

1989

Simulation of time-dependent compressible viscous flows using central and upwind-biased finite-difference techniques

Edward Joseph Hall
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Aerospace Engineering Commons](#)

Recommended Citation

Hall, Edward Joseph, "Simulation of time-dependent compressible viscous flows using central and upwind-biased finite-difference techniques" (1989). *Retrospective Theses and Dissertations*. 9128.
<https://lib.dr.iastate.edu/rtd/9128>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book. These are also available as one exposure on a standard 35mm slide or as a 17" x 23" black and white photographic print for an additional charge.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 9014903

**Simulation of time-dependent compressible viscous flows using
central and upwind-biased finite difference techniques**

Hall, Edward Joseph, Ph.D.

Iowa State University, 1989

U·M·I

**300 N. Zeeb Rd.
Ann Arbor, MI 48106**

**Simulation of time-dependent compressible viscous flows using
central and upwind-biased finite-difference techniques**

by

Edward Joseph Hall

**A Dissertation Submitted to the
Graduate Faculty in Partial Fulfillment of the
Requirements for the Degree of
DOCTOR OF PHILOSOPHY
Major: Mechanical Engineering**

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

**Iowa State University
Ames, Iowa
1989**

Copyright © Edward Joseph Hall, 1989. All rights reserved.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	xvii
NOTATION	xx
CHAPTER 1. INTRODUCTION	1
Description of the Problem	1
Historical Review	7
Analytical Methods	7
Numerical Methods	9
Scope of the Present Research	17
Description of the Geometries	20
CHAPTER 2. MATHEMATICAL DEVELOPMENT	24
Nondimensionalization	25
Navier-Stokes Equations	26
Coordinate Transformation	30
Strong Conservation Law Form	31
Boundary Conditions	35
CHAPTER 3. NUMERICAL ALGORITHMS	38

One-Dimensional Study	38
Two-Dimensional Time-Marching Algorithms	52
Grid Generation	55
Hopscotch Method	67
Runge-Kutta Method	73
Approximately Factored Linearized Block Implicit (AFLBI) Method .	77
Generalized Conjugate Gradient Total Variation Diminishing (GCGTVD) Method	83
CHAPTER 4. NUMERICAL RESULTS	99
Steady Inviscid Flow about an Isolated Cylinder	103
Critical Flow Case	105
Supercritical Flow Case	105
Steady Inviscid Flow Through a Low Stagger Turbine Cascade	113
Steady Inviscid Flow Through a High Turning Turbine Cascade	123
Steady Viscous Flow Through a High Turning Turbine Cascade	124
Steady Viscous Subsonic Flow over a Flat Plate	143
Unsteady Viscous Flow near an Oscillating Flat Plate	152
Unsteady Viscous Flow about an Impulsively Started Cylinder	166
Reynolds Number = 3000	168
Reynolds Number = 9500	172
CHAPTER 5. CONCLUDING REMARKS	235
Conclusions	235
Recommendations for Future Research	237

BIBLIOGRAPHY	239
APPENDIX . EXPLICIT METHOD OF CHARACTERISTICS	
BOUNDARY ALGORITHM	249

LIST OF TABLES

Table 3.1:	Uniform flow regions of one-dimensional shock tube flow . . .	40
Table 3.2:	Explicit algorithms for shock tube flow study	52
Table 3.3:	Implicit algorithms for shock tube flow study	53
Table 4.1:	Steady flow verification test cases	101
Table 4.2:	Unsteady flow verification test cases	102
Table 4.3:	Supercritical inviscid isolated cylinder CPU time comparison	113
Table 4.4:	Inviscid low stagger turbine cascade CPU time comparison .	123
Table 4.5:	Viscous high-turning turbine cascade CPU time comparison .	132
Table 4.6:	Oscillating flat plate flow data	164
Table 4.7:	Oscillating flat plate CPU time comparison	166
Table 4.8:	Impulsively started cylinder flow data, $Re=3000$	169
Table 4.9:	Impulsively started cylinder flow CPU time comparison, $Re=3000$	169
Table 4.10:	Impulsively started cylinder flow data, $Re=9500$	173
Table 4.11:	Impulsively started cylinder flow CPU time comparison, $Re=9500$	173

LIST OF FIGURES

Figure 1.1:	Two-dimensional planar channel flow geometry definition . .	21
Figure 1.2:	Spatially periodic planar cascade geometry definition	23
Figure 3.1:	One-dimensional shock tube geometry definition	39
Figure 3.2:	Wave propagation for one-dimensional shock tube flow problem	41
Figure 3.3:	Comparison of the predicted hopscotch and analytical density ratio for one-dimensional shock tube problem	45
Figure 3.4:	Comparison of the predicted second order Runge-Kutta and analytical density ratio for one-dimensional shock tube problem	46
Figure 3.5:	Comparison of the predicted fourth order Runge-Kutta and analytical density ratio for one-dimensional shock tube problem	47
Figure 3.6:	Comparison of the predicted explicit second order TVD and analytical density ratio for one-dimensional shock tube problem	48

Figure 3.7:	Comparison of the predicted linearized block implicit method with nonlinear damping and analytical density ratio for one-dimensional shock tube problem	49
Figure 3.8:	Comparison of the predicted implicit third order TVD and analytical density ratio for one-dimensional shock tube problem	50
Figure 3.9:	Sample H-type grid system for internal flow calculations . . .	56
Figure 3.10:	Sample O-type grid system for cascade flow calculations . . .	57
Figure 3.11:	Physical and computational mapping of H-type grid system .	59
Figure 3.12:	Physical and computational mapping of O-type grid system .	60
Figure 3.13:	Detail of leading edge region for high-turning turbine blade grid	66
Figure 3.14:	Grid-centered finite-volume arrangement for GCGTVD scheme	87
Figure 3.15:	Reimann problem at a grid cell interface for the GCGTVD scheme	88
Figure 4.1:	85x30 grid for inviscid isolated cylinder flow calculations . . .	104
Figure 4.2:	Comparison of the predicted cylinder surface Mach number distributions for critical inviscid isolated cylinder flow	106
Figure 4.3:	Comparison of the predicted cylinder surface Mach number distributions for supercritical inviscid isolated cylinder flow .	108
Figure 4.4:	Predicted hopscotch Mach number contours for supercritical inviscid isolated cylinder flow	109

Figure 4.5:	Predicted Runge-Kutta Mach number contours for supercritical inviscid isolated cylinder flow	110
Figure 4.6:	Predicted AFLBI Mach number contours for supercritical inviscid isolated cylinder flow	111
Figure 4.7:	Predicted GCGTVD Mach number contours for supercritical inviscid isolated cylinder flow	112
Figure 4.8:	91x9 grid system for low stagger turbine blade cascade	115
Figure 4.9:	Comparison of the predicted and experimental airfoil surface static to total pressure ratio distributions for inviscid flow through a low stagger turbine cascade (91x9 grid)	116
Figure 4.10:	171x15 grid system for low stagger turbine blade cascade . .	117
Figure 4.11:	Comparison of the predicted and experimental airfoil surface static to total pressure ratio distributions for inviscid flow through a low stagger turbine cascade (171x15 grid)	118
Figure 4.12:	Predicted hopscotch Mach number contours for inviscid flow through a low stagger turbine cascade	119
Figure 4.13:	Predicted Runge-Kutta Mach number contours for inviscid flow through a low stagger turbine cascade	120
Figure 4.14:	Predicted AFLBI Mach number contours for inviscid flow through a low stagger turbine cascade	121
Figure 4.15:	Predicted GCGTVD Mach number contours for inviscid flow through a low stagger turbine cascade	122
Figure 4.16:	High-turning turbine vane 157x15 grid	125

Figure 4.17: Comparison of the predicted and experimental airfoil surface static to total pressure ratio distributions for inviscid flow through a high-turning turbine cascade	126
Figure 4.18: Predicted hopscotch Mach number contours for inviscid flow through a high-turning turbine cascade	127
Figure 4.19: Predicted Runge-Kutta Mach number contours for inviscid flow through a high-turning turbine cascade	128
Figure 4.20: Predicted AFLBI Mach number contours for inviscid flow through a high-turning turbine cascade	129
Figure 4.21: Predicted GCGTVD Mach number contours for inviscid flow through a high-turning turbine cascade	130
Figure 4.22: 157x30 grid system for high-turning turbine cascade	133
Figure 4.23: Comparison of the predicted and experimental airfoil surface static to total pressure ratio distributions for viscous flow through a high-turning turbine cascade	134
Figure 4.24: Predicted hopscotch Mach number contours for viscous flow through a high-turning turbine cascade	135
Figure 4.25: Predicted Runge-Kutta Mach number contours for viscous flow through a high-turning turbine cascade	136
Figure 4.26: Predicted AFLBI Mach number contours for viscous flow through a high-turning turbine cascade	137
Figure 4.27: Predicted GCGTVD Mach number contours for viscous flow through a high-turning turbine cascade	138

Figure 4.28: Leading edge detail of the predicted hopscotch velocity vectors for viscous flow over a high-turning turbine cascade	139
Figure 4.29: Leading edge detail of the predicted Runge-Kutta velocity vec- tors for viscous flow over a high-turning turbine cascade	140
Figure 4.30: Leading edge detail of the predicted AFLBI velocity vectors for viscous flow over a high-turning turbine cascade	141
Figure 4.31: Leading edge detail of the predicted GCGTVD velocity vectors for viscous flow over a high-turning turbine cascade	142
Figure 4.32: Geometry definition for flow over a flat plate	144
Figure 4.33: 121x51 grid for flow over a flat plate	146
Figure 4.34: Comparison of predicted hopscotch and Blasius solution boundary-layer flow quantities for flow over a flat plate	148
Figure 4.35: Comparison of predicted Runge-Kutta and Blasius solution boundary-layer flow quantities for flow over a flat plate	149
Figure 4.36: Comparison of predicted AFLBI and Blasius solution boundary- layer flow quantities for flow over a flat plate	150
Figure 4.37: Comparison of predicted GCGTVD and Blasius solution boundary-layer flow quantities for flow over a flat plate	151
Figure 4.38: Geometry definition for oscillating flat plate flow	154
Figure 4.39: 41x51 grid for oscillating flat plate flow	155
Figure 4.40: Predicted hopscotch unsteady wall skin-friction coefficient for oscillating flat plate flow	156

Figure 4.41: Predicted Runge-Kutta unsteady wall skin-friction coefficient for oscillating flat plate flow	157
Figure 4.42: Predicted AFLBI unsteady wall skin-friction coefficient for oscillating flat plate flow	158
Figure 4.43: Predicted GCGTVD unsteady wall skin-friction coefficient for oscillating flat plate flow	159
Figure 4.44: Predicted hopscotch unsteady velocity profiles for oscillating flat plate flow	160
Figure 4.45: Predicted Runge-Kutta unsteady velocity profiles for oscillating flat plate flow	161
Figure 4.46: Predicted AFLBI unsteady velocity profiles for oscillating flat plate flow	162
Figure 4.47: Predicted GCGTVD unsteady velocity profiles for oscillating flat plate flow	163
Figure 4.48: Impulsively started cylinder flow description	175
Figure 4.49: Transient vortex development of impulsively started cylinder flow	176
Figure 4.50: 361x99 grid for impulsively started cylinder flow	177
Figure 4.51: Experimental flow visualization for impulsively started cylinder ($Re=3000$), $A:Vt/D=0.5$, $B:Vt/D=1.0$	178
Figure 4.52: Experimental flow visualization for impulsively started cylinder ($Re=3000$), $A:Vt/D=1.5$, $B:Vt/D=2.0$	179

Figure 4.53: Experimental flow visualization for impulsively started cylinder ($Re=3000$), $Vt/D=2.5$	180
Figure 4.54: Comparison of predicted hopscotch and experimental unsteady velocity profiles for impulsively started cylinder flow ($Re=3000$)	181
Figure 4.55: Comparison of predicted Runge-Kutta and experimental unsteady velocity profiles for impulsively started cylinder flow ($Re=3000$)	182
Figure 4.56: Comparison of predicted AFLBI and experimental unsteady velocity profiles for impulsively started cylinder flow ($Re=3000$)	183
Figure 4.57: Comparison of predicted GCGTVD and experimental unsteady velocity profiles for impulsively started cylinder flow ($Re=3000$)	184
Figure 4.58: Hopscotch predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=0.5$)	185
Figure 4.59: Hopscotch predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=1.0$)	186
Figure 4.60: Hopscotch predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=1.5$)	187
Figure 4.61: Hopscotch predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=2.0$)	188

Figure 4.62: Hopscotch predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=2.5$)	189
Figure 4.63: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=0.5$)	190
Figure 4.64: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=1.0$)	191
Figure 4.65: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=1.5$)	192
Figure 4.66: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=2.0$)	193
Figure 4.67: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=2.5$)	194
Figure 4.68: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=0.5$)	195
Figure 4.69: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=1.0$)	196
Figure 4.70: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=1.5$)	197
Figure 4.71: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=2.0$)	198
Figure 4.72: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=2.5$)	199

Figure 4.73: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=0.5$)	200
Figure 4.74: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=1.0$)	201
Figure 4.75: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=1.5$)	202
Figure 4.76: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=2.0$)	203
Figure 4.77: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=2.5$)	204
Figure 4.78: Experimental flow visualization for impulsively started cylinder ($Re=9500$), $A:Vt/D=0.6$, $B:Vt/D=0.8$, $C:Vt/D=1.0$. .	205
Figure 4.79: Experimental flow visualization for impulsively started cylinder ($Re=9500$), $A:Vt/D=1.4$, $B:Vt/D=1.6$, $C:Vt/D=2.0$. .	206
Figure 4.80: Comparison of predicted hopscotch and experimental unsteady velocity profiles for impulsively started cylinder flow ($Re=9500$)	207
Figure 4.81: Comparison of predicted Runge-Kutta and experimental unsteady velocity profiles for impulsively started cylinder flow ($Re=9500$)	208
Figure 4.82: Comparison of predicted AFLBI and experimental unsteady velocity profiles for impulsively started cylinder flow ($Re=9500$)	209

Figure 4.83: Comparison of predicted GCGTVD and experimental unsteady velocity profiles for impulsively started cylinder flow (Re=9500)	210
Figure 4.84: Hopscotch predicted streamline pattern for impulsively started cylinder flow. (Re=9500, t=0.6)	211
Figure 4.85: Hopscotch predicted streamline pattern for impulsively started cylinder flow. (Re=9500, t=0.8)	212
Figure 4.86: Hopscotch predicted streamline pattern for impulsively started cylinder flow. (Re=9500, t=1.0)	213
Figure 4.87: Hopscotch predicted streamline pattern for impulsively started cylinder flow. (Re=9500, t=1.4)	214
Figure 4.88: Hopscotch predicted streamline pattern for impulsively started cylinder flow. (Re=9500, t=1.6)	215
Figure 4.89: Hopscotch predicted streamline pattern for impulsively started cylinder flow. (Re=9500, t=2.0)	216
Figure 4.90: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. (Re=9500, t=0.6)	217
Figure 4.91: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. (Re=9500, t=0.8)	218
Figure 4.92: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. (Re=9500, t=1.0)	219
Figure 4.93: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. (Re=9500, t=1.4)	220

Figure 4.94: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=9500, t=1.6$)	221
Figure 4.95: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=9500, t=2.0$)	222
Figure 4.96: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=9500, t=0.6$)	223
Figure 4.97: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=9500, t=0.8$)	224
Figure 4.98: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=9500, t=1.0$)	225
Figure 4.99: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=9500, t=1.4$)	226
Figure 4.100: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=9500, t=1.6$)	227
Figure 4.101: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=9500, t=2.0$)	228
Figure 4.102: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=9500, t=0.6$)	229
Figure 4.103: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=9500, t=0.8$)	230
Figure 4.104: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=9500, t=1.0$)	231

Figure 4.105: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=1.4$)	232
Figure 4.106: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=1.6$)	233
Figure 4.107: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=2.0$)	234
Figure A.1: Characteristic paths for two-dimensional subsonic normal inlet flow calculations	252
Figure A.2: Characteristic paths for two-dimensional subsonic normal exit flow calculations	253

ACKNOWLEDGEMENTS

This research was supported in part by a grant from the Allison Gas Turbine Division of the General Motors Corporation. Additional support was provided by the National Science Foundation under grant CBT-8211713 and the Iowa State University Department of Mechanical Engineering. Support for completing the numerical computations was provided by a block grant from the Iowa State University Computation Center, a grant from the National Center for Supercomputing Applications at the University of Illinois, and a grant from the NASA Lewis Research Center for usage of the Numerical Aerodynamic Simulation facilities at the NASA Ames Research Center, Moffett Field, California. I am extremely grateful to the Allison Gas Turbine Division for choosing me as a recipient of a General Motors Fellowship, and the benefits that the program provides. I wholeheartedly applaud their support of employee graduate study and hope that many more people can benefit from this program. I am grateful for the helpful suggestions and friendship extended by Bob Delaney at Allison.

I am extremely grateful for the guidance and support provided by my advisor, Dr. Richard H. Pletcher, who somehow maintains a sense of humor through every crisis, and who always seemed to ask the right question when I couldn't find an answer. Many thanks are also extended to the other members of my graduate committee,

Drs. Ted Okiishi, Bruce Munson, John Tannehill, and Joe Prusa.

I am particularly grateful to my parents, Joseph L. and Anna Mae Hall for their patience with a son who never seems to finish school. I love you both very much, and I am grateful that you taught me the diligence required to finish this study. Similar wishes are extended to the other members of my immediate family: Becky, Jerry, Molly, and Jordan Clarke, Curt, Sara, Rachel, and Vanessa Fewell, for their participation in my much-needed retreats to Dubuque, Joanne Hall and Pat for their understanding of the trials and tribulations of graduate study, and Todd and Heidi Inskeep for keeping me in touch with my sanity and former life in Indianapolis.

I would also like to thank all my loyal friends for providing enough diversions to keep me from graduating early. In particular, Roxanne and Dave in Ames, Pat in Dubuque, and Kent, Neil, and the entire Dealers' Choice softball team in Indianapolis.

I appreciate the encouragement, friendship, and advice given by Kuditipudi Venkatram Rao and the other members of the Advanced Turbomachinery Department at Allison during the latter stages of this work.

To my best friend, Pete Hanger, thanks for the many trips to Ames and other parts unknown. I will always be thankful for the loyalty and friendship that you have always shown me, and I hope that you are thankful for mine.

To my son, David, (and any future sons or daughters), I promise to be around more to teach you the things my father taught me. I hope you can enjoy the benefits of education as much as I have. (Hint: do well in school.)

To my wife Sandra, who shared every moment of my anguish during the last year of this study, what can I say except to tell you I love you more now than ever before,

and I hope that I will never again have to know life without you. This is only the beginning.

NOTATION

Roman Symbols

a ... speed of sound

b ... row vector of $R_{\xi,\eta}^{-1}$

c_1 ... second order damping coefficient

c_2 ... fourth order damping coefficient

c_p ... specific heat at constant pressure

c_v ... specific heat at constant volume

c_x ... axial chord length

c_ξ ... damping coefficient in ξ direction

c_η ... damping coefficient in η direction

e ... internal energy

f ... external force

h ... convective heat transfer coefficient or enthalpy

i ... ξ index of numerical solution

j ... η index of numerical solution

k ... thermal conductivity

l ... iteration index

- n ... time step index of numerical solution
- \vec{n} ... outward unit normal vector
- p ... pressure
- q ... column vector of $R_{\xi,\eta}$ or heat flux
- r ... residual vector or cylinder radius
- s ... search vector for conjugate gradient or cascade pitch spacing
- t ... time
- u ... velocity in x direction
- v ... velocity in y direction
- x ... horizontal Cartesian coordinate
- y ... vertical Cartesian coordinate
- A ... inviscid ξ flux Jacobian matrix
- ADI ... alternating direction implicit
- AF ... approximate factorization
- $AFLBI$... approximately factored linearized block implicit
- B ... inviscid η flux Jacobian matrix
- C_f ... skin-friction coefficient
- CFL ... Courant-Freidrichs-Levy
- D ... dissipation flux vector or cylinder diameter
- E ... Cartesian flux vector in x direction
- E_t ... total internal energy
- F ... Cartesian flux vector in y direction
- G ... implicit ξ coefficient operator for TVD scheme

GCGTVD ... generalized conjugate gradient TVD
H ... implicit η coefficient operator for TVD scheme
H_t ... total enthalpy
I ... identity matrix
J ... grid transformation Jacobian
L ... length
M ... unfactored implicit coefficient matrix or Mach number
Nu ... Nusselt number
O ... measure of orthogonality
Pr ... Prandtl number
Q ... Cartesian vector of dependent variables
R ... gas constant
Re ... Reynolds number
RHS ... right hand side term
R_{ξ,η} ... eigenvector matrix of A, B decomposition
S ... Cartesian source vector or arc length
T ... temperature
TVD ... total variation diminishing
U ... contravariant velocity
V ... absolute velocity

Greek Symbols

α ... time-stepping factor
 ρ ... density

- γ ... specific heat ratio
 γ^0 ... cascade setting angle
 δ ... spatial central difference operator
 δ_2 ... cascade deviation angle
 ι_1 ... cascade incidence angle
 κ_1 ... cascade design inlet flow angle
 κ_2 ... cascade design exit flow angle
 λ ... eigenvalue of flux Jacobian matrix
 ξ ... horizontal transformed coordinate
 η ... vertical transformed coordinate
 ϕ ... row vector product of $\hat{R}^{-1} \Delta \hat{Q}$
 ψ ... source term for method of characteristics boundary condition
 σ ... damping factor
 τ ... viscous shear stress
 μ ... viscosity
 Υ ... relative pressure gradient coefficient
 Ψ ... entropy inequality function for TVD scheme
 Δ ... increment of change

Special Symbols

- ∇ ... spatial vector gradient operator
 $\Lambda_{\xi, \eta}$... eigenvalue matrix of A, B decomposition
 Δ ... spatial forward difference operator
 ∇ ... spatial backward difference operator

Superscripts

- [$\bar{\quad}$] ... dimensional variable or averaged variable
- [$\hat{\quad}$] ... transformed variable
- [$\tilde{\quad}$] ... flux limited variable
- [$\vec{\quad}$] ... vector variable
- [\ast] ... intermediate variable
- [$\text{}^R/L$] ... R/L side of a cell interface
- [$\text{}^n$] ... time step index of variable

Subscripts

- [$\text{}_{i,j}$] ... grid point index of variable
- [$\text{}_f$] ... related to friction
- [$\text{}_p$] ... related to pressure
- [$\text{}_t$] ... total quantity
- [$\text{}_x$] ... derivative or value with respect to x
- [$\text{}_y$] ... derivative or value with respect to y
- [$\text{}_L$] ... derivative along a characteristic surface
- [$\text{}_{inv}$] ... inviscid variable
- [$\text{}_{vis}$] ... viscous variable
- [$\text{}_\xi$] ... derivative or value with respect to ξ
- [$\text{}_\eta$] ... derivative or value with respect to η
- [$\text{}_{loc}$] ... local value
- [$\text{}_{ray}$] ... value along a ray or grid line

[]*ref* ... reference value

[]*surf* ... value on a surface

[]*wall* ... value on the wall

CHAPTER 1. INTRODUCTION

Description of the Problem

Modern aerodynamic design analysis relies heavily on the use of discrete numerical calculations to predict fluid flow-related performance. This trend has been spurred in recent years by increases in computational efficiency coupled with decreasing costs for computational studies versus experimental studies. In fact, most preliminary design decisions for aerodynamic-related components are now commonly based on the results of numerical flow simulations. Design practices constantly improve by utilizing more complicated and exact analyses as they become feasibly available. Dramatic improvements in solution accuracy and efficiency have been achieved due to advances in computational hardware and through numerical algorithms developed for specific flow conditions. As an example, space-marching steady flow analyses exploit the parabolic nature of certain flows [1] to reduce the complexity of the governing equations and enhance the efficiency of the numerical scheme. In general, no single algorithm has yet been found to adequately simulate all flow regimes, but for most well-defined situations, at least one scheme can be identified which may be expected to provide a reasonable solution.

Recently, a great deal of interest has focused on the ability to analyze and under-

stand the consequences of unsteady aerodynamic flow phenomena. Current problems of interest in this area include interblade row wake transport in turbomachines [2]-[5], helicopter rotor flow unsteadiness, airfoil aerodynamic-structural dynamic interaction [6], counter-rotating propeller systems [7], and the influence of unsteadiness on separated flow behavior [8]. Several factors motivate the need to investigate the prediction of unsteady flows. Current aerodynamic design analyses are typically based on the results of steady flow calculations, often even when the primary flow is known to be unsteady. In contrast, however, it has been observed that the aerodynamic time mean of unsteady flow activity does not necessarily correlate with steady state predictions. In addition, most unsteady flow analysis techniques are based on an overly simplified set of equations which neglect a significant portion of the physics governing such flows. In a review by Chi and Srinivasan [9], recent advances in the understanding and prediction of turbomachine subsonic stall flutter were discussed. Their study indicated that although the development of aerodynamic theories for the prediction of unsteady subsonic stall flutter were encouraging, a better understanding of unsteady separated flow and unsteady shock motion were required for reliable predictions.

Several aspects of unsteady viscous flow behavior remain completely unexplained. In a recent experimental study, Covert and Lorber [10] observed that the turbulent velocity profiles for the flow over an oscillating NACA0012 airfoil became independent of the mean adverse pressure gradient at higher reduced frequencies, indicating some decoupling of the viscous flow region from the outer flow. In fact, the sporadic nature of turbulence dictates that most flows of practical interest are inherently unsteady. The potential availability of high quality experimental data for unsteady viscous flows

obtained with laser velocimetry [11] also makes this an appealing area for study.

During the last decade, advances in computer architecture and computational algorithms have enhanced the feasibility of performing complex unsteady flow calculations. In a review by McCroskey et al. [12], the requirements for performing complex unsteady viscous flow simulations are outlined. To evaluate the potential for performing complex unsteady aerodynamic calculations accurately, the availability, performance, and memory capacity of modern computer architectures must be balanced against the efficiency, reliability, and storage needs of advanced numerical algorithms. The widespread availability of high speed supercomputers with a large internal memory capacity and multiprocessor or vector processor architectures, such as the Cray series of computers, is certainly a requirement for this undertaking using current computational methods. This study concludes that the technology required to perform detailed unsteady viscous flow calculations is presently available, and that such calculations are likely to become an important area of study in the near future. This report deals with an investigation into the application and development of numerical techniques for the prediction of complex unsteady compressible viscous flows.

A review of state-of-the-art finite-difference numerical techniques for solving the equations governing unsteady compressible flows reveals that most successful methods are based on time-marching, shock-capturing algorithms. Such formulations are advantageous in that the same solution routine may be used for both steady and unsteady flow calculations. By utilizing a strong conservation law form of the governing equations, flow discontinuities such as shock waves or contact discontinuities can be

automatically captured [13] as the solution develops without special treatment. This approach was originally proposed in a celebrated paper by Lax [14] which ultimately formed the basis for many modern compressible time-marching flow calculation techniques.

Time-marching algorithms may generally be categorized as being either explicit or implicit in nature. Explicit algorithms advance the solution on a point by point basis using only the information available at the known time level. Explicit algorithms are typically simpler to program and execute, but suffer from a grid-dependent time step limitation due to the restricted domain of dependence of the numerical solution. Implicit schemes, on the other hand, are formulated based on information at the new time level, and therefore require the solution of a complicated system of simultaneous equations. Implicit schemes are touted as being unconditionally stable based on linear stability analysis; however, the additional errors introduced when the equations are linearized and/or factored impose practical stability bounds on many implicit algorithms. In practice, for unsteady flow predictions, the time step must be further restricted for implicit schemes to prevent the accumulation of temporal inaccuracies. In fact, some investigators have reported that in order to obtain reasonably accurate unsteady results with inviscid implicit schemes, the explicit stability-related time step must be used. Many steady-state solution schemes seek to incorporate the advantages of both the explicit and implicit schemes in a type of hybrid algorithm. This approach, at first glance, appears not to be well suited for the prediction of unsteady flows due to the ad hoc nature of the implicit portion of the algorithm. Recently, however, Jorgensen and Chima [4] demonstrated that such algorithms may be tailored for the

unsteady flow problem to yield acceptable results.

A number of successful finite-difference schemes rely on some form of central differencing to approximate the spatial derivatives in the governing equations. Central differencing yields a compact, second-order accurate expression, which, in general, does not require special reformulation near boundaries. Unfortunately, central-differencing can also lead to nonphysical oscillations in the numerical solution near steep gradients and often requires the addition of artificial damping terms to stabilize the solution. These damping terms are typically constructed rather arbitrarily and generally rely on the experience of the user to specify the magnitude of the added numerical dissipation.

Recently, several algorithms have appeared which exploit the characteristic nature of the equations governing unsteady compressible flow in order to correctly apply upwind-differencing techniques [15] for systems of conservation laws. One example of this type of scheme is the flux vector splitting approach. Flux vector splitting is based on determining that portion of a numerical flux vector which is based on information from upstream traveling waves, and that portion which is based on information from downstream traveling waves. Each portion of the flux vector is then differenced in an upwind manner according to the directional propagation of the carrier wave. Examples of this type of scheme are given by Steger and Warming [16] and Van Leer [17]. A second upwind method expresses the difference between adjacent fluxes in terms of the positively and negatively traveling waves. This so-called flux-difference splitting approach has been utilized by a number of authors including Roe [18], Yee and Harten [19], Chakravarthy and Osher [20], and Chakravarthy [21],[22]. An important class of

algorithms of this type is referred to as total variation diminishing (TVD) since they are based on the principle of nonincreasing total variation for scalar conservation laws [19]. The nonoscillatory solution behavior associated with TVD schemes is obtained by a limiting process which can locally reduce the scheme to first order spatial accuracy. However, the resulting numerical solution is reasonably accurate and is free from the spurious oscillations near discontinuities which occur for the more standard central-difference schemes, without the introduction of added numerical dissipation. Most calculations involving upwind-difference techniques have been performed for steady, inviscid flows, and more intensive investigation is required to determine how these schemes perform for time-accurate calculations, and flows where viscous effects are considered.

Other numerical considerations often become important in the simulation of unsteady flows. One of the more demanding aspects of the numerical prediction of unsteady flows is the treatment of inflow and outflow boundaries. Physical reasoning suggests that throughflow boundaries must be correctly modelled to permit the unaltered propagation of radiating waves into and out of the flow domain without the destructive effects of artificial reflections. The numerical application of physically realistic inflow and outflow boundary conditions poses one of the greatest challenges in the prediction of unsteady flows. The boundary condition model has the responsibility of maintaining the desired time-mean flow, while still permitting the local unsteady activity to propagate unaltered. Erdos and Alzner [5], Tong [23], and Hedstrom [24] have all described techniques based on characteristic theory which approximate this behavior. Previous experience has shown that their methods perform acceptably for

simple planar wave fronts, but may be less accurate for nonplanar wave fronts due to an inherent one-dimensional assumption.

In summary, the preceding paragraphs have detailed the practical utility, motivation, and computational considerations for the aerodynamic analysis of unsteady compressible viscous flows. The desire to accurately calculate unsteady viscous flows and assess the impact of the many computational uncertainties form the basis of the remainder of this study. The analysis of compressible flows is particularly enticing in light of the numerous algorithmic developments which have occurred for compressible flow solvers in recent years. A perspective on the analysis of unsteady compressible viscous flows is given in the following section.

Historical Review

Unsteady flows fall into two general categories: transient flow occurring due to a change in some initial flow, and time-periodic flows resulting from periodic vortex shedding or oscillatory motion. It would be impossible to detail all of the pertinent works relating to unsteady flows of all kinds here; therefore, the literature review provided below will concentrate primarily on methods for predicting the characteristics of unsteady flows. Separate sections are provided for analytical and numerical studies.

Analytical Methods

Analytical solutions for unsteady compressible flow problems have been difficult to obtain with the exception of a few simple geometries or idealized flows. The

unsteady one-dimensional flow of a compressible gas in a shock tube is a classic example of such a solution. Unfortunately, the limited analytical results are generally only useful for understanding observable trends in more complicated unsteady flows. The underlying concepts describing the behavior of unsteady compressible flows are best described by the method of characteristics. Characteristic theory reduces the governing system of hyperbolic partial differential equations to a system of total differential equations which are applicable along unique characteristic surfaces. Each characteristic surface represents the propagation path of disturbances in the fluid, and hence governs the exchange of information throughout the flow. The characteristic representation permits continuous solutions which have discontinuous derivatives across characteristic surfaces. Thus, the movement of flow discontinuities such as shock waves and/or contact surfaces are also fully explained. A complete description and application of characteristic theory to compressible flow is beyond the realm of this report, and the interested reader is encouraged to seek more information in the references [25],[26]. Rather, the importance of the physical insight gained from characteristic analysis must be recognized in our attempt to numerically simulate unsteady compressible flows. A number of numerical algorithms are constructed on the basis of characteristic theory alone, and it is often possible to determine algorithmic limitations through assimilation of the numerical process with known characteristic behavior.

There exist several useful analytical solutions for unsteady viscous flows with simple geometries with which the time-dependent numerics of candidate algorithms might be compared. Stokes [27] derived an analytical solution for the Navier-Stokes

equations for an otherwise stagnant flowfield in the vicinity of a sinusoidally oscillating flat plate. The result is a time-periodic solution which illustrates the behavior of the velocity profiles near the plate as a function of time. It is also possible to derive an expression for the start-up transient solution behavior for the same case, as the solution approaches the time-periodic solution. This solution suggests that the initial transient solution exists for about one cycle of the plate oscillation before the time-periodic solution is reached.

A second series of Navier-Stokes solutions exist for flow between two parallel plates as one is suddenly accelerated to a uniform velocity. The solution provides the transient velocity of the fluid as the steady state Couette flow is approached. In fact, it is possible to derive an analytical expression for any arbitrary plate motion since the plate motion can be expressed as a Fourier series in time having sinusoidal components.

A second set of analytical solutions are available based on the unsteady boundary layer equations. Rosenhead [28] details a number of theoretical solutions for boundary layers including the expression for the transient boundary layer behavior behind a moving shock wave investigated by Mirels [29]. Analytical expressions also exist for a boundary layer acting under the influence of a sinusoidally oscillating freestream [30].

Numerical Methods

Previous numerical methods for solving unsteady flowfields can generally be classified according to the form of the governing equations upon which the analysis is

based. Many of the original works were directed at the problem of unsteady flow about airfoils in an attempt to deal with the problems of buffeting on highly loaded wings.

The earliest works were based on a linearized set of inviscid equations obtained from small perturbation theory. The solutions were therefore limited to small fluctuations about an aerodynamic mean, and suffered from the limitations of the inviscid approximation. Despite these weaknesses, the linearized analyses demonstrated some utility in the area of oscillating airfoils and the resultant unsteady aerodynamic loading. Unfortunately, these schemes were soon found to be accurate only for conditions of small amplitude fluctuations and low airfoil loading levels where viscous effects and the nonlinearities of the problem are less important. Chow and Goorjian [31] utilized an alternating direction implicit (ADI) procedure to perform a numerical solution of the two-dimensional nonlinear low-frequency transonic small disturbance equation for unsteady transonic flows about oscillating airfoils with supersonic freestreams. Their results compared favorably with linear theory. An additional high-frequency modification was found to improve their results in some cases. Caspar and Verdon [32] calculated unsteady subsonic flow about an oscillating cascade based on linear small disturbance theory for small amplitude fluctuations about a nonuniform steady state flow. Calculations were performed on a nonorthogonal periodic sheared H-type grid system using a least squares difference approximation. Their results suggest that blade thickness produces a strong steady-unsteady flow interaction, especially at high Mach numbers. Unfortunately, their results suggested different aerodynamic trends for bending vibration compared to torsional oscillation, indicating the complexity of

establishing definite trends for unsteady flow behavior. Calculations of this type are further complicated in that errors in the steady state solution are directly propagated into the unsteady flow calculation.

Some improvements were realized by utilizing a more general form of the governing equations based on the potential function. The unsteady full potential formulation benefits from the ability to handle unsteady flow phenomena beyond simple harmonic oscillations. However, since the potential function relies on an irrotational and isentropic flow assumption, only very weak shocks are permitted in the solution, and airfoil calculations must utilize potential "jumps" in wake regions in order to properly satisfy the trailing edge Kutta condition. In addition, some form of artificial viscosity or artificial compressibility must be used to overcome the nonuniqueness problem of the full potential equation for transonic flows [33].

Krammer [34] developed a model for the prediction of the unsteady forces on turbomachinery blades based on two-dimensional incompressible potential flow. Blade element and viscous wakes were modelled using distributed vortices. Although the model was extremely crude, reasonable agreement with experimental data for the unsteady forces on an axial flow turbine blade was achieved.

Chipman and Jameson [35] utilized the full potential equation in conservative form for the prediction of unsteady irrotational transonic flow about airfoils. In this study, a two-dimensional sheared H-type grid system was used with time-varying coordinates and an ADI algorithm with artificial viscosity. Calculations were presented for flap oscillations of a bicircular arc airfoil.

Malone and Sankar [36] solved the two-dimensional full potential equation using

an implicit approximate factorization algorithm on a moving, body-fitted C-type coordinate system for pitching oscillation of a NACA 64A410 airfoil.

Steger and Cardonna [37] solved the unsteady full potential equation in conservative law form with an unsteady circulation model for lifting airfoil calculations. Results were presented for steady and unsteady airfoil flows, and an impulsively plunging flat plate.

Shankar et al. [38] utilized an approximate factorization scheme with Newton iteration at each time step in the implicit numerical solution of the conservative full potential equation. Potential jumps across wakes were determined by solving the vorticity convection equation. Excellent agreement with experimental results was observed for a NACA-0012 airfoil undergoing pitch oscillation about the quarter chord for a reduced frequency of 0.081. Somewhat less accurate results were given for pitch oscillation about the 40% chord location of a NLR-7301 airfoil.

In a truly extensive application, Sankar et al. [39] solved the three-dimensional full potential equation in conservative form using a hybrid differencing scheme and a strongly implicit solution procedure. The potential solutions were compared with a more complex Euler solution scheme. Results were compared for steady flow and pitching oscillations of a three dimensional fighter wing. The Euler solution required roughly five times the CPU time of the potential code, with little difference in the results for low transonic Mach numbers. These results demonstrate the usefulness of the full potential procedure for this class of flows.

The next step was taken when the fully nonlinear inviscid form of the governing equations were solved using time-marching, shock-capturing techniques. These

schemes demonstrated superior accuracy over a wider range of compressible flows, but were again limited to low loading levels due to the occurrence of flow separation at higher loading levels.

Jameson and Venkatakrisnan [40] demonstrated a Runge-Kutta based Euler code for the prediction of inviscid flows over harmonically oscillating airfoils using the Euler equations. This scheme utilized a TVD-based dissipation and nonreflecting farfield boundary conditions. The conclusion was that the TVD dissipation scheme was well suited for the prediction of unsteady flows.

Lewis et al. [2] incorporated a hopscotch-based Euler solver for the prediction of turbine vane-blade interaction. In this case, the unsteady flow is complicated by the requirement of a phase-lagged boundary condition in order to properly account for unsteady effects resulting from uneven numbers of blades and vanes in a turbine stage.

Allmaras and Giles [41] presented two-dimensional Euler results for a transonic diffuser flow with a sinusoidally oscillating exit pressure using a second-order accurate flux split scheme.

Belk and Whitfield [42] utilized an implicit, two factor split flux finite-volume scheme for the Euler equations on dynamic blocked grids. This approach is particularly useful for complicated geometries where a single grid is not easily generated. Results were presented for rigid body oscillatory pitching of a NACA0012 airfoil.

Anderson et al. [43] applied a flux splitting technique for transonic pitch oscillations of a NACA0012 airfoil with a moving grid. Their study also investigated the use of multigrid to aid in reducing the overall execution time.

Clearly, this list is far from complete, with new applications and techniques being published on a daily basis. This sampling of the available results, however, demonstrates the usefulness of the Euler equations for moderately loaded transonic unsteady flow calculations.

The solution of problems involving massive flow separation were originally restricted to steady flows. Several investigators have demonstrated that steady separated flows can be accurately modelled using approximate forms of the Navier-Stokes equations (both time-marching, and pressure-relaxation or space-marching methods) or by coupling special boundary-layer techniques with an outer freestream through viscous-inviscid interaction [44]. Limitations of computational resources and the inefficiencies of certain algorithms inhibited early attempts to solve unsteady viscous flow problems. During the past decade, however, tremendous advances have been made in both computer architecture and numerical analysis, and it is now plausible to economically solve many unsteady viscous flow problems.

The first numerical analyses developed for predicting general unsteady viscous flows were based on the unsteady boundary-layer equations. The unsteady boundary-layer equations have been shown to be capable of predicting some unsteady separated flows (separated in the sense that a negative streamwise velocity exists) with promising results. However, a recent paper by Cebeci [45] suggests that a singularity exists for certain unsteady boundary-layer flows evolving under a prescribed pressure gradient. This singularity may not be unlike the steady state boundary-layer singularity encountered at points of separation in a direct (prescribed pressure) steady flow boundary-layer solution. There is some speculation that the unsteady singularity

may be removed by utilizing an inverse solution scheme with a viscous-inviscid interaction algorithm similar to those used to predict steady separated flows. To date, very little information can be found concerning this approach. Perhaps the closest may be the viscous-inviscid interaction solutions performed by Rizzetta and Borland [46] for unsteady transonic flows over wings. Their approach coupled a quasi-steady integral boundary-layer scheme for viscous regions with an unsteady small disturbance potential equation for the inviscid flow domain. The viscous-inviscid coupling was performed using a modified geometry based on the displacement thickness concept. Although the viscous and inviscid flowfields were not strongly coupled in this approach, their results were impressive in that the overall computation time was increased only 6 percent through the addition of the viscous terms.

The desire to analyze complex unsteady flowfields has prompted many investigators to utilize proven steady flow algorithms adapted for the solution of similar unsteady problems. Many steady flow algorithms utilize an unsteady or iterative form of the governing equations as a means to achieve the desired steady state, and therefore there is a natural extension to the unsteady flow problem.

Many publications have recently become available which deal with the use of the full Reynolds-averaged Navier-Stokes equations for unsteady viscous flow calculations. Weinberg et al. [47] presented a complete unsteady solution scheme for three-dimensional flows using the linearized block implicit technique, but only reported results for a flat plate geometry. Newsome [48] successfully calculated the self-sustaining inlet buzz phenomena associated with supersonic compression inlets operating in the subcritical flow regime using the Navier-Stokes equations and Mac-

Cormack's explicit algorithm. Osswald et al. [49] calculated the unsteady flow over a rearward-facing step resulting from the periodic shedding of vortices using the incompressible Navier-Stokes equations and a time-marching technique. Ramakrishnan and Rubin [50]-[52], detailed their use of the unsteady form of the reduced Navier-Stokes (RNS) equations for numerical prediction of unsteady flows past flat plates and airfoils, although there was some indication that their algorithm was troubled by instabilities. Perhaps the most comprehensive unsteady flow calculation resulting from moving geometries was reported by Rai [3]. Using patched and overlaid grids, Rai simulated the unsteady flow resulting from the relative motion of blade rows in an ideal turbine stage. Unfortunately, the enormous computation times reflected the complexity of his analysis.

Other unsteady flow predictions of interest using the Navier-Stokes equations include: the vortex shedding predictions given in Davis et al. [53] and Gustafson and Leben [54], three-dimensional viscous flow predictions over an oscillating wing given by Simpson and Whitfield [55], and the calculation of the aerodynamics about multiple bodies in relative motion given by Meakin and Suhs [56].

Although a number of these efforts have successfully demonstrated agreement with available unsteady data, most fail to address the fundamental questions concerning the application of numerical techniques for unsteady flow analysis and show no comparisons with other calculation schemes. Considering the possible time inconsistencies of ADI schemes, and the instabilities reported in other schemes, it is necessary to identify algorithms which yield reliable time-accurate solutions rather than simply obtaining an unsteady solution using a known steady flow algorithm.

Lerat [57] published a detailed study of noniterative implicit time-marching schemes with second order time accuracy. The results of such in-depth studies are likely to be beneficial for both steady and unsteady flow computations.

Scope of the Present Research

In this study, the numerical aspects of the general unsteady compressible viscous flow problem are addressed. A primary element of this research is the identification and development of efficient time-marching solution algorithms intended specifically for the analysis of unsteady viscous flowfields. Inspired by the increased interest in understanding the unsteady aerodynamics of practical flow systems demonstrated by both government agencies and industry, this research may aid in the eventual development of a general purpose unsteady viscous flow calculation algorithm.

A multitude of algorithms exist to implement a numerical solution of the Navier-Stokes equations. It is not the primary purpose of this research to add yet another scheme to this already lengthy list; however, there is certainly some value to be gained by examining new possibilities and untested combinations. An advantage of the upwind-differenced TVD schemes previously mentioned is that linearized implicit schemes based on this differencing technique are diagonally dominant, and may therefore be solved through relaxation, rather than direct elimination. Clearly, this offers a wide range of opportunities for previously untested solution techniques. In this study, an iterative, upwind-differenced TVD type scheme is developed based on a generalized conjugate gradient matrix solution technique. This scheme is based on an iterative updating of the implicit fluxes and completely eliminates the linearization

and factorization errors often associated with implicit schemes.

One important aspect of this research is to assess the utility of upwind-difference schemes versus central-difference schemes and the effect of various forms of numerical damping on time-marching flow predictions. As a preliminary test, the inviscid one-dimensional flow in a shock tube is used as a basis to compare 44 different time-marching algorithms in terms of shock-capturing ability, programming complexity, execution speed, and reliability. From the results of this one-dimensional study, four separate time-marching algorithms are developed for the solution of unsteady compressible viscous flows. To reduce development costs and complexity, only two-dimensional problems are computed, although some attention is directed towards extending each scheme to a fully three-dimensional analysis. The final analyses include the ability to accurately simulate laminar unsteady viscous flows involving complex internal flow geometries or planar cascaded geometries over a wide range of flow conditions up to and including the transonic regime.

To enhance the two-dimensional calculations, an advanced numerical grid generation procedure is developed for both internal duct flows and spatially-periodic planar cascade flows. This grid generation scheme allows direct control over the smoothness, orthogonality, and point density of the grid coordinates. This control is achieved through the use of a variational formulation combined with an algorithm which allows the surface boundary points to move to satisfy orthogonality, and a power law interpolation procedure.

The four time-marching algorithms are applied to a series of numerical experiments involving both steady and unsteady flows to determine the useful range and

accuracy of each scheme and to compare overall execution efficiency. As an initial test, predictions for the steady inviscid critical and supercritical flow over an isolated cylinder are compared to a previously published prediction for such a flow. This is followed by a comparison of the predictions for the steady inviscid flow in both a low stagger and a high turning turbine cascade geometry with available experimental data. Next, predictions for the steady laminar viscous flow through a high turning turbine cascade with a leading edge flow separation are compared to experimental data. In order to further verify the accuracy of the viscous calculations, predictions for the steady subsonic flow over a semi-infinite flat plate are compared to an analytical flat plate boundary-layer solution. Time-accurate viscous flow predictions are verified through comparison with the analytical solution of the flow resulting from a sinusoidally oscillating flat plate near an otherwise stagnant fluid. Finally, predictions from the four time-marching schemes are compared with experimental data and flow visualization for the complex vortical flow pattern resulting from the impulsively started motion of a circular cylinder.

In Chapter 2, the equations governing the unsteady two-dimensional flow of a nonradiating Newtonian perfect gas are developed in a form suitable for numerical solution. The equations are nondimensionalized, expressed in a generalized coordinate system, and then recast in the strong conservation law form to permit accurate shock-capturing. The mathematical system of equations is then closed by specifying proper boundary and initial conditions.

In Chapter 3, the details of the proposed numerical algorithms are discussed. A preliminary comparison of time-marching schemes for the prediction of an unsteady

one-dimensional inviscid shock tube flow problem is presented first. From this test, four time-marching algorithms are chosen for development for the solution of two-dimensional unsteady viscous compressible flow problems. The schemes include two explicit central-difference methods, an implicit central-difference approach based on approximate factorization, and a new implicit upwind-difference TVD-type scheme based on an iterative generalized conjugate gradient matrix reduction procedure. Separate sections are provided for the description of the two-dimensional grid generation procedure, and each of the time-marching algorithms.

