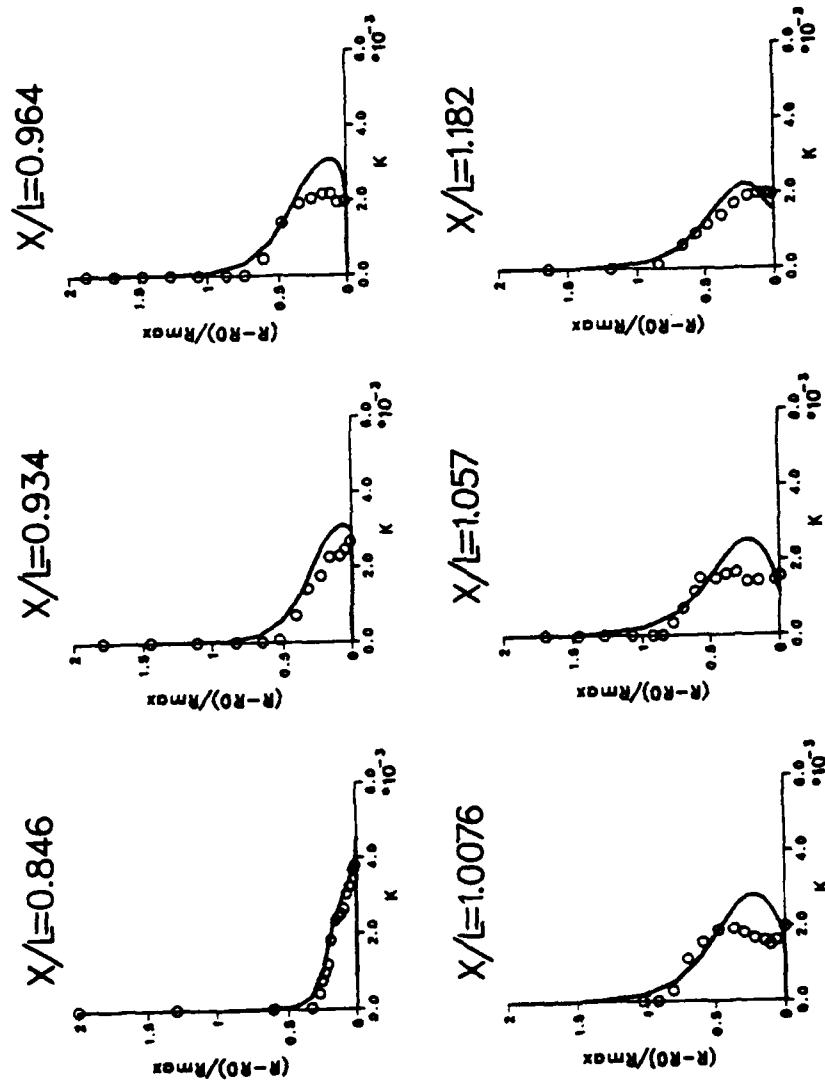


Fig.VI-38. Turbulent Kinetic Energy Profiles  
( Afterbody-1 )

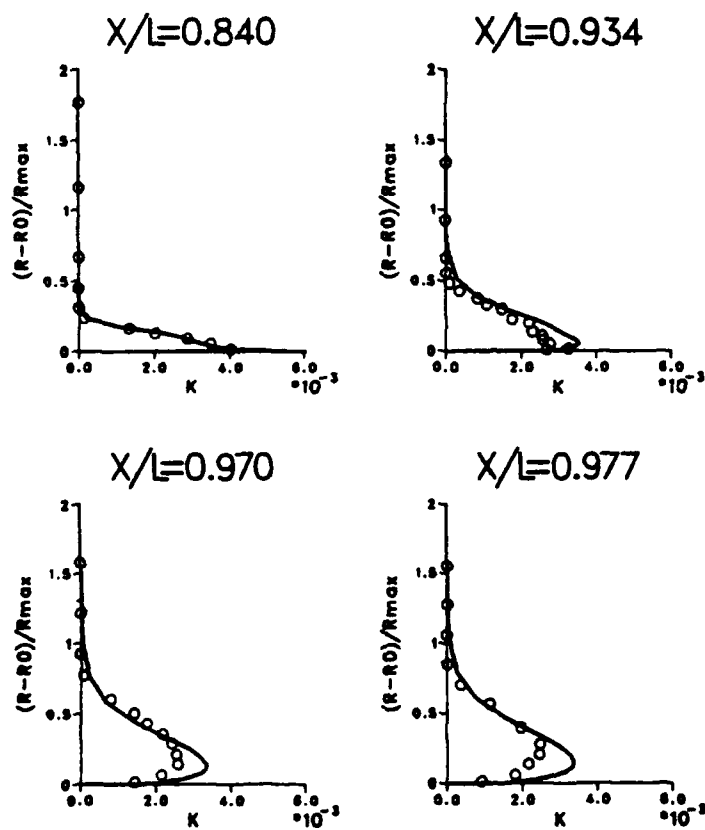


Fig.VI-39. Turbulent Kinetic Energy Profiles  
( Afterbody-2 )

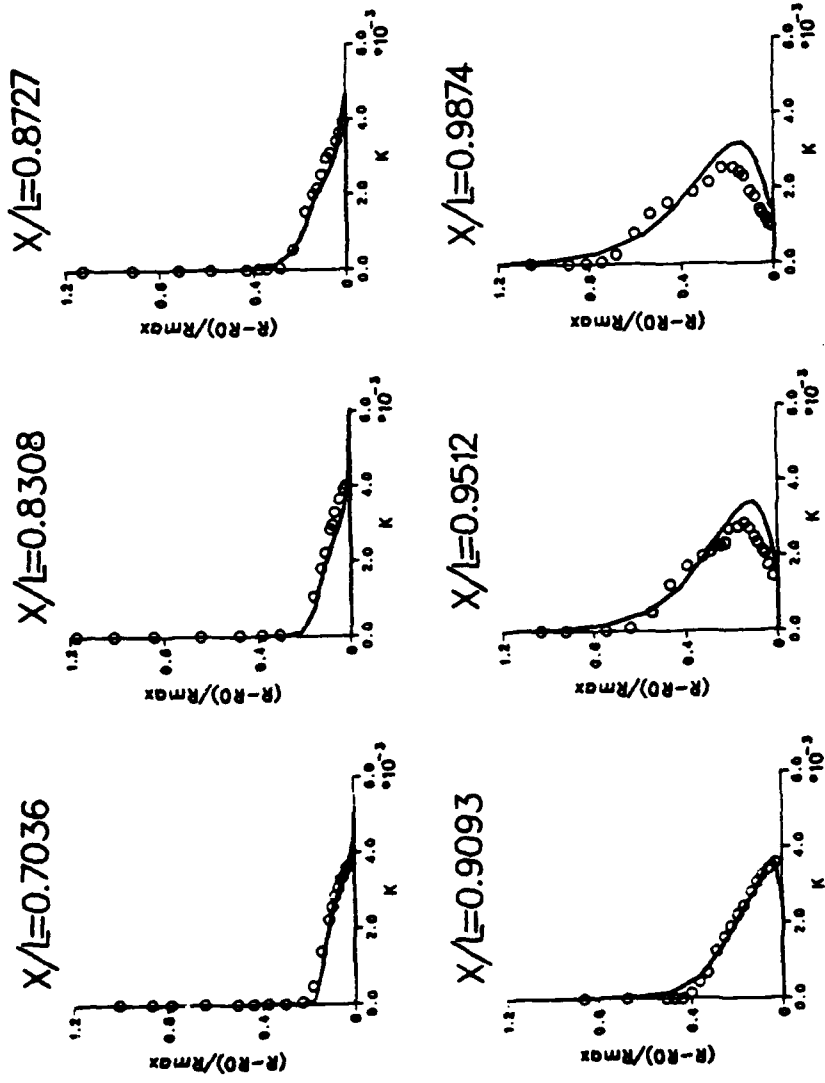


Fig.VI-40. Turbulent Kinetic Energy Profiles ( Afterbody-5 )

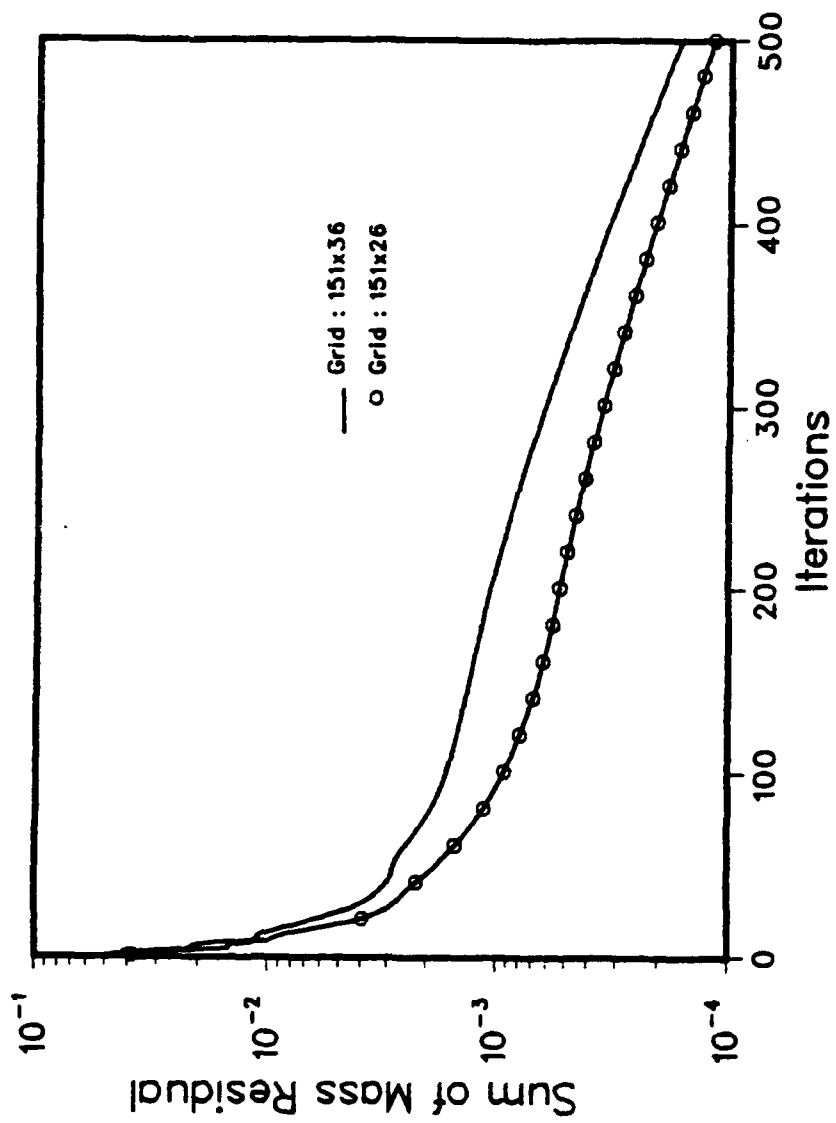


Fig.VII-1. Convergence History ( Afterbody-1 )

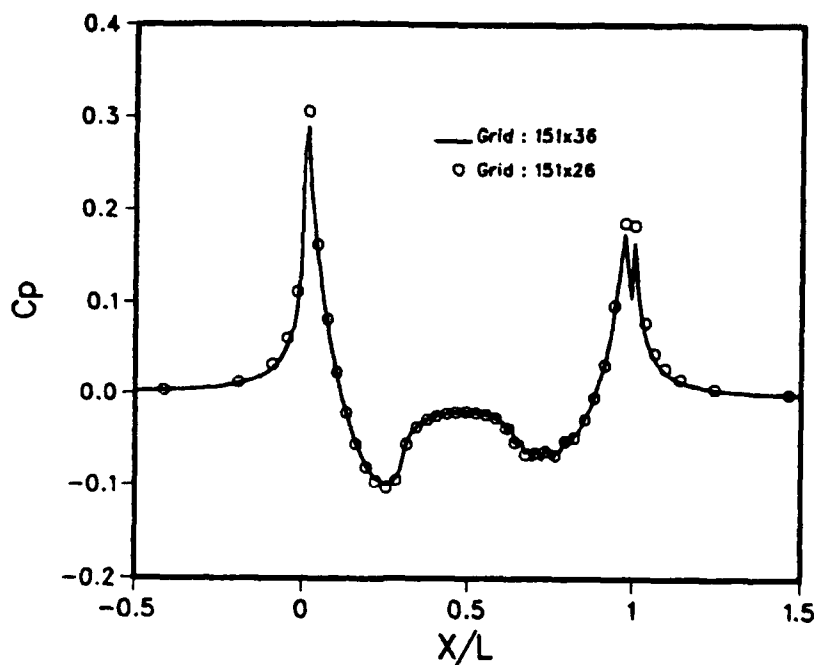


Fig.VII-2. Grid Dependence Test for Pressure  
( Afterbody-1 )

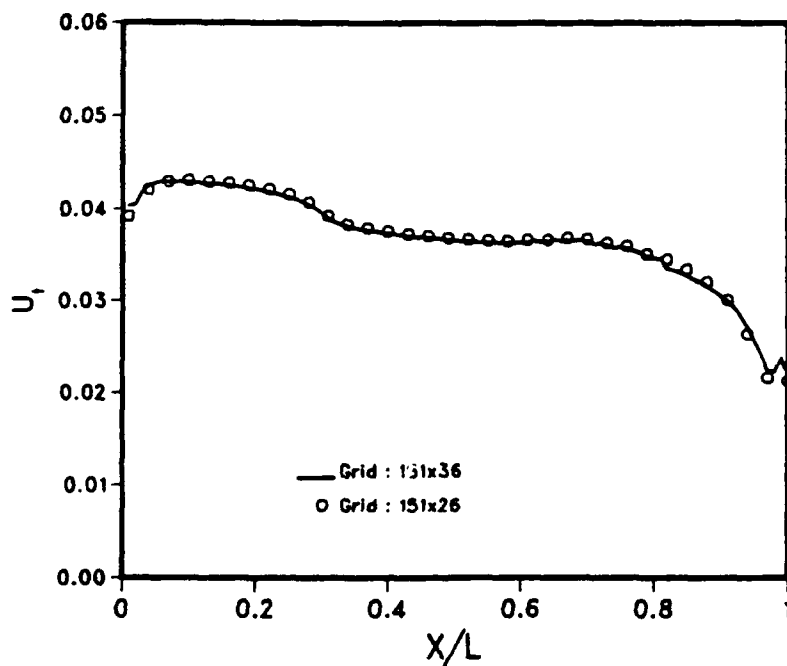


Fig.VII-3. Grid Dependence Test for Friction  
Velocity ( Afterbody-1 )

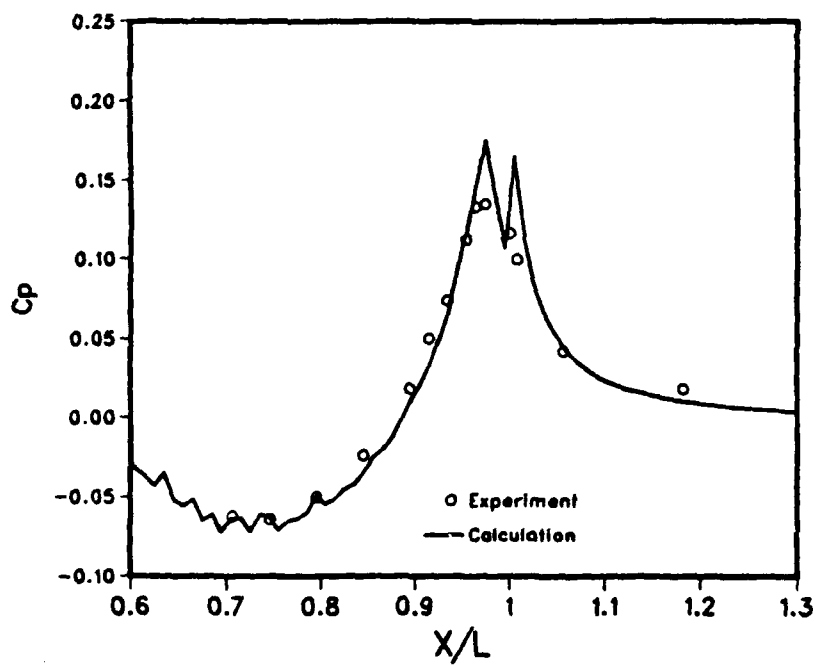
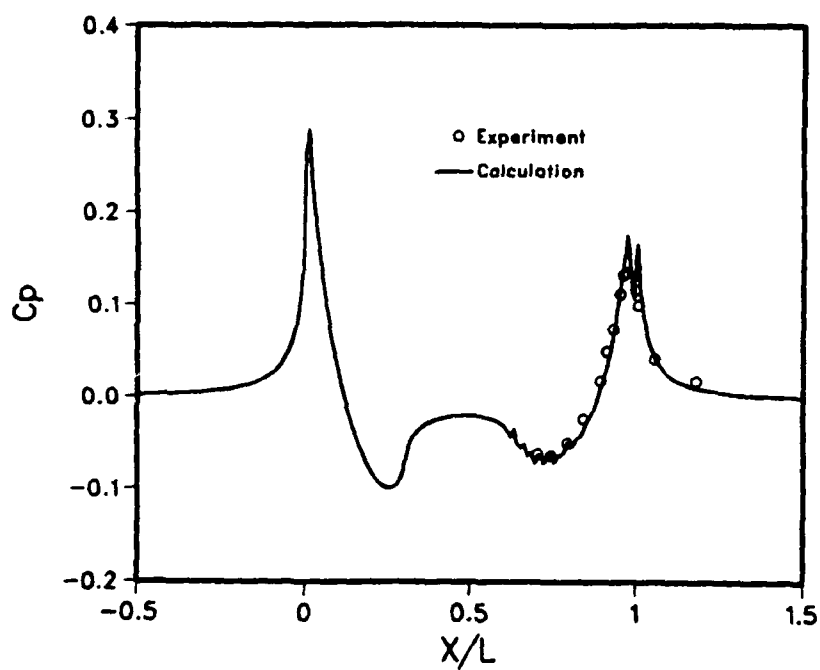


Fig.VII-4. Surface Pressure Distribution  
( Afterbody-1 )

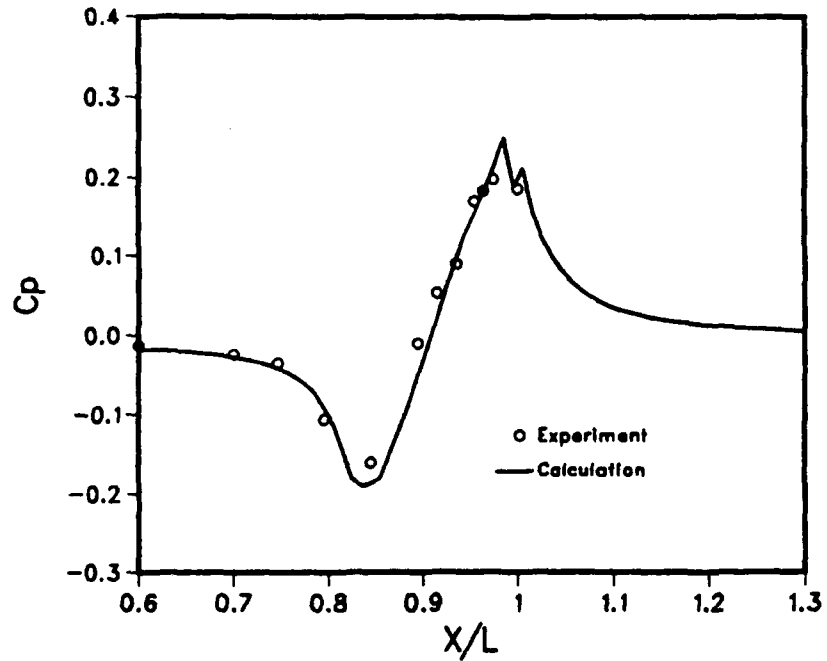
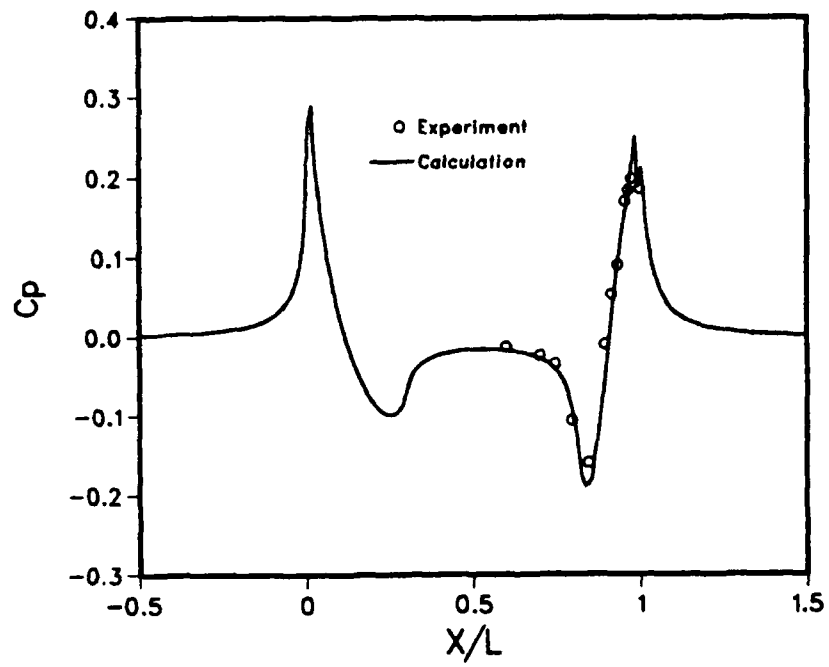


Fig.VII-5. Surface Pressure Distribution  
( Afterbody-2 )



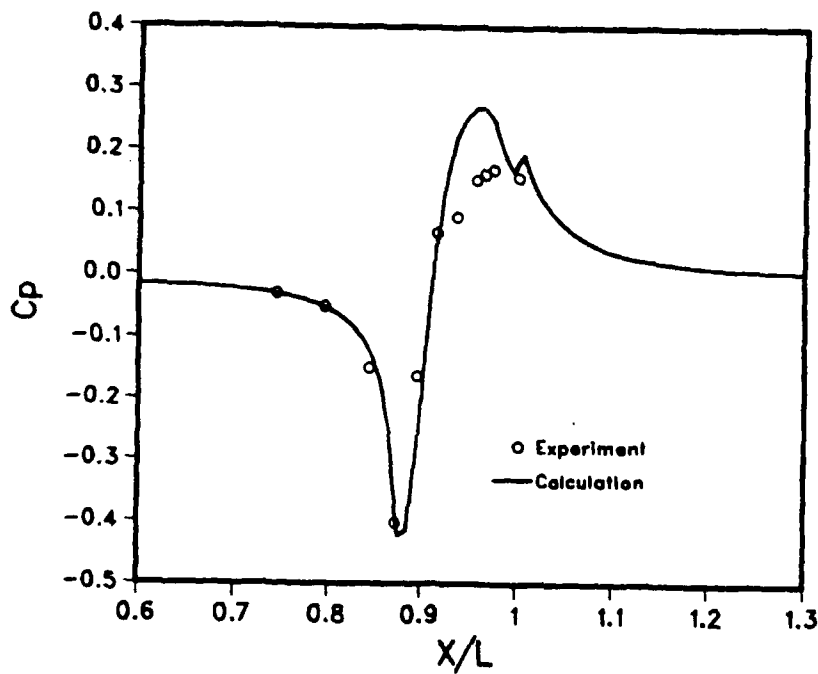
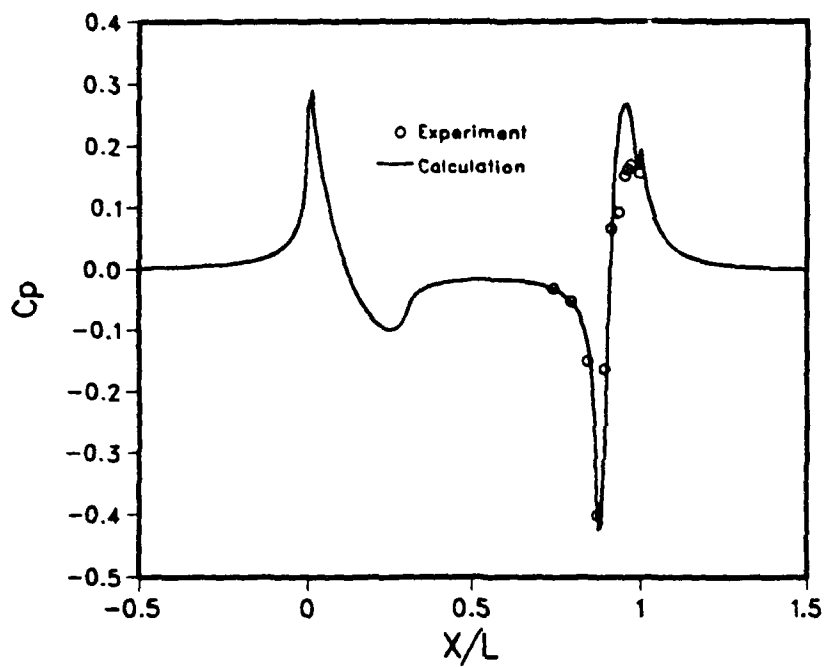


Fig.VII-6. Surface Pressure Distribution  
( Afterbody-3 )

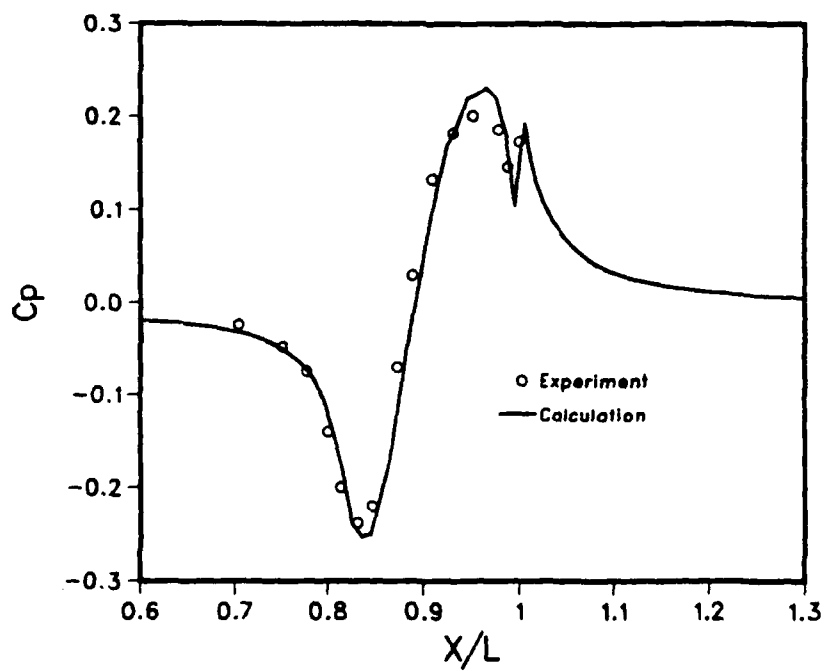
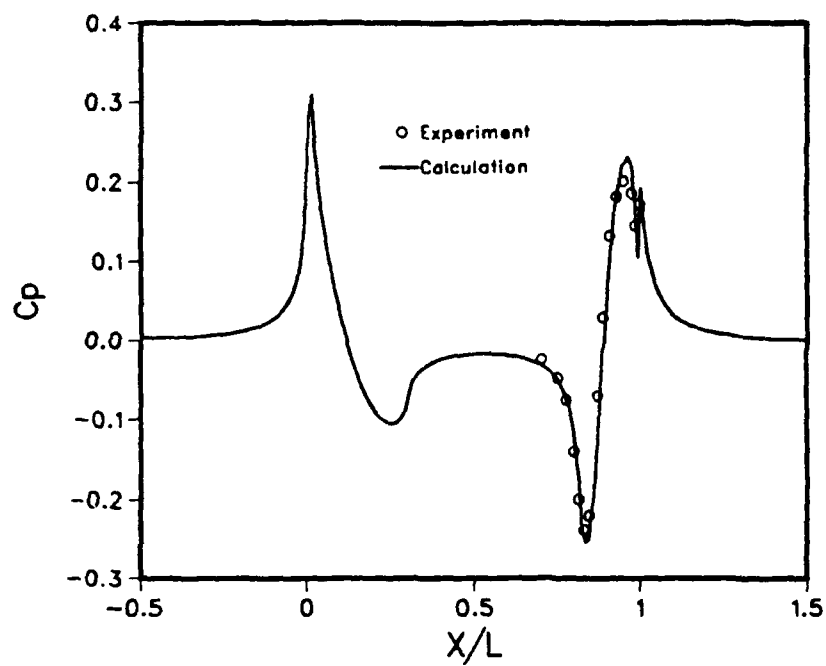


Fig.VII-7. Surface Pressure Distribution  
( Afterbody-5 )

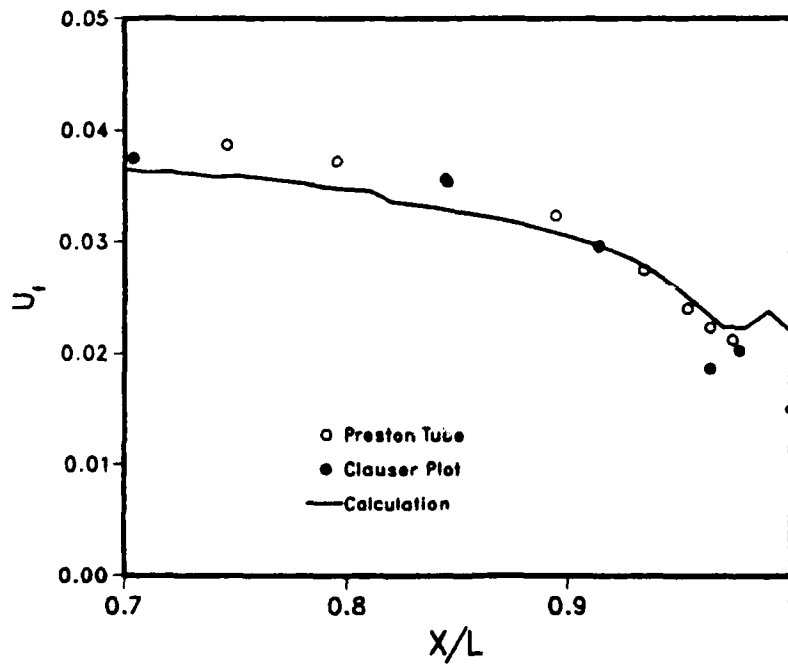
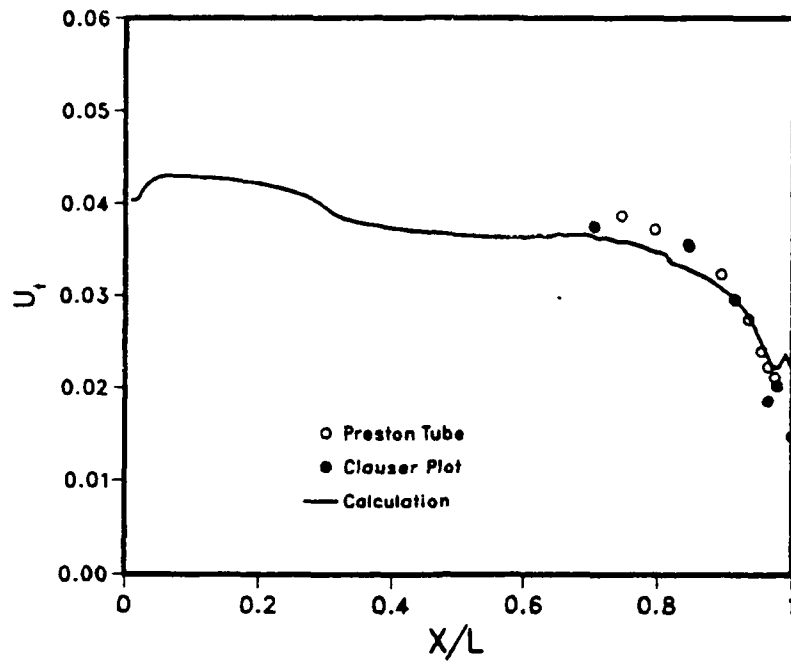


Fig.VII-8. Friction Velocity Distribution  
( Afterbody-1 )

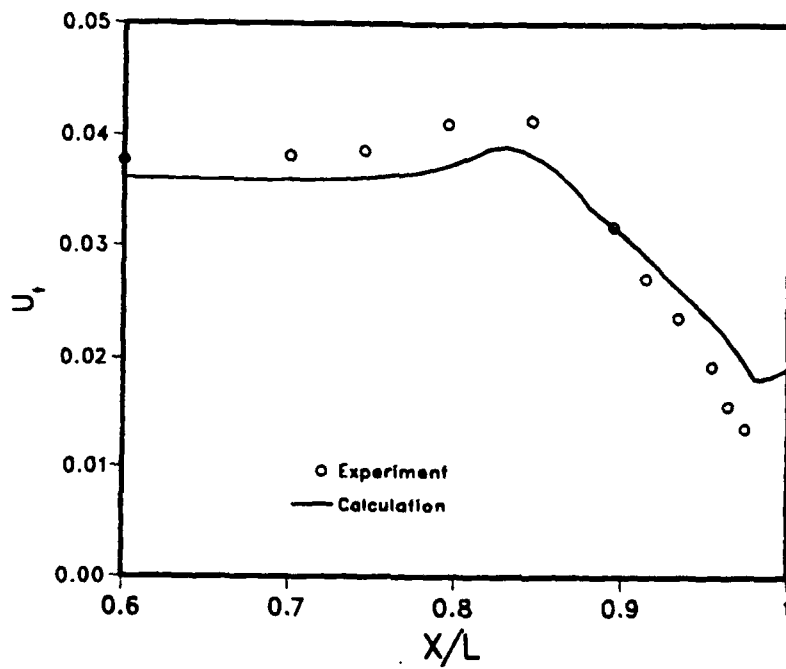
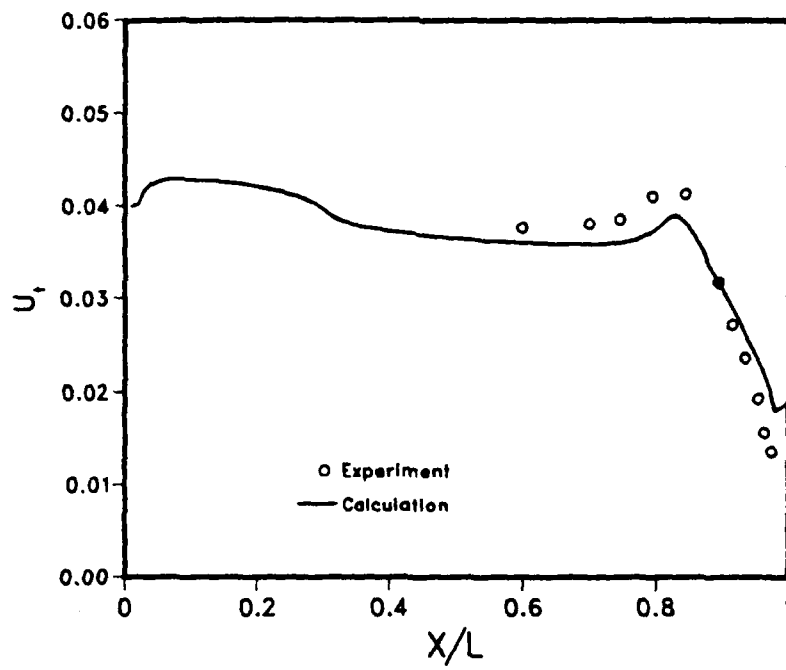


Fig.VII-9. Friction Velocity Distribution  
( Afterbody-2 )

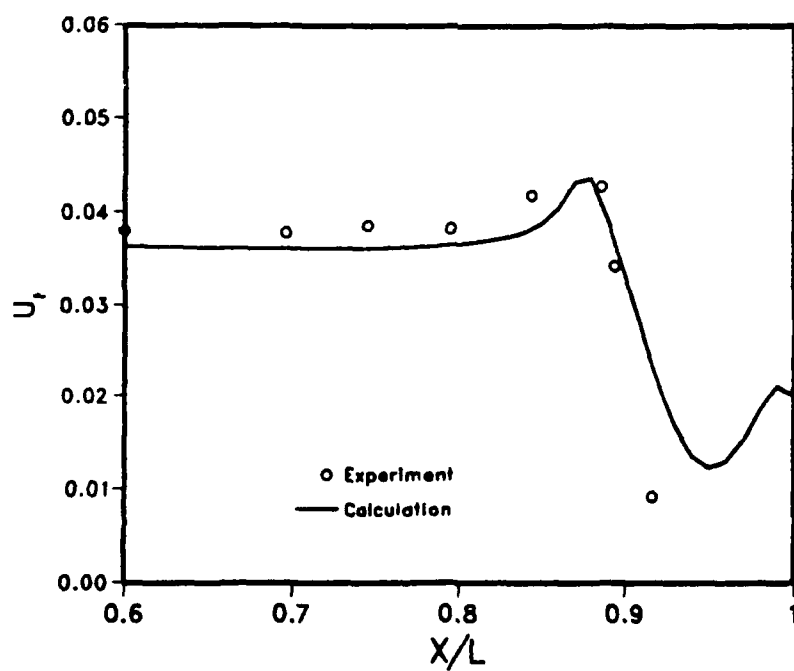
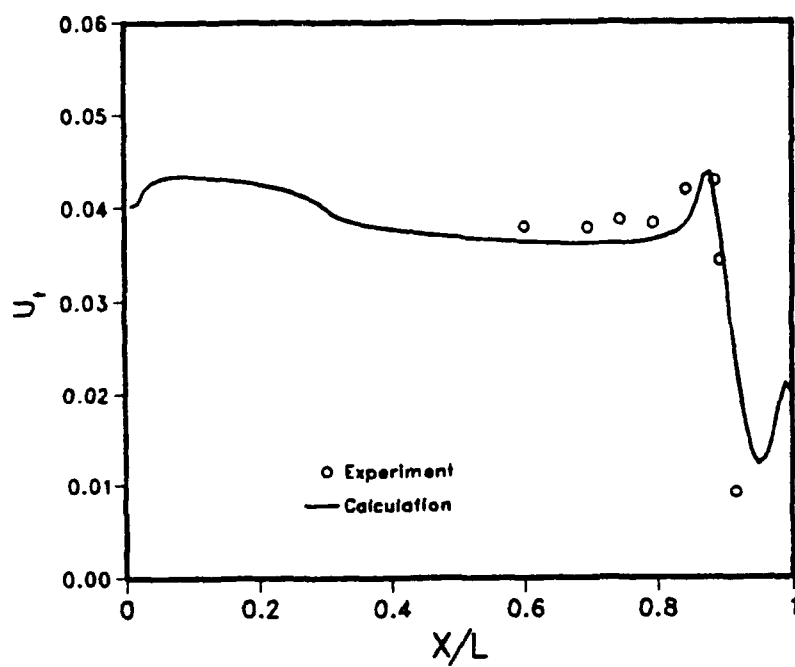


Fig.VII-10. Friction Velocity Distribution  
( Afterbody-3 )

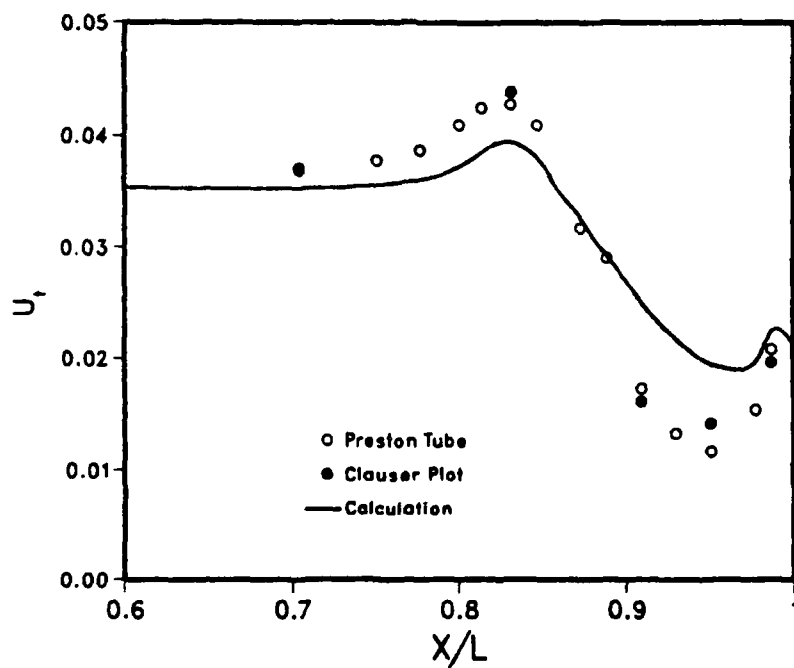
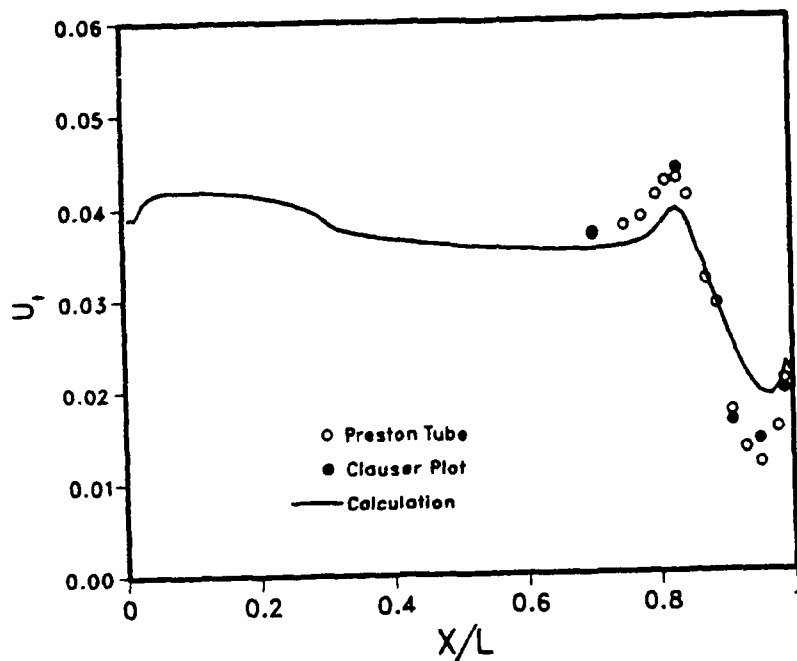


Fig.VII-11. Friction Velocity Distribution  
( Afterbody-5 )

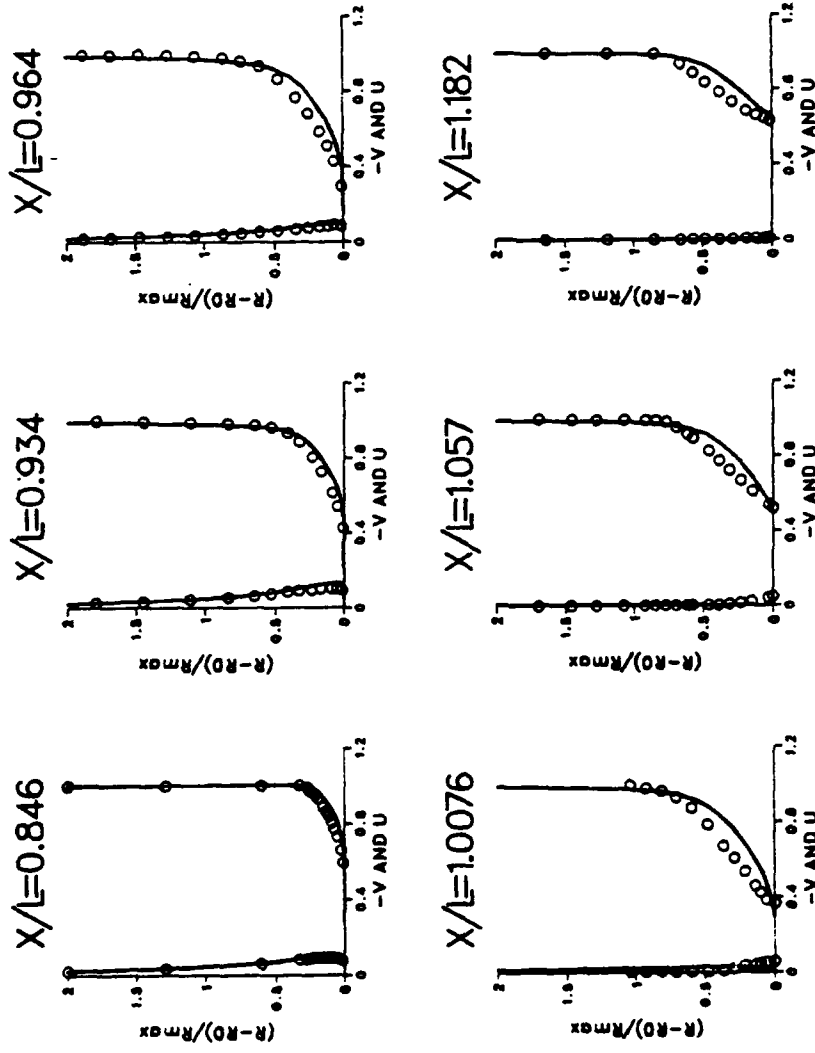


Fig.VII-12. Velocity Profiles ( Afterbody-1 )

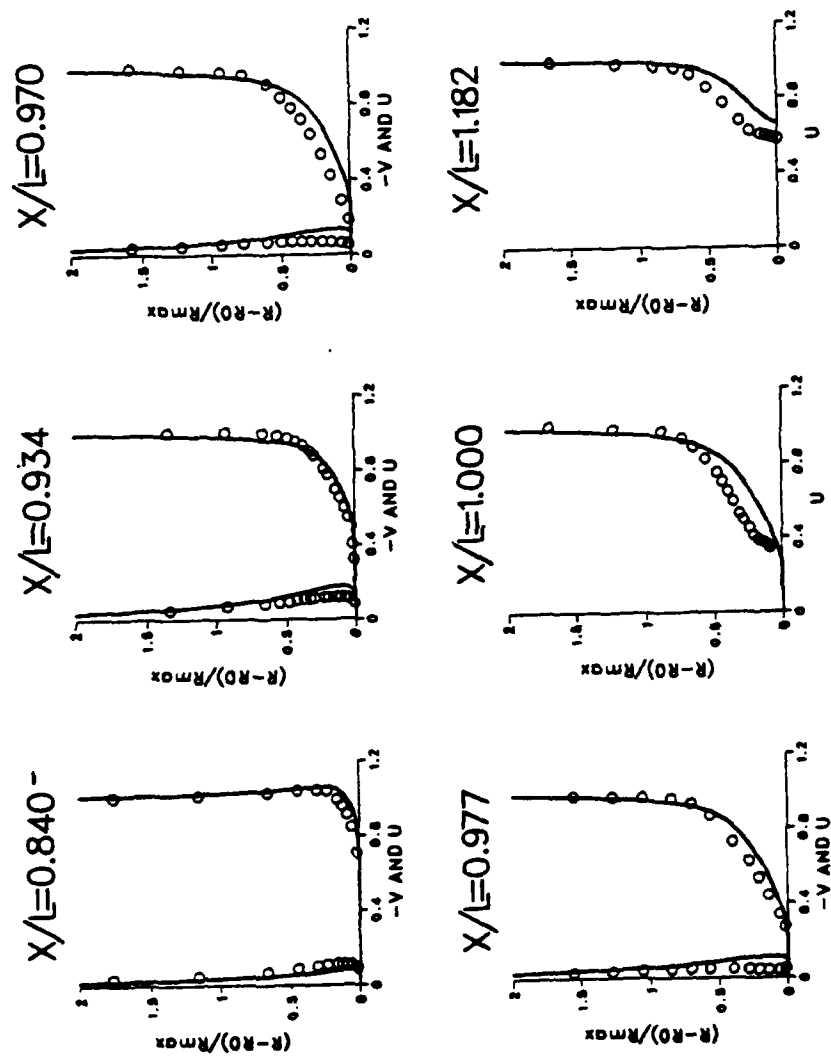


Fig.VII-13. Velocity Profiles ( Afterbody-2 )



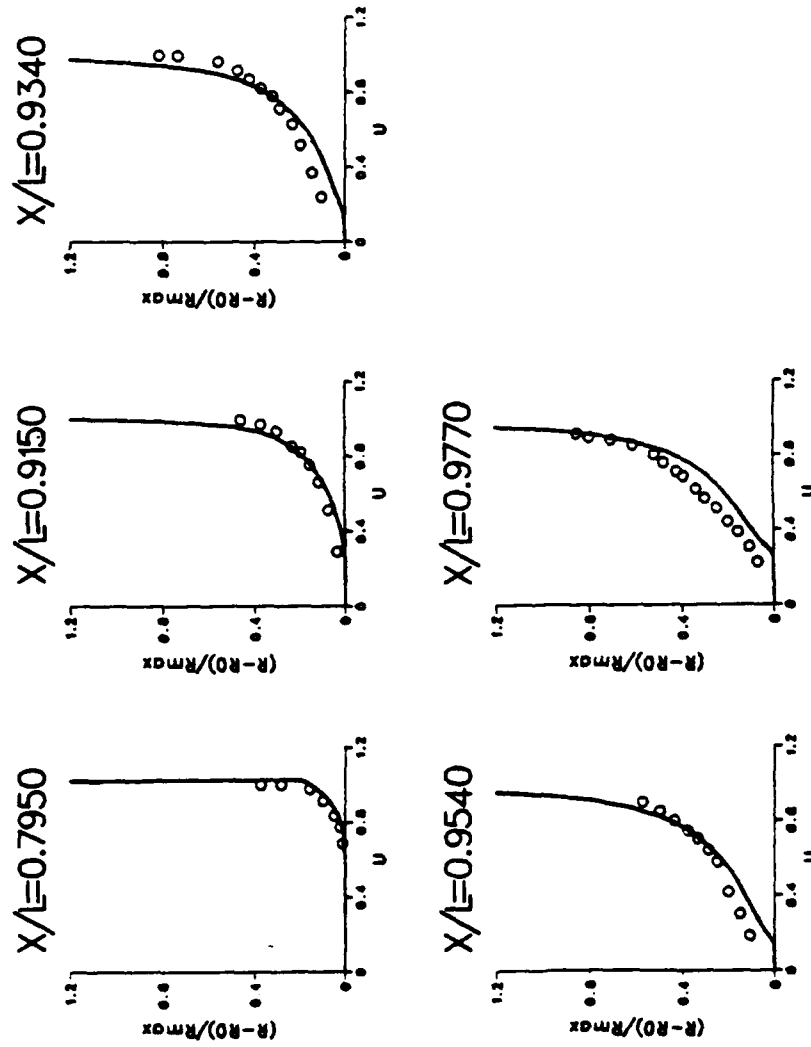


Fig.VII-14. Velocity Profiles ( Afterbody-3 )