In Chapter 4, numerical results from the four time-marching algorithms are presented for a variety of steady and unsteady compressible viscous and inviscid flows. For each case, a comparison of the predicted results with available reference data is given, as well as a comparison of the CPU time and convergence data for each run.

Chapter 5 summarizes the conclusions of the research presented in this report, and lists several recommended topics of future research in this area.

Description of the Geometries

Although unsteady flows of practical interest cover a wide range of geometries, only two configurations are considered in this study. The first, illustrated in Fig. 1.1 is a simple two-dimensional planar channel flow geometry. This geometry includes a number of devices including inlets, diffusers, nozzles, and ducts. This is perhaps the simplest type of two-dimensional geometry for the development of numerical schemes.

The second geometry considered in this study is the spatially periodic planar cascade arrangement shown in Fig. 1.2. The geometry is representative of a constant

TWO-DIMENSIONAL DUCT GEOMETRY

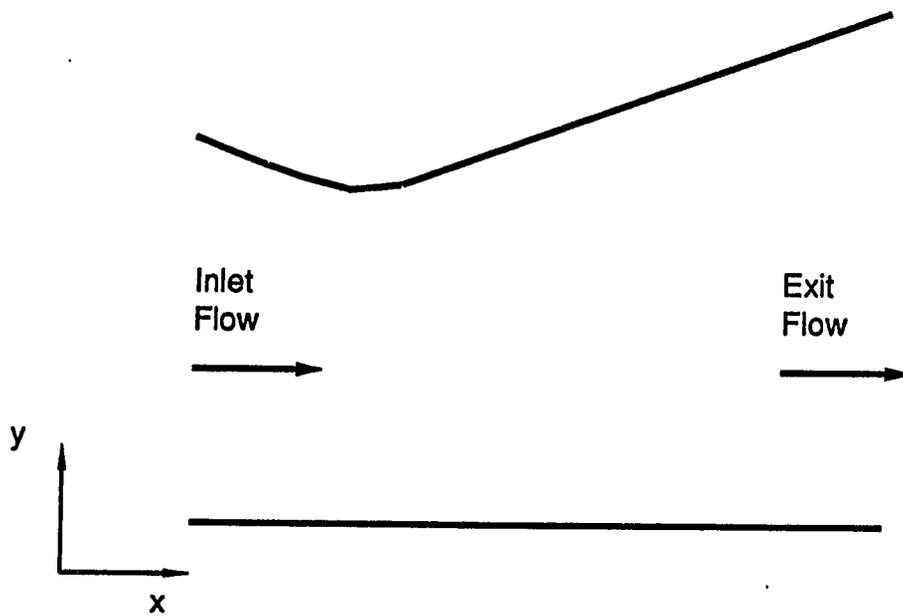


Figure 1.1: Two-dimensional planar channel flow geometry definition

radius cross section of a typical axial flow turbine blade row. This model is derived from the recent interest in the prediction of unsteady flows in turbomachines, and requires a more demanding numerical grid structure than the simple channel flow geometry.

The computer programs developed for the prediction of unsteady flows in this study are designed to handle either type of geometry through minor changes in the input.

TWO-DIMENSIONAL CASCADE GEOMETRY

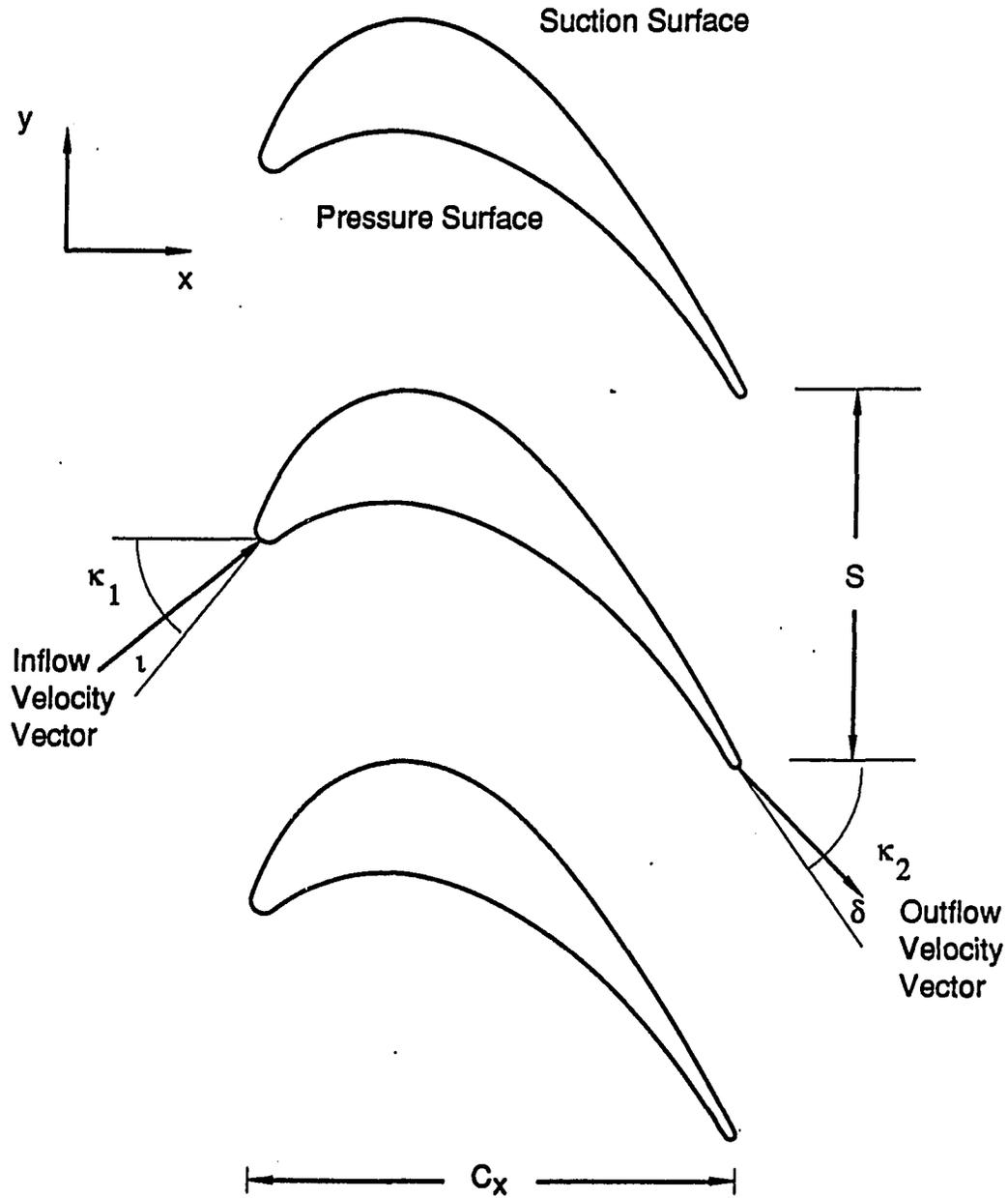


Figure 1.2: Spatially periodic planar cascade geometry definition

CHAPTER 2. MATHEMATICAL DEVELOPMENT

In this chapter, the equations governing the unsteady flow of a viscous, compressible, nonradiating, Newtonian fluid obeying Fourier's law of heat conduction are developed in a form suitable for numerical solution. The mathematical formulation utilizes the fully conservative form of the unsteady Navier-Stokes equations (e.g., see [58]) in two spatial dimensions. The conservative formulation allows flow discontinuities to be automatically captured as weak solutions of the partial differential equations without special treatment or fitting. The resulting coupled set of partial differential equations are mixed hyperbolic-parabolic in time, and are thus suitable for a time-marching solution without additional consideration for either subsonic or supersonic flows. To simplify the numerical calculations, the equations are written in an arbitrary coordinate system to allow for uneven grid spacing in the physical plane, while utilizing a uniform grid spacing in the computational plane.

The first section to follow describes a nondimensionalization procedure used to simplify the numerical solution. The Navier-Stokes equations in Cartesian coordinates for an ideal gas are then developed and expressed in strong conservation law form. Next, a general coordinate transformation is applied to the governing equations to simplify the numerical solution for uneven grid spacing. The resulting equations are then recast in the strong conservation law form. Finally, the system of equations is

closed by specifying proper initial and boundary conditions.

Nondimensionalization

It is a common practice to use nondimensionalized equations as the basis for numerical approximations. The advantage to this strategy is that it eliminates the need for many dimensional conversion factors within the program, and if the proper reference values are prescribed, it ensures that all variables have numerical values within a specific range (i.e., 0.0-1.0).

For convenience, the following nondimensional variables are defined:

$$\begin{aligned}
 x &= \frac{\bar{x}}{L_{ref}} & y &= \frac{\bar{y}}{L_{ref}} & t &= \frac{\bar{t}}{(L_{ref}/u_{ref})} & \rho &= \frac{\bar{\rho}}{\rho_{ref}} & u &= \frac{\bar{u}}{u_{ref}} \\
 v &= \frac{\bar{v}}{u_{ref}} & p &= \frac{\bar{p}}{\rho_{ref}u_{ref}^2} & E_t &= \frac{\bar{E}_t}{\rho_{ref}u_{ref}^2} & T &= \frac{\bar{T}}{(u_{ref}^2/R_{ref})} \\
 R &= \frac{\bar{R}}{R_{ref}} & c_p &= \frac{\bar{c}_p}{R_{ref}} & c_v &= \frac{\bar{c}_v}{R_{ref}} & k &= \frac{\bar{k}}{k_{ref}} & \mu &= \frac{\bar{\mu}}{\mu_{ref}} \\
 f &= \frac{\bar{f}}{(u_{ref}^2/L_{ref})} & H_t &= \frac{\bar{H}_t}{\rho_{ref}u_{ref}^2} \\
 Pr_{ref} &= \frac{R_{ref}\mu_{ref}}{k_{ref}} & Re_{ref} &= \frac{\rho_{ref}u_{ref}L_{ref}}{\mu_{ref}} & & & & & & (2.1)
 \end{aligned}$$

The definition of each of these variables may be found in the Notation (page xix). All fluid reference terms are determined by the inlet or freestream conditions for each case. The reference length, L_{ref} , is chosen as the larger of the maximum x or y dimension of the grid system for each calculation.

Navier-Stokes Equations

In order to derive the equations governing a general unsteady viscous flow, some assumptions about the nature of the fluid in question must be made. The first assumption is that the fluid acts essentially as a continuum. This imposes limits on the range of density for which the equations are valid. For most practical flows of engineering interest, this is not a concern, unless operation occurs at extremely high altitudes, or extremely low pressures. A second assumption commonly made is that the fluid is Newtonian and therefore the fluid shear stress is proportional to the element strain rate and the coefficient of viscosity (e.g., see [59]). A somewhat similar assumption is made for conduction heat transfer through the fluid by assuming that Fourier's law is applicable. This states that the local heat flux due to conduction, q , is proportional to the local gradient of temperature and the coefficient of thermal conductivity:

$$\vec{q} = -k\vec{\nabla}T \quad (2.2)$$

These simplifications are valid for a wide variety of common fluids. For the purposes of this study, radiation heat transfer and bouyancy forces are simply neglected. It is also convenient to limit the analysis to perfect fluids, indicating that the internal energy and enthalpy are functions of specific heats and temperature as:

$$e = c_v T \quad h = c_p T \quad (2.3)$$

For gases, fluid properties can then also be related through the ideal gas equation of state:

$$p = \rho RT \quad (2.4)$$

If the fluid temperature does not vary drastically, it is usually adequate to use constant values of the fluid properties. However, should the temperature vary significantly, the resulting variation in fluid properties is important and it is then required to provide specific temperature-based relationships to account for the property change. In this case, a constant Prandtl number is always assumed. The local thermal conductivity may then be determined from the constant Prandtl number:

$$k = \frac{c_p \mu}{Pr} \quad (2.5)$$

For all of the flow cases examined in this study, the working fluid is considered to be air, and unless otherwise noted, the following constant values of the fluid properties are used:

$$\begin{aligned} c_p &= 1005.0 \left(\frac{Nm}{kgK} \right) & c_v &= 717.857 \left(\frac{Nm}{kgK} \right) & \gamma &= 1.4 \\ \mu &= 1.6 * 10^{-5} \left(\frac{Ns}{m^2} \right) & R &= 287.143 \left(\frac{Nm}{kgK} \right) & Pr &= 0.7 \end{aligned} \quad (2.6)$$

Based on the above restrictions, the equations governing the two-dimensional unsteady flow of a viscous compressible ideal Newtonian nonradiating fluid obeying Fourier's law of heat conduction can be expressed nondimensionally in Cartesian coordinates as:

$$\frac{\partial Q}{\partial t} + \frac{\partial E_{inv}}{\partial x} + \frac{\partial F_{inv}}{\partial y} = S + \frac{\partial E_{vis}}{\partial x} + \frac{\partial F_{vis}}{\partial y} \quad (2.7)$$

where:

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E_t \end{bmatrix} \quad E_{inv} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E_t + p) \end{bmatrix} \quad F_{inv} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E_t + p) \end{bmatrix} \quad (2.8)$$

$$S = \begin{bmatrix} 0 \\ \rho f_x \\ \rho f_y \\ \rho(u f_x + v f_y) \end{bmatrix} \quad E_{vis} = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - q_x \end{bmatrix} \quad F_{vis} = \begin{bmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} - q_y \end{bmatrix} \quad (2.9)$$

and:

$$\begin{aligned} \tau_{xx} &= \frac{2}{3} \frac{\mu}{Re_{ref}} \left(2 \frac{\partial u}{\partial x} - \frac{\partial v}{\partial y} \right) \\ \tau_{xy} &= \frac{\mu}{Re_{ref}} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \\ \tau_{yy} &= \frac{2}{3} \frac{\mu}{Re_{ref}} \left(2 \frac{\partial v}{\partial y} - \frac{\partial u}{\partial x} \right) \end{aligned} \quad (2.10)$$

$$q_x = \frac{-k}{Pr_{ref} Re_{ref}} \frac{\partial T}{\partial x} \quad q_y = \frac{-k}{Pr_{ref} Re_{ref}} \frac{\partial T}{\partial y} \quad (2.11)$$

$$p = \rho RT \quad a = \sqrt{\gamma p / \rho} \quad E_t = \rho \left(c_v T + \frac{(u^2 + v^2)}{2} \right) \quad (2.12)$$

The terms f_x , and f_y represent external forces in the x and y directions, respectively. For the cases examined in this study, these terms were assumed to be zero. Heat sources and chemical reaction processes have also been neglected.

Equations (2.7-2.11) mathematically represent the principles of conservation of mass, momentum, and energy applied to a differential control volume element. The equations are written in the so-called strong conservation law form. This form is required for proper shock capturing in a time-marching calculation. Further discussion on this topic is given in a later section.

The individual flux derivatives have been separated into viscous and inviscid components in Eq. (2.8) since it can be shown that the limit behavior of each term can be

most effectively represented by different differencing techniques. In the inviscid limit, the character of the equations is strictly hyperbolic, and upwind spatial differencing is most appropriate. In the limit where viscous forces dominate, the equations are parabolic, and a centered-type spatial differencing scheme is likely to be best.

Standard central differencing schemes typically neglect this behavior. For the nonoscillatory TVD scheme to be described, upwind differencing is used for the inviscid flux terms, while centered differences are used for the viscous flux terms, consistent with the nature of the equations in the inviscid and viscous limit, respectively.

Finally, a note is required on the differences in calculating laminar and turbulent flows. Laminar flows progress in a smooth orderly fashion, but can only occur at relatively low Reynolds numbers and under stable operating conditions. Turbulent flows are characterized by large scale random fluctuations in flow properties and are inherently unsteady phenomena, although the mean flow may appear steady. It is generally agreed that the Navier-Stokes equations form a complete model for both laminar and turbulent flows; however, due to the minute scale of turbulent activity compared to the gross geometric scale of most problems of interest, it is impractical in most cases to capture turbulent effects through the nature of the governing equations. Instead, the problem is normally modelled according to the scale of the geometry and the associated turbulence effects are introduced through an appropriate turbulence model and the use of the Reynolds-averaged form of the governing equations. In the present study, only laminar flows are considered, and the extension to calculate flows with turbulence would require the formulation of the Reynolds-averaged form of Equations (2.7-2.11) and the development of an appropriate turbulence model. This

exercise is left as a basis for future research.

Coordinate Transformation

In order to solve Equations (2.7-2.11) numerically it is generally necessary to use a variably spaced numerical grid. The equations can be differenced in their present form for variable grid spacing, but the resulting system is complicated and difficult to program. Instead, a coordinate transformation is applied to the governing equations, and the equations may be solved in a uniformly-spaced computational plane while enjoying the advantages of variable spacing in the physical domain.

Applying a general nonsingular coordinate transformation of the form:

$$t = t \quad \xi = \xi(x, y, t) \quad \eta = \eta(x, y, t) \quad (2.13)$$

and making note of the following identities:

$$\xi_t = -J(x_t y_\eta - x_\eta y_t) \quad \xi_x = J y_\eta \quad \xi_y = -J x_\eta \quad (2.14)$$

$$\eta_t = J(x_t y_\xi - x_\xi y_t) \quad \eta_x = -J y_\xi \quad \eta_y = J x_\xi \quad (2.15)$$

with:

$$J = \frac{1}{(x_\xi y_\eta - x_\eta y_\xi)} \quad (2.16)$$

the governing equations may be transformed through application of the chain rule:

$$\frac{\partial}{\partial x} = \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta} \quad (2.17)$$

to yield the following equations:

$$\frac{\partial Q}{\partial t} + \xi_t \frac{\partial Q}{\partial \xi} + \eta_t \frac{\partial Q}{\partial \eta} + \xi_x \frac{\partial E}{\partial \xi} + \eta_x \frac{\partial E}{\partial \eta} + \xi_y \frac{\partial F}{\partial \xi} + \eta_y \frac{\partial F}{\partial \eta} = S \quad (2.18)$$

Here $E = E_{inv} - E_{vis}$, and $F = F_{inv} - F_{vis}$.

Strong Conservation Law Form

Although Equation (2.18) is valid for the flows to be tested, it is not prudent to use the equation in this form. The application of the coordinate transformation described in the previous section has resulted in the loss of the desired strong conservation law form of the equations. The strong conservation law form of the equations is essential for proper shock-capturing in the numerical solution. In a recent study, Hindman [13] presents a detailed discussion of shock-capturing schemes and the use of the strong conservation law form of the governing equations. Hindman notes that a conservative numerical scheme using the strong conservation law form of the governing equations can capture shocks properly and will automatically satisfy the Rankine-Hugoniot shock jump conditions [13], although the jump conditions are satisfied to first order accuracy only. Fortunately, it is possible to recombine the transformed equation to recover the strong conservation law property. Following the procedure outlined by Vinokur [60], the strong conservation law form may be recovered by first dividing the equations by the Jacobian, J , and adding and subtracting like terms which may later be combined to form the conservative derivative terms. The resulting equation becomes:

$$\begin{aligned}
 & \frac{\partial(Q/J)}{\partial t} + \frac{\partial(Q\xi_t/J)}{\partial \xi} + \frac{\partial(Q\eta_t/J)}{\partial \eta} - Q\left[\frac{\partial(1/J)}{\partial t} + \frac{\partial(\xi_t/J)}{\partial \xi} + \frac{\partial(\eta_t/J)}{\partial \eta}\right] + \\
 & \frac{\partial(E\xi_x/J)}{\partial \xi} + \frac{\partial(E\eta_x/J)}{\partial \eta} - E\left[\frac{\partial(\xi_x/J)}{\partial \xi} + \frac{\partial(\eta_x/J)}{\partial \eta}\right] + \\
 & \frac{\partial(F\xi_y/J)}{\partial \xi} + \frac{\partial(F\eta_y/J)}{\partial \eta} - F\left[\frac{\partial(\xi_y/J)}{\partial \xi} + \frac{\partial(\eta_y/J)}{\partial \eta}\right] = S/J
 \end{aligned} \tag{2.19}$$

Fortunately, this equation can be simplified by noting the identities

$$\frac{\partial(1/J)}{\partial t} + \frac{\partial(\xi_t/J)}{\partial \xi} + \frac{\partial(\eta_t/J)}{\partial \eta} = 0 \quad (2.20)$$

$$\frac{\partial(\xi_x/J)}{\partial \xi} + \frac{\partial(\eta_x/J)}{\partial \eta} = 0 \quad (2.21)$$

$$\frac{\partial(\xi_y/J)}{\partial \xi} + \frac{\partial(\eta_y/J)}{\partial \eta} = 0 \quad (2.22)$$

The important consideration here is that the numerical approximation of the derivative terms $(\xi_{x,y}, \eta_{x,y}, J)$ used in the digital solution must also satisfy these identities. This is the so-called grid conservation law which must be maintained in order to avoid the spurious introduction of nonphysical source terms in the numerical solution. There are several approaches which can be taken to ensure that the numerical grid derivative terms satisfy the identities. One method is to discretize the governing equations in terms of a finite-volume integral. A second approach is to use the same differencing scheme for the metric derivatives as is used for the spatial derivatives in the governing equation. Each of these approaches work adequately for two-dimensional cases, and for second order central differencing, the approaches are virtually identical. In three dimensions, the problem becomes somewhat more complicated. A common practice in this case is to average metric terms across a grid cell interface.

Elimination of the grid identities from Equation (2.19) and collecting like derivative terms yields the following strong conservation law form:

$$\frac{\partial(Q/J)}{\partial t} + \frac{\partial(Q\xi_t/J + E\xi_x/J + F\xi_y/J)}{\partial \xi} + \frac{\partial(Q\eta_t/J + E\eta_x/J + F\eta_y/J)}{\partial \eta} = S/J \quad (2.23)$$

By redefining a new set of flux terms, and substituting metric identities, the governing equation to be used in the two-dimensional numerical calculations may be expressed as:

$$\frac{\partial \hat{Q}}{\partial t} + \frac{\partial \hat{E}_{inv}}{\partial \xi} + \frac{\partial \hat{F}_{inv}}{\partial \eta} = \hat{S} + \frac{\partial \hat{E}_{vis}}{\partial \xi} + \frac{\partial \hat{F}_{vis}}{\partial \eta} \quad (2.24)$$

where now:

$$\hat{Q} = \frac{Q}{J} \quad \hat{E}_{inv} = \frac{\xi_t Q}{J} + \frac{\xi_x E_{inv}}{J} + \frac{\xi_y F_{inv}}{J} \quad \hat{F}_{inv} = \frac{\eta_t Q}{J} + \frac{\eta_x E_{inv}}{J} + \frac{\eta_y F_{inv}}{J} \quad (2.25)$$

$$\hat{S} = \frac{S}{J} \quad \hat{E}_{vis} = \frac{\xi_x E_{vis}}{J} + \frac{\xi_y F_{vis}}{J} \quad \hat{F}_{vis} = \frac{\eta_x E_{vis}}{J} + \frac{\eta_y F_{vis}}{J} \quad (2.26)$$

It should be noted that the mathematical nature of the equations is unchanged by the coordinate transformation [58]. The corresponding viscous shear stress (τ) and heat conduction (q) terms are transformed using a simple application of the chain rule.

$$\tau_{xx} = \frac{2}{3} \frac{\mu J}{Re_{ref}} \left[2(y_\eta \frac{\partial u}{\partial \xi} - y_\xi \frac{\partial u}{\partial \eta}) - (-x_\eta \frac{\partial v}{\partial \xi} - x_\xi \frac{\partial v}{\partial \eta}) \right] \quad (2.27)$$

$$\tau_{xy} = \frac{\mu J}{Re_{ref}} \left[-x_\eta \frac{\partial u}{\partial \xi} + x_\xi \frac{\partial u}{\partial \eta} + y_\eta \frac{\partial v}{\partial \xi} - y_\xi \frac{\partial v}{\partial \eta} \right] \quad (2.28)$$

$$\tau_{yy} = \frac{2}{3} \frac{\mu J}{Re_{ref}} \left[2(-x_\eta \frac{\partial v}{\partial \xi} + x_\xi \frac{\partial v}{\partial \eta}) - (y_\eta \frac{\partial u}{\partial \xi} - y_\xi \frac{\partial u}{\partial \eta}) \right] \quad (2.29)$$

$$q_x = \frac{-kJ}{Re_{ref} Pr_{ref}} \left[y_\eta \frac{\partial T}{\partial \xi} - y_\xi \frac{\partial T}{\partial \eta} \right] \quad (2.30)$$

$$q_y = \frac{-kJ}{Re_{ref} Pr_{ref}} \left[-x_\eta \frac{\partial T}{\partial \xi} + x_\xi \frac{\partial T}{\partial \eta} \right] \quad (2.31)$$

For the implicit schemes to be described, it is also necessary to describe the inviscid flux Jacobian matrices:

$$\hat{A} = \frac{\partial \hat{E}_{inv}}{\partial \hat{Q}} = \hat{R}_\xi \hat{\Lambda}_\xi \hat{R}_\xi^{-1} \quad \hat{B} = \frac{\partial \hat{F}_{inv}}{\partial \hat{Q}} = \hat{R}_\eta \hat{\Lambda}_\eta \hat{R}_\eta^{-1} \quad (2.32)$$

where:

$$\hat{A} = \begin{bmatrix} \xi_t & \xi_x & \xi_y & 0 \\ -uU_\xi + \xi_x\beta\alpha & U_\xi - (\beta - 1)\xi_x u & -\xi_y u - \beta\xi_x v & \beta\xi_x \\ -vU_\xi + \xi_y\beta\alpha & -\beta\xi_y u + \xi_x v & U_\xi - (\beta - 1)\xi_y v & \beta\xi_y \\ U_\xi[\alpha(\beta - 1) - a^2/\beta] & -\beta U_\xi u + \Omega\xi_x & -\beta U_\xi v + \Omega\xi_y & (\beta + 1)U_\xi \end{bmatrix} \quad (2.33)$$

$$\hat{R}_\xi = \begin{bmatrix} 1 & 0 & \rho/(a\sqrt{2}) & \rho/(a\sqrt{2}) \\ u & \rho\xi_y/\Phi & \rho(u/a + \xi_x/\Phi)/\sqrt{2} & \rho(u/a - \xi_x/\Phi)/\sqrt{2} \\ v & -\rho\xi_x/\Phi & \rho(v/a + \xi_y/\Phi)/\sqrt{2} & \rho(v/a - \xi_y/\Phi)/\sqrt{2} \\ \alpha & \rho(u\xi_y - v\xi_x)/\Phi & \rho(\alpha/a + U_\xi/\Phi + a/\beta)/\sqrt{2} & \rho(\alpha/a - U_\xi/\Phi + a/\beta)/\sqrt{2} \end{bmatrix} \quad (2.34)$$

$$\hat{R}_\xi^{-1} = \begin{bmatrix} 1.0 - (\alpha\beta/a^2) & \beta u/a^2 & \beta v/a^2 & -\beta/a^2 \\ (-u\xi_y + v\xi_x)/(\rho\Phi) & \xi_y/\rho\Phi & -\xi_x/\rho\Phi & 0 \\ (\alpha\beta/a - U_\xi/\Phi)/\zeta & (\xi_x/\Phi - \beta u/a)/\zeta & (\xi_y/\Phi - \beta v/a)/\zeta & \beta/\zeta a \\ (\alpha\beta/a + U_\xi/\Phi)/\zeta & -(\xi_x/\Phi + \beta u/a)/\zeta & -(\xi_y/\Phi + \beta v/a)/\zeta & \beta/\zeta a \end{bmatrix} \quad (2.35)$$

$$\hat{\Lambda}_\xi = \begin{bmatrix} U_\xi & 0 & 0 & 0 \\ 0 & U_\xi & 0 & 0 \\ 0 & 0 & U_\xi + a\Phi & 0 \\ 0 & 0 & 0 & U_\xi - a\Phi \end{bmatrix} \quad \Omega = \frac{a^2}{\beta} + \alpha \quad \zeta = \rho\sqrt{2} \quad (2.36)$$

where:

$$\alpha = \frac{1}{2}(u^2 + v^2) \quad \beta = \gamma - 1.0 \quad U_\xi = \xi_t + \xi_x u + \xi_y v \quad \Phi = \sqrt{\xi_x^2 + \xi_y^2} \quad (2.37)$$

The expressions for \hat{B} , \hat{R}_η^{-1} , $\hat{\lambda}_\eta$, and \hat{R}_η are determined by replacing ξ with η in Equations (2.33-2.37), respectively.

Boundary Conditions

In order to complete the system of equations governing the unsteady flows of interest, a proper set of boundary conditions must be supplied. Since the equations are to be marched in time, initial conditions must be known for all (x, y) at some initial time $t = 0$. Expressed mathematically this becomes:

$$(\rho, u, v, p, T)_{t=0} = f(x, y) \quad (2.38)$$

For steady and unsteady inviscid flows, solid surfaces must satisfy the condition of flow tangency, i.e.:

$$\vec{V} \cdot \vec{n}_{surf} = 0 \quad (2.39)$$

where \vec{V} is the velocity vector and \vec{n}_{surf} is the local surface outward unit normal vector. If the surface is aligned with a grid line of constant η value, this condition becomes simply:

$$U_\eta = 0 \quad (2.40)$$

which implies no flow through the surface.

For steady inviscid flows, a pressure condition may also be applied as:

$$\frac{\partial p}{\partial n} = \frac{-\rho V_{tan}^2}{R_{surf}} \quad (2.41)$$

where V_{tan} is the velocity tangent to the surface, and R_{surf} is the local radius of curvature at the surface. For steady inviscid flows with a uniform inflow and no external energy sources, a constant total enthalpy may also be assumed throughout the flow.

For viscous flows, the no slip condition is applied on the surface as:

$$u_{surf}, v_{surf} = 0 \quad (2.42)$$

The following pressure condition is a reasonable assumption based on boundary-layer approximations:

$$\left(\frac{\partial p}{\partial n}\right)_{surf} = 0 \quad (2.43)$$

If an adiabatic surface is assumed, a similar condition exists for temperature as:

$$\left(\frac{\partial T}{\partial n}\right)_{surf} = 0 \quad (2.44)$$

Otherwise, a surface temperature is specified as:

$$T_{surf} = T(x, y)_{surf} \quad (2.45)$$

For cascade calculations, only a single blade element is calculated for any solution, and therefore periodic boundary conditions must be applied to account for the presence of the neighboring blades. The periodic condition for the lower periodic boundary may be expressed as:

$$(\rho, u, v, p, T)_{x, y} = (\rho, u, v, p, T)_{x, y+S} \quad (2.46)$$

and for the upper periodic boundary as:

$$(\rho, u, v, p, T)_{x, y} = (\rho, u, v, p, T)_{x, y-S} \quad (2.47)$$

where S is the cascade spacing.

For some unsteady flows of practical interest [5], it may also be necessary to phase-lag the values used at the periodic boundaries to account for periodic disturbances not compatible with the spatial periodicity of the cascade.

For all cases in this study, the component of the velocity normal to the grid boundary is assumed to be subsonic for both the inlet and the exit, although the overall flow may still be supersonic. From characteristic theory, this implies that three boundary conditions are required at the inlet, while one condition is needed at the exit. At the inlet, for steady flows, total temperature, total pressure, and flow angle are specified. For unsteady flows, the Riemann invariants traveling along downstream-running waves at the boundary are specified, along with a specified flow angle. At the exit, static pressure is specified for steady flows, and the Riemann invariant traveling along upstream-running waves is imposed for unsteady flows. More details on this treatment are given in the Appendix. For the case of supersonic normal inflow boundaries, all flow variables must be specified, while for supersonic normal outflow, all flow variables are extrapolated from the interior.

CHAPTER 3. NUMERICAL ALGORITHMS

In this chapter, several time-marching, finite-difference numerical algorithms based on the governing equations developed in the preceding chapter are described. The algorithms are formulated for solution on a digital computer using VS-FORTRAN programming techniques.

In the first section to follow, a survey of 44 different time-marching algorithms is presented for a one-dimensional inviscid shock tube problem as a means to identify the characteristics of various solution schemes. In the second section, four of these algorithms are extended for two-dimensional viscous and inviscid flow problems. Separate subsections are provided for the grid generation procedure and the four numerical algorithms developed for the two-dimensional calculations.

One-Dimensional Study

In order to assess the utility of upwind-differencing versus central-differencing for potential application to complex two-dimensional compressible viscous flows, the numerical solution of a one-dimensional shock tube flow was chosen as an initial basis for comparison. This is a common case for testing algorithms for systems of conservation laws, and other examples of this type of comparison are given in [61].

Consider a constant area, constant cross section tube of infinite extent in which

ONE-DIMENSIONAL SHOCK TUBE GEOMETRY DEFINITION

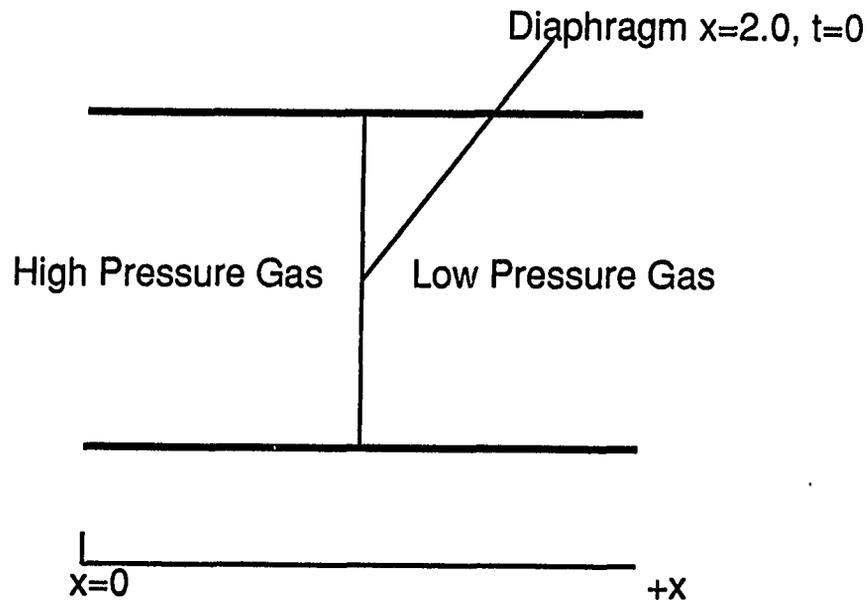


Figure 3.1: One-dimensional shock tube geometry definition

a diaphragm separates two gases at different pressures, but at a uniform temperature, as shown in Fig. 3.1. At time $t=0$, the diaphragm ruptures, allowing a shock wave, followed by a contact discontinuity, to propagate into the low pressure gas, while an expansion wave propagates into the high pressure gas, all in an approximately one-dimensional manner. Regions of uniform flow exist between each of the waves.

A graphical illustration of the development of such a flow is given in Fig. 3.2. If the specific heat ratio and gas constant of the two gases are identical, then the

Table 3.1: Uniform flow regions of one-dimensional shock tube flow

Region	Density Ratio ρ/ρ_1	Velocity Ratio u/a_1	Pressure Ratio p/p_1	Temperature Ratio T/T_1
1	1.0	0.0	1.0	1.0
2	0.40776	0.8212	0.2848	0.6985
3	0.20448	0.8212	0.2848	1.3931
4	0.1	0.0	0.1	1.0

solution is solely dependent on the initial pressure ratio across the diaphragm. This is the so-called Riemann problem, and analytical methods of solution are given in many texts [62].

For this test case, the initial pressure ratio across the diaphragm was chosen to be 10:1. The resulting properties of the uniform regions of flow indicated in Fig. 3.2 are given in Table 3.1.

The governing equations for the one-dimensional inviscid flow of an ideal gas may be expressed in Cartesian coordinates as:

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} = 0 \quad (3.1)$$

where:

$$Q = \begin{bmatrix} \rho \\ \rho u \\ E_t \end{bmatrix} \quad E = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(E_t + p) \end{bmatrix} \quad (3.2)$$

For implicit schemes, the Jacobian matrix A is required and can be derived as:

$$A = \frac{\partial E}{\partial Q} = \begin{bmatrix} 0 & 1 & 0 \\ u^2(\lambda - 3)/2 & -u(\lambda - 3) & (\lambda - 1) \\ (\lambda - 1)u^3 - \lambda E_t u/\rho & \lambda E_t/\rho - 3(\lambda - 1)u^2/2 & \lambda u \end{bmatrix} \quad (3.3)$$

Further, the A matrix may be decomposed into the diagonalized form:

$$A = R\Lambda R^{-1} \quad R = MT \quad R^{-1} = T^{-1}M^{-1} \quad (3.4)$$

ONE-DIMENSIONAL SHOCK TUBE FLOW DESCRIPTION

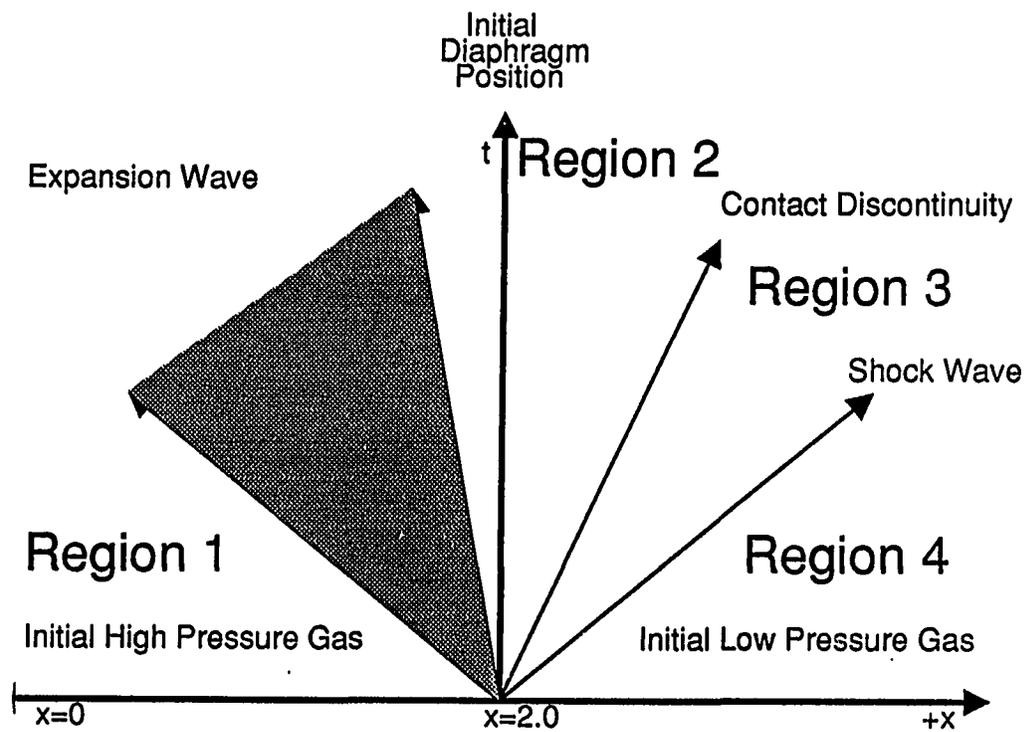


Figure 3.2: Wave propagation for one-dimensional shock tube flow problem

where:

$$M = \begin{bmatrix} 1 & 0 & 0 \\ u & \rho & 0 \\ u^2/2 & \rho u & 1/(\gamma - 1) \end{bmatrix} \quad T = \begin{bmatrix} 1 & \rho/(\sqrt{2}a) & \rho/(\sqrt{2}a) \\ 0 & 1/\sqrt{2} & -1/\sqrt{2} \\ 0 & \rho a/\sqrt{2} & \rho a/\sqrt{2} \end{bmatrix} \quad (3.5)$$

$$\Lambda = \begin{bmatrix} u & 0 & 0 \\ 0 & u + a & 0 \\ 0 & 0 & u - a \end{bmatrix} \quad (3.6)$$

$$M^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -u/\rho & 1/\rho & 0 \\ (\gamma - 1)u^2/2 & -(\gamma - 1)u & (\gamma - 1) \end{bmatrix} \quad T^{-1} = \begin{bmatrix} 1 & 0 & -1/a^2 \\ 0 & 1/\sqrt{2} & 1/(\sqrt{2}\rho a) \\ 0 & -1/\sqrt{2} & 1/(\sqrt{2}\rho a) \end{bmatrix} \quad (3.7)$$

A general computer code was written to predict the one-dimensional inviscid shock tube flow described above. The code was designed such that several algorithms could be easily implemented and compared for both accuracy and execution time. Due to the simplicity of the problem, calculations were performed on an IBM-PC-AT personal computer. A total of 44 different algorithms have been programmed and executed for this test case, including a wide variety of both central-difference and upwind-difference schemes. A listing and description of the algorithms tested is given in Tables 3.2-3.3 at the end of this section.

All of the calculations were based on a constant value of the time step resulting in an average CFL number near 0.8. The numerical grid consisted of 92 equally spaced points over a region which was 4.5 meters long. The diaphragm was located at $\bar{x} = 2.0$ meters. Each calculation was run to a nondimensional time $t = \bar{t}/(\Delta \bar{x}/\bar{a}_{tot}) = 20.0$. Results are presented at this time level. Each scheme was also tested for a 46 point grid and multiple damping levels when applicable. In some cases, calculations at

several different CFL values were also compared. The enormous number of plots generated through this exhaustive study are too numerous to display here; therefore, only a few of the more interesting results will be presented.

Figures 3.3-3.8 compare the predicted density ratio with the exact analytical solution for six of the algorithms tested. The prediction on Fig. 3.3 was generated using the centrally-differenced explicit hopscotch algorithm with blended second and product fourth order damping as described by Delaney [63]. This scheme was very well-behaved and extremely rapid, although like most central-difference schemes, small oscillations developed near the discontinuities in spite of the added damping. Figure 3.4 displays results from a second-order accurate centrally-differenced Runge-Kutta [64] explicit time marching procedure. Again, the blended second and product fourth order dissipation terms were used. Figure 3.5 displays results from a centrally-differenced Runge-Kutta [64] explicit time marching procedure which was fourth order accurate in both time and space, with added fourth order dissipation terms. Although this scheme has a significantly lower truncation error, the calculation is not noticeably better than the theoretically less accurate hopscotch scheme for this grid and choice of time step. Figure 3.6 is the upwind-differenced explicit second order TVD scheme presented by Roe [18]. This result is typical of most flux difference split methods using Roe's approximate Riemann solver. The advantage of the TVD scheme is obvious, in that the predicted discontinuities are clearly sharper than the previous central-difference results, and are void of any nonphysical oscillatory behavior. Figure 3.7 is the result of a one-dimensional version of the implicit time-linearized central-difference scheme described by Beam and Warming [65] with

explicit fourth order dissipation. This is a popular implicit central-difference scheme utilizing a complex nonlinear dissipation scheme of the form described by Pulliam [66]. Finally, Fig. 3.8 was generated by an implicit version of the explicit third order TVD scheme described by Chakravarthy [22]. This implicit algorithm utilizes a pointwise iteration at each time step to eliminate linearization errors (see e.g. [67]).

These results demonstrate the general solution characteristics of explicit, implicit, central-difference, and upwind-difference calculations for a simple unsteady flow problem. Several observations can be made from the predictions. All of the central-difference schemes produced oscillations near the discontinuities, in spite of the fact that the Runge-Kutta procedure in Fig. 3.5 was fully fourth order accurate in both time and space. In addition, the central-difference schemes smeared the discontinuities over several grid points. Clearly the best solution was obtained with the explicit TVD scheme in Fig. 3.6. No oscillations are evident, and the discontinuities are captured in a minimum number of points. Between the implicit schemes, the third order upwind TVD scheme of Fig. 3.8 is still better than the central-difference scheme of Fig. 3.7, although the implicit third order TVD scheme appears to be overly damped when compared to the explicit second order TVD solution. An analysis shows that for time-dependent flow calculations, the truncation error of the explicit TVD scheme is reduced by a fortuitous cancellation of terms which may not occur for the implicit scheme. It is likely that an implicit scheme will be required for detailed viscous flow calculations because of the severe time step limitation encountered by explicit schemes operating on the fine grids required to capture small scale viscous

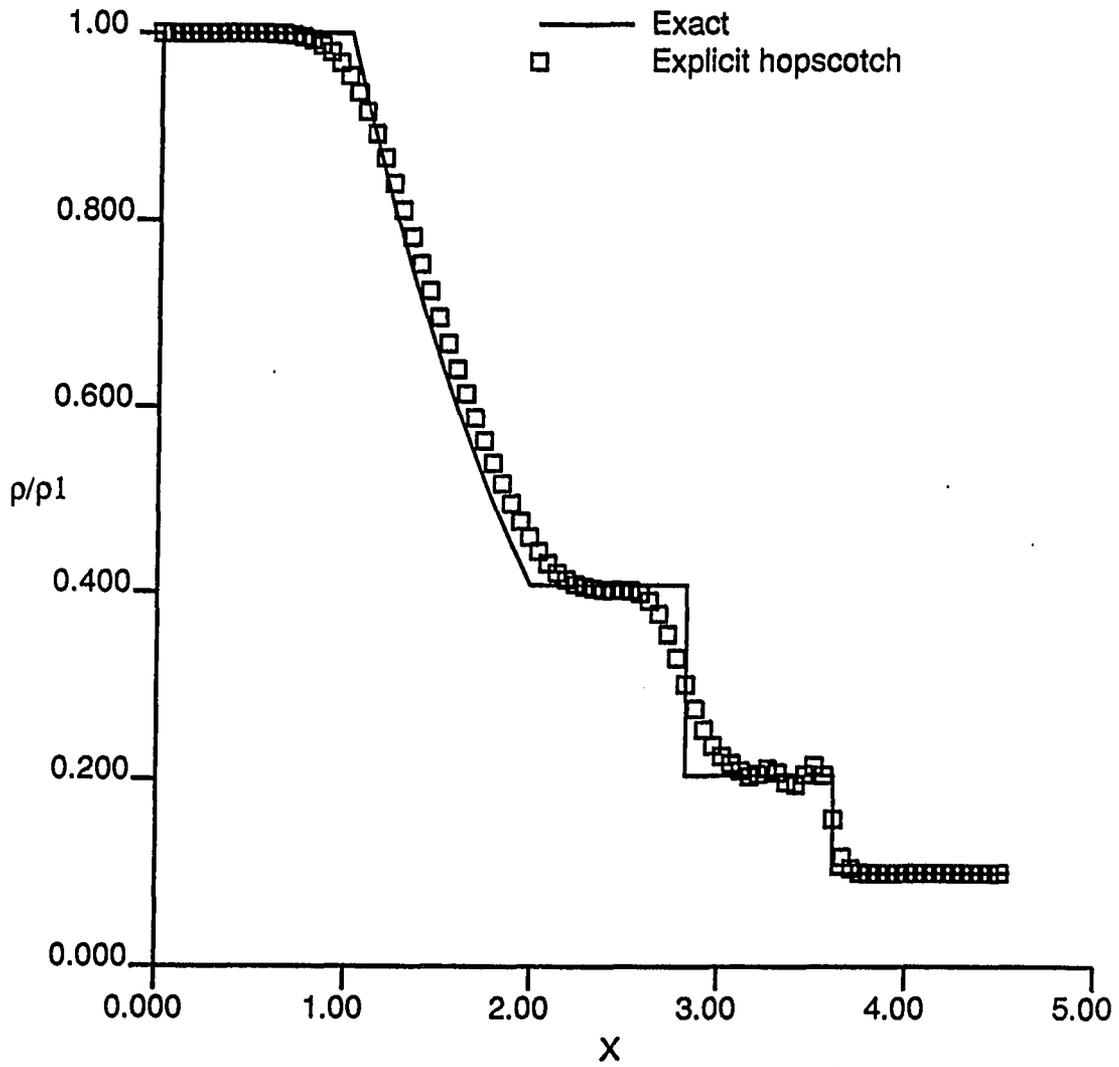
One-Dimensional Shock Tube Flow $P1/P4 = 10.0, t=20.0$ 

Figure 3.3: Comparison of the predicted hopscotch and analytical density ratio for one-dimensional shock tube problem

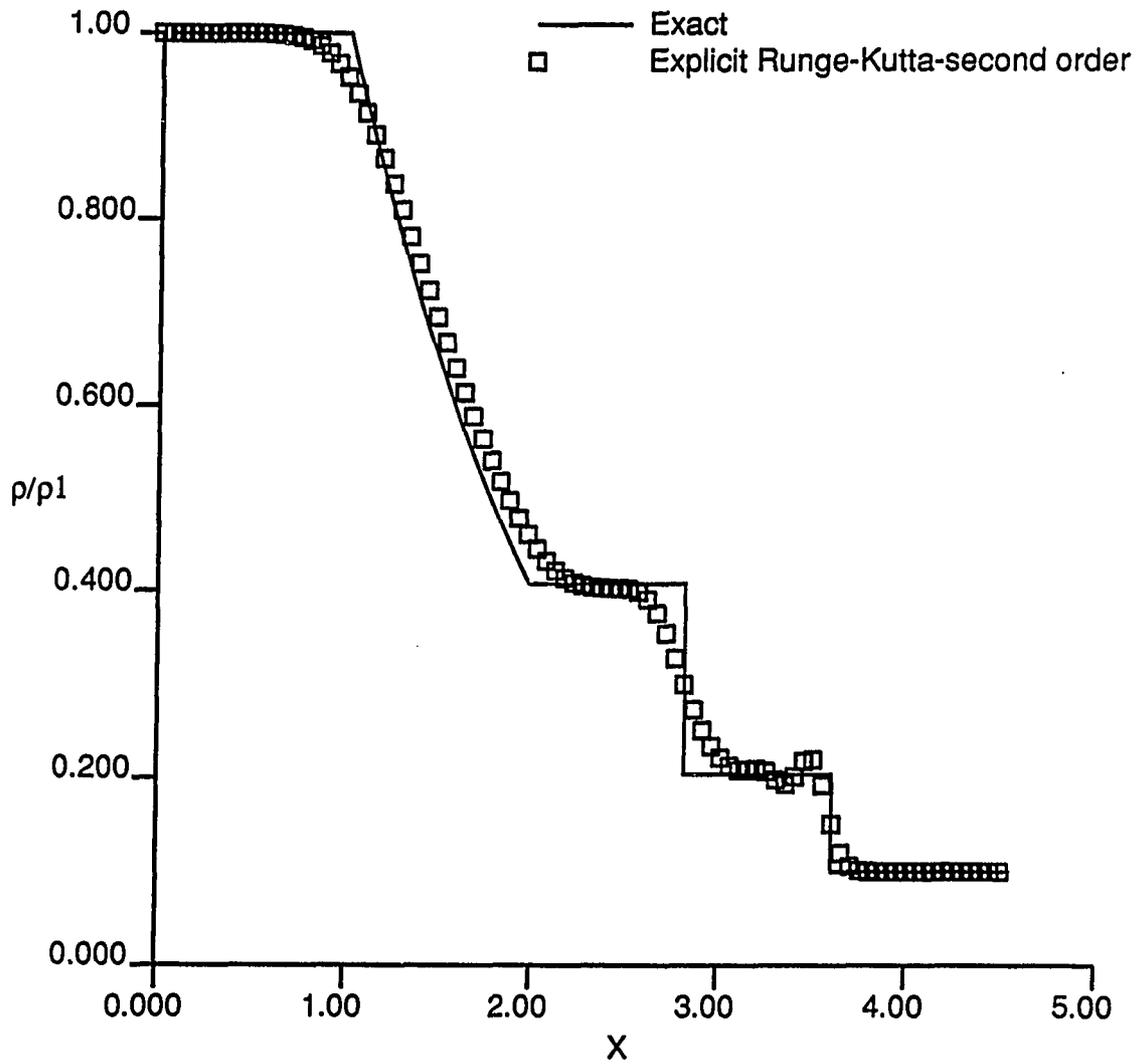
One-Dimensional Shock Tube Flow $P1/P4 = 10.0$, $t=20.0$ 

Figure 3.4: Comparison of the predicted second order Runge-Kutta and analytical density ratio for one-dimensional shock tube problem

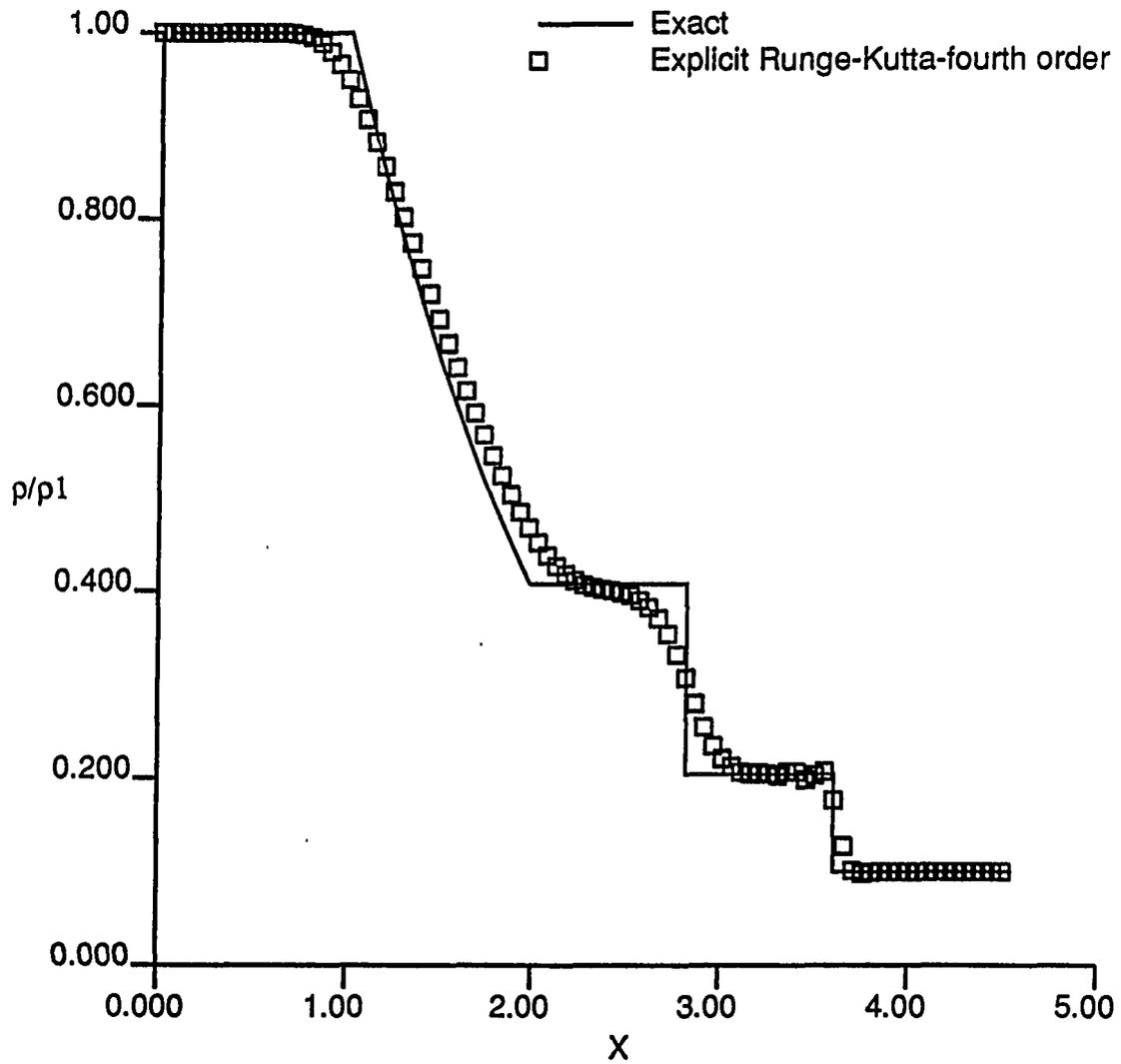
One-Dimensional Shock Tube Flow $P1/P4 = 10.0$, $t=20.0$ 

Figure 3.5: Comparison of the predicted fourth order Runge-Kutta and analytical density ratio for one-dimensional shock tube problem

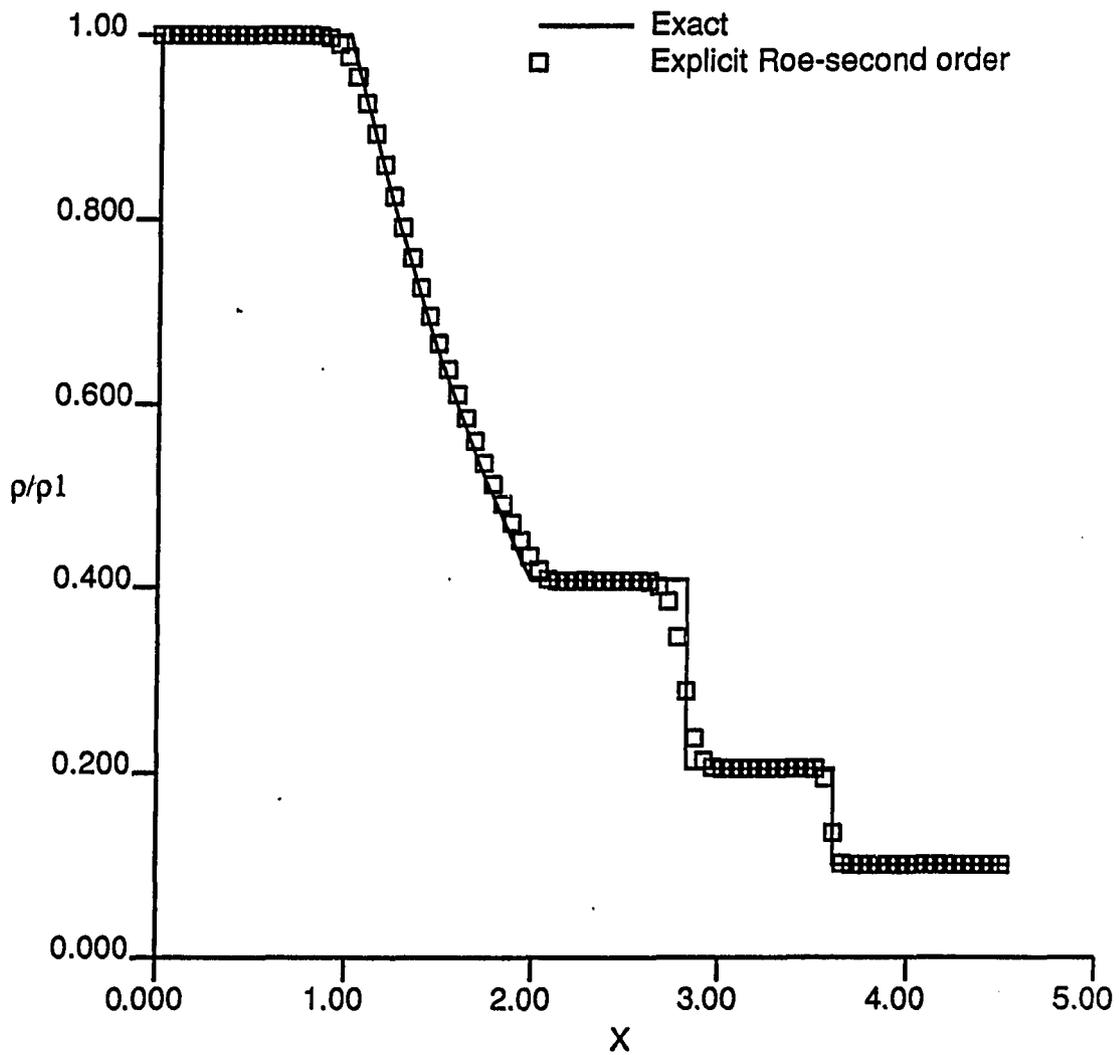
One-Dimensional Shock Tube Flow $P1/P4 = 10.0$, $t=20.0$ 

Figure 3.6: Comparison of the predicted explicit second order TVD and analytical density ratio for one-dimensional shock tube problem

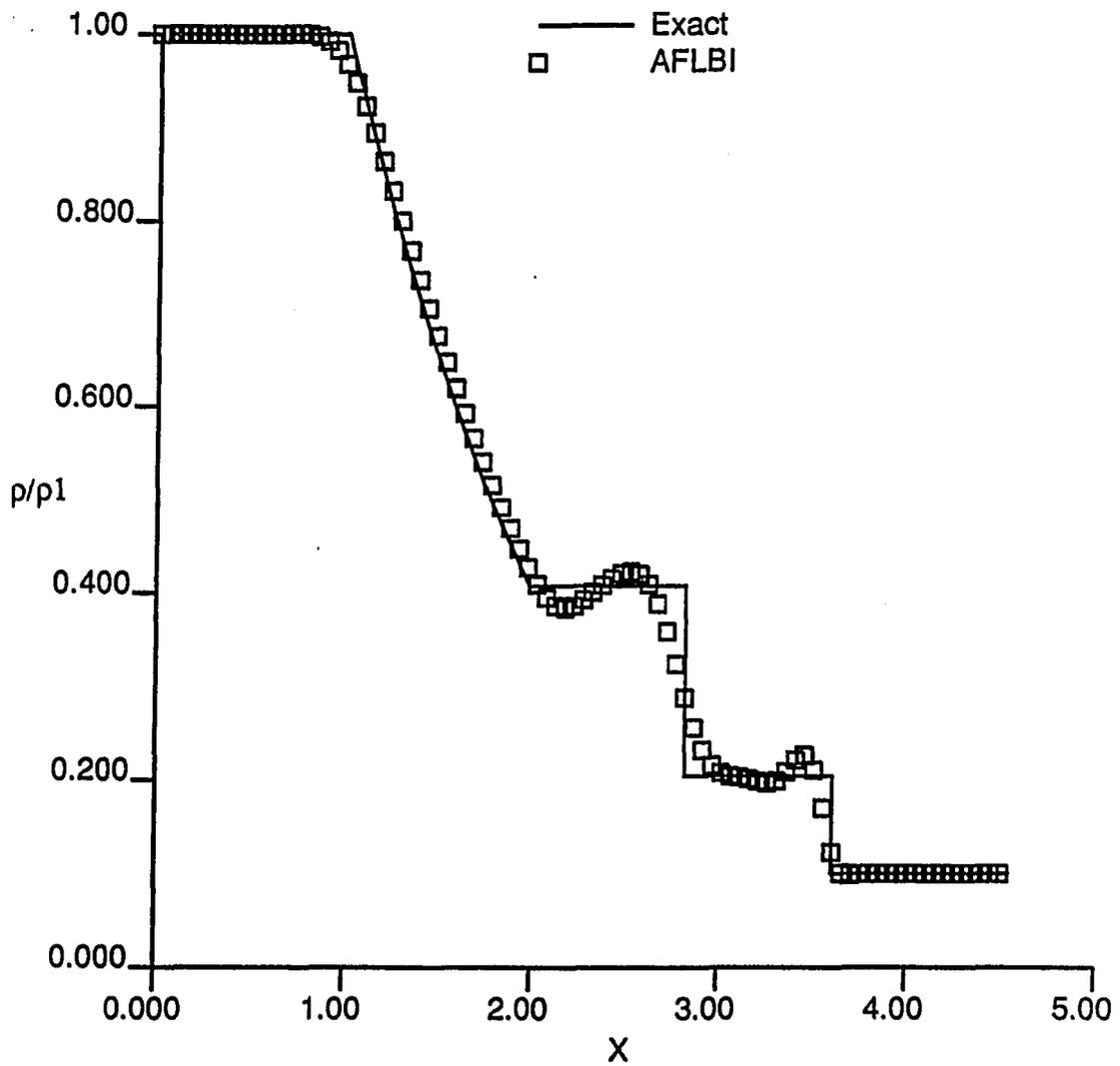
One-Dimensional Shock Tube Flow $P1/P4 = 10.0$, $t=20.0$ 

Figure 3.7: Comparison of the predicted linearized block implicit method with non-linear damping and analytical density ratio for one-dimensional shock tube problem

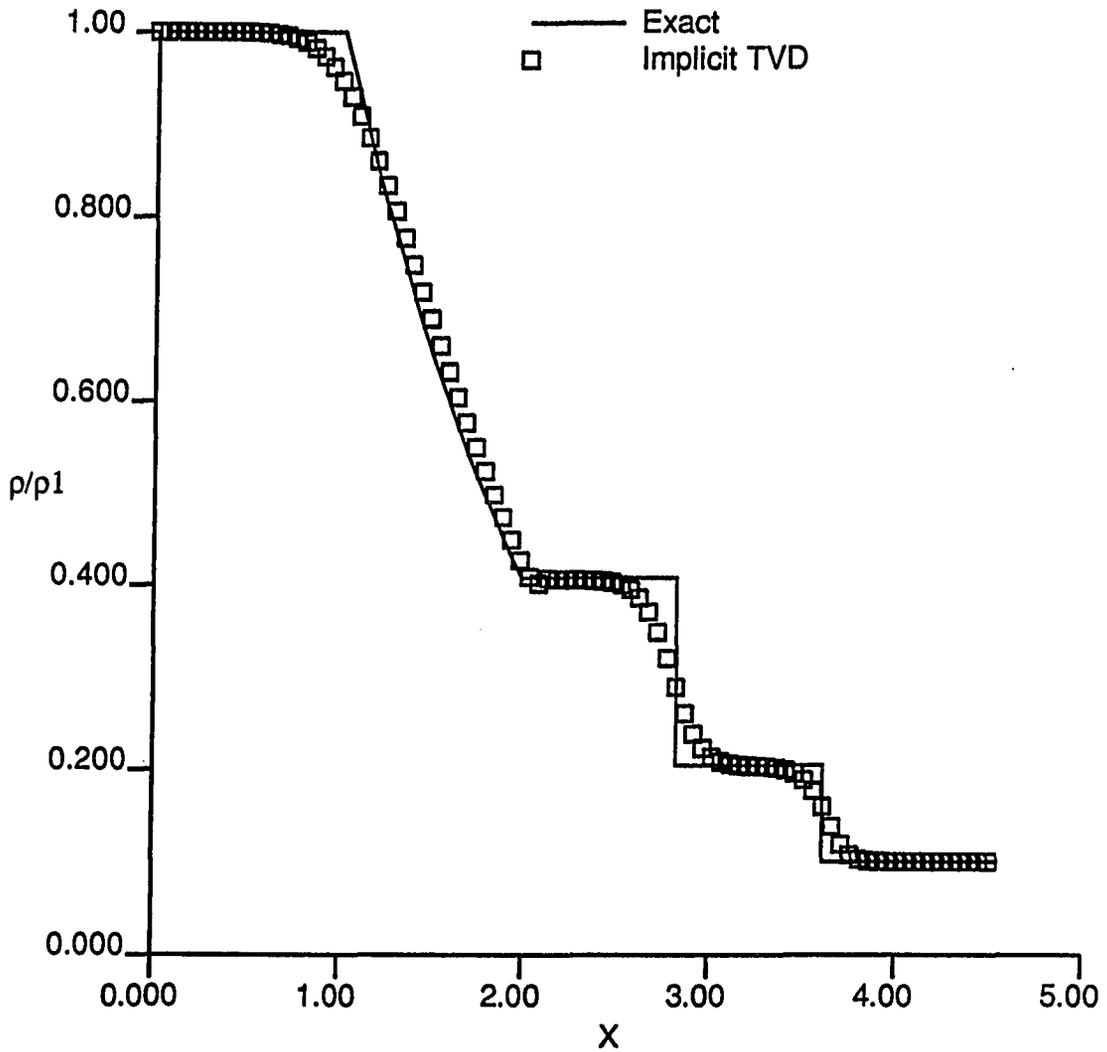
One-Dimensional Shock Tube Flow $P1/P4 = 10.0$, $t=20.0$ 

Figure 3.8: Comparison of the predicted implicit third order TVD and analytical density ratio for one-dimensional shock tube problem

flow phenomena.

It appears from the predicted results that even the most sophisticated explicitly added damping schemes are designed with only steady state calculations in mind. The one-dimensional unsteady predictions using complex (non-TVD) damping schemes (as in Fig. 3.7) were subject to overshoots and oscillations, while steady state solutions using the same schemes for a one-dimensional shocked nozzle flow were well behaved. This is likely to be an important consideration for complex unsteady flow calculations and it appears that some future development of intelligent damping schemes for unsteady flows is warranted.

The improved accuracy of the upwind difference schemes is not without cost. On the average, the scalar execution speed of the explicit upwind schemes were approximately 25% greater than the central difference schemes. In addition, the implicit TVD scheme with iteration at every time step required up to 2.5 times the execution time required for a similar central-difference scheme.

Table 3.2: Explicit algorithms for shock tube flow study

Algorithm	Type of Damping	Type of Difference	Time Acc.	Spatial Acc.	Ref.
Explicit hopscotch	2nd+prod 4th	central	1st	2nd	[63]
Explicit Lax	none	central	1st	2nd	[58]
Explicit Lax-Wendroff	none	central	1st	2nd	[58]
Explicit MacCormack	none	central	1st	2nd	[58]
Explicit Runge-Kutta	2nd+prod 4th	central	4th	2nd	[64]
Explicit Runge-Kutta	linear 4th	central	4th	4th	[64]
Explicit Van Leer	flux split	upwind	1st	1st	[17]
Explicit Van Leer	flux split	central	1st	2nd	[17]
Explicit Van Leer	flux split	upwind	1st	2nd	[17]
Explicit Van Leer	flux split	up-bias	1st	3rd	[17]
Explicit Chakravarthy	TVD ^a	upwind	1st	2nd	[22]
Explicit Chakravarthy	TVD	up-bias	1st	3rd	[22]
Explicit Chakravarthy	S limiter ^b	up-bias	1st	3rd	[22]
Explicit Harten	TVD	upwind	1st	1st	[68]
Explicit Harten	TVD	upwind	1st	2nd	[68]
Explicit Osher	TVD	upwind	1st	1st	[20]
Explicit Osher	TVD	upwind	1st	2nd	[20]
Explicit Roe	TVD	upwind	1st	1st	[18]
Explicit Roe	TVD	upwind	1st	2nd	[18]
Explicit Nonoscillatory	UNO ^c	upwind	1st	2nd	[69]

^aTVD scheme using min-mod flux limiter [70].

^bTVD scheme with Van Leers' S-type limiter [17].

^cHartens [69] UNO nonoscillatory flux construction.

Two-Dimensional Time-Marching Algorithms

The next stage of the research was to compare central-difference and upwind-difference algorithms for unsteady viscous flows in two dimensions for arbitrary geometries. For most of the schemes considered, the extension to multiple spatial dimensions is not particularly difficult. Typically, the one-dimensional differencing technique is simply duplicated for the added dimension to yield the appropriate finite-difference

Table 3.3: Implicit algorithms for shock tube flow study

Algorithm	Type of Damping	Type of Difference	Time Acc.	Spatial Acc.	Ref.
Implicit Beam-Warming	2nd+prod 4th	central	1st	2nd	[65]
Implicit Beam-Warming	2nd+prod 4th	cen-trap ^a	2nd	2nd	[65]
Implicit Beam-Warming	2nd+prod 4th	cen-3pt ^b	2nd	2nd	[65]
Implicit Beam-Warming	linear 4th	central	1st	2nd	[65]
Implicit Beam-Warming	linear 4th	cen-trap	2nd	2nd	[65]
Implicit Beam-Warming	linear 4th	cen-3pt	2nd	2nd	[65]
Implicit Beam-Warming	nonlinear	central	1st	2nd	[65]
Implicit Beam-Warming	nonlinear	cen-trap	2nd	2nd	[65]
Implicit Beam-Warming	nonlinear	cen-3pt	2nd	2nd	[65]
Implicit Steger-Warming	flux split	central	1st	2nd	[16]
Implicit Steger-Warming	flux split	cen-3pt	2nd	2nd	[16]
Implicit Steger-Warming	flux split	up-3pt	2nd	2nd	[16]
Implicit Steger-Warming	flux split	up-bias	2nd	2nd	[16]
Implicit Van Leer	flux split	central	1st	2nd	[71]
Implicit Van Leer	flux split	upwind	1st	2nd	[71]
Implicit Coakley	TVD ^c	upwind	1st	2nd	[72]
Implicit Harten LNI	TVD	upwind	1st	2nd	[70]
Implicit Harten	TVD	upwind	1st	2nd	[70]
Implicit Harten	TVD	upwind	1st	2nd	[70]
Relaxation ^d					[67]
Implicit Harten	TVD	upwind	1st	2nd	[70]
Relaxation w/compression ^e					[67]
Implicit Chakravarthy	TVD	upwind	1st	2nd	[22]
Relaxation					[67]
Implicit Chakravarthy	TVD	up-bias	1st	3rd	[22]
Relaxation					[67]
Implicit Chakravarthy	S limiter ^f	up-bias	1st	3rd	[22]
Relaxation					[67]
Implicit Chakravarthy	TVD	up-bias	2nd	3rd	[22]
Relaxation					[67]

^aCentral differences for spatial with trapezoidal time differencing.

^bCentral differences for spatial with three point time differencing.

^cTVD scheme using min-mod flux limiter [70].

^dIterative relaxation implicit scheme [67].

^eAdded compression to enhance linear (contact) discontinuities [70].

^fTVD scheme with Van Leers' S-type limiter [17].

equations. The manner in which these equations are solved, especially for implicit schemes, may require some special treatment such as approximate factorization of the coefficient matrix to yield an efficient algorithm. Advancing the upwind-differencing schemes to two dimensions is not completely straightforward, however. There is some concern that the one-dimensional nature of the approximate Riemann solver used in the determination of the cell-interface fluxes may limit the overall accuracy of the two-dimensional solution to first order; however, recent calculations by Chakravarthy for inviscid flow in two dimensions using the explicit TVD scheme demonstrated noticeable improvement over a central-difference formulation for flow over an isolated airfoil [22]. An additional concern is the proper treatment of the grid metric terms to maintain second order representation of grid derivatives, while simultaneously avoiding grid-induced sources of error in the approximate Riemann solver. At present it appears that this concern can best be satisfied by carefully formulating the difference equations through a finite volume analysis.

During the course of this study, several time-marching Navier-Stokes solution schemes have been developed for the prediction of two-dimensional internal flows on an H-type grid system and spatially periodic flows using a body-centered O-type grid system. Four of these methods were chosen as suitable for advanced calculations. The four methods studied extensively are the hopscotch method [63], a Runge-Kutta scheme [64], an approximately factored linearized block implicit (AFLBI) technique [65], and a new implicit generalized conjugate gradient total variation diminishing (GCGTVD) upwind differencing technique. A complete description is given for the grid generation and for each of the four time-marching schemes in the sections that

follow. Additional details on the actual numerical implementation for each scheme may be found in the references cited. For all of the algorithms to be described, a uniform grid spacing in the computational plane is assumed, (i.e., $\xi = (i - 1)\Delta\xi$ and $\eta = (j - 1)\Delta\eta$), and $\Delta t = t^{n+1} - t^n$.

Grid Generation

To successfully simulate the flow in complex geometries involved in many aerodynamic applications, adequate spatial resolution is essential. For the channel flows in this study, a simple H-type grid system as shown in Fig. 3.9 is adequate. However, grid resolution is especially crucial near the airfoil leading and trailing edge regions in the turbomachinery geometries considered later in this study. To achieve adequate grid resolution for two-dimensional cascaded airfoil geometries, a capped, body-centered, periodic, O-type grid system was chosen. A sample grid of this type is shown for a high turning turbine airfoil cascade in Fig. 3.10. This grid provides a maximum concentration of grid points in a reasonably-sized physical domain, allows clear resolution of the airfoil geometry, and provides planar inlet and exit flow surfaces, useful for the prediction of radiating waves in unsteady flow calculations. This type of grid system has been shown to yield superior resolution of the severe flow gradients encountered at the leading and trailing edges of modern turbine airfoils when compared to similar calculations performed on an H-type grid system (see [63]). In addition, use of the O grid eliminates grid singularities on the airfoil surface and the need for special treatment in the related boundary point calculation.

Grid points in the physical plane are mapped onto a rectangular space in the

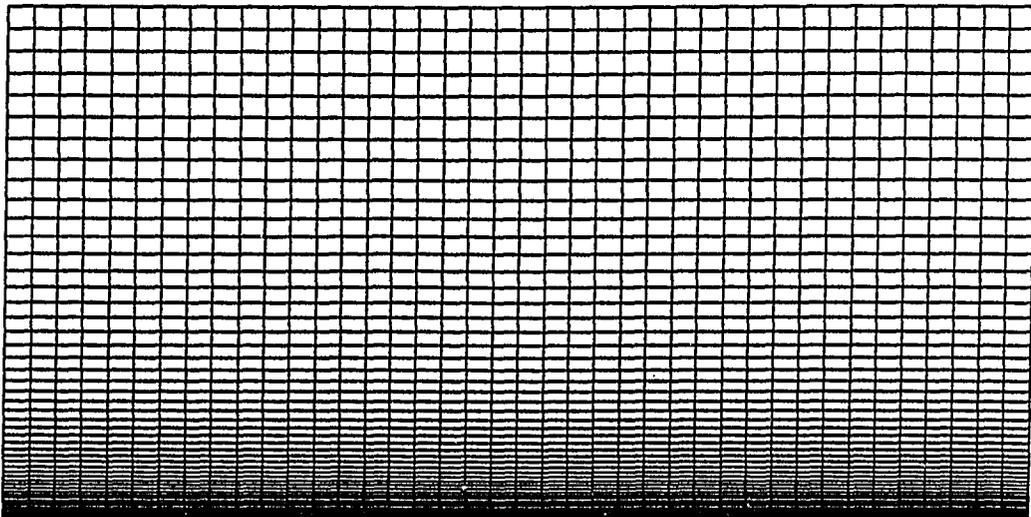


Figure 3.9: Sample H-type grid system for internal flow calculations

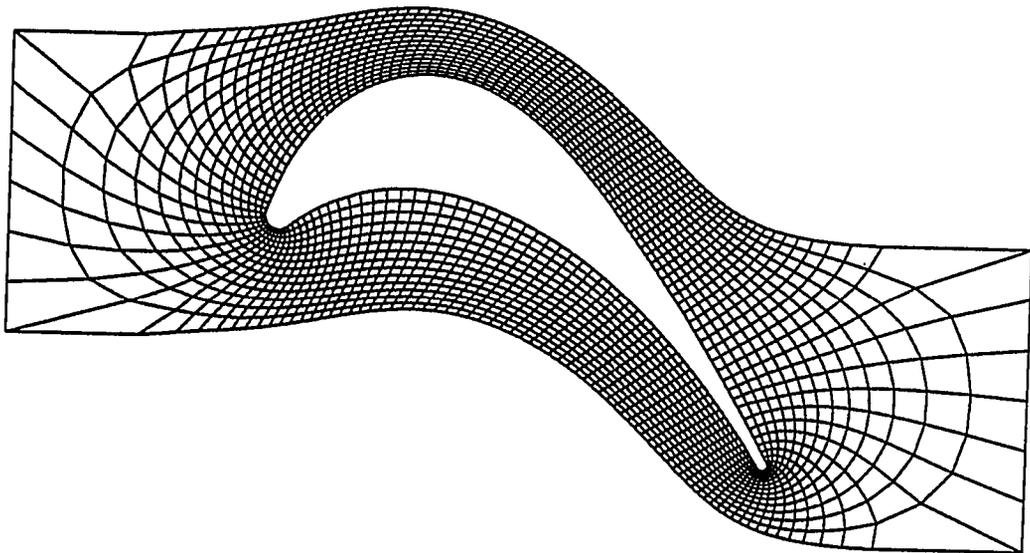


Figure 3.10: Sample O-type grid system for cascade flow calculations

computational domain for use in the numerical solution. This is a simple projection for the internal flow H-type grid system as shown in Fig. 3.11. For the cascade O-type grid, the mapping is accomplished by unwrapping the grid about the grid line extending from the airfoil to the middle of the downstream boundary as shown in Fig. 3.12. The number of points along the inlet and exit planes and the total number of points defining the airfoil surface must therefore both be odd numbers.

The determination of grid points for the H-type grids is a simplification of the procedure for the O-type grids, hence, only the O-grid procedure is described here. Grid points are determined in a two step procedure. The initial interior grid points are generated through the numerical solution of a set of elliptic equations controlling a weighted distribution of grid smoothness, orthogonality, and grid point density based on a variational formulation originally developed by Brackbill and Saltzman [73]. A brief description of this scheme is given below.

For a two-dimensional grid, the following integral expressions may be derived to evaluate critical aspects of the overall grid quality in physical space:

Grid smoothness:

$$I_s = \int \int [(\nabla\xi)^2 + (\nabla\eta)^2] dx dy \quad (3.8)$$

Grid orthogonality:

$$I_o = \int \int [(\nabla\xi) \cdot (\nabla\eta)] dx dy \quad (3.9)$$

Grid point density:

$$I_w = \int \int w(x, y) J dx dy \quad (3.10)$$

where:

$$\xi = \xi(x, y) \quad \eta = \eta(x, y) \quad (3.11)$$

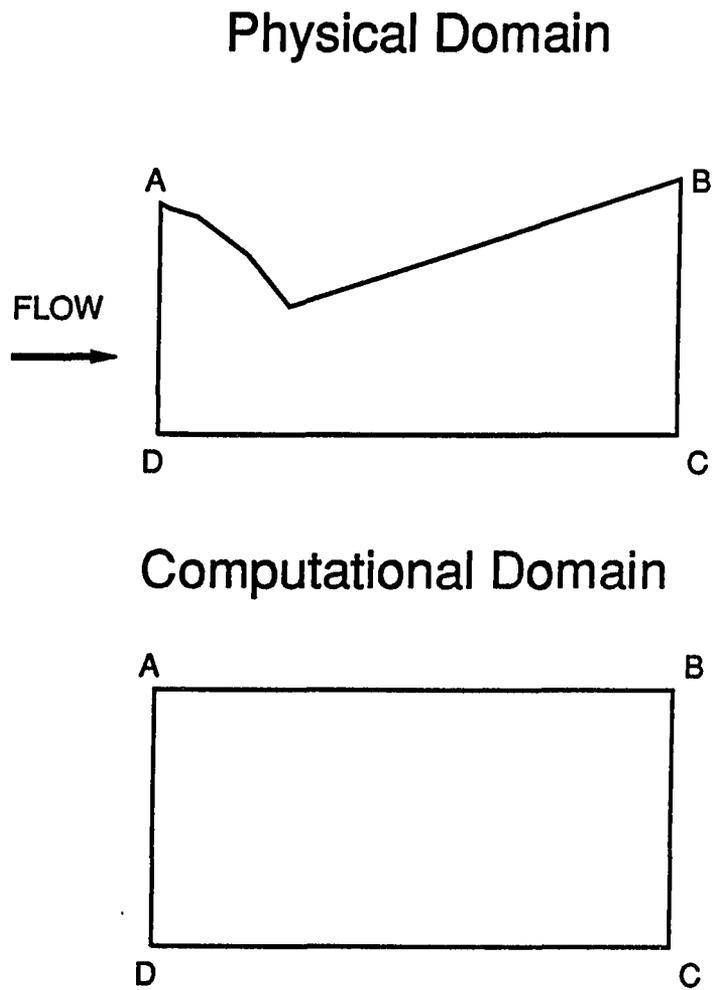
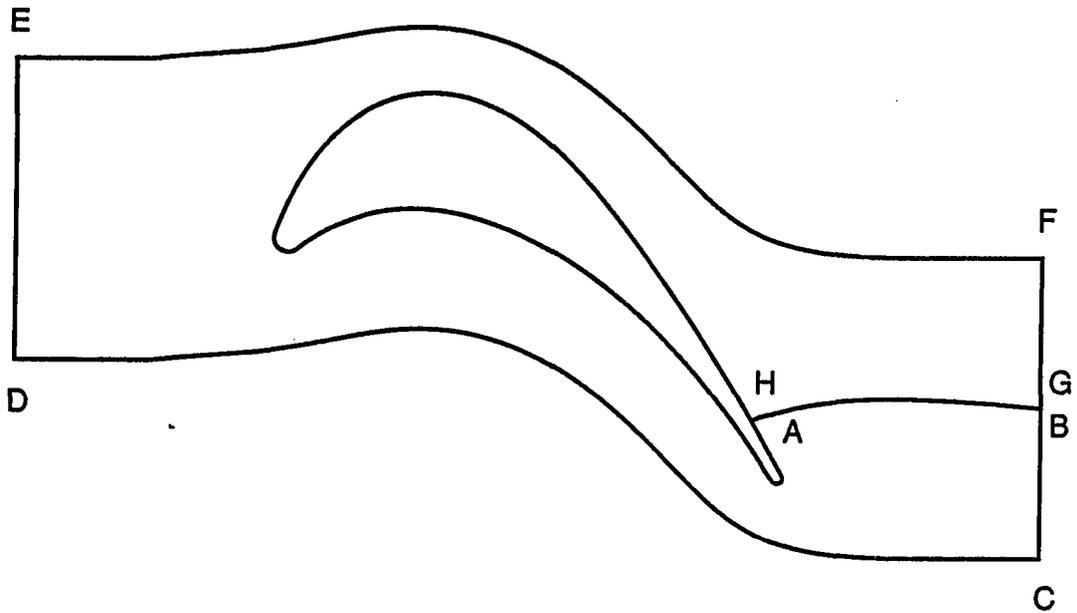


Figure 3.11: Physical and computational mapping of H-type grid system

PHYSICAL DOMAIN



COMPUTATIONAL DOMAIN

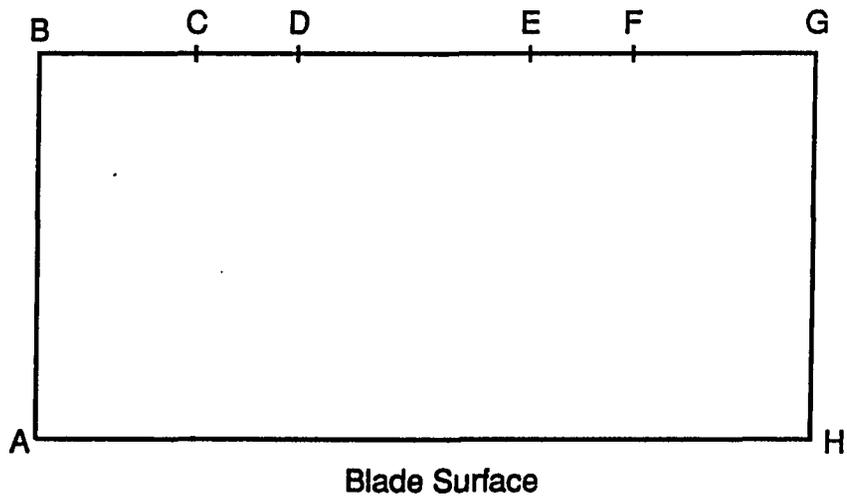


Figure 3.12: Physical and computational mapping of O-type grid system

$$J = \frac{\partial(\xi, \eta)}{\partial(x, y)} \quad (3.12)$$

and where ∇ is the Cartesian gradient vector operator. The term $w(x, y)$ is a user-specified function the magnitude of which is proportional to the desired grid point density in physical space.

Obviously, the smoothest possible grid is obtained when I_s is minimized, the most orthogonal grid is obtained when I_o is minimized, and the grid with the most desirable point density is obtained when I_w is minimized. By minimizing a weighted sum of these terms, i.e:

$$I = I_s + C_o I_o + C_w I_w \quad (3.13)$$

the constants C_o, C_w may be used to control the relative importance of orthogonality and point density, respectively. By exchanging dependent and independent variables, and applying the concepts of variational calculus for minimizing functions using the Euler-Lagrange equations, the following nonlinear coupled set of equations results:

$$b_1 x_{\xi\xi} + b_2 x_{\xi\eta} + b_3 x_{\eta\eta} + a_1 y_{\xi\xi} + a_2 y_{\xi\eta} + a_3 y_{\eta\eta} = -J^2 \frac{1}{2w} \frac{\partial w}{\partial x} \quad (3.14)$$

$$a_1 x_{\xi\xi} + a_2 x_{\xi\eta} + a_3 x_{\eta\eta} + c_1 y_{\xi\xi} + c_2 y_{\xi\eta} + c_3 y_{\eta\eta} = -J^2 \frac{1}{2w} \frac{\partial w}{\partial y} \quad (3.15)$$

where the coefficients ($a_i, b_i, c_i, i = 1, 3$) are all functions of the coordinate derivatives as:

$$\begin{aligned} a_1 &= a_{s1} + C_o a_{o1} + C_w a_{v1} & a_2 &= a_{s2} + C_o a_{o2} + C_w a_{v2} & a_3 &= a_{s3} + C_o a_{o3} + C_w a_{v3} \\ b_1 &= b_{s1} + C_o b_{o1} + C_w b_{v1} & b_2 &= b_{s2} + C_o b_{o2} + C_w b_{v2} & b_3 &= b_{s3} + C_o b_{o3} + C_w b_{v3} \\ c_1 &= c_{s1} + C_o c_{o1} + C_w c_{v1} & c_2 &= c_{s2} + C_o c_{o2} + C_w c_{v2} & c_3 &= c_{s3} + C_o c_{o3} + C_w c_{v3} \end{aligned} \quad (3.16)$$

where:

$$a_{s1} = -(aa)\alpha \quad a_{s2} = 2(aa)\beta \quad a_{s3} = -(aa)\gamma$$

$$b_{s1} = (bb)\alpha \quad b_{s2} = -2(bb)\beta \quad b_{s3} = (bb)\gamma$$

$$c_{s1} = (cc)\alpha \quad c_{s2} = -2(cc)\beta \quad c_{s3} = -(cc)\gamma$$

$$a_{o1} = x\eta y\eta \quad a_{o2} = x\xi y\eta + x\eta y\xi \quad a_{o3} = x\xi y\xi$$

$$b_{o1} = x_\eta^2 \quad b_{o2} = 2(2x_\xi x_\eta + y_\xi y_\eta) \quad b_{o3} = x_\xi^2$$

$$c_{o1} = y_\eta^2 \quad c_{o2} = 2(x_\xi x_\eta + 2y_\xi y_\eta) \quad c_{o3} = y_\xi^2$$

$$a_{v1} = -x\eta y\eta \quad a_{v2} = x\xi y\eta + x\eta y\xi \quad a_{v3} = -x\xi y\xi$$

$$b_{v1} = y_\eta^2 \quad b_{v2} = -2y_\xi y_\eta \quad b_{v3} = x_\xi^2$$

$$c_{v1} = x_\eta^2 \quad c_{v2} = -2x_\xi x_\eta \quad c_{v3} = x_\xi^2 \quad (3.17)$$

$$(aa) = x_\xi y_\xi + x\eta y\eta \quad (bb) = y_\xi^2 + y_\eta^2 \quad (cc) = x_\xi^2 + x_\eta^2 \quad (3.18)$$

$$\alpha = (x_\eta^2 + y_\eta^2)/J^3 \quad \beta = (x_\xi x_\eta + y_\xi y_\eta)/J^2 \quad \gamma = (x_\xi^2 + y_\xi^2)/J^3 \quad (3.19)$$

This system is more complex than the usual Poisson-type grid generation schemes [74], but is still solvable using standard relaxation techniques. In this case, an iterative successive overrelaxation Gauss-Seidel solution technique is applied to solve the finite-difference equations resulting from a second-order central-difference approximation of equations (3.14-3.15) above. For example:

$$x_{\xi\xi} \approx \frac{x_{i+1,j} - 2x_{i,j} + x_{i-1,j}}{(\Delta\xi)^2} \quad (3.20)$$

$$x_{\xi\eta} \approx \frac{x_{i+1,j+1} - x_{i+1,j-1} + x_{i-1,j-1} - x_{i-1,j+1}}{(\Delta\xi\Delta\eta)} \quad (3.21)$$

The numerical solution of the grid is completed through a specification of external boundary conditions. The number of points along the inlet and exit planes are directly specified, and must be an odd number. The x coordinate location of the inlet and exit planes are specified by the user and held fixed. The y coordinates along the inlet and exit planes are equally spaced between the corners of the grid (i.e., points C, D, E, and F in Fig. 3.12). The y coordinates of the corner points are adjusted up or down such that the grid lines at the corners form right angles. For example, at point E in Fig. 3.12 $y_{i,j} = y_{i+1,j}$. The periodic boundaries and overlap boundary grid points are determined through a modified form of the interior point scheme. Data for points lying outside the solution space are obtained by extrapolation of data along overlapping grid lines. A specification of the cascade airfoil spacing is also required to complete the periodic boundary point calculation.

The initial grid point distribution along the airfoil is also specified; however, for cascade geometries, this can often lead to highly skewed mesh lines near the airfoil surface, or across periodic boundaries. To circumvent this problem, grid points were allowed to "float" along the contour of the airfoil as the interior point calculation proceeded, to ensure orthogonality at the cascade surface. The movement of the airfoil grid points is controlled by a secant iteration procedure which optimizes the orthogonality of the surface grid point location as a function of arc length along the airfoil while maintaining a smooth transition through neighboring surface grid points. The new airfoil coordinates are determined from the updated value of arc length through a linear interpolation of the arc length and airfoil coordinates originally specified. The resulting grid thus possesses the property of orthogonality along the

airfoil contour, which is desirable in terms of solution accuracy, and also simplifies many numerical boundary conditions.

The secant iteration procedure is expressed as:

$$S_{i,1}^{k+1} = S_{i,1}^k + \frac{(0.0 - O_{i,1}^k)(S_{i,1}^k - S_{i,1}^{k-1})}{(O_{i,1}^k - O_{i,1}^{k-1})} \quad (3.22)$$

where $S_{i,1}$ is arc length measured clockwise around the airfoil from a fixed reference location (1,1) to the point (i,1), k is the secant iteration count, and $O_{i,1}$ is the measure of nonorthogonality,

$$O_{i,1} = (\nabla\xi \cdot \nabla\eta) = [(-y\eta y_\xi - x\eta x_\xi)/J^2]_{i,1} \quad (3.23)$$

The starting values for the secant iteration are determined as:

$$S_{i,1}^1 = S(x_{i+1,1}, y_{i+1,1}) \quad S_{i,1}^0 = S_{i,1} \quad (3.24)$$

In order to avoid overlapping grid lines and to maintain stability, the new surface grid point locations were never allowed to migrate more than one third of the distance between the previous surface location and that of the neighboring grid points. At the last iteration, the initial reference location for the arc length S is also corrected for orthogonality. This procedure is not completely arbitrary since it relies on a wise choice of the initial reference location from which the arc length S is determined. A better method would be to allow the reference location to float with the remaining points; however, during this study, a completely satisfactory method to accomplish this was not found. In each case, the reference location was chosen as the airfoil surface point corresponding to the grid line which extends to the middle of the downstream boundary, thus requiring a careful selection of the initial location for this point. When

the remainder of the values for the grid and airfoil surface coordinates have converged, the reference point is then also corrected to satisfy orthogonality.

The actual implementation of the surface boundary point procedure was delayed until several iterations (50-70) of the interior grid generation procedure had passed to allow a "reasonable" grid shape to develop. Following this initial waiting period, the surface points were updated every 10 iterations as the final solution developed. The overall scheme converged in 100-200 iterations for most cases, and required a slight underrelaxation (i.e., 0.7) during the iterative process.

A closeup of the detail provided by this grid scheme for the leading edge of the high turning turbine blade shown in Fig. 3.10 is given in Fig. 3.13.