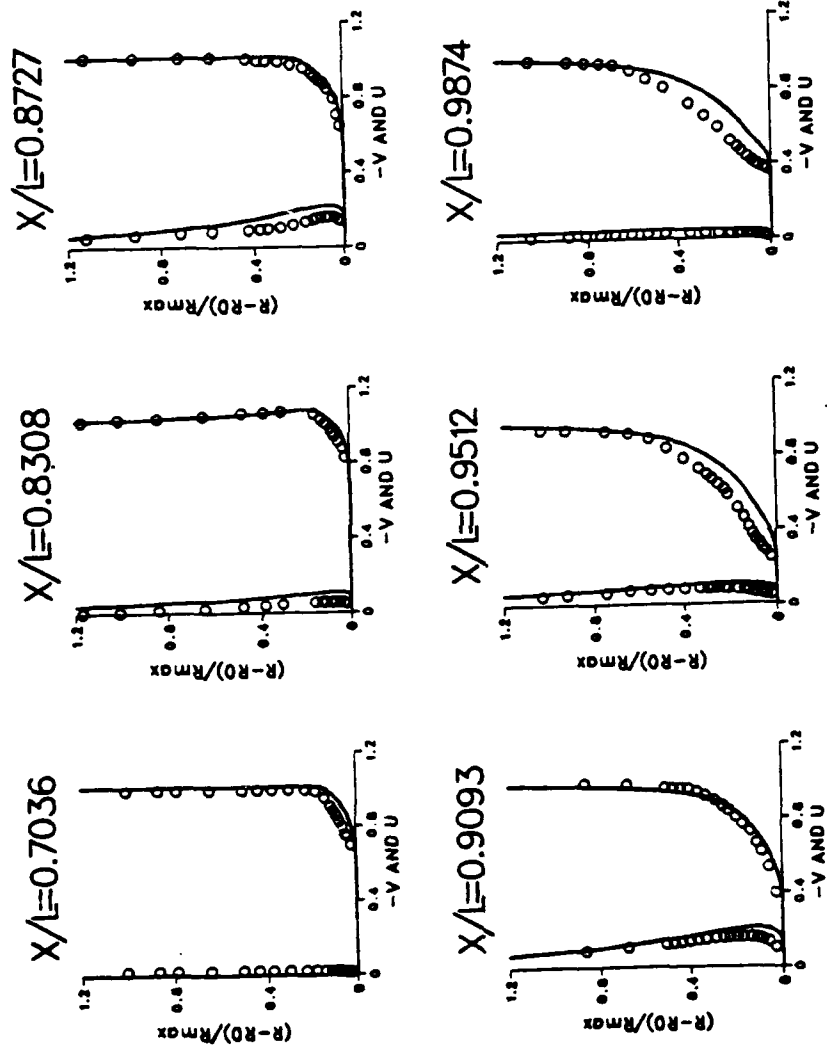


Fig.VII-15. Velocity Profiles ( Afterbody-5 )

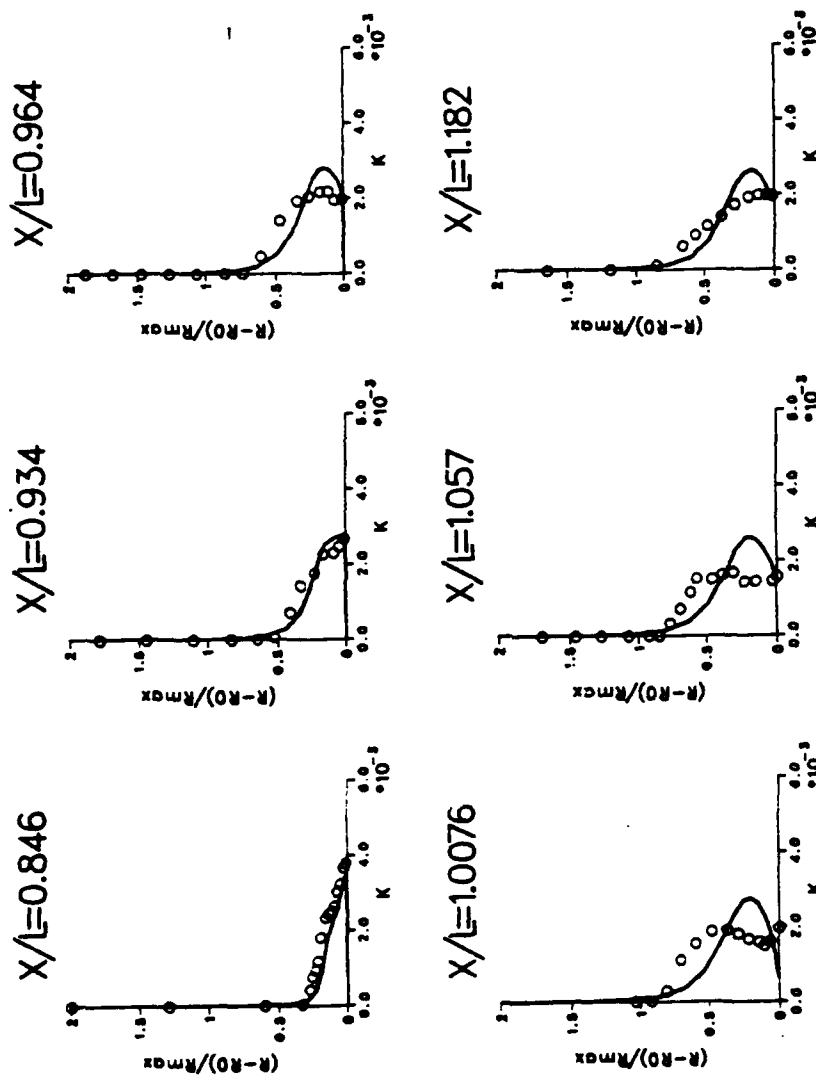


Fig. VII-16. Turbulent Kinetic Energy Profiles  
( Afterbody-1 )

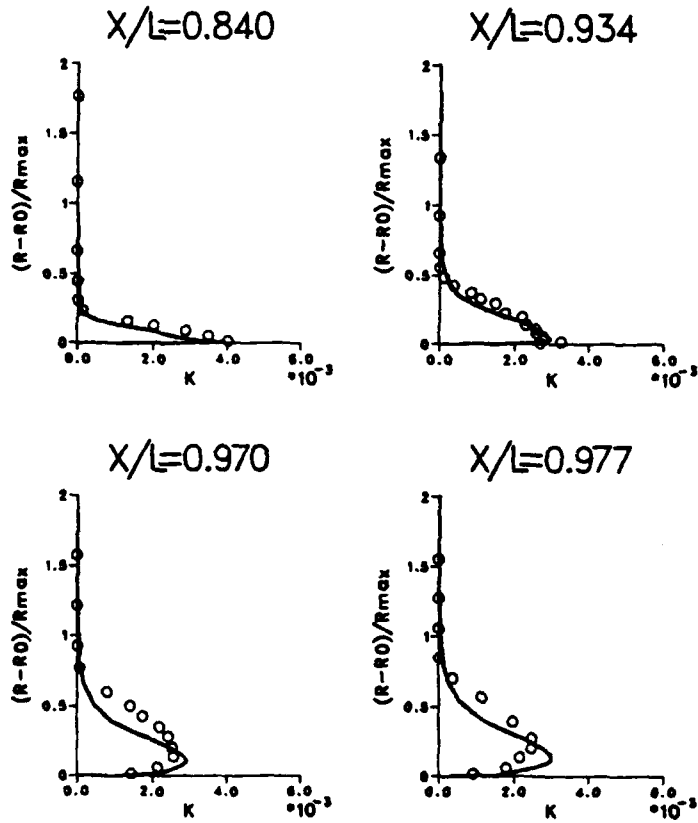


Fig.VII-17. Turbulent Kinetic Energy Profiles  
( Afterbody-2 )

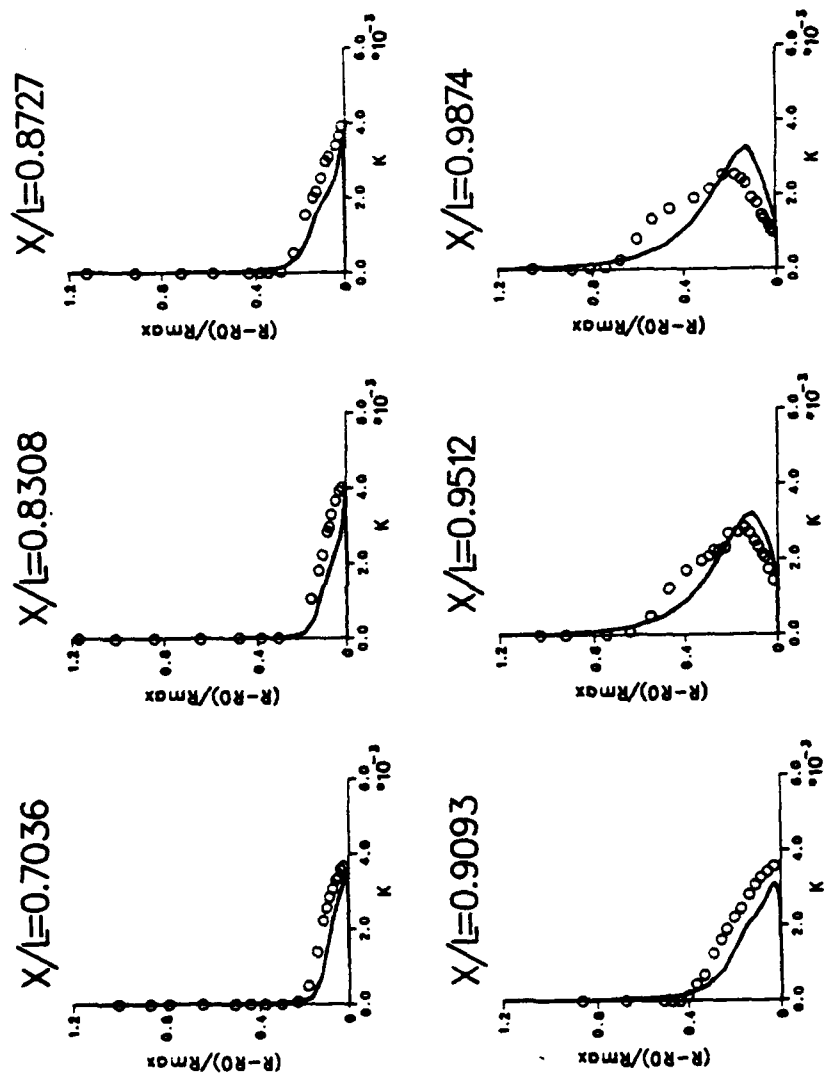


Fig. VII-18. Turbulent Kinetic Energy Profiles  
( Afterbody-5 )

## REFERENCES

1. S. V. Patankar and D. B. Spalding, " A Calculation Procedure for Heat, Mass Transfer and Momentum Transfer in Three-Dimensional Parabolic Flows," International Journal of Heat and Mass Transfer, Vol. 15, pp. 1787-1806, 1972.
2. S. V. Patankar, Numerical Heat Transfer and Fluid Flow, McGraw-Hill, New York, 1980.
3. C. J. Chen and H. C. Chen, "Finite Analytic Numerical Method for Unsteady Two Dimensional Navier-Stokes Equation," Journal of Computational Physics, Vol. 53, No.2, pp 209-226, 1984.
4. H. C. Chen and V. C. Patel, " Calculation of Trailing Edge, Stern and Wake Flows by a Time Marching Solution of the Partially Parabolic Equations," Iowa Institute of Hydraulic Research, Iowa City, IIHR Report No.285, 1985.
5. C. J. Chen and W. S. Cheng, " Finite Analytic Numerical Solutions of Incompressible Flow Past Inclined Axisymmetric Bodies," Iowa Institute of Hydraulic Research, Iowa City, IIHR Report No. 308, 1987.
6. K. M. Obasih and C. J. Chen, " Prediction of Laminar and Turbulent Flows Past Single and Twin Hydrofoils," Iowa Institute of Hydraulic Research, Iowa City, IIHR Report No. 317, 1987.
7. P. R. Eiseman, " A Multi-Surface Method of Coordinate Generation," Journal of Computational Physics, Vol. 30, PP. 118-150, 1979.
8. J. F. Thompson, Z.U.A Warsi and C. W. Mastin, " Boundary-Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations: A Reviews." Journal of Computational Physics, Vol. 47, PP. 1-108, 1982.

9. S. B. Pope, " The Calculation of Turbulent Recirculating Flows in General Orthogonal Coordinates," Journal of Computational Physics, Vol. 26, PP. 197-217, 1978.
10. A. D. Gosman and C. W. Rapley, " Fully Developed Flow in Passages of Arbitrary Cross-Section," in Recent Advances in Numerical Methods in Fluids, Taylor, C. and K. Morgan, Eds., pineridge Press, Swansea, 1980.
11. G. D. Raithby, F. P. Galpin and J. P. Van Doormaal, " Prediction of Heat and Fluid Flow in Complex Geometries Using General Orthogonal Coordinates," Numerical Heat Transfer, Vol.9, pp. 125-142, 1986.
12. C. F. Hsu, " A Curvilinear-Coordinate Method for Momentum, Heat and Mass transfer in Domains of Irregular Geometry," Ph.D Thesis, University of Minnesota, 1981.
13. S. Majumdar, " Role of Underrelaxation in Momentum Interpolation for Calculation of Flow with Nonstaggered Grids," Numerical Heat Transfer, Vol. 13, pp. 125-132, 1988.
14. T. F. Miller and F. W. Schmidt, " Use of A Pressure-Weighted Interpolation Method for the Solution of the Incompressible Navier-Stokes Equations on A Nonstaggered Grid System," Numerical Heat Transfer, Vol. 14, pp. 213-233, 1988.
15. M. Peric, R. Kessler and G. Scheuerer, " Comparision of Finite Volume Numerical Methods with Staggered and Collocated Grids," Computer and Fluids, Vol. 16, No. 4, pp. 389-403, 1988.
16. C. M. Rhie and W. L. Chow, " Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation," AIAA Journal, Vol. 21, No. 11, pp. 1525-1532, 1983.
17. S. Majumdar, W. Rodi and S. P. Vanka, " On the Use of Non-Staggered Pressure-Velocity Arrangement for Numerical Solution of Incompressible Flows, Rept. SFB 210/T/35, University of Karlsruhe, 1987.

18. M. Peric, " A Finite Volume Method for the Prediction of Three Dimensional Fluid Flow in Complex Ducts," Ph.d Thesis, University of London, 1985.
19. S. Majumar, " Development of A Finite Volume Procedure for Prediction of Fluid Flow Problems with Complex Irregular Boundaries," Rept. SFB 210/T/29, University of Karlsruhe, 1986.
20. C. M. Rhie, " A Pressure Based Navier-Stokes Solver Using the Multigrid Method," AIAA 24th Aerospace Science Meeting, Reno Nevada, 1986 AIAA-86-0207.
21. C. M. Rhie and S. T. Stowers, " Navier-Stokes Analysis for High Speed Flows Using a Pressure Correction Algorithm," AIAA/SAE/ASME/ASEE 23rd Joint Propulsion Conference, San Diago, California, 1987, AIAA-87-1980.
22. T. Han, " A Navier-Stokes Analysis of Three Dimensional Turbulent Flow around A Bluff Body in Ground Proximity," 1st National Fluid Dynamics Congress, Cincinnati, Ohio, 1988, AIAA-88-3766-CP.
23. W. Rodi, S. Majumdar and B. Schonung, " Finite Volume Methods for Two-Dimensional Incompressible Flows with Complex Boundaries," Proceedings of the Eighth International Conference on Computing Methods in Applied Science and Engineering, Volume 1, pp. 183-210, Versailles, France, 1987.
24. W. Shyy, S. S. Tong and S. M. Correa, " Numerical Recirculating Flow calculation Using a Body-Fitted Coordinate System," Numerical heat transfer, Vol.8, pp. 99-113, 1985.
25. M. Braaten and W. Shyy, " A Study of Recirculating Flow Computation Using the Boddy-Fitted Coordinates : Consistency Aspects and Mesh Skewness," Numerical Heat Transfer, Vol.9, pp.559-574, 1986.
26. A. Nakayama, " A Finite Difference Calculation Procedure for Three-Dimensional Turbulent Separated Flows," International Journal for Numerical Methods in Engineering, Vol. 20, pp. 1247-1260, 1984.



27. C. Hah, " A Navier-Stokes Analysis of Three-Dimensional Turbulent Flows Inside Turbine Blade Rows at Design and Off-Design conditions," Journal of Engineering for Gas Turbine and Power, vol. 106, pp. 383-390, 1984.
28. V. C. Patel, H. C. Chen and S. Ju, " Ship Stern and Wake Flows: Solutions of the Fully-Elliptic Reynolds-Averaged Navier-Stokes Equations and Comparisions with Experiments," Iowa Institute of Hydraulic Research, Iowa City, IIHR Rept. 323. 1988.
29. S. K. Choi and C. J. Chen, " Finite Analytic Numerical Solution of Turbulent Flow Past Axisymmetric Bodies by Zonal Modelling Approach," Advances and Applications in Computational Fluid Mechanics, 1988 Winter Annual Meeting of ASME, Chicago, Illinois, 1988.
30. C. R. Maliska and G. D. Raithby, " A Method for Computing Three-Dimensional Flows Using Non-Orthogonal Boundary-Fitted Coordinates," International Journal for Numerical Methods in Fluids, Vol. 4, pp. 519-537, 1984.
31. K. C. Karaki and S. V. Patankar, " Calculation Procedure for Viscous Incompressible Flows in Complex Geometries," Numerical Heat transfer, Vol. 14, pp. 295-307, 1988.
32. K. C. Karaki and S. V. Patankar, " Solution of Some Two-Dimensional Incompressible Flow Problems Using A Curvilinear Coordinate System Based Calculation Procedure," Numerical Heat transfer, Vol. 14, pp. 309-321, 1988.
33. K. C. Karaki and S. V. Patankar, " A Pressure Based Calculation Procedure for Viscous Flows at All Speeds in Arbitrary Configurations," AIAA 26th Aerospace Science Meeting, Reno, Nevada, 1988, AIAA-88-0058.
34. K. C. Karaki, " A Calculation Procedure for Viscous flows at All Speeds in Complex Geometries," Ph.D Thesis, University of Minnesota, Minneapolis, 1986.

35. I. Demirdzic, A. D. Gosman and R. I. Issa, " A Finite Volume Method for the Prediction of Turbulent Flow in Arbitrary Geometries," Lecture Notes in Physics, Vol. 141, Springer-Verlag, pp. 144-150, 1980.
36. I. Demirdzic, " A Finite Volume Method for Computation of Fluid Flow in Complex Geometries," Ph.D Thesis, University of London, 1982.
37. T. Gal-Chen and R. C. J. Somerville, " Numerical Solution of the Navier-Stokes Equations with Topography," Journal of Computational Physics, Vol. 17, pp. 276-310, 1975.
38. F. Stern, S. Y. Yoo and V. C. Patel, " Viscous-Inviscid Interaction with Higher-Order Viscous-Flow Equations," Iowa Institute of Hydraulic Research, Iowa City, IIHR Rept. 304, 1986.
39. M. C. Richmond, " Surface Curvature and Pressure Gradient Effects on Turbulent Flows: An Assessment Based on Numerical Solution of the Reynolds Equations," Ph.D Thesis, The University of Iowa, Iowa City, Iowa, 1987.
40. M. Faghri, E. M. Sparrow and A. T. Prata, " Finite Difference Solutions of Convection-Diffusion Problems in Irregular Domains Using a Non-Orthogonal Coordinate Transformation," Numerical Heat transfer, Vol. 7, pp. 183-109, 1984.
41. S. P. Vanka, B. C.- J. Chen and W. T. Sha, " A Semi-Implicit Calculation Procedure for Flow Described in Body-Fitted Coordinate system," Numerical Heat Transfer, Vol. 3, pp.1-19, 1980.
42. L. Prandtl, " Proc. 2nd. Intern. Congr. Appl. Mech. pp. 62-75, Zurich, 1926.
43. B. E. Launder, D. B. Spalding, " The Numerical Calculation of Turbulent Flows," Computer Methods in Applied Mechanics and Engineering, Vol. 3, No. 2, pp 269-289, 1974.

44. C. J. Chen, " Prediction of Turbulent Flows," Central Research Institute of Electric Power Industry, Abiko City, Chibu, Japan, 1983.
45. T. Cebeci and A. M. O. Smith, Analysis of Turbulent Boundary Layers, Academic Press, 1974.
46. B. S. Baldwin and H. Lomax, " Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA-78-257, 1978.
47. M. Wolfshtein, " The Velocity and Temperature Distribution in One-Dimensional Flow with Turbulence Augmentation and Pressure Gradient," International Journal of Heat and Mass Transfer, Vol. 12, pp. 301-318, 1969.
48. M. M. Gibson, D. B. Spalding, and W. Ziner, " Boundary-Layer Calculations Using the Hassid-Poreh One-Equation Energy Model," Letters in Heat and Mass Transfer, Vol. 5, pp. 73-80, 1978.
49. L. H. Norris and W. C. Reynolds, " Turbulent Channel Flow with a Moving Wavy Boundary," Rept. No. FM-10, Stanford University, Dept. Mech. Eng. Stanford, CA, 1975.
50. V. C. Patel, W. Rodi and G. Scheuerer, " Turbulence Models for Near-Wall and Low Reynolds Number Flows: A Review," AIAA Journal, Vol. 23, No. 9, pp. 1308-1319, 1985.
51. B. I. Sharma and B. E. Launder, " Application of the Energy Dissipation Model of Turbulence to the Calculation of Flow Near a Spinning Disc," Letters in Heat and Transfer, Vol. 1, pp. 131-138, 1974.
52. C. K. G. Lam and K. Bremhorst, " Modified Form of the  $k$ - $\epsilon$  Model for Predicting Wall Turbulence," Journal of Fluids Engineering, Vol. 103, pp.456-460, 1981.
53. K.- Y. Chien, " Predictions of Channel and Boundary-Layer Flows with Low-Reynolds-Number Model," AIAA Journal, Vol. 20, pp. 33-38, 1982.

54. W. Rodi and G. Scheuerer, "Scrutinizing the  $k-\epsilon$  Turbulence Model under the Adverse Pressure Gradient Conditions," ASME, Journal of Fluids Engineering, Vol. 108, pp. 174-179, 1986.
55. S. Majumdar and W. Rodi, "Numerical Calculation of Turbulent Flow Past Circular Cylinders," Proc. 3rd Symp. on Numerical and Physical Aspects of Aerodynamic Flows, Long Beach, California, 1985.
56. H. C. Chen and V. C. Patel, "Near-Wall Turbulence Models for Complex Flows Including Separation," AIAA Journal, Vol. 26, No. 6, pp.641-648, 1988.
57. H. C. Chen and V. C. Patel, "The Wake of An Axisymmetric Body with or without Tail Separation," 6th Symposium on Turbulent Shear Flows, Toulouse, France, 1987.
58. C. Yap, "Turbulent Heat and Momentum Transfer in Recirculating and Imprinting Flows," Ph.D Thesis, Faculty of Technology, The University of Manchester, 1987.
59. V. C. Patel and M. C. Richmond, "Pressure gradient and Surface Curvature Effects in Turbulent Boundary Layer," AIAA 19th Fluid Dynamics, Plasma Dynamics and Lasers Conference, Honolulu, Hawaii, 1987, AIAA-87-1301.
60. H. Iacovides and B. E. Launder, "Turbulent Momentum and Heat Transport in Square-Sectioned Ducts Rotating in Orthogonal Mode," Numerical Heat Transfer, Vol. 12, pp. 475-491, 1981.
61. O. K. Kwon and R. H. Pletcher, "Prediction of the Incompressible Flow over A Reward Facing Step," HTL-26, CFD-4, ISU-ERI-AMES-82019, Engineering Research Institute, Iowa State University, 1981.
62. I. E. Alber, "Turbulent Wake of A Thin Flat Plate," AIAA Journal, Vol. 18, pp. 1044-1051, 1980.
63. R. Chevray and L.S.G. Kovasznay, "Turbulent Measurement in the Wake of Thin Flat Plate," AIAA Journal, Vol. 7, No. 9, pp. 1641-1643, 1969.

64. B. R. Ramaprian, V. C. Patel, M. S. Sastry, " Turbulent Wake Development Behind Streamlined Bodies," Iowa Institute of Hydraulic Research, Iowa City, IIHR Report. 231, 1981.
65. P. J. Pot, " Measurement in a 2-D Wake and in A Wake Merging into A Boundary Layer," Data Report, NLR TR-79063 U, 1979.
66. J. Andreopoulos and P. Bradshaw, " Measurements of Interacting Turbulent Shear Layers in the Near Wake of A Flat Plate," Journal of Fluid Mechanics, Vol. 100, Part.3, pp. 639-668, 1980.
67. W. Rodi, " The Prediction of Free Turbulent Boundary Layers by Use of A Two-Equation Model of Turbulence," Ph.D Thesis, University of London, 1972.
68. B. E. Launder, A. Morse, W. Rodi, and D. B. Spalding, " Prediction of Free Shear Flows - A Comparision of the Performance of Six Turbulence Models," NASA-SP-321, 1973.
69. V. C. Patel and G. Scheuerer, " Calculation of Two-Dimensional Near and Far Wakes," AIAA Journal, Vol. 20, No. 7 pp. 900-907, 1982.
70. V. C. Patel and H. C. Chen, " Turbulent Wake of A Flat Plate," AIAA Journal, Vol. 25, No. 8, pp. 1078-1085, 1987.
71. H. Schlichting, Boundary Layer Theory, McGraw-Hill, New York, 1960.
72. A. M. Abbdelmeguid, N. C. Markatos, K. Muraoka, D. B. Spalding, " A Comparison between the Parabolic and Partially Parabolic Solution Procedures for Three-Dimensional Turbulent Flows around Ship Hulls," Appl. Math. Modelling, Vol. 3, pp. 249-258, 1979.
73. H. Iacovides and B. E. Launder, " PSL - An Economic Approach to the Numerical Analysis of Near-Wall, Elliptic Flow," ASME Journal of Fluids Engineering, Vol. 106, No. 2, pp. 241-242, 1984.

74. W. Rodi, " A Review of Experimental Data of Uniform-Density Free Turbulent Boundary Layers," Studies in Convection, Vol. 1, Edited by B.E. Launder, Academic Press, New York, 1975.
75. C. J. Chen and K. Singh, " Prediction of Buoyant Free Shear Flows by  $k$ - $\epsilon$  Model Based on Two Turbulence Scale Concept," Proceedings of International Symposium of Buoyant Flows, pp. 26-36, Athens, Greece, Sept. 1-5, 1986.
76. K. Stewartson, " On the Flow Near the Trailing Edge of A Flat Plate II," *Mathematica*, Vol. 16, pp. 106-121, 1969.
77. A. Messiter, " Boundary-Layer Flow Near the Trailing Edge of A Flat Plate," *SIAM Journal of Applied Mathematics*, Vol. 18, pp. 241-257, 1970.
78. R. E. Melnik and R. Chow, " Asymptotic Theory of Two-Dimensional Trailing-Edge Flows," *Aerodynamic Analysis Requiring Advanced Computers*, NASA-SP-347, pp. 177-249, 1975.
79. A. E. P. Veldman and A. I. van de Vooren, " Drag of A Finite Flat Plate," Proceedings of the 4th International Conference on Numerical Fluid Dynamics, Boulder, Colorado.
80. H. Macdonald and W. R. Briley, " A Survey of Recent Work on Interacting Boundary Layer Theory for Flow with Separation," Proceedings of the 2nd Symposium on Numerical and Physical Aspects of Aerodynamic Flows, Long Beach, California, 1983.
81. H. C. Chen and V. C. Patel, " Laminar Flow at the Trailing Edge of A Flat Plate," *AIAA Journal*, Vol. 25, No. 7, 1987.
82. Y. N. Xu and C. J. Chen, " Finite Analytic Numerical Prediction of Laminar and Turbulent Flows Past a Thin Finite Flat Plate," Proceedings of the 20th Midwestern Mechanics Conference, Volume 14(b), pp.

- 519-524, Purdue University, West Lafayette, Indiana, 1987.
83. C. M. Rhie, " A Numerical Study of the Flow Past An Isolated Airfoil with Separation," Ph.D Thesis, University of Illinois at Urbana Champaign, 1981.
  84. S. C. R. Dennis and J. Dunwoody, " The Steady Flow of A Viscous Fluid Past A Flat Plate," Journal of Fluid Mechanics, Vol. 24, Part 3, pp. 577-595, 1966.
  85. S. J. Shamroth, H. J. Gibeling and H. McDonald, " A Navier-Stokes Solution for Laminar and Turbulent Flow through A Cascade of Airfoils," AIAA 13th Fluid and Plasma Dynamics Conference, Snowmass, Colorado, 1980.
  86. D. G. Lilley and D. L. Rhode, " A Computer Code for Swirling Turbulent Axisymmetric Recirculating Flows in Practical Isothermal Combustor Geometries," NASA-CP-3442, 1982.
  87. F. D. White, Viscous Flow, McGraw-Hill, 1974.
  88. V. C. Patel, 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.
  89. V. C. Patel and H. C. Chen, " Flow Over Tail and in Wake of Axisymmetric Bodies: Review of the State of the Art," Journal of Ship Research., Vol. 30, No. 3, pp. 201-214, 1986.
  90. R. Chevray, " The Turbulent Wake of A Body of Revolution," ASME, Journal of Basic Engineering, Vol. 90, pp. 275-284, 1968.
  91. V. C. Patel and Y. T. Lee, " Thick Axisymmetric Turbulent Boundary Layer and Near wake of A Low-Drag Body of Revolution," Iowa Institute of Hydraulic Research, Iowa City, IIHR Report No.210, 1977.
  92. T. T. Huang, N. Santelli and G. Belt, " Stern Boundary Layer Flow on Axisymmetric Bodies" in

- Proceedings, 12th ONR Symposium on Naval Hydrodynamics, pp. 127-157, 1978.
93. T. T. Huang, N. C. Groves and G. Belt, " Boundary-Layer Flow on An Axisymmetric Body with An Inflected Stern," DTNSRDC Rept. 80/064, 1980.
  94. T. T. Huang, H. T. Wang, N. Santelli and N. C. Groves, " Propellor/Stern/Boundary-layer Interaction on Axisymmetric Bodies: Theory an Experiment," DTNSRDC Rept. 76-0113, 1976.
  95. G. H. Hoffman, " An Algebraic Turbulent Model Modified for Extra Rates of Strain in An Axisymmetric Boundary Layer," Applied Research Laboratory, Pennsylvania State University Technical Memorandum 82-201, State College, Pa., 1982.
  96. H. T. Wang and T. T. Huang, " Calculation of Potential Flow/Boundary-Layer Interaction on Axisymmetric Bodies," in Proceedings, ASME Symposium on Turbulent Boundary Layers, Niagara Falls, New York, pp. 47-57, 1979.
  97. K. Muraoka, " Calculation of Thick Boundary Layer and Wake of Ships by A Partially Parabolic Method," in Proceedings, 13th ONR Symposium on Naval Hydrodynamics, Tokyo, pp. 601-616, 1980.
  98. L-D. Zhou, " A Streamline Iteration Method for Calculating Turbulent Flow Around the Stern of A Body of Revolution and Its Wake," in Proceedings, 14th ONR Symposium on Naval Hydrodynamics, Ann Arbor, Mich., pp. 1041-1070, 1982.
  99. T. T. Huang and M-S. Chang, " Computation of Velocity and Pressure Variation Across Thick Turbulent Stern Flows," in Proceedings, 3rd Symposium on Numerical and Physical Aspects of Aerodynamic Flows, Long Beach, California, 1985.
  100. G. H. Hoffman, " A Modified Displacement Body Method for Testing the Axisymmetric Strong Interaction Problem," Journal of Ship Research, Vol. 24, No. 2, pp. 114-122, 1980.



101. M. S. Dietz, " An axisymmetric Strong-Interaction Procedure to Include Large, Normal Pressure Gradients," Applied Research Laboratory, Pennsylvania State University Technical Memorandum 80-160, State College, Pa., 1980.
102. N. C. Markatos, " The computation of Thick Axisymmetric Boundary Layers and Wakes Around Body of Revolution," in Proceedings, Institution of Mechanical Engineers, London, Vol. 198c, pp. 51-62, 1984.
103. K. Muraoka, " Calculation of Viscous Flow around Ships with Parabolic and Partially Parabolic Flows Solution Procedure," Trans. West Japan Soc. Naval Arch., Vol. 63, pp. 13-29, 1982.
104. K. Hanjalic and B. E. Launder, " Sensitizing the Dissipation Equation to Irrotational Strains," ASME, Journal of Fluids Engineering, Vol. 102, pp.34-40, 1980.
105. S. V. Ramakrishnan and S. G. Rubin, " Time-Consistent Pressure Relaxation Procedure for Compressible Reduced Navier-Stokes Equations," AIAA Journal, Vol. 25, No. 7, pp. 905-913, 1987.
106. H. C. Chen and V. C. Patel, " The Flow Around Wing-Body Junctions ". Proceedings of the Fourth Symposium on Numerical and Physical Aspects of Aerodynamic Flows, Long Beach, California, June, 1989.
107. J. J. Lee, " Heat and Mass Transfer Calculations for Recirculating Laminar and Turbulent Flows in An Abrupt Pipe Expansion," 1st National Fluid Dynamics Congress, Cincinnati, Ohio, 1988, AIAA-88-3791-CP.
108. C. C. Chieng and B. E. Lanunder, " On the Calculation of Turbulent Heat Transport Downstream from An Abrupt Pipe Expansion," Numerical Heat Transfer, Vol. 3, pp. 189-207, 1980.

APPENDIX  
COMPUTER CODES

## Computer Code for Chapter-VI

1. Wall function Method
  - 1-1) Geometric Data Files
  - 1-2) Computer Program for the Generation of Boundary-Fitted Coordinates
  - 1-3) Computer Program for Flow Calculation by Finite Analytic Method
  - 1-4) Computer Program for Flow Calculation by Finite Volume Method
  
2. Two-Layer Model
  - 2-1) Geometric Data Files
  - 2-2) Computer Program for the Generation of Boundary-Fitted Coordinates
  - 2-3) Computer Program for Flow Calculation by Finite Analytic Method



0.920000E+00 0.928000E+00 0.936000E+00 0.940000E+00  
0.922000E+00 0.960000E+00 0.968000E+00 0.976000E+00  
0.984000E+00 0.992000E+00 0.100000E+01 0.100800E+01  
0.101600E+01 0.102400E+01 0.103200E+01 0.104000E+01  
0.104800E+01 0.105600E+01 0.106400E+01 0.107200E+01  
0.108000E+01 0.109100E+01 0.110380E+01 0.111850E+01  
0.113536E+01 0.115477E+01 0.117708E+01 0.120276E+01  
0.123229E+01 0.126266E+01 0.130533E+01 0.135027E+01  
0.140197E+01 0.146144E+01 0.152984E+01 0.160852E+01  
0.169903E+01 0.180713E+01 0.192288E+01 0.206063E+01  
0.221908E+01 0.240133E+01 0.261098E+01 0.285213E+01  
0.312951E+01 0.34858E+01 0.381560E+01 0.423778E+01  
0.472338E+01 0.528198E+01 0.592451E+01 0.666359E+01  
0.751375E+01 0.849165E+01 0.961651E+01 0.109104E+02  
0.480000E-01 0.480000E-01 0.480000E-01 0.480000E-01  
0.480000E-01 0.480000E-01 0.480000E-01 0.480000E-01  
0.480000E-01 0.480000E-01 0.480000E-01 0.480000E-01  
0.480000E-01 0.480000E-01 0.480000E-01 0.480000E-01  
0.480000E-01 0.480000E-01 0.480000E-01 0.480000E-01  
0.473900E-01 0.466600E-01 0.456700E-01 0.443100E-01  
0.427100E-01 0.407200E-01 0.385000E-01 0.360100E-01  
0.334300E-01 0.306200E-01 0.277400E-01 0.248100E-01  
0.218300E-01 0.189800E-01 0.161300E-01 0.135100E-01  
0.110700E-01 0.885000E-02 0.711000E-02 0.595900E-02  
0.546200E-02 0.468700E-02 0.200000E-02 0.000000E+00  
0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00  
0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00  
0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00  
0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00  
0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00  
0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00  
0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00  
0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00  
0.000000E+00 0.000000E+00 0.000000E+00 0.000000E+00

6

```

C      READ(*,*) IBODY
C
C      REYNOLDS NUMBERS AND FRICTION VELOCITIES
C      AT THE MIDDLE OF THE BODY FROM EXPERIMENTAL DATA.
C
C      IF (IBODY.EQ.1) THEN
C      RE=6600000.D0
C      UTAU=0.04D0
C      IMA=11
C      ELSE IF (IBODY.EQ.2) THEN
C      RE=6800000.D0
C      U=0.040D0
C      IMA=21
C      ELSE IF (IBODY.EQ.5) THEN
C      RE=9300000.D0
C      UTAU=0.040D0
C      IMA=21
C      END IF
C
C      NUMBER OF GRID POINTS IN X-DIRECTION.
C
C      IMA=95
C      IPI=IMAX+1
C      INI=IMAX-1
C
C      TRAILING (M2), EDGE OF THE BODY.
C
C      INB=52
C
C      *****
C      READ BODY SHAPE AND X-COORDINATE.
C      *****
C
C      OPEN (UNIT=11, FILE=F1NAME(IBODY))
C      READ(11,1111) (X(I), I=1, IPI)
C      READ(11,1111) (RX(I), I=1, IPI)
C      FORMAT(4E14.6)
C      CLOSE(11)
C
C      *****
C      OPEN (UNIT=10, FILE=F2NAME(IBODY))
C      *****
C
C      SOME CONSTANTS AND RELAXATION FACTOR.
C
C      PI=3.141592653589793D0
C      EPS=0.001
C      RF=1.8
C
C      MAXIMUM NUMBER OF ITERATIONS.
C
C      ITMAX=200
C
C      NUMBER OF GRID IN Y-DIRECTION.
C
C      JMAX=34
C      JPI=JMAX+1
C      JNAM=JMAX-1

```

```

C      *****
C      NUMERICAL GENERATION OF BODY FITTED COORDINATES
C      * OF AXISYMMETRIC AND TWO DIMENSIONAL BODIES
C      * FOR LAMINAR AND TURBULENT FLOW CALCULATION.
C      *****
C
C      IMPLICIT REAL*8(A-H,O-Z)
C
C      CHARACTER*10 F1NAME(10)
C      CHARACTER*10 F2NAME(10)
C      CHARACTER*10 F3NAME(10)
C
C      COMMON/AAA/XC(200,70),YC(200,70)
C      COMMON/XXX/X(200),Y(200,70)
C      COMMON/ABC/AX(70),BY(70),CY(70),DY(70),TX(70)
C      COMMON/YYY/RX(200),RD(200),FR(200),DX1(70)
C      COMMON/BBB/AA(200),BB(200),CC(200),DD(200),A(200)
C      COMMON/AAA1/A11(200,70),A12(200,70),A22(200,70),AJ(200,70)
C      COMMON/AAA2/F1(200),F2(200,70)
C      COMMON/AAA3/FX(200,70),FY(200,70)
C
C      DATA F1NAME(1)/'ha1body',F1NAME(2)/'ha2body',
C      $ F1NAME(3)/'ha3body',F1NAME(4)/'ha4body',
C      $ F1NAME(5)/'ha5body',F1NAME(6)/'ha6body'
C
C      DATA F2NAME(1)/'halbout',F2NAME(2)/'ha2bout',
C      $ F2NAME(3)/'ha3bout',F2NAME(4)/'ha4bout',
C      $ F2NAME(5)/'ha5bout',F2NAME(6)/'ha6bout'
C
C      DATA F3NAME(1)/'halbdot',F3NAME(2)/'ha2bdot',
C      $ F3NAME(3)/'ha3bdot',F3NAME(4)/'ha4bdot',
C      $ F3NAME(5)/'ha5bdot',F3NAME(6)/'ha6bdot'
C
C      *****
C      * AXISYMMETRIC BODIES.
C      * *****
C      *
C      * IBODY=1 ; AFTERBODY1 OF DTNDSRDC (TURBULENT).
C      * IBODY=2 ; AFTERBODY2 OF DTNDSRDC (TURBULENT).
C      * IBODY=5 ; AFTERBODY5 OF DTNDSRDC (TURBULENT).
C      *
C      *****
C
C      *****
C      READ BODY TYPE FROM THE TERMINAL.
C      *****
C
C      WRITE(*,*) 'WHAT KIND OF BODY DO YOU WANT ?'
C      WRITE(*,*) 'TYPE 1 FOR AFTERBODY1 OF DTNDSRDC'
C      WRITE(*,*) 'TYPE 2 FOR AFTERBODY2 OF DTNDSRDC'
C      WRITE(*,*) 'TYPE 5 FOR AFTERBODY5 OF DTNDSRDC'
C
C

```

```

C C C NUMBER OF GRID IN Y-DIRECTION
C C C WHERE GRID IS ORTHOGONAL.
C C C NRP=19
C C C *****
C C C GENERATE Y-DIRECTIONAL GRID AT INLET.
C C C *****
C C C YPLUS=200.DO
C C C DY1(1)=0.
C C C DY1(2)=YPLUS/RE/UTAU
C C C DY1(3)=DY1(2)
C C C DO 10 J=4,JPI
C C C RATIO=1.20
C C C 10 DY1(J)=DY1(J-1)*RATIO
C C C
C C C Y(1,1)=RX(1)
C C C DO 82 J=2,JPI
C C C Y(1,J)=Y(1,J-1)+DY1(J)
C C C *****
C C C GENERATE INITIAL GRID IN THE WHOLE DOMAIN.
C C C *****
C C C DELTA=Y(1,NRP)
C C C
C C C DO 90 J=1,NRP
C C C FR(J)=Y(1,J)-RX(1)/(DELTA-RX(1))
C C C
C C C DO 91 I=1,IPI
C C C Y(I,1)=RX(I)
C C C Y(I,NRP)=DELTA
C C C 91 CONTINUE
C C C
C C C DO 750 I=1,IPI
C C C DO 751 J=2,NRP
C C C 751 Y(I,J)=Y(I,1)+FR(J)*(DELTA-Y(I,1))
C C C
C C C DO 750 J=NRP,JPI
C C C 750 Y(I,J)=Y(I,J)
C C C *****
C C C SPECIFY THE GRID CONTROL FUNCTIONS.
C C C *****
C C C DO 107 J=2,JMAX
C C C 1 F2(1,J)=Y(1,J+1)+Y(1,J-1)-2.*Y(1,J)
C C C 1 / (Y(1,J+1)+Y(1,J-1))
C C C 107 F2(IP1,J)=Y(IMB,J+1)+Y(IMB,J-1)-2.*Y(IMB,J)
C C C 1 / (Y(IMB,J+1)+Y(IMB,J-1))
C C C
C DO 110 J=2,JMAX
C DO 110 I=2,IMAX
C F2(I,J)=(F2(1,J)*(X(IMB)-X(I))+F2(IP1,J)*(X(I)-X(IMA)))/
C /X(IMB)-X(IMA))
C IF(I.GT.IMB) F2(1,J)=F2(IP1,J)
C IF(I.LT.IMA) F2(I,J)=F2(1,J)
C 110 CONTINUE
C
C DO 297 I=2,IMAX
C 297 F1(I)=(X(I+1)+X(I-1))-2.*X(I)/(X(I+1)-X(I-1))
C
C WRITE(10,2005)(Y(1,J),J=1,JPI)
C *****
C C START ITERATION PROCESS FOR GRID GENERATION.
C C *****
C C DO 300 IT=1,ITMAX
C C DRMAX=0.
C C
C DO 100 I=2,IMAX
C DO 100 J=2,JMAX
C XXI=0.5*(X(I+1)-X(I-1))
C XJI=0.5*(Y(I+1,J)-Y(I-1,J))
C XET=0.
C YET=0.5*(Y(I,J+1)-Y(I,J-1))
C G11=XXI*XXI+XJI*XJI
C G12=XXI*XET+XJI*YET
C G=(G11*G22-G12*G12)
C AJ(I,J)=DSORT(G)
C AI(I,J)=G22/G
C A22(I,J)=G11/G
C AI2(I,J)=-G12/G
C 100 CONTINUE
C C
C DO 500 I=2,IMAX
C AI=F1(I)
C IF(DABS(AI).LT.EPS) AI=EPS
C EPA=DEXP(AI)
C C
C DO 502 J=2,JMAX
C A2=F2(I,J)
C IF(DABS(A2).LT.EPS) A2=EPS
C EPB=DEXP(A2)
C SY=0.5*A12(I,J)*(Y(I+1,J+1)+Y(I-1,J-1))-Y(I+1,J-1)-Y(I-1,J+1)
C AY(J)=-2.*A22(I,J)*EPB+A2/(EPB-1./EPB)
C BY(J)=-2.*A11(I,J)*A1*(EPA+1./EPA)/(EPA-1./EPA)
C 1 +2.*A22(I,J)*A2*(EPB+1./EPB)/(EPB-1./EPB)
C CY(J)=-2.*A22(I,J)/EPB+A2/(EPB-1./EPB)
C DY(J)=-2.*A11(I,J)*A1/(EPA-1./EPA)*(Y(I-1,J)*EPA
C 2 +Y(I+1,J)/EPA)+SY
C 502 CONTINUE
C C
C DY(2)=DY(2)-AY(2)*Y(I,1)
C DY(JMAX)=DY(JMAX)-CY(JMAX)*Y(I,JPI)

```

```

A - 1
C
CALL TRIDAG(2, JMAX, AX, BY, CY, DY, TY)
DO 501 J=2, JMAX
YDIFF=DABS(TY(J)-Y(I,J))
IF(YDIFF.GT.DRMAX) DRMAX=YDIFF
Y(I,J)=Y(I,J)+RF*(TY(J)-Y(I,J))
501 CONTINUE
C
500 CONTINUE
C
C SPECIFY THE BOUNDARY CONDITIONS.
C
DO 130 J=1, JP1
Y(1,J)=Y(2,J)
Y(IP1,J)=Y(IMAX,J)
130 CONTINUE
C
C *****
C PRINT OUTPUT IN EACH ITERATION PROCESS.
C *****
WRITE(10,189) IT, DRMAX
189 FORMAT(5X, 'IT=', 15, 5X, 'DRMAX=', D12.4)
C
C PRINT 999, IT, DRMAX
999 FORMAT(5X, 'NO. OF ITERATION =', 15, 5X, 'DRMAX=', D12.4)
C
C 300 CONTINUE
C
C *****
C SPECIFY BOUNDARY CONDITIONS.
C *****
F1(1)=F1(2)
F1(IP1)=F1(IMAX)
C
DO 301 J=1, JP1
F2(1,J)=F2(2,J)
F2(IP1,J)=F2(IMAX,J)
301 CONTINUE
C
C REARRANGE DATA
C *****
DO 123 I=1, IP1
DO 123 J=1, JP1
FX(I,J)=F1(I)
FY(I,J)=F2(I,J)
XC(I,J)=X(I)
YC(I,J)=Y(I,J)
123 CONTINUE
C
C *****
C PRINT OUTPUT AFTER ITERATION PROCESS
C *****

```

```

DO 302 I=1, IP1
WRITE(10, 601) I, X(I)
601 FORMAT(5X, 'STATION ', 15, 5X, 'X =', F8.4)
WRITE(10, 2005) (Y(I, J), J=1, JP1)
302 CONTINUE
C
WRITE(10, 2007) (X(I), I=1, IP1)
2007 FORMAT(6F12.8)
C
DO 250 J=1, JP1
250 WRITE(10, 2005) (Y(I, J), I=1, IP1)
2005 FORMAT(4D14.6)
C
CLOSE(10)
C
C *****
C SAVE DATA FOR CALCULATION
C *****
C
OPEN(UNIT=11, FILE=FNAMEN(IBODY))
WRITE(11, 2009) (XC(I, J), I=1, IP1), (J=1, JP1)
WRITE(11, 2009) (YC(I, J), I=1, IP1), (J=1, JP1)
WRITE(11, 2009) (FX(I, J), I=1, IP1), (J=1, JP1)
WRITE(11, 2009) (FY(I, J), I=1, IP1), (J=1, JP1)
2009 FORMAT(6D13.6)
CLOSE(11)
C
STOP
END
C
SUBROUTINE TRIDAG(IF, L, A, B, C, D, V)
IMPLICIT REAL*(A-H, O-Z)
DIMENSION A(70), B(70), C(70), D(70), V(70), BETA(70), GAMMA(70)
BETA(IF)=B(IF)
GAMMA(IF)=D(IF)/BETA(IF)
IFFI=IF+1
DO 1 I=FFI, L
BETA(I)=B(I)-A(I)*C(I-1)/BETA(I-1)
V(I)=GAMMA(L)
LAST=L-IF
DO 2 K=1, LAST
I=L-K
V(I)=GAMMA(I)-C(I)*V(I+1)/BETA(I)
2 RETURN
END

```





```

COMMON/CONS4/CNU,CNU1,CEPS1,CEPS2,CEPK,CEPS,CKAR,E
COMMON/CONS5/RE,REI,TAU,TAUI
COMMON/CONS6/REF
COMMON/CONS7/IBODY
COMMON/CONS8/IFA
COMMON/BF1/B1U(ID,JD),B1V(ID,JD),B1W(ID,JD),B1X(ID,JD),B1Y(ID,JD)
COMMON/BF2/A1U(ID,JD),A1V(ID,JD),A1W(ID,JD),A1X(ID,JD),A1Y(ID,JD)
COMMON/BF3/B1V(ID,JD),B1W(ID,JD),B1X(ID,JD),B1Y(ID,JD),B1Z(ID,JD)
COMMON/BF4/A1V(ID,JD),A1W(ID,JD),A1X(ID,JD),A1Y(ID,JD),A1Z(ID,JD)
COMMON/BF5/B1P(ID,JD),B1Q(ID,JD),B1R(ID,JD),B1S(ID,JD),B1T(ID,JD)
COMMON/BF6/A1P(ID,JD),A1Q(ID,JD),A1R(ID,JD),A1S(ID,JD),A1T(ID,JD)
COMMON/BF7/B1C(ID,JD),B1D(ID,JD),B1E(ID,JD),B1F(ID,JD),B1G(ID,JD)
COMMON/PLOT/RV(ID,JD),RU(ID,JD),RP(ID,JD)

```

```

DATA F1NAME(1)/'ha1bdad',F1NAME(2)/'ha2bdad',
$ F1NAME(3)/'ha3bdad',F1NAME(4)/'ha4bdad',
$ F1NAME(5)/'ha5bdad',F1NAME(6)/'ha6bdad',
DATA F2NAME(1)/'hfaa101',F2NAME(2)/'hfaa201',
$ F2NAME(3)/'hfaa301',F2NAME(4)/'hfaa401',
$ F2NAME(5)/'hfaa501',F2NAME(6)/'hfaa601',
DATA F3NAME(1)/'hfaa102',F3NAME(2)/'hfaa202',
$ F3NAME(3)/'hfaa302',F3NAME(4)/'hfaa402',
$ F3NAME(5)/'hfaa502',F3NAME(6)/'hfaa602',
DATA F4NAME(1)/'hfaa103',F4NAME(2)/'hfaa203',
$ F4NAME(3)/'hfaa303',F4NAME(4)/'hfaa403',
$ F4NAME(5)/'hfaa503',F4NAME(6)/'hfaa603',
DATA F5NAME(1)/'hfaa104',F5NAME(2)/'hfaa204',
$ F5NAME(3)/'hfaa304',F5NAME(4)/'hfaa404',
$ F5NAME(5)/'hfaa504',F5NAME(6)/'hfaa604',

```

```

*****
* AXISYMMETRIC BODIES.
* *****
* IBODY=1 ; AFTERBODY1 OF DTNDSRDC (TURBULENT).
* IBODY=2 ; AFTERBODY2 OF DTNDSRDC (TURBULENT).
* IBODY=5 ; AFTERBODY5 OF DTNDSRDC (TURBULENT).
*****

```

```

*****
READ BODY TYPE FROM THE TERMINAL.
*****
WRITE(*,*) 'WHAT KIND OF BODY DO YOU WANT ?'
WRITE(*,*) 'TYPE 1 FOR AFTERBODY1 OF DTNDSRDC'
WRITE(*,*) 'TYPE 2 FOR AFTERBODY2 OF DTNDSRDC'
WRITE(*,*) 'TYPE 5 FOR AFTERBODY5 OF DTNDSRDC'
READ(*,*) IBODY

```

```

REYNOLDS NUMBERS AND FRICTION VELOCITIES
AT THE MIDDLE OF THE BODY FROM EXPERIMENTAL DATA.

```

```

IF (IBODY.EQ.1) THEN
RE=6600000.D0
ELSE IF (IBODY.EQ.2) THEN
RE=6800000.D0
ELSE IF (IBODY.EQ.5) THEN
RE=9300000.D0
END IF

```

```

*****
MAIN PROGRAM
*****
SPECIFY CONSTANTS.
*****
ELLIPTIC OR PARTIALLY PARABOLIC FORMULATION

```

```

IFA=1

```

```

GEOMETRIC CONSTANTS

```

```

IMAX=96
JMAX=35
IPI=IMAX+1
JPI=JMAX+1
M2=52
M2M1=M2-1

```

```

JAU=3
JAV=3
JAP=3
JA=3

```

```

JAU1=JAU-1
JAV1=JAV-1
JAP1=JAP-1
JAM1=JA-1

```

```

NUMBER OF ITERATIONS

```

```

IEND=200
ITERA=3
ITUV=3
ITKD=3
ITPP=5
ITM=1

```

```

NUMERIC CONSTANTS

```

```

EMAX=30.
EPE=0.0001
EPE1=0.00001

```

```

EPEW=0.0C01
PI=J.141592653589793D0
TURBULENCE MODEL CONSTANTS
CNU=0.09
CNU2=DSORT(CNU)
CNU4=DSORT(CNU2)
CNU3=CNU4+CNU4+CNU4
CNU2I=1./CNU2
CEPS1=1.44
CEPS2=1.92
CEPK=1.0
CEFS=1.3
CMAR=0.418
E=9.
C TIME STEP AND REYNOLDS NUMBER.
C
C REI=1./RE
C TAU=1.
C TAU1=1./TAU
C
C OPEN(UNIT=8, FILE=F2NAME(1BODY))
C OPEN(UNIT=7, FILE=F3NAME(1BODY))
C OPEN(UNIT=9, FILE=F4NAME(1BODY))
C OPEN(UNIT=10, FILE=F5NAME(1BODY))
C *****
C OFFSET OF NUMERICAL GRIDS
C XC, YC----- CONTROL VOLUME POINT
C XU, YU----- U CALCULATION POINT
C XV, YV----- V CALCULATION POINT
C XP, YP----- P, K, E CALCULATION POINT
C *****
C OPEN(UNIT=11, FILE=F1NAME(1BODY))
C
C READ(11, 2006) ((XC(I, J), I=1, IMAX), J=1, JMAX)
C READ(11, 2006) ((YC(I, J), I=1, IMAX), J=1, JMAX)
C READ(11, 2006) ((F1(I, J), I=1, IMAX), J=1, JMAX)
C READ(11, 2006) ((F2(I, J), I=1, IMAX), J=1, JMAX)
C 2006 FORMAT(6E13.6)
C CLOSE(11)
C
C XU(1, 1)=XC(1, 1)
C YU(1, 1)=YC(1, 1)
C XV(1, 1)=XC(1, 1)
C YV(1, 1)=YC(1, 1)
C XP(1, 1)=XC(1, 1)
C YP(1, 1)=YC(1, 1)
C
C XC(1, JPI)=XC(1, JMAX)
C YC(1, JPI)=YC(1, JMAX)+YC(1, JMAX)-YC(1, JMAX-1))
C DO 111 J=2, JPI

```

```

XU(1, J)=XC(1, J)
YU(1, J)=0.5*(YC(1, J)+YC(1, J-1))
XP(1, J)=XC(1, J)
YP(1, J)=0.5*(YC(1, J)+YC(1, J-1))
XV(1, J)=XC(1, J)
YV(1, J)=YC(1, J)
111 YV(1, J)=YC(1, J)
C
C DO 112 I=2, IMAX
C
C XU(I, 1)=XC(I, 1)
C XP(I, 1)=0.5*(XC(I, 1)+XC(I-1, 1))
C XV(I, 1)=0.5*(XC(I, 1)+XC(I-1, 1))
C YU(I, 1)=YC(I, 1)
C YP(I, 1)=0.5*(YC(I, 1)+YC(I-1, 1))
C YV(I, 1)=0.5*(YC(I, 1)+YC(I-1, 1))
C
C DO 112 J=2, JMAX
C
C XP(I, J)=0.25*(XC(I, J)+XC(I-1, J)+XC(I, J-1)+XC(I-1, J-1))
C XU(I, J)=0.5*(XC(I, J)+XC(I, J-1))
C XV(I, J)=0.5*(XC(I, J)+XC(I-1, J))
C YU(I, J)=0.5*(YC(I, J)+YC(I-1, J))
C YV(I, J)=0.5*(YC(I, J)+YC(I-1, J))
C 112 YP(I, J)=0.25*(YC(I, J)+YC(I-1, J)+YC(I, J-1)+YC(I-1, J-1))
C
C DO 113 I=1, IMAX
C
C YC(I, JPI)=YC(1, JPI)
C VV(I, JPI)=VV(1, JPI)
C YP(I, JPI)=YP(1, JPI)
C YU(I, JPI)=YU(1, JPI)
C XC(I, JPI)=XC(1, JMAX)
C XV(I, JPI)=XV(1, JMAX)
C XP(I, JPI)=XP(1, JMAX)
C 113 XU(I, JPI)=XU(1, JMAX)
C
C DO 114 J=1, JPI
C
C XC(IPI, J)=XC(IMAX, J)+0.5*(XC(IMAX, J)-XC(IMAX-1, J))
C XP(IPI, J)=XC(IPI, J)
C XV(IPI, J)=XC(IPI, J)
C YC(IPI, J)=XC(IPI, J)
C YP(IPI, J)=YC(IMAX, J)
C YU(IPI, J)=YP(IMAX, J)
C YV(IPI, J)=YV(IMAX, J)
C 114 YV(IPI, J)=YV(IMAX, J)
C
C DO 115 I=1, IPI
C
C YP2(I)=YP(I, 2)-YP(I, 1)
C YP3(I)=YP(I, 3)-YP(I, 1)
C YP4(I)=YP(I, 4)-YP(I, 1)
C YU2(I)=YU(I, 2)-YU(I, 1)
C YU3(I)=YU(I, 3)-YU(I, 1)
C YU4(I)=YU(I, 4)-YU(I, 1)
C YV2(I)=YV(I, 2)-YV(I, 1)
C YV3(I)=YV(I, 3)-YV(I, 1)
C YV4(I)=YV(I, 4)-YV(I, 1)
C 115 YV4(I)=YV(I, 4)-YV(I, 1)
C
C DO 116 I=1, IPI
C
C YP2SQ(I)=YP2(I)*YP2(I)
C YP3SQ(I)=YP3(I)*YP3(I)
C YP4SQ(I)=YP4(I)*YP4(I)
C YU2SQ(I)=YU2(I)*YU2(I)

```

```

YU3SQ(I)=YU3(I)*YU3(I)
YU4SQ(I)=YU4(I)*YU4(I)
116 CONTINUE
C
DO 117 I=2,IMAX
DO 117 J=2,JMAX
R=0.5*(YC(I,J)+YC(I,J-1))
YX1=0.25*(YC(I+1,J)+YC(I+1,J-1)-YC(I-1,J)-YC(I-1,J-1))
YET=YC(I,J)-YC(I,J-1)
XX1=0.25*(XC(I+1,J)+XC(I+1,J-1)-XC(I-1,J)-XC(I-1,J-1))
XET=0.
CALL GEOCOE
B11U(I,J)=B11
B12U(I,J)=B12
B21U(I,J)=B21
B22U(I,J)=B22
A11U(I,J)=A11
A12U(I,J)=A12
A21U(I,J)=A21
A22U(I,J)=A22
117 AJIU(I,J)=AJI
C
DO 118 I=2,IP1
DO 118 J=2,JMAX
R=0.5*(YC(I,J)+YC(I-1,J))
YX1=YC(I,J)-YC(I-1,J)
YET=0.25*(YC(I,J+1)+YC(I-1,J+1)-YC(I,J)-YC(I-1,J-1))
XX1=XC(I,J)-XC(I-1,J)
XET=0.
CALL GEOCOE
B11V(I,J)=B11
B12V(I,J)=B12
B21V(I,J)=B21
B22V(I,J)=B22
A11V(I,J)=A11
A12V(I,J)=A12
A21V(I,J)=A21
A22V(I,J)=A22
118 AJIV(I,J)=AJI
C
DO 119 I=2,IP1
DO 119 J=2,JMAX
R=0.25*(YC(I,J)+YC(I-1,J)+YC(I-1,J-1)+YC(I,J-1)+YC(I,J-1))
YX1=0.5*(YC(I,J)+YC(I,J-1)-YC(I-1,J)-YC(I-1,J-1))
YET=0.5*(YC(I,J)+YC(I-1,J)-YC(I,J-1)-YC(I-1,J-1))
XX1=0.5*(XC(I,J)+XC(I-1,J)-XC(I-1,J)-XC(I-1,J-1))
XET=0.
CALL GEOCOE
B11P(I,J)=B11
B12P(I,J)=B12
B21P(I,J)=B21
B22P(I,J)=B22
A11P(I,J)=A11
A12P(I,J)=A12
A21P(I,J)=A21
A22P(I,J)=A22
119 AJIP(I,J)=AJI
C
DO 120 I=2,IP1
DO 120 J=2,JP1
R=0.5*(YC(I,J)+YC(I,J-1))
B11C(I,J)=B11C(I,J)+YC(I,J-1)*R
B21C(I,J)=B21C(I,J)-XC(I,J)-XC(I,J-1)*R
R=0.5*(YC(I,J)+YC(I-1,J))
B12C(I,J)=B12C(I,J)-YC(I-1,J)*R
B22C(I,J)=B22C(I,J)-XC(I,J)-XC(I-1,J)*R
120 B22C(I,J)=XC(I,J)-XC(I-1,J)*R
C
*****
SET UP THE INITIAL CONDITIONS.
*****
DO 121 I=1,IP1
VSP(I)=0.
DO 121 J=1,JP1
BCU(I,J)=0.
BCV(I,J)=0.
TNU(I,J)=0.
PP(I,J)=0.
121 PR(I,J)=0.
C
*****
SET UP THE INLET CONDITION.
*****
UO(1,1)=0.
VTO(1,1)=0.
TDSO(1,1)=0.
UTAU(1)=0.038
DELTA=0.006
DELTAI=1./DELTA
DO 122 J=2,JP1
YY=YU(1,J)-YU(1,1)
ETA=YY*DELTAI
ETA2=(YU(1,2)-YU(1,1))*DELTAI
ETAM=(YU(1,JP1)-YU(1,1))*DELTAI
1/7 TH LAW
UO(1,J)=1.02*ETA**0.143
IF(ETA.GE.1.)UO(1,J)=1.02-0.02*(ETA-1.)/(ETAM-1.)
SPECIFY INLET U-VELOCITY PROFILE.
DPLUS=DELTA*RE*UTAU(1)
IF(DPLUS.LE.1000.D0) DPLUS=1001.D0
YPLUS=RE*YY*UTAU(1)
IF(YPLUS.LT.5.D0) THEN
UO(1,J)=UTAU(1)*YPLUS
ELSE IF(YPLUS.LT.50.D0) THEN
UO(1,J)=(5.D0*DLOG(YPLUS)-3.05D0)*UTAU(1)

```



```

C      DO 20 J=4,JPI
C      IF(TNU(I,J-1).LE.TNU(I,J-2).AND.TNU(I,J-1).LT.
1     TNU(I,J)) TNU(I,J)=TNU(I,J-1)
C      20 CONTINUE
C      IF(I.EQ.2) THEN
C      DO 21 J=1,JPI
C      21 TNU(1,J)=TNU(2,J)
C      END IF
C      IF(NITER.EQ.1) THEN
C      DO 22 J=1,JPI
C      22 TNU(I+1,J)=TNU(I,J)
C      END IF
C      *****
C      CALCULATES THE COEFFICIENTS FOR THE U-MOMENTUM EQUATION.
C      *****
C      DO 23 J=JAU,JMAX
C      TNUP=0.5*(TNU(I+1,J)+TNU(I,J))
C      REFF=1./(REI+TNUP)
C      CALL SOURCEV
C      UU=0.25*(UO(I,J)+UO(I,J+1)+UO(I-1,J)+UO(I-1,J+1))
C      VV=VO(I,J)
C      FX=-A11V(I,J)*(F1(I,J)+F1(I-1,J))
C      FY=-A22V(I,J)*(F2(I,J)+F2(I-1,J))+1./B11V(I,J)
C      ER=DSORT(A11V(I,J))
C      FR=DSORT(A22V(I,J))
C      AR=0.5*(REFF*AJIV(I,J)*(B11V(I,J)+UO+B21V(I,J)+VV)-FX)
C      BR=0.5*(REFF*AJIU(I,J)*(B12V(I,J)+UU+B22V(I,J)+VV)-FY)
C      DR=2.*AR
C      IF(IFA.EQ.2) THEN
C      CALL COEFF2
C      ELSE
C      CALL COEFF3
C      END IF
C      UWB(I,J)=CF(1,1)
C      UWC(I,J)=CF(1,2)
C      UWT(I,J)=CF(1,3)
C      UPB(I,J)=CF(2,1)
C      UPC(I,J)=CF(2,2)
C      UPT(I,J)=CF(2,3)
C      UPB(I,J)=CF(3,1)
C      UPC(I,J)=CF(3,2)
C      UPT(I,J)=CF(3,3)
C      IF(IFA.EQ.2) THEN
C      D2(I,J)=VWC(I,J)
C      ELSE
C      D2(I,J)=0.
C      END IF
C      E2(I,J)=REFF*TAUI
C      D=REFF*VPC(I,J)/(1.+D2(I,J)+E2(I,J)+VPC(I,J))
C      CV(I,J)=D/(YP(I,J+1)-YP(I,J))
C      BCV(I,J)=B22C(I,J)+CV(I,J)
C      25 CONTINUE

```

```

C      DO 20 J=4,JPI
C      IF(TNU(I,J-1).LE.TNU(I,J-2).AND.TNU(I,J-1).LT.
1     TNU(I,J)) TNU(I,J)=TNU(I,J-1)
C      20 CONTINUE
C      IF(I.EQ.2) THEN
C      DO 21 J=1,JPI
C      21 TNU(1,J)=TNU(2,J)
C      END IF
C      IF(NITER.EQ.1) THEN
C      DO 22 J=1,JPI
C      22 TNU(I+1,J)=TNU(I,J)
C      END IF
C      *****
C      CALCULATES THE COEFFICIENTS FOR THE U-MOMENTUM EQUATION.
C      *****
C      DO 23 J=JAU,JMAX
C      TNUP=0.5*(TNU(I+1,J)+TNU(I,J))
C      REFF=1./(REI+TNUP)
C      CALL SOURCEV
C      UU=UO(I,J)
C      VV=0.25*(VO(I,J-1)+VO(I+1,J-1)
1     +VO(I,J)+VO(I+1,J))
C      FX=-2.*A11U(I,J)*F1(I,J)
C      FY=-A22U(I,J)*(F2(I,J)+F2(I,J-1))+1./B11U(I,J)
C      ER=DSORT(A11U(I,J))
C      FR=DSORT(A22U(I,J))
C      AR=0.5*(REFF*AJIU(I,J)*(B11U(I,J)+UU+B21U(I,J)+VV)-FX)
C      BR=0.5*(REFF*AJIU(I,J)*(B12U(I,J)+UU+B22U(I,J)+VV)-FY)
C      DR=2.*AR
C      IF(IFA.EQ.2) THEN
C      CALL COEFF2
C      ELSE
C      CALL COEFF3
C      END IF
C      UWB(I,J)=CF(1,1)
C      UWC(I,J)=CF(1,2)
C      UWT(I,J)=CF(1,3)
C      UPB(I,J)=CF(2,1)
C      UPC(I,J)=CF(2,2)
C      UPT(I,J)=CF(2,3)
C      UPB(I,J)=CF(3,1)
C      UPC(I,J)=CF(3,2)
C      UPT(I,J)=CF(3,3)
C      IF(IFA.EQ.2) THEN
C      D1(I,J)=UWC(I,J)
C      ELSE
C      D1(I,J)=0.
C      END IF
C      E1(I,J)=REFF*TAUI

```

```

C BCV(I,JPI)=0.
C BCV(I,JAVM1)=0.
C
C US(I,JPI)=1.DO
C U(I,JPI)=1.DO
C UH(I,JPI)=1.DO
C
C *****
C START INNER ITERATIONS
C *****
C DO 777 ITA=1,ITERA
C *****
C TWO POINT WALL FUNCTION METHOD.
C *****88
C IF(I.GE.M2) GO TO 1000
C
C IF(I.EQ.M2M1) THEN
C R=10000.
C XXI=2.*(YC(I,1)-YP(I,1))
C YET=YC(I,2)-YC(I,1)
C XET=0.
C ELSE
C R=YC(I,1)
C YAI=YP(I+1,1)-YP(I,1)
C XXI=XP(I+1,1)-XP(I,1)
C YET=YC(I,2)-YC(I,1)
C XET=0.
C END IF
C
C U3=U(I,3)
C V3=0.25*(V(I,2)+V(I+1,2)
C +V(I,3)+V(I+1,3))
C CALL GEOCOE
C DG11=DSORT(G11)
C COSA=(B12*XET+B22*YET)/DSORT(G22+G+A22)
C GRADP=(PR(I+1,1)-PR(I,1))/(XP(I+1,1)-XP(I,1))*COSA
C IF(I.GE.M2.OR.NITER.LT.5) GRADP=0.
C Q3=DSORT(U3*U3+V3*V3)
C QXI=XI*DG11*(B11*U3+B21*V3)
C UTAU=UTAU(I)
C IF(NITER.EQ.1) UTAU=UTAU(I-1)
C
C DO 555 IJK=1,50
C DPR=GRADP/(RE*UTAU*UTAU*UTAU)
C DPR=DMAX1(DPR,EPEW)
C DTAU=0.5*DPR
C SORT3=DSQRT(1.+DTAU*RE*UTAU*YU3(I)*COSA)
C UTAUN=Q3/((DLOG(4.*(SQRT3-1.)/(SQRT3+1.)/DTAU)
C +2.*SQRT3-2.)/CKAR+5.45+3.7*DPR)
C 1 IF(DABS(UTAU-UTAU).LT.1.0D-5) GO TO 556
C UTAU=UTAU

```

```

555 CONTINUE
C 556 TAUW=2.*UTAU*UTAU
C UTAUA(I)=UTAU
C YPP=RE*UTAU*YU2(I)*COSA
C SQRT2=DSQRT(1.+DTAU*YPP)
C Q2=UTAU*((DLOG(4.*(SQRT2-1.)/(SQRT2+1.)/DTAU)
C +2.*SQRT2-2.)/CKAR+5.45+3.7*DPR) -
C Q2XI=Q3XI*Q2/Q3
C U(I,2)=Q2XI*XXI/DG11
C VSF(I)=Q2XI*YXI/DG11
C FC(I)=TAUW
C UTAUK=0.5*(UTAU(I-1)+UTAU)
C
C TKE(I,2)=UTAU*UTAU*CN2I
C TDS(I,2)=UTAU*UTAU*UTAU/(CKAR*YF2(I)*COSA)
C V2=0.5*(VSF(I)+VSF(I-1))
C HN=0.5*(YV3(I)-YV2(I))
C HS=0.5*(YV2(I)-YV2(I))
C V(I,2)=(HN*V2+HS*V0(I,3))/(HN+HS)
C US(I,2)=U(I,2)
C VS(I,2)=V(I,2)
C
C 1000 CONTINUE
C
C *****
C CALCULATE U-STAR VELOCITY.
C *****
C
C DO 26 JUV=1,ITUV
C DO 27 J=JAU,JMAX
C AA(J)=UPB(I,J)
C BB(J)=1.+D1(I,J)+E1(I,J)*UPC(I,J)
C CC(J)=UPT(I,J)
C DD(J)=UEC(I,J)*US(I+1,J)+UEB(I,J)*US(I+1,J-1)
C 1 +UET(I,J)*US(I+1,J+1)+UMC(I,J)*US(I-1,J)
C 2 +UWB(I,J)*US(I-1,J-1)+UWT(I,J)*US(I-1,J+1)
C 3 +UPC(I,J)*(E1(I,J)*UO(I,J)+SU(I,J))
C 4 -BB(J)*CUY(I,J)*(PR(I+1,J)-PR(I,J))
C 5 -BB(J)*CUY(I,J)*0.25*(PR(I+1,J+1)+PR(I,J-1)
C 6 -PR(I,J-1)-PR(I+1,J-1))
C DD(JAU)=DD(JAU)-AA(JAU)*US(I,JAU)
C DD(JMAX)=DD(JMAX)-CC(JMAX)*US(I,JPI)
C CALL TRIDAG(JAU,JMAX,AA,BB,CC,DD,T)
C DO 28 J=JAU,JMAX
C US(I,J)=T(J)
C IF(I.LT.M2) GO TO 26
C US(I,JAU-1)=(YU4SQ(I)-YU2SQ(I))*US(I,JAU)
C 1 -(YU3SQ(I)-YU2SQ(I))*US(I,JAU+1)
C 2 /YU4SQ(I)-YU3SQ(I)
C US(I,JAU-2)=(YU4SQ(I)*US(I,JAU)-YU3SQ(I)*US(I,JAU+1)
C 1 /YU4SQ(I)-YU3SQ(I))
C
C 26 CONTINUE
C
C *****
C CALCULATE V-STAR VELOCITY.
C *****

```

```

C
DO 30 IUV=1,ITUV
AA(J)=VPB(I,J)
BB(J)=1.+D2(I,J)+E2(I,J)*VPC(I,J)
CC(J)=VPT(I,J)
31 DD(J)=VEC(I,J)*VS(I+1,J)+VEB(I,J)*VS(I+1,J-1)
+VET(I,J)*VS(I+1,J+1)+VWC(I,J)*VS(I-1,J)
+VWB(I,J)*VS(I-1,J-1)+VWT(I,J)*VS(I-1,J+1)
+VPC(I,J)*E2(I,J)+V0(I,J)+SV(I,J)
- BB(J)*CV(I,J)+PR(I,J+1)-PR(I,J)
DD(JAV)=DD(JAV)-AA(JAV)*VS(I,JAV1)
DD(JMAX)=DD(JMAX)-CC(JMAX)*VS(I,JPI)
CALL TRIDAG(JAV,JMAX,AA,BB,CC,DD,T)
DO 32 J=JAV,JMAX
VS(I,J)=T(J)
32 VS(I,JPI)=VS(I,JMAX)*B22C(I,JMAX)/B22C(I,JPI)
IF(I.LT.N2) GO TO 30
VS(I,2)=VS(I,3)*V2(I)/V3(I)
VS(I,1)=0.
30 CONTINUE
C
C
C *****
C CALCULATE MASS SOURCE FOR PRESSURE CORRECTION EQUATION.
C *****
DO 35 J=JAP,JMAX
VE=0.25*(V(I,J)+V(I,J-1)+V(I+1,J)+V(I+1,J-1))
VM=0.25*(V(I-1,J)+V(I-1,J-1)+V(I,J)+V(I,J-1))
UN=0.25*(U(I,J)+U(I-1,J)+U(I,J+1)+U(I-1,J+1))
UB=0.25*(U(I,J)+U(I-1,J)+U(I,J-1)+U(I-1,J-1))
DF(I,J)=B21C(I,J)*VE-B21C(I-1,J)*VM
+ B12C(I,J)*UN-B12C(I,J-1)*UB
DS(I,J)=B11C(I,J)*US(I,J)-B11C(I-1,J)*US(I-1,J)
+ B22C(I,J)*VS(I,J)-B22C(I,J-1)*VS(I,J-1)
+ DF(I,J)
35 COMPUTE SUM OF ABSOLUTE MASS SOURCE
RESMM=RESMM+DARS(DS(I,J))
36 CONTINUE
C
C *****
C CALCULATE PRESSURE CORRECTION FIELD.
C *****
DO 36 J=1,JPI
PP(I-1,J)=0.
PP(I,J)=0.
36 CONTINUE
C
C *****
C DO 37 ITER=1,ITPP
DO 38 J=JAP,JMAX
AA(J)=BCV(I,J-1)
BB(J)=BCU(I,J)+BCU(I-1,J)*MM+BCV(I,J)+BCV(I,J-1)
CC(J)=BCV(I,J)
38 DD(J)=BCU(I,J)*PP(I+1,J)+BCU(I-1,J)*PP(I-1,J)
- DS(I,J)
CALL TRIDAG(JAP,JMAX,AA,BB,CC,DD,T)
DO 39 J=JAP,JMAX
PP(I,J)=T(J)
37 CONTINUE
C
C *****
C CALCULATE VELOCITY CORRECTION FIELD.
C *****
DO 40 J=JAU,JMAX
U(I,J)=US(I,J)-CU(I,J)*PP(I+1,J)-PP(I,J)
DO 41 J=JAV,JMAX
V(I,J)=VS(I,J)-CV(I,J)*PP(I,J+1)-PP(I,J)
V(I,JPI)=V(I,JMAX)*B22C(I,JMAX)/B22C(I,JPI)
IF(I.LT.N2) GO TO 42
U(I,JAU-1)={(YU4SQ(I)-YU2SQ(I))*U(I,JAU)
- (YU3SQ(I)-YU2SQ(I))*U(I,JAU+1)}
U(I,JAU-2)={(YU4SQ(I)-YU3SQ(I))*U(I,JAU)
- (YU4SQ(I)-YU3SQ(I))*U(I,JAU+1)}
V(I,2)=V(I,JAV)*V2(I)/V3(I)
V(I,1)=0.
42 CONTINUE
C
C *****
C CALCULATE F-A COEFFICIENT OF TURBULENT KINETIC ENERGY.
C *****
DO 50 J=JA,JMAX
TNUP=TNU(I,J)
REFF=1./(REI+TNUP)
CALL SOURCEK
VV=0.5*(U(I,J)+U(I-1,J))*CEPK
VW=0.5*(V(I,J-1)+V0(I,J))*CEPK
FX=-0.5*ALIP(I,J)*(F1(I,J)+F1(I-1,J)+F1(I,J-1)+F1(I-1,J-1))
FY=-0.5*A22P(I,J)*(F2(I,J)+F2(I-1,J)+F2(I,J-1)+F2(I-1,J-1))+
C 1./B11P(I,J)
ER=DSORT(ALIP(I,J))
FR=DSORT(A22P(I,J))
AR=0.5*(REFF*AJIP(I,J)*(B11P(I,J)*UU+B21P(I,J)*VV)-FX)
BR=0.5*(REFF*AJIP(I,J)*(B12P(I,J)*UU+B22P(I,J)*VV)-FY)
DR=2.*AR
IF(IFA.EQ.2) THEN
CALL COEFF2
ELSE
CALL COEFF3
END IF
TKWB(I,J)=CF(1,1)
TKWC(I,J)=CF(1,2)

```



```

C C C
TKWT(I,J)=CF(1,3)
TKPB(I,J)=CF(2,1)
TKPC(I,J)=CF(2,2)
TKPT(I,J)=CF(2,3)
TKEB(I,J)=CF(3,1)
TKEC(I,J)=CF(3,2)
TKET(I,J)=CF(3,3)
IF(IPA.EQ.2) THEN
D3(I,J)=TKWC(I,J)
ELSE
D3(I,J)=0.
END IF
E3(I,J)=CEPK*REFF*TAUI
CONTINUE
50
C C C
*****
CALCULATE F-A COEFFICIENT OF TURBULENT DISSIPATION RATE.
*****
DO 51 J=JA,JMAX
TNUP=TNUI(I,J)
REFF=1./ (REI*CEPS+TNUP)
CALL SOURCED
UU=0.5*(UO(I,J)+UO(I-1,J))*CEPS
VV=0.5*(VO(I,J-1)+VO(I,J))*CEPS
FX=-0.5*ALLP(I,J)*(F1(I,J)+F1(I-1,J)+F1(I,J-1)+F1(I-1,J-1))
FY=-0.5*A22P(I,J)*(F2(I,J)+F2(I-1,J)+F2(I,J-1)+F2(I-1,J-1))
FV=1./B11P(I,J)
ER=DSQRT(ALLP(I,J))
FR=DSQRT(A22P(I,J))
BR=0.5*(REFF*AJIP(I,J)*(B11P(I,J)+UU*B21P(I,J)+VV)*FV)
DR=2.*AR
IF(IPA.EQ.2) THEN
CALL COEFF2
ELSE
CALL COEFF3
END IF
TDMB(I,J)=CF(1,1)
TDMC(I,J)=CF(1,2)
TDMT(I,J)=CF(1,3)
TDPB(I,J)=CF(2,1)
TDDC(I,J)=CF(2,2)
TDDT(I,J)=CF(2,3)
TDEB(I,J)=CF(3,1)
TDEC(I,J)=CF(3,2)
TDET(I,J)=CF(3,3)
IF(IPA.EQ.2) THEN
O4(I,J)=TDMC(I,J)
ELSE
O4(I,J)=0.
END IF
E4(I,J)=CEPS*REFF*TAUI
CONTINUE
51
C C C
*****
CALCULATE TURBULENT KINETIC ENERGY FIELD.
*****

```

```

*****
DO 52 ITK=1,ITKD
DO 53 J=JA,JMAX
AA(J)=-TKPB(I,J)
RB(J)=1.+D3(I,J)+(E3(I,J)-SRP(I,J))*TKPC(I,J)
CC(J)=-TKPT(I,J)
DD(J)=TKEC(I,J)*TKE(I+1,J)+TKEB(I,J)*TKE(I+1,J-1)
1 +TKWB(I,J)*TKE(I+1,J+1)+TKMC(I,J)*TKE(I-1,J)
2 +TKPC(I,J)*(E3(I,J)+TKEO(I,J)*TKE(I-1,J+1)
3 +TKB(I,J)*TKE(I,J)+TKEO(I,J)+SK(I,J))
DD(JMAX)=DD(JA)-AA(JA)*TKE(I,JM1)
CALL TRIDAG(JA,JMAX,AA,BB,CC,DD,T)
DO 54 J=JA,JMAX
TKE(I,J)=T(J)
TKE(I,JM1)=TKE(I,JMAX)
IF(I.LT.M2) GO TO 52
TKE(I,JA-1)=-((YP4SQ(I)-YP2SQ(I))*TKE(I,JA)
1 -(YP3SQ(I)-YP2SQ(I))*TKE(I,JA+1)
2 / (YP4SQ(I)-YP3SQ(I)))
TKE(I,JA-2)=(YP4SQ(I)*TKE(I,JA)-YP3SQ(I)*TKE(I,JA+1)
1 / (YP4SQ(I)-YP3SQ(I)))
52 CONTINUE
C C C
*****
CALCULATE TURBULENT DISSIPATION RATE FIELD.
*****
DO 55 ITK=1,ITKO
DO 56 J=JA,JMAX
AA(J)=-TDPB(I,J)
BB(J)=1.+D4(I,J)+(E4(I,J)-SDP(I,J))*TDDC(I,J)
CC(J)=-TDPT(I,J)
DD(J)=TDEC(I,J)*TDS(I+1,J)+TDEB(I,J)*TDS(I+1,J-1)
1 +TDMB(I,J)*TDS(I+1,J+1)+TDMC(I,J)*TDS(I-1,J)
2 +TDDC(I,J)*(E4(I,J)+TDSO(I,J)+SD(I,J))
3 +TDDP(I,J)*TDS(I,J)+TDS(I,JM1)
DD(JMAX)=DD(JA)-AA(JA)*TDS(I,JM1)
CALL TRIDAG(JA,JMAX,AA,BB,CC,DD,T)
DO 57 J=JA,JMAX
TDS(I,J)=T(J)
TDS(I,JM1)=TDS(I,JMAX)
IF(I.LT.M2) GO TO 55
TDS(I,JA-1)=-((VP4SQ(I)-VP2SQ(I))*TDS(I,JA)
1 -(VP3SQ(I)-VP2SQ(I))*TDS(I,JA+1)
2 / (VP4SQ(I)-VP3SQ(I)))
TDS(I,JA-2)=(VP4SQ(I)*TDS(I,JA)-VP3SQ(I)*TDS(I,JA+1)
1 / (VP4SQ(I)-VP3SQ(I)))
55 CONTINUE
C C C
*****
CALCULATE THE PSEUDO VELOCITIES.
*****

```

```

*****
CALCULATE TURBULENT KINETIC ENERGY FIELD.
*****

```



```

C *****
C PROVIDE DATA FOR SOME ITERATIONS
C *****
C
C 1 IP(NITER.EQ.1.OR.NITER.EQ.20.OR.NITER.EQ.50.OR.
C 1 NITER.EQ.100.OR.NITER.EQ.150.OR.NITER.EQ.200.
C 2 OR.NITER.EQ.300) THEN
C
C WRITE(10,4114) NITER
C 4114 FORMAT(/5X,15)
C
C WRITE(10,2009) (PR(I,1), I=2, IMAX)
C
C WRITE(10,2009) (UTAU(I), I=2, M2M1)
C
C WRITE(10,2009) (UO(I,1), I=M2, IMAX)
C
C END IF
C
C 4000 CONTINUE
C
C *****
C OUTPUT AFTER CALCULATION
C *****
C DO 3333 I=1, IPI
C
C WRITE(8,2970) NITER, I, XP(I,1), TAU
C 2970 FORMAT(/5X, 'NO. OF ITERATION =', I3, 5X, 'STATION', I3
C 1 , 5X, 'X =', F8.4, 5X, 'TAU =', F6.3)
C
C IF(1.GE.M2) GO TO 3222
C
C 2098 WRITE(8,2099) YP,UTAU,TAU
C 2099 FORMAT(5X, 'YPLUS=', E12.4, 5X, 'UTAU=', E12.4, 5X,
C 1 'TAU=', E12.4)
C
C 2222 WRITE(8,3001)
C 3001 FORMAT(/5X, 'RP(I,J)', 6X, 'VEL U', 7X, 'VEL V', 5X, 'TURB KE', 4X,
C 1 'TURB DIS', 4X, 'PRESSURE'//)
C
C RMAX=YC(1,1)
C DO 3003 J=1, JPI
C RU(I,J)=(YU(I,J)-YU(I,1))/RMAX
C RP(I,J)=(YP(I,J)-YP(I,1))/RMAX
C RV(I,J)=(YV(I,J)-YV(I,1))/RMAX
C 3003 WRITE(8,2005) RP(I,J), UO(I,J), TKEO(I,J), TDSO(I,J)
C 1 , PR(I,J)
C
C 3333 CONTINUE
C
C DATA FOR PLOTTING.
C
C WRITE(7,2009) (XP(I,1), I=1, IPI)
C WRITE(7,2009) (XU(I,1), I=1, IPI)
C WRITE(7,2009) (YP(I,1), I=1, IPI)

```

```

WRITE(7,2009) (YU(I,1), I=1, IPI)
WRITE(7,2009) (PR(I,1), I=2, IMAX)
WRITE(7,2009) (UO(I,1), I=M2, IMAX)
WRITE(7,2009) (UTAU(I), I=2, M2M1)
C
C DO 5001 I=1, IPI
C DO 5001 J=1, JPI
C 5001 WRITE(7,2005) UO(I,J), VO(I,J), PR(I,J), TKEO(I,J),
C 1 TDSO(I,J), TNU(I,J)
C
C 2005 FORMAT(6E12.4)
C 2009 FORMAT(6E12.4)
C
C CLOSE(7)
C CLOSE(8)
C CLOSE(9)
C CLOSE(10)
C
C STOP
C END
C *****
C CALCULATE PRESSURE FIELD.
C *****
C SUBROUTINE PRESS
C
C IMPLICIT REAL*8(A-H, O-Z)
C
C PARAMETER (ID=142, JD=45)
C
C COMMON/PRES1/PRO(ID,JD), PP(ID,JD), PR(ID,JD)
C COMMON/PRES2/CU(ID,JD), CV(ID,JD), CUY(ID,JD)
C COMMON/PRES3/BCU(ID,JD), BCV(ID,JD)
C COMMON/GEOL/XP(ID,JD), YP(ID,JD)
C COMMON/MASS1/DH(ID,JD), DS(ID,JD), DF(ID,JD)
C COMMON/AA/AA(JD), BB(JD), CC(JD), DD(JD), T(JD)
C COMMON/CONS1/I,J,IMAX,JMAX,IPI,JPI,JAU,JAV,JAP,JA,M2,M2M1
C COMMON/CONS2/NITER,ITM,IEND,ITERA
C COMMON/CONS7/IBODY
C COMMON/CONS8/IFA
C
C IF(IFA.EQ.2) THEN
C RFP=0.5
C END IF
C
C FOR ELLIPTIC CALCULATION, IT IS NECESSARY
C TO UNDERRELAX SEVERELY IN FIRST FEW ITERATIONS
C DEPENDING ON THE BODY SHAPE.
C
C IF(IFA.EQ.1) THEN
C
C IF(NITER.LE.30) THEN
C RFP=0.1
C IF(IBODY.EQ.5) RFP=0.01
C ELSE IF(NITER.LE.50) THEN
C RFP=0.2

```

```

C IF (IBODY.EQ.5) RFP=0.05
C ELSE
C RFP=0.2
C IF (IBODY.EQ.5) RFP=0.1
C END IF
C
C END IF
C
C ITERP=10
C
C DO 300 I=1, IPI
C DO 300 J=1, JPI
C 300 PR(I,J)=PR(I,J)
C
C DO 400 ITERG=1,20
C DO 600 II=2, IMAX
C I=IMAX+2-II
C DO 500 ITER=1, ITERP
C DO 100 J=JAP, JMAX
C AA(J)=BCV(I,J-1)
C BB(J)=BCU(I,J)+GCU(I-1,J)+BCV(I,J)+BCV(I,J-1)
C CC(J)=BCV(I,J)
C 100 DD(J)=BCU(I,J)*PR(I+1,J)+BCU(I-1,J)*PR(I-1,J)
C -DH(I,J)
C CALL TRIDAG(JAP, JMAX, AA, BB, CC, DD, T)
C DO 200 J=JAP, JMAX
C 200 PR(I,J)=T(J)
C
C IF (I.LT.M2) THEN
C PR(I,2)=((VP(I,4)-YP(I,2))*PR(I,3)+YP(I,3)-YP(I,2))
C *PR(I,4))/(VP(I,4)-YP(I,3))
C PR(I,1)=((VP(I,4)-YP(I,1))*PR(I,3)-YP(I,3)-YP(I,1))
C *PR(I,4))/(VP(I,4)-YP(I,3))
C ELSE
C PR(I,2)=PR(I,3)
C PR(I,1)=PR(I,3)
C END IF
C
C 500 CONTINUE
C 600 CONTINUE
C
C DO 501 J=1, JPI
C 501 PR(IPI,J)=PR(IMAX,J)
C
C 400 CONTINUE
C
C DO 700 I=1, IPI
C DO 700 J=1, JPI
C 700 PR(I,J)=PRO(I,J)+RFP*(PR(I,J)-PRO(I,J))
C
C RETURN
C END
C
C *****
C CALCULATE SOURCE FUNCTION OF U-VELOCITY...
C *****
C
C SUBROUTINE SOURCEU

```

```

C IMPLICIT REAL*8(A-H,O-Z)
C
C PARAMETER (ID=142,JD=45)
C
COMMON/GE01/XP(ID,JD), YP(ID,JD)
COMMON/GE02/XU(ID,JD), YU(ID,JD)
COMMON/GE03/XV(ID,JD), YV(ID,JD)
COMMON/VEL1/UO(ID,JD), VO(ID,JD)
COMMON/VEL4/U(ID,JD), V(ID,JD)
COMMON/TURB1/TKE(ID,JD), TKEO(ID,JD)
COMMON/TURB3/TNU(ID,JD)
COMMON/ST/SU(ID,JD), SV(ID,JD), SK(ID,JD), SD(ID,JD)
COMMON/STLM/SKP(ID,JD), SDP(ID,JD)
COMMON/CONS1/I,J, IMAX, JMAX, IPI, JPI, JAU, JAV, JAP, JA, M2, M2M1
COMMON/CONS5/RE, REI, TAU, TAU1
COMMON/CONS6/REFF
COMMON/BODY1/B11, B12, B21, B22
COMMON/BODY2/G11, G12, G22, A11, A12, A22, G, AJ1
COMMON/BODY3/XXI, YXI, XET, YET, R
C
R=Yu(I,J)
XET=0.5*(XU(I,J+1)-XU(I,J-1))
YET=0.5*(YU(I,J+1)-YU(I,J-1))
XXI=0.5*(XU(I+1,J)-XU(I-1,J))
YXI=0.5*(YU(I+1,J)-YU(I-1,J))
CALL GEOCOE
DUXI=0.5*(U(I+1,J)-U(I-1,J))
DUYI=0.5*(V(I+1,J)-V(I-1,J))
DUX=AJ1*(B21*DUXI+B22*DUYI)
DUY=AJ1*(B11*DUXI+B12*DUYI)
SS=0.5*A12*(UO(I+1,J+1)+UO(I-1,J-1))-UO(I+1,J-1)
-UO(I-1,J+1))
XET=0.25*(XP(I,J+1)+XP(I+1,J+1)-XP(I,J-1)-XP(I+1,J-1))
YET=0.25*(YP(I,J+1)+YP(I+1,J+1)-YP(I,J-1)-YP(I+1,J-1))
XXI=XP(I+1,J)-XP(I,J)
YXI=YP(I+1,J)-YP(I,J)
CALL GEOCOE
TNXI=TNUI(I+1,J)-TNUI(I,J)
TNET=0.25*(TNU(I,J+1)+TNU(I+1,J+1)-TNU(I,J-1)-TNU(I+1,J-1))
DKXI=TKE(I+1,J)-TKE(I,J)
DKET=0.25*(TKE(I,J+1)+TKE(I+1,J+1)-TKE(I,J-1)-TKE(I+1,J-1))
DKX=AJ1*(B11*DKXI+B12*DKET)
TNUX=AJ1*(B21*TNXI+B22*TNET)
TNUX=AJ1*(B11*TNXI+B12*TNET)
XET=0.5*(XV(I+1,J)+XV(I,J)-XV(I+1,J-1)-XV(I,J-1))
YET=0.5*(YV(I+1,J)+YV(I,J)-YV(I+1,J-1)-YV(I,J-1))
XXI=0.5*(XV(I+1,J)+XV(I+1,J-1)-YV(I,J)-YV(I,J-1))
YXI=0.5*(YV(I+1,J)+YV(I+1,J-1)-YV(I,J)-YV(I,J-1))
CALL GEOCOE
DUXI=0.5*(U(I+1,J)+V(I+1,J-1)-V(I,J)-V(I,J-1))
DUYI=0.5*(V(I+1,J)+V(I,J)-V(I+1,J-1)-V(I,J-1))
DUX=AJ1*(B11*DUXI+B12*DUYI)

```

1 SU(I,J) = SS + REFF \* (TNUY \* (DUY + DVX)  
+ 2. \* TNUX \* DUX - 2. / 3. \* DKX)

C RETURN  
C END

\*\*\*\*\*  
CALCULATE SOURCE FUNCTION OF V-VELOCITY..  
\*\*\*\*\*

C SUBROUTINE SOURCEV

C IMPLICIT REAL\*8(A-H,O-Z)

C PARAMETER (ID=142,JD=45)

C COMMON/GEOL/XP(ID,JD), YP(ID,JD)  
C COMMON/GEOS3/XU(ID,JD), YU(ID,JD)  
C COMMON/GEOS4/XV(ID,JD), YV(ID,JD)  
C COMMON/VEL1/UO(ID,JD), VO(ID,JD)  
C COMMON/VEL4/U(ID,JD), V(ID,JD)  
C COMMON/TURB1/TKE(ID,JD), TKEO(ID,JD)  
C COMMON/TURB3/TNU(ID,JD)

C COMMON/ST/SU(ID,JD), SV(ID,JD), SK(ID,JD), SD(ID,JD)  
C COMMON/STLN/SKP(ID,JD), SDP(ID,JD)

C COMMON/CONS1/I,J,IMAX,JMAX,IP1,JPI,JAU,JAV,JAP,JA,M2,M2M1  
C COMMON/CONS6/REFP

C COMMON/BODY1/B11,B12,B21,B22  
C COMMON/BODY2/G11,G12,G22,A11,A12,A22,G,AJI  
C COMMON/BODY3/XXI,XXI,XXI,XET,YET,R

C R=YP(I,J)

C XXI=0.25\*(XP(I+1,J)+YP(I+1,J+1))-XP(I-1,J)-YP(I-1,J+1))

C YXI=0.25\*(YP(I+1,J)+XP(I+1,J+1))-YP(I-1,J)-XP(I-1,J+1))

C XET=XP(I,J+1)-XP(I,J)

C YET=YP(I,J+1)-YP(I,J)

C CALL GEOCOE

C DKET=TKE(I,J+1)-TKE(I,J)

C DNXT=0.25\*(TKE(I+1,J)+TKE(I+1,J+1))-TKE(I-1,J)-TKE(I-1,J+1))

C TNXT=TNU(I,J+1)-TNU(I,J)

C TNXI=0.25\*(TNU(I+1,J)+TNU(I+1,J+1))-TNU(I-1,J)-TNU(I-1,J+1))

C DKV-AJI\*(B21\*DKXI+B22\*DKET)

C TNUY-AJI\*(B21\*TNXI+B22\*TNXT)

C TNUX-AJI\*(B11\*TNXI+B12\*TNXT)

C XET=0.5\*(XU(I,J+1)+XU(I-1,J+1))-XU(I,J)-XU(I-1,J)

C YET=0.5\*(YU(I,J+1)+YU(I-1,J+1))-YU(I,J)-YU(I-1,J)

C XXI=0.5\*(XU(I,J)+XU(I-1,J))-XU(I-1,J)

C YXI=0.5\*(YU(I,J)+YU(I-1,J))-YU(I-1,J)

C XET=0.5\*(XV(I+1,J)+XV(I-1,J))-XV(I,J)

C YET=0.5\*(YV(I+1,J)+YV(I-1,J))-YV(I,J)

CALL GEOCOE  
DVET=0.5\*(V(I,J+1)-V(I,J-1))  
DVXI=0.5\*(V(I+1,J)-V(I-1,J))  
DVX-AJI\*(B11\*DVXI+B12\*DVET)  
DVY-AJI\*(B21\*DVXI+B22\*DVET)

C SS=0.5\*A12\*(VO(I+1,J+1)+VO(I-1,J-1))-VO(I+1,J-1)  
C -VO(I-1,J+1)

C R=YP(I,J)  
C VO=V(I,J)

C SV(I,J) = SS + REFF \* (TNUX \* (DVX + DUY)  
C + 2. \* TNUY \* DVY - 2. / 3. \* DKY) - VO / R

C RETURN

C END

\*\*\*\*\*  
CALCULATE SOURCE FUNCTION OF KINETIC ENERGY FIELD..  
\*\*\*\*\*

C SUBROUTINE SOURCEK

C IMPLICIT REAL\*8(A-H,O-Z)

C PARAMETER (ID=142,JD=45)

C COMMON/GEOL/XP(ID,JD), YP(ID,JD)  
C COMMON/GEOS3/XU(ID,JD), YU(ID,JD)  
C COMMON/GEOS4/XV(ID,JD), YV(ID,JD)  
C COMMON/VEL1/UO(ID,JD), VO(ID,JD)  
C COMMON/VEL4/U(ID,JD), V(ID,JD)  
C COMMON/TURB1/TKE(ID,JD), TKEO(ID,JD)  
C COMMON/TURB2/TDS(ID,JD), TDSO(ID,JD)  
C COMMON/TURB3/TNU(ID,JD)  
C COMMON/TURB4/TGE(ID,JD)  
C COMMON/ST/SU(ID,JD), SV(ID,JD), SK(ID,JD), SD(ID,JD)  
C COMMON/STLN/SKP(ID,JD), SDP(ID,JD)  
C COMMON/CONS1/I,J,IMAX,JMAX,IP1,JPI,JAU,JAV,JAP,JA,M2,M2M1  
C COMMON/CONS4/CNU,CNU3,CNU21,CEPS1,CEPS2,CEPK,CEPS,CKAR,E  
C COMMON/CONS6/REFP

C COMMON/BODY1/B11,B12,B21,B22  
C COMMON/BODY2/G11,G12,G22,A11,A12,A22,G,AJI  
C COMMON/BODY3/XXI,XXI,XXI,XET,YET,R

C R=YP(I,J)

C YXI=YP(I,J)-YU(I-1,J)

C XET=0.25\*(XU(I,J+1)+XU(I-1,J+1))-XU(I,J)-XU(I-1,J-1))

C YET=0.25\*(YU(I,J+1)+YU(I-1,J+1))-YU(I,J)-YU(I-1,J-1))

C CALL GEOCOE

C DUET=0.25\*(U(I,J+1)+U(I-1,J+1))-U(I,J)-U(I-1,J-1))

C DUXI=U(I,J)-U(I-1,J)

C DUX-AJI\*(B11\*DUXI+B12\*DUET)

C DUY-AJI\*(B21\*DUXI+B22\*DUET)

COMMON/TURB3/TNU(ID,JD)  
COMMON/TURB4/TGE(ID,JD)  
COMMON/ST/SU(ID,JD),SV(ID,JD),SK(ID,JD),SD(ID,JD)  
COMMON/STLN/SKP(ID,JD),SDP(ID,JD)  
COMMON/CONS1/I,J,IMAX,IP1,JPI,JAU,JAV,JAP,JA,M2,MZM1  
COMMON/CONS4/CNU,CNU3,CNU21,CEPS1,CEPS2,CEPK,CEFS,CKAR,E  
COMMON/CONS6/REFP  
COMMON/BODY1/B11,B12,B21,B22  
COMMON/BODY2/G11,G12,G22,A11,A12,A22,G,AJ1  
COMMON/BODY3/XXI,YXI,XET,YET,R

R=YP(I,J)  
XXI=XU(I,J)-XU(I-1,J)  
YXI=YU(I,J)-YU(I-1,J)  
XET=0.25\*(XU(I,J+1)+XU(I-1,J+1)-XU(I,J-1)-XU(I-1,J-1))  
YET=0.25\*(YU(I,J+1)+YU(I-1,J+1)-YU(I,J-1)-YU(I-1,J-1))  
CALL GEOCOE  
DUET=0.25\*(U(I,J+1)+U(I-1,J+1)-U(I,J-1)-U(I-1,J-1))  
DUXI=U(I,J)-U(I-1,J)  
DUY=AJI\*(B11\*DUXI+B12\*DUET)  
DUY=AJI\*(B21\*DUXI+B22\*DUET)  
XET=XV(I,J)-XV(I,J-1)  
YET=YV(I,J)-YV(I,J-1)  
XXI=0.25\*(XV(I+1,J)+XV(I+1,J-1)-XV(I-1,J)-XV(I-1,J-1))  
YXI=0.25\*(YV(I+1,J)+YV(I+1,J-1)-YV(I-1,J)-YV(I-1,J-1))  
CALL GEOCOE  
DVET=V(I,J)-V(I,J-1)  
DVXI=0.25\*(V(I+1,J)+V(I+1,J-1)-V(I-1,J)-V(I-1,J-1))  
DVY=AJI\*(B11\*DVXI+B12\*DVET)  
DVY=AJI\*(B21\*DVXI+B22\*DVET)

XXI=0.5\*(XP(I+1,J)-XP(I-1,J))  
YXI=0.5\*(YP(I+1,J)-YP(I-1,J))  
XET=0.5\*(XP(I,J+1)-XP(I,J-1))  
YET=0.5\*(YP(I,J+1)-YP(I,J-1))  
CALL GEOCOE  
DEET=0.5\*(TDS(I,J+1)-TDS(I,J-1))  
TNET=0.5\*(TDS(I,J+1)-TDS(I,J-1))  
DEXI=0.5\*(TDS(I+1,J)-TDS(I-1,J))  
TNXI=0.5\*(TNU(I+1,J)-TNU(I-1,J))  
DEX=AJI\*(B11\*DEXI+B12\*DEET)  
DEV=AJI\*(B21\*DEXI+B22\*DEET)  
TNUY=AJI\*(B21\*TNXI+B22\*TNET)  
TNUY=AJI\*(B11\*TNXI+B12\*TNET)  
SS=0.5\*A12\*(TDS0(I+1,J+1)+TDS0(I-1,J-1)-TDS0(I+1,J-1)  
-TDS0(I-1,J+1))

R=YP(I,J)  
VO=0.5\*(V(I,J)+V(I,J-1))  
TGE(I,J)=TNU(I,J)\*(2.\*DUX\*DUX+2.\*DVI\*DVI+  
(DVX+DUY)\*\*2+2.\*(VO/R)\*\*2)

ADK=TDS(I,J)/TKE(I,J)  
SDP(I,J)--ADK\*CEPK\*REFP\*CEPS2  
SD(I,J)--SS+CEPK\*REFP\*CEPS1\*TGE(I,J)\*TDS(I,J)/TKE(I,J)  
+REFP\*(TNUY\*DEV+TNUX\*DEX)