Following the initial grid point calculation, a second calculation is performed which redistributes and interpolates grid points along the rays extending from the airfoil surface. Through this procedure, extra grid points may be added to the grid without a complete regeneration, and grid refinement near the airfoil surface necessary for viscous flow calculations may be easily performed. The resulting grid preserves the desirable surface orthogonality property present in the original grid. The current formulation utilizes an interpolation function based on the one-sided Roberts transformation (e.g., see [58]) to achieve refined meshes. This procedure is implemented as follows. Grid points from the original grid are used to establish an interpolation function for coordinates (x, y) based on arc length S_{ray} along the rays extending from the airfoil surface to the outer boundaries of the grid. New coordinate values may then be determined by the following formula [58]:

$$(S_{ray})_j = \frac{(\beta + 1) - (\beta - 1)[(\beta + 1)/(\beta - 1)]^{1-\bar{y}}}{[(\beta + 1)/(\beta - 1)]^{1-\bar{y}} + 1} \quad (3.25)$$

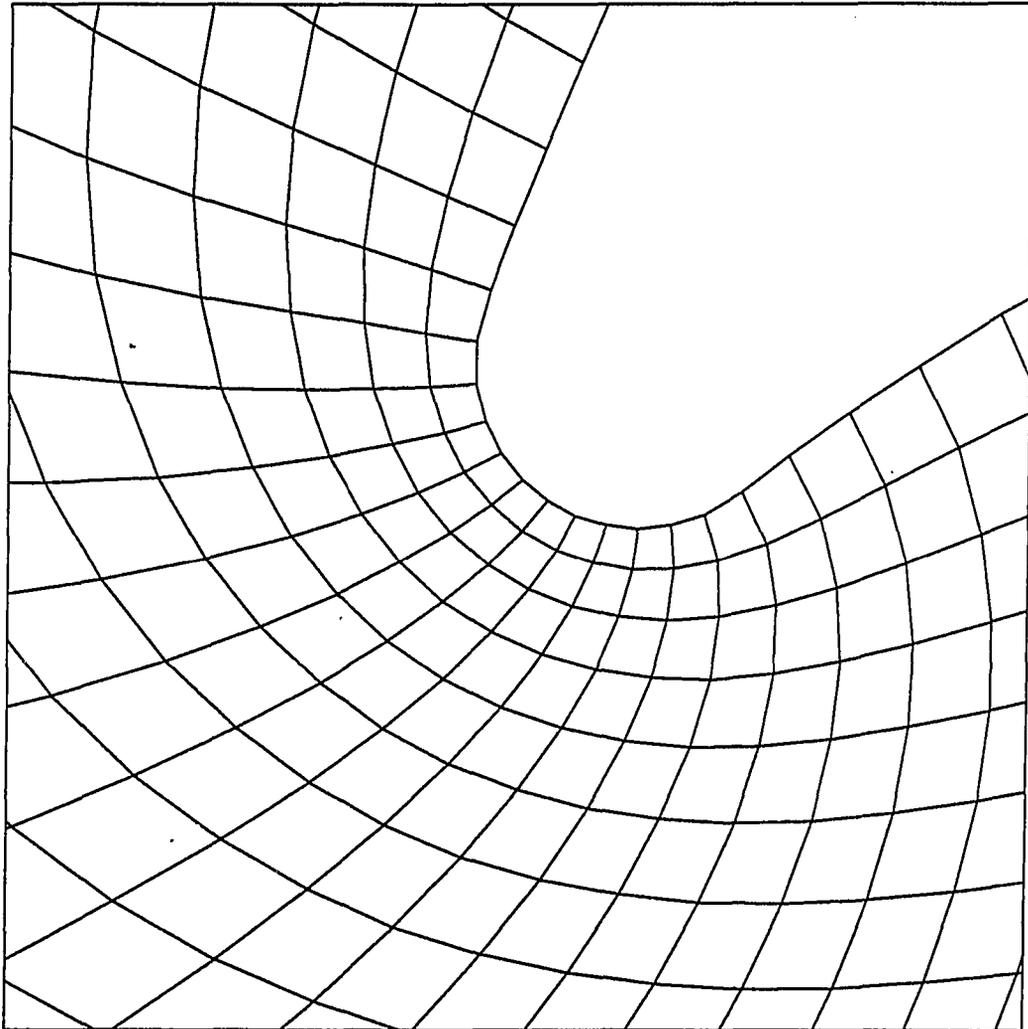
Grid Detail Near Leading Edge

Figure 3.13: Detail of leading edge region for high-turning turbine blade grid

Here $(S_{ray})_j$ is the value of arc length along the ray extending from the airfoil surface used to interpolate the new x, y coordinates from the initial grid, $\bar{y} = (j-1)/(jmax-1)$ where j is the current η grid index and $jmax$ is the new maximum η grid index, and β is a user-supplied parameter which controls the amount of clustering near the airfoil surface. (More clustering occurs as β approaches 1.0.) This technique eliminates the slow convergence problems often associated with grid generation schemes which utilize source functions and a Poisson solver to attain grid refinement.

This two-step grid generation procedure is rapid, robust, and efficient. No cases have been encountered in which the scheme failed to produce an acceptable set of coordinates, and the resulting grids are quite acceptable for use in complex numerical calculations.

Hopscotch Method

The hopscotch class of algorithms is based on the use of different finite-difference expressions at alternating points or groups of points in a calculation domain. It appears that Gordon [75] was the first to suggest the use of mixed explicit and implicit alternating difference schemes for the solution of differential equations, reasoning that the consistency of the individual equations ensured the consistency of the combined equations. Shortly thereafter, Scala and Gordon [76] utilized a mixed difference scheme approach to solve the time-dependent Navier-Stokes equations in nonconservative form for the flow over a circular cylinder. Gourlay [77] also utilized a hopscotch approach for the solution of parabolic and elliptic partial differential equations, and likened the approach to alternating-direction type methods.

Gourlay and Morris [78] formulated a generalized hopscotch-Lax approach for the solution of hyperbolic partial differential equations. This approach is computationally explicit, efficient, easily programmable, has minimal storage requirements, and possesses optimal pseudo-viscosity damping characteristics. Gottlieb and Gustaffson [79] presented a modified hopscotch approach including a special linearization for diffusion terms which maintained the overall explicit nature of the algorithm.

In a comparative study of four explicit schemes for the prediction of quasi-one-dimensional viscous nozzle flows, Holst [80] found that the hopscotch approach was faster than the other methods by a factor of two or more. Holst observed that although the hopscotch scheme was more sensitive to a viscous stability requirement than the other schemes, faster steady state convergence was still obtained. A similar comparison including implicit schemes was reported by Rudy et al. [81], and again the hopscotch procedure was determined to be favorable in terms of both execution efficiency and storage.

Gourlay [82] discussed the use of computationally implicit hopscotch schemes. Through his analysis, it was shown that the solution scheme could be tailored to be directionally implicit. Several implicit and semi-implicit schemes of this type were investigated for the solution of the compressible Navier-Stokes equations by Greenberg [83]. In a case by case comparison, Greenberg noted that the implicit hopscotch approach was twice as fast as fully implicit schemes of the Beam-Warming variety [65], since only half of the normal block tridiagonal inversions were required. This extension of the hopscotch methodology is applicable to other implicit algorithms as well, and could prove to be useful in reducing execution times of more complicated

algorithms.

A number of complex aerodynamic applications utilizing the hopscotch approach have been reported. Delaney [63] utilized Gourlay's hopscotch-Lax scheme for the prediction of steady inviscid transonic flow in two-dimensional turbomachinery cascades using the Euler equations. Lewis et al. [2] extended this approach for the solution of an unsteady turbine vane-blade interaction problem. Shieh and Delaney [84] demonstrated the procedure for steady three-dimensional inviscid turbine flows.

Applications to the compressible Navier-Stokes equations are equally abundant. Kwon and Delaney [85] utilized the modified hopscotch-Lax scheme for the prediction of transonic viscous nozzle flows. Kwon [86] later presented results for an altered form of the modified approach by solving a coupled form of the momentum equations implicitly in a hopscotch-type algorithm for the calculation of viscous transonic flow and heat transfer in turbine cascades.

In this study, a version of the modified hopscotch-Lax approach developed by Gourlay and Morris [78] and Gottlieb and Gustaffson [79] is used in the numerical solution of the governing Equations (2.24-2.31). The modified hopscotch method is an explicit central-difference algorithm that is first order accurate in time and spatially second order accurate [$O(\Delta t, \Delta x^2)$]. The overwhelming advantage of the hopscotch method lies in its speed. Predictions based on the hopscotch approach have been shown to be extremely economical for many calculations. Like many explicit methods, the stability of the method is dependent on a limitation of the calculation time step such that the Courant-Friedrichs-Lewy (*CFL*) number is less than 1.0. An additional viscous stability condition must also be considered. As a result, the practicality of

this method for some unsteady viscous flow calculations may be questionable.

The hopscotch algorithm utilizes a two sweep solution procedure. Both explicit and implicit formulas are used at alternating grid points, although the overall implementation is computationally explicit in nature. The algorithm proceeds as:

$$\hat{Q}_{i,j}^{n+1} = \hat{Q}_{i,j}^n - \Delta t \left(\frac{(\hat{E}_{i+1,j}^k - \hat{E}_{i-1,j}^k)}{2\Delta\xi} + \frac{(\hat{F}_{i,j+1}^k - \hat{F}_{i,j-1}^k)}{2\Delta\eta} \right) + D_{i,j}^k \quad (3.26)$$

where:

$$\hat{E} = \hat{E}_{inv} - \hat{E}_{vis} \quad \hat{F} = \hat{F}_{inv} - \hat{F}_{vis} \quad (3.27)$$

$$k = \begin{cases} n, & \text{first sweep, } i + j + n \text{ odd;} \\ n + 1 & \text{second sweep, } i + j + n \text{ even} \end{cases} \quad (3.28)$$

and $D_{i,j}^k$ is an added numerical damping term to stabilize the solution. In evaluating \hat{E}^{n+1} and \hat{F}^{n+1} , the modified approach is used. Viscous terms are evaluated using the latest updated values of the dependent variables, thus introducing an effective lagging to avoid compromising the linearity of the equations. The overall time accuracy of the method is not impaired by this treatment; however, it is pointed out in [79] that an additional dissipative truncation error term is generated. Metric derivatives are also centrally differenced to satisfy the grid conservation law [58].

Gourlay and Morris [78] noted that for some cases the explicit sweep can be replaced by a simple linear extrapolation in time:

$$\hat{Q}_{i,j}^{n+1} = 2\hat{Q}_{i,j}^n - \hat{Q}_{i,j}^{n-1} \quad (3.29)$$

Equation (3.29) is obtained through repetitive application of the explicit and implicit finite-difference formulas at the same grid point over two time levels under the additional assumption that the local time step (Δt) and damping coefficients (c_ξ, c_η)

(see description below) remain constant. Use of the extrapolation formula for steady flow calculations results in a dramatic saving in computational time; however, for unsteady flow predictions, this simplification is not always applicable, and in those cases the full two sweep version must be employed. It should be noted, however, that numerical calculations for an unsteady, one-dimensional inviscid flow showed no discernable difference between the use of the full two sweep algorithm and the extrapolation scheme.

The use of central differencing for the prediction of discontinuous flows requires the addition of an explicit artificial dissipation term $D_{i,j}^k$, to control oscillations in the predicted solution. The present damping terms utilize the blended second and product fourth order dissipation suggested by Delaney [63] for inviscid cascade predictions and are calculated as:

$$D_{i,j}^k = CFL_{loc}((D_{\xi}^k)_{i,j} + (D_{\eta}^k)_{i,j}) \quad CFL_{loc} = \frac{(\Delta t)_{i,j}}{(\Delta t_{CFL=1.0})_{i,j}} \quad (3.30)$$

$$(D_{\xi}^k)_{i,j} = \frac{(c_{\xi})_{i,j}}{J_{i,j}}(Q_{i+1,j}^k - 2Q_{i,j}^k + Q_{i-1,j}^k)$$

$$(D_{\eta}^k)_{i,j} = \frac{(c_{\eta})_{i,j}}{J_{i,j}}(Q_{i,j+1}^k - 2Q_{i,j}^k + Q_{i,j-1}^k) \quad (3.31)$$

$$(c_{\xi})_{i,j} = c_1 + c_2 \frac{|p_{i+1,j}^n - 2p_{i,j}^n + p_{i-1,j}^n|}{p_{i,j}^n}$$

$$(c_{\eta})_{i,j} = c_1 + c_2 \frac{|p_{i,j+1}^n - 2p_{i,j}^n + p_{i,j-1}^n|}{p_{i,j}^n} \quad (3.32)$$

In the above:

$$(\Delta t)_{CFL=1.0} = \left(abs(U_{\xi}) + abs(U_{\eta}) + a(\sqrt{\xi_x^2 + \xi_y^2} + \sqrt{\eta_x^2 + \eta_y^2}) \right)^{-1} \quad (3.33)$$

where U_ξ and U_η are contravariant velocity components pertaining to the ξ and η coordinates, respectively (See Equation (2.37)).

The damping terms are multiplied by a local CFL number in order to ensure that the dissipation does not overwhelm the physical fluxes for small local time steps. In a linear stability analysis, Gourlay and Morris [78] showed that this scheme is stable for all positive values of the damping coefficients (c_ξ, c_η) and for local time steps satisfying the CFL stability criteria

$$CFL_{loc} \leq 1.0 \quad (3.34)$$

(see Equation (3.30), above.) In contrast, the original Lax scheme [14] requires an upper limit on the damping coefficients for stability.

Gottlieb and Gustaffson [79] performed a rigorous stability analysis of a modified hopscotch scheme for the one-dimensional Navier-Stokes equations and discovered an additional viscous stability requirement which becomes dominant in low Reynolds number flows. This additional requirement expressed in one dimension becomes:

$$\frac{\mu \Delta t}{\rho (\Delta x)^2} \leq 1 \quad (3.35)$$

For high Reynolds number flow calculations, this condition may be approximated by running at local CFL numbers less than 1.0, (0.7 is a typical value). For low Reynolds number predictions, this procedure does not guarantee stability, and Equation (3.35) must be satisfied explicitly.

Inlet and exit plane boundary conditions are enforced through an explicit reference plane method of characteristics calculation (see Appendix). Periodic and overlap boundary points are updated using the full interior point scheme by extrapolating data across grid lines for points outside the solution space.

Along solid boundaries, inviscid flows are treated using a full two sweep (no extrapolation) modified version of the interior point scheme (see also [63]). Difference terms in the direction normal to the surface are replaced by one-sided differences and the damping terms in that direction are omitted. For uniform grids, a three-point one-sided difference may be used, while for highly stretched grids a two point one-sided difference must be used. Once a preliminary solution is obtained at the boundary at the new time step, the predicted velocities are projected onto the surface to satisfy flow tangency (i.e., $U_\eta = 0$; see Equation (2.37)). For viscous flows, the no slip condition is specified, (i.e., $u, v = 0$), and either two point or three point one-sided differences (depending on the grid) are applied to satisfy zero normal pressure gradient and adiabatic conditions. (When rapid changes in the grid spacing occur, the two point formula must be used.) For the appropriate cases, a surface temperature may be specified in place of the adiabatic relationship.

Runge-Kutta Method

The Runge-Kutta class of finite-difference algorithms has rapidly evolved to become a highly popular and reliable temporal differencing strategy for solving time-marching problems. The schemes are characterized as computationally explicit, with a variable number of calculation stages and combination parameters. A general description of the Runge-Kutta technique for gasdynamic problems is given by Jameson et al. [64]. Utilizing various techniques such as multigrid, residual smoothing, and enthalpy damping, calculations have been successfully performed utilizing time increments much larger than the traditional explicit stability limited time step, resulting

in improved convergence rates and CPU times when compared to the unmodified scheme. Unfortunately, these acceleration techniques are generally invalid for the calculation of unsteady flows, and are therefore not used in the present study.

Centered differences are commonly used in the discretization of spatial derivative terms, although it is also possible to utilize upwind differencing techniques. Turkel and Van Leer [87] utilized the Runge-Kutta time-marching procedure with a split flux vector formulation of the spatial derivative terms for the solution of the Euler equations. Although an implicit residual smoothing scheme and multigrid acceleration were applied, the resulting code was still five times slower than an equivalent standard central-difference scheme.

The Runge-Kutta formulation has been successfully applied to a wide range of problems for both inviscid and viscous flows. Jameson and Baker [88] utilized a four stage Runge-Kutta procedure based on a finite volume formulation with implicit residual smoothing to solve the Euler equations for the steady inviscid flow about a three-dimensional complex wing-body-tail aircraft. Subramanian and Bozzola [89] utilized a similar approach to solve for steady inviscid flows in turbomachines. Swanson and Turkel [90] also utilized a four stage Runge-Kutta procedure to solve the thin-layer Navier-Stokes equations for laminar and turbulent viscous flows about flat plates and airfoils. Several of the previously mentioned acceleration techniques were utilized and an improved convergence rate for steady flows was demonstrated with the addition of each technique. Morinishi and Satofuka [91] performed similar calculations using a two stage Runge-Kutta scheme.

An application of the Runge-Kutta procedure for unsteady flow calculations is

given by Jameson and Venkatakrishnan [40]. In this case, a three stage Runge-Kutta scheme was applied to solve the Euler equations for transonic unsteady flow about a harmonically oscillating airfoil. Both total variation diminishing (TVD) and blended second/fourth dissipation schemes were examined, with little discernable difference in the computed results, although the TVD type dissipation required about twice as much computer time. Jorgenson and Chima [4] successfully applied the four-stage Runge-Kutta technique for the calculation of unsteady turbulent quasi-three-dimensional flow in a rotor/stator interaction problem.

It is also possible to apply the Runge-Kutta scheme for the prediction of incompressible flows. Merkle and Tsai [92] utilized an artificial compressibility technique with implicit residual smoothing for the calculation of incompressible flows.

The present application of the Runge-Kutta technique is based on a four-stage time-marching solution procedure with spatial central differencing, and is fourth order accurate in time while maintaining second order spatial accuracy. $[O(\Delta t^4, \Delta x^2)]$. Numerical experiments were performed utilizing a fully fourth order accurate centered spatial differencing technique in an attempt to improve the overall spatial accuracy of the code. Unfortunately, stability problems were encountered due to the use of non-centered fourth order differencing techniques near boundaries where the centered scheme could not be applied. A Fourier stability analysis was performed on several one-sided differencing techniques of fourth and fifth order accuracy using the symbolic manipulation code MACSYMA on the Iowa State University Engineering Vax 11-785 computer. Plotted stability curves for these higher-order one-sided difference formula indicated that they were unconditionally unstable. The fourth order accurate

differencing scheme was therefore abandoned due to the apparent inability to maintain fourth order spatial accuracy near boundaries.

The four-stage calculation may be expressed as:

$$\begin{aligned}
\hat{Q}_{i,j}^{(0)} &= \hat{Q}_{i,j}^n \\
\hat{Q}_{i,j}^{(1)} &= \hat{Q}_{i,j}^{(0)} - \alpha_1(\Delta t(RHS)_{i,j}^{(0)} - D_{i,j}^n) \\
\hat{Q}_{i,j}^{(2)} &= \hat{Q}_{i,j}^{(0)} - \alpha_2(\Delta t(RHS)_{i,j}^{(1)} - D_{i,j}^n) \\
\hat{Q}_{i,j}^{(3)} &= \hat{Q}_{i,j}^{(0)} - \alpha_3(\Delta t(RHS)_{i,j}^{(2)} - D_{i,j}^n) \\
\hat{Q}_{i,j}^{(4)} &= \hat{Q}_{i,j}^{(0)} - \alpha_4(\Delta t(RHS)_{i,j}^{(3)} - D_{i,j}^n) \\
\hat{Q}_{i,j}^{n+1} &= \hat{Q}_{i,j}^{(4)}
\end{aligned} \tag{3.36}$$

$$\begin{aligned}
\alpha_1 &= \frac{1}{4} & \alpha_2 &= \frac{1}{3} & \alpha_3 &= \frac{1}{2} & \alpha_4 &= 1 \\
(RHS)_{i,j}^{(k)} &= \left(\frac{(\hat{E}_{i+1,j}^{(k)} - \hat{E}_{i-1,j}^{(k)})}{2\Delta\xi} + \frac{(\hat{F}_{i,j+1}^{(k)} - \hat{F}_{i,j-1}^{(k)})}{2\Delta\eta} \right)
\end{aligned} \tag{3.37}$$

Here $\hat{E}_{i,j}^{(k)} = \hat{E}(\hat{Q}_{i,j}^{(k)})$. Again, due to the use of centered differences, an explicit artificial dissipation term is required to control odd-even point decoupling. The dissipation term $D_{i,j}^n$, described by Delaney [63] and defined in the previous section is also used here. Hence, in light of the similarity of the boundary point calculation schemes, the Runge-Kutta scheme and the hopscotch scheme produce the same steady state solution when equivalent damping parameters are used.

Use of the four stage formula allows a larger calculation time step than most other explicit formulas at the expense of some additional computational work. A

Fourier stability analysis indicates that the scheme is stable for time increments satisfying ($CFL \leq 2\sqrt{2}$) [64]. The Runge-Kutta procedure thus has advantages due to the relaxed stability requirement, and the additional time accuracy afforded by the multistage procedure. Although it is possible to fine-tune the coefficients ($\alpha_1 - \alpha_4$) for specific cases to improve convergence, the values of α given above are generally reliable.

Approximately Factored Linearized Block Implicit (AFLBI) Method

The approximately factored linearized block implicit (AFLBI) procedure is widely used for implicit calculations of both viscous and inviscid flows. The algorithm is based on an approximate factorization of a large, banded block coefficient matrix resulting from a time linearization of an implicit finite-difference formula, and belongs to class of algorithms known as alternating direction implicit (ADI) methods. Following the discretization presented by Beam and Warming [65] among others, the approach utilizes central differences for the spatial derivatives in conjunction with several alternative representations for temporal derivatives. The present application utilizes a first order accurate time representation. A nonlinear adaptive damping scheme is used to stabilize the oscillations associated with the central-difference formulation. The implicit formula is linearized through a Taylor series expansion in time and application of the chain rule formula. An approximate linearization is used for the damping terms, resulting in an implicit artificial damping term as well.

The AFLBI algorithm is based on a fully implicit second order central-difference

discretization of the governing Equations (2.24-2.31)

$$\frac{\hat{Q}_{i,j}^{n+1} - \hat{Q}_{i,j}^n}{\Delta t} + \frac{\hat{E}_{i+1,j}^{n+1} - \hat{E}_{i-1,j}^{n+1}}{2\Delta\xi} + \frac{\hat{F}_{i,j+1}^{n+1} - \hat{F}_{i,j-1}^{n+1}}{2\Delta\eta} = D_{i,j}^n \quad (3.38)$$

where:

$$\hat{E} = \hat{E}_{inv} - \hat{E}_{visc} \quad \hat{F} = \hat{F}_{inv} - \hat{F}_{visc}$$

Here again, due to the use of centered spatial differencing, an artificial dissipation term, $D_{i,j}^n$, is added to control nonphysical oscillations. This discretization has truncation error $O(\Delta t, \Delta x^2)$. As written, the above equation is nonlinear due to the dependence of the fluxes $\hat{E}^{n+1}, \hat{F}^{n+1}$ on the vector of dependent variables \hat{Q}^{n+1} . Separate linearization schemes are utilized for the inviscid and viscous components of the flux vectors \hat{E} and \hat{F} . The inviscid components of the flux vectors are linearized using a Taylor series expansion in time as:

$$(\hat{E}_{inv})_{i,j}^{n+1} = (\hat{E}_{inv})_{i,j}^n + \left(\frac{\partial \hat{E}_{inv}}{\partial t} \right)_{i,j}^n \Delta t + O(\Delta t)^2 \quad (3.39)$$

The linearization is completed through an application of the chain rule as:

$$(\hat{E}_{inv})_{i,j}^{n+1} = (\hat{E}_{inv})_{i,j}^n + \left(\frac{\partial \hat{E}_{inv}}{\partial Q} \right)_{i,j}^n \left(\frac{\partial \hat{Q}}{\partial t} \right)_{i,j}^n \Delta t + O(\Delta t)^2 \quad (3.40)$$

The derivative $\partial \hat{Q} / \partial t$ is then approximated by a first order forward difference in time, resulting in an overall linearization error which is second order accurate in time. The final linearization formula becomes:

$$(\hat{E}_{inv})_{i,j}^{n+1} \approx (\hat{E}_{inv})_{i,j}^n + \hat{A}_{i,j}^n (\hat{Q}_{i,j}^{n+1} - \hat{Q}_{i,j}^n) \quad (3.41)$$

The viscous components of the flux vectors are approximated by simply lagging the desired values from the previous time step as:

$$(\hat{E}_{vis})_{i,j}^{n+1} \approx (\hat{E}_{vis})_{i,j}^n \quad (3.42)$$

Without iteration, this linearization results in a first order accurate truncation error in time. It is also possible to utilize a more accurate linearization procedure such as the Taylor series method described above for the inviscid flux terms. Unfortunately, the Taylor series technique is expensive in terms of computational effort for viscous terms, and requires special modification for cross derivative terms as well. Therefore, the simpler lagging procedure was adopted.

The resulting simultaneous matrix equation which results from the application of the linearization procedures may be written in the so-called delta formulation as:

$$[I + \delta_\xi \hat{A}_{i,j}^n + \delta_\eta \hat{B}_{i,j}^n - \frac{(c_\xi)_{i,j}}{J_{i,j}} \Delta_\xi \nabla_\xi J_{i,j} - \frac{(c_\eta)_{i,j}}{J_{i,j}} \Delta_\eta \nabla_\eta J_{i,j}] [\Delta \hat{Q}_{i,j}]^n = \quad (3.43)$$

$$- \Delta t \left(\frac{(\hat{E}_{i+1,j}^n - \hat{E}_{i-1,j}^n)}{2\Delta\xi} + \frac{(\hat{F}_{i,j+1}^n - \hat{F}_{i,j-1}^n)}{2\Delta\eta} + D_{i,j}^n \right) \quad (3.44)$$

where:

$$[\Delta \hat{Q}_{i,j}]^n = \hat{Q}_{i,j}^{n+1} - \hat{Q}_{i,j}^n$$

$$\delta_\xi w = 0.5(w_{i+1,j} - w_{i-1,j}) \quad \delta_\eta w = 0.5(w_{i,j+1} - w_{i,j-1})$$

$$\Delta_\xi w = w_{i+1,j} - w_{i,j} \quad \Delta_\eta w = w_{i,j+1} - w_{i,j}$$

$$\nabla_\xi w = w_{i,j} - w_{i-1,j} \quad \nabla_\eta w = w_{i,j} - w_{i,j-1}$$

It should be mentioned that the artificial dissipation term $D_{i,j}$ has been approximately linearized in the implicit coefficient array. Studies indicate that more accurate linearization of explicit damping terms can enhance convergence and stability. However, this form was thought to be suitable for the present applications.

As written, the direct inversion of the matrix equation would be extremely time consuming. Instead, the matrix is approximately factored. This factorization is

not unique, and in fact, a multitude of algorithms may be derived from various combinations of factorization and subsequent solution. Mitchell and Griffiths [93] gives a thorough description of a number of popular methods of this type. In this study, a variation of the Douglas-Rachford splitting proposed by D'Yakanov (e.g., see [93]) is utilized. The approximate factorization allows independent reduction in each of the coordinate directions as:

$$[I + \delta_{\xi} \hat{A}_{i,j}^n - \frac{(c_{\xi})_{i,j}}{J_{i,j}} \Delta_{\xi} \nabla_{\xi} J_{i,j}] [I + \delta_{\eta} \hat{B}_{i,j}^n - \frac{(c_{\eta})_{i,j}}{J_{i,j}} \Delta_{\eta} \nabla_{\eta} J_{i,j}] [\Delta \hat{Q}_{i,j}]^n =$$

$$RHS_{i,j} = -\Delta t \left(\frac{(\hat{E}_{i+1,j}^n - \hat{E}_{i-1,j}^n)}{2\Delta\xi} + \frac{(\hat{F}_{i,j+1}^n - \hat{F}_{i,j-1}^n)}{2\Delta\eta} \right) + \Delta t D_{i,j}^n \quad (3.45)$$

and a two sweep alternating direction type reduction proceeds as:

$$[I + \delta_{\xi} \hat{A}_{i,j}^n + \frac{(c_{\xi})_{i,j}}{J_{i,j}} \Delta_{\xi} \nabla_{\xi} J_{i,j}] [\Delta \hat{Q}_{i,j}]^* = RHS_{i,j}$$

$$[I + \delta_{\eta} \hat{B}_{i,j}^n + \frac{(c_{\eta})_{i,j}}{J_{i,j}} \Delta_{\eta} \nabla_{\eta} J_{i,j}] [\Delta \hat{Q}_{i,j}]^n = [\Delta \hat{Q}_{i,j}]^*$$

$$\hat{Q}_{i,j}^{n+1} = \hat{Q}_{i,j}^n + [\Delta \hat{Q}_{i,j}]^n \quad (3.46)$$

Here, the first sweep is performed along each of the ξ family of grid lines, and the second sweep is performed along each of the η family of grid lines. The decoupling of the solution in each direction resulting from the approximate factorization yields an additional second order time error.

For H-grid solutions, the solution is obtained by a sequence of standard, non-periodic block tridiagonal reductions. For O-grid calculations, the first sweep above (implicit along a line of constant η), requires the solution of a periodic 4x4 block element tridiagonal matrix due to the overlapping O-type grid system. The second sweep

(implicit along a line of constant ξ), involves the inversion of a non-periodic 4x4 block element tridiagonal matrix. The solution remains fully implicit across the spatially periodic boundary due to a slight reordering of the grid points in the solution matrix (see [94]). The tridiagonal reduction is performed using a simplified version of Gaussian elimination designed specifically for block tridiagonal matrices (e. g., see [58]). In order to improve the performance of this algorithm on vector operation computers such as the Cray X-MP, the reduction algorithms are written to take advantage of the fact that while the non-vectorizable reduction occurs for variables along one set of coordinate lines, the solution will vectorize across the remaining set of coordinate lines. Hence, the reductions for every coordinate line during a sweep are performed simultaneously. This requires storing all of the coefficient arrays during each sweep of the reduction procedure; however, the improvement in execution efficiency due to increased vectorization is well worth the increase in memory required.

Linear stability analysis of the implicit discretization described above indicates that the scheme is unconditionally stable. However, errors introduced through the approximate factorization, and the uncertainties associated with the imposition of boundary conditions impose a practical stability limit of about $CFL = 10$.

The adaptive nonlinear dissipation function suggested by Pulliam [66] was utilized in the implicit algorithm to monitor the central difference oscillations, and is determined by:

$$D_{i,j}^n = (D_{\xi}^n)_{i,j} + (D_{\eta}^n)_{i,j} \quad (3.47)$$

$$(D_{\xi}^n)_{i,j} = \left(\frac{1}{\sigma_{i+1,j} J_{i+1,j}} + \frac{1}{\sigma_{i,j} J_{i,j}} \right) \left((\epsilon_{i,j}^{(2)})_{\xi} \Delta_{\xi} Q_{i,j} - (\epsilon_{i,j}^{(4)})_{\xi} \Delta_{\xi} \nabla_{\xi} \Delta_{\xi} Q_{i,j} \right) \quad (3.48)$$

$$(D_\eta^n)_{i,j} = \left(\frac{1}{\sigma_{i,j+1} J_{i,j+1}} + \frac{1}{\sigma_{i,j} J_{i,j}} \right) \left((\epsilon_{i,j}^{(2)})_\eta \Delta_\eta Q_{i,j} - (\epsilon_{i,j}^{(4)})_\eta \Delta_\eta \nabla_\eta \Delta_\eta Q_{i,j} \right) \quad (3.49)$$

$$\begin{aligned} (\epsilon_{i,j}^{(2)})_\xi &= c_1 (\max[(\Upsilon_\xi)_{i+1,j}, (\Upsilon_\xi)_{i,j}, (\Upsilon_\xi)_{i-1,j}]) \\ (\epsilon_{i,j}^{(2)})_\eta &= c_1 (\max[(\Upsilon_\eta)_{i,j+1}, (\Upsilon_\eta)_{i,j}, (\Upsilon_\eta)_{i,j-1}]) \end{aligned} \quad (3.50)$$

$$(\Upsilon_\xi)_{i,j} = \frac{|p_{i+1,j}^n - 2p_{i,j}^n + p_{i-1,j}^n|}{|p_{i+1,j}^n + 2p_{i,j}^n + p_{i-1,j}^n|} \quad (\Upsilon_\eta)_{i,j} = \frac{|p_{i,j+1}^n - 2p_{i,j}^n + p_{i,j-1}^n|}{|p_{i,j+1}^n + 2p_{i,j}^n + p_{i,j-1}^n|} \quad (3.51)$$

$$(\epsilon_{i,j}^{(4)})_\xi = \max[0, (c_2 - (\epsilon_{i,j}^{(2)})_\xi)] \quad (\epsilon_{i,j}^{(4)})_\eta = \max[0, (c_2 - (\epsilon_{i,j}^{(2)})_\eta)] \quad (3.52)$$

$$\sigma_{i,j} = \left(\text{abs}(U_\xi) + \text{abs}(U_\eta) + a(\sqrt{\xi_x^2 + \xi_y^2} + \sqrt{\eta_x^2 + \eta_y^2}) \right)_{i,j}^{-1} \quad (3.53)$$

where the recommended values for c_1 and c_2 are (0.25, 0.007), respectively. Results using this dissipation scheme, and a more advanced conservative dissipation scheme described by Caughey and Turkel [95] showed little discernable difference in the computed results. Near a solid boundary, the fourth order dissipation in the η direction is eliminated. The second order dissipation at the surface is determined by adding a small constant (i.e., 0.01) to $\epsilon_{i,j}^{(2)}$ to maintain some damping when the pressure derivative term is very small. This modification was found to enhance convergence for viscous flows.

Here again, a central difference representation must be used for all metric derivative terms to satisfy the grid conservation law. Boundary conditions are imposed implicitly utilizing a linearization procedure similar to the Taylor series linearization described above for inviscid fluxes. Various implicit boundary condition treatments are described in [96], [97]. It is also possible to use explicit boundary conditions, and

in fact, this is a simpler procedure, although convergence can be somewhat adversely affected by their use.

Generalized Conjugate Gradient Total Variation Diminishing (GCGTVD) Method

The generalized conjugate gradient total variation diminishing (GCGTVD) scheme is the most complicated of the four time-marching algorithms compared in this study. The overall finite-difference representation is based on a centered finite-volume representation of the spatial derivative terms. Based on the high and low Reynolds number limit behavior of each term in the governing equations, the GCGTVD algorithm utilizes different differencing techniques for the inviscid and viscous flux derivative terms. The inviscid terms, which exhibit a hyperbolic wavelike behavior for compressible flows, are differenced in an upwind manner, while the viscous terms, which maintain a parabolic character, are centrally-differenced. The representative inviscid fluxes are constructed by an approximate Riemann solution of the piecewise continuous data between grid points. The approximate Riemann solver recognizes the direction and magnitude of waves in the solution, and therefore upwind differencing techniques may be applied in the flux construction. Further, the manner in which the upwinding is applied is chosen such that the resulting scheme is inherently nonoscillatory and does not require the explicit addition of an artificial viscosity term [69]. The class of algorithms of this type used in this study are referred to as total variation diminishing (TVD), since they are based on the principle of nonincreasing total variation for scalar conservation laws [19]. The present upwind formulation utilizes an

implicit TVD type formulation and a generalized conjugate gradient minimal residual descent matrix relaxation procedure which resembles the method of steepest descent. At every time step, a calculation loop is established which iteratively updates the implicit flux values. This procedure eliminates linearization errors in the resulting solution [67], and avoids the factorization errors associated with approximately factored schemes. The algorithm is inherently robust and is capable of capturing shock waves with remarkable clarity on nonrefined grids.

The development of upwind-biased algorithms for systems of hyperbolic conservation laws has progressed rapidly in recent years. The original development can be traced to the conservation principle established by Lax [14]; however, it was Godunov [98] who first suggested that fluxes at a grid cell interface could be determined from a solution of the Riemann problem associated with the piecewise continuous data between grid points. Godunov's scheme solved an exact Riemann problem and was successful in producing solutions demonstrating a smooth, monotone transition across predicted flow discontinuities. Unfortunately, Godunov's scheme was extremely slow due to the complex Riemann solver in the flux construction, and was no better than first order accurate spatially. Roe [18] developed a similar solution scheme based on an approximate Riemann solver which effectively duplicated Godunov's results without incurring the expense of the full Riemann solver. A number of algorithms quickly followed which in many cases reduce to Roe's scheme. Yee and Harten [19] extended the approach to second order spatial accuracy through their modified flux approach. Chakravarthy presented two schemes [99], [21] which approached third order accuracy through an interpolation procedure. Chakravarthy and Osher [20] and Coakley

[72] also proposed similar schemes. Further work is currently in progress to improve the overall accuracy of the TVD approach by basing solutions on a nonoscillatory principle [100], [69].

Another set of algorithms was developed based on the flux vector splitting technique described by Steger and Warming [16], and later extensively redefined and applied by Van Leer [17]. However, recent evidence suggests that the flux vector splitting approach may not be as suitable for viscous flow calculations as the flux difference splitting technique in Roe's approximate Riemann solver; hence, the flux difference splitting approach was pursued in this study.

The underlying implicit algorithm in the GCGTVD approach may be expressed by the following semi-discrete finite-volume conservation statement:

$$\begin{aligned} \hat{Q}_{i,j}^{n+1} &= \hat{Q}_{i,j}^n \\ \Delta t \left(\frac{((\hat{E}_{inv})_{i+\frac{1}{2},j}^{n+1} - (\hat{E}_{inv})_{i-\frac{1}{2},j}^{n+1})}{\Delta \xi} + \frac{((\hat{F}_{inv})_{i,j+\frac{1}{2}}^{n+1} - (\hat{F}_{inv})_{i,j-\frac{1}{2}}^{n+1})}{\Delta \eta} - \right. \\ &\quad \left. \frac{((\hat{E}_{vis})_{i+\frac{1}{2},j}^{n+1} - (\hat{E}_{vis})_{i-\frac{1}{2},j}^{n+1})}{\Delta \xi} - \frac{((\hat{F}_{vis})_{i,j+\frac{1}{2}}^{n+1} - (\hat{F}_{vis})_{i,j-\frac{1}{2}}^{n+1})}{\Delta \eta} \right) \end{aligned} \quad (3.54)$$

The conservative formula shown above is representative of a finite-volume formulation if one considers the metric Jacobian term, $1/J$, equivalent to the area of the cell under consideration, and the associated metric derivatives $\xi_{x,y}$, $\eta_{x,y}$ to be representative of the cell face normal vector components. The finite volume is centered about each grid point as shown in Fig. 3.14. The vertices of the finite volume are determined by averaging the coordinates of the four grid points which surround each corner of

the finite volume. Volumes adjacent to a computational boundary therefore require care to ensure that the proper volume is calculated from the non trapezoidal shape of the cell (see the cell bounded by points C-D-E-F in Fig. 3.14). In this manner, the sum of the volumes exactly matches the volume of the computational domain. This arrangement permits storing computational variables at the grid points themselves (instead of a staggered grid arrangement), but does not guarantee that the grid point is in the geometric center of the finite volume. Cell face normal vector components are calculated using the finite volume corner vertices, and therefore represent grid derivatives along the face of the cell, not at the grid point itself. This method was chosen to simplify the calculation of fluxes along the boundary of each cell. Exact formulas for the cell face area-weighted normal components are written below. For a vertical cell face (constant ξ):

$$\vec{n}_x = \xi_x \quad \vec{n}_y = \eta_x \quad (3.55)$$

For a horizontal cell face (constant η):

$$\vec{n}_x = \xi_y \quad \vec{n}_y = \eta_y \quad (3.56)$$

If we interpret the flux at a grid cell interface as being based on the flux differences resulting from a Riemann problem in an approximately one-dimensional manner as shown in Fig. 3.15, then the representative fluxes may be expressed as:

$$\hat{E}_{i+\frac{1}{2},j} = \hat{E}_{i,j} + \Delta \hat{E}_{i+\frac{1}{2},j}^- \quad (3.57)$$

or alternately as:

$$\hat{E}_{i+\frac{1}{2},j} = \hat{E}_{i+1,j} - \Delta \hat{E}_{i+\frac{1}{2},j}^+ \quad (3.58)$$

GCGTVD Grid-Centered Finite-Volume

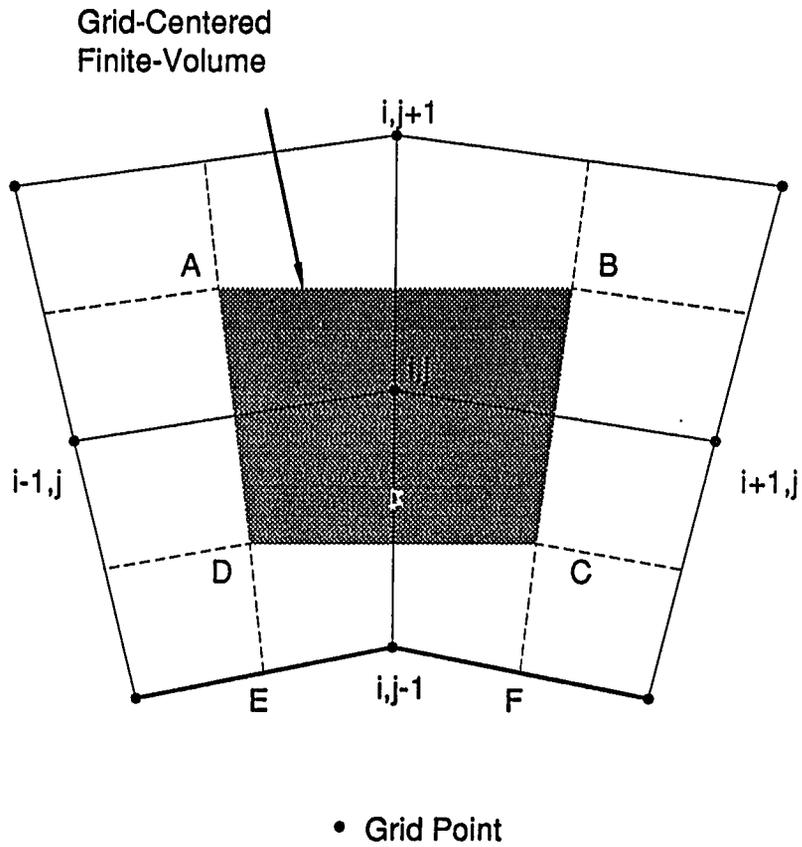


Figure 3.14: Grid-centered finite-volume arrangement for GCGTVD scheme

Grid Cell Interface Riemann Problem

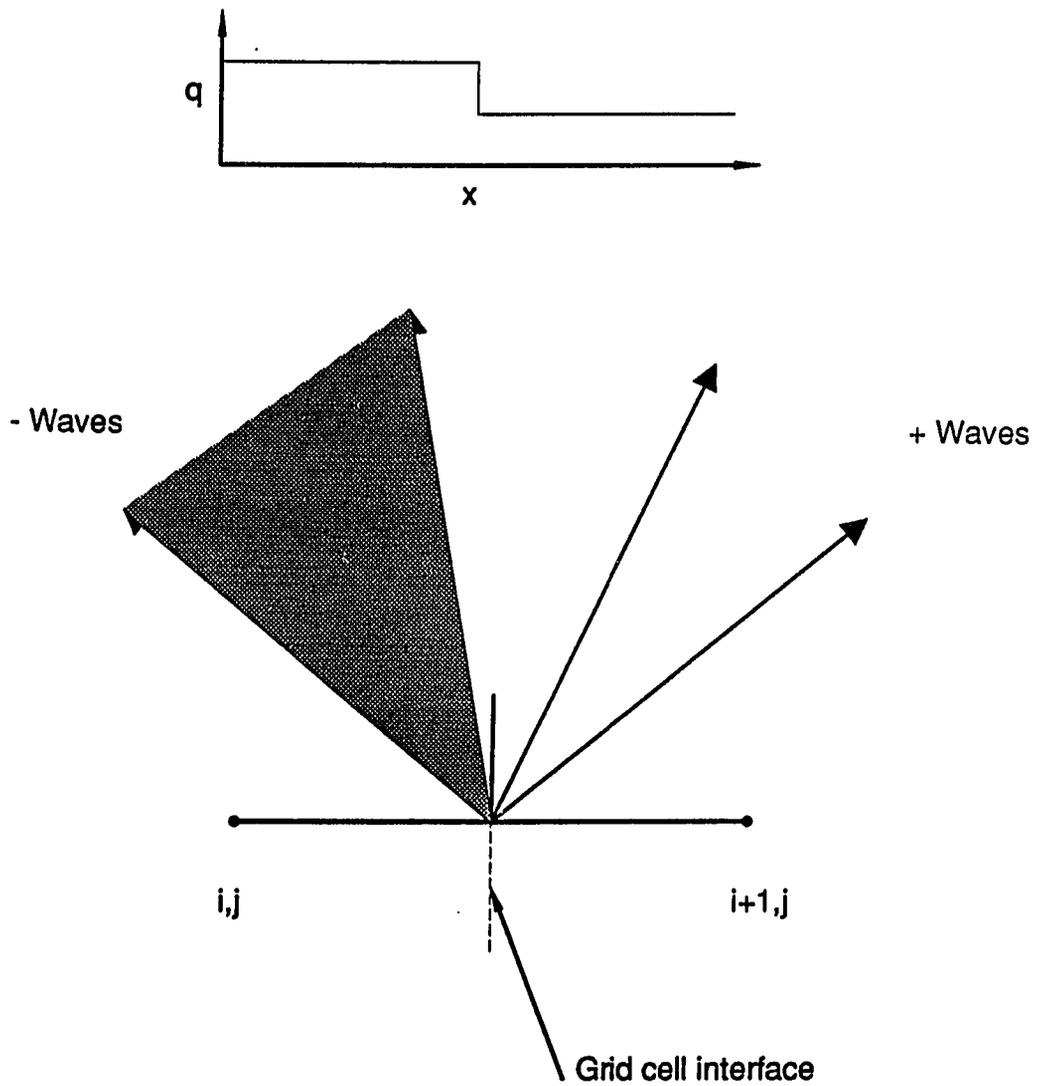


Figure 3.15: Riemann problem at a grid cell interface for the GCGTVD scheme

Roes' approximate Riemann solver determines the flux differences by

$$\Delta \hat{E} = \bar{A} \Delta \hat{Q} \quad (3.59)$$

where \bar{A} is the flux Jacobian based on some averaged state between grid points.

Clearly then, \bar{A} can be diagonalized into eigenvalues and eigenvectors as:

$$\bar{A} = \bar{R}_\xi \bar{\Lambda}_\xi \bar{R}_\xi^{-1} \quad (3.60)$$

where the sign of the eigenvalues in $\bar{\Lambda}_\xi$ determines whether the contribution to the flux travels along positive (+) or negative (-) waves.

Based on this observation, the flux formulas listed above may be combined to yield the compact expression:

$$\hat{E}_{i+\frac{1}{2},j} = \frac{1}{2}(\hat{E}_{i+1,j} + \hat{E}_{i,j} - [\bar{R}_\xi |\hat{\Lambda}_\xi| \bar{R}_\xi^{-1} \Delta \hat{Q}]_{i+\frac{1}{2},j}) \quad (3.61)$$

The inviscid flux values are calculated as:

$$\begin{aligned} (\hat{E}_{inv})_{i+\frac{1}{2},j} &= \frac{1}{2}((\hat{E}_{inv})_{i+\frac{1}{2},j}^L + (\hat{E}_{inv})_{i+\frac{1}{2},j}^R \\ &- (\hat{R}_\xi)_{i+\frac{1}{2},j} \Psi[(\hat{\Lambda}_\xi)_{i+\frac{1}{2},j}] (\hat{R}_\xi^{-1})_{i+\frac{1}{2},j} \Delta \hat{Q}_{i+\frac{1}{2},j}) \end{aligned} \quad (3.62)$$

where:

$$\begin{aligned} \Delta \hat{Q}_{i+\frac{1}{2},j} &= \frac{(Q_{i+\frac{1}{2},j}^R - Q_{i+\frac{1}{2},j}^L)}{0.5(J_{i+1,j} + J_{i,j})} \\ \Psi(\hat{\Lambda}) &= \Psi \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 \\ 0 & 0 & 0 & \lambda_4 \end{pmatrix} = \text{diag}(\lambda_k^*)_{k=1,4} \end{aligned}$$

$$\lambda_k^* = \begin{cases} |\lambda_k|, & |\lambda_k| > \epsilon \\ \frac{1}{2}(\lambda_k^2/\epsilon + \epsilon) & |\lambda_k| < \epsilon \end{cases} \quad (3.63)$$

where λ_k represents the eigenvalues of the inviscid flux Jacobian matrix decomposition (see Eq. (2.32)), and the term $diag(\lambda_k)$ is a diagonal matrix containing λ_k as its diagonal entries in the order shown above. The diagonal matrix operator $\Psi(\hat{\Lambda})$ forces the solution to satisfy an entropy inequality and suppresses the formation of expansion shocks [19]. A value of $\epsilon = 0.001$ was used for all calculations.

It should be noted that this flux function is consistent, such that:

$$\hat{E}(\hat{Q}_{i+2,j}, \hat{Q}_{i+1,j}, \hat{Q}_{i,j}, \hat{Q}_{i-1,j}, \dots) = \hat{E}(\hat{Q})$$

for any number of contributing sources:

$$\hat{Q}_{i+2,j} = \hat{Q}_{i+1,j} = \hat{Q}_{i,j} = \hat{Q}_{i-1,j} = \hat{Q}$$

The fluxes $(\hat{E}_{inv})_{i+\frac{1}{2},j}^{R/L}$ are determined using values of the dependent flow variables $Q_{i+\frac{1}{2},j}^{R/L}$, that are interpolated in an upwind manner to either side (R/L) of the interface at $i+\frac{1}{2},j$ (see Fig. 3.15). A first order scheme results when $Q_{i+\frac{1}{2},j}^L = Q_{i,j}$ and $Q_{i+\frac{1}{2},j}^R = Q_{i+1,j}$. Higher overall accuracy is obtained by using more accurate upwind biased interpolation procedures to obtain these values. This technique for achieving higher accuracy, often referred to as the MUSCL approach, was also used by Van Leer [17] for a flux vector splitting scheme, and by Chakravarthy [22] for a flux difference splitting scheme.

The matrices $(\hat{R}_\xi)_{i+\frac{1}{2},j}$, $(\hat{\Lambda}_\xi)_{i+\frac{1}{2},j}$, and $(\hat{R}_\xi^{-1})_{i+\frac{1}{2},j}$ are constructed using the (R/L) interpolated variables and the symmetric averaging procedure derived by Roe

[18].

$$\begin{aligned}
\rho_{i+\frac{1}{2},j} &= \frac{(\sqrt{\rho_{i+\frac{1}{2},j}^R} \rho_{i+\frac{1}{2},j}^R + \sqrt{\rho_{i+\frac{1}{2},j}^L} \rho_{i+\frac{1}{2},j}^L)}{\sqrt{\rho_{i+\frac{1}{2},j}^R} + \sqrt{\rho_{i+\frac{1}{2},j}^L}} \\
u_{i+\frac{1}{2},j} &= \frac{(\sqrt{\rho_{i+\frac{1}{2},j}^R} u_{i+\frac{1}{2},j}^R + \sqrt{\rho_{i+\frac{1}{2},j}^L} u_{i+\frac{1}{2},j}^L)}{\sqrt{\rho_{i+\frac{1}{2},j}^R} + \sqrt{\rho_{i+\frac{1}{2},j}^L}} \\
v_{i+\frac{1}{2},j} &= \frac{(\sqrt{\rho_{i+\frac{1}{2},j}^R} v_{i+\frac{1}{2},j}^R + \sqrt{\rho_{i+\frac{1}{2},j}^L} v_{i+\frac{1}{2},j}^L)}{\sqrt{\rho_{i+\frac{1}{2},j}^R} + \sqrt{\rho_{i+\frac{1}{2},j}^L}} \\
H_{i+\frac{1}{2},j} &= \frac{(\sqrt{\rho_{i+\frac{1}{2},j}^R} H_{i+\frac{1}{2},j}^R + \sqrt{\rho_{i+\frac{1}{2},j}^L} H_{i+\frac{1}{2},j}^L)}{\sqrt{\rho_{i+\frac{1}{2},j}^R} + \sqrt{\rho_{i+\frac{1}{2},j}^L}}. \tag{3.64}
\end{aligned}$$

The remaining variables are derived from these values.

In this study, the values of $Q_{i+\frac{1}{2},j}^{R/L}$ are determined through the following generally third order accurate flux-limited upwind-biased interpolation procedure:

$$\begin{aligned}
Q_{i+\frac{1}{2},j}^L &= Q_{i,j} + \sum_{k=1,4} \left(\frac{1}{3} \bar{\sigma}_{i+\frac{1}{2}}^k + \frac{1}{6} \bar{\sigma}_{i-\frac{1}{2}}^k \right) w_{i,j}^k \\
Q_{i-\frac{1}{2},j}^R &= Q_{i,j} - \sum_{k=1,4} \left(\frac{1}{3} \bar{\sigma}_{i-\frac{1}{2}}^k + \frac{1}{6} \bar{\sigma}_{i+\frac{1}{2}}^k \right) w_{i,j}^k \tag{3.65}
\end{aligned}$$

$$\begin{aligned}
\bar{\sigma}_{i+\frac{1}{2},j}^k &= \frac{1}{2} (\text{sgn}(\phi_{i+\frac{1}{2},j}^k) + \text{sgn}(\phi_{i-\frac{1}{2},j}^k)) \min[\text{abs}(\phi_{i+\frac{1}{2},j}^k), \text{abs}(\phi_{i-\frac{1}{2},j}^k)] \\
\phi_{i+\frac{1}{2},j}^k &= b_{i,j}^k \Delta \hat{Q}_{i+\frac{1}{2},j}, \quad \text{sgn}(\phi) = \begin{cases} 1, & \text{if } \phi > 0 \\ -1, & \text{if } \phi < 0 \end{cases} \tag{3.66}
\end{aligned}$$

In the above notation, $w_{i,j}^k$ represents the columns of the matrix $\hat{R}_{i,j}$, and $b_{i,j}^k$ represents the rows of the matrix $\hat{R}_{i,j}^{-1}$. The overall term $\sum \bar{\phi}^k w^k$ contains the limited

variables $\tilde{\phi}^k$ and therefore represents a flux limited form of the product $\hat{R}^{-1}\Delta\hat{Q}$. The limiting occurs near points of extrema in the decoupled variable system, and can revert to a locally first order accurate scheme to control oscillations. The limiting is applied in a manner consistent with the derivation of TVD schemes for scalar conservation laws; however, it should be noted that no distinct TVD property has been established for systems of conservation laws.

A characteristic-type decomposition is applied in the interpolation to allow limiting on characteristic, rather than primitive variables. This is consistent with the nonoscillatory formulations developed for scalar conservation laws and allows the limiters to act upon the individual waves of the hyperbolic system. It is also possible to utilize direct interpolation of the primitive flow variables if the limiting functions are carefully designed to minimize oscillations. This procedure was tested several times during the development of this code, and while the results were not unsatisfactory, the characteristic interpolation scheme consistently resulted in a more uniform nonoscillatory solution.

Nonoscillatory flux constructions of this type are documented in [22]. A similar construction is used for the flux term $\hat{F}_{i+\frac{1}{2}}$ by utilizing the appropriate terms, derivatives and interpolation in the η direction. All viscous fluxes are determined through standard central differencing of the flux terms at each mid-point interface.

For example:

$$(u_{\xi})_{i+\frac{1}{2},j} = u_{i+1,j} - u_{i,j}$$

$$(u_{\eta})_{i+\frac{1}{2},j} = \frac{1}{4}((u_{i+1,j+1} - u_{i+1,j-1}) + (u_{i,j+1} - u_{i,j-1})) \quad (3.67)$$

Since the underlying differencing scheme for the inviscid fluxes is upwinded,

the coefficient matrix which results when these terms are linearized is diagonally dominant in nature, and the matrix system can thus be solved through relaxation, rather than direct elimination [67]. This permits use of a number of additional solution techniques not previously feasible for time-marching calculations. Although it is possible to construct viscous fluxes which also contribute to the diagonal dominance of the linearized coefficient matrix [21], the present results were obtained by using a first order linearization of the inviscid fluxes to obtain the implicit coefficient matrix, and lagging the viscous fluxes during the relaxation iteration. Several linearization procedures are discussed by Yee [70]. A Newton-type relaxation procedure is used to determine the flow variables $\hat{Q}_{i,j}^{n+1}$ at the new time level as:

$$\begin{aligned}
& [I + \hat{G}_{i+\frac{1}{2},j} - \hat{G}_{i-\frac{1}{2},j} + \hat{H}_{i,j+\frac{1}{2}} - \hat{H}_{i,j-\frac{1}{2}}][\Delta\hat{Q}_{i,j}]^{(l+1)} = \\
& \quad -[\hat{Q}_{i,j}^l - \hat{Q}_{i,j}^n + \Delta t* \\
& \quad \left(\frac{((\hat{E}_{inv})_{i+\frac{1}{2},j}^* - (\hat{E}_{inv})_{i-\frac{1}{2},j}^*)}{\Delta\xi} + \frac{((\hat{F}_{inv})_{i,j+\frac{1}{2}}^* - (\hat{F}_{inv})_{i,j-\frac{1}{2}}^*)}{\Delta\eta} + \right. \\
& \quad \left. \frac{((\hat{E}_{vis})_{i+\frac{1}{2},j}^* - (\hat{E}_{vis})_{i-\frac{1}{2},j}^*)}{\Delta\xi} + \frac{((\hat{F}_{vis})_{i,j+\frac{1}{2}}^* - (\hat{F}_{vis})_{i,j-\frac{1}{2}}^*)}{\Delta\eta} \right)] \quad (3.68)
\end{aligned}$$

The implicit operator can take many forms, many of which are discussed by Yee [70].

Perhaps the most direct linearization may be derived as:

$$\begin{aligned}
& \hat{G}_{i+\frac{1}{2},j}[\Delta\hat{Q}_{i,j}]^{l+1} = \\
& \left[(\hat{R}_\xi)_{i+\frac{1}{2},j} (\hat{\Lambda}_\xi^-)_{i+\frac{1}{2},j} (\hat{R}_\xi^{-1})_{i+\frac{1}{2},j} \right] \left[(\Delta\hat{Q}_{i+1,j})^{l+1} - (\Delta\hat{Q}_{i,j})^{l+1} \right]
\end{aligned}$$

$$\hat{G}_{i-\frac{1}{2},j}[\Delta\hat{Q}_{i,j}]^{l+1} = \left[(\hat{R}_\xi)_{i-\frac{1}{2},j} (\hat{\Lambda}_\xi^+)_{i-\frac{1}{2},j} (\hat{R}_\xi^{-1})_{i-\frac{1}{2},j} \right] \left[(\Delta\hat{Q}_{i,j})^{l+1} - (\Delta\hat{Q}_{i-1,j})^{l+1} \right] \quad (3.69)$$

where l is the iteration index for the Newton relaxation procedure. Here, $(\hat{\Lambda}_\xi^+)$ and $(\hat{\Lambda}_\xi^-)$ represent the eigenvector matrices containing only the positive and negative eigenvalues, respectively. Similar relations hold for $\hat{H}_{j+/-\frac{1}{2}}$ by taking differences in the η direction. This linearization is suitable for many calculations, but the interest in this study was the application of conjugate gradient solvers for compressible flow predictions. Although this approach contains some diagonally dominant quality (in a matrix sense), conjugate gradient solvers for block diagonally dominant matrix systems are not well defined. In fact, a suitable conjugate gradient solver which would consistently solve this type of equation could not be found. Instead, a simpler, ad hoc linearization approach was adopted and may be expressed as:

$$\begin{aligned} \hat{G}_{i+\frac{1}{2},j}[\Delta\hat{Q}_{i,j}]^{l+1} &= \text{diag}[\max(\Psi(\lambda_k^-)_{k=1,4})]_{i+\frac{1}{2},j} ([\Delta\hat{Q}_{i+1,j}]^{l+1} - [\Delta\hat{Q}_{i,j}]^{l+1}) \\ \hat{G}_{i-\frac{1}{2},j}[\Delta\hat{Q}_{i,j}]^{l+1} &= \text{diag}[\max(\Psi(\lambda_k^+)_{k=1,4})]_{i-\frac{1}{2},j} ([\Delta\hat{Q}_{i,j}]^{l+1} - [\Delta\hat{Q}_{i-1,j}]^{l+1}) \end{aligned} \quad (3.70)$$

Similar relations hold for $\hat{H}_{j+/-\frac{1}{2}}$ by taking differences in the η direction. The terms $\lambda^+/-$ are formed by utilizing only the positive (+) and negative (-) eigenvalues, respectively.

$$\lambda_k^+ = 0.5(\lambda_k + \text{abs}(\lambda_k)) \quad \lambda_k^- = 0.5(\lambda_k - \text{abs}(\lambda_k)) \quad (3.71)$$

In Eqs. (3.68-3.70) above:

$$[\Delta\hat{Q}]_{i,j}^{l+1} = [\hat{Q}_{i,j}^{n+1}]^{l+1} - [\hat{Q}_{i,j}^n]^l \quad (3.72)$$

where the * superscript indicates the most recent value during the Newton relaxation iteration. In a physical sense, this approach attempts to dictate how far the influence of the local residual (*RHS*) is spread throughout the local grid system. A local residual with a large magnitude eigenvalue is effectively more dominant in the implicit scheme than a smaller magnitude eigenvalue. This scheme does not have the desirable robust quality of the full matrix linearization, but does permit a wider variety of iterative solvers for the solution of the implicit equation. Second order accurate linearization of the flux values requires exact knowledge of the flux Jacobians $\hat{A}_{i+\frac{1}{2},j}$ and $\hat{B}_{i+\frac{1}{2},j}$ and is rather difficult to obtain.

The overall solution proceeds by iteratively solving the implicit equations during each time step to remove the nonlinearities of the implicit flux calculation. The inner iteration proceeds until changes between iterations are negligibly small, then move on to the next time step. A slight underrelaxation is required during this iterative cycle for optimal convergence.

For each iteration of the above linearization loop, a linear system of equations of the form:

$$M^l[\Delta\hat{Q}_{i,j}]^{l+1} = RHS^l \quad (3.73)$$

must be solved, where M is an irregular, nonsymmetric matrix. Other investigators have used approximate factorization, iterative methods, and sparse matrix solvers to update the dependent variables in the relaxation iteration loop. In this study, a generalized conjugate gradient-type iteration procedure is used. The generalized conjugate gradient procedure is free of factorization errors and takes advantage of the diagonal dominance of the coefficient matrix. In addition, for poorly conditioned matrices, the

conjugate gradient approach will often have better convergence characteristics than other iterative solvers. The conjugate gradient method was originally developed as a multistep direct method for the solution of linear symmetric positive definite matrices by Hestenes and Stiefel [101]. It was quickly realized that a reasonable solution was obtained after a relatively small number of steps, and the method gained popularity as an iterative technique. The procedure was later further extended to nonsymmetric matrices with positive definite symmetric part by Concus and Golub [102]. The class of descent methods commonly referred to as generalized conjugate gradient used in this study was described by Elman [103]. In this case, the descent direction vector is chosen in a manner that is based on the method of steepest descent, since a strictly conjugate relationship is somewhat difficult to achieve for nonsymmetric matrices. The conjugate gradient algorithm may be expected to yield faster convergence for linear systems. In addition, the algorithm is ideally suited for vector and parallel computer architectures. Although the M matrix is somewhat difficult to handle due to a high concentration of nonuniformly ordered coefficients, it only appears in the solution scheme as a matrix-vector product, thus the abnormalities in the matrix are accounted for through specific programming of the known irregular matrix structure, and only the non-zero matrix coefficients need to be stored. Thus another advantage of the relaxation procedure is that complicated grid arrangements may be handled implicitly more easily than with approximately factored schemes. The extension of this procedure to three dimensional problems is straightforward.

The generalized conjugate gradient solution proceeds as follows:

$$\text{To solve: } M^l[\Delta\hat{Q}]^{l+1} = RHS^l$$

Choose: $([\Delta\hat{Q}]^{l+1})^{(0)} = 0$

Compute: $r^{(0)} = RHS^l - M^l([\Delta\hat{Q}_{i,j}]^{l+1})^{(0)}$

Set: $s^{(0)} = r^{(0)}$

For $k = 1$ step 1 until convergence DO:

$$\begin{aligned}
 z^{(k)} &= \frac{(r^{(k-1)} \cdot M^l s^{(k-1)})}{(M^l s^{(k-1)} \cdot M^l s^{(k-1)})} \\
 ([\Delta\hat{Q}]^{l+1})^{(k)} &= ([\Delta\hat{Q}]^{l+1})^{(k-1)} + z^{(k)} s^{(k-1)} \\
 r^{(k)} &= r^{(k-1)} - z^{(k)} \cdot M^l s^{(k-1)} \\
 s^{(k)} &= r^{(k)} \tag{3.74}
 \end{aligned}$$

The choice of the vector $s^{(k)}$ likens the scheme to the method of steepest descent. For practical calculations, the iteration is terminated before convergence to allow the fluxes and coefficients to be updated. Typically, for most calculations, only 10-20 iterations were used, then coefficients and *RHS* terms were updated. Approximately 1-5 linearization iterations are required at each time step to fully converge the implicit flux values. For steady flows, the linearization loop need not be fully converged, although numerical tests indicated that a larger time step could be tolerated for steady flow calculations when the implicit flux values were fully converged.

The overall scheme is highly vectorizable and requires only slightly more storage than the vectorized AFLBI algorithm. If the relaxation iterations are sufficiently converged, the scheme is first order time accurate and generally third order spatially accurate $[O(\Delta t, \Delta x^3)]$. (The scheme is generally third order accurate since the effect of the limiter is to reduce the scheme to first order accuracy in regions of discontinuous

flow.) It is possible to upgrade the scheme to second order time accuracy if necessary by utilizing a higher order time difference, although this procedure does not coincide with the convolution integral formulation of the general upwind difference scheme [19].

CHAPTER 4. NUMERICAL RESULTS

Several steady and unsteady flow test cases are presented below to demonstrate the accuracy, efficiency, and reliability of the four two-dimensional time-marching algorithms previously described. In each case, identical grids and initial conditions were utilized for each of the codes. Most calculations were performed on the Cray X-MP 48 computer at the University of Illinois, although some additional calculations were also performed on the Iowa State University DEC Vax 11-780 and NAS AS-9160, and a Silicon Graphics 4D/240-GTX workstation. Impulsively started cylinder flow calculations were performed on Cray-2 and Cray Y-MP computers. This variety of devices served to verify the machine independence of the codes. Every reasonable effort was made to construct each code to operate at maximum efficiency on each machine.

The time-marching algorithms were applied for the calculation of steady flows by specifying time-independent boundary conditions, and marching the solution in time until no significant changes occurred between time steps. For steady state calculations, the solution was deemed converged when the maximum relative difference in pressure at any grid point satisfied the criteria:

$$\max_{\substack{i=1,imax \\ j=1,jmax}} \left(\frac{|p_{i,j}^{n+1} - p_{i,j}^n|}{p_{i,j}^{n+1}} \right) \leq (10)^{-4} \quad (4.1)$$

unless otherwise specified. Although many techniques are available to accelerate convergence for steady flow calculations, the primary goal of the overall research was the eventual prediction of unsteady flows; therefore, the only convergence enhancement was the application of the local maximum time step at every point.

In addition to the test cases described in detail in the following sections, a number of steady and unsteady flows were calculated initially to verify the formulation and range of application for each code. No results for these cases are presented here; however, some discussion of each case is warranted. The conservative nature of each code was tested by simulating a simple uniform flow. (Nonconservative formulations often cannot accurately reproduce a uniform flow.) Each code accurately reproduced a uniform flow and no indications of spurious nonphysical source terms were observed. Steady state viscous solutions were examined by calculating the developing viscous flow in a constant area two-dimensional planar channel. Each code accurately reproduced the parabolic fully developed velocity profile, although the relatively coarse grids and artificial damping resulted in an excessive negative pressure gradient in the flow direction. Attempts to compare the flow in the entrance region of the channel with other calculations were thwarted by the fact that a nonuniform inlet profile is established by the characteristic inlet boundary algorithm. Although this is likely to be consistent with actual experimental observations, other calculations of this type are generally initiated with a uniform inlet profile. Steady state shock-capturing performance was examined in the two-dimensional solution of a converging-diverging Laval nozzle flow. Approximate damping values were established to minimize oscillations near discontinuities. An interesting result of this study was the observation that the

Table 4.1: Steady flow verification test cases

Case	Description
1.	Critical inviscid flow about an isolated cylinder. (Inlet Mach number=0.4, 85x30 grid)
2.	Supercritical inviscid flow about an isolated cylinder. (Inlet Mach number=0.45, 85x30 grid)
3.	Transonic inviscid flow about a low stagger turbine cascade. (Exit Mach number=1.1, 91x9 and 171x15 grids)
4.	Transonic inviscid flow about a high turning turbine cascade. (Exit Mach number=1.1, 157x15 grid)
5.	Transonic laminar viscous flow about a high turning turbine cascade. Exit Mach Number=1.1, 157x30 grid)
6.	Subsonic laminar viscous flow over a flat plate. (Mach Number=0.2, 121x51 grid)

nonlinear damping scheme in the AFLBI algorithm produced shocks equally sharp as the TVD scheme, although the shock locations differed by one grid point. Preliminary transient tests were performed by solving a two-dimensional version of the one-dimensional Riemann problem discussed in Chapter 3. In this case, discontinuous initial conditions were specified along one coordinate of a Cartesian grid, while initial conditions along the remaining coordinate were constant. The one-dimensional predictions were essentially duplicated by the two-dimensional codes.

Table 4.1 lists the steady flow verification test cases and the grids tested which are discussed in detail in the following sections. Cases 1 and 2 were used to verify the overall inviscid formulation and shock-capturing capabilities of each code. Case

Table 4.2: Unsteady flow verification test cases

Case	Description
1.	Laminar Unsteady Flow near an Oscillating Flat Plate (41x51 grid)
2.	Laminar Unsteady Flow about an Impulsively Started Cylinder (Mach number=0.10, 361x99 grid, Re=3000, 9500)

3 verified the individual formulations for cascade flow calculations, and formed the basis for a comparison of predictions for a design-type calculation. Predictions were performed for two different grid sizes to evaluate this effect on the solutions. Cases 4 and 5 compared both inviscid and viscous flow predictions for a turbomachinery cascade geometry. Case 6 provided a detailed viscous flow verification of both velocity and temperature profiles, as well as viscous flow capabilities for H-type grids.

Table 4.2 lists the unsteady flow test cases performed using the four time-marching algorithms. Unsteady flow calculations were terminated when a time-periodic solution was observed or at the end of a specified finite time interval.

Case 1 was used as a verification of the time-dependent numerics of each of the codes and to estimate grid sizes necessary for unsteady viscous flow predictions. Case 2 provides a detailed comparison of numerical predictions and experimental data for a complicated unsteady viscous flow.

A description of each test case and the predicted results are given in the sections which follow.

Steady Inviscid Flow about an Isolated Cylinder

As the first test of the two-dimensional codes, the steady inviscid flow over an isolated cylinder was computed. The isolated cylinder was approximated by a zero angle of attack cascade of cylinders with a relatively large interblade spacing (6 cylinder diameters) to minimize the aerodynamic interference between cylinders.

Results from two inlet Mach numbers were compared. The first, with an inlet Mach number of 0.4, results in a peak Mach number on the surface of the cylinder near 1.0, and therefore this condition is referred to as the critical flow case. In the second case, the inlet Mach number is increased to 0.45. In this case, the flow is transonic, and a symmetric shock wave pattern develops on the leeward side of the cylinder. The second flow is referred to as the supercritical case.

A grid utilizing 85 normals extending from the surface of the cylinder and 30 contours surrounding the cylinder was used for all of the calculations, and is pictured in Fig. 4.1. Calculations were performed using damping values suggested by Delaney [63] ($c_1 = 0.02, c_2 = 0.5$) and Pulliam [66] ($c_1 = 0.25, c_2 = 0.007$). Although "fine-tuning" the coefficients can slightly improve the solutions, the present results are considered typical of those obtained from each code.

The primary emphasis of this test case was to verify the numerical formulation of each algorithm. In this respect, predicted results were compared with other calculations rather than experimental data. Therefore, in order to validate the predictions, a comparison was made with results from a third order accurate TVD-type Euler code developed by Chakravarthy [22] utilizing a highly refined grid near the cylinder surface.

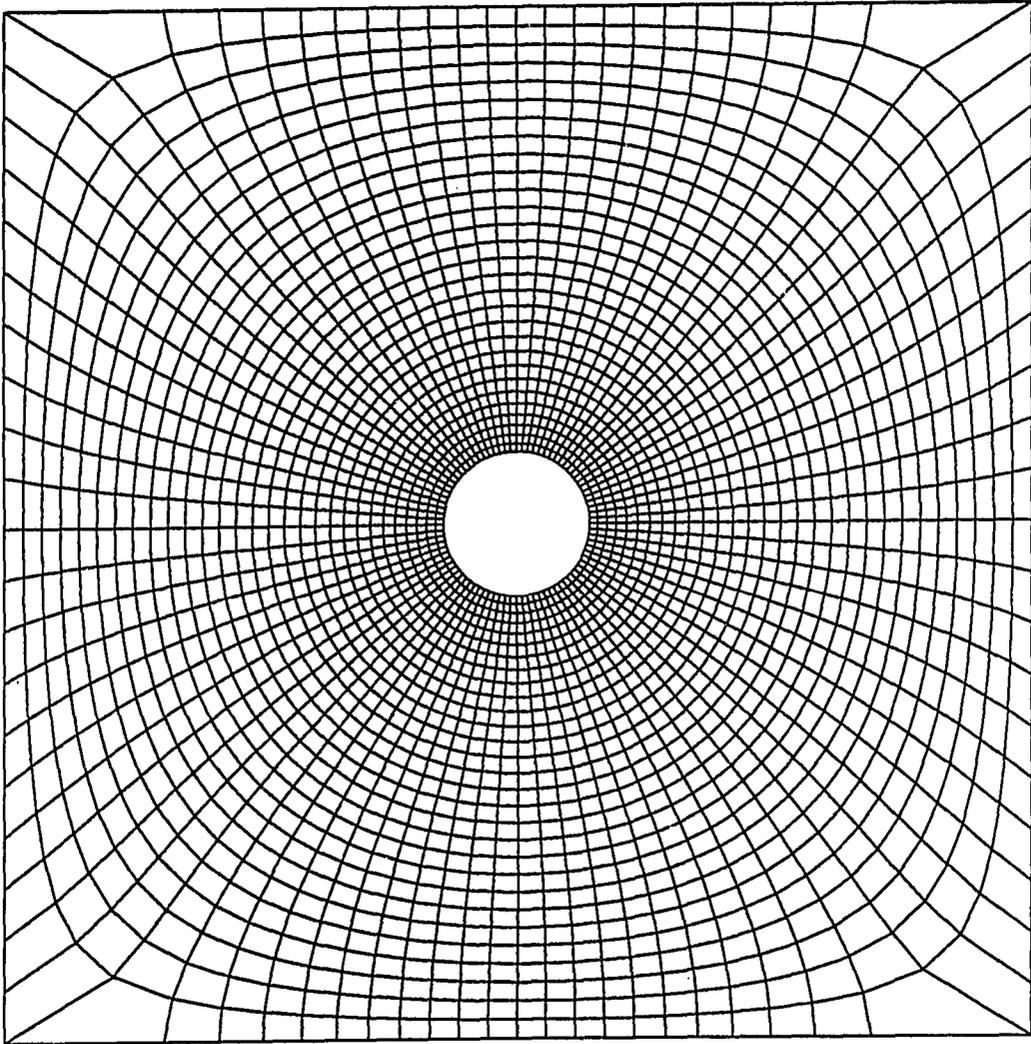


Figure 4.1: 85x30 grid for inviscid isolated cylinder flow calculations

Critical Flow Case

A comparison of the predicted cylinder surface Mach number distribution from each of the four time-marching algorithms with the TVD prediction of [22] for the critical flow case is given in Fig. 4.2. In each case, excellent agreement between all predictions is found on the windward surface of the cylinder. All of the present predictions indicate a peak Mach number near 1.0, located near the 90 degree point on the cylinder. The discrepancy between predictions on the downwind portion of the cylinder is attributed to spurious entropy generation on the cylinder surface which is then convected downstream. This phenomenon is caused by lack of a refined grid in the present calculations, locally excessive damping, and small errors caused by numerical application of the inviscid surface boundary conditions. Aside from this small abnormality, each code has adequately captured the significant features of the flow.

Supercritical Flow Case

The supercritical case affords an examination of the shock capturing capabilities of each algorithm. A comparison of the predicted cylinder surface Mach number distributions for the supercritical case is given in Fig. 4.3. Again, the small discrepancy on the downwind surface of the cylinder can be at least partially attributed to small numerical errors resulting from the coarse grid and small differences in predicted shock intensity. The effects of the shock wave on the predicted results is evident and clearly visible near the 100 degree point of the cylinder. For the hopscotch and Runge-Kutta predictions in Fig. 4.3 the peak Mach number is slightly underpredicted. This is

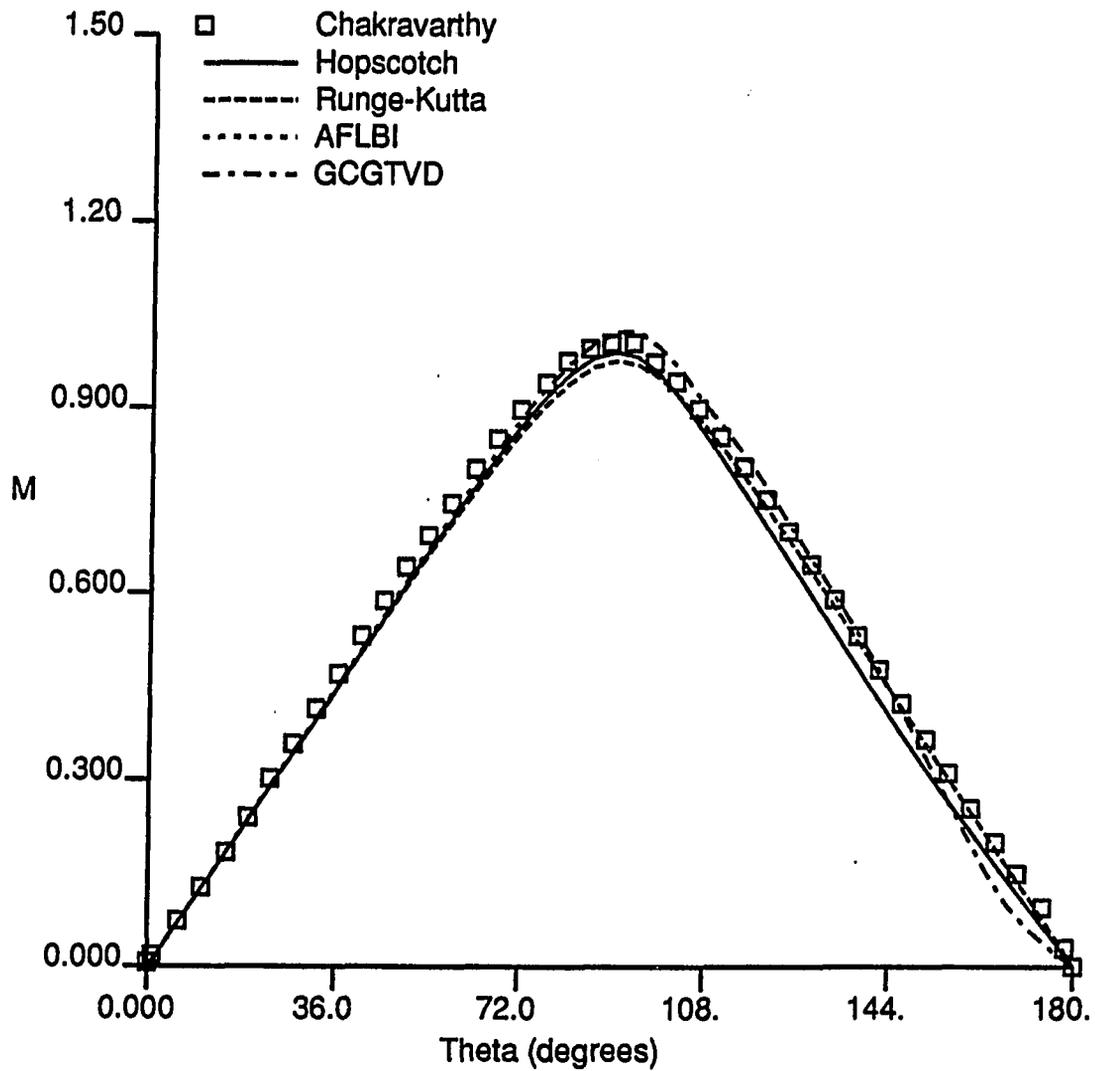
Inviscid Cylinder Flow $M = 0.4$ 

Figure 4.2: Comparison of the predicted cylinder surface Mach number distributions for critical inviscid isolated cylinder flow

thought to be due to the use of the product fourth order dissipation (second-order dissipation with a second derivative coefficient), which tends to significantly smear shock waves, and thus clips the peak value. A slight improvement is noted for the nonlinear damping scheme used in the AFLBI code. The sharpest shock (based on the number of grid points required for the shock transition) for this case was predicted by the GCGTVD scheme. This feature is even more apparent in the predicted Mach number contour plots given in Figs. 4.4-4.7. Although the shock wave is visible in all four calculations, the sharpest representation is given by the upwind nonoscillatory scheme. This excellent resolution is even more amazing considering the overall coarseness of the grid used in the present calculations, and is typical of the outstanding shock-capturing quality of flux-difference split algorithms.

A comparison of computational run time for each of the four codes is given in Table 4.3 for the supercritical flow case. All runs and central processor unit (CPU) times are for calculations performed on a Cray XMP-48 supercomputer. Although each of the codes has to a certain degree been optimized for use on vector processors, the modular nature of the codes inhibits optimal performance for certain subroutines. The CPU times obtained varied considerably for back to back identical runs. This was due to the crudeness of the timing algorithm available during the time these calculations were made. With this in mind, a direct comparison of CPU times between codes must be interpreted with this deficiency in mind. In these initial calculations it appears that the central difference schemes all execute with nearly the same efficiency, with a slight edge given to the hopscotch method. Although the GCGTVD code

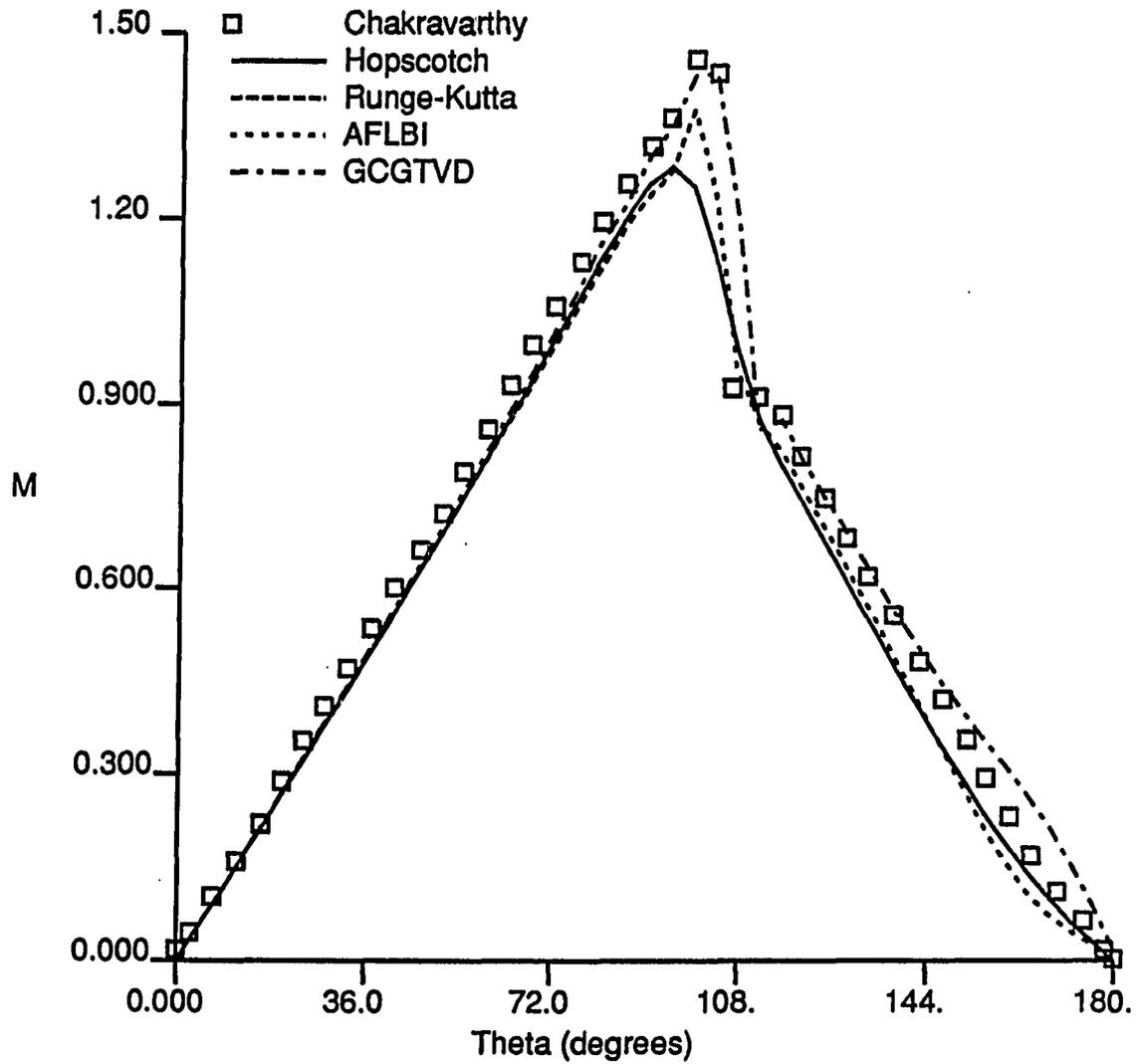
Inviscid Cylinder Flow $M = 0.45$ 

Figure 4.3: Comparison of the predicted cylinder surface Mach number distributions for supercritical inviscid isolated cylinder flow

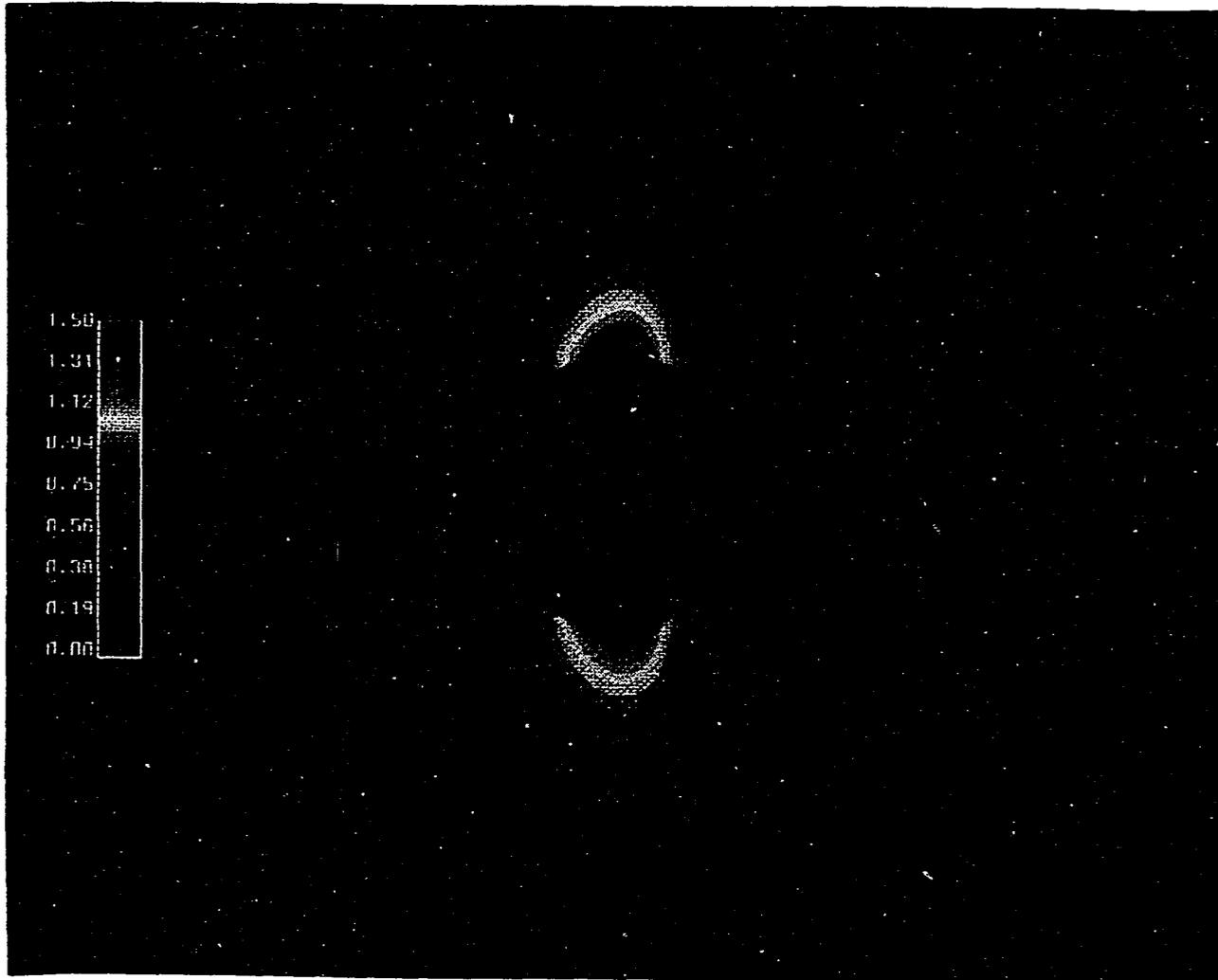


Figure 4.4: Predicted hopscotch Mach number contours for supercritical inviscid isolated cylinder flow

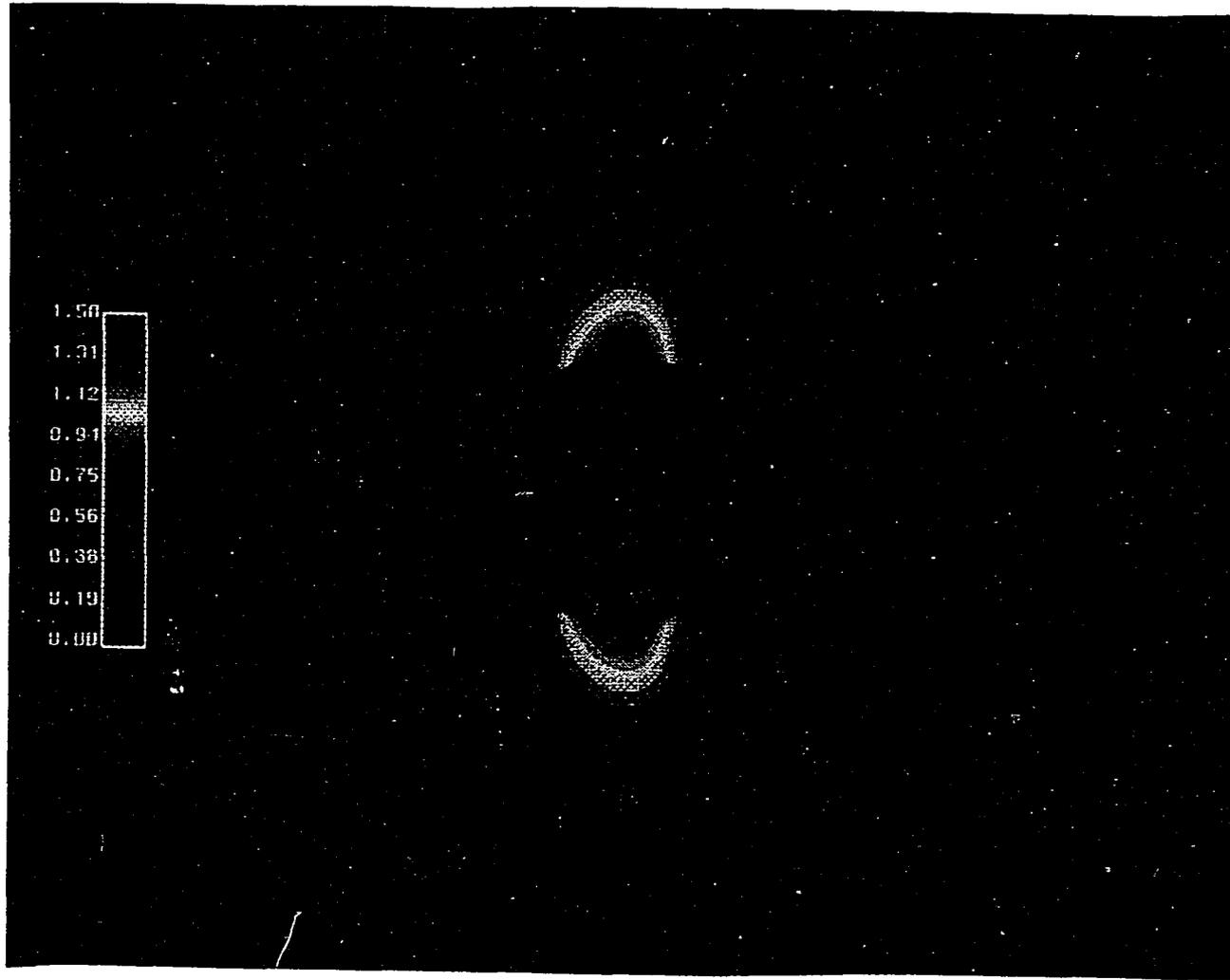


Figure 4.5: Predicted Runge-Kutta Mach number contours for supercritical inviscid isolated cylinder flow

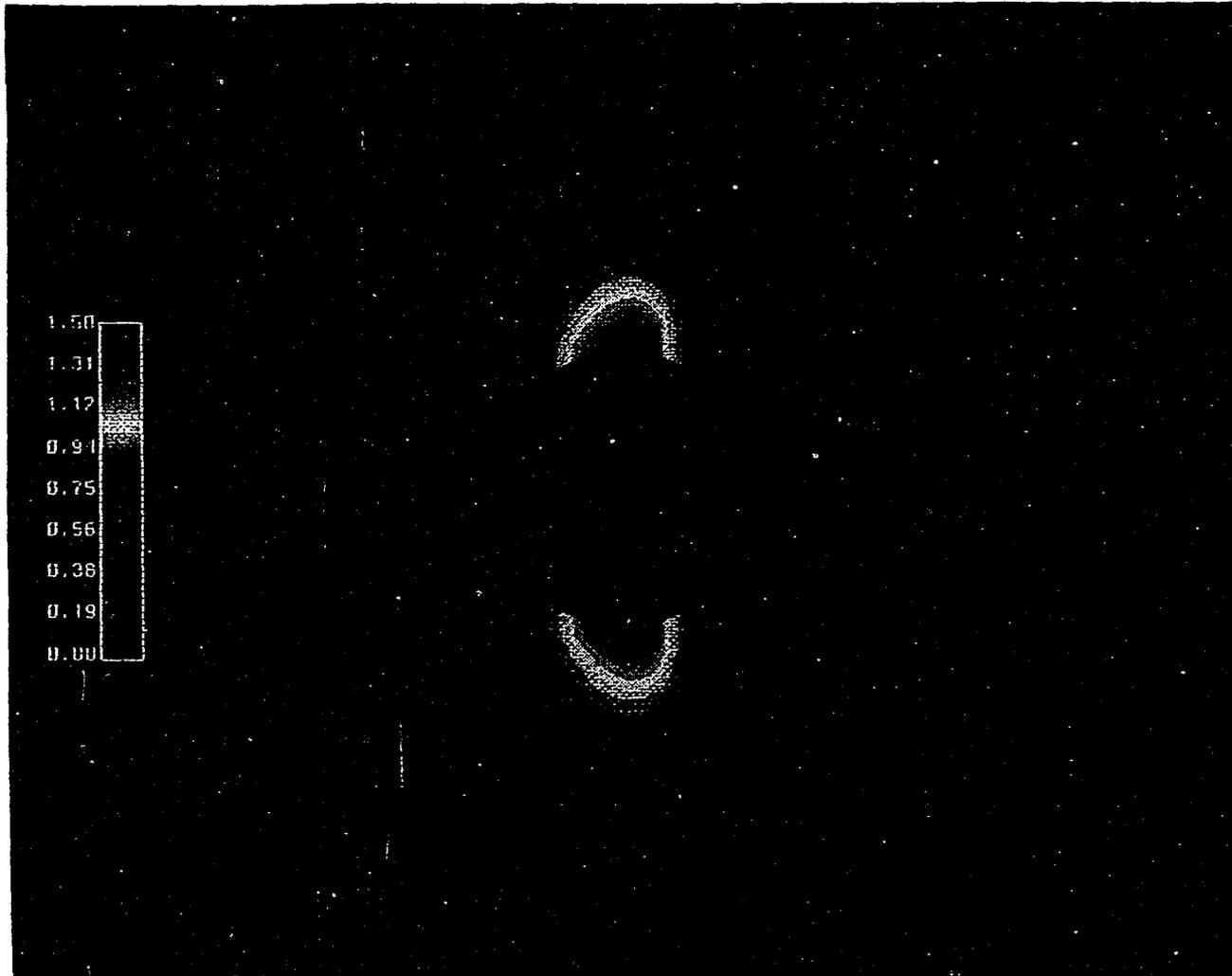


Figure 4.6: Predicted AFLBI Mach number contours for supercritical inviscid isolated cylinder flow

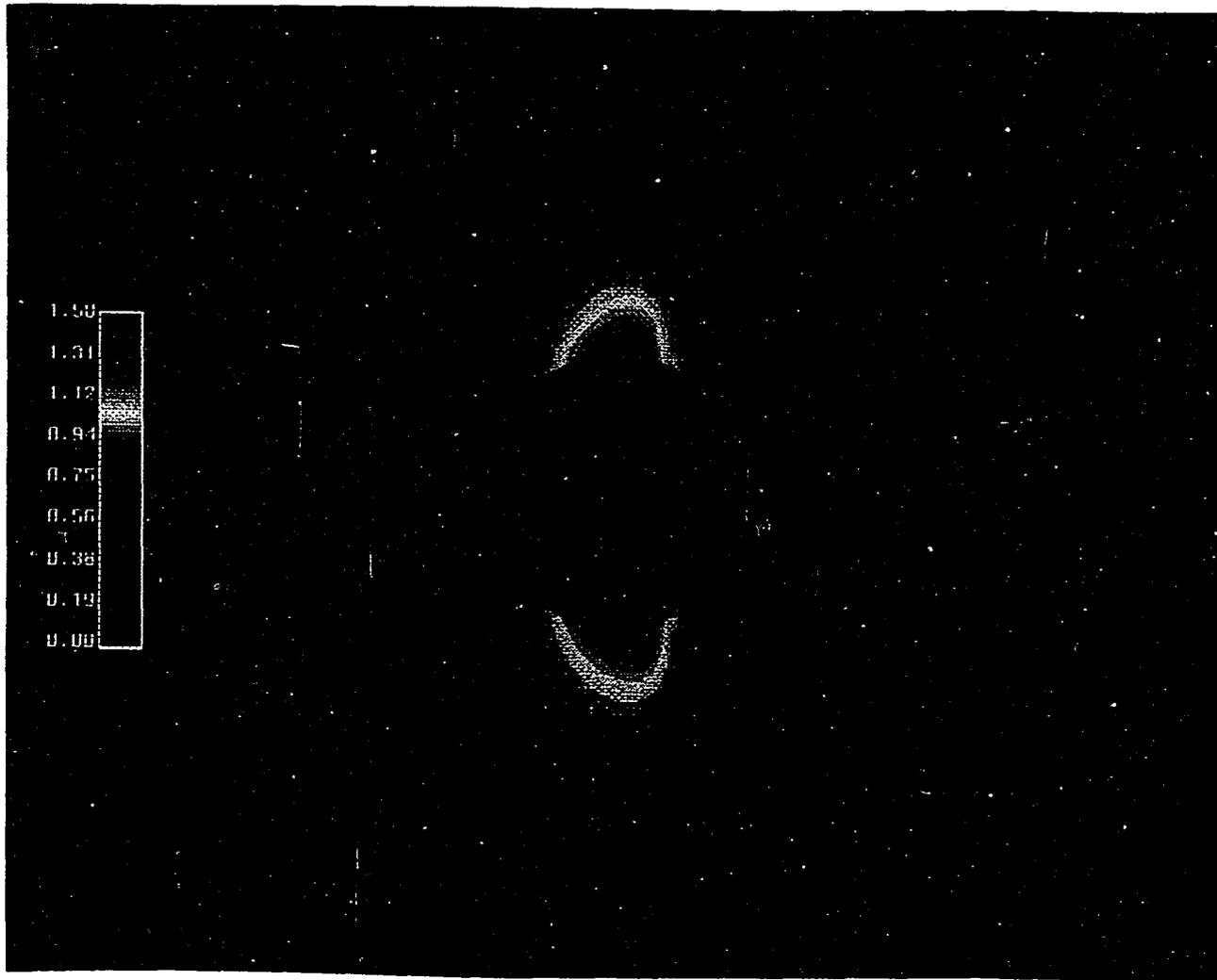


Figure 4.7: Predicted GCGTVD Mach number contours for supercritical inviscid isolated cylinder flow

Table 4.3: Supercritical inviscid isolated cylinder CPU time comparison

Method	CFL Number	Iterations for Convergence	CPU seconds for Convergence	CPU seconds Grid Point / It.
Hopscotch	1.0	1745	45.99	1.03×10^{-5}
Runge-Kutta	2.5	1195	57.66	1.88×10^{-5}
AFLBI	8.0	320	57.09	6.99×10^{-5}
GCGTVD	8.0	305	93.45	12.42×10^{-5}

requires about 60 percent more time than the central-difference codes, further testing and enhancement of the generalized conjugate gradient solution procedure is expected to reduce this margin.