XET=XV(I,J)-XV(I,J-1)  
YET=YV(I,J)-YV(I,J-1)  
XXI=0.25\*(XV(I+1,J)+XV(I+1,J-1)-XV(I-1,J)-XV(I-1,J-1))  
YXI=0.25\*(YV(I+1,J)+YV(I+1,J-1)-YV(I-1,J)-YV(I-1,J-1))  
CALL GEOCOE  
DVET=V(I,J)-V(I,J-1)  
DVXI=0.25\*(V(I+1,J)+V(I+1,J-1)-V(I-1,J)-V(I-1,J-1))  
DVY=AJI\*(B11\*DVXI+B12\*DVET)  
DVY=AJI\*(B21\*DVXI+B22\*DVET)

XXI=0.5\*(XP(I+1,J)-XP(I-1,J))  
YXI=0.5\*(YP(I+1,J)-YP(I-1,J))  
XET=0.5\*(XP(I,J+1)-XP(I,J-1))  
YET=0.5\*(YP(I,J+1)-YP(I,J-1))  
CALL GEOCOE  
DKET=0.5\*(TKE(I,J+1)-TKE(I,J-1))  
TNET=0.5\*(TNU(I,J+1)-TNU(I,J-1))  
DKXI=0.5\*(TKE(I+1,J)-TKE(I-1,J))  
TNXI=0.5\*(TNU(I+1,J)-TNU(I-1,J))  
DKX=AJI\*(B11\*DKXI+B12\*DKET)  
DNX=AJI\*(B21\*DKXI+B22\*DKET)  
TNUY=AJI\*(B21\*TNXI+B22\*TNET)  
TNUY=AJI\*(B11\*TNXI+B12\*TNET)

SS=0.5\*A12\*(TKE0(I+1,J+1)+TKE0(I-1,J-1)-TKE0(I+1,J-1)  
-TKE0(I-1,J+1))  
R=YP(I,J)  
VO=0.5\*(V(I,J)+V(I,J-1))  
TGE(I,J)=TNU(I,J)\*(2.\*DUX\*DUX+2.\*DVI\*DVI+  
(DVX+DUY)\*\*2+2.\*(VO/R)\*\*2)

ADK=TDS(I,J)/TKE(I,J)  
SDP(I,J)--ADK\*CEPK\*REFP  
SK(I,J)--SS+CEPK\*REFP\*TGE(I,J)  
+CEPK\*REFP\*(TNUY\*DNX+TNUX\*DKX)

RETURN  
END  
\*\*\*\*\*  
CALCULATE SOURCE FUNCTION OF DISSIPATION RATE.  
\*\*\*\*\*  
SUBROUTINE SOURCED  
IMPLICIT REAL\*8(A-H,O-Z)  
PARAMETER (ID=142,JD=45)  
COMMON/GEOL/XP(ID,JD),YP(ID,JD)  
COMMON/GEOS/XU(ID,JD),YU(ID,JD)  
COMMON/GEO4/XV(ID,JD),YV(ID,JD)  
COMMON/VEL1/UO(ID,JD),VO(ID,JD)  
COMMON/VEL4/U(ID,JD),V(ID,JD)  
COMMON/TURB1/TKE(ID,JD),TKE0(ID,JD)  
COMMON/TURB2/TDS(ID,JD),TDS0(ID,JD)

C

```

RETURN
END
*****
SUBROUTINE COEFF2 FOR CALCULATING PARTIALLY PARABOLIC
FINITE ANALYTIC COEFFICIENTS OF TRANSPORT EQUATIONS
*****

```

C

```

SUBROUTINE COEFF2
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/COEF7/CF(3,3)
COMMON/FAL/AR,BR,DR,ER,FR
COMMON/CONS3/EMAX,FMAX,EPE,EPE1,PI
IF(DABS(BR).LT.EPE) GO TO 1
BR=BR
BR=BR/FR
HV=1./FR
BK=BR*HV
DBK=DMIN1(DABS(BK),FMAX)
BK=DSIGN(DBK,BK)
EPBK=DEXP(BK)
EPBK=1./EPBK
TANHB=(EPBK-EPBK1)/(EPBK+EPBK1)
CF(2,3)=EPBK1/EPBK+EPBK
CF(2,2)=TANHB/(2.*BRO)
CF(2,1)=EPBK/(EPBK1+EPBK)
GO TO 2

```

1

```

CF(2,1)=0.5
CF(2,2)=0.5/FR/FR
CF(2,3)=0.5

```

2

```

CONTINUE
CF(1,1)=0.
CF(1,2)=CF(2,2)*DR
CF(1,3)=0.
CF(3,1)=0.
CF(3,2)=0.
CF(3,3)=0.
RETURN
END

```

C

```

*****
COEFF3 FOR CALCULATING 9-POINT FINITE-ANALYTIC
COEFFICIENTS OF TRANSPORT EQUATIONS-OLD VERSION.
*****

```

C

```

SUBROUTINE COEFF3
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON/COEF7/CF(3,3)
COMMON/FAL/AR,BR,DR,ER,FR
COMMON/CONS3/EMAX,FMAX,EPE,EPE1,PI
MAX=12
DAR=DMAX1(DABS(AR),EPE)
DBR=DMAX1(DABS(BR),EPE)
AR=DSIGN(DAR,AR)
BR=DSIGN(DBR,BR)
AR=AR/ER
BR=BR/FR

```

C

```

HX=1./ER
HY=1./FR
ER2=ER*ER
FR2=FR*FR
AB2=AR*AR+BR*BR
AH=AR*HX
AK=AR*HY
BH=BR*HX
BK=BR*HY
DAH=DMIN1(DABS(AH),FMAX)
DBK=DMIN1(DABS(BK),FMAX)
AH=DSIGN(DAH,AH)
BK=DSIGN(DBK,BK)
AH2=AH*AH
BK2=BK*BK
EPAH=DEXP(AH)
EPBK=DEXP(BK)
EPAHI=1./EPAH
EPBKI=1./EPBK
SINHA=EPAH-EPBKI
SINHB=EPBK-EPBKI
COSHA=EPAH+EPBKI
COSH=EPBK+EPBKI
COTHA=COSHA/SINHA
COTHB=COSHB/SINHB
IF(DMAX1(DAH,DBK).GE.EMAX) GO TO 120
PWR=1.
IF(HX.GT.HY) GO TO 21
EX2=0.
DO 10 M=1,MAX
ZA=(M-0.5)*PI
ZA2=ZA*ZA
PWR=PWR
AB=DEXP(DSQRT(AB2+ZA2*ER2))*HY
10 EX2=EX2-PWR*ZA/((AB+1./AB)*(AH2+ZA2))*(AH2+ZA2)
PA=2.*AH*COTHA*COSHA*COSH*EX2
PB=1.+BH*COTHB/(AK*COTHA)*(PA-1.)
Q=1.-PA-PB
CF(2,2)=0.5*HX/(AR*COTHA)*(1.-PA)
GO TO 160
21 EY2=0.
DO 20 M=1,MAX
ZA=(M-0.5)*PI
ZA2=ZA*ZA
PWR=PWR
AB=DEXP(DSQRT(AB2+ZA2*FR2))*HX
20 EY2=EY2-PWR*ZA/((AB+1./AB)*(BK2+ZA2))*(BK2+ZA2)
PB=2.*BK*COTHB*COSHA*COSH*EY2
PA=1.+AK*COTHA/(BH*COTHB)*(PB-1.)
Q=1.-PA-PB
CF(2,2)=0.5*HY/(AR*COTHA)*(1.-PA)
GO TO 160
120 AI=AH/SINHA
BI=BK/SINHB
AB1=A1*A1*HY/HX+BI*BI*HX/HY
DAK=DABS(AK)
DBH=DABS(BH)
Q=DAK/(AB1+DBH)
Q=DBH/(AB1+DAK)
PA=1.-((1.+Q)*AK*COTHA/(AK*COTHA+BH*COTHB))

```

```

G=R*R*(XXI*YET-XET*YXI)*(XXI*YET-XET*YXI)
GI=1./G
A11=R*R*GI*G22
A22=R*R*GI*G11
A12=-R*R*GI*G12
AJI=DSORT(GI)
RETURN
END

```

```

PB=1.-PA-Q
CF(2,2)=0.5*HX/(AR*COTNA)*(1.-PA)
BE=EPAH/COSHA
BN=EPBK/COSHB
BS=EPBK/COSHB
CF(1,1)=BS*BN*Q
CF(3,1)=BS*BE*Q
CF(1,3)=BN*BN*Q
CF(3,3)=BN*BE*Q
CF(2,1)=BS*PA
CF(2,3)=BN*PB
CF(3,2)=BE*PB
RETURN
END

```

160

A-21

```

C *****
C SUBROUTINE TRIDAG TO SOLVE ALGEBRAIC EQUATIONS
C SIMULTANEOUSLY FOR EACH ROW OR COLUMN
C *****

```

```

SUBROUTINE TRIDAG(IF,L,A,B,C,D,V)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
PARAMETER ( JD=45 )
DIMENSION A(JD),B(JD),C(JD),D(JD),V(JD),BETA(JD),GAMMA(JD)
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)
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
V(I)=GAMMA(I)-C(I)*V(I+1)/BETA(I)
RETURN
END

```

```

C *****
C CALCULATE THE GEOMETRIC CONSTANTS.
C *****

```

```

SUBROUTINE GEOCOS
IMPLICIT REAL*8(A-H,O-Z)
COMMON/BODY1/B11,B12,B21,B22
COMMON/BODY2/G11,G12,G22,A11,A12,A23,G,AJI
COMMON/BODY3/XXI,YXI,XET,YET,R
B11=R*YET
B12=-R*YXI
B21=-R*XET
B22=R*XXI
G11=XXI*XXI+YXI*YXI
G22=YET*YET+XET*XET
G12=YXI*YET+XXI*XET

```

C



```

C C *****
C C * THIS PROGRAM CALCULATES PAST AXISYMMETRIC BODIES *
C C * USING THE K-E TURBULENCE MODEL AND WALL FUNCTION *
C C * METHOD WITH A SPACE MARCHING TECHNIQUES. *
C C *****
C C *****
C C * PROGRAMMED BY S.K.CHOI *
C C *****
C C *****
C C * THE BASIC FEATURE OF THIS CODE. *
C C *****
C C * ONE-VELOCITY STAGGERED GRID SYSTEM. *
C C * SIMPLER NUMERICAL ALGORITHM. *
C C * SPACE MARCHING TECHNIQUE. *
C C * BODY-FITTED COORDINATE USING PARTIAL TRANSFORMATION. *
C C * ( CARTESIAN, CYLINDRICAL BASE COORDINATE. ) *
C C *****
C C PROGRAM HAWKEYE

```

```

COMMON/VEL5/UO(ID,JD)
COMMON/VEL6/VO(ID,JD)
COMMON/TURB1/AKE(ID,JD),ADS(ID,JD)
COMMON/TURB2/VIS(ID,JD),GEN(ID,JD)
COMMON/ST1/SCUE(ID,JD),SPE(ID,JD)
COMMON/ST2/SCVN(ID,JD),SPVN(ID,JD)
COMMON/ST3/SCK(ID,JD),SPK(ID,JD)
COMMON/ST4/SCD(ID,JD),SPD(ID,JD)
COMMON/ST5/SUKD(ID,JD),SPKD(ID,JD)
COMMON/ST6/SUDD(ID,JD),SPDD(ID,JD)
COMMON/COEF1/AEE(ID,JD),AWE(ID,JD),ANE(ID,JD)
COMMON/COEF2/AESE(ID,JD),APEU(ID,JD),APVE(ID,JD)
COMMON/COEF3/AEN(ID,JD),AWN(ID,JD),ANN(ID,JD)
COMMON/COEF4/ASN(ID,JD),APUN(ID,JD),APVN(ID,JD)
COMMON/COEF5/AEK(ID,JD),APK(ID,JD),ANK(ID,JD)
COMMON/COEF6/ASK(ID,JD),APK(ID,JD),ADK(ID,JD)
COMMON/COEF7/AED(ID,JD),APD(ID,JD),ADD(ID,JD)
COMMON/COEF8/ASD(ID,JD),APD(ID,JD),ADD(ID,JD)
COMMON/AAA/AA(JD),BB(JD),CC(JD),DD(JD),T(JD)
COMMON/CONS1/IMAX,JMAX,IPI,JPI,M2,M2M1
COMMON/CONS2/NSWPU,NSWPP,NSWPP,NSWPP
COMMON/CONS3/RESUE,RESVE,RESAK,RESAD,RESHM
COMMON/CONS4/URFU,URFU,URFU,URFU,URFD,URFVIS,URFPR
COMMON/CONS5/RENUM,DENSIT,VISCOS,FRANDT
COMMON/CONS6/IBODY,ICORO
COMMON/CONS7/NITER,MAXIT,ITMAX,NINER,ITOUT
COMMON/CONS8/CNU,CD,C1,C2,ELOG,CAPPA,PRTE,PRED
COMMON/CONS9/PI,GREAT

```

```

C C DATA FINAME(1)/'ha1bdat',FINAME(2)/'ha2bdat',
C C $ FINAME(3)/'ha3bdat',FINAME(4)/'ha4bdat',
C C $ FINAME(5)/'ha5bdat',FINAME(6)/'ha6bdat',
C C DATA F2NAME(1)/'hfva101',F2NAME(2)/'hfva201',
C C $ F2NAME(3)/'hfva301',F2NAME(4)/'hfva401',
C C $ F2NAME(5)/'hfva501',F2NAME(6)/'hfva601',
C C DATA F3NAME(1)/'hfva102',F3NAME(2)/'hfva202',
C C $ F3NAME(3)/'hfva302',F3NAME(4)/'hfva402',
C C $ F3NAME(5)/'hfva502',F3NAME(6)/'hfva602',
C C DATA F4NAME(1)/'hfva103',F4NAME(2)/'hfva203',
C C $ F4NAME(3)/'hfva303',F4NAME(4)/'hfva403',
C C $ F4NAME(5)/'hfva503',F4NAME(6)/'hfva603',
C C DATA F5NAME(1)/'hfva104',F5NAME(2)/'hfva204',
C C $ F5NAME(3)/'hfva304',F5NAME(4)/'hfva404',
C C $ F5NAME(5)/'hfva504',F5NAME(6)/'hfva604',
C C *****
C C * MAIN PROGRAM
C C *****
C C *****
C C * AXISYMMETRIC BODIES.
C C * *****
C C *

```





```

C C PRESSURE CORRECTION EQUATION.
DO 119 J=1,JPI
C 119 VOL(1,J)=0.DO
C C
DO 120 I=1,IPI
C 120 VOL(I,1)=0.DO
C C
*****
C C SET VARIABLES TO ZERO.
C C *****
C C
DO 14 I=1,IPI
C C WALL SURFACE VELOCITY
C C
VSP(I)=0.DO
C C
DO 14 J=1,JPI
C C
CARTESIAN ( OR CYLINDRICAL ) VELOCITY COMPONENTS.
C C
UE(I,J)=0.DO
C C VN(I,J)=0.DO
C C UO(I,J)=0.DO
C C VO(I,J)=0.DO
C C
PSEUDO-VELOCITY COMPONENTS.
C C
UHE(I,J)=0.DO
C C VHN(I,J)=0.DO
C C
PRESSURE, PRESSURE CORRECTION FIELD.
C C
PR(I,J)=0.DO
C C PRO(I,J)=0.DO
C C PP(I,J)=0.DO
C C
KINETIC ENERGY AND DISSIPATION RATE.
C C
AKE(I,J)=0.DO
C C ADS(I,J)=0.DO
C C
DENSITY AND VISCOSITY
C C
DEN(I,J)=DENSIT
C C VIS(I,J)=VISCOS
C C
COEFFICIENTS OF PRESSURE TERM IN
C C MOMENTUM EQUATIONS.
C C
CXUE(I,J)=0.DO
C C CYUR(I,J)=0.DO
C C
CXVN(I,J)=0.DO
C C CYVN(I,J)=0.DO
C C
QXUE(I,J)=0.DO
C C QYVN(I,J)=0.DO
C C
COEFFICIENTS OF PRESSURE AND

```

```

C C PRESSURE CORRECTION EQUATION.
BCU(I,J)=0.DO
BCV(I,J)=0.DO
SOURCE TERMS.
FOR VELOCITY, UE.
SCUE(I,J)=0.DO
SPUE(I,J)=0.DO
FOR VELOCITY, VN.
SCVN(I,J)=0.DO
SPVN(I,J)=0.DO
FOR KINETIC ENERGY, AKE.
SCK(I,J)=0.DO
SPK(I,J)=0.DO
SUKD(I,J)=0.DO
SPKD(I,J)=0.DO
FOR DISSIPATION RATE, ADS.
SCD(I,J)=0.DO
SPD(I,J)=0.DO
SUDD(I,J)=0.DO
SPDD(I,J)=0.DO
FOR PRESSURE AND PRESSURE CORRECTION EQUATION.
SPP(I,J)=0.DO
SPR(I,J)=0.DO
14 CONTINUE
*****
INLET CONDITIONS
*****
FLOWIN=0.DO
ARDEN=0.DO
ARDENT=0.DO
XNONIN=0.DO
UTAU(1)=0.038
DELTA=0.006
DELTAI=1./DELTA
UE(1,1)=0.0
VN(1,1)=0.0
AKE(1,1)=0.0
ADS(1,1)=0.0
DO 20 J=2,JPI
YV=YE(1,J)-YE(1,1)

```

```

ETA=YY*DELTAI
ETAM=(YE(1,JP1)-YE(1,1))*DELTAI
1/7TH LAW FOR U-VELOCITY.
UE(1,J)=1.02*ETA**0.143
IF(ETA.GE.1.) UE(1,J)=1.02-0.02*(ETA-1.)/(ETAM-1.)
DPLUS=DELTA*RENUM*UTAU(1)
IF(DPLUS.LE.1000.) DPLUS=1001.
YPLUS=RENUM*YY*UTAU(1)
IF(YPLUS.LE.5.) THEN
UE(1,J)=UTAU(1)*YPLUS
ELSE IF(YPLUS.LE.50.) THEN
UE(1,J)=UTAU(1)*(5.D0*DLOG(YPLUS)-3.05D0)
ELSE IF(YPLUS.LE.1000.) THEN
UE(1,J)=UTAU(1)*(2.5*DLOG(YPLUS)+5.5)
ELSE IF(YPLUS.LE.DPLUS) THEN
UE(1,J)=UTAU(1)*(2.5*DLOG(YPLUS)+5.5
+ 0.5*2./CAPPA*(DSIN(PI/2.*YY/DELTA))**2)
ELSE
UE(1,J)=1.
END IF
AKE(1,J)=UTAU(1)*UTAU(1)*(1.D0-ETA)/(CHU**0.5)
IF(ETA.GE.1.) AKE(1,J)=1.D-8
ADS(1,J)=(CHU**0.75)*AKE(1,J)**1.5/(CAPPA*YY)
UHR(1,J)=UE(1,J)
UO(1,J)=UE(1,J)
VN(1,J)=0.D0
VO(1,J)=0.D0
20 CONTINUE
COMPUTE INLET MEAN VELOCITY, MASS FLUX
AND MOMENTUM FLUX.
DO 201 J=2, JMAX
ARDEN=DEN(1,J)*0.5*(R(1,J)+R(1,J-1))*(YC(1,J)-YC(1,J-1))
XMONIN=XMONIN+ARDEN*UE(1,J)*UE(1,J)
ARDEN=ARDEN+ARDEN
201 FLOWIN=FLOWIN+ARDEN*UE(1,J)
UMEAN=FLOWIN/ARDEN
*****
RETURN POINT OF ITERATION.
*****
4000 NITER=NITER+1
RESUE=0.

```

```

RESVN=0.
RESAK=0.
RESAD=0.
RESMN=0.
DO 19 J=1, JPI
DO 21 I=1, IPI
SPP(I,J)=0.D0
21 PP(I,J)=0.D0
IF(NITER.LT.4) GO TO 19
VN(1,J)=VN(2,J)
VO(1,J)=VO(2,J)
AKE(1,J)=AKE(2,J)
ADS(1,J)=ADS(2,J)
19 CONTINUE
*****
RETURN POINT OF SPACE MARCHING PROCESS.
*****
DO 3000 I=2, IMAX
IF(NITER.EQ.1) THEN
UTAU(1)=UTAU(I-1)
DO 23 J=1, JPI
UE(I,J)=UE(I-1,J)
VN(I,J)=VN(I-1,J)
UO(I,J)=UE(I-1,J)
VO(I,J)=VN(I-1,J)
AKE(I,J)=AKE(I-1,J)
ADS(I,J)=ADS(I-1,J)
UE(I+1,J)=UE(I,J)
VN(I+1,J)=VN(I,J)
AKE(I+1,J)=AKE(I,J)
ADS(I+1,J)=ADS(I,J)
UO(I+1,J)=UE(I,J)
VO(I+1,J)=VN(I,J)
23 CONTINUE
END IF
YE2SQ=YE2(I)*YE2(I)
YE3SQ=YE3(I)*YE3(I)
YE4SQ=YE4(I)*YE4(I)
YN2SQ=YN2(I)*YN2(I)
YN3SQ=YN3(I)*YN3(I)
YN4SQ=YN4(I)*YN4(I)
YP2SQ=YP2(I)*YP2(I)

```

```
YP3Q=YP3(I)*YP3(I)
YP4Q=YP4(I)*YP4(I)
```

```
*****
CALCULATE THE EFFECTIVE EDDY VISCOSITY
*****
```

```
DO 240 J=2,JP1
VISOLD=VIS(I,J)
IF(ABS(I,J).EQ.0.D0) GO TO 241
VIS(I,J)=CRUDEN(I,J)*AKE(I,J)+VISCOS
GO TO 242
241 VIS(I,J)=VISCOS
UNDER-RELAX VISCOSITY
242 VIS(I,J)=URFVIS*VIS(I,J)+(1.D0-URFVIS)*VISOLD
240 CONTINUE
```

```
DO 244 J=4,JP1
IF(VIS(I,J-1).LE.VIS(I,J-2)).AND.VIS(I,J-1).LT.
VIS(I,J)) VIS(I,J)=VIS(I,J-1)
244 CONTINUE
ALONG THE BOTTOM WALL AND SYMMETRY LINE.
IF(I.GE.M2) VIS(I,1)=VIS(I,2)
IF(I.LT.M2) VIS(I,1)=VISCOS
INLET AND OUTLET CONDITIONS
IF(I.EQ.2) THEN
DO 26 J=1,JP1
VIS(I,J)=VIS(2,J)
END IF
IF(I.EQ.IMAX) THEN
DO 261 J=1,JP1
VIS(IP1,J)=VIS(IMAX,J)
END IF
```

```
IF(ITER.EQ.1) THEN
DO 262 J=1,JP1
VIS(I+1,J)=VIS(I,J)
END IF
270 CONTINUE
*****
STARTING POINT OF INNER ITERATION.
*****
```

```
DO 2000 ITERN=1,NINER
*****
CALCULATE THE COEFFICIENTS AND SOURCE TERMS FOR
NORTH CELL SURFACE VELOCITIES.
*****
DO 380 J=3,JMAX
COMPUTE GEOMETRIC CONSTANTS
FOR EAST CONTROL VOLUME SURFACE
RE=R(I,J)
B1E=YE(I,J+1)-YE(I,J)
B2E=(XE(I,J+1)-XE(I,J))
B21E=(VN(I+1,J)-VN(I,J))
B22E=XN(I+1,J)-XN(I,J)
D1E=RE*RE*(B1E*B1E+B2E*B2E)
D2E=RE*RE*(B1E*B21E+B2E*B22E)
VOLE=0.25*(VOL(I+1,J)+VOL(I,J)+VOL(I,J+1)+VOL(I,J+1))
FOR WEST CONTROL VOLUME SURFACE
RW=R(I-1,J)
B1W=YE(I-1,J+1)-YE(I-1,J)
B2W=(XE(I-1,J+1)-XE(I-1,J))
B21W=(VN(I,J)-VN(I-1,J))
B22W=XN(I,J)-XN(I-1,J)
D1W=RW*RW*(B1W*B1W+B2W*B2W)
D2W=RW*RW*(B1W*B21W+B2W*B22W)
VOLW=0.25*(VOL(I-1,J)+VOL(I-1,J+1)+VOL(I,J)+VOL(I,J+1))
FOR NORTH CONTROL VOLUME SURFACE
RN=0.25*(R(I-1,J)+R(I-1,J+1)+R(I,J)+R(I,J+1))
B1N=VN(I,J+1)-VN(I,J)
B2N=(XN(I,J+1)-XN(I,J))
B21N=(YE(I,J+1)-YE(I-1,J+1))
B22N=XE(I,J+1)-XE(I-1,J+1)
D21N=RN*RN*(B1N*B21N+B22N*B22N)
D22N=RN*RN*(B21N*B22N+B22N*B22N)
VOLN=VOL(I,J+1)
FOR SOUTH CONTROL VOLUME SURFACE
RS=0.25*(R(I-1,J)+R(I-1,J-1)+R(I,J)+R(I,J-1))
B1S=VN(I,J)-VN(I,J-1)
B2S=(XN(I,J)-XN(I,J-1))
B21S=(YE(I,J)-YE(I-1,J))
B22S=XE(I,J)-XE(I-1,J)
D21S=RS*RS*(B1S*B21S+B22S*B22S)
D22S=RS*RS*(B21S*B22S+B22S*B22S)
VOLS=VOL(I,J)
FOR CENTER POINT
RP=0.5*(R(I,J)+R(I-1,J))
```

```

      B11P=YP(I,J+1)-YP(I,J)
      B12P=-XP(I,J+1)-XP(I,J)
      B21P=-(YC(I,J)-YC(I-1,J))
      B22P=XC(I,J)-XC(I-1,J)
      VOLP=0.5*(VOL(I,J)+VOL(I,J+1))
C
C
C
C
      COMPUTE CELL SURFACE VALUES
C
      UNN=0.5*(UE(I,J+1)+UE(I-1,J+1))
      VNN=0.5*(VN(I,J)+VN(I,J+1))
C
      USS=0.5*(UE(I,J)+UE(I-1,J))
      VSS=0.5*(VN(I,J)+VN(I-1,J))
C
      UEE=0.5*(UE(I,J)+UE(I,J+1))
      VEE=0.5*(VN(I,J)+VN(I+1,J))
C
      UVM=0.5*(UE(I-1,J)+UE(I-1,J+1))
      VVM=0.5*(VN(I,J)+VN(I-1,J))
C
      AKNN=AKE(I,J+1)
      AKSS=AKE(I,J)
      AKKE=0.25*(AKE(I,J)+AKE(I,J+1)+AKE(I+1,J+1)+AKE(I+1,J))
      AKKW=0.25*(AKE(I-1,J)+AKE(I-1,J+1)+AKE(I,J+1)+AKE(I,J))
C
      VINN=VIS(I,J+1)
      VISS=VIS(I,J)
      VIEE=0.25*(VIS(I,J)+VIS(I,J+1)+VIS(I+1,J+1)+VIS(I+1,J))
      VINM=0.25*(VIS(I-1,J)+VIS(I-1,J+1)+VIS(I,J+1)+VIS(I,J))
C
      DENN=DEN(I,J+1)
      DESS=DEN(I,J)
      DEEE=0.25*(DEN(I,J)+DEN(I,J+1)+DEN(I+1,J+1)+DEN(I+1,J))
      DEWM=0.25*(DEN(I-1,J)+DEN(I-1,J+1)+DEN(I,J+1)+DEN(I,J))
C
C
      COMPUTE CONVECTION MASS FLUX.
C
      FN=DENN*RN*(B21M*UNN+B22N*VNN)
      FS=DESS*RS*(B21S*USS+B22S*VSS)
      FE=DEE*RE*(B11E*UEE+B12E*VEE)
      FM=DEWM*RM*(B11M*UVM+B12M*VVM)
C
C
      COMPUTE DIFFUSION COEFFICIENTS
C
      DM=VINN*D22N/VOLM
      DS=VISS*D22S/VOLS
      DE=VIEE*D11E/VOLE
      DM=VINM*D11M/VOLM
C
      COMPUTE COEFFICIENTS OF SOURCE TERM
C
      SMP=FN+FS+FE+FM
      CP=DHAXI(0.00,SMP)
      CPO=CP
C
C
      ASSEMBLE MAIN COEFFICIENTS
C
      ANN(I,J)=DHAXI(DABS(0.500*FN),DN)-0.500*FN
      ASN(I,J)=DHAXI(DABS(0.500*FS),DS)+0.500*FS
C
      AEN(I,J)=DHAXI(DABS(0.500*FE),DE)-0.500*FE
      ANN(I,J)=DHAXI(DABS(0.500*FM),DM)+0.500*FM
C
      COEFFICIENTS OF PRESSURE TERM IN MOMENTUM EQUATION
C
      CAVN(I,J)=RP*B12P
      CAVN(I,J)=RP*B22P
C
      COMPUTE DERIVATIVE VALUES FOR SOURCE TERMS.
C
      DUDX=RP*(B11P*(UEE-UVM)+B21P*(UNN-USS))/VOLP
      DVDX=RP*(B11P*(VEE-VVM)+B21P*(VNN-VSS))/VOLP
      DKDX=RP*(B11P*(AKE-AKW)+B21P*(AKNN-AKSS))/VOLP
      DVIDX=RP*(B11P*(VIEE-VIWM)+B21P*(VINN-VISS))/VOLP
C
      KDUDY=RP*(B12P*(UEE-UVM)+B22P*(UNN-USS))/VOLP
      KDVDY=RP*(B12P*(VEE-VVM)+B22P*(VNN-VSS))/VOLP
      DVVIDY=RP*(B12P*(VIEE-VIWM)+B22P*(VINN-VISS))/VOLP
C
      DVNDEE=0.25*(VN(I+1,J+1)+VN(I,J+1)
      1 -VN(I+1,J-1)-VN(I,J-1))
      DVNDEM=0.25*(VN(I-1,J+1)+VN(I,J+1)
      1 -VN(I-1,J-1)-VN(I,J-1))
      DVNDKN=0.25*(VN(I+1,J+1)+VN(I+1,J)
      1 -VN(I-1,J+1)-VN(I-1,J))
      DVNDKS=0.25*(VN(I+1,J-1)+VN(I+1,J)
      1 -VN(I-1,J-1)-VN(I-1,J))
C
      SOURCE FUNCTION BY NON-ORTHOGONALITY.
C
      SBVNE=VIEE*D12E/VOLE*DVNDEE
      SBVNH=VIWM*D12H/VOLM*DVNDEW
      SBVNN=VINN*D21N/VOLN*DVNDKN
      SBVNS=VISS*D21S/VOLS*DVNDKS
C
      SBVN=SBVNE-SBVNH+SBVNN-SBVNS
C
      COMPUTE SOURCE FUNCTIONS
C
      SORCVN=DVIDX*DUDY+DVIDY*DVDX+DVIDY*DVDX+DVIDY*DVDX
C
      SCVN(I,J)=CPO*VN(I,J)+SORCVN*VOLP+SBVN
C
      SPVN(I,J)=-CP
C
      IF(ICORD.EQ.2) THEN
      SPVN(I,J)=SPVN(I,J)
      1 -0.5*(VIS(I,J)+VIS(I,J+1))*VOLP/RP/RP
      END IF
C
      380 CONTINUE
C
      *****
      CONSIDER THE BOUNDARY CONDITIONS
      *****
      OUTLET CONDITION
C
```

```

      B11P=YP(I,J+1)-YP(I,J)
      B12P=-XP(I,J+1)-XP(I,J)
      B21P=-(YC(I,J)-YC(I-1,J))
      B22P=XC(I,J)-XC(I-1,J)
      VOLP=0.5*(VOL(I,J)+VOL(I,J+1))
C
C
C
C
      COMPUTE CELL SURFACE VALUES
C
      UNN=0.5*(UE(I,J+1)+UE(I-1,J+1))
      VNN=0.5*(VN(I,J)+VN(I,J+1))
C
      USS=0.5*(UE(I,J)+UE(I-1,J))
      VSS=0.5*(VN(I,J)+VN(I-1,J))
C
      UEE=0.5*(UE(I,J)+UE(I,J+1))
      VEE=0.5*(VN(I,J)+VN(I+1,J))
C
      UVM=0.5*(UE(I-1,J)+UE(I-1,J+1))
      VVM=0.5*(VN(I,J)+VN(I-1,J))
C
      AKNN=AKE(I,J+1)
      AKSS=AKE(I,J)
      AKKE=0.25*(AKE(I,J)+AKE(I,J+1)+AKE(I+1,J+1)+AKE(I+1,J))
      AKKW=0.25*(AKE(I-1,J)+AKE(I-1,J+1)+AKE(I,J+1)+AKE(I,J))
C
      VINN=VIS(I,J+1)
      VISS=VIS(I,J)
      VIEE=0.25*(VIS(I,J)+VIS(I,J+1)+VIS(I+1,J+1)+VIS(I+1,J))
      VINM=0.25*(VIS(I-1,J)+VIS(I-1,J+1)+VIS(I,J+1)+VIS(I,J))
C
      DENN=DEN(I,J+1)
      DESS=DEN(I,J)
      DEEE=0.25*(DEN(I,J)+DEN(I,J+1)+DEN(I+1,J+1)+DEN(I+1,J))
      DEWM=0.25*(DEN(I-1,J)+DEN(I-1,J+1)+DEN(I,J+1)+DEN(I,J))
C
C
      COMPUTE CONVECTION MASS FLUX.
C
      FN=DENN*RN*(B21M*UNN+B22N*VNN)
      FS=DESS*RS*(B21S*USS+B22S*VSS)
      FE=DEE*RE*(B11E*UEE+B12E*VEE)
      FM=DEWM*RM*(B11M*UVM+B12M*VVM)
C
C
      COMPUTE DIFFUSION COEFFICIENTS
C
      DM=VINN*D22N/VOLM
      DS=VISS*D22S/VOLS
      DE=VIEE*D11E/VOLE
      DM=VINM*D11M/VOLM
C
      COMPUTE COEFFICIENTS OF SOURCE TERM
C
      SMP=FN+FS+FE+FM
      CP=DHAXI(0.00,SMP)
      CPO=CP
C
C
      ASSEMBLE MAIN COEFFICIENTS
C
      ANN(I,J)=DHAXI(DABS(0.500*FN),DN)-0.500*FN
      ASN(I,J)=DHAXI(DABS(0.500*FS),DS)+0.500*FS
C
      AEN(I,J)=DHAXI(DABS(0.500*FE),DE)-0.500*FE
      ANN(I,J)=DHAXI(DABS(0.500*FM),DM)+0.500*FM
C
      COEFFICIENTS OF PRESSURE TERM IN MOMENTUM EQUATION
C
      CAVN(I,J)=RP*B12P
      CAVN(I,J)=RP*B22P
C
      COMPUTE DERIVATIVE VALUES FOR SOURCE TERMS.
C
      DUDX=RP*(B11P*(UEE-UVM)+B21P*(UNN-USS))/VOLP
      DVDX=RP*(B11P*(VEE-VVM)+B21P*(VNN-VSS))/VOLP
      DKDX=RP*(B11P*(AKE-AKW)+B21P*(AKNN-AKSS))/VOLP
      DVIDX=RP*(B11P*(VIEE-VIWM)+B21P*(VINN-VISS))/VOLP
C
      KDUDY=RP*(B12P*(UEE-UVM)+B22P*(UNN-USS))/VOLP
      KDVDY=RP*(B12P*(VEE-VVM)+B22P*(VNN-VSS))/VOLP
      DVVIDY=RP*(B12P*(VIEE-VIWM)+B22P*(VINN-VISS))/VOLP
C
      DVNDEE=0.25*(VN(I+1,J+1)+VN(I,J+1)
      1 -VN(I+1,J-1)-VN(I,J-1))
      DVNDEM=0.25*(VN(I-1,J+1)+VN(I,J+1)
      1 -VN(I-1,J-1)-VN(I,J-1))
      DVNDKN=0.25*(VN(I+1,J+1)+VN(I+1,J)
      1 -VN(I-1,J+1)-VN(I-1,J))
      DVNDKS=0.25*(VN(I+1,J-1)+VN(I+1,J)
      1 -VN(I-1,J-1)-VN(I-1,J))
C
      SOURCE FUNCTION BY NON-ORTHOGONALITY.
C
      SBVNE=VIEE*D12E/VOLE*DVNDEE
      SBVNH=VIWM*D12H/VOLM*DVNDEW
      SBVNN=VINN*D21N/VOLN*DVNDKN
      SBVNS=VISS*D21S/VOLS*DVNDKS
C
      SBVN=SBVNE-SBVNH+SBVNN-SBVNS
C
      COMPUTE SOURCE FUNCTIONS
C
      SORCVN=DVIDX*DUDY+DVIDY*DVDX+DVIDY*DVDX+DVIDY*DVDX
C
      SCVN(I,J)=CPO*VN(I,J)+SORCVN*VOLP+SBVN
C
      SPVN(I,J)=-CP
C
      IF(ICORD.EQ.2) THEN
      SPVN(I,J)=SPVN(I,J)
      1 -0.5*(VIS(I,J)+VIS(I,J+1))*VOLP/RP/RP
      END IF
C
      380 CONTINUE
C
      *****
      CONSIDER THE BOUNDARY CONDITIONS
      *****
      OUTLET CONDITION
C
```

```

C      IF(I.EQ.IMAX) THEN
C      DO 381 J=3,JMAX
C      381 AEN(IMAX,J)=0.DO
C      END IF
C
C      *****
C      FINAL COEFF ASSEMBLY AND RESIDUAL SOURCE FOR VN
C      *****
C      DO 384 J=3,JMAX
C      APVN(I,J)=ANN(I,J)+ASN(I,J)+AEN(I,J)+AWN(I,J)-SPVN(I,J)
C      RESOR=ANN(I,J)*VN(I,J+1)+ASN(I,J)*VN(I,J)+AWN(I,J)*VN(I,J-1)
C      1 +AEN(I,J)*VN(I+1,J)+AWN(I,J)*VN(I,J)+VN(I-1,J)
C      2 -APVN(I,J)*VN(I,J)+SCVN(I,J)
C      3 +CKVN(I,J)*0.25*(PR(I-1,J+1)+PR(I-1,J)
C      4 -PR(I+1,J+1)-PR(I+1,J))
C      5 +CVVN(I,J)*(PR(I,J)-PR(I,J+1))
C      VOLP=0.5*(VOL(I,J)+VOL(I,J+1))
C      RESOR=GREAT*VOLP
C      IF(-SPVN(I,J).GT.0.5*SORVOL) RESOR=RESOR/SORVOL
C      RESVN=RESVN+DABS(RESOR)
C      UNDER-RELAXATION
C      SCVN(I,J)=SCVN(I,J)
C      1 +(1.DO-URFV)/URFV*APVN(I,J)*VN(I,J)
C      384 CONTINUE
C      *****
C      CALCULATE THE COEFFICIENTS AND SOURCE TERMS FOR
C      EAST CELL SURFACE VELOCITIES.
C      *****
C      DO 280 J=3,JMAX
C      COMPUTE GEOMETRIC CONSTANTS
C      FOR EAST CONTROL VOLUME SURFACE
C      RE=0.25*(R(I+1,J)+R(I+1,J-1)+R(I,J-1)+R(I,J))
C      B12E=YN(I+1,J)-YN(I+1,J-1)
C      B21E=-(XN(I+1,J)-XN(I+1,J-1))
C      B22E=XE(I+1,J)-YE(I,J)
C      D12E=RE*RE*(B12E*B12E+B22E*B22E)
C      D21E=RE*RE*(B12E*B21E+B21E*B22E)
C      VOL2E=VOL(I+1,J)
C      FOR WEST CONTROL VOLUME SURFACE
C      RW=0.25*(R(I,J)+R(I,J-1)+R(I-1,J-1)+R(I-1,J))
C      B12W=YN(I,J)-YN(I,J-1)

```

```

B12W=-(XN(I,J)-XN(I,J-1))
B21W=-(YE(I,J)-YE(I-1,J))
B22W=XE(I,J)-XE(I-1,J)
D11W=RW*RW*(B11W*B11W+B12W*B12W)
D12W=RW*RW*(B11W*B21W+B12W*B22W)
VOL1W=VOL(I,J)
FOR NORTH CONTROL VOLUME SURFACE
RN=R(I,J)
B11N=YE(I,J+1)-YE(I,J)
B12N=-(XE(I,J+1)-XE(I,J))
B21N=-(YN(I+1,J)-YN(I,J))
B22N=XN(I+1,J)-XN(I,J)
D21N=RN*RN*(B11N*B21N+B12N*B22N)
D22N=RN*RN*(B21N*B21N+B22N*B22N)
VOL1N=0.25*(VOL(I,J)+VOL(I,J+1)+VOL(I+1,J+1)+VOL(I+1,J))
FOR SOUTH CONTROL VOLUME SURFACE
RS=R(I,J-1)
B11S=YE(I,J)-YE(I,J-1)
B12S=-(XE(I,J)-XE(I,J-1))
B21S=-(YN(I+1,J-1)-YN(I,J-1))
B22S=XN(I+1,J-1)-XN(I,J-1)
D21S=RS*RS*(B11S*B21S+B12S*B22S)
D22S=RS*RS*(B21S*B21S+B22S*B22S)
VOL1S=0.25*(VOL(I,J-1)+VOL(I,J)+VOL(I+1,J)+VOL(I+1,J-1))
FOR CENTER POINT
RP=0.5*(R(I,J)+R(I,J-1))
B11P=YC(I,J)-YC(I,J-1)
B12P=-(XC(I,J)-XC(I,J-1))
B21P=-(YP(I+1,J)-YP(I,J))
B22P=XP(I+1,J)-XP(I,J)
VOL1P=0.5*(VOL(I,J)+VOL(I+1,J))
COMPUTE CELL SURFACE VALUES
UNN=0.5*(UE(I,J)+UE(I,J+1))
VNN=0.5*(VN(I,J)+VN(I+1,J))
USS=0.5*(UE(I,J)+UE(I,J-1))
VSS=0.5*(VN(I,J-1)+VN(I+1,J-1))
UEE=0.5*(UE(I,J)+UE(I+1,J))
VEE=0.5*(VN(I+1,J)+VN(I+1,J-1))
UWW=0.5*(UE(I,J)+UE(I-1,J))
VWW=0.5*(VN(I,J)+VN(I-1,J))
AKNN=0.25*(AKE(I,J)+AKE(I,J+1)+AKE(I+1,J+1)+AKE(I+1,J))
AKSS=0.25*(AKE(I,J-1)+AKE(I,J)+AKE(I+1,J)+AKE(I+1,J-1))
AKWW=AKE(I+1,J)
AKWW=AKE(I,J)
VINN=0.25D0*(VIS(I,J)+VIS(I,J+1)+VIS(I+1,J)+VIS(I+1,J+1))
VISS=0.25D0*(VIS(I,J)+VIS(I,J-1)+VIS(I+1,J)+VIS(I+1,J-1))
VIEE=VIS(I+1,J)

```





1 +2.\*SQRT3-2.)/CAPPA+5.45+3.7\*DPR)  
 IF (DABS (UTAUN-UTAU) .LT. 1.D-5) GO TO 556  
 UTAU=UTAU  
 555 CONTINUE

C 556 TAUW=2.\*UTAU\*UTAU  
 UTAUA(I)=UTAU  
 CFS(I)=TAUW  
 YPLUS=RENUM\*UTAU\*YE2(I)\*COSA

C SQRT2=DSQRT(1.DO+DTAU\*YPLUS)  
 Q2=UTAU\*((DLOG(4.\*(SORT2-1.)/(SORT2+1.)/DTAU)  
 +2.\*SQRT2-2.)/CAPPA+5.45+3.7\*DPR)  
 1 Q2XI=Q3XI\*Q2/Q3

C UE(I,2)=Q2XI\*XXI/DG11  
 VSF(I)=Q2XI\*YXI/DG11

C UTAUK=0.5\*(UTAUA(I-1)+UTAU)  
 AKE(I,2)=UTAUK\*UTAUK/(CMU\*\*0.5)  
 ADS(I,2)=UTAUK\*UTAUK\*UTAUK/(CAPPA\*YP2(I)\*COSA)

C V2=0.5\*(VSF(I)+VSF(I-1))  
 HN=0.5\*(VN3(I)-VN2(I))  
 HS=0.5\*(VN2(I)-VP2(I))  
 VN(I,2)=(HN\*V2+HS\*V0(I,3))/(HN+HS)

C 554 CONTINUE

C \*\*\*\*\*  
 C SOLUTION OF DIFFERENCE EQUATIONS FOR VN.  
 C \*\*\*\*\*

DO 388 ITUV=1,NSWPV  
 DO 389 J=3,JMAX  
 AA(J)=-ASN(I,J)  
 BB(J)=APVN(I,J)/URFV  
 CC(J)=-ANN(I,J)

389 DD(J)=AEN(I,J)\*VN(I+1,J)+AMN(I,J)\*VN(I-1,J)+SCVN(I,J)  
 +CXVN(I,J)\*0.25\*(PR(I-1,J+1)+PR(I-1,J)  
 -PR(I+1,J+1)-PR(I+1,J))  
 4 +CYVN(I,J)\*(PR(I,J)-PR(I,J+1))  
 5

C DD(3)=DD(3)-AA(3)\*VN(I,2)  
 DD(JMAX)=DD(JMAX)-CC(JMAX)\*VN(I,JMAX+1)

C CALL TRIDAG(3,JMAX,AA,BB,CC,DD,T)

C DO 390 J=3,JMAX  
 390 VN(I,J)=T(J)

C VN(I,JP1)=VN(I,JMAX)\*(R(I,JMAX)+R(I-1,JMAX))  
 1 VN(I,1)=0.  
 / (R(I,JMAX+1)+R(I-1,JMAX+1))

C IF(I.LT.M2) GO TO 388  
 VN(I,2)=VN(I,3)\*VN2(I)/VN3(I)

C 388 CONTINUE

C 283 CONTINUE

C IF(I.GE.M2) GO TO 554

C \*\*\*\*\*  
 C TWO-POINT WALL FUNCTION METHOD.  
 C \*\*\*\*\*

C IF(I.EQ.M2M1) THEN  
 RM=100000.  
 YXI=2.\*(YC(I,1)-YP(I,1))  
 XXI=2.\*(XC(I,1)-XP(I,1))  
 YET=YC(I,2)-YC(I,1)  
 XET=0.

C ELSE  
 RM=VC(I,1)  
 YXI=YP(I+1,1)-YP(I,1)  
 XXI=XP(I+1,1)-XP(I,1)  
 YET=YP(I,2)-YC(I,1)  
 XET=0.  
 END IF

C B11=RM\*YET  
 B12=-RM\*YXI  
 B21=-RM\*XET  
 B22=RM\*XXI

C G11=XXI\*XXI+YXI\*YXI  
 G22=XET\*XET+YET\*YET  
 G12=YXI\*YET+XXI\*XET  
 G=RM\*RM\*(XXI\*YET-XET\*YXI)\*(XXI\*YET-XET\*YXI)  
 GI=1./G

C A11=RM\*RM\*GI\*G22  
 A22=RM\*RM\*GI\*G11  
 A12=-RM\*RM\*GI\*G12  
 AJI=DSQRT(GI)

C DG11=DSQRT(G11)  
 COSA=(B12\*XET+B22\*YET)/DSQRT(G22\*G+A22)  
 GRADP=(PR(I+1,1)-PR(I,1))/(XP(I+1,1)-XP(I,1))\*COSA  
 IF(I.GE.M2.OR.NITER.LT.5) GRADP=0.

C U3=UE(I,3)  
 V3=0.25\*(VN(I,2)+VN(I+1,2)+VN(I,3)+VN(I+1,3))  
 Q3=DSQRT(U3\*U3+V3\*V3)  
 Q3XI=AJI\*DG11\*(B11\*U3+B21\*V3)

C UTAU=UTAUA(I)  
 IF(NITER.EQ.1) UTAU=UTAUA(I-1)

C DO 555 IJK=1,50  
 DPR=GRADP/(RENUM\*UTAU\*UTAU)  
 DPR=DMAXI(DPR,0.000100)  
 DTAU=0.5\*DPR  
 SQRT3=DSQRT(1.DO+DTAU\*RENUM\*UTAU\*YE3(I)\*COSA)  
 UTAUN=Q3/((DLOG(4.\*(SQRT3-1.)/(SQRT3+1.)/DTAU)

```

C *****
C SOLUTION OF DIFFERENCE EQUATIONS FOR UE.
C *****
C
C DO 285 ITUV=1,NSWPU
C DO 286 J=3,JMAX
C AA(J)=-ASE(I,J)
C BB(J)=APUE(I,J)/URFU
C CC(J)=-ANE(I,J)
C DD(J)=AEZ(I,J)*UE(I+1,J)+ANE(I,J)*UE(I-1,J)+SCUR(I,J)
C 1 +CXUE(I,J)*(PR(I,J)-PR(I+1,J))
C 2 +CYUE(I,J)*0.25*(PR(I,J-1)+PR(I+1,J-1)
C 3 -PR(I,J+1)-PR(I+1,J+1))
C
C DD(3)=DD(3)-AA(3)*UE(I,2)
C DD(JMAX)=DD(JMAX)-CC(JMAX)*UE(I,JMAX+1)
C
C CALL TRIDAG(3,JMAX,AA,BB,CC,DD,T)
C
C DO 287 J=3,JMAX
C 287 UE(I,J)=T(J)
C
C IF(I.LT.N2) GO TO 285
C
C UE(I,2)=((YE4SQ-YE2SQ)*UE(I,3)-(YE3SQ-YE2SQ)*UE(I,4))
C 1 / (YE4SQ-YE3SQ)
C UE(I,1)=(VE4SQ*UE(I,3)-YE3SQ*UE(I,4))
C 1 / (YE4SQ-YE3SQ)
C
C 285 CONTINUE
C
C *****
C CALCULATE THE MASS SOURCE FOR PRESSURE CORRECTION
C EQUATIONS
C *****
C
C DO 480 J=3,JMAX
C
C COMPUTE GEOMETRIC CONSTANTS
C
C FOR EAST CONTROL VOLUME SURFACE
C
C RE=0.5*(R(I,J)+R(I,J-1))
C B11E=YC(I,J)-YC(I,J-1)
C B12E=-(XC(I,J)-XC(I,J-1))
C
C FOR WEST CONTROL VOLUME SURFACE
C
C RN=0.5*(R(I-1,J)+R(I-1,J-1))
C B11W=YC(I-1,J)-YC(I-1,J-1)
C B12W=-(XC(I-1,J)-XC(I-1,J-1))
C
C FOR NORTH CONTROL VOLUME SURFACE
C
C RN=0.5*(R(I-1,J)+R(I,J))

```

```

B21N=-(YC(I,J)-YC(I-1,J))
B22N=XC(I,J)-XC(I-1,J)
FOR SOUTH CONTROL VOLUME SURFACE
C
C
C RS=0.5*(R(I-1,J-1)+R(I,J-1))
C B21S=-(YC(I,J-1)-YC(I-1,J-1))
C B22S=XC(I,J-1)-XC(I-1,J-1)
C
C COMPUTE FACE DENSITIES.
C
C DENN=0.5D0*(DEN(I,J)+DEN(I,J+1))
C DESS=0.5D0*(DEN(I,J)+DEN(I,J-1))
C DEEE=0.5D0*(DEN(I,J)+DEN(I+1,J))
C DEWM=0.5D0*(DEN(I,J)+DEN(I-1,J))
C
C COMPUTE CONVECTION MASS FLUX.
C
C FN=DENN*RN*B22N*VN(I,J)
C FS=DESS*RS*B22S*VN(I,J-1)
C FE=DEEE*RE*B11E*UE(I,J)
C FW=DEWM*RW*B11W*UE(I-1,J)
C
C SHP=FN-FS+FE-FW
C
C UNN=0.25D0*(UO(I,J)+UO(I,J+1)+UO(I-1,J)+UO(I-1,J+1))
C VSS=0.25D0*(VO(I,J)+VO(I,J-1)+VO(I-1,J)+VO(I-1,J-1))
C VES=0.25D0*(VO(I,J)+VO(I,J-1)+VO(I+1,J)+VO(I+1,J-1))
C VWN=0.25D0*(VO(I,J)+VO(I,J-1)+VO(I-1,J)+VO(I-1,J-1))
C
C FN=DENN*RN*B21N*UNN
C FS=DESS*RS*B21S*USS
C FE=DEEE*RE*B12E*VEE
C FW=DEWM*RW*B12W*VWW
C
C SNP=FN-FS+FE-FW
C
C SPP(I,J)=SHP+SNP
C
C COMPUTE SUM OF ABSOLUTE MASS SOURCE
C
C RESMM=RESMM+DABS(SPP(I,J))
C
C COMPUTE COEFFICIENTS OF PRESSURE
C AND PRESSURE CORRECTION.
C
C QXUE(I,J)=URFU*RE*B11E/APUE(I,J)
C QYVN(I,J)=URFV*RN*B22W/APVN(I,J)
C
C BCU(I,J)=DEEE*RE*B11E*QXUE(I,J)
C BCY(I,J)=DENN*RN*B22N*QYVN(I,J)
C
C 480 CONTINUE
C
C *****
C CONSIDER THE BOUNDARY CONDITIONS FOR
C THE PRESSURE AND PRESSURE CORRECTION EQUATION.
C *****
C
C

```

C C ALONG THE BOTTOM WALL AND SYMMETRY LINE.

C BCV(I,2)=0.D0

C C ALONG THE INLET

C IF(I.EQ.2) THEN

C DO 481 J=3,JMAX

C 481 BCU(I,J)=0.D0

C END IF

C C \*\*\*\*\*  
C C CALCULATE THE PRESSURE-CORRECTION FIELD  
C C \*\*\*\*\*

C ITPP=15

C DO 482 J=1,JPI

C PP(I-1,J)=0.

C 482 PP(I+1,J)=0.

C DO 483 ITER=1,ITPP

C CALL PRESU(PP,SPP,I,0)

C 483 CONTINUE

C C \*\*\*\*\*  
C C CORRECT THE VELOCITY FIELD.  
C C \*\*\*\*\*

C DO 170 J=3,JMAX

C 170 VN(I,J)=VN(I,J)+QYVN(I,J)\*(PP(I,J)-PP(I+1,J))

C VN(I,JPI)=VN(I,JMAX)\*(R(I,JMAX)+R(I-1,JMAX))

C 1 / (R(I,JMAX+1)+R(I-1,JMAX+1))

C VN(I,1)=0.

C DO 171 J=3,JMAX

C 171 UE(I,J)=UE(I,J)+QXUE(I,J)\*(PP(I,J)-PP(I+1,J))

C UE(I,JPI)=1.D0

C IF(I.GE.M2) THEN

C UE(I,2)=((YE4SQ-YE2SQ)\*UE(I,3)-(YE3SQ-YE2SQ)\*UE(I,4))

C 1 / (YE4SQ-YE3SQ)

C UE(I,1)=((YE4SQ\*UE(I,3)-YE3SQ\*UE(I,4))

C 1 / (YE4SQ-YE3SQ)

C VN(I,2)=VN(I,3)\*YN2(I)/YN3(I)

C END IF

C C \*\*\*\*\*  
C C OUTLET CONDITION FOR VELOCITIES.  
C C \*\*\*\*\*

C IF(I.EQ.IMAX) THEN

C DO 489 J=1,JPI

C UE(IP1,J)=UE(IMAX,J)

C 489 VN(IP1,J)=VN(IMAX,J)

C END IF

C C \*\*\*\*\*  
C C SAVE CONTINUITY SATISFYING VELOCITIES.  
C C \*\*\*\*\*

C DO 689 J=1,JPI

C UO(I,J)=UE(I,J)

C 689 VO(I,J)=VN(I,J)

C C \*\*\*\*\*  
C C RETURN POINT OF INNER ITERATION.  
C C \*\*\*\*\*

C 2000 CONTINUE

C C \*\*\*\*\*  
C C CALCULATE THE PSEUDO-VELOCITY AND THE  
C C ASSOCIATED MASS SOURCE  
C C \*\*\*\*\*

C CALCULATE PSEUDO-VELOCITIES.

C DO 490 J=3,JMAX

C 490 URE(I,J)=-(ANE(I,J)\*UE(I,J+1)+ASE(I,J)\*UE(I,J-1)

C 1 +AEE(I,J)\*UE(I+1,J)+AWE(I,J)\*UE(I-1,J)

C 2 +SCUE(I,J)/(APUE(I,J)/URFU)

C 3 +CVUE(I,J)/(APUE(I,J)/URFU)

C 4 \*0.25\*(PR(I,J-1)+PR(I+1,J-1)

C 5 -PR(I,J+1)-PR(I+1,J+1))

C DO 491 J=3,JMAX

C 491 VHN(I,J)=(ANN(I,J)\*VN(I,J+1)+ASN(I,J)\*VN(I,J-1)

C 1 +AEN(I,J)\*VN(I+1,J)+AVN(I,J)\*VN(I-1,J)

C 2 +SCVN(I,J)/(APVN(I,J)/URFV)

C 3 +CXVN(I,J)/(APVN(I,J)/URFV)

C 4 \*0.25\*(PR(I-1,J+1)+PR(I-1,J)

C 5 -PR(I+1,J+1)-PR(I+1,J))

C VHN(I,2)=VN(I,2)

C VHN(I,JPI)=VN(I,JPI)

C C CALCULATE MASS-SOURCE OF PRESSURE EQUATION.

C DO 493 J=3,JMAX

```

C C COMPUTE GEOMETRIC CONSTANTS
C C FOR EAST CONTROL VOLUME SURFACE
C C RE=0.5*(R(I,J)+R(I,J-1))
C C B11E=YC(I,J)-YC(I,J-1)
C C B12E=-(XC(I,J)-XC(I,J-1))
C C FOR WEST CONTROL VOLUME SURFACE
C C RW=0.5*(R(I-1,J)+R(I-1,J-1))
C C B11W=YC(I-1,J)-YC(I-1,J-1)
C C B12W=-(XC(I-1,J)-XC(I-1,J-1))
C C FOR NORTH CONTROL VOLUME SURFACE
C C RN=0.5*(R(I,J)+R(I,J+1))
C C B21N=-(YC(I,J)-YC(I,J+1))
C C B22N=XC(I,J)-XC(I,J+1)
C C FOR SOUTH CONTROL VOLUME SURFACE
C C RS=0.5*(R(I,J)+R(I,J-1))
C C B21S=-(YC(I,J)-YC(I,J-1))
C C B22S=XC(I,J)-XC(I,J-1)
C C COMPUTE FACE DENSITIES.
C C DENH=0.5D0*(DEN(I,J)+DEN(I,J+1))
C C DEES=0.5D0*(DEN(I,J)+DEN(I,J-1))
C C DEEW=0.5D0*(DEN(I,J)+DEN(I,J-1))
C C DENW=0.5D0*(DEN(I,J)+DEN(I,J+1))
C C COMPUTE CONVECTION MASS FLUX.
C C FN=DENN*RN*B22N*VHN(I,J)
C C FS=DESS*RS*B22S*VHN(I,J-1)
C C FE=DEEE*RE*B11E*UHE(I,J)
C C FW=DEWW*RW*B11W*UHE(I-1,J)
C C SMP=FN-FS+FE-FW
C C UHN=0.25D0*(UO(I,J)+UO(I,J+1)+UO(I-1,J)+UO(I-1,J+1))
C C USS=0.25D0*(UO(I,J)+UO(I,J-1)+UO(I-1,J)+UO(I-1,J-1))
C C VEE=0.25D0*(VO(I,J)+VO(I,J-1)+VO(I-1,J)+VO(I-1,J-1))
C C VWN=0.25D0*(VO(I,J)+VO(I,J+1)+VO(I-1,J)+VO(I-1,J+1))
C C FN=DENN*RN*B21N*UHN
C C FS=DESS*RS*B21S*USS
C C FE=DEEE*RE*B12E*VEE
C C FW=DEWW*RW*B12W*VWN
C C SMP=FN-FS+FE-FW
C C SPR(I,J)=SMP/SNP
C C 493 CONTINUE
C C

```

A-34

```

*****
CALCULATE THE COEFFICIENTS AND SOURCE FUNCTIONS
FOR TURBULENCE KINETIC ENERGY
AND DISSIPATION RATE EQUATIONS.
*****
DO 580 J=3,JMAX
COMPUTE GEOMETRIC CONSTANTS
FOR EAST CONTROL VOLUME SURFACE
RE=0.5*(R(I,J)+R(I,J-1))
B11E=YC(I,J)-YC(I,J-1)
B12E=-(XC(I,J)-XC(I,J-1))
B21E=-(YC(I+1,J)-YP(I,J))
B22E=XP(I+1,J)-XP(I,J)
D11E=RE*RE*(B11E*B11E+B12E*B12E)
D12E=RE*RE*(B11E*B21E+B12E*B22E)
VOLE=0.5*(VOL(I,J)+VOL(I+1,J))
FOR WEST CONTROL VOLUME SURFACE
RW=0.5*(R(I-1,J)+R(I-1,J-1))
B11W=YC(I-1,J)-YC(I-1,J-1)
B12W=-(XC(I-1,J)-XC(I-1,J-1))
B21W=-(YP(I,J)-YP(I-1,J))
B22W=XP(I,J)-XP(I-1,J)
D11W=RW*RW*(B11W*B11W+B12W*B12W)
D12W=RW*RW*(B11W*B21W+B12W*B22W)
VOLW=0.5*(VOL(I,J)+VOL(I-1,J))
FOR NORTH CONTROL VOLUME SURFACE
RN=0.5*(R(I-1,J)+R(I,J))
B11N=YP(I,J+1)-YP(I,J)
B12N=-(XP(I,J+1)-XP(I,J))
B21N=-(YC(I,J)-YC(I-1,J))
B22N=XC(I,J)-XC(I-1,J)
D21N=RN*RN*(B11N*B21N+B12N*B22N)
D22N=RN*RN*(B21N*B21N+B22N*B22N)
VOLN=0.5*(VOL(I,J)+VOL(I,J+1))
FOR SOUTH CONTROL VOLUME SURFACE
RS=0.5*(R(I-1,J-1)+R(I,J-1))
B11S=YP(I,J)-YP(I,J-1)
B12S=-(XP(I,J)-XP(I,J-1))
B21S=-(YC(I,J-1)-YC(I-1,J-1))
B22S=XC(I,J-1)-XC(I-1,J-1)
D21S=RS*RS*(B11S*B21S+B12S*B22S)
D22S=RS*RS*(B21S*B21S+B22S*B22S)
VOLS=0.5*(VOL(I,J-1)+VOL(I,J))
FOR CENTER POINT
RP=0.25*(R(I,J)+R(I-1,J)+R(I-1,J-1)+R(I,J-1))
B11P=YN(I,J)-YN(I,J-1)

```



```

C C      END IF
C C      COMPUTE SOURCE FUNCTIONS
C C      FOR KINETIC ENERGY EQUATION.
C C      SCK(I,J)=CPO*AKE(I,J)+SBAKE
C C      SUND(I,J)=SCK(I,J)
C C      SCK(I,J)=SCK(I,J)+GEN(I,J)*VOLP
C C      SPK(I,J)=CP
C C      SPKD(I,J)=SPK(I,J)
C C      SPK(I,J)=SPK(I,J)
C C      1 -DEN(I,J)*ADS(I,J)/AKE(I,J)*VOLP
C C      FOR DISSIPATION RATE EQUATION.
C C      SCD(I,J)=CPO*ADS(I,J)+SBADES
C C      SUDD(I,J)=SCD(I,J)
C C      SCD(I,J)=SCD(I,J)
C C      1 +DEN(I,J)*C1*GEN(I,J)*ADS(I,J)*VOLP/AKE(I,J)
C C      SPD(I,J)=CP
C C      SPDD(I,J)=SPD(I,J)
C C      SPD(I,J)=SPD(I,J)
C C      1 -C3*DEN(I,J)*ADS(I,J)*VOLP/AKE(I,J)
C C      580 CONTINUE
C C      *****
C C      CONSIDER THE BOUNDARY CONDITIONS
C C      *****
C C      ALONG THE OUTLET.
C C      IF(I.EQ.IMAX) THEN
C C      DO 582 J=3,JMAX
C C      AKE(IMAX,J)=0.DO
C C      582 AED(IMAX,J)=0.DO
C C      END IF
C C      *****
C C      FINAL COEFF ASSEMBLY AND RESIDUAL SOURCE FOR AKE
C C      *****
C C      DO 583 J=3,JMAX
C C      APK(I,J)=ANK(I,J)+ASK(I,J)+AWK(I,J)+AKK(I,J)-SPK(I,J)
C C      RESOR=ANK(I,J)+AKE(I,J+1)+ASK(I,J)+AKE(I,J-1)
C C      +AEK(I,J)+AKE(I+1,J)+AWK(I,J)+AKE(I-1,J)
C C      -APK(I,J)+AKE(I,J)+SCK(I,J)
C C      VOLP=VOL(I,J)
C C      SORVOL=GREAT*VOLP
C C      IF(-SPK(I,J).GT.0.5*SORVOL) RESOR=RESOR/SORVOL
C C      RESAK=RESAK+DABS(RESOR)
C C      *****
C C      UNDER-RELAXATION
C C      APK(I,J)=APK(I,J)/URFK
C C      SCK(I,J)=SCK(I,J)+(1.DO-URFK)*APK(I,J)+AKE(I,J)
C C      583 CONTINUE
C C      *****
C C      SOLUTION OF DIFFERENCE EQUATIONS FOR AKE
C C      *****
C C      DO 584 ITUV=1,NSWPK
C C      DO 585 J=3,JMAX
C C      AA(J)=ASK(I,J)
C C      BB(J)=APK(I,J)
C C      CC(J)=ANK(I,J)
C C      585 DD(J)=AEK(I,J)+AKE(I+1,J)+AWK(I,J)+AKE(I-1,J)
C C      +SCK(I,J)
C C      DD(3)=DD(3)-AA(3)*AKE(I,2)
C C      DD(JMAX)=DD(JMAX)-CC(JMAX)*AKE(I,JMAX+1)
C C      CALL TRIDAG(3,JMAX,AA,BB,CC,DD,T)
C C      DO 586 J=3,JMAX
C C      586 AKE(I,J)=T(J)
C C      AKE(I,JMAX+1)=AKE(I,JMAX)
C C      IF(I.IT.N2) GO TO 584
C C      AKE(I,2)=(YP4SQ-YP2SQ)*AKE(I,3)-(YP3SQ-YP2SQ)*AKE(I,4)
C C      1 / (YP4SQ-YP3SQ)
C C      AKE(I,1)=(YP4SQ*AKE(I,3)-YP3SQ*AKE(I,4))
C C      1 / (YP4SQ-YP3SQ)
C C      AKE(I,2)=AKE(I,3)
C C      AKE(I,1)=AKE(I,3)
C C      584 CONTINUE
C C      *****
C C      FINAL COEFF ASSEMBLY AND RESIDUAL SOURCE FOR ADS
C C      *****
C C      DO 683 J=3,JMAX
C C      APD(I,J)=AND(I,J)+ASD(I,J)+AED(I,J)+AWD(I,J)+AWD(I,J)-SPD(I,J)
C C      RESOR=AND(I,J)+ADS(I,J+1)+ASD(I,J)+ADS(I,J-1)
C C      +AED(I,J)+ADS(I+1,J)+AWD(I,J)+ADS(I-1,J)
C C      -APD(I,J)+ADS(I,J)+SCD(I,J)
C C      VOLP=VOL(I,J)
C C      SORVOL=GREAT*VOLP
C C      IF(-SPD(I,J).GT.0.5*SORVOL) RESOR=RESOR/SORVOL
C C      RESAD=RESAD+DABS(RESOR)
C C      *****

```

```

C C UNDER-RELAXATION
C C APD(I,J)=APD(I,J)/URFD
C C SCD(I,J)=SCD(I,J)/(1.DO-URFD)*APD(I,J)*ADS(I,J)
C C
C C 683 CONTINUE
C C *****
C C SOLUTION OF DIFFERENCE EQUATIONS FOR ADS
C C *****
C C DO 684 ITUV=1,NSWPD
C C DO 685 J=3,JMAX
C C AA(J)=-ASD(I,J)
C C BB(J)=APD(I,J)
C C CC(J)=-AND(I,J)
C C 685 DD(J)=ASD(I,J)*ADS(I+1,J)+AND(I,J)*ADS(I-1,J)
C C +SCD(I,J)
C C
C C DD(J)=DD(J)-AA(J)*ADS(I,2)
C C DD(JMAX)=DD(JMAX)-CC(JMAX)*ADS(I,JMAX+1)
C C
C C CALL TRIDAG(3,JMAX,AA,BB,CC,DD,T)
C C
C C DO 686 J=3,JMAX
C C 686 ADS(I,J)=T(J)
C C
C C ADS(I,JMAX+1)=ADS(I,JMAX)
C C
C C IF(I.LT.M2) GO TO 684
C C
C C ADS(I,2)=((YP4SQ-YP2SQ)*ADS(I,3)-(YP3SQ-YP2SQ)*ADS(I,4))
C C /((YP4SQ-YP3SQ))
C C 1 ADS(I,1)=((YP4SQ*ADS(I,3)-YP3SQ*ADS(I,4))
C C /((YP4SQ-YP3SQ))
C C
C C ADS(I,2)=ADS(I,2)
C C ADS(I,1)=ADS(I,3)
C C
C C 684 CONTINUE
C C
C C IF(I.EQ.IMAX) THEN
C C DO 56 J=1,JPI
C C ARE(IP1,J)=ARE(IMAX,J)
C C ADS(IP1,J)=ADS(IMAX,J)
C C 56 CONTINUE
C C END IF
C C
C C 688 CONTINUE
C C
C C *****
C C PRINT SOME VARIABLES ON THE TERMINAL
C C AND RETURN SPACE MARCHING PROCESS.
C C *****
C C PRINT 9997, I,XE(I,1),UE(I,2),YPLUS,UTAU(I)
C C 9997 FORMAT(5X, I=,I3,2X, X=,E11.4,2X, 'UE2=',E11.4,2X,

```

```

1 'YPLUS=',E11.4,2X, 'UTAU=',E11.4)
C C 3000 CONTINUE
C C *****
C C PRINT GLOBAL ERRORS IN EACH ITERATION
C C *****
C C RESMM=RESMM/FLOWIN
C C RESUE=RESUE/XMONIN
C C RESVN=RESVN/XMONIN
C C RESAK=RESAK/(0.5DO*FLOWIN*UMEAN*UMZAN)
C C
C C PRINT 701, NITER, RESMM, RESUE, RESVN, RESAK
C C 701 FORMAT(3X, IT=',I5,2X, 'ERRM=',E11.4,2X, 'ERRUE=',E11.4
C C ,2X, 'ERRVN=',E11.4,2X, 'ERRK=',E11.4)
C C
C C STORE THE ERRORS
C C
C C WRITE(9,7010) NITER, RESMM, RESUE, RESVN, RESAK
C C 7010 FORMAT(3X, I5,2X, E11.4,2X, E11.4,2X, E11.4,2X, E11.4)
C C *****
C C UPDATE THE PRESSURE FIELD BY GLOBAL ITERATIONS
C C *****
C C IF(NITER.LT.ITPR) GO TO 1234
C C
C C ITERP=10
C C
C C DO 1660 I=1,IP1
C C DO 1660 J=1,JPI
C C 1660 PRO(I,J)=PR(I,J)
C C
C C DO 4001 ITERG=1,20
C C
C C DO 661 II=2,IMAX
C C I=IMAX+2-II
C C DO 660 ITER=1,ITERP
C C 660 CALL PRESU(PR,SPR,I,1)
C C
C C IF(I.LT.M2) THEN
C C PR(I,2)=((YP(I,4)-YP(I,2))*PR(I,3)-(YP(I,3)-YP(I,2))
C C *PR(I,4))/(YP(I,4)-YP(I,3))
C C PR(I,1)=((YP(I,4)-YP(I,1))*PR(I,3)-(YP(I,3)-YP(I,1))
C C *PR(I,4))/(YP(I,4)-YP(I,3))
C C ELSE
C C PR(I,2)=PR(I,2)
C C PR(I,1)=PR(I,3)
C C END IF
C C
C C 661 CONTINUE
C C
C C DO 954 J=1,JPI
C C 954 PR(IP1,J)=PR(IMAX,J)

```



```

C 4001 CONTINUE
C DO 9999 I=1,IP1
DO 9999 J=1,JP1
9999 PR(I,J)=PRO(I,J)+URFPR*(PR(I,J)-PRO(I,J))
C 1234 CONTINUE
C
C SELECT NUMBER OF ITERATION FOR OUTPUT
C ITPRINT=MAXIT-ITOUT
C IF(NITER.LE.ITPRINT) GO TO 6666
C
C *****
C OUTPUT FOR EACH ITERATION
C *****
C
C WRITE(8,4111) NITER
4111 FORMAT(/5X,'NO. OF ITERATION =',I5,
15X,'WALL OR CENTERLINE PRESSURE')
WRITE(8,2009)(PR(I,1),I=2,IMAX)
C
C WRITE(8,4112)
4112 FORMAT(/5X,'FRICTION VELOCITY')
WRITE(8,2009)(UE(I,1),I=M2,IMAX)
C
C WRITE(8,4113)
4113 FORMAT(/5X,'CENTERLINE VELOCITY')
WRITE(8,2009)(UE(I,1),I=M2,IMAX)
C
C 6666 CONTINUE
C CHECK FOR OVERFLOW.
C DO 4999 I=1,IP1
DO 4999 J=1,JP1
IF(DABS(PR(I,J)).GT.10.DO) GO TO 5555
IF(DABS(UE(I,J)).GT.10.DO) GO TO 5555
4999 CONTINUE
C
C *****
C PROVIDE DATA FOR CONVERGENCE HISTORY
C *****
C IF(NITER.EQ.1.OR.NITER.EQ.20.OR.NITER.EQ.50.OR.
1 NITER.EQ.100.OR.NITER.EQ.150.OR.NITER.EQ.200.
2 OR.NITER.EQ.300) THEN
C WRITE(10,4114) NITER
4114 FORMAT(/5X,I5)
C WRITE(10,2009)(PR(I,1),I=2,IMAX)

```

00  
C  
I  
K

```

C WRITE(10,2009)(UTAU(I),I=2,M2M1)
C WRITE(10,2009)(UE(I,1),I=M2,IMAX)
C END IF
C IF(NITER.GE.MAXIT) GO TO 702
C GO TO 4000
C 702 CONTINUE
C *****
C OUTPUT AFTER CALCULATION
C *****
C WRITE(8,2097) URFU,URFV,URFK,URFD,URFVIS,URFPR
2097 FORMAT(/5X,'RELAXATION FACTORS(U,V,K,E,VIS,PR) =',
1 6F6.3)
C DO 3333 I=1,IP1
C WRITE(8,12)NITER,I,XP(I,1),URFU
12 FORMAT(/5X,'NO. OF ITERATION =',I3,5X,'STATION',I3,
1,5X,'X =',F8.4,5X,'U-RF =',F6.3)
C IF(I.GE.M2) GO TO 2222
C 2098 WRITE(8,2099) UTAUA(I),CFS(I)
2099 FORMAT(5X,'UTAU=',E12.4,5X,'TAUM=',E12.4)
C 2222 WRITE(8,3001)
C 3001 FORMAT(/5X,'Y-CORD',6X,'VEL U',7X,'VEL V',6X,
1 'TURB KE',5X,'TURB DIS',3X,'PRESSURE'//)
C DO 3003 J=1,JP1
3003 WRITE(8,2007) YP(I,J),UE(I,J),VN(I,J)
1 'AKE(I,J),ADS(I,J),PR(I,J)
C 3333 CONTINUE
C 2117 FORMAT(4E12.4)
2007 FORMAT(6E12.4)
C
C SAVE DATA FOR PLOTTING.
C WRITE(7,2009)(XP(I,1),I=1,IP1)
WRITE(7,2009)(XE(I,1),I=1,IP1)
WRITE(7,2009)(YP(I,1),I=1,IP1)
WRITE(7,2009)(YN(I,1),I=1,IP1)
WRITE(7,2009)(PR(I,1),I=2,IMAX)
WRITE(7,2009)(UE(I,1),I=M2,IMAX)

```

```

C      WRITE(7,2009) (UTAU(A(I),I=2,M2H1)
C      DO 5001 I=1,IPI
C      DO 5001 J=1,JPI
5001  WRITE(7,2005)UE(I,J),VM(I,J),PR(I,J)
C      1  ,AKE(I,J),ADS(I,J),VIS(I,J)
C      2005  FORMAT(6E12.4)
C      2009  FORMAT(6E12.4)
C      C
C      C
C      5555  CONTINUE
C      CLOSE(7)
C      CLOSE(8)
C      CLOSE(9)
C      CLOSE(10)
C      STOP
C      END
C      C
C      SUBROUTINE TRIDAG(IF,L,A,B,C,D,V)
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C      PARAMETER(JD=45)
C      DIMENSION A(JD),B(JD),C(JD),D(JD),V(JD),BETA(JD),GAMMA(JD)
C      BETA(IF)=B(IF)
C      GAMMA(IF)=D(IF)/BETA(IF)
C      IFPI=IF+1
C      DO 1 I=IFPI,L
C      BETA(I)=B(I)-A(I)*C(I-1)/BETA(I-1)
C      1  GAMMA(I)=(D(I)-A(I)*GAMMA(I-1))/BETA(I)
C      V(I)=GAMMA(L)
C      LAST=L-IF
C      DO 2 K=1, LAST
C      I=L-K
C      2  V(I)=GAMMA(I)-C(I)*V(I+1)/BETA(I)
C      RETURN
C      END
C      C
C      SUBROUTINE PRESU(P,DHS,I,H)
C      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C      PARAMETER(ID=142,JD=45)
C      DIMENSION P(ID,JD)
C      DIMENSION DHS(ID,JD)
C      COMMON/PRES/BCU(ID,JD),BCV(ID,JD)
C      COMMON/AAA/AA(JD),BB(JD),CC(JD),DD(JD),T(JD)
C      COMMON/CONS1/IMAX,JMAX,IPI,JPI,M2,M2H1
C      DO 660 J=3,JMAX
C      AA(J)=-BCV(I,J-1)
C      BB(J)=-BCU(I-1,J)+M+BCU(I,J)+BCV(I,J-1)
C      1  +BCV(I,J)
C      CC(J)=-BCV(I,J)
C      660  DD(J)=-BCU(I,J)*P(I+1,J)+BCU(I-1,J)*P(I-1,J)
C      1  -DHS(I,J)
C      C

```

```

C      CALL TRIDAG(3,JMAX,AA,BB,CC,DD,T)
C      DO 671 J=3,JMAX
C      671  P(I,J)=T(J)
C      RETURN
C      END

```





```

C *****
C * NUMERICAL GENERATION OF BODY FITTED COORDINATES *
C * OF AXISYMMETRIC BODIES FOR TURBULENT CALCULATION. *
C *****
C
C IMPLICIT REAL*(A-H,O-Z)
C
C CHARACTER*10 F1NAME(6)
C CHARACTER*10 F2NAME(6)
C CHARACTER*10 F3NAME(6)
C
C COMMON/AAA/XC(200,70),YC(200,70)
C COMMON/XXX/X(200),Y(200,70)
C COMMON/ABC/AY(70),BY(70),CY(70),DY(70),TY(70)
C COMMON/YYY/RX(200),RY(200),FR(200),DY1(70)
C COMMON/BBB/AA(200),BB(200),CC(200),DD(200),A(200)
C COMMON/AAA1/AL1(200,70),AL2(200,70),A22(200,70),AJ(200,70)
C COMMON/AAA2/FL(200),F2(200,70)
C COMMON/AAA3/FX(200,70),FY(200,70)
C
C DATA F1NAME(1)/'FA1BODY',F1NAME(2)/'FA2BODY',
C $ F1NAME(3)/'FA3BODY',F1NAME(4)/'FA4BODY',
C $ F1NAME(5)/'FA5BODY',F1NAME(6)/'FA6BODY'//
C
C DATA F2NAME(1)/'FALBOUT',F2NAME(2)/'FA2BOUT',
C $ F2NAME(3)/'FALBOUT',F2NAME(4)/'FA4BOUT',
C $ F2NAME(5)/'FALBOUT',F2NAME(6)/'FALBOUT'//
C
C DATA F3NAME(1)/'FALBDAT',F3NAME(2)/'FA2BDAT',
C $ F3NAME(3)/'FALBDAT',F3NAME(4)/'FA4BDAT',
C $ F3NAME(5)/'FALBDAT',F3NAME(6)/'FALBDAT'//
C *****
C * AXISYMMETRIC BODIES. *
C * ***** *
C * IBODY-1 ; AFTERBODY1 OF DTNDSRDC (TURBULENT) . *
C * IBODY-2 ; AFTERBODY2 OF DTNDSRDC (TURBULENT) . *
C * IBODY-3 ; AFTERBODY3 OF DTNDSRDC (TURBULENT) . *
C * IBODY-5 ; AFTERBODY5 OF DTNDSRDC (TURBULENT) . *
C *****
C READ BODY TYPE FROM THE TERMINAL.
C *****
C WRITE(*,*) 'WHAT KIND OF BODY DO YOU WANT ?'
C WRITE(*,*) 'TYPE 1 FOR AFTERBODY1 OF DTNDSRDC'
C WRITE(*,*) 'TYPE 2 FOR AFTERBODY2 OF DTNDSRDC'
C WRITE(*,*) 'TYPE 3 FOR AFTERBODY3 OF DTNDSRDC'
C WRITE(*,*) 'TYPE 5 FOR AFTERBODY5 OF DTNDSRDC'
C

```

```

C READ(*,*) IBODY
C
C REYNOLDS NUMBERS AND FRICTION VELOCITIES
C AT THE STARTING POINT OF THE BODY FROM EXPERIMENTAL DATA.
C
C
C JMAX=52
C NRP=45
C RATIO=10.**0.125
C YPLUS=0.1
C
C IF (IBODY.EQ.1) THEN
C RE=6600000.D0
C UTAU=0.04D0
C IMA=12
C IMB=52
C IMAX=84
C
C ELSE IF (IBODY.EQ.2) THEN
C RE=6800000.D0
C UTAU=0.038D0
C IMA=25
C IMB=52
C IMAX=84
C
C ELSE IF (IBODY.EQ.3) THEN
C RE=6000000.D0
C UTAU=0.038D0
C IMA=32
C IMB=52
C IMAX=89
C
C ELSE IF (IBODY.EQ.5) THEN
C RE=9300000.D0
C UTAU=0.038D0
C IMA=26
C IMB=52
C IMAX=84
C
C END IF
C
C NUMBER OF GRID POINTS IN X-DIRECTION.
C
C IPI=IMAX+1
C IPI=IMAX-1
C *****
C READ BODY SHAPE AND X-COORDINATE.
C *****
C OPEN(UNIT=7, FILE=FINAME(IBODY))
C READ(7,1111)(X(I),I=1,IPI)
C READ(7,1111)(RX(I),I=1,IPI)
C 1111 FORMAT(4E14.6)
C CLOSE(7)
C
C MODIFY THE LEADING EDGE AND TRAILING EDGE.
C

```





```
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      JRP=22
C      IWAKE=52
C      JMAX=52
C      IPI=JMAX+1
C      JPI=JMAX+1

C      IFA=2
C      ITURD=1

C      IZND=999
C      ITM=1
C      ITERA=J

C      EMAX=30.00
C      FMAX=35.00
C      EPFE=0.000100
C      EPFL=0.0000100
C      FI=3.14159265358979300

C      CNU=0.0900
C      CNU2=DSQRT(CNU)
C      CNU4=DSQRT(CNU2)
C      CNU3=CNU4*CNU4*CNU4
C      CNU2I=1.00/CNU2
C      CEPS1=1.4400
C      CEPS2=1.9200
C      CEPK=1.00
C      CEPS=1.300
C      CKAR=0.41800
C      E=9.00
C      REI=1.00/RE

C      FOR WOLFSHTEIN MODEL CONSTANT CORRECTED BY
C      BY PATEL AND CHEN.

C      AU=70.00
C      CL=CKAR/CNU**0.7500
C      AE=2.00*CL

C      FOR HASSID AND POREH MODEL.

C      CNU=0.54800
C      CD1=0.16400
C      CD2=0.33600
C      AMU=0.02900
C      AK=0.4100

C      *****
C      SET UP THE GEOMETRIC COORDINATES AND CALCULATE
C      THE GEOMETRIC COEFFICIENTS.
C      *****
C      *****
C      OFFSET OF AXISYMMETRIC BODY.
C      INPUT FROM THE GRID-GENERATION PROGRAM : A3BFC

```

7-1-80

```

C      THE GRID LINE FROM BODY-FITTED COORDINATE SYSTEM IS TREATED
C      AS U-VELOCITY CALCULATION POINT XU,YU.
C      XC,YC----- CONTROL VOLUME POINT
C      XU,YU----- U CALCULATION POINT
C      XV,YV----- V CALCULATION POINT
C      XP,YP----- P,K,E CALCULATION POINT
C      *****
C      OPEN(UNIT=11,FILE=FINAME(1BODYV))
C      READ(11,7009)((XU(I,J),I=1,IPI),J=1,JPI)
C      READ(11,7009)((YU(I,J),I=1,IPI),J=1,JPI)
C      READ(11,7009)((F1(I,J),I=1,IPI),J=1,JPI)
C      READ(11,7009)((F2(I,J),I=1,IPI),J=1,JPI)
C      7009 FORMAT(4D14.8)
C      CLOSE(11)
C      XC(1,1)=XU(1,1)
C      YC(1,1)=YU(1,1)
C      XV(1,1)=XU(1,1)
C      YV(1,1)=YU(1,1)
C      XP(1,1)=XU(1,1)
C      YP(1,1)=YU(1,1)
C      DO 10 J=2,JMAX
C      XC(I,J)=XU(I,J)
C      YC(I,J)=YU(I,J)
C      XV(I,J)=XU(I,J)
C      YV(I,J)=YU(I,J)
C      10 YV(1,J)=0.5D0*(YU(1,J)+YU(1,J+1))
C      DO 12 I=2,IPI
C      XC(I,1)=XU(I,1)
C      YC(I,1)=YU(I,1)
C      XV(I,1)=0.5D0*(XU(I,1)+XU(I-1,1))
C      YV(I,1)=0.5D0*(YU(I,1)+YU(I-1,1))
C      12 YV(I,1)=0.5D0*(YU(I,1)+YU(I-1,1))
C      DO 11 I=2,IPI
C      DO 11 J=2,JMAX
C      XP(I,J)=0.5D0*(XU(I,J)+XU(I-1,J))
C      YP(I,J)=0.5D0*(YU(I,J)+YU(I-1,J))
C      11 YV(I,J)=0.25D0*(YU(I,J)+YU(I-1,J))+YU(I,J+1)+YU(I-1,J+1))
C      DO 13 I=1,IPI
C      YU(I,JPI)=YU(I,JMAX)+0.5D0*(YU(I,JMAX)-YU(I,JMAX-1))
C      YV(I,JPI)=YV(I,JMAX)+0.5D0*(YV(I,JMAX)-YV(I,JMAX-1))
C      YP(I,JPI)=YP(I,JPI)
C      XV(I,JPI)=XV(I,JMAX)
C      YV(I,JPI)=YV(I,JMAX)
C      13 XP(I,JPI)=XP(I,JMAX)
C      DO 131 J=1,JPI
C      XU(IPI,J)=XP(IPI,J)

```

```

131 YU(IP1,J)=YP(IP1,J)
C
DO 14 I=1,IP1
  YP2(I)=YP(I,2)-YP(I,1)
  YP3(I)=YP(I,3)-YP(I,2)
  YP4(I)=YP(I,4)-YP(I,3)
  YU2(I)=YU(I,2)-YU(I,1)
  YU3(I)=YU(I,3)-YU(I,2)
  YU4(I)=YU(I,4)-YU(I,3)
  YV1(I)=0.5D0*(YP(I,2)+YV(I,1))-YV(I,1)
  YV2(I)=YV(I,2)-YV(I,1)
  YV3(I)=YV(I,3)-YV(I,2)
  YV4(I)=YV(I,4)-YV(I,3)
14 YV4(I)=YV(I,4)-YV(I,3)
C
DO 15 I=1,IP1
  YP2SQ(I)=YP2(I)*YP2(I)
  YP3SQ(I)=YP3(I)*YP3(I)
  YP4SQ(I)=YP4(I)*YP4(I)
  YU2SQ(I)=YU2(I)*YU2(I)
  YU3SQ(I)=YU3(I)*YU3(I)
  YU4SQ(I)=YU4(I)*YU4(I)
15 CONTINUE
C
DO 16 I=2,IMAX
DO 16 J=2,JMAX
R=YU(I,J)
YXI=0.5D0*(YU(I+1,J)-YU(I-1,J))
YET=0.5D0*(YU(I,J+1)-YU(I,J-1))
XXI=0.5D0*(XU(I+1,J)-XU(I-1,J))
XET=0.5D0*(XU(I,J+1)-XU(I,J-1))
CALL GEOCOE
B11U(I,J)=B11
B12U(I,J)=B12
B21U(I,J)=B21
B22U(I,J)=B22
A11U(I,J)=A11
A12U(I,J)=A12
A21U(I,J)=A21
A22U(I,J)=A22
16 AJIU(I,J)=AJI
C
DO 17 I=2,IMAX
DO 17 J=2,JMAX
R=YV(I,J)
YXI=0.5D0*(YV(I+1,J)-YV(I-1,J))
YET=0.5D0*(YV(I,J+1)-YV(I,J-1))
XXI=0.5D0*(XV(I+1,J)-XV(I-1,J))
XET=0.5D0*(XV(I,J+1)-XV(I,J-1))
CALL GEOCOE
B11V(I,J)=B11
B12V(I,J)=B12
B21V(I,J)=B21
B22V(I,J)=B22
A11V(I,J)=A11
A12V(I,J)=A12
A21V(I,J)=A21
A22V(I,J)=A22
17 AJIV(I,J)=AJI
C
DO 18 I=2,IMAX
DO 18 J=2,JMAX
R=YP(I,J)
YXI=0.5D0*(YP(I+1,J)-YP(I-1,J))
YET=0.5D0*(YP(I,J+1)-YP(I,J-1))
XXI=0.5D0*(XP(I+1,J)-XP(I-1,J))
XET=0.5D0*(XP(I,J+1)-XP(I,J-1))
CALL GEOCOE
B11P(I,J)=B11
B12P(I,J)=B12
B21P(I,J)=B21
B22P(I,J)=B22
A11P(I,J)=A11
A12P(I,J)=A12
A21P(I,J)=A21
A22P(I,J)=A22
18 AJIP(I,J)=AJI
C
DO 19 I=2,IP1
DO 19 J=2,JP1
R=0.5D0*(YC(1,J)+YC(1,J-1))
B11C(1,J)=(YC(1,J)-YC(1,J-1))*R
R=0.5D0*(YC(1,J)+YC(1,J-1))
B11C(1,J)=(YC(1,J)-YC(1,J-1))*R
B21C(1,J)=(XC(1,J)-XC(1,J-1))*R
R=0.5D0*(YC(1,J)+YC(1,J-1))
B12C(1,J)=(YC(1,J)-YC(1,J-1))*R
B22C(1,J)=(XC(1,J)-XC(1,J-1))*R
19 B22C(1,J)=(XC(1,J)-XC(1,J-1))*R
C
DO 191 I=2,IMAX
DO 191 J=2,JMAX
FUX(I,J)=-2.0D0*A11U(I,J)
  / (XU(I+1,J)-XU(I-1,J))
FUY(I,J)=-2.0D0*A22U(I,J)*F2(I,J)+1.0D0/B11U(I,J)
FVX(I,J)=-2.0D0*A11V(I,J)*(XV(I+1,J)-2.0D0*XV(I,J)+XV(I-1,J))
  / (XV(I+1,J)-XV(I-1,J))
FVY(I,J)=-0.5D0*A22V(I,J)
  + (F2(I,J)+F2(I,J+1)+F2(I-1,J)+F2(I-1,J+1))
  + 1.0D0/B11V(I,J)
FPX(I,J)=-2.0D0*A11P(I,J)*(XP(I+1,J)-2.0D0*XP(I,J)+XP(I-1,J))
  / (XP(I+1,J)-XP(I-1,J))
FPY(I,J)=-A22P(I,J)*(F2(I-1,J)+F2(I,J)+1.0D0/B11P(I,J))
191 CONTINUE
C
*****
C SET UP THE INITIAL CONDITIONS.
*****
DO 20 I=1,IP1
VSF(I)=0.0D0
DO 20 J=1,JP1
BCU(I,J)=0.0D0
BCV(I,J)=0.0D0
TNU(I,J)=0.0D0
PP(I,J)=0.0D0
PR(I,J)=0.0D0
20 PR(I,J)=0.0D0
C
*****
C SET UP THE INLET CONDITION.
*****
DELTA=0.006D0

```

```

          DELTAI=1.DO/DELTAX
          UO(1,1)=0.DO
          VO(1,1)=0.DO
          TREO(1,1)=0.DO
          UTAU=0.036DO
          UTAUA(1)=UTAU
          ETAK=(YU(1,JPI)-YU(1,1))*DELTAI
          DPLUS=DELTAI*RE*UTAU
          IF(DPLUS.LE.1000.DO) DPLUS=1001.DO
C
C      VELOCITY PROFILE FROM SPALDING'S INNER LAW.
C      (TAKEN FROM H.C.RICHMOND'S PH.D THESIS,1987)
C
          DO 935 J=2,JPI
          RR=(YU(1,J)-YU(1,1))
          ETA=RR*DELTAI
          YPLUS=RE*RR*UTAU
C
          IF(YPLUS.LT.10.DO) THEN
              UO(1,J)=UTAU*YPLUS
C
          ELSE IF(YPLUS.LT.50.DO) THEN
              UKO=DLOG(YPLUS)+CKAR*5.45DO
C
          DO 171 K=1,50
          DUM=DEXP(UKO)-1.DO-UKO-0.5DO*UKO**2
          FU=UKO/CKAR-YPLUS+0.10248DO*(DUM-UKO**3/6.DO)
          FPU=1.DO+AK*0.10248DO*DUM
          UK=UKO-FU*AK/FPU
          IF(DABS(UK-UKO).LT.1.DO-6) GO TO 181
          UKO=UK
          171 CONTINUE
          181 UO(1,J)=UK*UTAU/CKAR
C
          UO(1,J)=(5.DO*DLOG(YPLUS)-3.05DO)*UTAU
C
          ELSE IF(YPLUS.LT.1000.DO) THEN
              UO(1,J)=(2.5DO*DLOG(YPLUS)+5.5DO)*UTAU
C
          ELSE IF(YPLUS.LT.DPLUS) THEN
              UO(1,J)=UTAU*(2.5DO*DLOG(YPLUS)+5.5DO)
          C   +0.55DO*2.DO/0.41DO*(DSIN(PI/2.DO*RR/DELTAX))**2)
C
          ELSE
              UO(1,J)=1.DO
          END IF
          VO(1,J)=0.DO
          935 DO 111 J=2,JPI
C
          YPLUS=(YU(1,J)-YU(1,1))*RE*UTAU
          DPLUS=DELTAI*RE*UTAU
          UTAUSQ=UTAU*UTAU
C
          IF(YPLUS.LE.5.DO) THEN
              TREO(1,J)=0.05DO*YPLUS*YPLUS*UTAU*U
          ELSE IF(YPLUS.LE.15.DO) THEN

```

```

C C C *****
C C C DO 22 J=1,JPI
C C C V(1,J)=V0(1,J)
C C C VS(1,J)=V(1,J)
C C C 22 VH(1,J)=V(1,J)
C C C DO 23 J=1,JPI
C C C DO 23 I=1,IPI
C C C CU(I,J)=0.DO
C C C CV(I,J)=0.DO
C C C CUV(I,J)=0.DO
C C C DS(I,J)=0.DO
C C C 23 PP(I,J)=0.DO
C C C IF(IT.LT.4) GO TO 24
C C C DO 25 J=1,JPI
C C C V0(1,J)=V0(2,J)
C C C TKEO(1,J)=TKEO(2,J)
C C C TDSO(1,J)=TDSO(2,J)
C C C 25 CONTINUE
C C C 24 CONTINUE
C C C *****
C C C START SPACE MARCHING FROM UPSTREAM TO DOWN STREAM.
C C C *****
C C C DO 3000 I=2,IMAX
C C C JAU=2
C C C JAP=2
C C C JAV=2
C C C JA=2
C C C JAUH1=JAU-1
C C C JAVH1=JAV-1
C C C JAPH1=JAP-1
C C C JAH1=JA-1
C C C *****
C C C INITIAL GUESS FOR THE FIRST MARCHING PROCESS.
C C C *****
C C C IF(IT.GT.1) GO TO 26
C C C UTAUA(I)=UTAU(A(I-1))
C C C DO 27 J=1,JPI
C C C U(I,J)=U(I-1,J)
C C C V(I,J)=V(I-1,J)
C C C VS(I,J)=U(I,J)
C C C VS(I,J)=V(I,J)
C C C UO(I,J)=U(I,J)
C C C VO(I,J)=V(I,J)
C C C TKEO(I,J)=TKEO(I-1,J)
C C C TDSO(I,J)=TDSO(I-1,J)
C C C TKE(I,J)=TKEO(I,J)
C C C TDS(I,J)=TDSO(I,J)
C C C U(I+1,J)=U(I,J)

```

```

V(I+1,J)=V(I,J)
US(I+1,J)=US(I,J)
VS(I+1,J)=VS(I,J)
UO(I+1,J)=UO(I,J)
VO(I+1,J)=VO(I,J)
TKEO(I+1,J)=TKEO(I,J)
TDSO(I+1,J)=TDSO(I,J)
TKE(I+1,J)=TKE(I,J)
27 TDS(I+1,J)=TDS(I,J)
26 CONTINUE
C C C *****
C C C CALCULATE THE COSINE VALUE OF GRID LINE.
C C C *****88
C C C R=1.DO
C C C YXI=YC(I,1)-YC(I-1,1)
C C C XXI=XC(I,1)-XC(I-1,1)
C C C YET=YP(I,2)-YP(I,1)
C C C XET=0.DO
C C C CALL GEOCOE
C C C COSA=(B12*XET+B22*YET)/DSQRT(G22*G*A22)
C C C *****
C C C CALCULATE THE EDDY VISCOSITY.
C C C *****
C C C DO 28 J=2,JPI
C C C IF(I.LE.IWAKE.AND.J.LE.JRP) THEN
C C C IF(ITURB.EQ.1) THEN
C C C IF(TKEO(I,J).LE.0.DO) TKEO(I,J)=1.D-8
C C C RK=RE*DSQRT(TKEO(I,J))*(YP(I,J)-YP(I,1))*COSA
C C C ALU=CL*(1.DO-DEXP(-RK/ALU))*(YP(I,J)-YP(I,1))*COSA
C C C TNU(I,J)=CNU*DSQRT(TKEO(I,J))*ALU
C C C ELSE
C C C IF(TKEO(I,J).LE.0.DO) TKEO(I,J)=1.D-8
C C C ALEN=AK*(YP(I,J)-YP(I,1))*COSA
C C C RK=RE*DSQRT(TKEO(I,J))*(YP(I,J)-YP(I,1))*COSA
C C C TNU(I,J)=CMU*ALEN*DSQRT(TKEO(I,J))*(1.DO-DEXP(-AMU*RK))
C C C END IF
C C C ELSE
C C C TNU(I,J)=CNU*TKEO(I,J)*TKEO(I,J)/TDSO(I,J)
C C C END IF
C C C 28 CONTINUE
C C C DO 29 J=4,JPI
C C C IF(TNU(I,J-1).LE.TNU(I,J-2).AND.TNU(I,J-1).LT.
C C C 1TNU(I,J)) TNU(I,J)=TNU(I,J-1)
C C C 29 CONTINUE
C C C IF(I.EQ.2) THEN
C C C DO 30 J=1,JPI
C C C 30 TNU(1,J)=TNU(2,J)
C C C END IF
C C C IF(I.LT.IWAKE) TNU(I,1)=0.DO
C C C IF(I.GE.IWAKE) TNU(I,1)=TNU(I,2)

```

```

C IF(IT.GT.ITM) GO TO 31
C DO 32 J=1,JPI
C BCUI(I,J)=0.DO
C 35 BCU(IPI,J)=0.DO

```

```

C *****
C CALCULATE F-A COEFFICIENT FOR V-MOMENTUM EQUATION.
C *****

```

```

C DO 36 J=JAV,JMAX
C TNUP=0.5D0*(TNU(I,J+1)+TNU(I,J))
C REFF=1.DO/(REI+TNUP)
C CALL SRCV

```

```

C UU=0.25D0*(UO(I,J)+UO(I,J+1)+UO(I-1,J)+UO(I-1,J+1))-TNUX
C VV=VO(I,J)-2.DO*THUY
C FX=FVX(I,J)
C FY=FYI(I,J)
C ER=DSORT(A11V(I,J))
C FR=DSORT(A22V(I,J))
C AR=0.5D0*(REFF*AJIV(I,J)*(B11V(I,J)*UU+B21V(I,J)*VV)-FX)
C IF(AR.LE.0.DO) AR=0.5D0*FX
C BR=0.5D0*(REFF*AJIV(I,J)*(B12V(I,J)*UU+B22V(I,J)*VV)-FY)
C DR=2.DO*AR
C IF(IPA.EQ.2) THEN
C CALL COEFF2
C ELSE
C CALL COEFF3

```

```

C END IF
C VMB(I,J)=CF(1,1)
C VMC(I,J)=CF(1,2)
C VMT(I,J)=CF(1,3)
C VPB(I,J)=CF(2,1)
C VPC(I,J)=CF(2,2)
C VPT(I,J)=CF(2,3)
C VEB(I,J)=CF(3,1)
C VEC(I,J)=CF(3,2)
C VET(I,J)=CF(3,3)
C IF(IPA.EQ.2) THEN
C D2(I,J)=VVC(I,J)
C ELSE
C D2(I,J)=0.DO
C END IF

```

```

C E2(I,J)=REFF*TAUI
C D=REFF*VPC(I,J)/(1.DO+D2(I,J)+E2(I,J)*VPC(I,J))
C CV(I,J)=D/(VP(I,J+1)-VP(I,J))
C BCV(I,J)=B22C(I,J)+CV(I,J)
C 36 CONTINUE

```

```

C BCV(I,JPI)=0.DO
C BCV(I,JAVH1)=0.DO

```

```

C IF(I.LT.IWAKE) THEN
C U(I,1)=0.DO
C US(I,1)=0.DO
C UH(I,1)=0.DO
C TKE(I,1)=0.DO
C END IF

```

```

C US(I,JPI)=1.DO

```

```

C *****
C CALCULATE THE FIRST FICTICIOUS V-VELOCITY BY INTERPOLATION.
C *****

```

```

C V(I,1)=V(I,2)+VVI(I)/YV2(I)
C VO(I,1)=V(I,1)

```

```

C *****
C CALCULATES F-A COEFFICIENTS FOR THE U-MOMENTUM EQUATION.
C *****

```

```

C DO 34 J=JAU,JMAX
C TNUP=0.5D0*(TNU(I+1,J)+TNU(I,J))
C REFF=1.DO/(REI+TNUP)
C CALL SRCU

```

```

C UU=UO(I,J)-2.DO*TMUX
C VV=0.25D0*(VO(I,J-1)+VO(I+1,J-1)-TMUY
C 1 +VO(I,J)+VO(I+1,J))-TMUY
C FX=FUX(I,J)
C FY=FYU(I,J)
C ER=DSORT(A11U(I,J))
C FR=DSORT(A22U(I,J))
C AR=0.5D0*(REFF*AJIU(I,J)*(B11U(I,J)*UU+B21U(I,J)*VV)-FX)
C IF(AR.LE.0.DO) AR=0.5D0*FX
C BR=0.5D0*(REFF*AJIU(I,J)*(B12U(I,J)*UU+B22U(I,J)*VV)-FY)
C DR=2.DO*AR
C IF(IPA.EQ.2) THEN
C CALL COEFF2
C ELSE
C CALL COEFF3

```

```

C END IF
C UMB(I,J)=CF(1,1)
C UMC(I,J)=CF(1,2)
C UMT(I,J)=CF(1,3)
C UPB(I,J)=CF(2,1)
C UPC(I,J)=CF(2,2)
C UPT(I,J)=CF(2,3)
C UEB(I,J)=CF(3,1)
C UEC(I,J)=CF(3,2)
C UET(I,J)=CF(3,3)
C IF(IPA.EQ.2) THEN
C D1(I,J)=UMC(I,J)
C ELSE
C D1(I,J)=0.DO
C END IF