Steady Inviscid Flow Through a Low Stagger Turbine Cascade

The second inviscid flow case tested was the low stagger turbine cascade tested in [104]. This case was chosen to demonstrate the potential for analyzing turbomachinery blade row flowfields. The cascade geometry and initial calculation grid utilizing 91 normals and 9 contours is shown in Fig. 4.8. This low stagger cascade was tested over a range of expansion ratios and eventually compared to similar test results for a high stagger cascade. In this study, calculations were performed for an exit Mach number of 1.1. The inlet flow angle was fixed at 31 degrees. Figure 4.9 compares the predicted ratio of airfoil surface local static to inlet total pressure with the experimental data for an exit static to inlet total pressure ratio of 0.314. These predictions required slightly larger damping constants in the central-difference codes due to the drastic flow gradients encountered near the trailing edge of the airfoil. Due to the somewhat erratic behavior of the transient solution, slightly higher damping

values were required in the Runge-Kutta calculation than the hopscotch scheme, even though in theory the same steady state solution should result from each code. Many techniques are available to smooth this false transient while marching to the steady state solution, but were not considered essential for this study.

Each of the codes indicates a small shock-like transition near 60 percent chord on the suction surface of the airfoil. There is insufficient experimental data to determine whether this is a physical or numerical phenomenon, although the flow is likely to be extremely sensitive in this region since it is near the throat. In addition, disturbances emanating from the trailing edge of an adjacent airfoil may well be interfering with the flow on the suction surface.

In order to examine the effect of grid density on the predicted results for this case, a second set of calculations were performed on a grid utilizing 171 normals and 15 contours, as shown in Fig. 4.10. The predicted airfoil surface static/total pressure ratios for the new grid are compared with the experimental data in Fig. 4.11. No significant differences between these results and those predicted using the 91x9 grid are evident, in spite of the fact that nearly three times as many grid points were used in the second grid. Further refinement of the grid is not expected to improve the calculations.

The predicted Mach number contours for this flow are given for each code in Figs. 4.12-4.15, respectively.

Each of the codes demonstrated good agreement with the experimental data, with no single code showing clear superiority. The convergence and CPU time data for each code are given in Table 4.4. Overall, relative CPU times were consistent with the

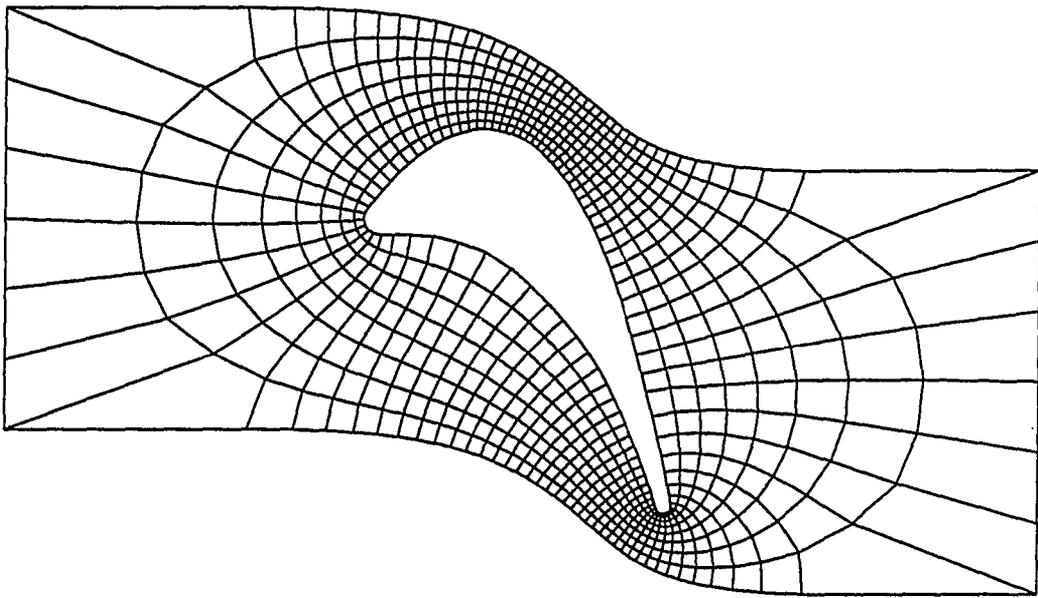


Figure 4.8: 91x9 grid system for low stagger turbine blade cascade

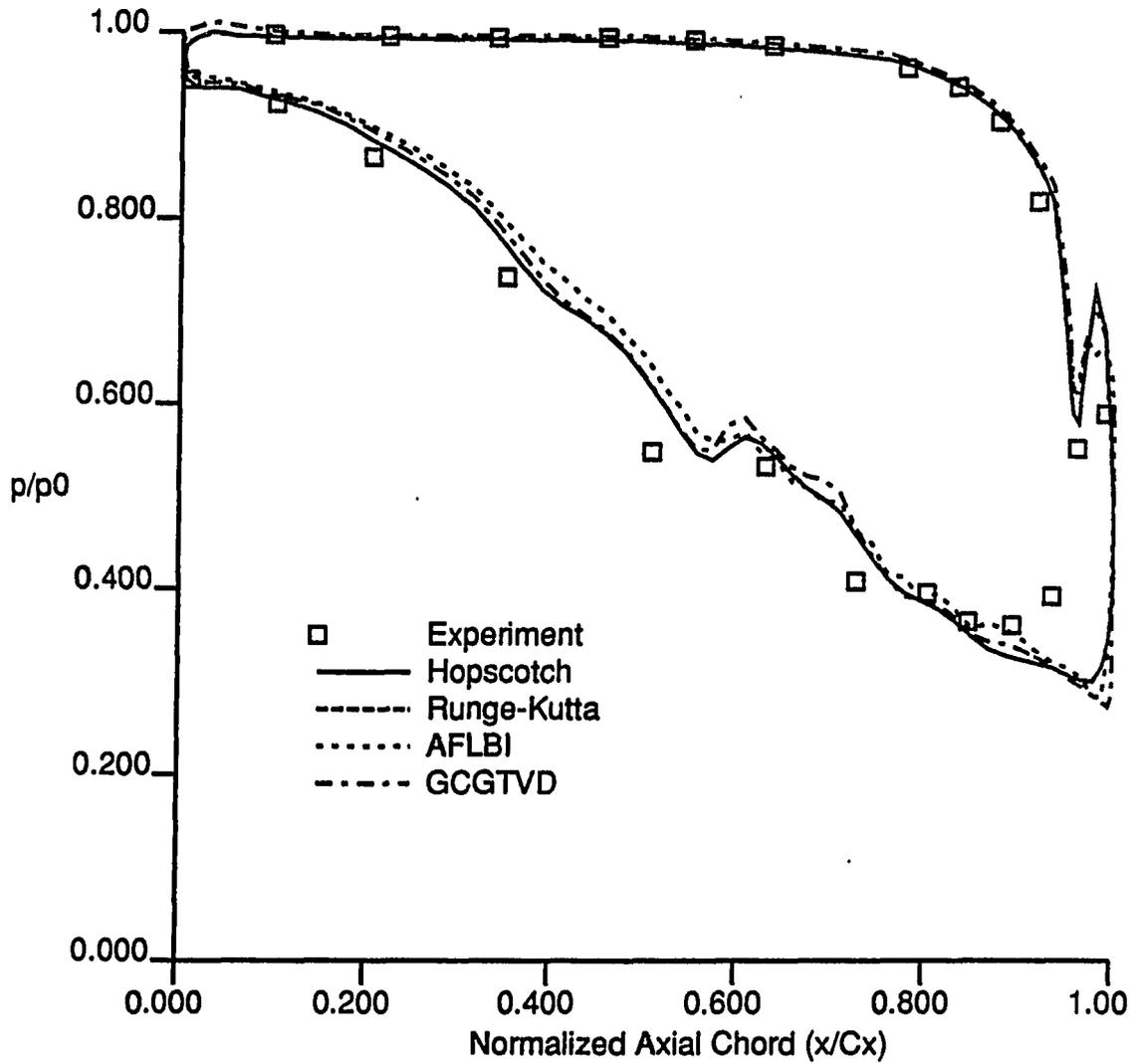
Low-Stagger Airfoil Turbine Cascade ($M_2=1.1$)

Figure 4.9: Comparison of the predicted and experimental airfoil surface static to total pressure ratio distributions for inviscid flow through a low stagger turbine cascade (91x9 grid)

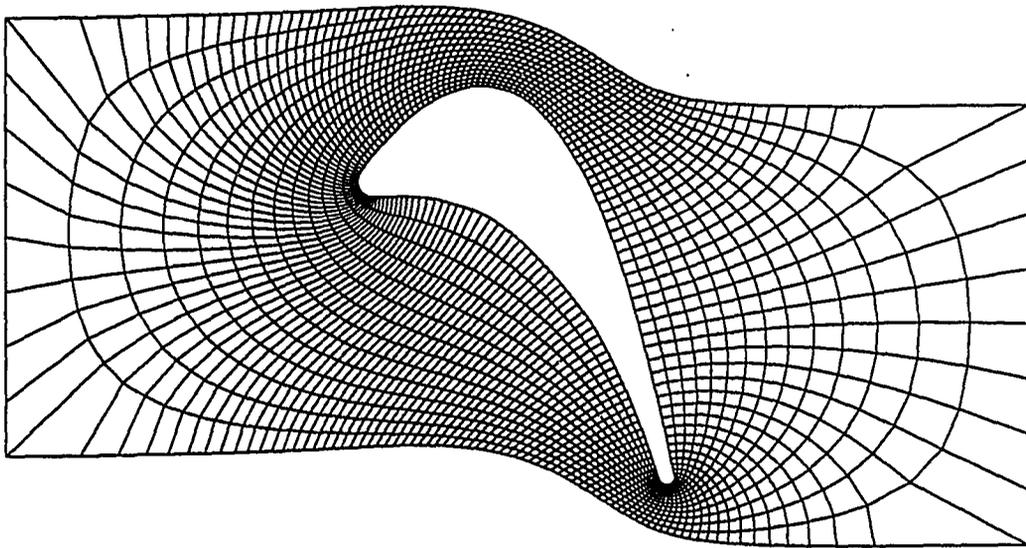


Figure 4.10: 171x15 grid system for low stagger turbine blade cascade

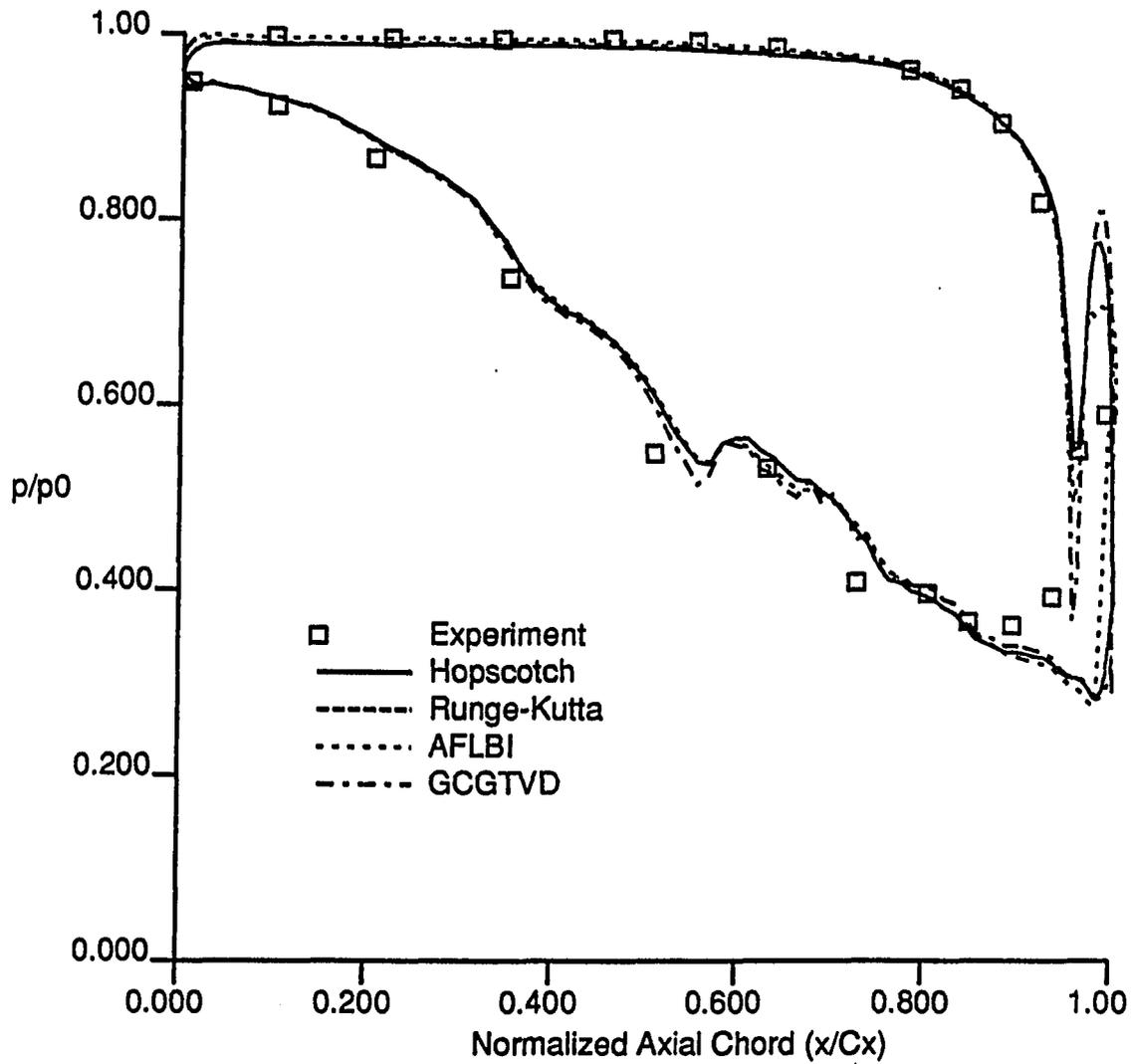
Low-Stagger Airfoil Turbine Cascade ($M_2=1.1$)

Figure 4.11: Comparison of the predicted and experimental airfoil surface static to total pressure ratio distributions for inviscid flow through a low stagger turbine cascade (171x15 grid)

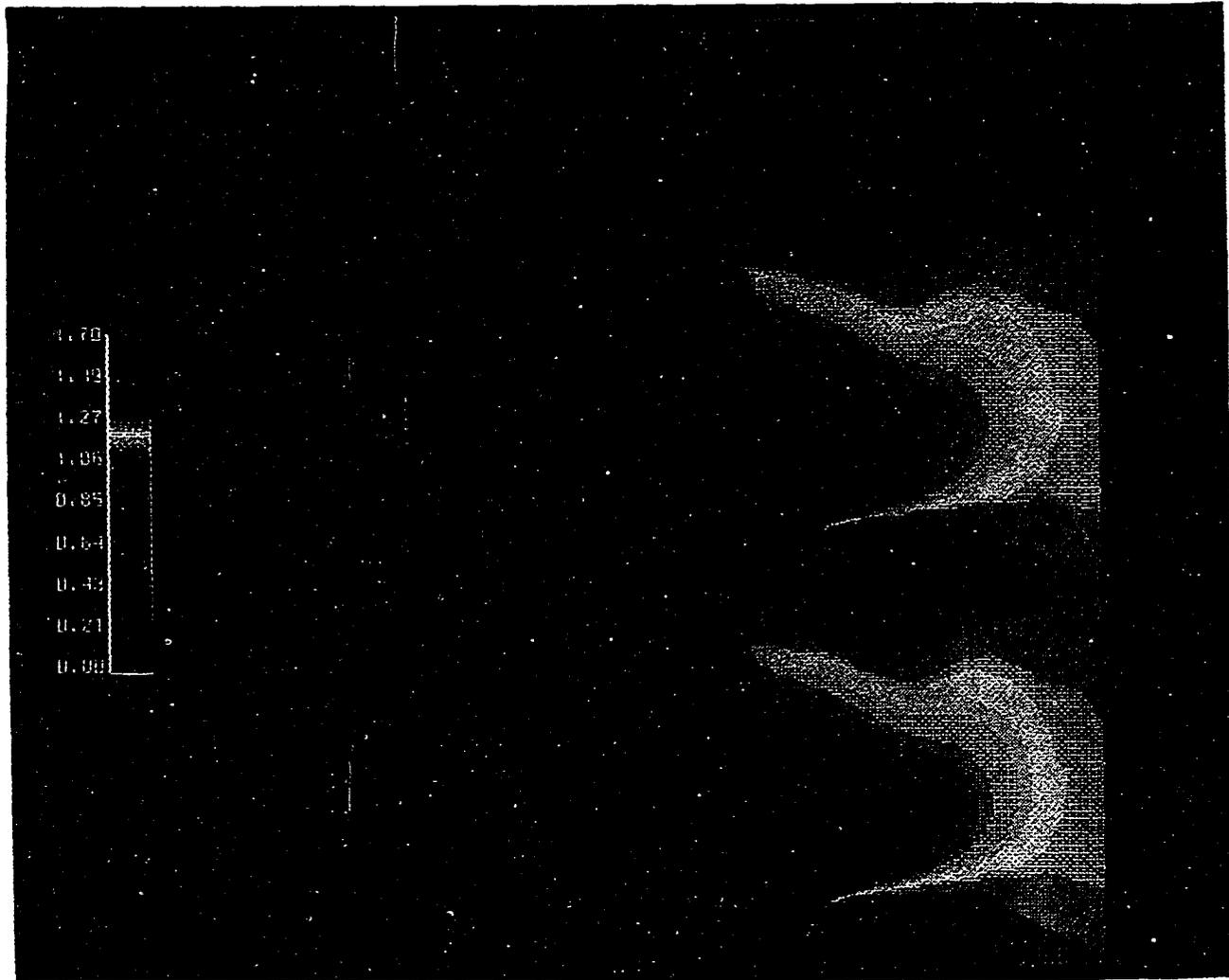


Figure 4.12: Predicted hopscotch Mach number contours for inviscid flow through a low stagger turbine cascade



Figure 4.13: Predicted Runge-Kutta Mach number contours for inviscid flow through a low stagger turbine cascade

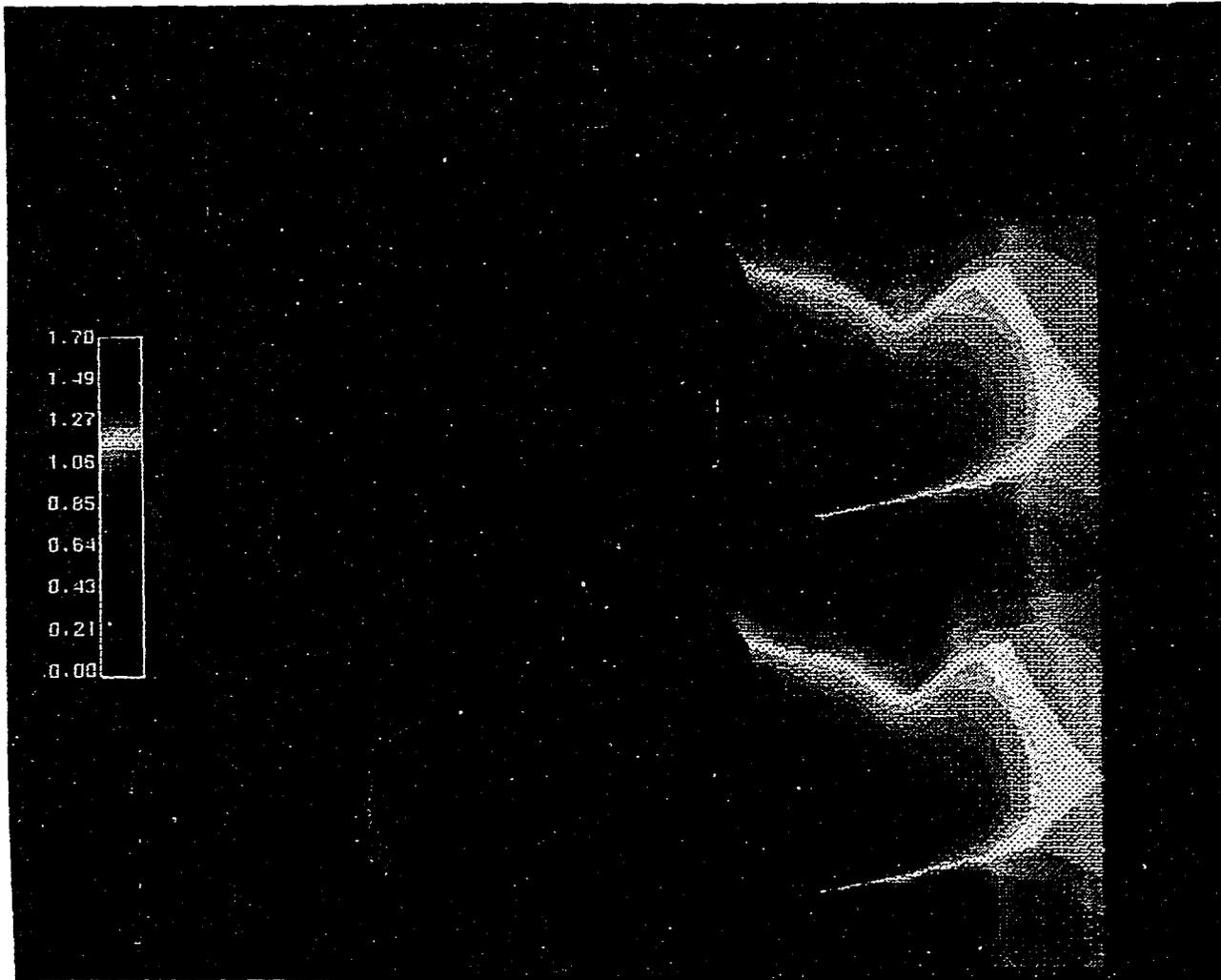


Figure 4.14: Predicted AFLBI Mach number contours for inviscid flow through a low stagger turbine cascade

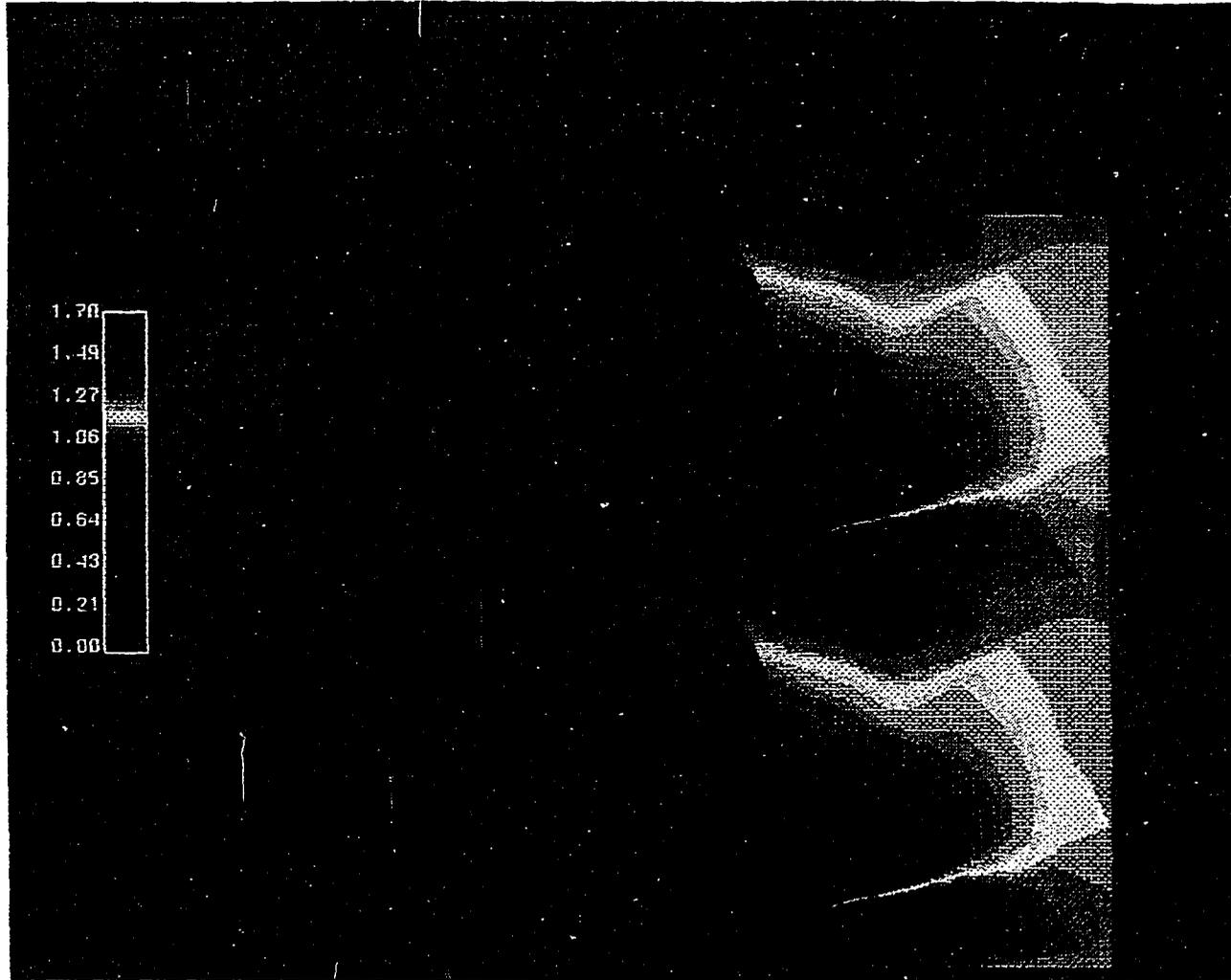


Figure 4.15: Predicted GCGTVD Mach number contours for inviscid flow through a low stagger turbine cascade

Table 4.4: Inviscid low stagger turbine cascade CPU time comparison

Method	CFL Number	Iterations for Convergence	CPU seconds for Convergence	CPU seconds Grid Point / It.
Hopscotch	1.0	3035	131.97	1.69×10^{-5}
Runge-Kutta	2.2	2190	161.21	2.87×10^{-5}
AFLBI	5.0	759	146.01	7.50×10^{-5}
GCGTVD	5.0	1472	357.00	9.46×10^{-5}

results obtained from the isolated cylinder test case. The large CPU time required for the GCGTVD code was traced to the effect of the limiter as the results were nearing convergence. The effect of the limiter turning on and off acted to destabilize the calculation slightly, which delayed convergence considerably. This behavior could best be improved through a smoother limiting function.

Steady Inviscid Flow Through a High Turning Turbine Cascade

A somewhat more demanding test case is the prediction of the flow through the high-turning turbine cascade pictured in Fig. 4.16. Experimental data for this cascade were reported in [105] for several exit static to inlet total pressure ratios. This cascade operates at a fixed inlet flow angle of 46 degrees, resulting in a negative incidence angle ($\iota = -6.9$ degrees). The ideal exit flow angle is -63 degrees resulting in a total of 109 degrees of ideal fluid turning by the cascade. Experimental flow visualization indicated the presence of a leading edge flow separation on the pressure surface of the airfoil (a result of the negative incidence), which should provide a rather challenging computational case.

Inviscid flow predictions were performed using the 157x15 grid system pictured in Fig. 4.16. A comparison of inviscid predicted local static/inlet total pressure ratio dis-

tributions with experimental data for an exit Mach number of 1.1 is given in Fig. 4.17. The inviscid calculations clearly fail to accurately predict surface pressure ratios in the leading edge region due to the experimentally observed separation, although good agreement is obtained for the remainder of the airfoil surface. The predicted Mach number contours for each code are given in Figs. 4.18-4.21, respectively.

Similar inviscid calculations using the 157x30 grid pictured in Fig. 4.22 led to no significant improvement or difference in the predicted results, and these results are thus considered nearly grid independent, although grid refinement in the circumferential coordinate direction has not been examined.

Execution times and convergence data were consistent with the previous inviscid test cases. Predicted Mach number contours for each of the predictions are given in Figs. 4.18-4.21, respectively. A pair of oblique trailing edge shocks are clearly displayed in each plot. It is interesting to note that the apparent shock angle near the trailing edge varies dramatically from code to code. This is most likely due to differences in airfoil surface boundary conditions and the numerically generated entropy associated with each scheme.

Steady Viscous Flow Through a High Turning Turbine Cascade

In order to examine the viscous flow prediction qualities of each of the time-marching algorithms, the laminar, viscous flow through the high-turning turbine cascade described in the previous section was calculated. In order to resolve the small-length scale viscous activity of the airfoil surface boundary layer, the 157x30

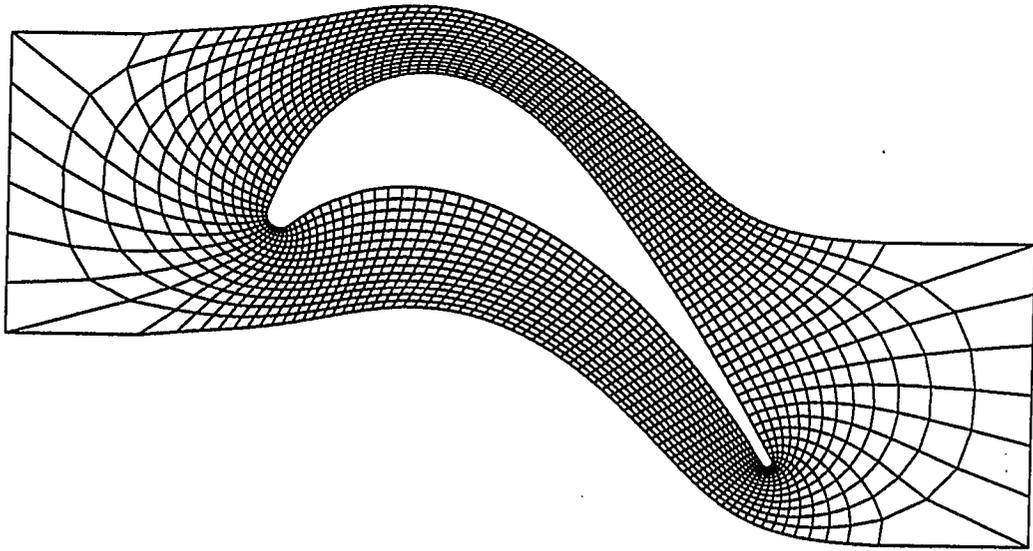


Figure 4.16: High-turning turbine vane 157x15 grid

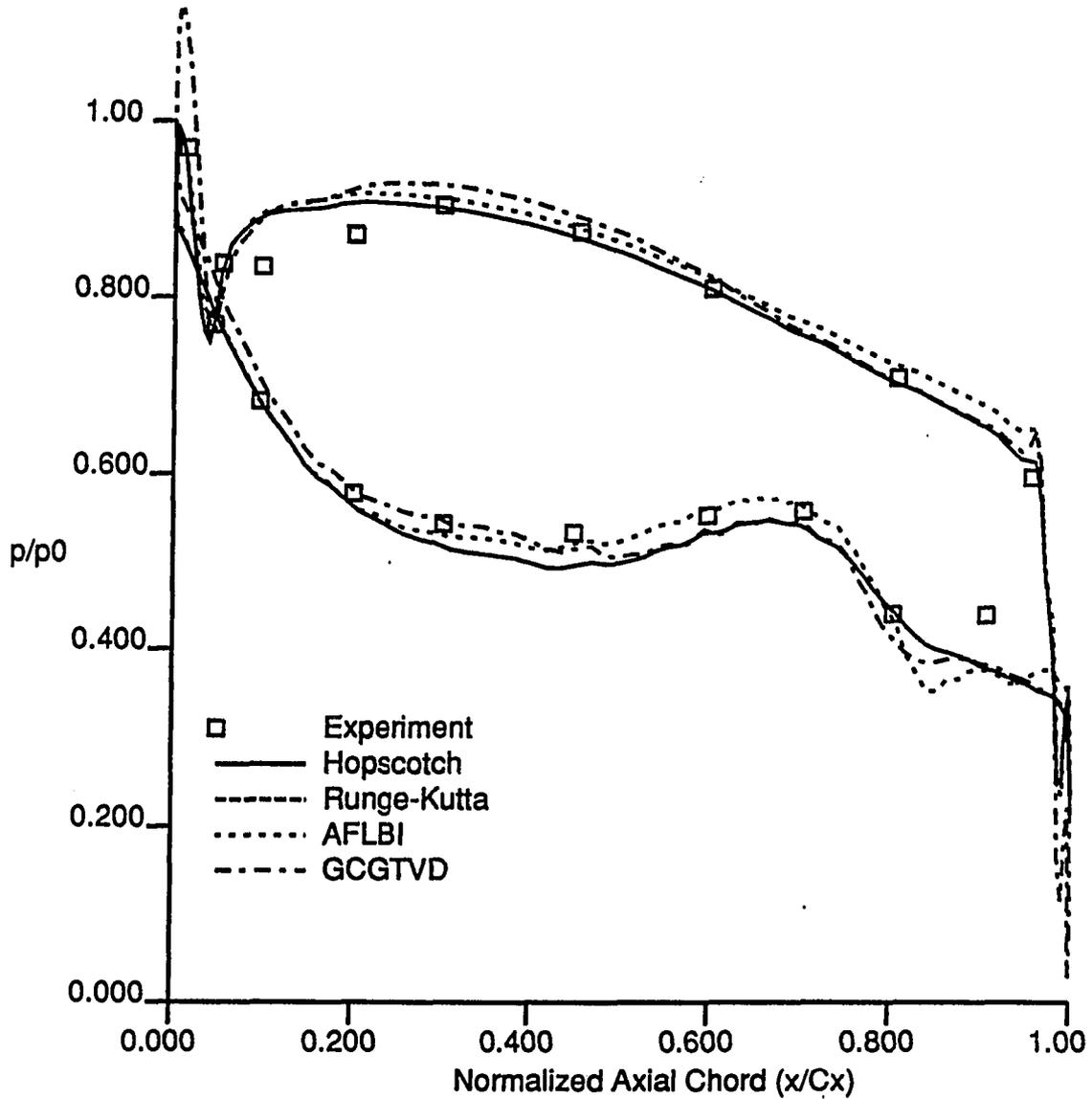
High-Turning Turbine Airfoil Cascade ($M_2=1.1$)

Figure 4.17: Comparison of the predicted and experimental airfoil surface static to total pressure ratio distributions for inviscid flow through a high-turning turbine cascade

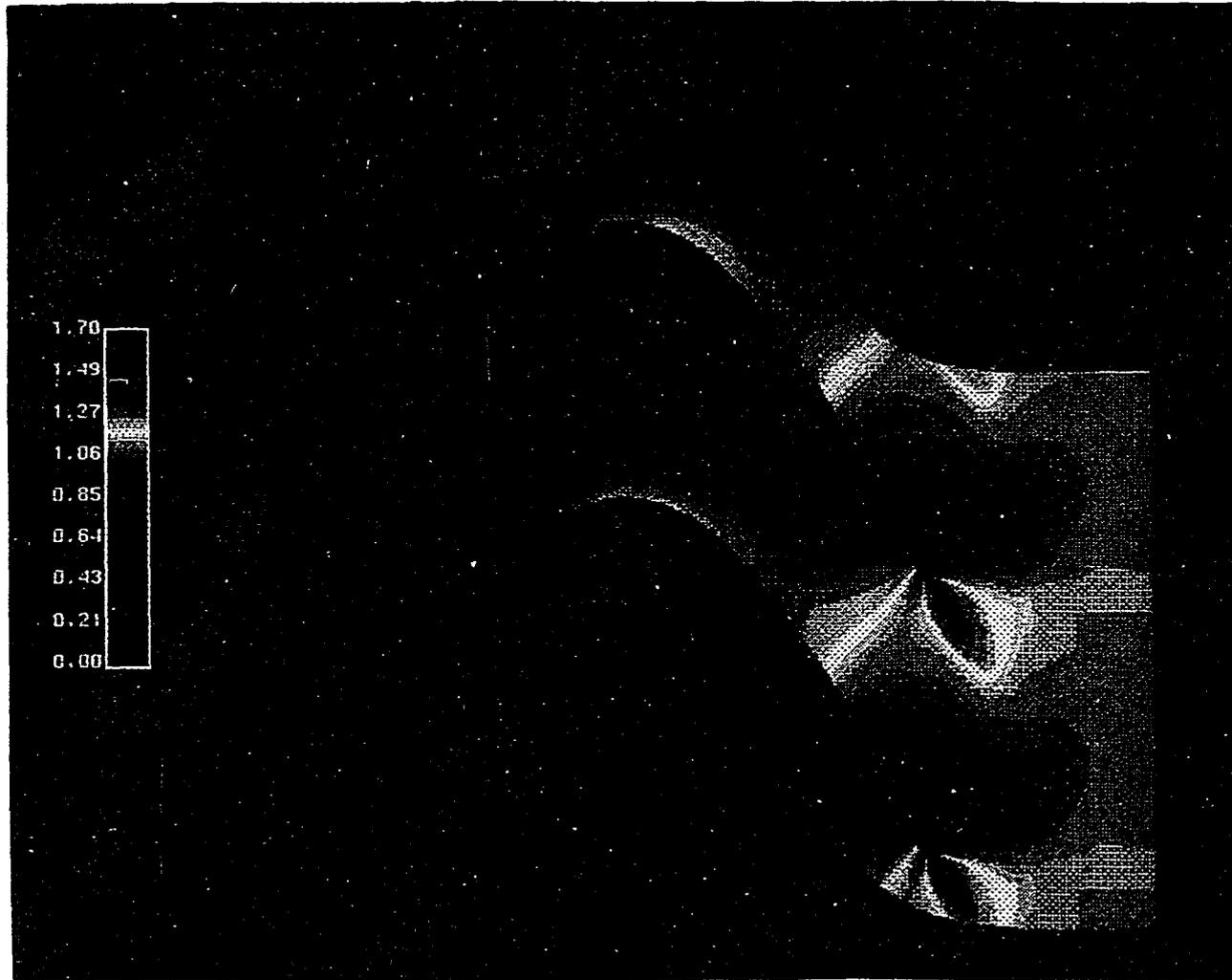


Figure 4.18: Predicted hopscotch Mach number contours for inviscid flow through a high-turning turbine cascade

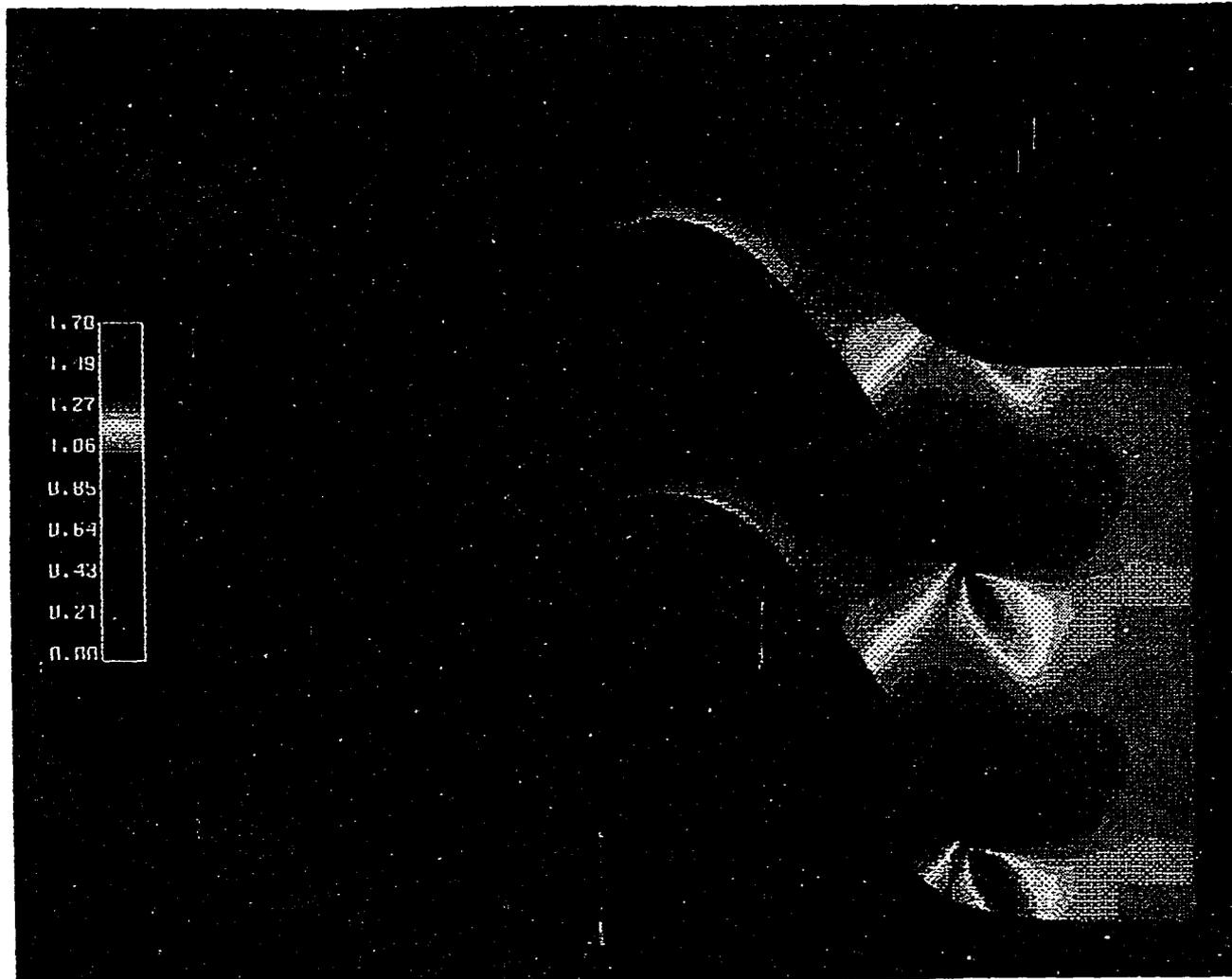


Figure 4.19: Predicted Runge-Kutta Mach number contours for inviscid flow through a high-turning turbine cascade

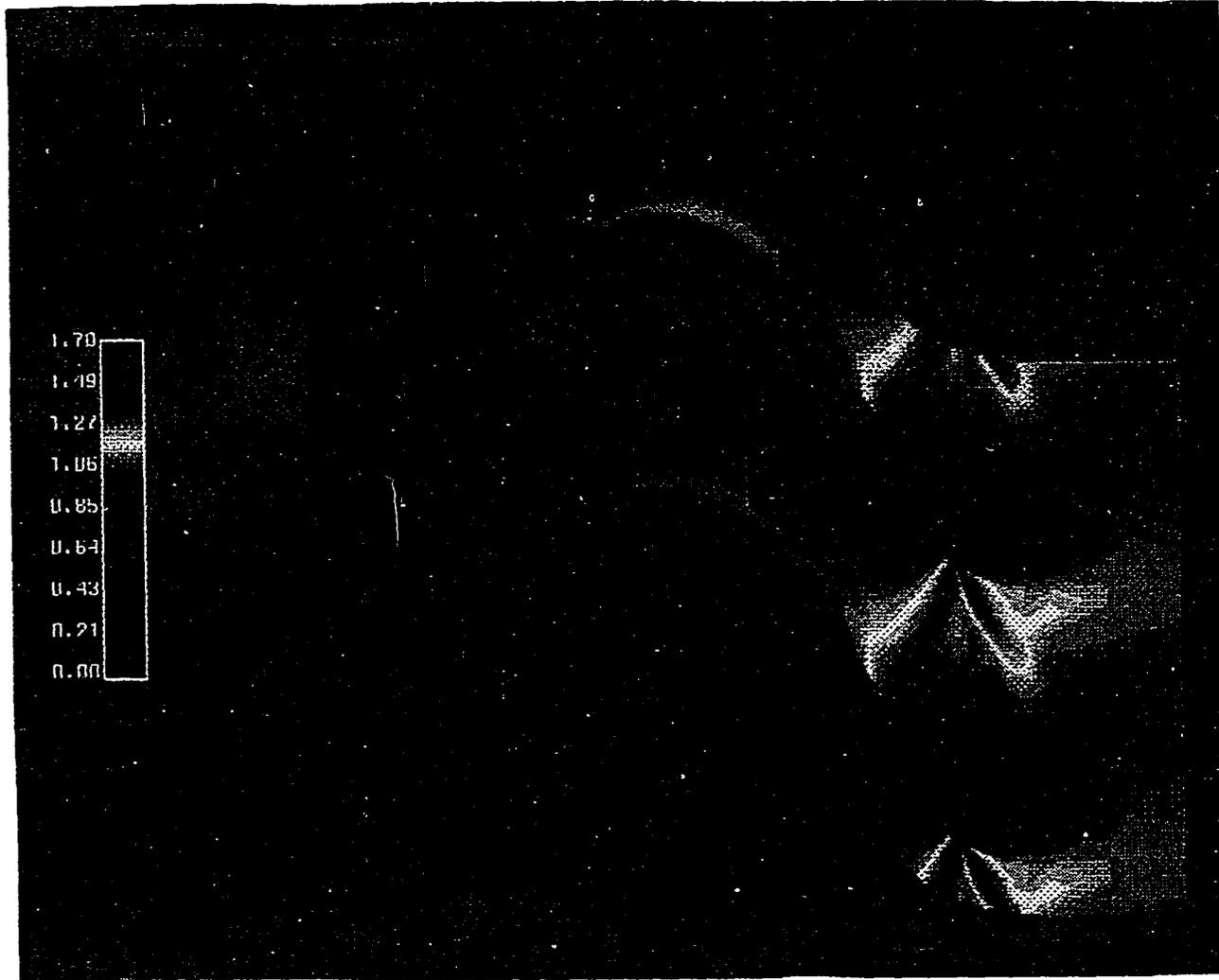


Figure 4.20: Predicted AFLBI Mach number contours for inviscid flow through a high-turning turbine cascade



Figure 4.21: Predicted GCGTVD Mach number contours for inviscid flow through a high-turning turbine cascade

grid pictured in Fig. 4.22 was generated. This grid was constructed by reinterpolating points along the grid normals of the original 157x15 grid used for the inviscid calculations presented in the previous section. Calculations were performed using the same damping constants as the inviscid predictions, although the calculation CFL number was reduced 20% to avoid compromising any viscous stability criteria. A larger number of iterations was required by each code to obtain a converged solution because of the small grid spacing and effect of the viscous terms.