```

```

C E1(I,J)=REFF*TAUI
C D=REFF*UPC(I,J)/(1.DO+D1(I,J)+E1(I,J)*UPC(I,J))
C CU(I,J)=D/(XP(I+1,J)-XP(I,J))
C CUY(I,J)=D*(VP(I,J)-VP(I+1,J))/(XP(I+1,J)-XP(I,J))
C / (0.25D0*(VP(I,J+1)+VP(I+1,J+1)-YP(I+1,J-1)-YP(I,J-1)))
C BCU(I,J)=B11C(I,J)+CU(I,J)
C 34 CONTINUE

```

```

50 CONTINUE
C
C *****
C CALCULATE MASS SOURCE FOR PRESSURE CORRECTION EQUATION.
C *****
C
C DO 44 J=JAP,JMAX
C
C VE=0.25D0*(V(I,J)+V(I,J-1)+V(I+1,J)+V(I+1,J-1))
C VM=0.25D0*(V(I-1,J)+V(I-1,J-1)+V(I,J)+V(I,J-1))
C UN=0.25D0*(U(I,J)+U(I-1,J)+U(I,J+1)+U(I-1,J+1))
C UB=0.25D0*(U(I,J)+U(I-1,J)+U(I,J-1)+U(I-1,J-1))
C
C DF(I,J)=B21C(I,J)*VE-B21C(I-1,J)*VM
C 1 +B12C(I,J)*UN-B12C(I,J-1)*UB
C
C DS(I,J)=B11C(I,J)*US(I,J)-B11C(I-1,J)*US(I-1,J)
C 1 +B22C(I,J)*VS(I,J)-B22C(I,J-1)*VS(I,J-1)
C 2 +DF(I,J)
C
C RESMM=RESMM+DABS(DS(I,J))
C
C 44 CONTINUE
C *****
C CALCULATE THE PRESSURE CORRECTION FIELD.
C *****
C
C DO 521 J=1,JPI
C PP(I-1,J)=0.D0
C PP(I+1,J)=0.D0
C
C 521 CONTINUE
C
C ITPP=5
C
C DO 531 ITP=1,ITPP
C DO 53 J=JAP,JMAX
C AA(J)=-BCV(I,J-1)
C BB(J)=-BCU(I,J)+BCU(I-1,J)+BCV(I,J)+BCV(I,J-1)
C CC(J)=-BCV(I,J)
C DD(J)=-BCU(I,J)+PP(I+1,J)+BCU(I-1,J)+PP(I-1,J)
C 1 -DS(I,J)
C CALL TRIDAG(JAP,JMAX,AA,BB,CC,DD,T)
C DO 54 J=JAP,JMAX
C 54 PP(I,J)=T(J)
C 531 CONTINUE
C *****
C CALCULATE CORRECTED VELOCITIES.
C *****
C
C DO 55 J=JAU,JMAX
C 55 U(I,J)=US(I,J)-CU(I,J)*(PP(I+1,J)-PP(I,J))
C
C DO 57 J=JAV,JMAX
C 57 V(I,J)=VS(I,J)-CV(I,J)*(PP(I,J+1)-PP(I,J))
C V(I,JPI)=V(I,JMAX)*B22C(I,JMAX)/B22C(I,JPI)
C

```

```

U(I,JPI)=1.D0
UH(I,JPI)=1.D0
C
C *****
C START INNER ITERATION.
C *****
C
C DO 6000 ITA=1,ITERA
C
C *****
C CALCULATE THE U-STAR VELOCITY.
C *****
C
C ITUV=3
C
C DO 45 IUV=1,ITUV
C DO 46 J=JAU,JMAX
C AA(J)=-UPB(I,J)
C BB(J)=-1.D0+DI(I,J)+EI(I,J)+UFC(I,J)
C CC(J)=-UPT(I,J)
C DD(J)=-UEC(I,J)+US(I+1,J)+UBR(I,J)+US(I+1,J-1)
C 1 +UBT(I,J)+US(I+1,J+1)+UMC(I,J)+US(I-1,J)
C 2 +UMB(I,J)+US(I-1,J-1)+UMT(I,J)+US(I-1,J)
C 3 +UPC(I,J)+EI(I,J)+UO(I,J)+SU(I,J)
C 4 -BB(J)+CU(I,J)+PR(I+1,J)-PR(I,J)
C 5 -BB(J)+CU(I,J)+0.25D0*(PR(I+1,J+1)+PR(I,J+1)
C 6 -PR(I,J-1)-PR(I+1,J-1))
C DD(JAU)=DD(JAU)-AA(JAU)*US(I,JAU1)
C DD(JMAX)=DD(JMAX)-CC(JMAX)*US(I,JPI)
C CALL TRIDAG(JAU,JMAX,AA,BB,CC,DD,T)
C DO 47 J=JAU,JMAX
C 47 U(I,J)=T(J)
C IF(I.LT.IWAKE) GO TO 45
C US(I,1)=(YU3SQ(I)*US(I,2)-YU2SQ(I)*US(I,3))
C 1 / (YU3SQ(I)-YU2SQ(I))
C
C 45 CONTINUE
C *****
C CALCULATE THE V-STAR VELOCITY.
C *****
C
C ITUV=3
C
C DO 50 IUV=1,ITUV
C DO 51 J=JAV,JMAX
C AA(J)=-VPB(I,J)
C BB(J)=-1.D0+D2(I,J)+E2(I,J)+VPC(I,J)
C CC(J)=-VPT(I,J)
C DD(J)=-VEC(I,J)+VS(I+1,J)+VEB(I,J)+VS(I+1,J-1)
C 1 +VET(I,J)+VS(I+1,J+1)+VMC(I,J)+VS(I-1,J)
C 2 +VMB(I,J)+VS(I-1,J-1)+VMT(I,J)+VS(I-1,J)
C 3 +VPC(I,J)+E2(I,J)+V0(I,J)+SV(I,J)
C 4 -BB(J)+CV(I,J)+PR(I,J+1)-PR(I,J)
C DD(JAV)=DD(JAV)-AA(JAV)*VS(I,JAV1)
C DD(JMAX)=DD(JMAX)-CC(JMAX)*VS(I,JPI)
C CALL TRIDAG(JAV,JMAX,AA,BB,CC,DD,T)
C DO 52 J=JAV,JMAX
C 52 VS(I,J)=T(J)
C VS(I,1)=V(I,1)
C VS(I,JPI)=VS(I,JMAX)*B22C(I,JMAX)/B22C(I,JPI)
C

```

A  
U  
C  
C



```

IF(I.LT.IWAKE) GO TO 6000
U(I,1)=(YU2SQ(I)*U(I,2)-YU2SQ(I)*U(I,3))
1 / (YU3SQ(I)-YU2SQ(I))
6000 CONTINUE
C
C *****
C CALCULATE THE FINITE ANALYTIC COEFFICIENT FOR TURBULENT
C KINETIC ENERGY EQUATION.
C *****
DO 60 J=JA,JMAX
TNUP=TNUI(I,J)
REFF=1.00/(REI+TNUP)
CALL SRCX
UU=0.500*(UU(I,J)+UU(I-1,J))*CEPK - TNUX
VV=0.500*(VV(I,J-1)+VV(I,J))*CEPK - TNUY
FX=FPX(I,J)
FY=FPY(I,J)
ER=DSQRT(A11P(I,J))
FR=DSQRT(A22P(I,J))
AR=0.500*(REFF*AJIP(I,J)*(B11P(I,J)*UU+B21P(I,J)*VV)-FX)
IF(AR.LE.0.00) AR=0.500*FX
BR=0.500*(REFF*AJIP(I,J)*(B12P(I,J)*UU+B22P(I,J)*VV)-FY)
DR=2.00*AR
IF(IFA.EQ.2) THEN
CALL COEFF2
ELSE
CALL COEFF3
END IF
E4(I,J)=CEPS*REFF*TAUI
61 CONTINUE
C
C *****
C CALCULATE THE TURBULENT KINETIC ENERGY FIELD.
C *****
ITKD=3
DO 62 ITK=1,ITKD
DO 63 J=JA,JMAX
AA(J)=TKPB(I,J)
BB(J)=1.00+D3(I,J)+(E3(I,J)-SKP(I,J))*TKPC(I,J)
CC(J)=TKPT(I,J)
63 DD(J)=TREC(I,J)*TKE(I+1,J)+TKEB(I,J)*TKE(I+1,J-1)
1 +TKET(I,J)*TKE(I+1,J+1)+TKWC(I,J)*TKE(I-1,J)
2 +TKWB(I,J)*TKE(I-1,J-1)+TKWT(I,J)*TKE(I-1,J+1)
3 +TKPC(I,J)*(E3(I,J)*TKE(I,J)+SK(I,J))
DD(JA)=DD(JA)-AA(JA)*TKE(I,JAH1)
DD(JMAX)=DD(JMAX)-CC(JMAX)*TNE(I,JPI)
CALL TRIDAG(JA,JMAX,AA,BB,CC,DD,T)
DO 64 J=JA,JMAX
TKE(I,J)=T(J)
TKE(I,JPI)=TKE(I,JMAX)
IF(I.LT.IWAKE) GO TO 62
TKE(I,1)=(YP3SQ(I)*TKE(I,2)-YP2SQ(I)*TKE(I,3))
1 / (YP3SQ(I)-YP2SQ(I))
64 CONTINUE
C
C *****
C CALCULATE THE TURBULENT DISSIPATION RATE.
C *****
IF(I.LE.IWAKE) THEN

```

```

FR=DSQRT(A22P(I,J))
AR=0.500*(REFF*AJIP(I,J)*(B11P(I,J)*UU+B21P(I,J)*VV)-FX)
IF(AR.LE.0.00) AR=0.500*FX
BR=0.500*(REFF*AJIP(I,J)*(B12P(I,J)*UU+B22P(I,J)*VV)-FY)
DR=2.00*AR
IF(IFA.EQ.2) THEN
CALL COEFF2
ELSE
CALL COEFF3
END IF
E4(I,J)=CEPS*REFF*TAUI
61 CONTINUE
C
C *****
C CALCULATE THE TURBULENT KINETIC ENERGY FIELD.
C *****
ITKD=3
DO 62 ITK=1,ITKD
DO 63 J=JA,JMAX
AA(J)=TKPB(I,J)
BB(J)=1.00+D3(I,J)+(E3(I,J)-SKP(I,J))*TKPC(I,J)
CC(J)=TKPT(I,J)
63 DD(J)=TREC(I,J)*TKE(I+1,J)+TKEB(I,J)*TKE(I+1,J-1)
1 +TKET(I,J)*TKE(I+1,J+1)+TKWC(I,J)*TKE(I-1,J)
2 +TKWB(I,J)*TKE(I-1,J-1)+TKWT(I,J)*TKE(I-1,J+1)
3 +TKPC(I,J)*(E3(I,J)*TKE(I,J)+SK(I,J))
DD(JA)=DD(JA)-AA(JA)*TKE(I,JAH1)
DD(JMAX)=DD(JMAX)-CC(JMAX)*TNE(I,JPI)
CALL TRIDAG(JA,JMAX,AA,BB,CC,DD,T)
DO 64 J=JA,JMAX
TKE(I,J)=T(J)
TKE(I,JPI)=TKE(I,JMAX)
IF(I.LT.IWAKE) GO TO 62
TKE(I,1)=(YP3SQ(I)*TKE(I,2)-YP2SQ(I)*TKE(I,3))
1 / (YP3SQ(I)-YP2SQ(I))
64 CONTINUE
C
C *****
C CALCULATE THE TURBULENT DISSIPATION RATE.
C *****
IF(I.LE.IWAKE) THEN

```

```

JE-JRP+1
DO 717 J=2,JRP
IF(ITURB.EQ.1) THEN
  IF(TKE(I,J).LE.0.DO) TKE(I,J)=1.D-8
  RK-RE*DSQRT(TKE(I,J))*(YP(I,J)-YP(I,1))*COSA
  ALE=CL*(1.DO-DEXP(-RK/AE))*(YP(I,J)-YP(I,1))*COSA
  TDS(I,J)=TKE(I,J)*.500/ALE
ELSE
  IF(TKE(I,J).LE.0.DO) TKE(I,J)=1.D-8
  ALEN=AK*(YP(I,J)-YP(I,1))*COSA
  RK-RE*DSQRT(TKE(I,J))*(YP(I,J)-YP(I,1))*COSA
  TDS(I,J)=CD1*TKE(I,J)*.500*(1.DO-DEXP(-AKU*RK))/ALEN
  +CD2*TKE(I,J)/ALEN/ALEN/RE
1 END IF
717 CONTINUE
C
YL=(YP(I,2)-YP(I,1))*COSA
TDS(I,1)=4.DO*TKE(1,2)/YL/YL/RE-TDS(I,2)
ELSE
  JE=2
END IF
C
ITKD=3
DO 65 ITK=1,ITKD
DO 66 J=JE,JMAX
AA(J)=TDPB(I,J)
BB(J)=1.DO+D4(I,J)*(E4(I,J)-SDP(I,J))*TDPC(I,J)
CC(J)=TDPT(I,J)
66 DD(J)=TDEC(I,J)*TDS(I+1,J)+TDEB(I,J)*TDS(I+1,J-1)
+TDEB(I,J)*TDS(I+1,J+1)+TDMC(I,J)*TDS(I-1,J)
+TDMC(I,J)*TDS(I-1,J-1)+TDMT(I,J)*TDS(I-1,J+1)
+TDMT(I,J)*TDS(I-1,J-1)+TDSO(I,J)*SD(I,J)
DD(JE)=DD(JE)-AA(JE)*TDS(I,JE-1)
DD(JMAX)=DD(JMAX)-CC(JMAX)*TDS(I,JPI)
CALL TRIDAG(JE,JMAX,AA,BB,CC,DD,T)
DO 67 J=JE,JMAX
TDS(I,J)=T(J)
TDS(I,JPI)=TDS(I,JMAX)
IF(I.LT.IWAKE) GO TO 65
TDS(I,1)=(VP3SQ(I)*TDS(I,2)-YP2SQ(I))*TDS(I,3)
1 / (VP3SQ(I)-YP2SQ(I))
65 CONTINUE
C
*****
CALCULATE THE PSEUDO VELOCITIES.
*****
C
DO 40 J=JAP,JMAX
40 UH(I,J)=(UMC(I,J)+U(I-1,J)+UEC(I,J)+U(I+1,J)
+UPT(I,J)+U(I,J+1)+UPB(I,J)+U(I,J-1)
+UET(I,J)+U(I+1,J+1)+UEB(I,J)+U(I+1,J-1)
+UMT(I,J)+U(I-1,J-1)+UWB(I,J)+U(I-1,J-1)
+UPC(I,J)+U(I,J)+E1(I,J)+SU(I,J))
/(1.DO+D1(I,J)+UPC(I,J)+E1(I,J))
-CUY(I,J)+.25DU*(PR(I+1,J+1)+PR(I,J+1)
-PR(I,J-1)-PR(I+1,J-1))
C
*****
CALCULATE THE MASS SOURCE FOR PRESSURE EQUATION.
*****
C
DO 43 J=JAP,JMAX
VE=0.25DU*(V(I,J)+V(I,J-1)+V(I+1,J)+V(I+1,J-1))
VM=0.25DU*(V(I-1,J)+V(I-1,J-1)+V(I,J)+V(I,J-1))
UN=0.25DU*(U(I,J)+U(I-1,J)+U(I,J+1)+U(I-1,J+1))
UB=0.25DU*(U(I,J)+U(I-1,J)+U(I,J-1)+U(I-1,J-1))
DF(I,J)=B21C(I,J)*VE-B21C(I-1,J)*VW
+ B21C(I,J)*UN-B12C(I,J-1)*UB
43 DH(I,J)=B11C(I,J)*UH(I,J)-B11C(I-1,J)*UH(I-1,J)
+B22C(I,J)*VH(I,J)-B22C(I,J-1)*VH(I,J-1)
+DF(I,J)
*****
UPDATE THE TRANSPORT QUANTITIES AT UPSTREAM
STATION AND PREVIOUS TIME STEP.
*****
DO 679 J=1,JPI
UO(I,J)=U(I,J)
VO(I,J)=V(I,J)
US(I,J)=U(I,J)
VS(I,J)=V(I,J)
TKEO(I,J)=TKE(I,J)
TDSO(I,J)=TDS(I,J)
679 IF(I.EQ.IMAX) THEN
DO 680 J=1,JPI
U(IPI,J)=U(IMAX,J)
V(IPI,J)=V(IMAX,J)
VS(IPI,J)=VS(IMAX,J)
VO(IPI,J)=VO(IMAX,J)
TKE(IPI,J)=TKE(IMAX,J)
TDS(IPI,J)=TDS(IMAX,J)
TREO(IPI,J)=TRE(IMAX,J)
TDSO(IPI,J)=TDSO(IMAX,J)
TNU(IPI,J)=TNU(IMAX,J)
680 CONTINUE
END IF
*****
CALCULATE FRICTION COEFFICIENT.
*****

```

```

DO 42 J=JAV,JMAX
42 VH(I,J)=(VMC(I,J)+V(I-1,J)+VEC(I,J)+V(I+1,J)
+VPT(I,J)+V(I,J+1)+VPB(I,J)+V(I,J-1)
+VET(I,J)+V(I+1,J+1)+VEB(I,J)+V(I+1,J-1)
+VMT(I,J)+V(I-1,J-1)+VMB(I,J)+V(I-1,J-1)
+VPC(I,J)+V(I,J)+E2(I,J)+SV(I,J))
/(1.DO+D2(I,J)+VPC(I,J)+E2(I,J))
VH(I,JAV-1)=V(I,JAV-1)
VH(I,JPI)=V(I,JPI)
C
*****
CALCULATE THE MASS SOURCE FOR PRESSURE EQUATION.
*****
C
DO 43 J=JAP,JMAX
VE=0.25DU*(V(I,J)+V(I,J-1)+V(I+1,J)+V(I+1,J-1))
VM=0.25DU*(V(I-1,J)+V(I-1,J-1)+V(I,J)+V(I,J-1))
UN=0.25DU*(U(I,J)+U(I-1,J)+U(I,J+1)+U(I-1,J+1))
UB=0.25DU*(U(I,J)+U(I-1,J)+U(I,J-1)+U(I-1,J-1))
DF(I,J)=B21C(I,J)*VE-B21C(I-1,J)*VW
+ B21C(I,J)*UN-B12C(I,J-1)*UB
43 DH(I,J)=B11C(I,J)*UH(I,J)-B11C(I-1,J)*UH(I-1,J)
+B22C(I,J)*VH(I,J)-B22C(I,J-1)*VH(I,J-1)
+DF(I,J)
*****
UPDATE THE TRANSPORT QUANTITIES AT UPSTREAM
STATION AND PREVIOUS TIME STEP.
*****
DO 679 J=1,JPI
UO(I,J)=U(I,J)
VO(I,J)=V(I,J)
US(I,J)=U(I,J)
VS(I,J)=V(I,J)
TKEO(I,J)=TKE(I,J)
TDSO(I,J)=TDS(I,J)
679 IF(I.EQ.IMAX) THEN
DO 680 J=1,JPI
U(IPI,J)=U(IMAX,J)
V(IPI,J)=V(IMAX,J)
VS(IPI,J)=VS(IMAX,J)
VO(IPI,J)=VO(IMAX,J)
TKE(IPI,J)=TKE(IMAX,J)
TDS(IPI,J)=TDS(IMAX,J)
TREO(IPI,J)=TRE(IMAX,J)
TDSO(IPI,J)=TDSO(IMAX,J)
TNU(IPI,J)=TNU(IMAX,J)
680 CONTINUE
END IF
*****
CALCULATE FRICTION COEFFICIENT.
*****

```



```

C 207 FORMAT(7E12.4)
C
C 4001 CONTINUE
C
C *****
C DATA FOR PLOTTING.
C *****
C
C IAWAKE=IAWAKE-1
C WRITE(8,2009) (UTAU(A,I), I=2, IMAWAKE)
C WRITE(8,2009) (PR(I,1), I=1, IPI)
C WRITE(8,2009) (XP(I,1), I=1, IPI)
C WRITE(8,2009) (XU(I,1), I=1, IPI)
C WRITE(8,2009) (VP(I,1), I=1, IPI)
C WRITE(8,2009) (YU(I,1), I=1, IPI)
C
C DO 5001 I=1, IPI
C   VO(I,1)=0.D0
C   DO 5001 J=1, JPI
C     5001 WRITE(8,2005) XU(I,J), YU(I,J), UO(I,J), VO(I,J), PR(I,J)
C     C , TKEO(I,J), TDSO(I,J)
C
C 205 FORMAT(7E12.4)
C
C 2009 FORMAT(6E12.4)
C
C *****
C SAVE FOR THE FULLY ELLIPTIC CALCULATION.
C ( IF NECESSARY )
C *****
C OPEN(UNIT=10, FILE='A'P'OUT')
C WRITE(10,2009) (U(I,J), I=1, IPI), J=1, JPI)
C WRITE(10,2009) (V(I,J), I=1, IPI), J=1, JPI)
C WRITE(10,2009) (TKE(I,J), I=1, IPI), J=1, JPI)
C WRITE(10,2009) (TDS(I,J), I=1, IPI), J=1, JPI)
C WRITE(10,2009) (TNU(I,J), I=1, IPI), J=1, JPI)
C CLOSE(10)
C
C 5555 CONTINUE
C
C CLOSE(8)
C CLOSE(9)
C CLOSE(7)
C
C STOP
C END
C *****
C CALCULATE SOURCE FUNCTION FOR U-MOMENTUM EQUATION.
C *****
C
C SUBROUTINE SRCU

```

```

C
C *****
C IMPLICIT REAL*8(A-H,O-Z)
C COMMON/VEL1/UO(90,53)
C COMMON/VEL4/U(90,53)
C COMMON/VEL9/V(90,53)
C COMMON/TURB1/TKE(90,53)
C COMMON/TURB2/TKEO(90,53)
C COMMON/TURB5/TNU(90,53)
C COMMON/ST/SU(90,53), SV(90,53), SK(90,53), SD(90,53)
C COMMON/CONS1/I,J,IMAX,JMAX,IPI,JPI,JAU,JAV,JAP,JA,IFA,IWAKE
C COMMON/CONS5/RE,REI,TAU,TAU2
C COMMON/CONS6/REFF
C COMMON/GEOL/XP(90,53), YP(90,53)
C COMMON/GEOS/XU(90,53), YU(90,53)
C COMMON/GEOA/XV(90,53), YV(90,53)
C COMMON/BODY1/B11,B12,B21,B22
C COMMON/BODY2/G11,G12,G22,A11,A12,A22,G,AJI
C COMMON/BODY3/XXI,XXI,XXI,XET,YET,R
C COMMON/ZUT/TNUX,TNUY
C
C R=YU(I,J)
C XET=0.5D0*(XU(I,J+1)-XU(I,J-1))
C YET=0.5D0*(YU(I,J+1)-YU(I,J-1))
C XXI=0.5D0*(XU(I+1,J)-XU(I-1,J))
C YXI=0.5D0*(YU(I+1,J)-YU(I-1,J))
C CALL GEOCOE
C DUXI=0.5D0*(U(I+1,J)-U(I-1,J))
C DUYI=0.5D0*(V(I+1,J)-V(I-1,J))
C DUV=AJI*(B21*DUXI+B22*DUYI)
C DUX=AJI*(B11*DUXI+B12*DUYI)
C SS=0.5D0*A12*(UO(I+1,J+1)+UO(I-1,J-1)-UO(I+1,J-1)
C   -UO(I-1,J+1))
C
C XET=0.25D0*(XP(I,J+1)+XP(I+1,J+1)-XP(I,J-1)-XP(I+1,J-1))
C YET=0.25D0*(YP(I,J+1)+YP(I+1,J+1)-YP(I,J-1)-YP(I+1,J-1))
C XXI=XP(I+1,J)-XP(I,J)
C YXI=YP(I+1,J)-YP(I,J)
C CALL GEOCOE
C TNXI=TNU(I+1,J)-TNU(I,J)
C TNXT=0.25D0*(TNU(I,J+1)+TNU(I+1,J+1)-TNU(I,J-1)
C   -TNU(I+1,J-1))
C DKXI=TKE(I+1,J)-TKE(I,J)
C DKET=0.25D0*(TKE(I,J+1)+TKE(I+1,J+1)-TKE(I,J-1)
C   -TKE(I+1,J-1))
C DKX=AJI*(B11*DKXI+B12*DKET)
C TNUI=AJI*(B21*TNXI+B22*TNXT)
C TNUX=AJI*(B11*TNXI+B12*TNXT)
C
C XET=0.5D0*(XV(I+1,J)+XV(I,J)-XV(I+1,J-1)-XV(I,J-1))
C YET=0.5D0*(YV(I+1,J)+YV(I,J)-YV(I+1,J-1)-YV(I,J-1))
C XXI=0.5D0*(XV(I+1,J)+XV(I+1,J-1)-XV(I,J)-XV(I,J-1))
C YXI=0.5D0*(YV(I+1,J)+YV(I+1,J-1)-YV(I,J)-YV(I,J-1))
C CALL GEOCOE
C DVXI=0.5D0*(V(I+1,J)+V(I+1,J-1)-V(I,J)-V(I,J-1))
C DVY=0.5D0*(V(I+1,J)+V(I,J)-V(I+1,J-1)-V(I,J-1))
C DVX=AJI*(B11*DVXI+B12*DVY)
C
C SU(I,J)=SS+REFF*(TNUI*DVX
C   -2.D0/3.D0*DKX)

```

```

1 SS=0.5D0*A12*(V0(I+1,J+1)+V0(I-1,J-1))-V0(I+1,J-1)
1 -V0(I-1,J+1)
C R=VY(I,J)
C V0=V(I,J)
C SV(I,J)=SS+REFP*(TNUX*DUY
1 -2.D0/3.D0*DKY)-V0/R/R
C RETURN
C END
C *****
C CALCULATE THE SOURCE FUNCTION FOR TURBULENT KINETIC
C ENERGY EQUATION.
C *****
C SUBROUTINE SRCK
C IMPLICIT REAL*8(A-H,O-Z)
C COMMON/VEL4/V(90,53)
C COMMON/VEL9/V(90,53)
C COMMON/TURB1/TKE(90,53)
C COMMON/TURB2/TKE0(90,53)
C COMMON/TURB3/TDS(90,53)
C COMMON/TURB5/TNU(90,53)
C COMMON/TURB6/TGE(90,53)
C COMMON/ST/SU(90,53),SV(90,53),SK(90,53),SD(90,53)
C COMMON/STLN/SKP(90,53),SDP(90,53)
C COMMON/CONS1/I,J,IMAX,JMAX,IP1,JPI,JAU,JAV,JAP,JA,IFA,IWAKE
C COMMON/CONS4/CNU,CNU3,CNU21,CEPS1,CEPS2,CEPK,CEPS,CKAR,E
C COMMON/CONS6/REFP
C COMMON/GEOL/XP(90,53),YP(90,53)
C COMMON/GE03/XU(90,53),YU(90,53)
C COMMON/GE04/XV(90,53),YV(90,53)
C COMMON/BODY1/BI1,BI2,B21,B22
C COMMON/BODY2/G11,G12,G22,A11,A12,A22,G,AJI
C COMMON/BODY3/XXI,XXI,XXI,XET,YET,R
C COMMON/ZUT/TNUX,TNUY
C *****
C R=YV(I,J)
C XXI=XU(I,J)-XU(I-1,J)
C YXI=YU(I,J)-YU(I-1,J)
C XET=0.25D0*(XU(I,J+1)+XU(I-1,J+1)+XU(I,J-1)-XU(I-1,J-1))
C YET=0.25D0*(YU(I,J+1)+YU(I-1,J+1)+YU(I,J-1)-YU(I-1,J-1))
C CALL GEOCOE
C DUET=0.25D0*(U(I,J+1)+U(I-1,J+1)-U(I,J-1)-U(I-1,J-1))
C DUXI=U(I,J)-U(I-1,J)
C DUY=AJI*(B21*DUXI+B12*DUET)
C *****
C XET=XV(I,J)-XV(I-1,J)
C YET=YV(I,J)-YV(I-1,J)
C XXI=0.25D0*(XV(I+1,J)+XV(I-1,J-1)-XV(I-1,J)-XV(I-1,J-1))
C YXI=0.25D0*(YV(I+1,J)+YV(I-1,J-1)-YV(I-1,J)-YV(I-1,J-1))
C CALL GEOCOE
C DVET=V(I,J)-V(I-1,J)
C DVXI=0.25D0*(V(I+1,J)+V(I-1,J)-V(I-1,J)-V(I-1,J-1))
C DVY=AJI*(B11*DVXI+B12*DVET)
C *****
C DVY=AJI*(B21*DVXI+B22*DVET)

```

```

C *****
C CALCULATE THE SOURCE FUNCTION FOR V-MOMENTUM EQUATION.
C *****
C SUBROUTINE SRCV
C IMPLICIT REAL*8(A-H,O-Z)
C COMMON/VEL4/V(90,53)
C COMMON/VEL6/V0(90,53)
C COMMON/VEL9/V(90,53)
C COMMON/TURB1/TKE(90,53)
C COMMON/TURB5/TNU(90,53)
C COMMON/ST/SU(90,53),SV(90,53),SK(90,53),SD(90,53)
C COMMON/CONS1/I,J,IMAX,JMAX,IP1,JPI,JAU,JAV,JAP,JA,IFA,IWAKE
C COMMON/CONS6/REFP
C COMMON/GEOL/XP(90,53),YP(90,53)
C COMMON/GE03/XU(90,53),YU(90,53)
C COMMON/GE04/XV(90,53),YV(90,53)
C COMMON/BODY1/BI1,BI2,B21,B22
C COMMON/BODY2/G11,G12,G22,A11,A12,A22,G,AJI
C COMMON/BODY3/XXI,XXI,XXI,XET,YET,R
C COMMON/ZUT/TNUX,TNUY
C *****
C R=VY(I,J)
C XXI=0.25D0*(XP(I+1,J)+XP(I-1,J+1)-XP(I-1,J)-XP(I-1,J+1))
C YXI=0.25D0*(YP(I+1,J)+YP(I-1,J+1)-YP(I-1,J)-YP(I-1,J+1))
C XET=XP(I,J+1)-XP(I,J)
C YET=YP(I,J+1)-YP(I,J)
C CALL GEOCOE
C DKET=TKE(I,J+1)-TKE(I,J)
C DKXI=0.25D0*(TKE(I+1,J)+TKE(I-1,J+1)-TKE(I-1,J)-TKE(I-1,J+1))
C TNET=TNU(I,J+1)-TNU(I,J)
C TNXI=0.25D0*(TNU(I+1,J)+TNU(I-1,J+1)-TNU(I-1,J)-TNU(I-1,J+1))
C DKY=AJI*(B21*DKXI+B22*DKET)
C TNUY=AJI*(B21*TNXI+B22*TNET)
C *****
C XET=0.5D0*(XU(I,J+1)+XU(I-1,J+1)-XU(I,J)-XU(I-1,J))
C YET=0.5D0*(YU(I,J+1)+YU(I-1,J+1)-YU(I,J)-YU(I-1,J))
C XXI=0.5D0*(XU(I,J)+XU(I-1,J)-XU(I-1,J)-XU(I-1,J+1))
C YXI=0.5D0*(YU(I,J)+YU(I-1,J)-YU(I-1,J)-YU(I-1,J+1))
C CALL GEOCOE
C DUET=0.5D0*(U(I,J+1)+U(I-1,J+1)-U(I,J)-U(I-1,J))
C DUXI=0.5D0*(U(I,J)+U(I-1,J)-U(I-1,J)-U(I-1,J+1))
C DUY=AJI*(B21*DUXI+B22*DUET)
C *****
C XXI=0.5D0*(XV(I+1,J)+XV(I-1,J))
C YXI=0.5D0*(YV(I+1,J)+YV(I-1,J))
C XET=0.5D0*(XV(I,J+1)+XV(I-1,J)-XV(I-1,J)-XV(I-1,J+1))
C YET=0.5D0*(YV(I,J+1)+YV(I-1,J)-YV(I-1,J)-YV(I-1,J+1))
C CALL GEOCOE
C DVET=0.5D0*(V(I,J+1)+V(I-1,J)-V(I-1,J)-V(I-1,J+1))
C DVXI=0.5D0*(V(I+1,J)+V(I-1,J)-V(I-1,J)-V(I-1,J+1))
C DVY=AJI*(B11*DVXI+B12*DVET)
C *****
C DVY=AJI*(B21*DVXI+B22*DVET)

```

```

C *****
C CALCULATE THE SOURCE FUNCTION FOR V-MOMENTUM EQUATION.
C *****
C SUBROUTINE SRCV
C IMPLICIT REAL*8(A-H,O-Z)
C COMMON/VEL4/V(90,53)
C COMMON/VEL6/V0(90,53)
C COMMON/VEL9/V(90,53)
C COMMON/TURB1/TKE(90,53)
C COMMON/TURB5/TNU(90,53)
C COMMON/ST/SU(90,53),SV(90,53),SK(90,53),SD(90,53)
C COMMON/CONS1/I,J,IMAX,JMAX,IP1,JPI,JAU,JAV,JAP,JA,IFA,IWAKE
C COMMON/CONS6/REFP
C COMMON/GEOL/XP(90,53),YP(90,53)
C COMMON/GE03/XU(90,53),YU(90,53)
C COMMON/GE04/XV(90,53),YV(90,53)
C COMMON/BODY1/BI1,BI2,B21,B22
C COMMON/BODY2/G11,G12,G22,A11,A12,A22,G,AJI
C COMMON/BODY3/XXI,XXI,XXI,XET,YET,R
C COMMON/ZUT/TNUX,TNUY
C *****
C R=VY(I,J)
C XXI=0.25D0*(XP(I+1,J)+XP(I-1,J+1)-XP(I-1,J)-XP(I-1,J+1))
C YXI=0.25D0*(YP(I+1,J)+YP(I-1,J+1)-YP(I-1,J)-YP(I-1,J+1))
C XET=XP(I,J+1)-XP(I,J)
C YET=YP(I,J+1)-YP(I,J)
C CALL GEOCOE
C DKET=TKE(I,J+1)-TKE(I,J)
C DKXI=0.25D0*(TKE(I+1,J)+TKE(I-1,J+1)-TKE(I-1,J)-TKE(I-1,J+1))
C TNET=TNU(I,J+1)-TNU(I,J)
C TNXI=0.25D0*(TNU(I+1,J)+TNU(I-1,J+1)-TNU(I-1,J)-TNU(I-1,J+1))
C DKY=AJI*(B21*DKXI+B22*DKET)
C TNUY=AJI*(B21*TNXI+B22*TNET)
C *****
C XET=0.5D0*(XU(I,J+1)+XU(I-1,J+1)-XU(I,J)-XU(I-1,J))
C YET=0.5D0*(YU(I,J+1)+YU(I-1,J+1)-YU(I,J)-YU(I-1,J))
C XXI=0.5D0*(XU(I,J)+XU(I-1,J)-XU(I-1,J)-XU(I-1,J+1))
C YXI=0.5D0*(YU(I,J)+YU(I-1,J)-YU(I-1,J)-YU(I-1,J+1))
C CALL GEOCOE
C DUET=0.5D0*(U(I,J+1)+U(I-1,J+1)-U(I,J)-U(I-1,J))
C DUXI=0.5D0*(U(I,J)+U(I-1,J)-U(I-1,J)-U(I-1,J+1))
C DUY=AJI*(B21*DUXI+B22*DUET)
C *****
C XXI=0.5D0*(XV(I+1,J)+XV(I-1,J))
C YXI=0.5D0*(YV(I+1,J)+YV(I-1,J))
C XET=0.5D0*(XV(I,J+1)+XV(I-1,J)-XV(I-1,J)-XV(I-1,J+1))
C YET=0.5D0*(YV(I,J+1)+YV(I-1,J)-YV(I-1,J)-YV(I-1,J+1))
C CALL GEOCOE
C DVET=0.5D0*(V(I,J+1)+V(I-1,J)-V(I-1,J)-V(I-1,J+1))
C DVXI=0.5D0*(V(I+1,J)+V(I-1,J)-V(I-1,J)-V(I-1,J+1))
C DVY=AJI*(B11*DVXI+B12*DVET)
C *****
C DVY=AJI*(B21*DVXI+B22*DVET)

```



COMMON/FAI/AR, BR, DR, ER, FR

```

IF (DABS (BR) .LT. EPE) GO TO 1
BRO=BR
BR=BR/FR
HY=1. DO/FR
BK=BR*HY
DBK=DMIN1 (DABS (BK) , FMAX)
BK=DSIGN (DBK, BK)
EPBK=DEXP (BK)
EPBK1=1. DO/EPBK
TANHB=(EPBK-EPBK1)/(EPBK+EPBK1)
CF(2,3)=EPBK1/EPBK1+EPBK
CF(2,2)=TANHB/(2. DO*BRO)
CF(2,1)=EPBK/(EPBK1+EPBK)
GO TO 2
1 CF(2,1)=0.5DO
CF(2,2)=0.5DO/FR/FR
CF(2,3)=0.5DO
2 CONTINUE
CF(1,1)=0. DO
CF(1,2)=CF(2,2)*DR
CF(1,3)=0. DO
CF(3,1)=0. DO
CF(3,2)=0. DO
CF(3,3)=0. DO
RETURN
END

```

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 EPH=DEXP (AH)
EPBK=DEXP (BK)
EPAN1=1./EPAN
EPBK1=1./EPBK
COSH=0.5*(EPAN+EPAN1)
COSH=0.5*(EPBK+EPBK1)
COTHA=2.*COSH/(EPAN+EPAN1)
COTHB=2.*COSH/(EPBK+EPBK1)
AKCTHA=AK*COTHA
BHCTHB=BH*COTHB
PWR=1.
IF (HX. GT. HY) GO TO 11
EX2=0.
DO 10 M=1, MAX
ZA=(M-0.5)*PI
ZA2=ZA*ZA
PWR=-PWR
DABK=DSORT (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*COSH+COSH*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 M=1, MAX
ZA=(M-0.5)*PI

```

C

IF (DABS (BR) .LT. EPE) GO TO 1

```

BRO=BR
BR=BR/FR
HY=1. DO/FR
BK=BR*HY
DBK=DMIN1 (DABS (BK) , FMAX)
BK=DSIGN (DBK, BK)
EPBK=DEXP (BK)
EPBK1=1. DO/EPBK
TANHB=(EPBK-EPBK1)/(EPBK+EPBK1)
CF(2,3)=EPBK1/EPBK1+EPBK
CF(2,2)=TANHB/(2. DO*BRO)
CF(2,1)=EPBK/(EPBK1+EPBK)
GO TO 2
1 CF(2,1)=0.5DO
CF(2,2)=0.5DO/FR/FR
CF(2,3)=0.5DO
2 CONTINUE
CF(1,1)=0. DO
CF(1,2)=CF(2,2)*DR
CF(1,3)=0. DO
CF(3,1)=0. DO
CF(3,2)=0. DO
CF(3,3)=0. DO
RETURN
END

```

C C C C C C C C

SUBROUTINE TRIDAG TO SOLVE ALGEBRAIC EQUATIONS SIMULTANEOUSLY FOR EACH ROW OR COLUMN

```

*****
SUBROUTINE TRIDAG (IF, L, A, B, C, D, V)
IMPLICIT DOUBLE PRECISION (A-H, O-Z)
DIMENSION A(53), B(53), C(53), D(53), V(53), BETA(53), GAMMA(53)
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)
GAMMA (I)=(D (I)-A (I)*GAMMA (I-1))/BETA (I)
LAST=L-IF
DO 2 K=1, LAST
I=L-K
V (I)=GAMMA (I)-C (I)*V (I+1)/BETA (I)
RETURN
END

```

CALCULATE THE FINITE ANALYTIC COEFFICIENT FOR CONVECTIVE-DIFFUSION PROBLEM ---- ELLIPTIC CASE ( 9 POINT FORMULA.)

```

*****
SUBROUTINE COEFFJ
IMPLICIT DOUBLE PRECISION (A-H, O-Z)
COMMON/COEF7/CF(3,3)

```

C C C C C C C C

IF(AKCTHA.GT.BHCTHB) GO TO 32

EY2=0.

FY2=0.

DO 30 II=1,MAX

ZA=(II-0.5)\*PI

ZAZ=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.-COSH(A/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

ZAZ=ZA\*ZA

PWR=-PWR

DABK=DK-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(AK)

DBH=DABS(BH)

COTHA=DSIGN(CI,AR)

COTHB=DSIGN(CI,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./COTHB

BE=0.5\*(1.-TANHA)

BW=0.5\*(1.+TANHA)

BN=0.5\*(1.-TANHB)

BS=0.5\*(1.+TANHB)

CF(1,1)=BS\*BW\*Q

CF(3,1)=BS\*BE\*Q

CF(1,3)=BN\*BW\*Q

CF(3,3)=BN\*BE\*Q

CF(2,1)=BS\*PA

CF(2,3)=BN\*PA

ZAZ=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\*COSH\*EV2

PA=1.+AKCTHA/BHCTHB\*(PB-1.)

CF(2,2)=0.5\*HY/(BR\*COTHB)\*(1.-PB)

GO TO 100

2 EPBK=DEXP(BK)

EPBK1=1./EPBK

COSH=0.5\*(EPBK+EPBK1)

COTHB=2.\*COSH/(EPBK-EPBK1)

COTHA=DSIGN(CI,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

ZAZ=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.-COSH(A/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

ZAZ=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\*COSH\*EY2

PA=1.+AKCTHA/BHCTHB\*(PB-1.)

CF(2,2)=0.5\*HY/(BR\*COTHB)\*(1.-PB)

GO TO 100

3 EPAH=DEXP(AH)

EPAH1=1./EPAH

COSHA=0.5\*(EPAH+EPAH1)

COTHA=2.\*COSH/(EPAH-EPAH1)

COTHB=DSIGN(CI,BR)

AKCTHA=AK\*COTHA

BHCTHB=BH\*COTHB

PWR=1.





Computer Code for Chapter-VII

1. Full Body Calculation
  - 1-1) Computer Program for the Generation of Body Shape and X-Coordinates
  - 1-2) Geometric Data Files  
( Generated by Program 1-1 )
  - 1-3) Computer Program for the Generation of Boundary-Fitted Coordinates
  - 1-4) Computer Program for Input Data for Flow calculation
  - 1-5) Computer Program for Flow Calculation by Finite Volume Method







```

1 +0.17730*AK2**2
RX(I)=RM*DSQRT(SUM2)
ELSE
RX(I)=0.
END IF
170 CONTINUE
C
C
C *****
C GENERATE 10% THICK ARC AIRFOIL.
C *****
C ELSE IF (IBODY.EQ.8) THEN
C
C AM=0.1
C AN1=1./2./AH*(AH*AM-0.25)
C AN2=1./2./AH*(AH*AM+0.25)
C
C DO 180 I=1, IPI
C XT=X(I)
C IF(XT.LE.0.DO.OR.XT.GE.1.DO) THEN
C RX(I)=0.DO
C ELSE
C RX(I)=AN1+DSQRT(AN2*AN2-(XT-0.5)**2)
C END IF
180 CONTINUE
C
C *****
C GENERATE 5% THICK ARC AIRFOIL.
C *****
C ELSE IF (IBODY.EQ.9) THEN
C
C AM=0.05
C AN1=1./2./AH*(AH*AM-0.25)
C AN2=1./2./AH*(AH*AM+0.25)
C
C DO 190 I=1, IPI
C XT=X(I)
C IF(XT.LE.0.DO.OR.XT.GE.1.DO) THEN
C RX(I)=0.DO
C ELSE
C RX(I)=AN1+DSQRT(AN2*AN2-(XT-0.5)**2)
C END IF
190 CONTINUE
C
C END IF
C
C OPEN (UNIT=7, FILE=FNAME(IBODY))
C WRITE(7,1111)(X(I), I=1, IPI)
C WRITE(7,1111)(RX(I), I=1, IPI)
C 1111 FORMAT(6D13.6)
C CLOSE(7)
C
C 999 CONTINUE
C STOP
C END

```











```

750 Y(I,J)=Y(IMC,J)
C *****
C SPECIFY THE GRID CONTROL FUNCTIONS.
C *****
C DO 107 J=2,JMAX
C
C F2(1,J)=(Y(1,J+1)+Y(1,J-1)-2.*Y(1,J))
1 / (Y(1,J+1)+Y(1,J-1))
C
C F2(IMC,J)=(Y(IMC,J+1)+Y(IMC,J-1)-2.*Y(IMC,J))
1 / (Y(IMC,J+1)+Y(IMC,J-1))
C
C F2(IPI,J)=(Y(IMB,J+1)+Y(IMB,J-1)-2.*Y(IMB,J))
1 / (Y(IMB,J+1)+Y(IMB,J-1))
C
C DO 108 I=1,IMAX
C 108 RD(I)=DELTA-RX(I)
C
C DO 110 J=2,JMAX
C DO 110 I=2,IMAX
C F2(I,J)=(F2(IMC,J)+(RD(IMB)-RD(I))+F2(IPI,J)*(RD(I)-RD(IMC)))
1 / (RD(IMB)-RD(IMC))
C IF(I.GT.IMB) F2(I,J)=F2(IPI,J)
C IF(I.LT.IMA) F2(I,J)=F2(1,J)
C 110 CONTINUE
C
C DO 297 I=2,IMAX
C 297 F1(I)=(X(I+1)+X(I-1)-2.*X(I))/(X(I+1)-X(I-1))
C
C WRITE(10,2005) (Y(IMC,J),J=1,JP1)
C *****
C START ITERATION PROCESS FOR GRID GENERATION.
C *****
C DO 300 IT=1,ITMAX
C DRMAX=0.
C
C DO 100 I=2,IMAX
C DO 100 J=2,JMAX
C XXI=0.5*(X(I+1)+X(I-1))
C YXI=0.5*(Y(I+1,J)+Y(I-1,J))
C XET=0.
C YET=0.5*(Y(I,J+1)+Y(I,J-1))
C G11=XXI*XET+YXI*YET
C G22=XET*XET+YET*YET
C G12=XXI*XET+YXI*YET
C G=(G11*G22-G12*G12)
C AJ(I,J)=DSQRT(G)
C A11(I,J)=G22/G
C A22(I,J)=G11/G
C A12(I,J)=-G12/G
C 100 CONTINUE
C
C DO 500 I=2,IMAX
C
C *****
C PRINT OUTPUT IN EACH ITERATION PROCESS.
C *****
C WRITE(10,189) IT,DRMAX
C 189 FORMAT(5X,'IT=',15,5X,'DRMAX=',D12.4)
C
C PRINT 999, IT,DRMAX
C 999 FORMAT(5X,'NO. OF ITERATION = ',15,5X,'DRMAX=',D12.4)
C
C 300 CONTINUE
C *****
C SPECIFY BOUNDARY CONDITIONS.
C *****
C F1(1)=F1(2)
C F1(IPI)=F1(IMAX)

```

```

A1=F1(I)
IF(DABS(A1).LT.EPS) A1=EPS
EPA=DEXP(A1)
C
C DO 502 J=2,JMAX
C A2=F2(I,J)
C IF(DABS(A2).LT.EPS) A2=EPS
C EPA=DEXP(A2)
C SY=0.5*A12(I,J)*(Y(I+1,J+1)+Y(I-1,J-1)-Y(I+1,J-1)-Y(I-1,J+1))
C AY(J)=-2.*A22(I,J)*EPA*A2/(EPB-1./EPB)
C BY(J)=-2.*A11(I,J)*A1*(EPA+1./EPA)/(EPA-1./EPA)
1 +2.*A22(I,J)*A2*(EPB+1./EPB)/(EPB-1./EPB)
C CY(J)=-2.*A22(J,J)/EPA*A2/(EPB-1./EPB)
C DY(J)=-2.*A11(I,J)*A1/(EPA-1./EPA)*(Y(I-1,J)+EPA
2 +Y(I+1,J)/EPA)+SY
C 502 CONTINUE
C
C DY(2)=DY(2)-AY(2)*Y(I,1)
C DY(JMAX)=DY(JMAX)-CY(JMAX)*Y(I,JP1)
C
C CALL TRIDAG(2,JMAX,AY,BY,CY,DY,TV)
C
C DO 501 J=2,JMAX
C YDIFF=DABS(TY(J)-Y(I,J))
C IF(YDIFF.GT.DRMAX) DRMAX=YDIFF
C Y(I,J)=Y(I,J)+RF*(TY(J)-Y(I,J))
C 501 CONTINUE
C
C 500 CONTINUE
C
C SPECIFY THE BOUNDARY CONDITIONS.
C
C DO 130 J=1,JP1
C Y(1,J)=Y(2,J)
C Y(IP1,J)=Y(IMAX,J)
C 130 CONTINUE
C
C *****
C PRINT OUTPUT IN EACH ITERATION PROCESS.
C *****
C WRITE(10,189) IT,DRMAX
C 189 FORMAT(5X,'IT=',15,5X,'DRMAX=',D12.4)
C
C PRINT 999, IT,DRMAX
C 999 FORMAT(5X,'NO. OF ITERATION = ',15,5X,'DRMAX=',D12.4)
C
C 300 CONTINUE
C *****
C SPECIFY BOUNDARY CONDITIONS.
C *****
C F1(1)=F1(2)
C F1(IP1)=F1(IMAX)

```

END

```
A-72
DO 301 J=1,JPI
F2(1,J)=F2(2,J)
301 F2(IP1,J)=F2(IMAX,J)
C *****
C REARRANGE DATA *****
C *****
C DO 123 I=1,IP1
DO 123 J=1,JPI
YC(I,J)=Y(I,J)
123 XC(I,J)=X(I)
C *****
C PRINT OUTPUT AFTER ITERATION PROCESS *****
C *****
C DO 302 I=1,IP1
WRITE(10,601) I,X(I)
601 FORMAT(5X,'STATION ',15,5X,'X =',F8.4)
WRITE(10,2005)(Y(I,J),J=1,JPI)
302 CONTINUE
C *****
C WRITE(10,2007)(X(I),I=1,IP1)
2007 FORMAT(6F12.8)
C *****
C DO 250 J=1,JPI
250 WRITE(10,2005)(Y(I,J),I=1,IP1)
2005 FORMAT(4D14.6)
C *****
C CLOSE(10)
C *****
C SAVE DATA FOR CALCULATION *****
C *****
C OPEN(UNIT=11,FILE=FJNAME(10BODY))
WRITE(11,2009)((XC(I,J),I=1,IP1),J=1,JPI)
WRITE(11,2009)((YC(I,J),I=1,IP1),J=1,JPI)
2009 FORMAT(6D13.6)
C *****
C STOP
C END
C SUBROUTINE TRIDAG(IF,L,A,B,C,D,V)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION A(70),B(70),C(70),D(70),V(70),BETA(70),GAMMA(70)
BETA(IF)=B(IF)
GAMMA(IF)=D(IF)/BETA(IF)
IFPI=IF+1
DO 1 I=IFPI,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
```

```

C *****
C THIS PROGRAM PROVIDES INPUT DATA
C FOR THE PROGRAM IOWA.FTN
C *****
C
C IMPLICIT REAL*8(A-H,O-Z)
C
C CHARACTER*7 F$NAME(10)
C
C DATA F$NAME(1) , A1DAT , F$NAME(2) , A2DAT , /
C $ F$NAME(3) , A3DAT , F$NAME(4) , A4DAT , /
C $ F$NAME(5) , A5DAT , F$NAME(6) , A6DAT , /
C $ F$NAME(7) , A7DAT , F$NAME(8) , A8DAT , /
C $ F$NAME(9) , A9DAT , F$NAME(10) , A0DAT , /
C *****
C
C *****
C AXISYMMETRIC AND TWO-DIMENSIONAL BODIES.
C *****
C
C *****
C AXISYMMETRIC BODIES.
C *****
C
C * IBODY=1 ; AFTERBODY1 OF DTNDSRDC (TURBULENT) .
C * IBODY=2 ; AFTERBODY2 OF DTNDSRDC (TURBULENT) .
C * IBODY=3 ; AFTERBODY3 OF DTNDSRDC (TURBULENT) .
C * IBODY=5 ; AFTERBODY5 OF DTNDSRDC (TURBULENT) .
C * IBODY=6 ; 6 TO 1 RE-MODIFIED SPHEROID (TURBULENT) .
C * IBODY=7 ; F57 BODY (TURBULENT) .
C *****
C
C *****
C TWO-DIMENSIONAL BODIES.
C *****
C
C * IBODY=8 ; 10% THICK ARC AIRFOIL (LAMINAR) .
C * IBODY=9 ; 5% THICK ARC AIRFOIL (LAMINAR) .
C * IBODY=10 ; FLAT-PLATE (TURBULENT) .
C * IBODY=11 ; FLAT-PLATE (LAMINAR) .
C *****
C
C *****
C READ BODY TYPE FROM THE TERMINAL.
C *****
C
C *****
C WRITE(*,*) 'WHAT KIND OF BODY DO YOU WANT ?'
C WRITE(*,*) 'TYPE 1 FOR AFTERBODY1 OF DTNDSRDC'
C WRITE(*,*) 'TYPE 2 FOR AFTERBODY2 OF DTNDSRDC'
C WRITE(*,*) 'TYPE 3 FOR AFTERBODY3 OF DTNDSRDC'
C WRITE(*,*) 'TYPE 5 FOR AFTERBODY5 OF DTNDSRDC'
C WRITE(*,*) 'TYPE 6 FOR 6 TO 1 RE-MODIFIED SPHEROID'
C WRITE(*,*) 'TYPE 7 FOR F57 BODY'
C WRITE(*,*) 'TYPE 8 FOR 10% THICK ARC AIRFOIL'
C WRITE(*,*) 'TYPE 9 FOR 5% THICK ARC AIRFOIL'
C *****

```

```

C READ(*,*) IBODY
C
C REYNOLDS NUMBERS
C
C IF (IBODY.EQ.1) THEN
C   RENUM=6600000.D0
C ELSE IF (IBODY.EQ.2) THEN
C   RENUM=6800000.D0
C ELSE IF (IBODY.EQ.3) THEN
C   RENUM=6000000.D0
C ELSE IF (IBODY.EQ.5) THEN
C   RENUM=9300000.D0
C ELSE IF (IBODY.EQ.6) THEN
C   RENUM=1300000.D0
C ELSE IF (IBODY.EQ.7) THEN
C   RENUM=1200000.D0
C END IF
C
C NUMBER OF GRID POINTS IN X AND Y-DIRECTION.
C
C IMAX=151
C JMAX=36
C
C LEADING EDGE OF THE BODY ( I=M1 ).
C STARTING CALCULATION POINT OF THE WAKE ( I=M2 ).
C
C M1=21
C M2=122
C
C LAMINAR (IFLOW=1) OR TURBULENT FLOW (IFLOW=2) .
C IFLOW=2
C
C 2-DIMENSIONAL BODY (ICORD=1) OR AXISYMMETRIC BODY (ICORD=2) .
C ICORD=2
C
C TURBULENT INTENSITY AND LENGTH SCALE.
C
C TURBIN=0.5
C TUSCALE=0.001D0
C
C UNDER-RELAXATION FACTORS
C
C URFU=0.7D0
C URFB=0.7D0
C URFC=0.8D0
C URFD=0.8D0
C URFVIS=0.8D0
C URFPR=0.5
C
C MAXIMUM ITERATION(ITMAX) AND INNER ITERATION(NINER) .
C
C ITMAX=500
C NINER=3
C ITOUT=100
C
C NUMBER OF SWEEPS.
C

```

```
C      NSWPU=5
      NSWPV=5
      NSWPK=5
      NSWPD=5
C      SAVE DATA FOR THE FLOW CALCULATION.
C      OPEN(UNIT=12,FILE=FSNAME(IBODY))
C      WRITE(12,1)  IMAX,JMAX,M1,M2
C      1  FORMAT(4I5)
C      WRITE(12,2)  IFLOW,ICORD
C      2  FORMAT(2I5)
C      WRITE(12,3)  RENUM,TURBIN,TUSCALE
C      3  FORMAT(3E12.4)
C      WRITE(12,4)  ITMAX,NINER,ITOUT
C      4  FORMAT(3I5)
C      WRITE(12,5)  URFU,URFV,UREK,URFD,URFVIS,URFPR
C      5  FORMAT(6F7.3)
C      WRITE(12,6)  NSWPU,NSWPV,NSWPK,NSWPD
C      6  FORMAT(4I5)
C      CLOSE(12)
C      STOP
      END
```

```

C *****
C THIS PROGRAM CALCULATES LAMINAR AND TURBULENT FLOW *
C PAST A TWO-DIMENSIONAL OR AXISYMMETRIC BODIES *
C USING THE K-E TURBULENCE MODEL AND WALL FUNCTION *
C METHOD WITH A SPACE MARCHING TECHNIQUES. *
C *****
C
C *****
C PROGRAMMED BY S. K. CHOI *
C *****
C
C *****
C THE BASIC FEATURE OF THIS CODE.
C *****
C TWO-VELOCITY STAGGERED GRID SYSTEM.
C (MODIFIED VERSION OF METHOD OF MALISKA AND RAITHBY.) *
C *****
C SIMPLER NUMERICAL ALGORITHM.
C *****
C SPACE MARCHING TECHNIQUE.
C *****
C BODY-FITTED COORDINATE USING PARTIAL TRANSFORMATION.
C ( CARTESIAN, CYLINDRICAL BASE COORDINATE. ) *
C *****
C PROGRAM HAWKEYE
C *****
C IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C *****
C CHARACTER*10 FNAME(10)
C CHARACTER*10 F2NAME(10)
C CHARACTER*10 F3NAME(10)
C CHARACTER*10 F4NAME(10)
C CHARACTER*10 F5NAME(10)
C *****
C PARAMETER(ID=152,JD=45)
C *****
C COMMON/PRES1/PR(ID,JD),PP(ID,JD),PRO(ID,JD)
C COMMON/PRES2/CKUE(ID,JD),CYUE(ID,JD)
C COMMON/PRES3/CKXU(ID,JD),CYXU(ID,JD)
C COMMON/PRES4/CKVE(ID,JD),CYVE(ID,JD)
C COMMON/PRES5/CKVN(ID,JD),CYVN(ID,JD)
C COMMON/PRES6/QXUE(ID,JD),QXVE(ID,JD)
C COMMON/PRES7/QYUN(ID,JD),QYVN(ID,JD)
C COMMON/PRES8/BCU(ID,JD),BCV(ID,JD)
C COMMON/PRES9/SPP(ID,JD),SPR(ID,JD)
C COMMON/GE0C/KC(ID,JD),YC(ID,JD)
C COMMON/GE0P/XP(ID,JD),YP(ID,JD)
C COMMON/GE0E/XE(ID,JD),YE(ID,JD)
C COMMON/GE0N/XN(ID,JD),YN(ID,JD)
C COMMON/GE0I/YE2(ID,JD),YE3(ID,JD)
C COMMON/GE0J/YN2(ID,JD),YN3(ID,JD)
C COMMON/GE03/YP2(ID,JD),YP3(ID,JD)
C COMMON/GE04/R(ID,JD),VOL(ID,JD)
C COMMON/WALL1/UTAU(ID,JD),CFS(ID,JD)
C COMMON/WALL2/YPLUSN(ID,JD),TAUN(ID,JD)
C COMMON/DENSITY/DEN(ID,JD)
C COMMON/VELL/UE(ID,JD),VE(ID,JD)

```

```

COMMON/VEL2/UN(ID,JD),VN(ID,JD)
COMMON/VEL3/UHE(ID,JD),VHE(ID,JD)
COMMON/VEL4/UHN(ID,JD),VHN(ID,JD)
COMMON/VEL5/UUN(ID,JD),VUN(ID,JD)
COMMON/VEL6/UUE(ID,JD),VUE(ID,JD)
COMMON/TURB1/AKE(ID,JD),ADS(ID,JD)
COMMON/TURB2/VIS(ID,JD),GEN(ID,JD)
COMMON/ST1/SCUE(ID,JD),SPUE(ID,JD)
COMMON/ST2/SCVE(ID,JD),SPVE(ID,JD)
COMMON/ST3/SCUN(ID,JD),SPUN(ID,JD)
COMMON/ST4/SCVN(ID,JD),SPVN(ID,JD)
COMMON/ST5/SCK(ID,JD),SPK(ID,JD)
COMMON/ST6/SCD(ID,JD),SPD(ID,JD)
COMMON/ST7/SUKD(ID,JD),SPKD(ID,JD)
COMMON/ST8/SUDDI(ID,JD),SPDD(ID,JD)
COMMON/COEF1/AEE(ID,JD),AME(ID,JD),ANE(ID,JD)
COMMON/COEF2/AEE(ID,JD),APUE(ID,JD),APVE(ID,JD)
COMMON/COEF3/AEN(ID,JD),ANN(ID,JD),ANN(ID,JD)
COMMON/COEF4/ASN(ID,JD),AFUN(ID,JD),APVN(ID,JD)
COMMON/COEF5/AEK(ID,JD),AWK(ID,JD),ANK(ID,JD)
COMMON/COEF6/ASK(ID,JD),APK(ID,JD),ADK(ID,JD)
COMMON/COEF7/AED(ID,JD),AWD(ID,JD),AND(ID,JD)
COMMON/COEF8/ASD(ID,JD),APD(ID,JD),ADD(ID,JD)
COMMON/AAA/AA(JD),BB(JD),CC(JD),DD(JD),T(JD)
COMMON/CONS1/IMAX,JMAX,IPI,JPI,JPP1
COMMON/CONS2/NSWPU,NSWVP,NSWPP,NSWPK,NSMPD
COMMON/CONS3/RESUE,RESVE,RESAK,RESAD,RESMH
COMMON/CONS4/URFU,URFV,URFK,URFD,URFVIS,URFPR
COMMON/CONS5/RENUM,DENSIT,VISCOS,PRANDT
COMMON/CONS6/IBODY,IFLOW,ICORD
COMMON/CONS7/NITER,MAXIT,ITMAX,NINER,ITOUT
COMMON/CONS8/CHU,CD,C1,C2,ELOG,CAPPA,PRTE,PRED
COMMON/CONS9/PI,GREAT
COMMON/CONS10/UIN,TEIN,EDIN,FLOWIN,ALAMDA
COMMON/CONS11/M1,M2,MIP1,MIM1,M2P1,M2M1
C *****
C DATA FNAME(1)/'A1BDAT',FNAME(2)/'A2BDAT',
C $ FNAME(3)/'A3BDAT',FNAME(4)/'A4BDAT',
C $ FNAME(5)/'A5BDAT',FNAME(6)/'A6BDAT',
C $ FNAME(7)/'A7BDAT',FNAME(8)/'A8BDAT',
C $ FNAME(9)/'A9BDAT',FNAME(10)/'A0BDAT',
C *****
C DATA F2NAME(1)/'A1OUT1',F2NAME(2)/'A2OUT1',
C $ F2NAME(3)/'A3OUT1',F2NAME(4)/'A4OUT1',
C $ F2NAME(5)/'A5OUT1',F2NAME(6)/'A6OUT1',
C $ F2NAME(7)/'A7OUT1',F2NAME(8)/'A8OUT1',
C $ F2NAME(9)/'A9OUT1',F2NAME(10)/'A0OUT1',
C *****
C DATA F3NAME(1)/'A1OUT2',F3NAME(2)/'A2OUT2',
C $ F3NAME(3)/'A3OUT2',F3NAME(4)/'A4OUT2',
C $ F3NAME(5)/'A5OUT2',F3NAME(6)/'A6OUT2',
C $ F3NAME(7)/'A7OUT2',F3NAME(8)/'A8OUT2',
C $ F3NAME(9)/'A9OUT2',F3NAME(10)/'A0OUT2',
C *****
C DATA F4NAME(1)/'A1OUT3',F4NAME(2)/'A2OUT3',
C $ F4NAME(3)/'A3OUT3',F4NAME(4)/'A4OUT3',
C $ F4NAME(5)/'A5OUT3',F4NAME(6)/'A6OUT3',
C $ F4NAME(7)/'A7OUT3',F4NAME(8)/'A8OUT3',
C $ F4NAME(9)/'A9OUT3',F4NAME(10)/'A0OUT3',
C *****
C

```

```

C     DATA F$NAME(1)='A1DAT', F$NAME(2)='A2DAT',
C     $ F$NAME(3)='A3DAT', F$NAME(4)='A4DAT',
C     $ F$NAME(5)='A5DAT', F$NAME(6)='A6DAT',
C     $ F$NAME(7)='A7DAT', F$NAME(8)='A8DAT',
C     $ F$NAME(9)='A9DAT', F$NAME(10)='A0DAT',
C
C     *****
C     * MAIN PROGRAM
C     *****
C
C     *****
C     * AXISYMMETRIC AND TWO-DIMENSIONAL BODIES.
C     *****
C
C     * AXISYMMETRIC BODIES.
C     *****
C
C     * IBODY=1 ; AFTERBODY1 OF DTNSRDC (TURBULENT).
C     * IBODY=2 ; AFTERBODY2 OF DTNSRDC (TURBULENT).
C     * IBODY=3 ; AFTERBODY3 OF DTNSRDC (TURBULENT).
C     * IBODY=5 ; AFTERBODY5 OF DTNSRDC (TURBULENT).
C     * IBODY=6 ; 6 TO 1 RE-MODIFIED SPHEROID (TURBULENT).
C     * IBODY=7 ; F57 BODY (TURBULENT).
C
C     * TWO-DIMENSIONAL BODIES.
C     *****
C
C     * IBODY=8 ; 10% THICK ARC AIRFOIL (LAMINAR).
C     * IBODY=9 ; 5% THICK ARC AIRFOIL (LAMINAR).
C
C     * IBODY=10 ; FLAT-PLATE (TURBULENT).
C     * IBODY=11 ; FLAT-PLATE (LAMINAR).
C
C     *****
C
C     *****
C     * READ BODY TYPE FROM THE TERMINAL.
C     *****
C
C     WRITE(*,*) 'WHAT KIND OF BODY DO YOU WANT ?'
C     WRITE(*,*) 'TYPE 1 FOR AFTERBODY1 OF DTNSRDC'
C     WRITE(*,*) 'TYPE 2 FOR AFTERBODY2 OF DTNSRDC'
C     WRITE(*,*) 'TYPE 3 FOR AFTERBODY3 OF DTNSRDC'
C     WRITE(*,*) 'TYPE 5 FOR AFTERBODY5 OF DTNSRDC'
C     WRITE(*,*) 'TYPE 6 FOR 6 TO 1 RE-MODIFIED SPHEROID'
C     WRITE(*,*) 'TYPE 7 FOR F57 BODY'
C     WRITE(*,*) 'TYPE 8 FOR 10% THICK ARC AIRFOIL'
C     WRITE(*,*) 'TYPE 9 FOR 5% THICK ARC AIRFOIL'
C
C     READ(*,*) IBODY
C
C     *****
C     * READ INPUT DATA FROM FILE (F$NAME)
C     *****

```

```

C     OPEN (UNIT=12, FILE=F$NAME (IBODY))
C
C     READ (12,1) IMAX, JMAX, M1, M2
C     FORMAT (4I5)
C
C     READ (12,2) IFLOW, ICORD
C     FORMAT (2I5)
C
C     READ (12,3) RENUM, TURBIN, TUSCALE
C     FORMAT (3E12.4)
C
C     READ (12,4) ITHAX, NINER, ITOUT
C     FORMAT (3I5)
C
C     READ (12,5) URFU, URFV, URFK, URFD, URFVIS, URFPR
C     FORMAT (6F7.3)
C
C     READ (12,6) NSWPU, NSWPV, NSWPK, NSWPD
C     FORMAT (4I5)
C
C     CLOSE (12)
C
C     *****
C     * GEOMETRIC CONSTANTS.
C     *****
C
C     IPI=IMAX+1
C     JPI=JMAX+1
C     JPP1=JMAX+2
C     IM1=IMAX-1
C     JM1=JMAX-1
C     M2M1=M2-1
C     M1M1=M1-1
C     M1P1=M1+1
C
C     NUMBER OF ITERATIONS.
C
C     NITER=0
C     MAXIT=NITER+ITMAX
C     ITPR=1
C
C     FLUID PROPERTIES.
C
C     DENSIT=1.
C     VISCOS=1./RENUM
C     PRANDT=0.9D0
C
C     TURBULENT CONSTANTS.
C
C     CMU=0.09
C     CD=1.
C     C1=1.44
C     C2=1.92
C     CAPPA=0.4187
C     ELOG=9.793
C     PRED=CAPPA*CAPPA/(C2-C1)/(CMU**0.5D0)

```



```

C C PRED=1.3
C C PRTE=1.00
C C INLET CONDITIONS.
C C ALNGTH=1.00
C C UIN=1.00
C C TEIN=1.5D0*(TURBIN/100.00)**2
C C ALAMDA=TUSCALE*ALENGTH
C C EDIN=CHU*TEIN**1.5/ALAMDA
C C GREAT=1.00
C C PI=3.141592653589793D0
C C *****
C C OPEN FILES FOR OUTPUT
C C *****
C C OPEN(UNIT=8, FILE=FNAME(IBODY))
C C OPEN(UNIT=7, FILE=FNAME(IBODY))
C C OPEN(UNIT=9, FILE=FNAME(IBODY))
C C *****
C C *****
C C ***** OFFSET OF NUMERICAL GRID.
C C *****
C C * XE, YE----- EAST CELL SURFACE CALCULATION POINT (UE, VE) .
C C * XN, YN----- NORTH CELL SURFACE CALCULATION POINT (UN, VN) .
C C * XP, YP----- CELL CENTER CALCULATION POINT (PR, PP, AKE, ADS) .
C C * XC, YC----- CELL CONTROL VOLUME.
C C *****
C C *****
C C 2006 FORMAT(6D13.6)
C C OPEN(UNIT=11, FILE=FINAME(IBODY))
C C READ(11, 2006) ((XC(I, J), I=1, IMAX), J=1, JMAX)
C C READ(11, 2006) ((YC(I, J), I=1, IMAX), J=1, JMAX)
C C CLOSE(11)
C C DO 110 J=1, JMAX
C C XC(1, J)=0.5*(XC(1, J)+XC(2, J))
C C 110 YC(1, J)=0.5*(YC(1, J)+YC(2, J))
C C XE(1, 1)=XC(1, 1)
C C XN(1, 1)=XC(1, 1)
C C XP(1, 1)=XC(1, 1)
C C YE(1, 1)=YC(1, 1)
C C YN(1, 1)=YC(1, 1)
C C YP(1, 1)=YC(1, 1)
C C XC(1, JPI)=XC(1, JMAX)
C C YC(1, JPI)=YC(1, JMAX)+(YC(1, JMAX)-YC(1, JMAX-1))
C C DO 111 J=2, JPI
C C XE(1, J)=XC(1, J)
C C XN(1, J)=XC(1, J)
C C XP(1, J)=XC(1, J)

```

```

C C YE(1, J)=0.5*(YC(1, J)+YC(1, J-1))
C C YN(1, J)=YC(1, J)
C C 111 YP(1, J)=0.5*(YC(1, J)+YC(1, J-1))
C C DO 112 I=2, IMAX
C C XE(I, 1)=XC(I, 1)
C C XN(I, 1)=0.5*(XC(I, 1)+XC(I-1, 1))
C C XP(I, 1)=0.5*(XC(I, 1)+XC(I-1, 1))
C C YE(I, 1)=YC(I, 1)
C C YN(I, 1)=0.5*(YC(I, 1)+YC(I-1, 1))
C C YP(I, 1)=0.5*(YC(I, 1)+YC(I-1, 1))
C C DO 112 J=2, JMAX
C C XE(I, J)=0.5*(XC(I, J)+XC(I, J-1))
C C XN(I, J)=0.5*(XC(I, J)+XC(I-1, J))
C C XP(I, J)=0.25*(XC(I, J)+XC(I, J-1)+XC(I-1, J)+XC(I-1, J-1))
C C YE(I, J)=0.5*(YC(I, J)+YC(I, J-1))
C C YN(I, J)=0.5*(YC(I, J)+YC(I-1, J))
C C YP(I, J)=0.25*(YC(I, J)+YC(I, J-1)+YC(I-1, J)+YC(I-1, J-1))
C C DO 113 I=2, IMAX
C C XC(I, JPI)=XC(I, JMAX)
C C XE(I, JPI)=XE(I, JMAX)
C C XN(I, JPI)=XN(I, JMAX)
C C XP(I, JPI)=XP(I, JMAX)
C C YC(I, JPI)=YC(I, JPI)
C C YE(I, JPI)=YE(I, JPI)
C C YN(I, JPI)=YN(I, JPI)
C C YP(I, JPI)=YP(I, JPI)
C C DO 113 J=1, JPI
C C XC(I, J)=XC(IMAX, J)+0.5*(XC(IMAX, J)-XC(IMAX-1, J))
C C XE(I, J)=XC(I, J)
C C XN(I, J)=XC(I, J)
C C XP(I, J)=XC(I, J)
C C YC(I, J)=YC(IMAX, J)
C C YE(I, J)=YE(IMAX, J)
C C YN(I, J)=YN(IMAX, J)
C C YP(I, J)=YP(IMAX, J)
C C DO 116 I=1, JPI
C C YE2(I)=YE(I, 2)-YE(I, 1)
C C YE3(I)=YE(I, 3)-YE(I, 1)
C C YN2(I)=YN(I, 2)-YN(I, 1)
C C YN3(I)=YN(I, 3)-YN(I, 1)
C C YP2(I)=YP(I, 2)-YP(I, 1)
C C YP3(I)=YP(I, 3)-YP(I, 1)
C C 116 CONTINUE
C C RADIUS FOR AXISYMMETRIC BODY.
C C DO 117 I=1, JPI
C C IF(ICORD.EQ.1) THEN
C C R(I, J)=1.00
C C ELSE IF(ICORD.EQ.2) THEN
C C R(I, J)=YC(I, J)
C C END IF
C C 117 CONTINUE

```

```

C          C          VOLUME SIZE OF EACH CONTROL VOLUME CELL.
C          C          PR(I,J)=0.D0
C          C          PRO(I,J)=0.D0
C          C          PP(I,J)=0.D0
C          C          KINETIC ENERGY AND DISSIPATION RATE.
C          C          AKE(I,J)=0.D0
C          C          ADS(I,J)=0.D0
C          C          DENSITY AND VISCOSITY
C          C          DEN(I,J)=DENSIT
C          C          VIS(I,J)=VISCOS
C          C          COEFFICIENTS OF PRESSURE TERM IN
C          C          MOMENTUM EQUATIONS.
C          C          CXUE(I,J)=0.D0
C          C          CYUE(I,J)=0.D0
C          C          CXVE(I,J)=0.D0
C          C          CYVE(I,J)=0.D0
C          C          CXUN(I,J)=0.D0
C          C          CYUN(I,J)=0.D0
C          C          CXVN(I,J)=0.D0
C          C          CYVN(I,J)=0.D0
C          C          QXUE(I,J)=0.D0
C          C          QXVE(I,J)=0.D0
C          C          QYUN(I,J)=0.D0
C          C          QYVN(I,J)=0.D0
C          C          COEFFICIENTS OF PRESSURE AND
C          C          PRESSURE CORRECTION EQUATION.
C          C          BCU(I,J)=0.D0
C          C          BCV(I,J)=0.D0
C          C          SOURCE TERMS.
C          C          FOR VELOCITY, UE AND VE.
C          C          SCUE(I,J)=0.D0
C          C          SPUE(I,J)=0.D0
C          C          SCVE(I,J)=0.D0
C          C          SPVE(I,J)=0.D0
C          C          FOR VELOCITY, UN AND VN.
C          C          SCUN(I,J)=0.D0
C          C          SPUN(I,J)=0.D0
C          C          SCVN(I,J)=0.D0
C          C          SPVN(I,J)=0.D0
C          C          FOR KINETIC ENERGY, AKE.
C          C          SCK(I,J)=0.D0
C          C          SPK(I,J)=0.D0

```

```

C          C          VOLUME SIZE OF EACH CONTROL VOLUME CELL.
C          C          DO 118 I=2,IP1
C          C          DO 118 J=2,JP1
C          C          IF(ICORD.EQ.1) THEN
C          C          VOL(I,J)=0.5*((XC(I,J)-XC(I-1,J))*(YC(I-1,J)-YC(I,J))
C          C          - (XC(I-1,J)-XC(I,J))*(YC(I,J)-YC(I-1,J-1)))
C          C          ELSE IF(ICORD.EQ.2) THEN
C          C          VOL(I,J)=1.D0/6.D0*
C          C          1 ((XC(I,J-1)-XC(I,J))
C          C          2 *(YC(I,J-1)-YC(I,J))**2+YC(I,J-1)*YC(I,J))
C          C          3 +(XC(I,J)-XC(I-1,J))
C          C          4 *(YC(I,J)-YC(I-1,J))**2+YC(I-1,J)*YC(I-1,J))
C          C          5 +(XC(I-1,J)-XC(I-1,J-1))
C          C          6 *(YC(I-1,J)-YC(I-1,J-1))**2+YC(I-1,J)*YC(I-1,J-1))
C          C          7 +(XC(I-1,J-1)-XC(I-1,J-1))
C          C          8 *(YC(I-1,J-1)-YC(I-1,J-1))**2+YC(I-1,J-1)*YC(I-1,J-1))
C          C          END IF
C          C          118 CONTINUE
C          C          DO 119 J=1,JP1
C          C          119 VOL(I,J)=0.D0
C          C          DO 120 I=1,IP1
C          C          120 VOL(I,1)=0.D0
C          C          *****
C          C          SET VARIABLES TO ZERO.
C          C          *****
C          C          DO 14 I=1,IP1
C          C          WALL SHEAR STRESS
C          C          TAUN(I)=1.D0
C          C          DO 14 J=1,JP1
C          C          CARTESIAN ( OR CYLINDRICAL ) VELOCITY COMPONENTS.
C          C          UE(I,J)=0.D0
C          C          VE(I,J)=0.D0
C          C          UN(I,J)=0.D0
C          C          VN(I,J)=0.D0
C          C          PSEUDO-VELOCITY COMPONENTS.
C          C          UHE(I,J)=0.D0
C          C          VHE(I,J)=0.D0
C          C          UHN(I,J)=0.D0
C          C          VHN(I,J)=0.D0
C          C          CONTRAVARIANT VELOCITY COMPONENTS.
C          C          UUE(I,J)=0.D0
C          C          VVE(I,J)=0.D0
C          C          UUN(I,J)=0.D0
C          C          VVN(I,J)=0.D0

```

```

C SUKD(I,J)=0.DO
C SPKD(I,J)=0.DO
C FOR DISSIPATION RATE, ADS.
C
C SCD(I,J)=0.DO
C SPD(I,J)=0.DO
C SUDD(I,J)=0.DO
C SPDD(I,J)=0.DO
C
C FOR PRESSURE AND PRESSURE CORRECTION EQUATION.
C
C SPP(I,J)=0.DO
C SPR(I,J)=0.DO
C
C 14 CONTINUE
C
C *****
C INITIAL PROFILES AND INLET CONDITIONS.
C *****
C
C ITM=1
C IF(ITM.EQ.1) GO TO 15
C
C INITIAL PROFILES FROM FILE FVINPOT
C IF AVAILABLE
C
C OPEN(UNIT=11,FILE='FVINPOT')
C DO 16 I=1,IPI
C DO 16 J=1,JPI
C 16 READ(11,2008) UE(I,J),VE(I,J),UN(I,J),VN(I,J)
C 1
C CLOSE(11)
C 2008 FORMAT(7E12.4)
C
C 15 CONTINUE
C *****
C INLET CONDITIONS
C *****
C
C FLOWIN=0.DO
C ARDEN=0.DO
C ARDENT=0.DO
C XMONIN=0.DO
C
C DO 20 J=1,JPI
C UE(1,J)=UIN
C VE(1,J)=0.DO
C UN(1,J)=UIN
C VN(1,J)=0.DO
C
C UME(1,J)=UIN
C UHN(1,J)=UIN
C
C VVE(1,J)=0.
C VVN(1,J)=0.

```

```

C
C AKE(1,J)=TEIN
C ADS(1,J)=EDIN
C
C 20 CONTINUE
C
C COMPUTE INLET MEAN VELOCITY, MASS FLUX
C AND MOMENTUM FLUX.
C
C DO 201 J=2,JMAX
C ARDEN=DEN(1,J)*0.5*(R(1,J)+R(1,J-1))*(YC(1,J)-YC(1,J-1))
C XMONIN=XMONIN+ARDEN*UE(1,J)*UE(1,J)
C ARDENT=ARDENT+ARDEN
C UMEAN=FLOWIN+ARDEN*UE(1,J)
C 201 FLOWIN=FLOWIN+ARDEN*UE(1,J)
C
C COMPUTE INLET CONTRAVARIANT VELOCITY.
C
C DO 202 J=2,JPI
C UUE(1,J)=DEN(1,J)*0.5*(R(1,J)+R(1,J-1))*(YC(1,J)-YC(1,J-1))
C 1
C UUE(1,1)=0.
C
C INITIAL GUESS FOR YPLUS
C
C DO 205 I=2,IMAX
C 205 YPLASN(I)=11.DO
C
C *****
C RETURN POINT OF ITERATION.
C *****
C
C 4000 NITER=NITER+1
C
C RESUE=0.
C RESVE=0.
C RESUN=0.
C RESVN=0.
C RESAK=0.
C RESAD=0.
C RESMN=0.
C
C DO 19 J=1,JPI
C DO 21 I=1,IPI
C SPP(I,J)=0.DO
C PP(I,J)=0.DO
C 21 IF(NITER.LT.4) GO TO 19
C VE(1,J)=VE(2,J)
C VN(1,J)=VN(2,J)
C VVN(1,J)=VVN(2,J)
C AKE(1,J)=AKE(2,J)
C ADS(1,J)=ADS(2,J)
C 19 CONTINUE
C
C

```

```

*****
RETURN POINT OF SPACE MARCHING PROCESS.
*****
DO 3000 I=2, IMAX
  IF(ITN.EQ.2) GO TO 231
  IF(NITER.EQ.1) THEN
    DO 23 J=1, JPI
      UE(I,J)=UE(I-1,J)
      VE(I,J)=VE(I-1,J)
      UN(I,J)=UN(I-1,J)
      VN(I,J)=VN(I-1,J)
      UUE(I,J)=UUE(I-1,J)
      VVE(I,J)=VVE(I-1,J)
      UUN(I,J)=UUN(I-1,J)
      VVN(I,J)=VVN(I-1,J)
      AKE(I,J)=AKE(I-1,J)
      ADS(I,J)=ADS(I-1,J)
      UE(I+1,J)=UE(I,J)
      VE(I+1,J)=VE(I,J)
      UN(I+1,J)=UN(I,J)
      VN(I+1,J)=VN(I,J)
      UUE(I+1,J)=UUE(I,J)
      VVE(I+1,J)=VVE(I,J)
      UUN(I+1,J)=UUN(I,J)
      VVN(I+1,J)=VVN(I,J)
      AKE(I+1,J)=AKE(I,J)
      ADS(I+1,J)=ADS(I,J)
    23 CONTINUE
  END IF
  231 CONTINUE
  VE2SQ=VE2(I)*VE2(I)
  VE3SQ=VE3(I)*VE3(I)
  VN2SQ=VN2(I)*VN2(I)
  VN3SQ=VN3(I)*VN3(I)
  VP2SQ=VP2(I)*VP2(I)
  VP3SQ=VP3(I)*VP3(I)
  IF(IFLOW.EQ.1) GO TO 270
  *****
  CALCULATE THE EFFECTIVE EDDY VISCOSITY
  *****
  DO 240 J=2, JPI
    VISOLD=VIS(I,J)
    IF(ADS(I,J).EQ.0.DO) GO TO 241
    VIS(I,J)=CHU*DEN(I,J)*AKE(I,J)*AKE(I,J)/ADS(I,J)
    +VISCOS
    GO TO 242
  241 VIS(I,J)=VISCOS
    UNDER-RELAX VISCOSITY
  242 VIS(I,J)=URFVIS*VIS(I,J)+(1.DO-URFVIS)*VISOLD
  240 CONTINUE
    ALONG THE BOTTOM WALL AND SYMMETRY LINE.
    IF(I.LE.M1.OR.I.GE.M2) VIS(I,1)=VIS(I,2)
    IF(I.GT.M1.AND.I.LT.M2) VIS(I,1)=VISCOS
    VIS(I,1)=VIS(I,2)
    INLET AND OUTLET CONDITIONS
    IF(I.EQ.2) THEN
      DO 26 J=1, JPI
        26 VIS(I,J)=VIS(2,J)
      END IF
    IF(I.EQ.IMAX) THEN
      DO 261 J=1, JPI
        261 VIS(IPI,J)=VIS(IMAX,J)
      END IF
    IF(NITER.EQ.1) THEN
      DO 262 J=1, JPI
        262 VIS(I+1,J)=VIS(I,J)
      END IF
  270 CONTINUE
  *****
  STARTING POINT OF INNER ITERATION.
  *****
  DO 2000 ITERN=1, MINER
    *****
    CALCULATE THE COEFFICIENTS AND SOURCE TERMS FOR
    NORTH CELL SURFACE VELOCITIES.
    *****
    DO 380 J=2, JMAX
      COMPUTE GEOMETRIC CONSTANTS
      FOR EAST CONTROL VOLUME SURFACE

```

```

RE=R(I,J)
B11E=YE(I,J+1)-YE(I,J)
B12E=-XE(I,J+1)-XE(I,J)
B21E=-YN(I+1,J)-YN(I,J)
B22E=XN(I+1,J)-XN(I,J)
C
IF(J.EQ.2) THEN
B11E=YE(I,3)-YE(I,1)
B12E=-XE(I,3)-XE(I,1)
END IF
C
D11E=RE*RE*(B11E*B11E+B12E*B12E)
D12E=RE*RE*(B11E*B21E+B12E*B22E)
D21E=RE*RE*(B11E*B21E+B12E*B22E)
D22E=RE*RE*(B11E*B21E+B12E*B22E)
VOL=0.25*(VOL(I+1,J)+VOL(I+1,J+1)+VOL(I,J)+VOL(I,J+1))
C
IF(J.EQ.2) THEN
VOL=0.25*(VOL(I+1,2)+VOL(I+1,3)+VOL(I,2)+VOL(I,3))
1
+0.25*(VOL(I+1,2)+VOL(I,2))
END IF
C
FOR WEST CONTROL VOLUME SURFACE
C
RN=R(I-1,J)
B11W=YE(I-1,J+1)-YE(I-1,J)
B12W=-XE(I-1,J+1)-XE(I-1,J)
C
IF(J.EQ.2) THEN
B11W=YE(I-1,3)-YE(I-1,1)
B12W=-XE(I-1,3)-XE(I-1,1)
END IF
C
B21W=-YN(I,J)-YN(I-1,J)
B22W=XN(I,J)-XN(I-1,J)
D11W=RN*RN*(B11W*B11W+B12W*B12W)
D12W=RN*RN*(B11W*B21W+B12W*B22W)
D21W=RN*RN*(B11W*B21W+B12W*B22W)
D22W=RN*RN*(B11W*B21W+B12W*B22W)
VOLW=0.25*(VOL(I-1,J)+VOL(I-1,J+1)+VOL(I,J)+VOL(I,J+1))
C
IF(J.EQ.2) THEN
VOLW=0.25*(VOL(I-1,2)+VOL(I-1,3)+VOL(I,2)+VOL(I,3))
1
+0.25*(VOL(I-1,2)+VOL(I,2))
END IF
C
FOR NORTH CONTROL VOLUME SURFACE
C
RN=0.25*(R(I-1,J)+R(I-1,J+1)+R(I,J)+R(I,J+1))
B11N=YN(I,J+1)-YN(I,J)
B12N=-XN(I,J+1)-XN(I,J)
B21N=-YE(I,J+1)-YE(I-1,J+1)
B22N=XE(I,J+1)-XE(I-1,J+1)
D21N=RN*RN*(B11N*B21N+B12N*B22N)
D22N=RN*RN*(B11N*B21N+B22N*B22N)
VOLN=VOL(I,J+1)
C
FOR SOUTH CONTROL VOLUME SURFACE
C
RS=0.25*(R(I-1,J)+R(I-1,J-1)+R(I,J)+R(I,J-1))
B11S=YN(I,J)-YN(I,J-1)
B12S=-XN(I,J)-XN(I,J-1)
P21S=-YE(I,J)-YE(I-1,J)

```

```

B22S=XE(I,J)-XE(I-1,J)
D21S=RS*RS*(B11S*B21S+B12S*B22S)
D22S=RS*RS*(B11S*B21S+B12S*B22S)
VOLS=VOL(I,J)
C
FOR CENTER POINT
C
RP=0.5*(R(I,J)+R(I-1,J))
B11P=YP(I,J+1)-YP(I,J)
B12P=-XP(I,J+1)-XP(I,J)
B21P=-YC(I,J)-YC(I-1,J)
B22P=XZ(I,J)-XZ(I-1,J)
VOLP=0.5*(VOL(I,J)+VOL(I,J+1))
C
IF(J.EQ.2) THEN
VOLP=VOL(I,2)+0.5*VOL(I,3)
END IF
C
COMPUTE CELL SURFACE VALUES
C
UNN=0.5*(UE(I,J+1)+UE(I-1,J+1))
VNN=0.5*(VN(I,J)+VN(I,J+1))
USS=0.5*(UE(I,J)+UE(I-1,J))
VSS=0.5*(VN(I,J)+VN(I,J-1))
UEE=0.5*(UE(I,J)+UE(I,J+1))
VEF=0.5*(VN(I,J)+VN(I+1,J))
UWN=0.5*(UE(I-1,J)+UE(I-1,J+1))
VWN=0.5*(VN(I,J)+VN(I-1,J))
AKNN=AKE(I,J+1)
AKSS=AKE(I,J)
AKEE=0.25*(AKE(I,J)+AKE(I+1,J+1)+AKE(I+1,J))
AKNW=0.25*(AKE(I-1,J)+AKE(I-1,J+1)+AKE(I,J+1)+AKE(I,J))
VINN=VIS(I,J+1)
VISS=VIS(I,J)
VIEE=0.2500*(VIS(I,J)+VIS(I,J+1)+VIS(I-1,J+1)+VIS(I+1,J))
VIWN=0.2500*(VIS(I-1,J)+VIS(I-1,J+1)+VIS(I,J+1)+VIS(I,J))
DENN=DEN(I,J+1)
DESS=DEN(I,J)
DEEE=0.25*(DEN(I,J)+DEN(I,J+1)+DEN(I-1,J+1)+DEN(I+1,J))
DEWN=0.25*(DEN(I-1,J)+DEN(I-1,J+1)+DEN(I,J+1)+DEN(I,J))
C
COMPUTE CONVECTION MASS FLUX.
C
FN=0.5*(VNN(I,J)+VNN(I,J+1))
FS=0.5*(VNN(I,J)+VNN(I,J-1))
FE=0.5*(UUE(I,J)+UUE(I,J+1))
FW=0.5*(UUE(I-1,J)+UUE(I-1,J+1))
C
COMPUTE DIFFUSION COEFFICIENTS
C
DN=VINN*D22N/VOLN
DS=VISS*D22S/VOLS
DE=VIEE*D11E/VOLE
DW=VIMW*D11W/VOLW
C
COMPUTE COEFFICIENTS OF SOURCE TERM
C

```

```

C
20 SMP=FN-FS+FE-FW
1 CP=DMAX1(0.DO,SMP)
4 CPO=CP

C
C ASSEMBLE MAIN COEFFICIENTS
C
C ANN(I,J)=DMAX1(DABS(0.5D0*FN),DN)-0.5D0*FN
C ASN(I,J)=DMAX1(DABS(0.5D0*FS),DS)+0.5D0*FS
C AEN(I,J)=DMAX1(DABS(0.5D0*FE),DE)-0.5D0*FE
C AMN(I,J)=DMAX1(DABS(0.5D0*FW),DM)+0.5D0*FW

C
C COEFFICIENTS OF PRESSURE TERM IN MOMENTUM EQUATION
C
C CXUN(I,J)=RP*B11P
C CYUN(I,J)=RP*B21P
C CXVN(I,J)=RP*B12P
C CYVN(I,J)=RP*B22P

C
C COMPUTE DERIVATIVE VALUES FOR SOURCE TERMS.
C
C DDX=RP*(B11P*(UEE-UWV)+B21P*(UNN-USS))/VOLP
C DDX=RP*(B11P*(VEE-VWV)+B21P*(VNN-VSS))/VOLP
C DDX=RP*(B11P*(AKEE-AKVM)+B21P*(AKNN-AKSS))/VOLP
C DVIDX=RP*(B11P*(VIEE-VIWM)+B21P*(VINN-VISS))/VOLP

C
C DDY=RP*(B12P*(UEE-UWV)+B22P*(UNN-USS))/VOLP
C DDY=RP*(B12P*(VEE-VWV)+B22P*(VNN-VSS))/VOLP
C DDY=RP*(B12P*(AKEE-AKVM)+B22P*(AKNN-AKSS))/VOLP
C DVIDY=RP*(B12P*(VIEE-VIWM)+B22P*(VINN-VISS))/VOLP

C
C DUNDEE=0.25*(UN(I+1,J+1)+UN(I,J+1)
1 -UN(I+1,J-1)-UN(I,J-1))
C DVNDEE=0.25*(VN(I+1,J+1)+VN(I,J+1)
1 -VN(I+1,J-1)-VN(I,J-1))
C DUNDEN=0.25*(UN(I-1,J+1)+UN(I,J+1)
1 -UN(I-1,J-1)-UN(I,J-1))
C DVNDEEN=0.25*(VN(I-1,J+1)+VN(I,J+1)
1 -VN(I-1,J-1)-VN(I,J-1))
C DUNDKN=0.25*(UN(I+1,J+1)+UN(I+1,J)
1 -UN(I-1,J+1)-UN(I-1,J))
C DVNDKN=0.25*(VN(I+1,J+1)+VN(I+1,J)
1 -VN(I-1,J+1)-VN(I-1,J))
C DUNDKS=0.25*(UN(I+1,J-1)+UN(I+1,J)
1 -UN(I-1,J-1)-UN(I-1,J))
C DVNDKS=0.25*(VN(I+1,J-1)+VN(I+1,J)
1 -VN(I-1,J-1)-VN(I-1,J))

C
C SOURCE FUNCTION BY NON-ORTHOGONALITY.
C
C SBUNE=VIEE*D12E/VOLE*DUNDEE
C SBVNE=VIEE*D12E/VOLE*DVNDEE
C SBUNW=VINW*D12W/VOLW*DUNDEN
C SBVNW=VINW*D12W/VOLW*DVNDEW
C SBUNN=VINN*D21N/VOLN*DUNDKN
C SBVNN=VINN*D21N/VOLN*DVNDKN

C
C IF(J.EQ.2) THEN
C SBUNS=0.DO
C SBVNS=0.DO

```

20  
1  
4

```

ELSE
SBUNS=VISS*D21S/VOLS*DUNDKS
SBVNS=VISS*D21S/VOLS*DVNDKS
END IF

C
C SBUN=SBUNE-SBUNW+SBUNN-SBUNS
C SBVN=SBVNE-SBVNW+SBVNN-SBVNS

C
C COMPUTE SOURCE FUNCTIONS
C
C SORCUN=DVIDX*DDUX+DVIDY*DDY-2./3.*DKDX
C SORCVN=DVIDX*DUDY+DVIDY*D'DY-2./3.*DKDY

C
C SCUN(I,J)=CPO*UN(I,J)+SORCUN*VOLP+SBUN
C SCVN(I,J)=CPO*VN(I,J)+SORCVN*VOLP+SBVN

C
C IF(IFLOW.EQ.1) THEN
C SCUN(I,J)=CPO*UN(I,J)+SBUN
C SCVN(I,J)=CPO*VN(I,J)+SBVN
END IF

C
C SPUN(I,J)=CP
C SPVN(I,J)=CP

C
C IF(ICORD.EQ.2) THEN
C SPVN(I,J)=SPVN(I,J)
1 -0.5*(VIS(I,J)+VIS(I,J+1))*VOLP/RP/RP
END IF

C
C 380 CONTINUE
C
C *****
C CONSIDER THE BOUNDARY CONDITIONS
C (WALL FUNCTION AND SYMMETRY CONDITIONS)
C *****
C
C CDTERM=CMU*0.25

C
C ALONG THE BOTTOM WALL- BY WALL FUNCTION.
C *****
C
C IF(I.GT.M1.AND.I.LT.M2) THEN
C
C DIN=(XN(I,2)-XN(I,1))*2+(YN(I,2)-YN(I,1))*2
C DIW=(XC(I,1)-XC(I-1,1))*2+(YC(I,1)-YC(I-1,1))*2
C DISN=DSQRT(DIN)
C DISW=DSQRT(DIW)
C AN1=-/YC(I,1)-YC(I-1,1))/DISW
C AN2=(XC(I,1)-XC(I-1,1))/DISW

C
C AREAW=DISW*0.5*(R(I,1)+R(I-1,1))
C YW=DISN*AN2
C SORTK=DSQRT(0.5*(AKE(I,2)+AKE(I,3)))
C DENUE=0.5D0*(DEN(I,2)+DEN(I,3))
C YPLUSA=CDTERM*SORTK*YW/VISCOS
C
C IF(YPLUSA.LE.11.63D0) THEN
C TRULT=VISCOS/YW
C ELSE
C TRULT=DENUE*CDTERM*SORTK+CAPPA/DLOG(ELOG*YPLUSA)

```

```

END IF
C
C LAMINAR FLOW CALCULATION.
C
C IF(IFLOW.EQ.1) TMULT=VISCOS/YW
C
C SPUN(I,2)=SPUN(I,2)-TMULT*AREAH*AN2*AN2
C SPVN(I,2)=SPVN(I,2)-TMULT*AREAH*AN1*AN1
C
C SCUN(I,2)=SCUN(I,2)+TMULT*AREAH*AN1*AN2*VN(I,2)
C SCVN(I,2)=SCVN(I,2)+TMULT*AREAH*AN1*AN2*UN(I,2)
C
C ASN(I,2)=0.DO
C
C ALONG THE BOTTOM WALL AND SYMMETRY LINE.
C *****
C ELSE
C ASN(I,2)=0.DO
C
C END IF
C
C OUTLET CONDITION
C *****
C IF(I.EQ.IMAX) THEN
C DO 381 J=2,JMAX
C 381 AEN(IMAX,J)=0.DO
C END IF
C
C TOP FREE STREAM BOUNDARY.
C *****
C UN(I,JPI)=1.DO
C
C *****
C FINAL COEFF ASSEMBLY AND RESIDUAL SOURCE FOR UN
C *****
C DO 383 J=2,JMAX
C
C APUN(I,J)=ANN(I,J)+ASN(I,J)+AEN(I,J)+AMN(I,J)-SPUN(I,J)
C
C RESOR=ANN(I,J)*UN(I,J)+ASN(I,J)+ASN(I,J)*UN(I,J-1)
C 1 +AEN(I,J)*UN(I+1,J)+AMN(I,J)*UN(I,J-1)
C 2 -APUN(I,J)*UN(I,J)+SCUN(I,J)
C 3 +CXUN(I,J)*0.25*(PR(I-1,J+1)+PR(I-1,J)
C 4 -PR(I+1,J+1)-PR(I+1,J))
C 5 +CYUN(I,J)*(PR(I,J)-PR(I,J+1))
C
C VOLP=0.5*(VOL(I,J)+VOL(I,J+1))
C
C SORVOL=GREAT*VOLP
C IF(-SPVN(I,J).GT.0.5*SORVOL) RESOR=RESOR/SORVOL
C RESVN=RESVN+DABS(RESOR)
C
C UNDER-RELAXATION
C
C SCVN(I,J)=SCVN(I,J)
C 1 +(1.00-UREFV)/URFV*APVN(I,J)*VN(I,J)
C
C 384 CONTINUE
C
C *****
C CALCULATE THE COEFFICIENTS AND SOURCE TERMS FOR
C EAST CELL SURFACE VELOCITIES.
C *****
C DO 280 J=2,JMAX
C
C COMPUTE GEOMETRIC CONSTANTS
C
C FOR EAST CONTROL VOLUME SURFACE
C
C RE=0.25*(R(I+1,J)+R(I+1,J-1)+R(I,J-1)+R(I,J))
C B1E=VN(I+1,J)-VN(I+1,J-1)
C B12E=-(XN(I+1,J)-XN(I+1,J-1))
C B2E=-(YE(I+1,J)-YE(I,J))
C B22E=XE(I+1,J)-XE(I,J)
C D1E=RE*RE*(B1E*B1E+B12E*B12E)
C D12E=RE*RE*(B1E*B2E+B12E*B22E)
C VOLE=VOL(I+1,J)
C
C FOR WEST CONTROL VOLUME SURFACE
C
C RW=0.25*(R(I,J)+R(I,J-1)+R(I-1,J-1)+R(I-1,J))
C B1W=VN(I,J)-VN(I,J-1)

```