A comparison of viscous predicted airfoil surface static/total pressure ratios with experimental data for the 1.1 exit Mach number case with a chord-based Reynolds number of 230,000 is given in Fig. 4.23. For the viscous calculations, the predicted local static/inlet total pressure ratios now clearly more accurately match the experimental data in the leading edge region. The grid is not sufficiently refined near the trailing edge to adequately capture a trailing edge separation or wake, and thus a small difference in the predictions and experimental data is still observed there.

The predicted Mach number contour plots for each code are displayed in Figs. 4.24-4.27, respectively. Some evidence of the airfoil surface boundary layer and wake is displayed in each plot.

An examination of the velocity vector patterns for the viscous predictions of each code indicates that each code has adequately captured the leading edge separation bubble. Predicted velocity vector patterns near the leading edge region are presented for each code in Figs. 4.28-4.31, respectively. The separation region is clearly defined in each case. It is interesting to note that the largest separation bubble is given by the AFLBI calculation in Fig. 4.30. This may well be an indication of the level of damping

Table 4.5: Viscous high-turning turbine cascade CPU time comparison

Method	CFL Number	Iterations for Convergence	CPU seconds for Convergence	CPU seconds Grid Point / It.
Hopscotch	0.7	4043	224.70	1.18x10(-5)
Runge-Kutta	1.5	3100	365.03	2.50x10(-5)
AFLBI	3.0	1745	616.42	7.50x10(-5)
GCGTVD	5.0	1497	706.50	10.02x10(-5)

being applied by each code. The magnitude of the velocities in the recirculation region differ significantly for each calculation. The experimental data is unfortunately not a good guide to select a best solution, since the predictions were completely laminar, while the experimental flow was likely transitional, or turbulent.

The CPU requirements and convergence data for each of these runs are given in Table 4.5. The viscous flow solver is not considerably more expensive per iteration, but delays in convergence were observed for each code compared to the inviscid schemes. Again, back to back CPU time comparisons often gave erratic results, so these data must be interpreted with this in mind. In addition, since the convergence criteria do not take into consideration the magnitude of the time step, the scheme with the smallest time step will likely "converge" (in terms of the relative change in pressure during a time step) first.

In order to evaluate the efficiency of the generalized conjugate gradient solution technique in comparison with other matrix relaxation procedures, this same flow was calculated using a point relaxation, and two types of line relaxation iterative methods. For each relaxation loop, the matrix solution was assumed to be converged when the maximum error for each variable was reduced by four orders of magnitude. Although the generalized conjugate gradient procedure required much more work per iteration,

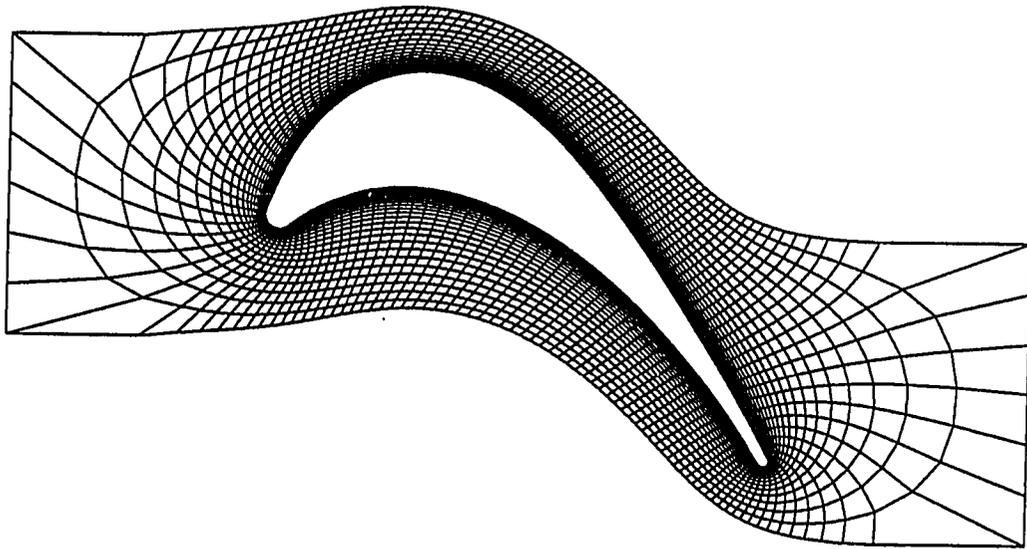


Figure 4.22: 157x30 grid system for high-turning turbine cascade

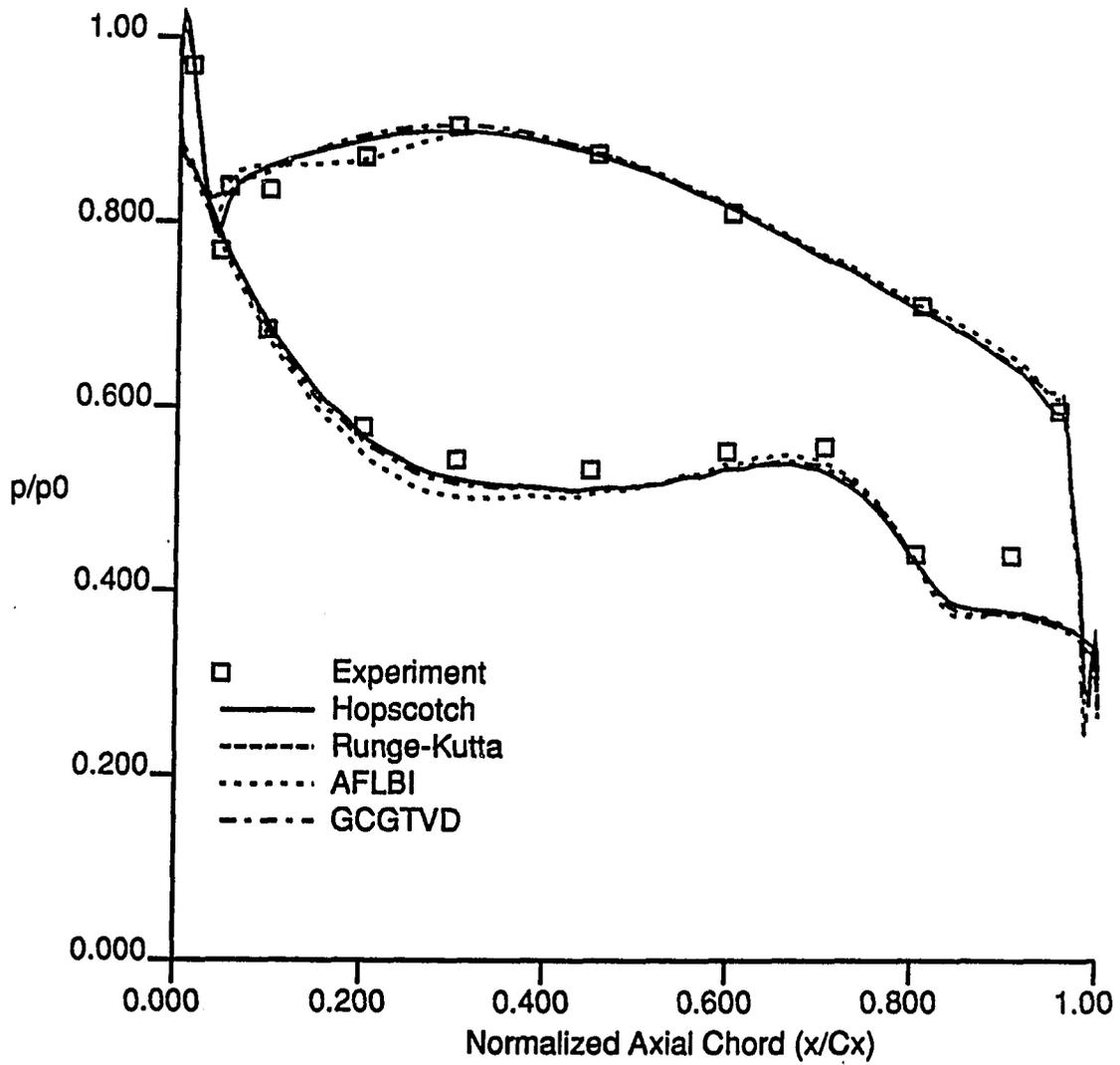
High-Turning Turbine Airfoil Cascade ($M_2=1.1$)

Figure 4.23: Comparison of the predicted and experimental airfoil surface static to total pressure ratio distributions for viscous flow through a high-turning turbine cascade

High-Turning Turbine Cascade ($M_2=1.1$)
Hopscotch Viscous Predicted Mach Number Contours



Figure 4.24: Predicted hopscotch Mach number contours for viscous flow through a high-turning turbine cascade

High-Turning Turbine Cascade ($M_2=1.1$)
Runge-Kutta Viscous Predicted Mach Number Contours

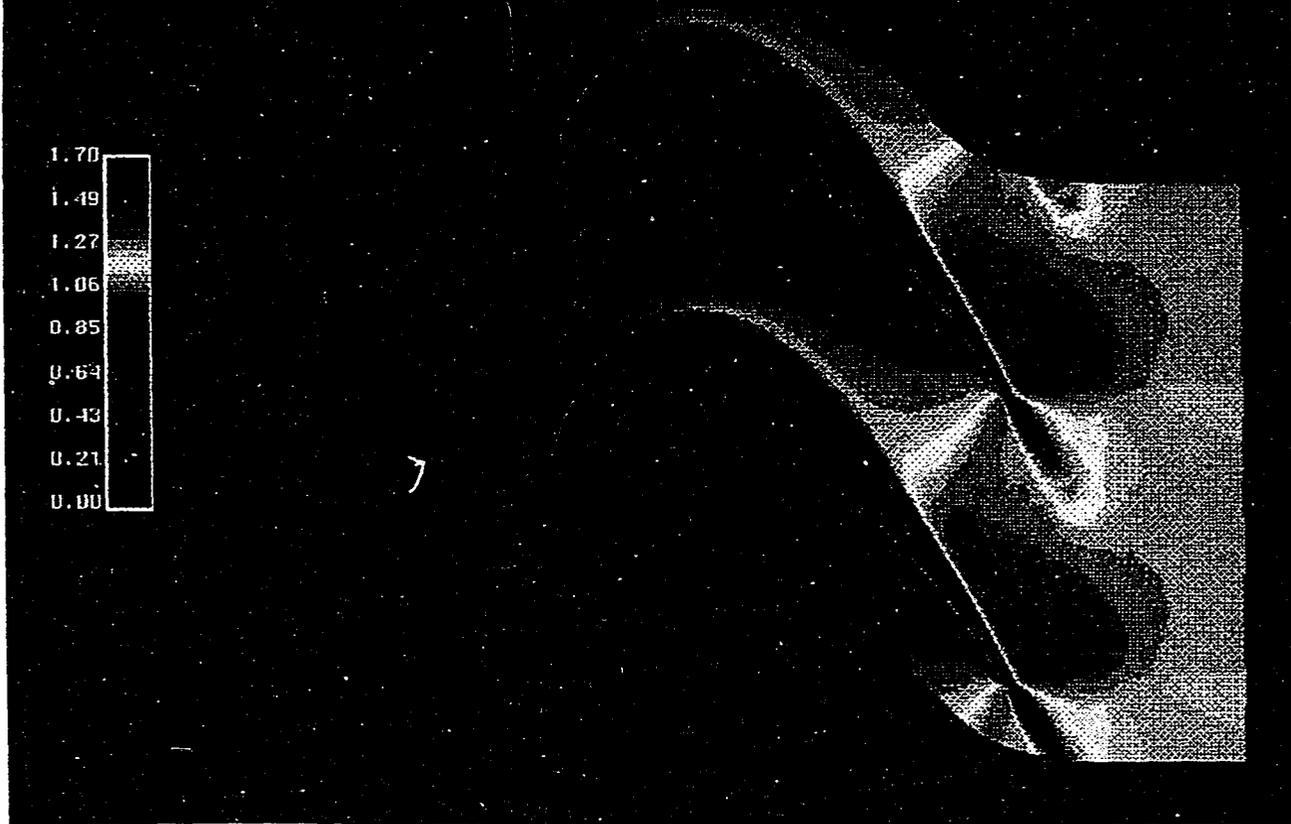


Figure 4.25: Predicted Runge-Kutta Mach number contours for viscous flow through a high-turning turbine cascade

High-Turning Turbine Cascade ($M_2=1.1$)
AFLBI Viscous Predicted Mach Number Contours

1.70
1.40
1.20
1.00
0.80
0.60
0.40
0.20
0.00



Figure 4.26: Predicted AFLBI Mach number contours for viscous flow through a high-turning turbine cascade

High-Turning Turbine Cascade ($M_2=1.1$)
GCGTVD Viscous Predicted Mach Number Contours

1.70
1.49
1.27
1.06
0.85
0.64
0.43
0.21
0.00

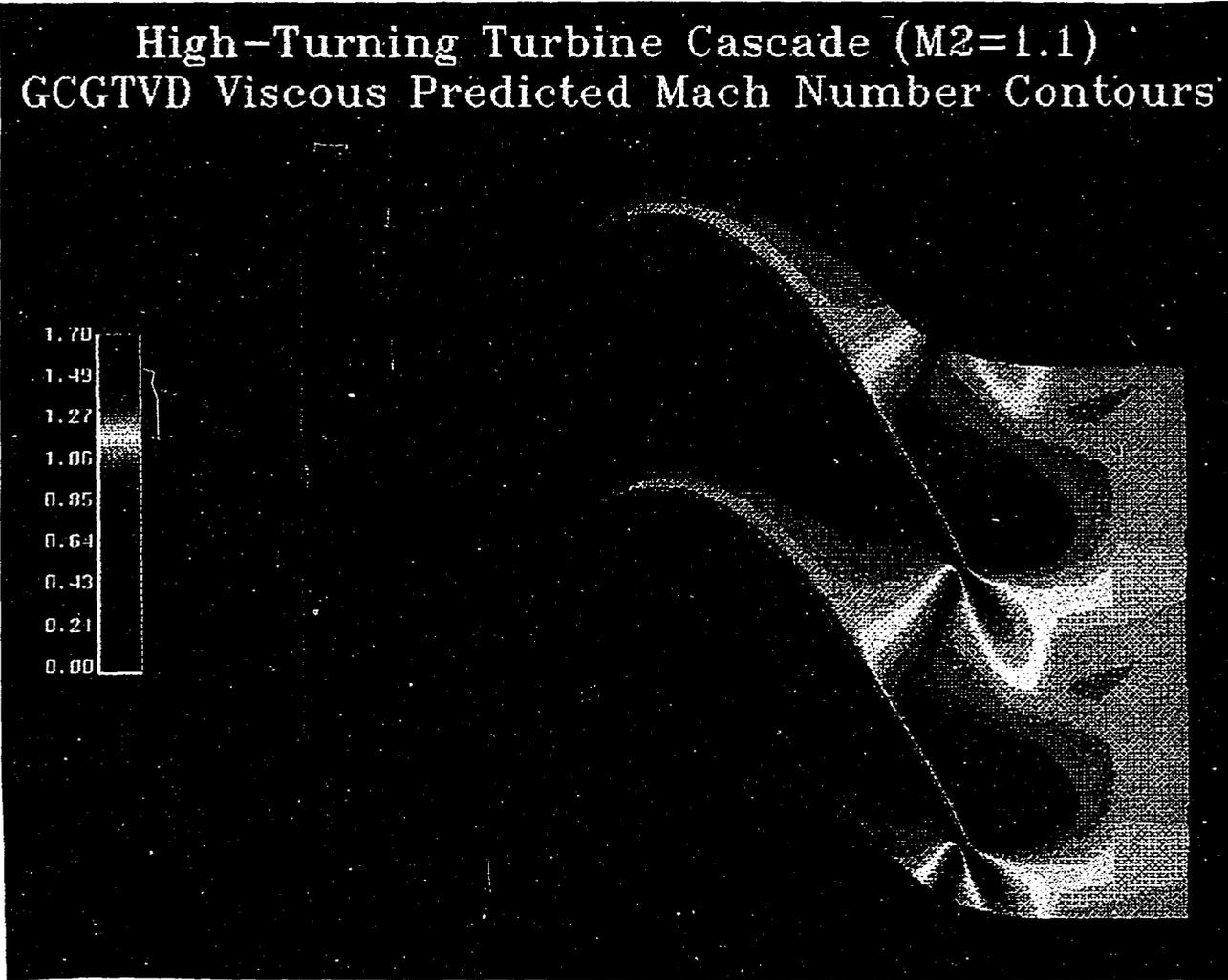


Figure 4.27: Predicted GCGTVD Mach number contours for viscous flow through a high-turning turbine cascade

Leading Edge Detail of Predicted Hopscotch Velocity Vectors

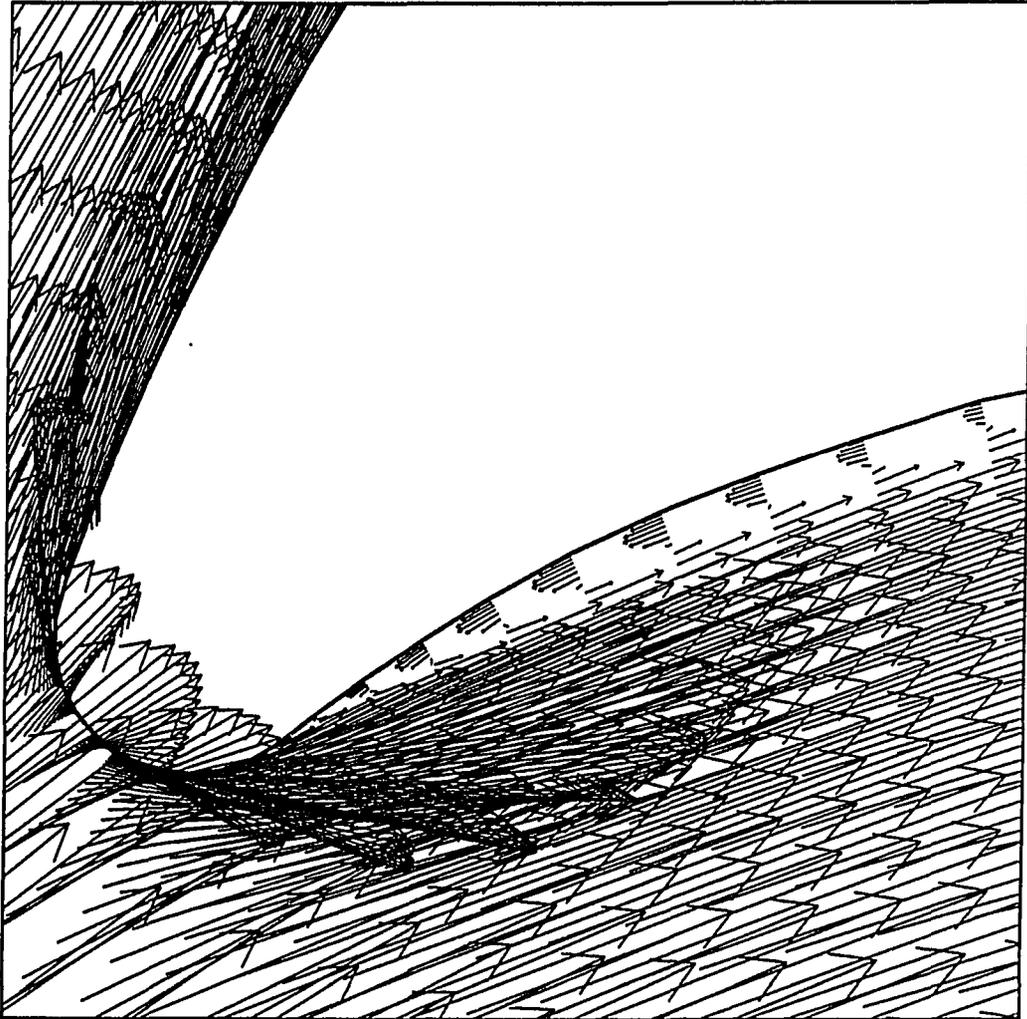


Figure 4.28: Leading edge detail of the predicted hopscotch velocity vectors for viscous flow over a high-turning turbine cascade

Leading Edge Detail of Predicted Runge-Kutta Velocity Vectors

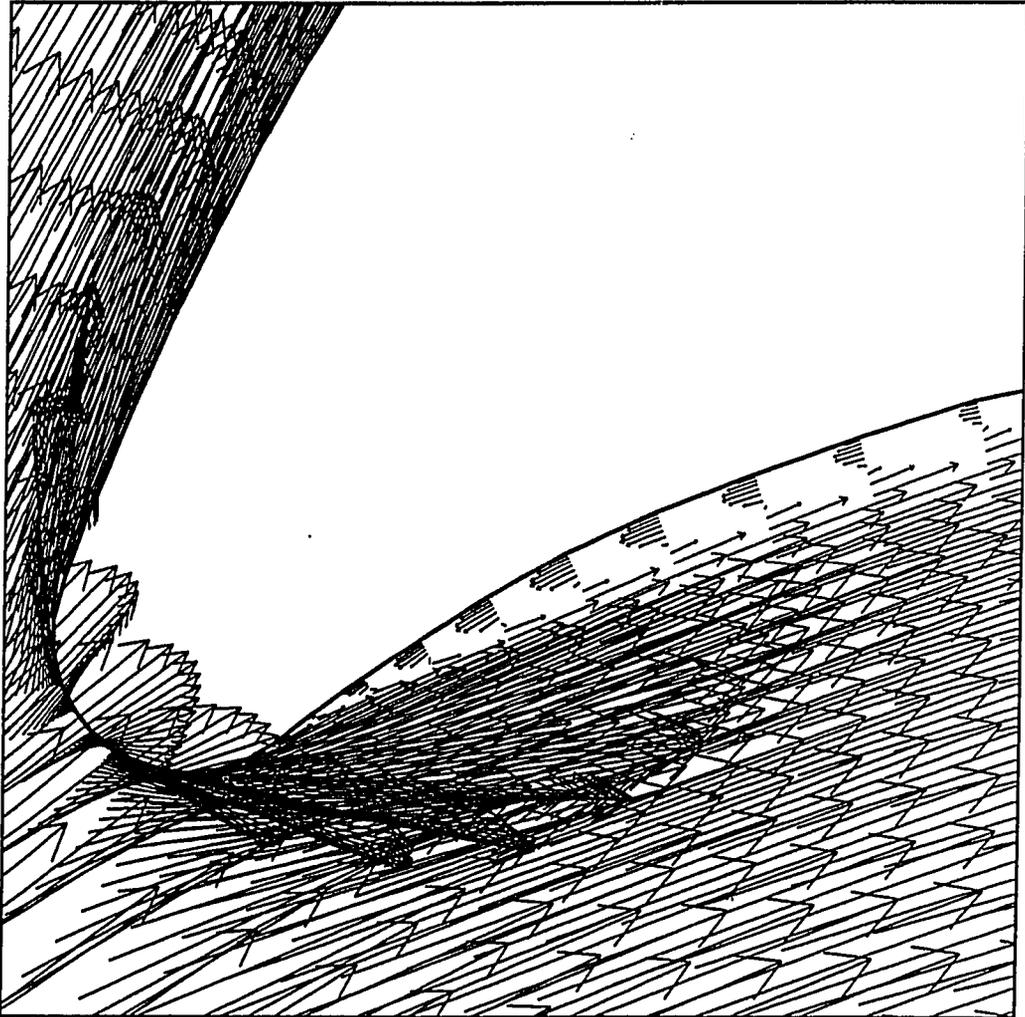


Figure 4.29: Leading edge detail of the predicted Runge-Kutta velocity vectors for viscous flow over a high-turning turbine cascade

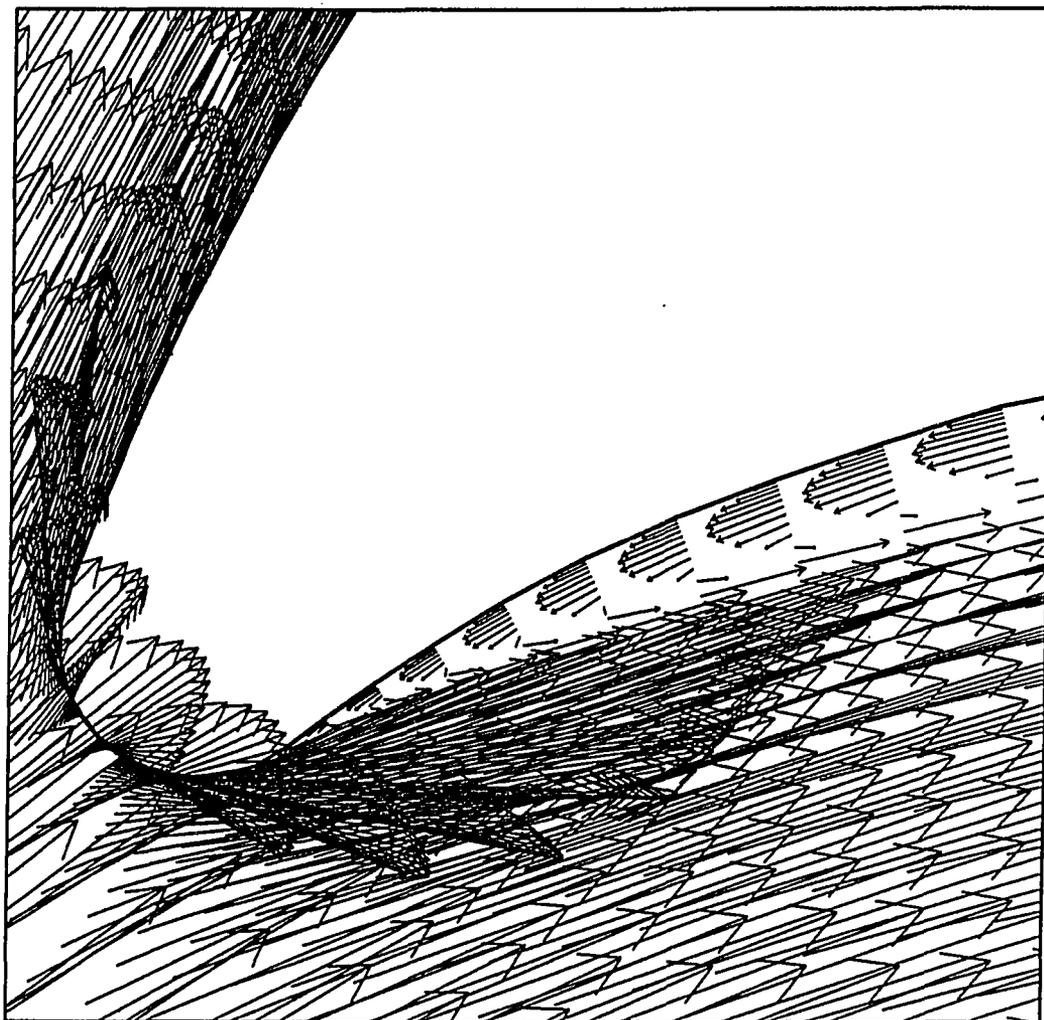
Leading Edge Detail of Predicted AFLBI Velocity Vectors

Figure 4.30: Leading edge detail of the predicted AFLBI velocity vectors for viscous flow over a high-turning turbine cascade

Leading Edge Detail of Predicted GCGTVD Velocity Vectors

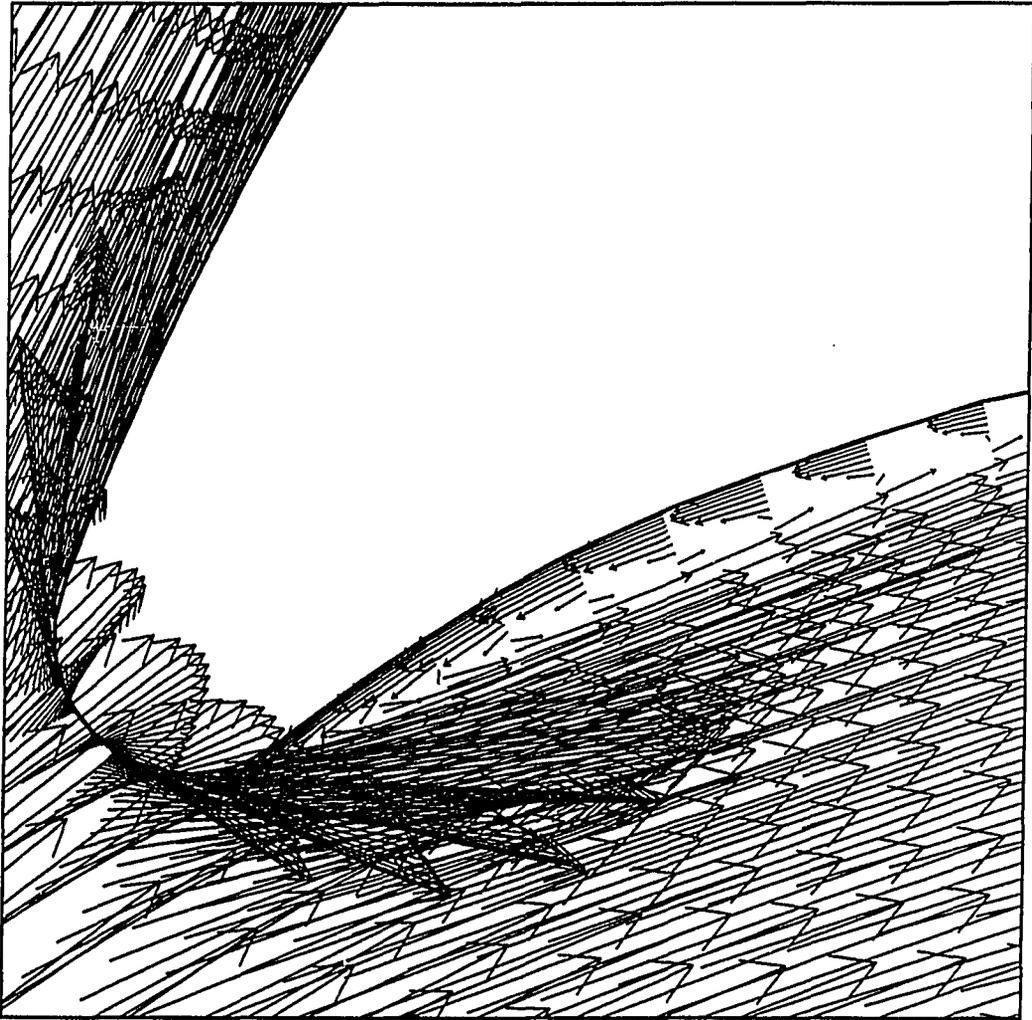


Figure 4.31: Leading edge detail of the predicted GCGTVD velocity vectors for viscous flow over a high-turning turbine cascade

a payoff was achieved through faster convergence and an overall slightly lower execution time ($\sim 5\%$). Further benefits are anticipated with the use of preconditioned generalized conjugate gradient routines.

Steady Viscous Subsonic Flow over a Flat Plate

A common test case for steady viscous flow calculations is the prediction of the developing laminar boundary-layer flow over a semi-infinite flat plate. The geometry and flow conditions for such a flow are illustrated in Fig. 4.32. Computed results are compared with the classical Blasius [106] flat plate similarity solution for laminar incompressible boundary-layers. Although the Blasius solution is based on the incompressible boundary-layer equations, for relatively low Mach numbers and wall heat flux values a reasonable agreement between computed and Blasius results can be expected.

For the purposes of this comparison, the plate length L was limited to 6.0 meters. The remaining freestream conditions were chosen such that the overall plate length Reynolds number ($Re_L = \rho_\infty u_\infty L / \mu_\infty$) based on freestream (∞) conditions was 10,000. A constant wall temperature of 352.286° K was used resulting in a temperature difference of 16.514° K across the boundary-layer. The 121×51 numerical grid used for the numerical predictions is pictured in Fig. 4.33. The grid is equally spaced in the x direction with a spacing $\Delta x = 0.05\text{m}$. The grid is highly stretched in the y direction to provide finer resolution of the boundary-layer region. The grid spacing in the y direction at the plate is $\Delta y = 0.0007\text{m}$. The extent of the grid in the y direction was 4 meters from the plate. The leading edge of the flat plate coincides

Flat Plate Flow Definition

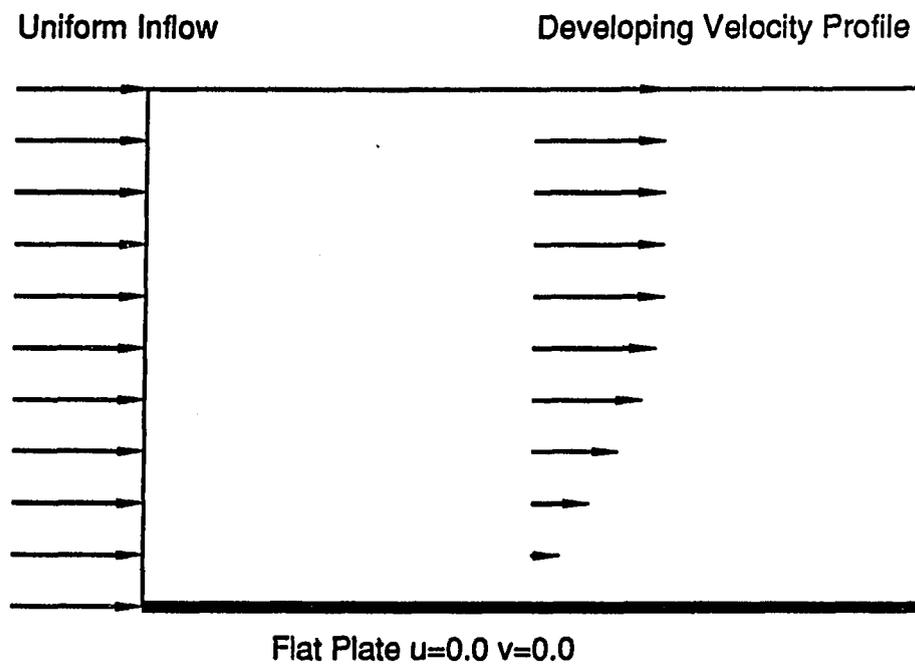


Figure 4.32: Geometry definition for flow over a flat plate

with the inlet of the grid.

Initial calculations showed that the predicted results were highly sensitive to the boundary conditions used at the exit and far field boundaries. In order to adequately simulate this case and achieve good agreement with the Blasius solution, some modification of the boundary algorithms in each code was required. The plate boundary was treated with the usual no slip conditions ($u, v = 0$) with a zero normal pressure gradient ($\partial p / \partial y = 0$) and a uniform fixed wall temperature. The inlet boundary was treated using the method of characteristics approach outlined in the Appendix. The freestream boundary was treated with the boundary layer flow conditions: $\partial u / \partial y = \partial v / \partial y = \partial T / \partial y = \partial p / \partial y = 0$. The exit boundary was also treated in a boundary-layer flow fashion by utilizing a modified form of the method of characteristics boundary algorithm described in the Appendix, with the additional condition $\partial^2 T / \partial x^2 = 0$. Other formulations tested often interfered with the development of the flow, or caused poor convergence due to reflections of transient waves within the computational domain.

Several predicted boundary-layer flow quantities for the four schemes are compared with the Blasius solution in Figs. 4.34-4.37. A comparison of the predicted wall skin friction coefficient distributions with the Blasius solution using

$$C_f = \mu \frac{\left(\frac{\partial u}{\partial y}\right)_{wall}}{\frac{1}{2} \rho_{\infty} u_{\infty}^2} \quad (4.2)$$

is given in Figs. 4.34-4.37 for the hopscotch, Runge-Kutta, AFLBI and GCGTVD algorithms, respectively. Although the predicted results deviate somewhat from the Blasius solution near the inlet for each solution, this is only a local effect, and good agreement between predicted and Blasius results is quickly established over the re-

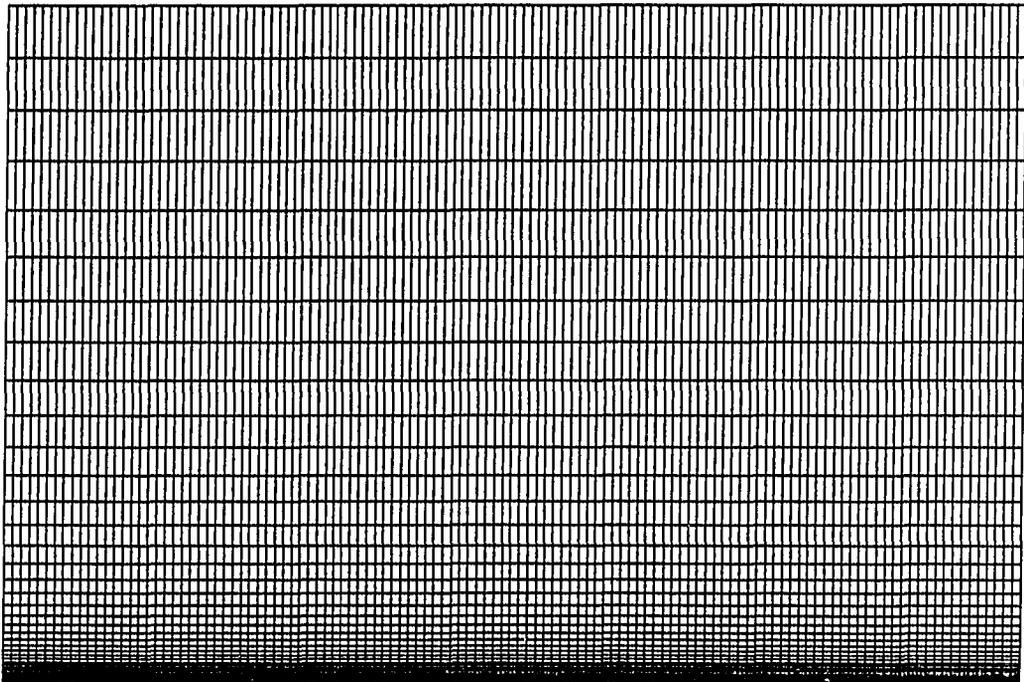


Figure 4.33: 121x51 grid for flow over a flat plate

maining length of the plate. The deviation near the downstream boundary in these plots results from incomplete convergence of the steady solution. The data presented here are from solutions converged to $\epsilon = 10^{-6}$ (see Eq. (4.1)). Convergence to a lower value (i.e., $\epsilon = 10^{-8}$) yields good agreement over the entire surface of the plate. Unfortunately, such a level of convergence is unattainable or extremely expensive for some codes. Obtaining this level of convergence with an explicit scheme may well take over 20,000 iterations. This behavior is cause for concern over the accuracy of Navier-Stokes solvers in general. In this case, the accuracy of the solver is not questioned, but the normally acceptable convergence levels often advertised as providing accurate solutions may well be overly optimistic. Because of the convergence issue, no CPU time comparison is presented for this case.

A similar trend is observed for the local Nusselt number predictions for the same order of algorithms illustrated in Figs. 4.34-4.37, where:

$$Nu = \frac{hx}{k} \quad (4.3)$$

Again, downstream of the leading edge region, where the Blasius solution is likely to be invalid, excellent agreement between predicted and Blasius results is displayed.