```

      B12M=-(XN(I,J)-XN(I,J-1))
      B21M=-(VE(I,J)-VE(I-1,J))
      B22M=XE(I,J)-XE(I-1,J)
      D11M=RM*RM*(B11M*B11M+B12M*B12M)
      D12M=RM*RM*(B11M*B21M+B12M*B22M)
      VOLM=VOL(I,J)
      C
      C
      C
      FOR NORTH CONTROL VOLUME SURFACE
      C
      C
      C
      RN=R(I,J)
      B11N=XE(I,J+1)-XE(I,J)
      B12N=-(XE(I,J+1)-XE(I,J))
      B21N=-(YN(I+1,J)-YN(I,J))
      B22N=XN(I+1,J)-XN(I,J)
      D21N=RN*RN*(B11N*B21N+B12N*B22N)
      D22N=RN*RN*(B21N*B21N+B22N*B22N)
      VOLN=0.25*(VOL(I,J)+VOL(I,J+1)+VOL(I+1,J+1)+VOL(I+1,J))
      C
      C
      C
      FOR SOUTH CONTROL VOLUME SURFACE
      C
      C
      C
      RS=R(I,J-1)
      B11S=XE(I,J)-XE(I,J-1)
      B12S=-(XE(I,J)-XE(I,J-1))
      B21S=-(YN(I+1,J-1)-YN(I,J-1))
      B22S=XN(I+1,J-1)-XN(I,J-1)
      D21S=RS*RS*(B11S*B21S+B12S*B22S)
      D22S=RS*RS*(B21S*B21S+B22S*B22S)
      VOLS=0.25*(VOL(I,J-1)+VOL(I,J)+VOL(I+1,J)+VOL(I+1,J-1))
      C
      C
      C
      FOR CENTER POINT
      C
      C
      C
      RP=0.5*(R(I,J)+R(I,J-1))
      B11P=YC(I,J)-YC(I,J-1)
      B12P=-(XC(I,J)-XC(I,J-1))
      B21P=-(YP(I+1,J)-YP(I,J))
      B22P=XP(I+1,J)-XP(I,J)
      VOLP=0.5*(VOL(I,J)+VOL(I+1,J))
      C
      C
      C
      COMPUTE CELL SURFACE VALUES
      C
      C
      UNN=0.5*(UE(I,J)+UE(I,J+1))
      VNN=0.5*(VN(I,J)+VN(I+1,J))
      USS=0.5*(UE(I,J)+UE(I,J-1))
      VSS=0.5*(VN(I,J-1)+VN(I+1,J-1))
      UEE=0.5*(UE(I,J)+UE(I+1,J))
      VEE=0.5*(VN(I+1,J)+VN(I+1,J-1))
      UVM=0.5*(UE(I,J)+UE(I-1,J))
      VVM=0.5*(VN(I,J)+VN(I,J-1))
      C
      AKNN=0.25*(AKE(I,J)+AKE(I,J+1)+AKE(I+1,J+1)+AKE(I+1,J))
      AKSS=0.25*(AKE(I,J-1)+AKE(I,J)+AKE(I+1,J)+AKE(I+1,J-1))
      AKEE=AKE(I+1,J)
      AKWM=AKE(I,J)
      C
      VMM=0.2500*(VIS(I,J)+VIS(I,J+1)+VIS(I+1,J)+VIS(I+1,J+1))
      VSS=0.2500*(VIS(I,J)+VIS(I,J-1)+VIS(I+1,J)+VIS(I+1,J-1))
      VEE=VIS(I+1,J)
      VVM=VIS(I,J)
      C
      DENN=0.25*(DEN(I,J)+DEN(I,J+1)+DEN(I+1,J)+DEN(I+1,J+1))
      DUEE=0.25*(UE(I,J)+UE(I,J+1)+UE(I+1,J+1)+UE(I+1,J))
      DENM=0.25*(UUE(I,J)+UUE(I+1,J))
      DEVM=0.25*(UUE(I,J)+UUE(I-1,J))
      C
      C
      C
      COMPUTE CONVECTION MASS FLUX.
      C
      C
      C
      FN=0.5*(VWN(I,J)+VWN(I+1,J))
      FS=0.5*(VWN(I,J-1)+VWN(I+1,J-1))
      FE=0.5*(UUE(I,J)+UUE(I+1,J))
      FW=0.5*(UUE(I,J)+UUE(I-1,J))
      C
      C
      C
      COMPUTE DIFFUSION COEFFICIENTS
      C
      C
      C
      DN=VINN*D22N/VOLN
      DS=VISS*D22S/VOLS
      DE=VIEE*D11E/VOLE
      DW=VIWM*D11W/VOLW
      C
      C
      C
      COMPUTE COEFFICIENTS OF SOURCE TERM
      C
      C
      C
      SMP=FN-FS+FE-FW
      CP=DMAX1(0.00,SMP)
      CP0=CP
      C
      C
      C
      ASSEMBLE MAIN COEFFICIENTS
      C
      C
      C
      ANE(I,J)=DMAX1(DABS(0.500*FN),DN)-0.500*FN
      ASE(I,J)=DMAX1(DABS(0.500*FS),DS)+0.500*FS
      AEE(I,J)=DMAX1(DABS(0.500*FE),DE)-0.500*FE
      AWE(I,J)=DMAX1(DABS(0.500*FW),DW)+0.500*FW
      C
      C
      C
      COEFFICIENTS OF PRESSURE TERM IN MOMENTUM EQUATION
      C
      C
      C
      CXUE(I,J)=RP*B11P
      CYUE(I,J)=RP*B21P
      CXVE(I,J)=RP*B12P
      CYVE(I,J)=RP*B22P
      C
      C
      C
      COMPUTE DERIVATIVE VALUES FOR SOURCE TERMS.
      C
      C
      C
      DUDX=RP*(B11P*(UEE-VMM)+B21P*(UNN-USS))/VOLP
      DVDX=RP*(B11P*(VEE-VMM)+B21P*(VNN-VSS))/VOLP
      DKDX=RP*(B11P*(AKEE-AKWM)+B21P*(AKNN-AKSS))/VOLP
      DIVDX=RP*(B11P*(VIEE-VIWM)+B21P*(VINN-VISS))/VOLP
      C
      DUDY=RP*(B12P*(UEE-VMM)+B22P*(UNN-USS))/VOLP
      DVDY=RP*(B12P*(VEE-VMM)+B22P*(VNN-VSS))/VOLP
      DKDY=RP*(B12P*(AKEE-AKWM)+B22P*(AKNN-AKSS))/VOLP
      DIVDY=RP*(B12P*(VIEE-VIWM)+B22P*(VINN-VISS))/VOLP
      C
      DUEDES=0.25*(UE(I+1,J+1)+UE(I,J+1)
      -UE(I+1,J-1)-UE(I,J-1))
      DVEDES=0.25*(VE(I+1,J+1)+VE(I,J+1)
      -VE(I+1,J-1)-VE(I,J-1))
      DUEDEM=0.25*(UE(I-1,J+1)+UE(I,J+1)
      -UE(I-1,J-1)-UE(I,J-1))
      DVEDEM=0.25*(VE(I-1,J+1)+VE(I,J+1)
      -VE(I-1,J-1)-VE(I,J-1))
      DUEDKN=0.25*(UE(I+1,J+1)+UE(I+1,J)
      -UE(I-1,J+1)-UE(I-1,J))
      C

```





```

C *****
C FINAL COEFF ASSEMBLY AND RESIDUAL SOURCE FOR UE
C *****
C
C DO 283 J=2,JMAX
C
C APUE(I,J)=AWE(I,J)+ASE(I,J)+AEE(I,J)+AWE(I,J)-SPUE(I,J)
C
C RESOR=AWE(I,J)+AEE(I,J)+ASE(I,J)+AWE(I,J)+AWE(I,J-1)
C 1 +AEE(I,J)+AEE(I+1,J)+AWE(I,J)+AWE(I,J-1)
C 2 -APUE(I,J)+AWE(I,J)+SCUE(I,J)
C 3 +CXUE(I,J)+PR(I,J)-PR(I+1,J)
C 4 +CYUE(I,J)+0.25*(PR(I,J-1)+PR(I+1,J)-1)
C 5 -PR(I,J+1)-PR(I+1,J+1))
C
C VOLP=0.5*(VOL(I,J)+VOL(I+1,J))
C
C SORVOL=GREAT*VOLP
C IF(-SPUE(I,J).GT.0.5*SORVOL) RESOR=RESOR/SORVOL
C RESUE=RESUE+DABS(RESOR)
C
C UNDER-RELAXATION
C
C SCUE(I,J)=SCUE(I,J)
C 1 +(1.DO-URFU)/URFU*APUE(I,J)+UE(I,J)
C
C 283 CONTINUE
C
C *****
C FINAL COEFF ASSEMBLY AND RESIDUAL SOURCE FOR VE
C *****
C
C DO 284 J=2,JMAX
C
C APVE(I,J)=AWE(I,J)+ASE(I,J)+AEE(I,J)+AWE(I,J)-SPVE(I,J)
C
C RESOR=AWE(I,J)+AEE(I,J)+ASE(I,J)+AWE(I,J)+AWE(I,J-1)
C 1 +AEE(I,J)+AEE(I+1,J)+AWE(I,J)+AWE(I,J-1)
C 2 -APVE(I,J)+AWE(I,J)+SCVE(I,J)
C 3 +CXVE(I,J)+PR(I,J)-PR(I+1,J)
C 4 +CYVE(I,J)+0.25*(PR(I,J-1)+PR(I+1,J)-1)
C 5 -PR(I,J+1)-PR(I+1,J+1))
C
C VOLP=0.5*(VOL(I,J)+VOL(I+1,J))
C
C SORVOL=GREAT*VOLP
C IF(-SPVE(I,J).GT.0.5*SORVOL) RESOR=RESOR/SORVOL
C RESVE=RESVE+DABS(RESOR)
C
C UNDER-RELAXATION
C
C SCVE(I,J)=SCVE(I,J)
C 1 +(1.DO-URFV)/URFV*APVE(I,J)+VE(I,J)
C
C 284 CONTINUE
C
C *****
C SOLUTION OF DIFFERENCE EQUATIONS FOR UN.
C *****

```

```

C
C DO 385 ITUV=1,NSWPU
C DO 386 J=2,JMAX
C AA(J)=-ASN(I,J)
C BB(J)=APUN(I,J)/URFU
C CC(J)=-ANN(I,J)
C 386 DD(J)=AEN(I,J)+UN(I+1,J)+AWN(I,J)+UN(I-1,J)+SCUN(I,J)
C 3 -PR(I-1,J+1)+PR(I-1,J)
C 4 +CXUN(I,J)+0.25*(PR(I,J)-PR(I+1,J))
C 5 +CYUN(I,J)*(PR(I,J)-PR(I,J+1))
C
C DD(2)=DD(2)-AA(2)*UN(I,1)
C DD(JMAX)=DD(JMAX)-CC(JMAX)*UN(I,JMAX+1)
C
C CALL TRIDAG(2,JMAX,AA,BB,CC,DD,T)
C
C DO 387 J=2,JMAX
C UN(I,J)=T(J)
C
C IF(I.LE.M1.OR.I.GE.M2) THEN
C UN(I,1)=(YN3SQ*UN(I,2)-YN2SQ*UN(I,3))
C 1 / (YN3SQ-YN2SQ)
C ELSE
C UN(I,1)=0.
C END IF
C
C 385 CONTINUE
C
C *****
C SOLUTION OF DIFFERENCE EQUATIONS FOR VN.
C *****
C
C DO 388 ITUV=1,NSHPV
C DO 389 J=2,JMAX
C AA(J)=-ASN(I,J)
C BB(J)=APVN(I,J)/URFV
C CC(J)=-ANN(I,J)
C 389 DD(J)=AEN(I,J)+VN(I+1,J)+AWN(I,J)+VN(I-1,J)+SCVN(I,J)
C 3 +CXVN(I,J)+0.25*(PR(I-1,J+1)+PR(I-1,J)
C 4 -PR(I+1,J+1)-PR(I+1,J))
C 5 +CYVN(I,J)*(PR(I,J)-PR(I,J+1))
C
C DD(2)=DD(2)-AA(2)*VN(I,1)
C DD(JMAX)=DD(JMAX)-CC(JMAX)*VN(I,JMAX+1)
C
C CALL TRIDAG(2,JMAX,AA,BB,CC,DD,T)
C
C DO 390 J=2,JMAX
C VN(I,J)=T(J)
C
C VN(I,JPL)=VN(I,JMAX)*(R(I,JMAX)+R(I-1,JMAX))
C 1 / (R(I,JMAX+1)+R(I-1,JMAX+1))
C VN(I,1)=0.
C
C 388 CONTINUE
C
C *****
C SOLUTION OF DIFFERENCE EQUATIONS FOR UE.
C *****

```

```

C C
C *****
C DO 285 ITUV=1, NSMPV
C DO 286 J=2, JMAX
C AA(J)=ASE(I,J)
C BB(J)=APUE(I,J)/URFU
C CC(J)=-ANE(I,J)
C
C IF(J.EQ.2) THEN
C PRVUE=-0.25*(PR(I,2)+PR(I,3)+PR(I+1,2)+PR(I+1,3))
C 1 +0.5*(PR(I,1)+PR(I+1,1))
C ELSE
C PRVUE=0.25*(PR(I,J-1)+PR(I+1,J-1)
C -PR(I,J+1)-PR(I+1,J+1))
C 3
C END IF
C
C 286 DD(J)=AEE(I,J)*UE(I+1,J)+ANE(I,J)*UE(I-1,J)+SCUE(I,J)
C 1 +CXUE(I,J)*(PR(I,J)-PR(I+1,J))
C 2 +CYUE(I,J)*PRVUE
C
C DD(2)=DD(2)-AA(2)*UE(I,1)
C DD(JMAX)=DD(JMAX)-CC(JMAX)*UE(I,JMAX+1)
C
C CALL TRIDAG(2,JMAX,AA,BB,CC,DD,T)
C
C DO 287 J=2,JMAX
C 287 UE(I,J)=T(J)
C
C IF(I.LT.M1.OR.I.GE.M2) THEN
C UE(I,1)=(YESSQ*UE(I,2)-YE2SQ*UE(I,3))
C 1 /{(YESSQ-YE2SQ)}
C ELSE
C UE(I,1)=0.
C END IF
C
C 285 CONTINUE
C
C *****
C SOLUTION OF DIFFERENCE EQUATIONS FOR VE.
C *****
C
C DO 288 ITUV=1, NSMPV
C DO 289 J=2, JMAX
C AA(J)=ASE(I,J)
C BB(J)=APVE(I,J)/URFV
C CC(J)=-ANE(I,J)
C
C IF(J.EQ.2) THEN
C PRVVE=-0.25*(PR(I,2)+PR(I,3)+PR(I+1,2)+PR(I+1,3))
C 1 +0.5*(PR(I,1)+PR(I+1,1))
C ELSE
C PRVVE=0.25*(PR(I,J-1)+PR(I+1,J-1)
C -PR(I,J+1)-PR(I+1,J+1))
C 3
C END IF
C
C 289 DD(J)=AEE(I,J)*VE(I+1,J)+AVE(I,J)*VE(I-1,J)+SCVE(I,J)
C 1 +CXVE(I,J)*(PR(I,J)-PR(I+1,J))
C 2 +CYVE(I,J)*PRVVE
C
C *****
C DD(2)=DD(2)-AA(2)*VE(I,1)
C DD(JMAX)=DD(JMAX)-CC(JMAX)*VE(I,JMAX+1)
C
C CALL TRIDAG(2,JMAX,AA,BB,CC,DD,T)
C
C DO 290 J=2,JMAX
C 290 VE(I,J)=T(J)
C
C VE(I,JP1)=0.25*(VN(I,JP1)+VN(I,JMAX)+VN(I+1,JP1)+VN(I+1,JMAX))
C VE(I,1)=0.
C
C 288 CONTINUE
C
C *****
C CALCULATE THE MASS SOURCE FOR PRESSURE CORRECTION
C EQUATIONS
C *****
C
C DO 480 J=2,JMAX
C
C COMPUTE GEOMETRIC CONSTANTS
C
C FOR EAST CONTROL VOLUME SURFACE
C
C RE=0.5*(R(I,J)+R(I,J-1))
C B11E=YC(I,J)-YC(I,J-1)
C B12E=-(XC(I,J)-XC(I,J-1))
C
C FOR WEST CONTROL VOLUME SURFACE
C
C RW=0.5*(R(I-1,J)+R(I-1,J-1))
C B11W=YC(I-1,J)-YC(I-1,J-1)
C B12W=-(XC(I-1,J)-XC(I-1,J-1))
C
C FOR NORTH CONTROL VOLUME SURFACE
C
C RN=0.5*(R(I-1,J)+R(I,J))
C B21N=-(YC(I,J)-YC(I-1,J))
C B22N=XC(I,J)-XC(I-1,J)
C
C FOR SOUTH CONTROL VOLUME SURFACE
C
C RS=0.5*(R(I-1,J-1)+R(I,J-1))
C B21S=-(YC(I,J-1)-YC(I-1,J-1))
C B22S=XC(I,J-1)-XC(I-1,J-1)
C
C COMPUTE FACE DENSITIES.
C
C DENN=0.5D0*(DEN(I,J)+DEN(I,J+1))
C DESS=0.5D0*(DEN(I,J)+DEN(I,J-1))
C DEEE=0.5D0*(DEN(I,J)+DEN(I+1,J))
C DENW=0.5D0*(DEN(I,J)+DEN(I-1,J))
C
C COMPUTE CONVECTION MASS FLUX.
C
C FN=DENN*RN*(B21N*UN(I,J)+B22N*VN(I,J))

```

```

C C
C C C CALCULATE THE PRESSURE-CORRECTION FIELD
C C *****
C C ITPP=15
C C
C C DO 482 J=1,JP1
C C PP(I-1,J)=0.
C C 482 PP(I+1,J)=0.
C C
C C DO 483 ITER=1,ITPP
C C CALL PRESU(PP,SPP,I,0)
C C 483 CONTINUE
C C
C C *****
C C CORRECT THE VELOCITY FIELD.
C C *****
C C CORRECT CONTRAVARIANT VELOCITIES.
C C
C C DO 170 J=2,JMAX
C C VVN(I,J)=VVN(I,J)+BCV(I,J)*(PP(I,J)-PP(I,J+1))
C C VVN(I,1)=0.DO
C C VVN(I,JP1)=VVN(I,JMAX)
C C
C C DO 171 J=2,JMAX
C C UUE(I,J)=UUE(I,J)+BCU(I,J)*(PP(I,J)-PP(I+1,J))
C C UUE(I,JP1)=0.5D0*(DEN(I,JP1)+DEN(I+1,JP1))
C C 1 *0.5D0*(R(I,JMAX)+R(I,JP1))
C C 2 *(YC(I,JP1)-YC(I,JMAX))*UUE(I,JP1)
C C UUE(I,1)=0.DO
C C
C C CALCULATE THE OTHER CONTRAVARIANT VELOCITIES
C C BY INTERPOLATION.
C C
C C DO 172 J=2,JMAX
C C VVE(I,J)=0.25*(VVN(I,J)+VVN(I,J-1))+VVN(I+1,J)+VVN(I+1,J-1))
C C UUN(I,J)=0.25*(UUE(I,J)+UUE(I-1,J))+UUE(I,J+1)+UUE(I-1,J+1))
C C
C C CALCULATE CORRECTED CARTESIAN VELOCITY COMPONENTS
C C USING THE CORRECTED CONTRAVELOCITIES.
C C
C C FOR NORTH CONTROL VOLUME SURFACE
C C VELOCITY COMPONENTS, UN AND VN.
C C
C C DO 174 J=2,JMAX
C C DENN=0.5*(DEN(I,J)+DEN(I,J+1))
C C RN=0.5*(R(I,J)+R(I-1,J))
C C B11N=YP(I,J+1)-YP(I,J)
C C B12N=(XP(I,J+1)-XP(I,J))
C C B21N=(YC(I,J)-YC(I-1,J))
C C B22N=XC(I,J)-XC(I-1,J)
C C DIVN=DENN*RN*(B11N*B22N-B12N*B21N)
C C UN(I,J)=(B22N*UUN(I,J)-B12N*VUN(I,J))/DIVN
C C 174 VN(I,J)=(B21N*UUN(I,J)-B11N*VUN(I,J))/DIVN
C C
C C
C C *****
C C
C C FS=DESS*RS*(B21S*UN(I,J-1)+B22S*VN(I,J-1))
C C FE=DEEE*RE*(B11E*UE(I,J)+B12E*VE(I,J))
C C FM=DEMN*RN*(B11M*UE(I-1,J)+B12M*VE(I-1,J))
C C
C C SMP=FN-FS+FE-FM
C C
C C SPP(I,J)=SMP
C C
C C COMPUTE CONTRAVARIANT VELOCITY COMPONENTS
C C
C C UUE(I,J)=FE
C C VVN(I,J)=FM
C C
C C COMPUTE SUM OF ABSOLUTE MASS SOURCE
C C
C C RESNM=RESNM+DABS(SMP)
C C
C C COMPUTE COEFFICIENTS OF PRESSURE
C C AND PRESSURE CORRECTION.
C C
C C QXUE(I,J)=URFU*RE*B11E/AFUE(I,J)
C C QXVE(I,J)=URFV*RE*B12E/AFVE(I,J)
C C QYUN(I,J)=URFU*RN*B21N/AFUN(I,J)
C C QYVN(I,J)=URFV*RN*B22N/AFVN(I,J)
C C
C C BCU(I,J)=DEEE*RE*(B11E*QXUE(I,J)+B12E*QXVE(I,J))
C C BCV(I,J)=DEMN*RN*(B21N*QYUN(I,J)+B22N*QYVN(I,J))
C C
C C 480 CONTINUE
C C *****
C C CONSIDER THE BOUNDARY CONDITIONS FOR
C C THE PRESSURE AND PRESSURE CORRECTION EQUATION.
C C *****
C C ALONG THE BOTTOM WALL AND SYMMETRY LINE.
C C *****
C C
C C VVN(I,1)=0.DO
C C UUE(I,1)=0.DO
C C BCV(I,1)=0.DO
C C
C C TOP FREE STREAM BOUNDARY
C C *****
C C
C C VVN(I,JP1)=VVN(I,JMAX)
C C UUE(I,JP1)=0.5D0*(DEN(I,JP1)+DEN(I+1,JP1))
C C 1 *0.5D0*(R(I,JMAX)+R(I,JP1))
C C 2 *(YC(I,JP1)-YC(I,JMAX))*UUE(I,JP1)
C C
C C ALONG THE INLET
C C *****
C C
C C IF(I,Eq.2) THEN
C C DO 481 J=2,JMAX
C C 481 BCU(I,J)=0.DO
C C END IF
C C *****
C C
C C *****
C C *****

```



```

1      +AEN(I,J)*VN(I+1,J)+AWN(I,J)*VN(I-1,J)
2      +SCVN(I,J)/(APVN(I,J)/URFV)
3      +CAVN(I,J)/(APVN(I,J)/URFV)
4      *0.25*(PR(I-1,J+1)+PR(I-1,J)
5      -PR(I+1,J+1)-PR(I+1,J))
C
C      VHN(I,1)=VN(I,1)
C      UHN(I,1)=UN(I,1)
C
C      IF(I.EQ.2) THEN
C      DO 492 J=2,JMAX
C      UHE(I,J)=UE(I,J)
C      VHE(I,J)=VE(I,J)
C      END IF
C      492
C
C      CALCULATE MASS-SOURCE OF PRESSURE EQUATION.
C
C      DO 493 J=2,JMAX
C
C      COMPUTE GEOMETRIC CONSTANTS
C
C      FOR EAST CONTROL VOLUME SURFACE
C
C      RE=0.5*(R(I,J)+R(I,J-1))
C      B1E=YC(I,J)-YC(I,J-1)
C      B2E=-(XC(I,J)-XC(I,J-1))
C      B21E=-(YP(I+1,J)-YP(I,J))
C      B22E=XP(I+1,J)-XP(I,J)
C      D11E=RE*RE*(B11E*B11E+B12E*B12E)
C      D12E=RE*RE*(B11E*B21E+B12E*B22E)
C      VOLM=0.5*(VOL(I,J)+VOL(I+1,J))
C
C      FOR WEST CONTROL VOLUME SURFACE
C
C      RW=0.5*(R(I-1,J)+R(I-1,J-1))
C      B1W=YC(I-1,J)-YC(I-1,J-1)
C      B2W=-(XC(I-1,J)-XC(I-1,J-1))
C      B21W=-(YP(I,J)-YP(I-1,J))
C      B22W=XP(I,J)-XP(I-1,J)
C      D11W=RW*RW*(B11W*B11W+B12W*B12W)
C      D12W=RW*RW*(B11W*B21W+B12W*B22W)
C      VOLM=0.5*(VOL(I,J)+VOL(I-1,J))
C
C      FOR NORTH CONTROL VOLUME SURFACE
C
C      RN=0.5*(R(I-1,J)+R(I,J))
C      B1N=YC(I-1,J)-YC(I,J)
C      B2N=-(XC(I-1,J)-XC(I,J))
C      B21N=-(YP(I,J)-YP(I-1,J))
C      B22N=XP(I,J)-XP(I-1,J)
C      D11N=RN*RN*(B11N*B11N+B12N*B12N)
C      D12N=RN*RN*(B11N*B21N+B12N*B22N)
C      VOLM=0.5*(VOL(I,J)+VOL(I,J+1))
C
C      FOR SOUTH CONTROL VOLUME SURFACE
C
C      RS=0.5*(R(I-1,J-1)+R(I,J-1))
C      B1S=YP(I,J)-YP(I,J-1)
C      B2S=-(XP(I,J)-XP(I,J-1))
C      B21S=-(YC(I,J-1)-YC(I-1,J-1))
C      B22S=XC(I,J-1)-XC(I-1,J-1)
C
C      COMPUTE FACE DENSITIES.
C
C      DENN=0.5D0*(DEN(I,J)+DEN(I,J+1))
C      DESS=0.5D0*(DEN(I,J)+DEN(I,J-1))
C      DEEP=0.5D0*(DEN(I,J)+DEN(I+1,J))
C      DENW=0.5D0*(DEN(I,J)+DEN(I-1,J))
C
C      COMPUTE CONVECTION MASS FLUX.
C
C      FN=DENN*RN*(B21N*UHN(I,J)+B22N*VHN(I,J))
C      FS=DESS*RS*(B21S*UHN(I,J-1)+B22S*VHN(I,J-1))
C      FE=DEEP*RE*(B11E*UHE(I,J)+B12E*VHE(I,J))
C      FW=DEMW*RW*(B11W*UHE(I-1,J)+B12W*VHE(I-1,J))
C
C      SMP=FN+FS+FE+FW
C      SPR(I,J)=SMP
C      493 CONTINUE
C
C      IF(IFLOW.EQ.1) GO TO 688
C
C      *****
C      CALCULATE THE COEFFICIENTS AND SOURCE FUNCTIONS
C      FOR TURBULENCE KINETIC ENERGY
C      AND DISSIPATION RATE EQUATIONS.
C      *****
C
C      DO 580 J=2,JMAX
C
C      COMPUTE GEOMETRIC CONSTANTS
C
C      FOR EAST CONTROL VOLUME SURFACE
C
C      RE=0.5*(R(I,J)+R(I,J-1))
C      B1E=YC(I,J)-YC(I,J-1)
C      B2E=-(XC(I,J)-XC(I,J-1))
C      B21E=-(YP(I+1,J)-YP(I,J))
C      B22E=XP(I+1,J)-XP(I,J)
C      D11E=RE*RE*(B11E*B11E+B12E*B12E)
C      D12E=RE*RE*(B11E*B21E+B12E*B22E)
C      VOLM=0.5*(VOL(I,J)+VOL(I+1,J))
C
C      FOR WEST CONTROL VOLUME SURFACE
C
C      RW=0.5*(R(I-1,J)+R(I-1,J-1))
C      B1W=YC(I-1,J)-YC(I-1,J-1)
C      B2W=-(XC(I-1,J)-XC(I-1,J-1))
C      B21W=-(YP(I,J)-YP(I-1,J))
C      B22W=XP(I,J)-XP(I-1,J)
C      D11W=RW*RW*(B11W*B11W+B12W*B12W)
C      D12W=RW*RW*(B11W*B21W+B12W*B22W)
C      VOLM=0.5*(VOL(I,J)+VOL(I-1,J))
C
C      FOR NORTH CONTROL VOLUME SURFACE
C
C      RN=0.5*(R(I-1,J)+R(I,J))
C      B1N=YC(I-1,J)-YC(I,J)
C      B2N=-(XC(I-1,J)-XC(I,J))
C      B21N=-(YP(I,J)-YP(I-1,J))
C      B22N=XP(I,J)-XP(I-1,J)
C      D11N=RN*RN*(B11N*B11N+B12N*B12N)
C      D12N=RN*RN*(B11N*B21N+B12N*B22N)
C      VOLM=0.5*(VOL(I,J)+VOL(I,J+1))
C
C      FOR SOUTH CONTROL VOLUME SURFACE
C
C      RS=0.5*(R(I-1,J-1)+R(I,J-1))
C      B1S=YP(I,J)-YP(I,J-1)
C      B2S=-(XP(I,J)-XP(I,J-1))
C      B21S=-(YC(I,J-1)-YC(I-1,J-1))
C      B22S=XC(I,J-1)-XC(I-1,J-1)

```

```

D21S=RS*RS*(B11S*B21S+B12S*B22S)
D22S=RS*RS*(B21S*B21S+B22S*B22S)
VOLS=0.5*(VOL(I,J-1)+VOL(I,J))
FOR CENTER POINT
  RP=0.25*(R(I,J)+R(I-1,J)+R(I-1,J-1)+R(I,J-1))
  B1P=VN(I,J)-VN(I,J-1)
  B12P=-(XN(I,J)-XN(I,J-1))
  B21P=-(YF(I,J)-YF(I,J-1))
  B22P=XE(I,J)-XE(I-1,J)
  VOLP=VOL(I,J)
COMPUTE CELL SURFACE VALUES
  UNN=0.25*(UE(I-1,J)+UE(I-1,J+1)+UE(I,J)+UE(I,J+1))
  VNN=VN(I,J)
  USS=0.25*(UE(I-1,J-1)+UE(I-1,J)+UE(I,J-1)+UE(I,J))
  VSS=VN(I,J-1)
  UEE=UE(I,J)
  VEE=0.25*(VN(I,J)+VN(I,J-1)+VN(I+1,J)+VN(I+1,J-1))
  UMM=UE(I-1,J)
  VMM=0.25*(VN(I-1,J)+VN(I-1,J-1)+VN(I,J)+VN(I,J-1))
IF(J.EQ.2) THEN
  USS=0.5*(UE(I-1,1)+UE(I,1))
END IF
COMPUTE DERIVATIVE VALUES FOR SOURCE TERMS.
  DUDX=RP*(B11P*(UEE-UWM)+B21P*(UNN-USS))/VOLP
  DUDY=RP*(B12P*(VEE-UWM)+B22P*(VNN-USS))/VOLP
  DVDI=RP*(B12P*(VEE-UWM)+B22P*(VNN-USS))/VOLP
  DAKEDEE=0.25*(AKE(I+1,J+1)+AKE(I,J+1)
  1 -AKE(I+1,J-1)-AKE(I,J-1))
  DADSDEE=0.25*(ADS(I+1,J+1)+ADS(I,J+1)
  1 -ADS(I+1,J-1)-ADS(I,J-1))
  DAKEDEM=0.25*(AKE(I-1,J+1)+AKE(I,J+1)
  1 -AKE(I-1,J-1)-AKE(I,J-1))
  DADSDEM=0.25*(ADS(I-1,J+1)+ADS(I,J+1)
  1 -ADS(I-1,J-1)-ADS(I,J-1))
  DAKEDKN=0.25*(AKE(I+1,J+1)+AKE(I+1,J)
  1 -AKE(I-1,J+1)-AKE(I-1,J))
  DADSCKN=0.25*(ADS(I+1,J+1)+ADS(I+1,J)
  1 -ADS(I-1,J+1)-ADS(I-1,J))
  DAKEDKS=0.25*(AKE(I+1,J-1)+AKE(I+1,J)
  1 -AKE(I-1,J-1)-AKE(I-1,J))
  DADSCKS=0.25*(ADS(I+1,J-1)+ADS(I+1,J)
  1 -ADS(I-1,J-1)-ADS(I-1,J))
SOURCE FUNCTION BY NON-ORTHOGONALITY.
  SBAKEE=VIEE*D12E/VOLE*DAKEDEE
  SBADSE=VIEE*D12E/VOLE*DAKSDEE
  SBAKEM=VINW*D12W/VOLM*DAKEDEW
  SBADSW=VINW*D12W/VOLM*DAKSDEW
  SBAKEN=VINN*D21N/VOLN*DAKEDKN
  SBADSN=VINN*D21N/VOLN*DAKSDKN
IF(J.EQ.2) THEN
  SBAKES=0.D0
  SBADSS=0.D0

```

```

ELSE
SBAKES=VISS*0.215/VOLS*DAKEDKS
SBADSS=VISSD*0.215/VOLS*DADSDKS
END IF
C
SBAKE=SBAKE-SBAKEN+SBAKEN-SBAKES
SBADSS=SBADSE-SBADSN+SBADSN-SBADSS
C
IF (J.EQ.2) THEN
SBAKE=0.00
SBADSS=0.00
END IF
C
C
COMPUTE GENERATION TERM
C
C
GEN(I,J)=VIS(I,J)*(2.00*(DUDX*DUDX+DVDY*DVDY)
1 + (DUDY*DUDY)**2)
IF(ICORD.EQ.2) THEN
GEN(I,J)=GEN(I,J)+VIS(I,J)*0.5*(VN(I,J)+VN(I,J-1))**2/RP/RP
END IF
C
COMPUTE SOURCE FUNCTIONS
C
FOR KINETIC ENERGY EQUATION.
C
C
SCK(I,J)=CPO*AKE(I,J)+SBAKE
SUKO(I,J)=SCK(I,J)
SCK(I,J)=SCK(I,J)+GEN(I,J)*VOLP
SPK(I,J)=CP
SPKO(I,J)=SPK(I,J)
SPK(I,J)=SPK(I,J)
1 -DEN(I,J)+ADS(I,J)/AKE(I,J)*VOLP
C
FOR DISSIPATION RATE EQUATION.
C
C
SCD(I,J)=CPO*ADS(I,J)+SBADS
SUDD(I,J)=SCD(I,J)
SCD(I,J)=SCD(I,J)
1 +DEN(I,J)*C1*GEN(I,J)+ADS(I,J)*VOLP/AKE(I,J)
C
SPD(I,J)=CP
SPDO(I,J)=SPD(I,J)
SPD(I,J)=SPD(I,J)
1 -C2*DEN(I,J)+ADS(I,J)*VOLP/AKE(I,J)
C
580 CONTINUE
C
*****
C CONSIDER THE BOUNDARY CONDITIONS *****
(WALL, FUNCTION AND SYMMETRY CONDITIONS)
*****
C
CDTERM=CHU**0.2500
C
ALONG THE BOTTOM WALL - WALL FUNCTION
*****
C
IF (I.GE.N1.AND.I.LE.N2) THEN

```

```

C
RP=0.25*(R(I,2)+R(I-1,2))+R(I-1,1)+R(I,1)
B11P=VN(I,2)-VN(I,1)
B12P=(XN(I,2)-XN(I,1))/VN(I,1)
B21P=(YE(I,2)-YE(I-1,2))/VN(I,2)
B22P=(XE(I,2)-XE(I-1,2))/VN(I,2)
VOLP=VOL(I,2)
C
DIN=(XP(I,2)-XP(I,1))**2+(YP(I,2)-YP(I,1))**2
DIM=(XC(I,1)-XC(I-1,1))**2+(YC(I,1)-YC(I-1,1))**2
DISN=DSORT(DIN)
DISN=DSORT(DIM)
AN1=(YC(I,1)-YC(I-1,1))/DISW
AN2=(XC(I,1)-XC(I-1,1))/DISW
C
AREAW=DISW*0.5*(R(I,1)+R(I-1,1))
YW=DISN*AN2
C
UP=0.5*(UE(I,2)+UE(I-1,2))
VP=0.5*(VN(I,2)+VN(I-1,2))
UAVG=AN2*UP-AN1*VP
DENU=DEN(I,2)
SQRK=DSORT(AKE(I,2))
YPLUSN(I)=DENU*SQRK*CDTERM*YW/VISCOS
C
IF (YPLUSN(I).LE.11.63D0) THEN
TMULT=VISCOS/YW
TAUN(I)=TMULT*UAVG
DITERM=DEN(I,2)*(CMU**0.75D0)**SQRK*YPLUSN(I)/YW
ELSE
TMULT=DENU*CDTERM*SQRK*CAPPA/DLOG(ELOG*YPLUSN(I))
TAUN(I)=TMULT*UAVG
DITERM=DEN(I,2)*(CMU**0.75D0)*SQRK
1 *DLOG(ELOG*YPLUSN(I))/(CAPPA*YW)
END IF
C
UNN=0.25*(UE(I-1,2)+UE(I-1,3))+UE(I,2)+UE(I,3)
USS=0.5*(UE(I-1,1)+UE(I,1))
UEE=UE(I,2)
UWM=UE(I-1,2)
VNN=VN(I,2)
VSS=VN(I,1)
VEE=0.25*(VN(I,2)+VN(I+1,2)+VN(I,1)+VN(I+1,1))
VMM=0.25*(VN(I,2)+VN(I-1,2)+VN(I,1)+VN(I-1,1))
C
DUDX=RP*(B11P*(UEE-UWM)+B21P*(UNN-USS))/VOLP
DVDX=RP*(B11P*(VEE-VMM)+B21P*(VNN-VSS))/VOLP
DUY=RP*(B12P*(UEE-UWM)+B22P*(VNN-USS))/VOLP
DVDY=RP*(B12P*(VEE-VMM)+B22P*(VNN-VSS))/VOLP
C
GENCOU=TAUN(I)**2/(DENU*CDTERM*SQRK*CAPPA*YW)
GENRES=VIS(I,2)*(2.00*(DUDX*DUDX+DVDY*DVDY))
1 +GENRES
C
IF(ICORD.EQ.2) THEN
GEN(I,2)=GEN(I,2)+VIS(I,2)*0.5*(VN(I,2)+VN(I,1))**2/RP/RP
END IF
C

```



```

C C FOR KINETIC ENERGY EQUATION
C C SCK(I,2)=GEN(I,2)*VOLP-SUKD(I,2)
C C SPK(I,2)=-DITERM*VOLP+SPKD(I,2)
C C ASK(I,2)=0.DO
C C FOR DISSIPATION RATE EQUATION.
C C TERM=(CNU**0.75DO)/(CAPPA*YH)
C C SCD(I,2)=GREAT*TERM*AKE(I,2)**1.5
C C SPD(I,2)=-GREAT
C C ALONG THE SYMMETRY AXIS
C C *****
C C ELSE
C C RP=0.25*(R(I,2)+R(I-1,2))+R(I-1,1)+R(I,1)
C C B11P=VN(I,2)-VN(I,1)
C C B12P=-XN(I,2)-XN(I,1)
C C B21P=-{YE(I,2)-YE(I-1,2)}
C C B22P=XE(I,2)-XE(I-1,2)
C C VOLP=VOL(I,2)
C C UNN=0.25*(UE(I-1,2)+UE(I-1,3))+UE(I,2)+UE(I,3)
C C USS=0.5*(UE(I-1,1)+UE(I,1))
C C UEE=UE(I,2)
C C UMW=UE(I-1,2)
C C VNN=VN(I,2)
C C VSS=VN(I,1)
C C VEE=0.25*(VN(I,2)+VN(I+1,2)+VN(I,1)+VN(I+1,1))
C C VNM=0.25*(VN(I,2)+VN(I-1,2)+VN(I,1)+VN(I-1,1))
C C DVDX=RP*(B11P*(VEE-UWV)+B21P*(UNN-USS))/VOLP
C C DVDX=RP*(B11P*(VEE-UWV)+B21P*(VNN-VSS))/VOLP
C C DVDY=RP*(B12P*(VEE-UWV)+B22P*(UNN-USS))/VOLP
C C DVDY=RP*(B12P*(VEE-UWV)+B22P*(VNN-VSS))/VOLP
C C GEN(I,2)=VIS(I,2)*(2.DO*(DVDX*DUD.+DVY*DVDY)+(DVDX+DVY)**2)
C C IF(ICORD.EQ.2) THEN
C C GEN(I,2)=GEN(I,2)+VIS(I,2)*0.5*(VN(I,2)+VN(I,1))**2/RP/RP
C C END IF
C C SCK(I,2)=SUKD(I,2)+GEN(I,2)*VOLP
C C SCD(I,2)=SUDD(I,2)
C C 1 +DEN(I,2)*C1*GEN(I,2)*ADS(I,2)*VOLP/AKE(I,2)
C C ASK(I,2)=0.DO
C C ASD(I,2)=0.DO
C C END IF
C C ALONG THE OUTLET.
C C *****

```

```

C C IF(I.EQ.IMAX) THEN
C C DO 582 J=2,JMAX
C C AKE(IMAX,J)=0.DO
C C 582 AED(IMAX,J)=0.DO
C C END IF
C C *****
C C FINAL COEFF ASSEMBLY AND RESIDUAL SOURCE FOR AKE
C C *****
C C DO 583 J=2,JMAX
C C APK(I,J)=ANK(I,J)+ASK(I,J)+AEK(I,J)+AMK(I,J)+SPK(I,J)
C C RESOR=ANK(I,J)*AKE(I,J+1)+ASK(I,J)*AKE(I,J-1)
C C 1 +AEK(I,J)*AKE(I+1,J)+AMK(I,J)*AKE(I-1,J)
C C 2 -APK(I,J)*AKE(I,J)+SCK(I,J)
C C VOLP=VOL(I,J)
C C SORVOL=GREAT*VOLP
C C IF(-SPK(I,J).GT.0.5*SORVOL) RESOR=RESOR/SORVOL
C C RESAK=RESAK+DABS(RESOR)
C C UNDER-RELAXATION
C C APK(I,J)=APK(I,J)/URFK
C C SCK(I,J)=SCK(I,J)+(1.DO-URFK)*APK(I,J)*AKE(I,J)
C C 583 CONTINUE
C C *****
C C SOLUTION OF DIFFERENCE EQUATIONS FOR AKE
C C *****
C C DO 584 ITUV=1,NSWPK
C C DO 585 J=2,JMAX
C C AA(J)=-ASK(I,J)
C C BB(J)=APK(I,J)
C C CC(J)=-ANK(I,J)
C C 585 DD(J)=AEK(I,J)*AKE(I+1,J)+AMK(I,J)*AKE(I-1,J)
C C 1 +SCK(I,J)
C C DD(2)=DD(2)-AA(2)*AKE(I,1)
C C DD(JMAX)=DD(JMAX)-CC(JMAX)*AKE(I,JMAX+1)
C C CALL TRIDAG(2,JMAX,AA,BB,CC,DD,T)
C C DO 586 J=2,JMAX
C C 586 AKE(I,J)=T(J)
C C AKE(I,JMAX+1)=AKE(I,JMAX)
C C IF(I.LE.M1.OR.I.GE.M2) THEN
C C AKE(I,1)=(YP3SQ*AKE(I,2)-YP2SQ*AKE(I,3))
C C 1 ELSE
C C AKE(I,1)=0.

```

```

C      END IF
C      584 CONTINUE
C      DO 56 J=1,JP1
C      AKE(IPI,J)=AKE(IMAX,J)
C      ADS(IPI,J)=ADS(IMAX,J)
C      56 CONTINUE
C      END IF
C
C      688 CONTINUE
C
C      *****
C      CALCULATE SOME IMPORTANT VARIABLES.
C      *****
C      FOR LAMINAR FLOW CALCULATION.
C      IF(IFLOW.EQ.1.AND.I.GT.M1.AND.I.LT.M2) THEN
C      DIN=(XP(I,2)-XP(I,1))*2+(YP(I,2)-YP(I,1))*2
C      DIW=(XC(I,1)-XC(I-1,1))*2+(YC(I,1)-YC(I-1,1))*2
C      DISW=DSQRT(DIN)
C      AN1=(YC(I,1)-YC(I-1,1))/DISW
C      AN2=(XC(I,1)-XC(I-1,1))/DISW
C      YW=DISN*AN2
C      UP=0.5*(UE(I,2)+UE(I-1,2))
C      VP=0.5*(VN(I,2)+VN(I,1))
C      UAVG=AN2*UP-AN1*VP
C      TMULT=VISCOS/YW
C      TAUN(I)=TMULT*UAVG
C      END IF
C      CALCULATE FRICTION VELOCITY.
C      IF(I.GT.M1.AND.I.LT.M2) THEN
C      CFS(I)=DABS(TAUN(I))/(0.5D0*DENSIT*UIN*UIN)
C      UTAU(I)=DSQRT(0.5D0*CFS(I))
C      IF(TAUN(I).LE.0.D0) THEN
C      CFS(I)=-CFS(I)
C      UTAU(I)=-UTAU(I)
C      END IF
C      END IF
C      *****
C      PRINT SOME VARIABLES ON THE TERMINAL
C      AND RETURN SPACE MARCHING PROCESS.
C      *****
C      PRINT 9997, I,XE(I,1),UE(I,2),VN(I,2),AKE(I,2),ADS(I,2)
C      9997 FORMAT(5X,'I=',I3,2X,'X=',E11.4,2X,'UE2=',E11.4,2X,

```

```

C      END IF
C      584 CONTINUE
C      *****
C      FINAL COEFF ASSEMBLY AND RESIDUAL SOURCE FOR ADS
C      *****
C      DO 683 J=2,JMAX
C      APD(I,J)=AND(I,J)+ASD(I,J)+AED(I,J)+AMD(I,J)+SPD(I,J)
C      RESOR=AND(I,J)*ADS(I,J+1)+ASD(I,J)*ADS(I,J-1)
C      1 +AED(I,J)*ADS(I+1,J)+AMD(I,J)*ADS(I-1,J)
C      2 -APD(I,J)*ADS(I,J)+SCD(I,J)
C      VOLP=VOL(I,J)
C      SORVOL=GREAT*VOLP
C      IF(-SPD(I,J).GT.0.5*SORVOL) RESOR=RESOR/SORVOL
C      RESAD=RESAD*DABS(RESOR)
C      UNDER-RELAXATION
C      APD(I,J)=APD(I,J)/URFD
C      SCD(I,J)=SCD(I,J)+(1.D0-URFD)*APD(I,J)*ADS(I,J)
C      683 CONTINUE
C      *****
C      SOLUTION OF DIFFERENCE EQUATIONS FOR ADS
C      *****
C      DO 684 ITUV=1,NSWPD
C      DO 685 J=2,JMAX
C      AA(J)=-ASD(I,J)
C      BB(J)=APD(I,J)
C      CC(J)=-AND(I,J)
C      685 DD(J)=AED(I,J)*ADS(I+1,J)+AMD(I,J)*ADS(I-1,J)
C      1 +SCD(I,J)
C      DD(2)=DD(2)-AA(2)*ADS(I,1)
C      DD(JMAX)=DD(JMAX)-CC(JMAX)*ADS(I,JMAX+1)
C      CALL TRIDAG(2,JMAX,AA,BB,CC,DD,T)
C      DO 686 J=2,JMAX
C      686 ADS(I,J)=T(J)
C      ADS(I,JMAX+1)=ADS(I,JMAX)
C      IF(I.LE.M1.OR.I.GE.M2) THEN
C      ADS(I,1)=(YP2SQ*ADS(I,2)-YP2SQ*ADS(I,3))
C      1 /((YP2SQ-YP2SQ))
C      ELSE
C      ADS(I,1)=0.
C      END IF
C      684 CONTINUE

```

```

A-95
1 'VN2=' ,E11.4,2X,'AKE2=' ,E11.4,2X,'ADS2=' ,E11.4)
C
C 3000 CONTINUE
C *****
C PRINT GLOBAL ERRORS IN EACH ITERATION
C *****
C
C RESM=RESM/FLOWIN
C RESUE=RESUE/XMONIN
C RESVN=RESVN/XMONIN
C RESAK=RESAK/(0.500*FLOWIN*UMEAN*UMEAN)
C
C PRINT 701 ,NITER,RESM,RESUE,RESVN,RESAK
701 FORMAT(3X,'I=' ,I5,2X,'ERRM=' ,E11.4,2X,'ERRUE=' ,E11.4
1 ,2X,'ERRVN=' ,E11.4,2X,'ERRK=' ,E11.4)
C
C STORE THE ERRORS
C
C WRITE(9,7010) NITER,RESM,RESUE,RESVN,RESAK
7010 FORMAT(3X,I5,2X,E11.4,2X,E11.4,2X,E11.4,2X,E11.4)
C
C *****
C UPDATE THE PRESSURE FIELD BY GLOBAL ITERATIONS
C *****
C
C
C
C IF(NITER.LT.ITPR) GO TO 1234
C
C ITERP=10
C
C DO 1660 I=1,IP1
C DO 1660 J=1,JP1
1660 PR(I,J)=PR(I,J)
C
C DO 4001 ITERG=1,20
C
C DO 661 II=2,IMAX
C I=IMAX+2-II
C DO 660 ITER=1,ITERP
660 CALL PRESU(PR,SPR,I,1)
661 CONTINUE
C
C DO 662 I=1,IMAX
662 PR(I,1)=(YP3(I)*PR(I,2)-YP2(I)*PR(I,3))
1 / (YP3(I)-YP2(I))
C
C DO 954 J=1,JP1
954 PR(IP1,J)=PR(IMAX,J)
C
C 4001 CONTINUE
C
C DO 9999 I=1,IP1
C DO 9999 J=1,JP1
9999 PR(I,J)=PRO(I,J)+URFPR*(PR(I,J)-PRO(I,J))
C
C 1234 CONTINUE
C
C SELECT NUMBER OF ITERATION FOR OUTPUT
C
C ITPRINT=MAXIT-ITOUT
C IF(NITER.LE.ITPRINT) GO TO 6666
C
C *****
C OUTPUT FOR EACH ITERATION
C *****
C
C WRITE(8,4111) NITER
4111 FORMAT(/5X,'NO. OF ITERATION =' ,I5,
15X,'WALL OR CENTERLINE PRESSURE')
C WRITE(8,2009) (PR(I,1),I=2,IMAX)
C
C WRITE(8,4112)
4112 FORMAT(/5X,'FRICTION VELOCITY')
C WRITE(8,2009) (UTAU(I),I=MIP1,M2M1)
C
C WRITE(8,4113)
4113 FORMAT(/5X,'CENTERLINE VELOCITY')
C WRITE(8,2009) (UE(I,1),I=M2,IMAX)
C
C 6666 CONTINUE
C
C CHECK FOR OVERFLOW.
C
C DO 4999 I=1,IP1
C DO 4999 J=1,JP1
C IF(DABS(PR(I,J)).GT.10.DO) GO TO 5555
C IF(DABS(UE(I,J)).GT.10.DO) GO TO 5555
C IF(DABS(UN(I,J)).GT.10.DO) GO TO 5555
4999 CONTINUE
C
C IF(NITER.GE.MAXIT) GO TO 702
C
C GO TO 4000
C
C 702 CONTINUE
C
C *****
C OUTPUT AFTER CALCULATION
C *****
C
C DO 3333 I=1,IP1
C
C WRITE(8,12)NITER,I,XP(I,1),URFU
12 FORMAT(/5X,'NO. OF ITERATION =' ,I3,5X,'STATION' ,I3
1,5X,'X =' ,F8.4,5X,'U-RF =' ,F6.3)
C

```

```

C          IF(I.LE.M1.OR.I.GE.M2) GO TO 2222
C          2098 WRITE(9,2099) UTAU(I),CFS(I)
C          2099 FORMAT(5X,'UTAU=',E12.4,5X,'TAUW=',E12.4)
C          2222 WRITE(9,3001)
C          3001 FORMAT(/,5X,'Y-CORD',6X,'VEL U',7X,'VEL V',6X,
C              1,'TURB KE',5X,'EFF VISC',3X,'PRESSURE'//)
C          DO 3003 J=1,JP1
C          3003 WRITE(9,2007) YP(I,J),UE(I,J),VN(I,J)
C              1,'AKE(I,J),VIS(I,J),PR(I,J)
C          3333 CONTINUE
C          2117 FORMAT(4E12.4)
C          2007 FORMAT(6E12.4)
C          C          GO TO 5555
C          C          SAVE DATA FOR ANOTHER CALCULATION
C          C          IF NECESSARY
C          C          OPEN(UNIT=11,FILE='FVINPUT')
C          C          DO 169 I=1,IP1
C          C          DO 169 J=1,JP1
C          C          169 WRITE(11,2008) UE(I,J),VE(I,J),UN(I,J),VN(I,J)
C              1,'AKE(I,J),ADS(I,J),PR(I,J)
C          C          CLOSE(11)
C          C          SAVE DATA FOR PLOTTING.
C          C          WRITE(7,2009) (PR(I,1),I=2,IMAX)
C          C          WRITE(7,2009) (UE(I,1),I=M2,IMAX)
C          C          WRITE(7,2009) (UTAU(I),I=M1PI,M2M1)
C          C          DO 5001 I=1,IP1
C          C          DO 5001 J=1,JP1
C          C          5001 WRITE(7,2005)UE(I,J),VN(I,J),PR(I,J)
C              1,'AKE(I,J),ADS(I,J),VIS(I,J)
C          C          2005 FORMAT(6E12.4)
C          C          2009 FORMAT(6E12.4)
C          C          5555 CONTINUE
C          C          CLOSE(7)
C          C          CLOSE(8)
C          C          CLOSE(9)
C          C          STOP
C          C          END
C          C          SUBROUTINE TRIDAG(IF,L,A,B,C,D,V)

```

```

C          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C          PARAMETER(JD=45)
C          DIMENSION A(JD),B(JD),C(JD),D(JD),V(JD),BETA(JD),GAMMA(JD)
C          BETA(IF)=B(IF)
C          GAMMA(IF)=D(IF)/BETA(IF)
C          IFFI=IF+1
C          DO 1 I=IFFI,L
C          BETA(I)=B(I)-A(I)*C(I-1)/BETA(I-1)
C          1 GAMMA(I)=(D(I)-A(I)*GAMMA(I-1))/BETA(I)
C          V(L)=GAMMA(L)
C          LAST=L-IF
C          DO 2 K=1,LAST
C          I=L-K
C          2 V(I)=GAMMA(I)-C(I)*V(I+1)/BETA(I)
C          RETURN
C          END
C          SUBROUTINE PRESU(P,DHS,I,M)
C          IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C          PARAMETER(ID=152,JD=45)
C          DIMENSION P(ID,JD)
C          DIMENSION DHS(ID,JD)
C          COMMON/PRES/BCU(ID,JD),BCV(ID,JD)
C          COMMON/AA/AA(JD),BB(JD),CC(JD),DD(JD),T(JD)
C          COMMON/CONSI/IMAX,JMAX,IPI,JP1,JPP1
C          DO 660 J=2,JMAX
C          AA(J)=-BCV(I,J-1)
C          BB(J)=BCU(I-1,J)*M+BCU(I,J)+BCV(I,J-1)
C          1 +BCV(I,J)
C          CC(J)=-BCV(I,J)
C          660 DD(J)=BCU(I,J)*P(I+1,J)+BCU(I-1,J)*P(I-1,J)
C              1 -DHS(I,J)
C          CALL TRIDAG(2,JMAX,AA,BB,CC,DD,T)
C          DO 671 J=2,JMAX
C          671 P(I,J)=T(J)
C          RETURN
C          END

```