A comparison of predicted and Blasius velocity profiles at $Re_x=5,000$ is given in Figs. 4.34-4.37 for each of the four codes. The predicted results show excellent agreement with the Blasius results, except for the highly curved region near the boundary-layer edge. It is as yet unknown whether this discrepancy is a result of the limiting boundary-layer assumption of the Blasius solution, or a result of the dissipation or compressibility of the Navier-Stokes predictions. Predictions from other

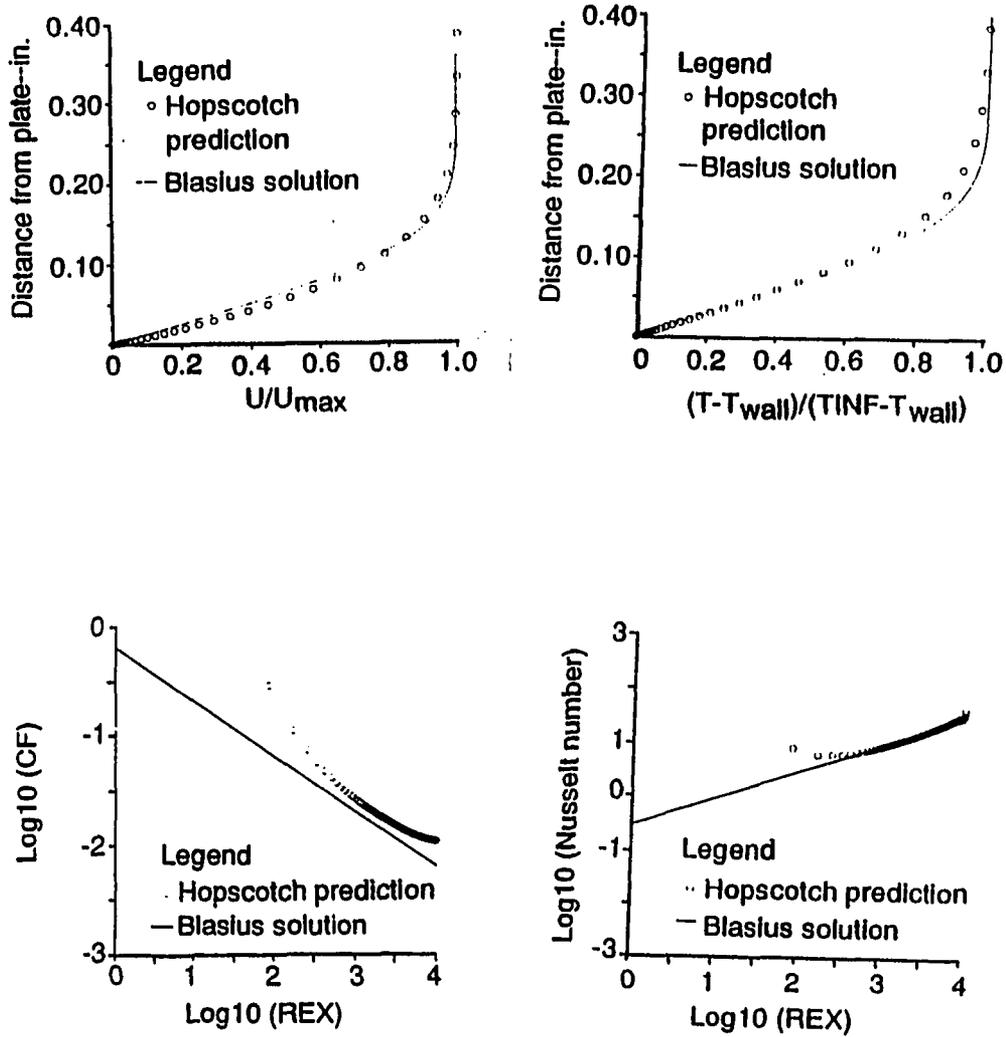


Figure 4.34: Comparison of predicted hopscotch and Blasius solution boundary-layer flow quantities for flow over a flat plate

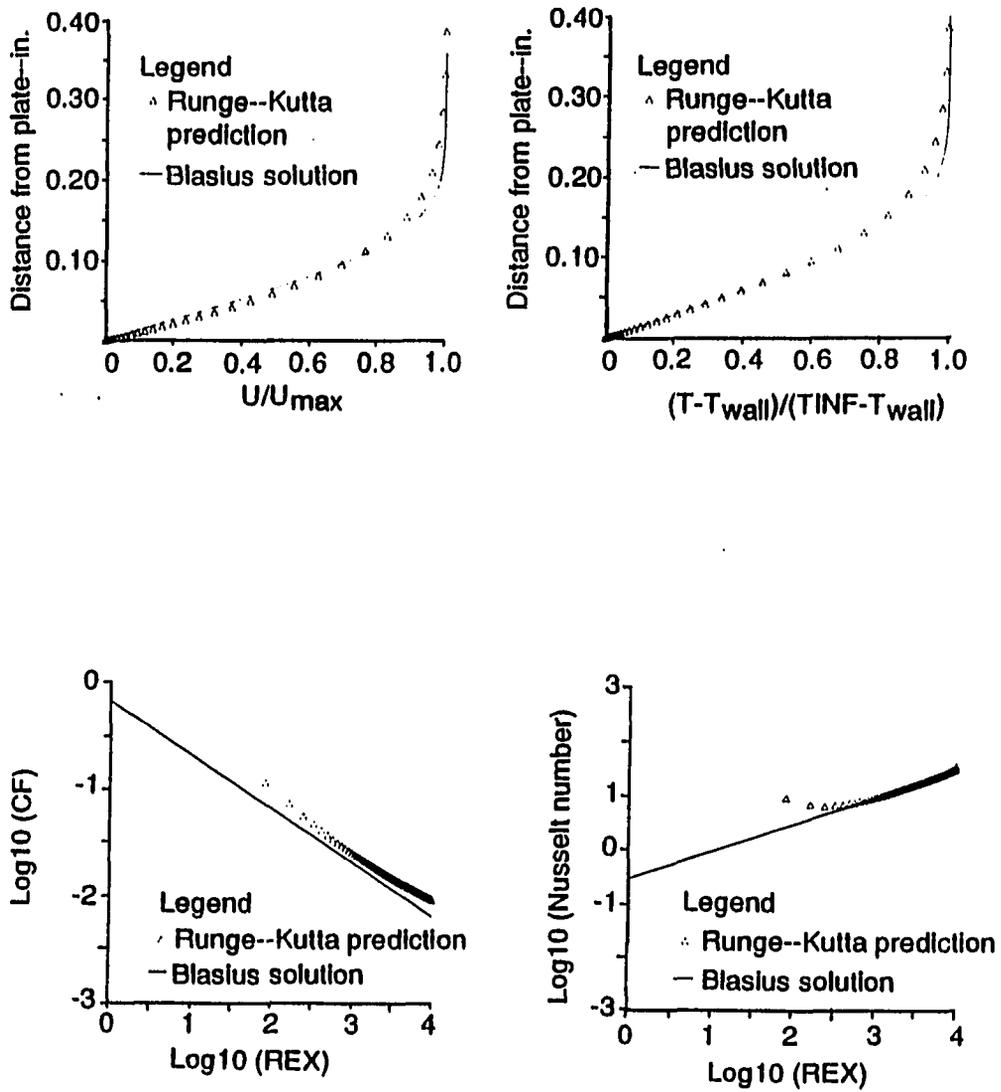


Figure 4.35: Comparison of predicted Runge-Kutta and Blasius solution boundary-layer flow quantities for flow over a flat plate

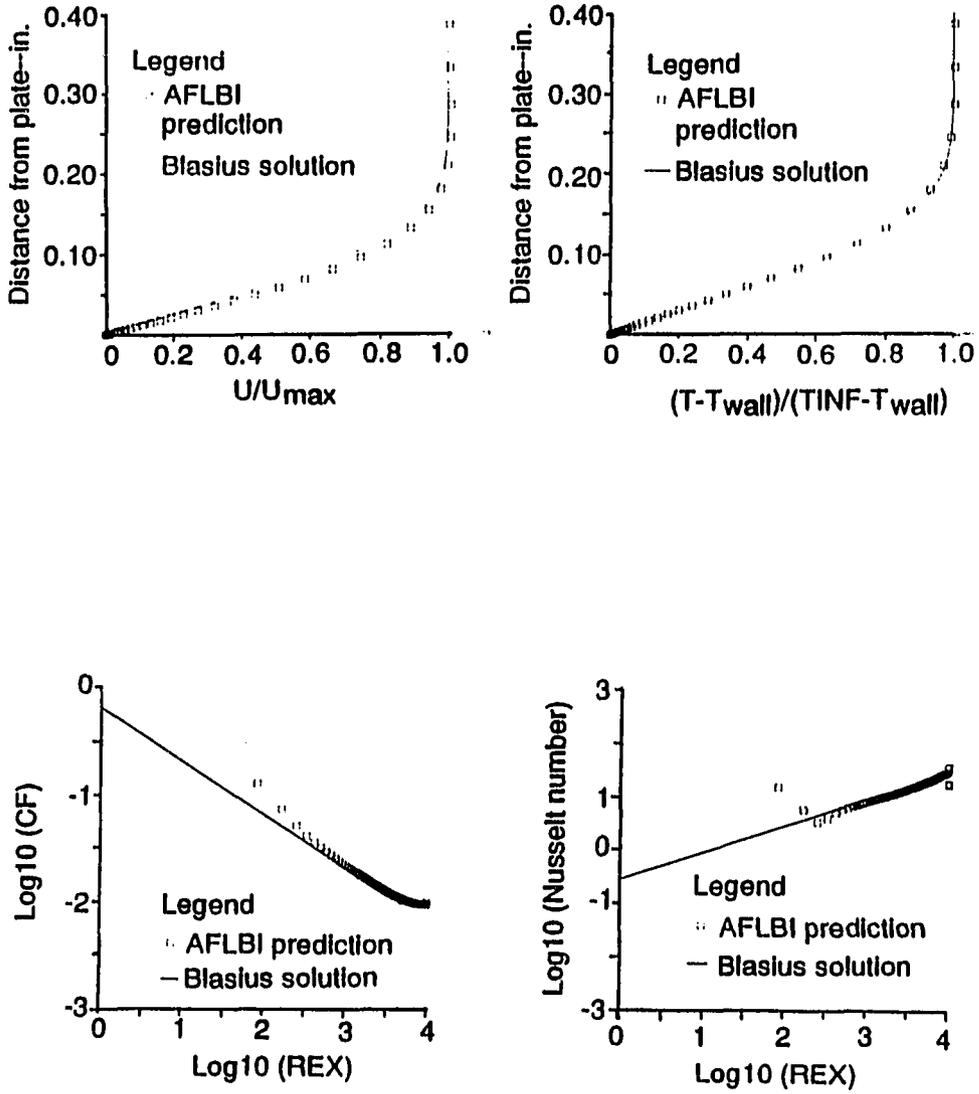


Figure 4.36: Comparison of predicted AFLBI and Blasius solution boundary-layer flow quantities for flow over a flat plate

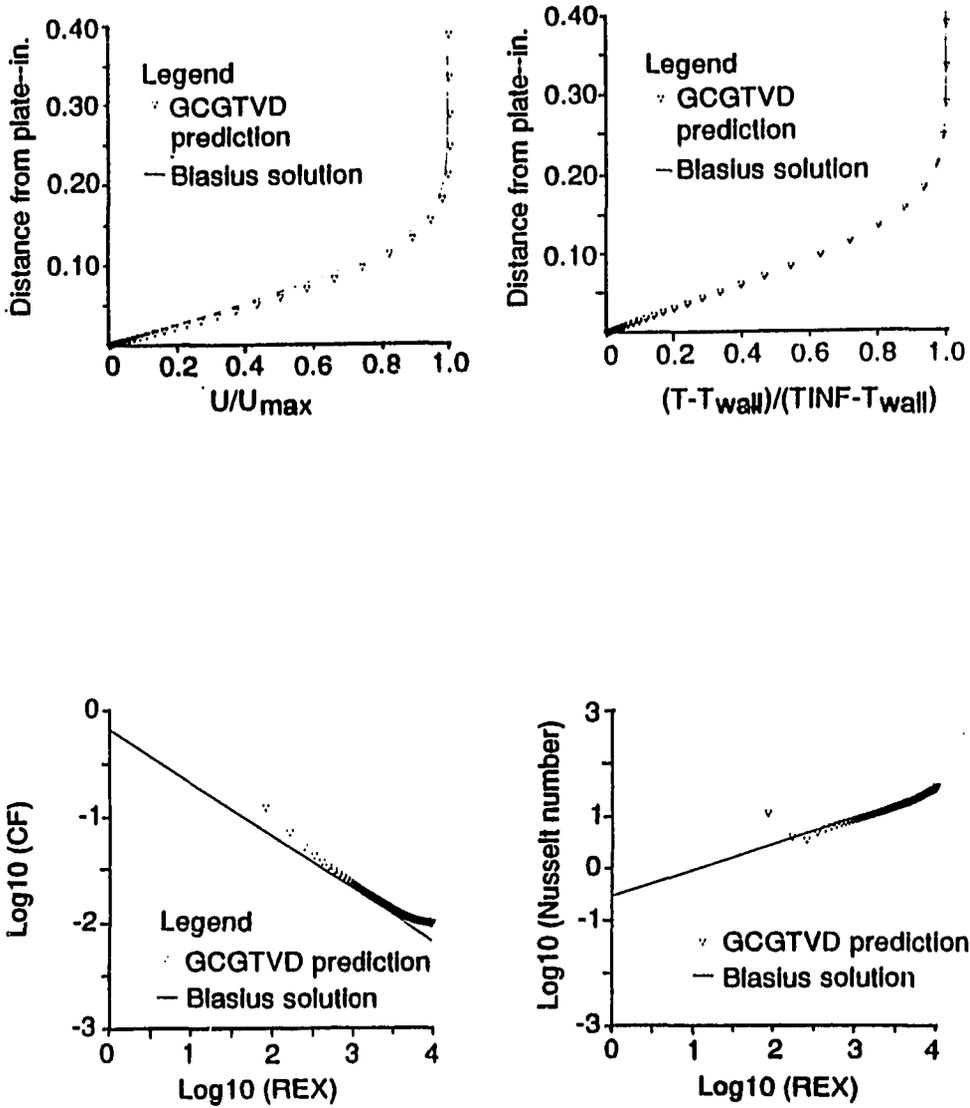


Figure 4.37: Comparison of predicted GCGTVD and Blasius solution boundary-layer flow quantities for flow over a flat plate

Navier-Stokes solution schemes [90] have demonstrated a corresponding behavior near the boundary-layer edge. A similar comparison of temperature profiles at $Re_x=10,000$ is given in Figs. 4.34-4.37. Again, the agreement between the predictions and the Blasius solution is excellent, with a small difference near the boundary-layer edge.

Overall, each code demonstrated good agreement with the Blasius solution for both velocity and temperature fields, enhancing the confidence in the accuracy of computed viscous flow results for more complex cases.

Unsteady Viscous Flow near an Oscillating Flat Plate

In order to verify the time-dependent numerics of the two-dimensional time-marching algorithms in this study, the unsteady flow resulting from a sinusoidally oscillating flat plate was chosen as an unsteady viscous flow test case. This flow is one of many discussed in a celebrated paper by Stokes [27]. The flow results from an infinite flat plate undergoing a time-periodic oscillatory translational motion in the vicinity of an otherwise stagnant fluid. A visual interpretation of this flow is given in Fig. 4.38. The time-periodic plate velocity is given by:

$$u(t)_{plate} = u_{max} \cos(\omega t) \quad (4.4)$$

where u_{max} is the maximum plate velocity, ω dictates the frequency of oscillation, and t represents time. For large values of time, the resulting flow is time-periodic and independent of the initial conditions. Stokes [27] derived an analytical expression for the time-periodic velocity profiles based on the incompressible Navier-Stokes equations. This analytical solution may be used to validate the time-marching calcu-

lations, and may be expressed as:

$$u(y, t) = u_{max} e^{-y\sqrt{\frac{\omega}{2\nu}}} \cos(\omega t - y\sqrt{\frac{\omega}{2\nu}}) \quad (4.5)$$

Several points are evident from this expression. First, the resulting velocity profiles are independent of the x location due to the infinite flat plate assumption. Second, a phase shift is evident due to the diffusion of the flat plate motion into the fluid. It is also useful to define a local skin friction coefficient as:

$$C_f = \frac{\tau_{wall}}{\frac{1}{2}\rho u_{max}^2} \quad (4.6)$$

where the analytical wall shear stress is:

$$\tau_{wall} = u_{max} \sqrt{\frac{\omega}{\nu}} \sin(\omega t - \frac{\pi}{4}) \quad (4.7)$$

It is apparent, then, that the maximum analytical shear stress lags the maximum plate velocity by 135 degrees.

In order to calculate this flow numerically, a limited computational domain was established as shown in Fig. 4.38. The length of the domain was 4.0 meters, and the height was 4.0 meters. A highly refined 41 by 51 grid was constructed for the calculations and is shown in Fig. 4.39. The minimum Δy spacing at the plate is 0.0025 meters. The flow parameters chosen for this comparison are given in Table 4.6. These values were chosen to allow a reasonable number of time steps per cycle for the explicit schemes.

The numerical solutions were obtained by utilizing a special set of boundary conditions for this flow. Along the plate surface, the plate velocity $[u(t)]_{wall}$ was specified as a function of time, along with the remaining no slip condition, $v = 0$, and

Oscillating Flat Plate Flow Definition

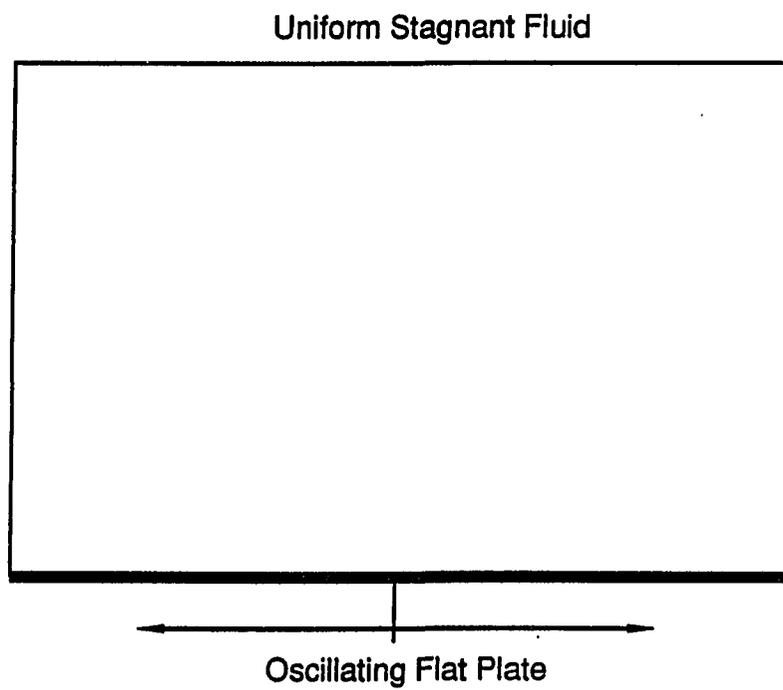


Figure 4.38: Geometry definition for oscillating flat plate flow

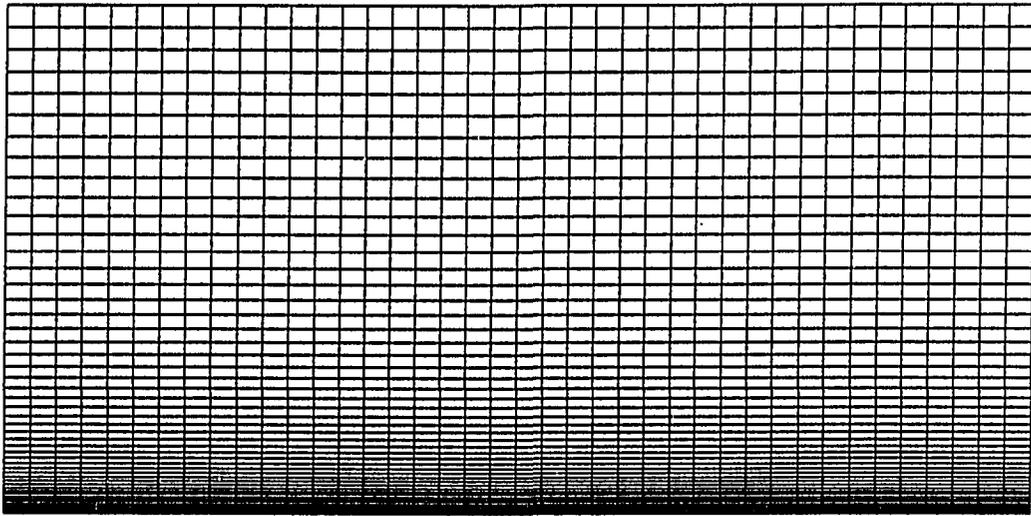


Figure 4.39: 41x51 grid for oscillating flat plate flow

Oscillating Flat Plate Skin Friction Coefficient

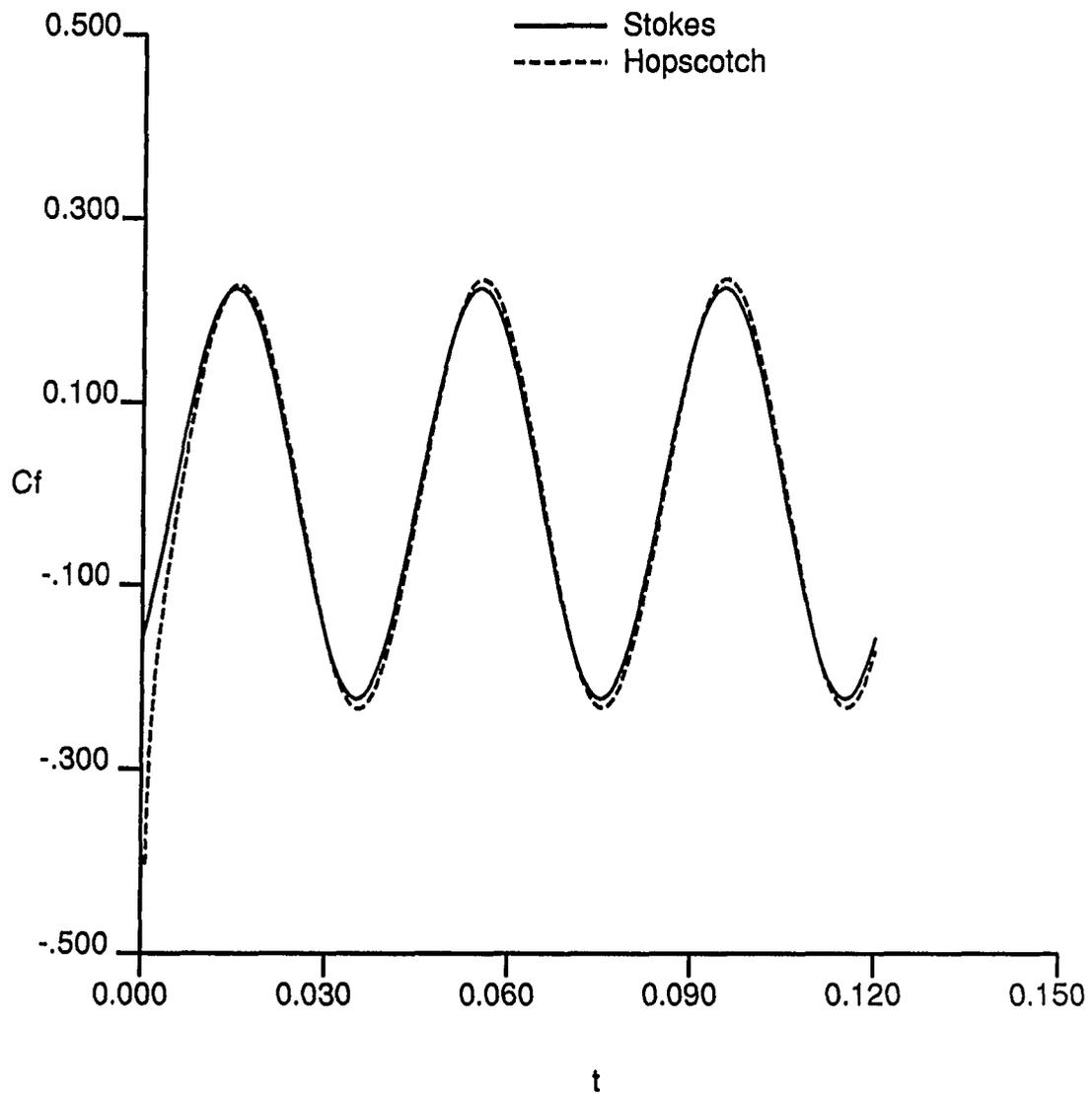


Figure 4.40: Predicted hopscotch unsteady wall skin-friction coefficient for oscillating flat plate flow

Oscillating Flat Plate Skin Friction Coefficient

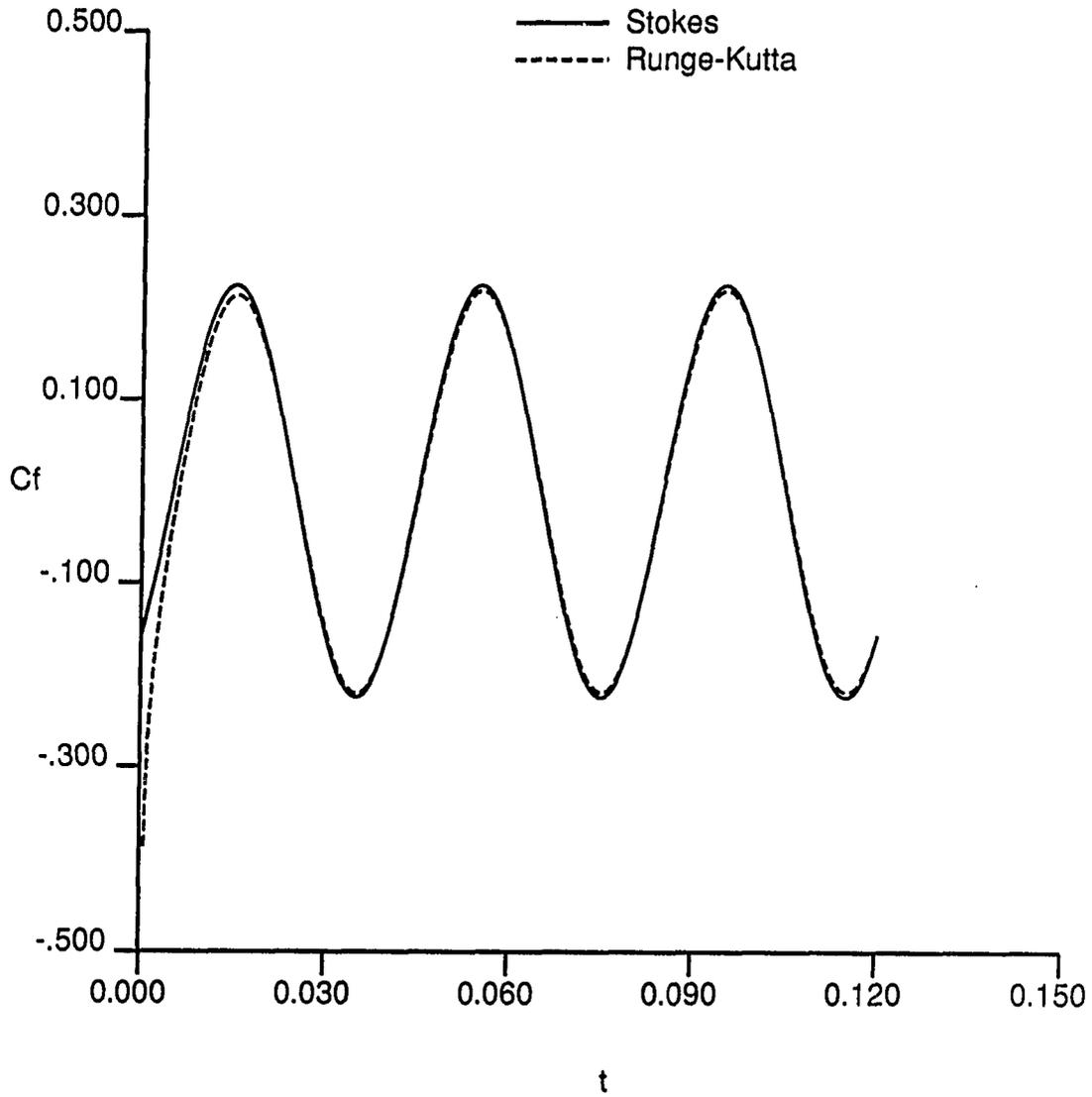


Figure 4.41: Predicted Runge-Kutta unsteady wall skin-friction coefficient for oscillating flat plate flow

Oscillating Flat Plate Skin Friction Coefficient

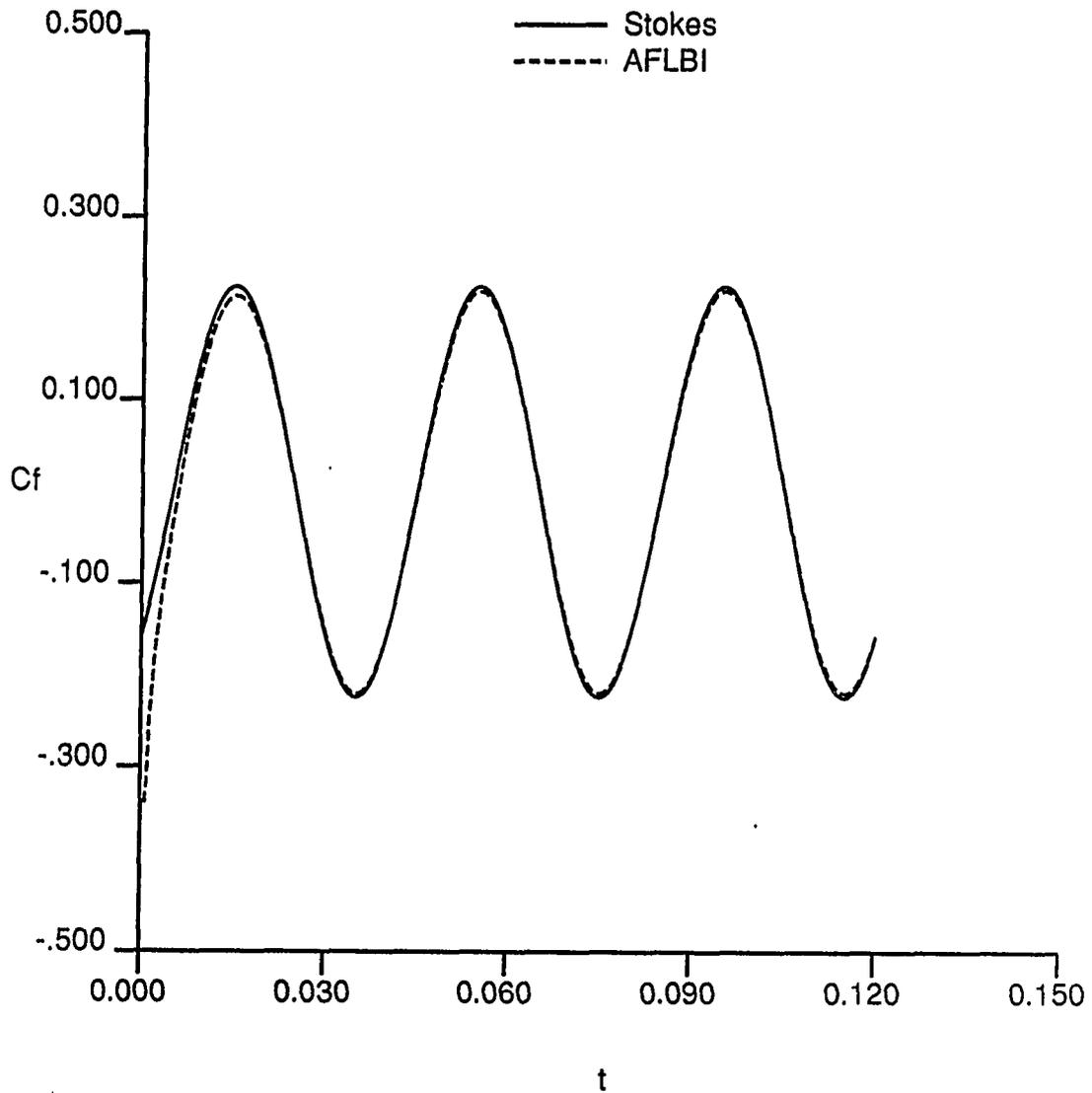


Figure 4.42: Predicted AFLBI unsteady wall skin-friction coefficient for oscillating flat plate flow

Oscillating Flat Plate Skin Friction Coefficient

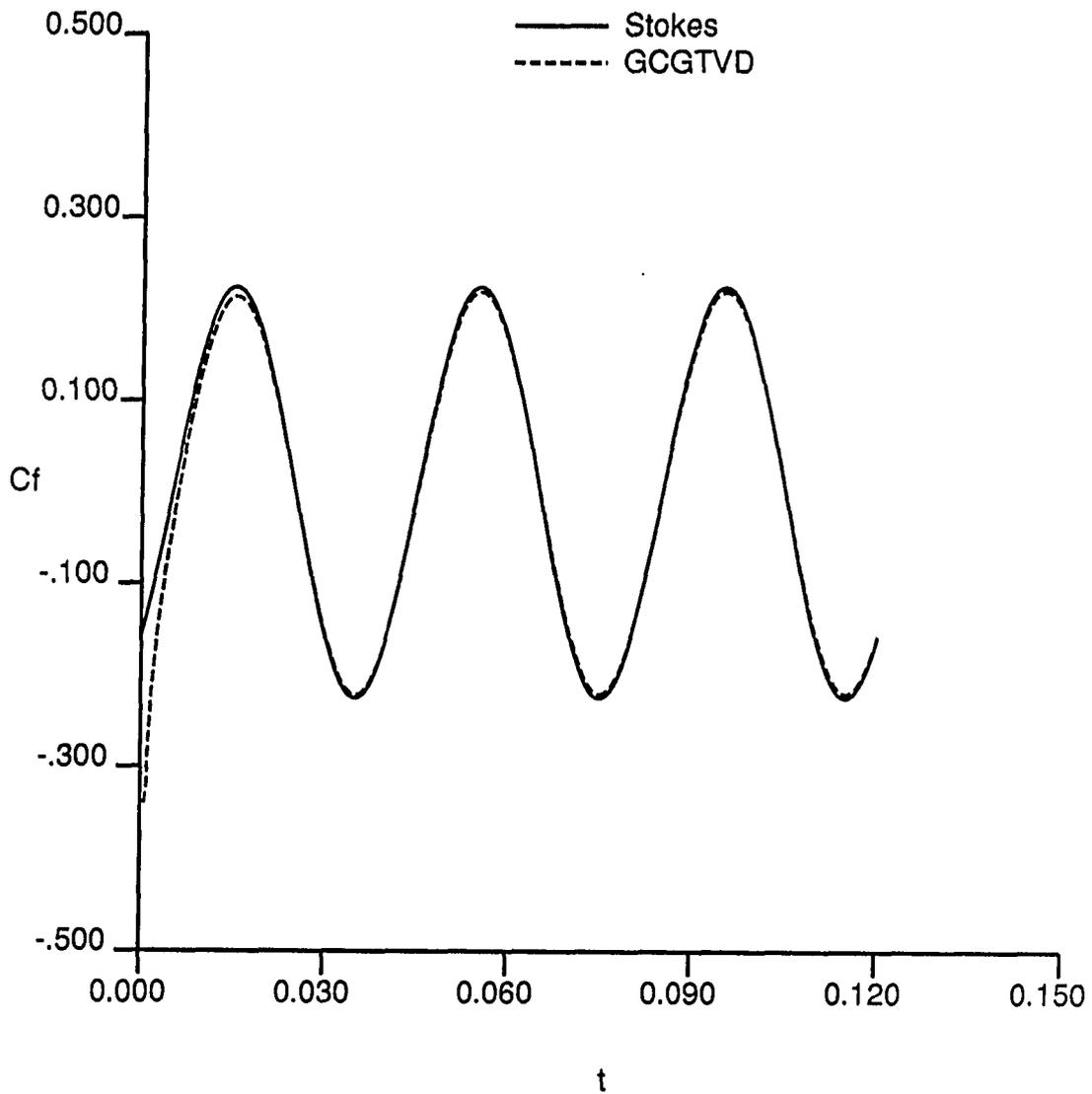


Figure 4.43: Predicted GCGTVD unsteady wall skin-friction coefficient for oscillating flat plate flow

Oscillating Flat Plate Velocity Profiles

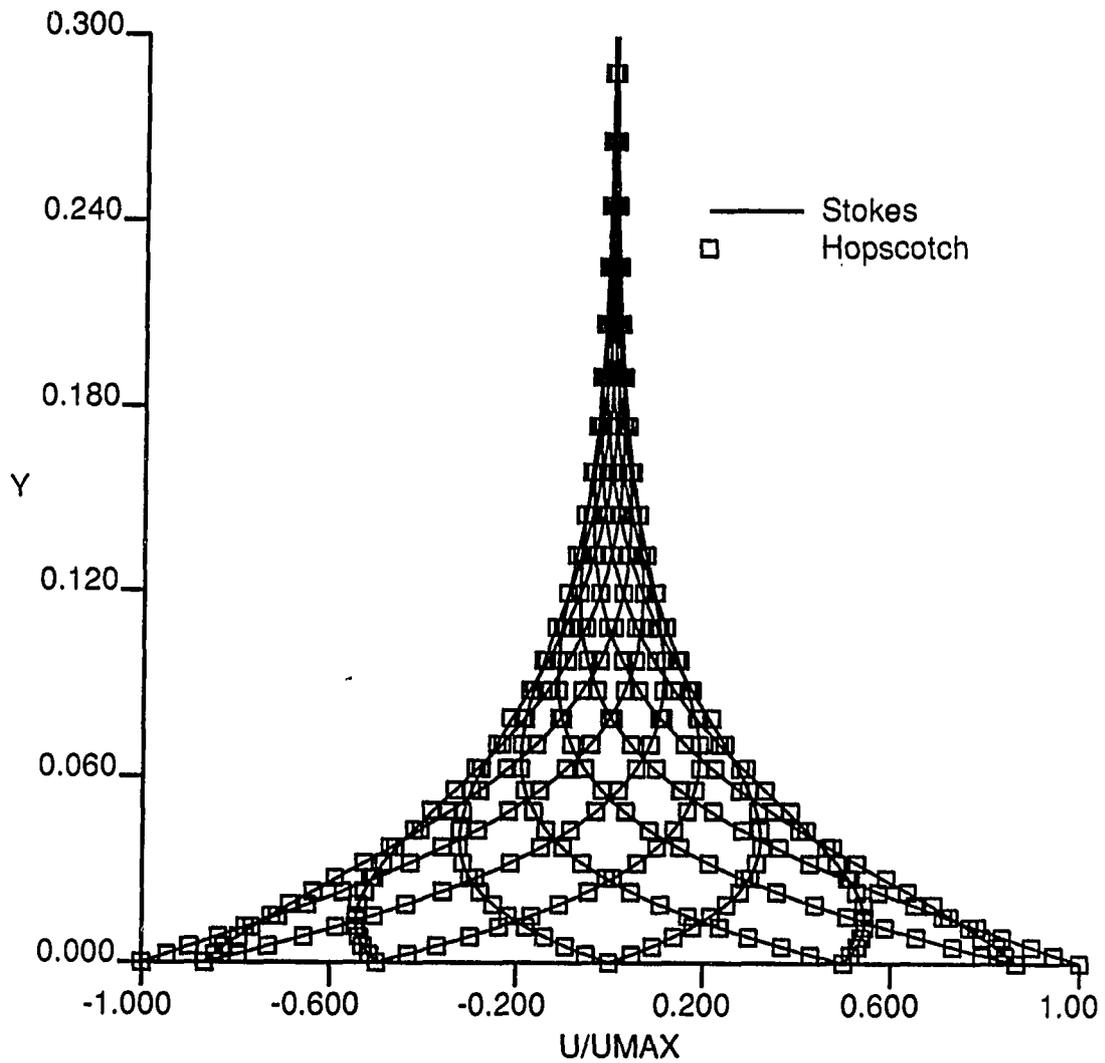


Figure 4.44: Predicted hopscotch unsteady velocity profiles for oscillating flat plate flow

Oscillating Flat Plate Velocity Profiles

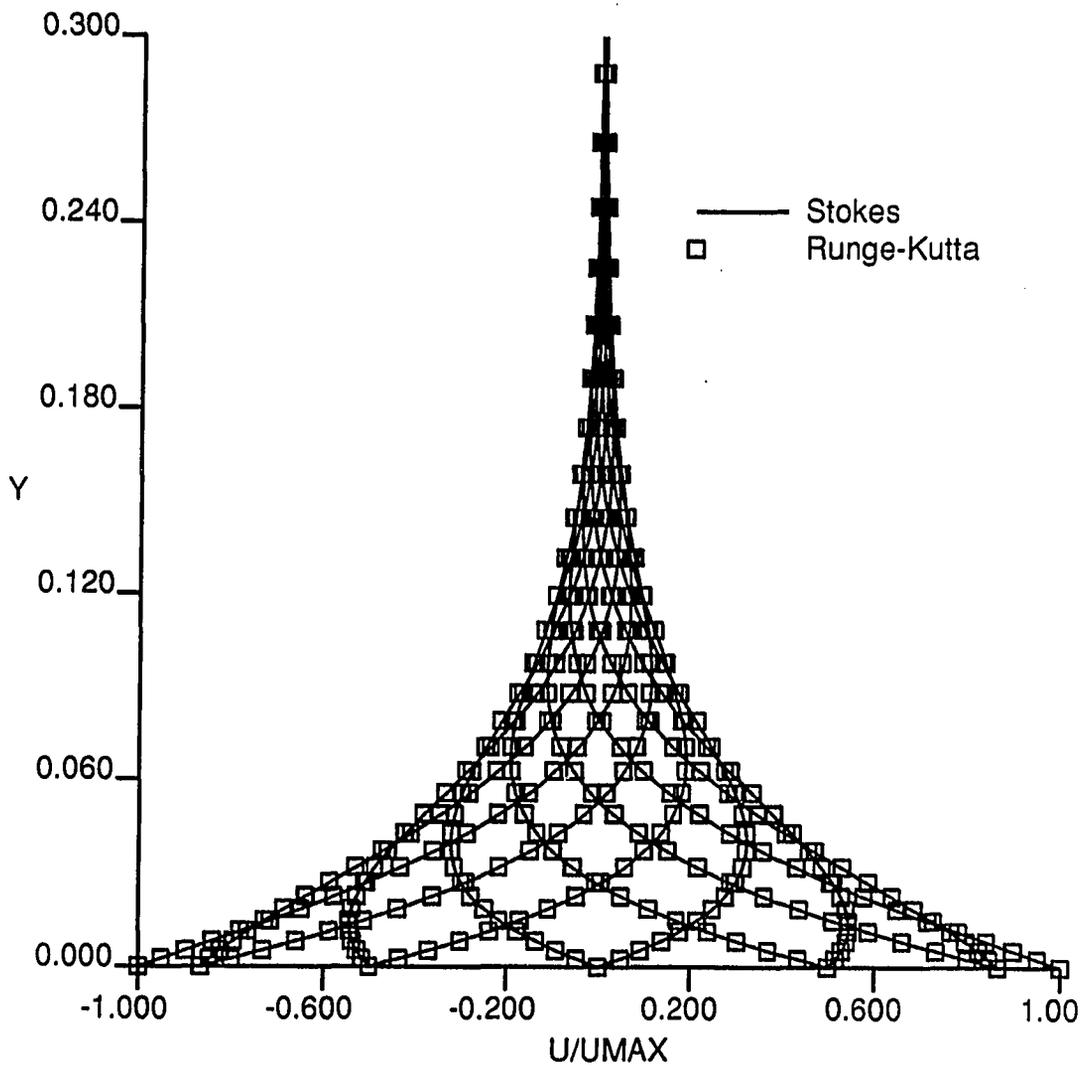


Figure 4.45: Predicted Runge-Kutta unsteady velocity profiles for oscillating flat plate flow

Oscillating Flat Plate Velocity Profiles

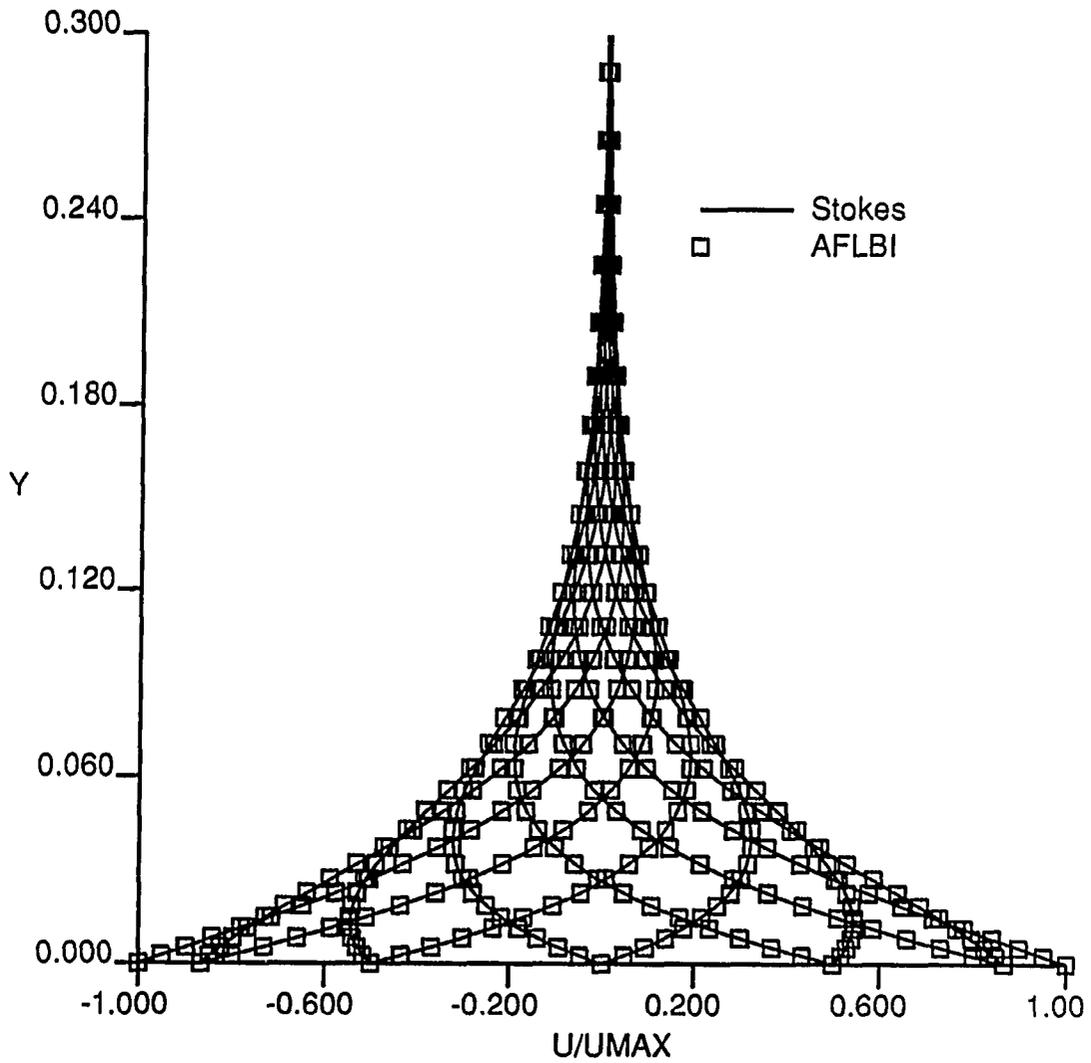


Figure 4.46: Predicted AFLBI unsteady velocity profiles for oscillating flat plate flow

Oscillating Flat Plate Velocity Profiles

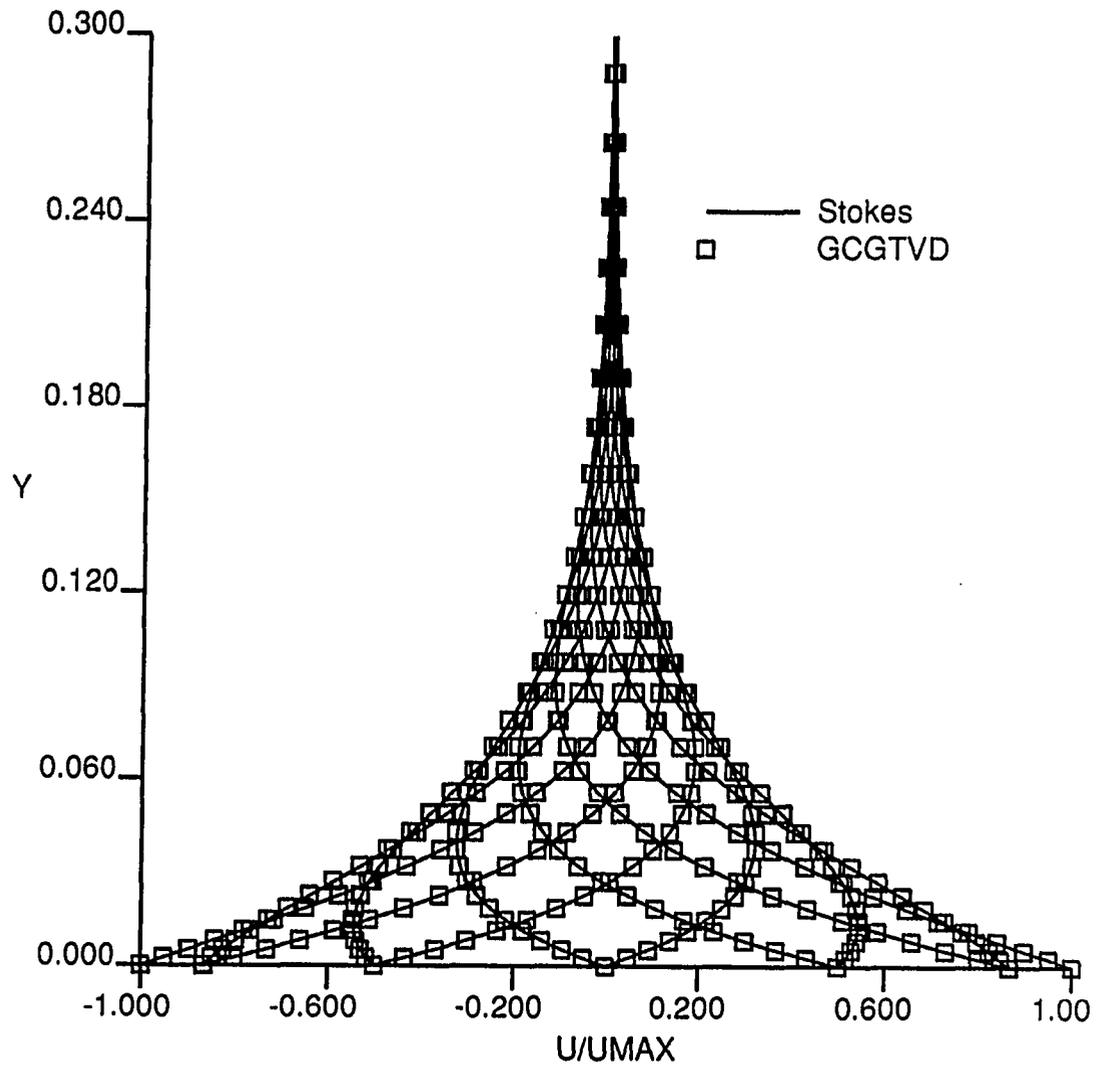


Figure 4.47: Predicted GCGTVD unsteady velocity profiles for oscillating flat plate flow

Table 4.6: Oscillating flat plate flow data

Parameter	Description	Value
ω	Frequency	50π
u_{max}	Maximum Velocity	$100.0m/s$
ρ	Density	$0.0005kg/m^3$
p	Pressure	$14.28574N/m^2$
T	Temperature	$99.5024deg.K$
μ	Viscosity	$0.0001Ns/m^2$

the usual boundary layer assumption $(\partial p/\partial y)_{wall} = 0$. A constant wall temperature equal to the fluid stagnation temperature was specified. The upper boundary of the domain simply specified a uniform, stagnant flow. The left and right boundaries were updated by assuming that all flow derivatives across the boundary were zero. For the AFLBI and GCGTVD schemes, these conditions were treated implicitly through the usual Taylor series linearization in time. The flow was initially assumed to be stagnant, and the plate oscillation is imposed impulsively at $t = 0$ to initiate the calculation. Each of the codes were advanced in time the equivalent of three cycles of the plate oscillation. Analytical evidence suggests that the startup transient for such a flow exists for approximately one cycle of the oscillation after which the time-periodic solution dominates. The run parameters for each of the codes is given in Table 4.7. The time increment for each code was varied based on the maximum allowable CFL number and an attempt to utilize approximately the same overall CPU time for each run.

A comparison of the time-asymptotic analytical and predicted time-dependent skin-friction coefficient for each of the codes is given in Figs. 4.40-4.43. The initial startup transient is immediately obvious in each case, and appears to last for about

one cycle of the plate oscillation. Beyond this point, excellent agreement with the time-periodic analytical result is demonstrated for each code. It should be noted, however, that a very small phase error is evident in the calculation utilizing the hopscotch algorithm, and more importantly, the results based on the hopscotch code were sensitive to the damping coefficients used. This may be due to the odd-even point calculation procedure in the hopscotch algorithm.

A comparison of the predicted velocity profiles with the analytical solution for 30 degree increments of the plate oscillation are given in Figs. 4.44-4.47 for each of the time-marching algorithms. The predicted velocity profiles were taken from the final oscillation cycle of the calculation. Again, excellent agreement is achieved between the calculated and analytical results for this case.

A comparison of the CPU time requirements for each code is given in Table 4.7. Each code executes at a slightly different CFL number chosen such that the resulting overall CPU time for is each code is roughly the same. The increase in speed per iteration observed for each code in this case compared to the previous cases was due to an upgraded FORTRAN compiler on the Cray computer.

The successful calculation and comparison with the analytical solution for this flow verifies the accuracy of the numerical formulation of the time-marching algorithms for unsteady viscous flows and lends optimism for the success of more demanding time-dependent flow problems.

Table 4.7: Oscillating flat plate CPU time comparison

Method	Time Step	Iterations	CPU seconds	CPU seconds Grid Point / It.
Hopscotch	1.111×10^{-5}	10,800	149.946	6.639×10^{-6}
Runge-Kutta	2.222×10^{-5}	5,400	216.187	1.915×10^{-5}
AFLBI	6.666×10^{-5}	1,800	175.098	4.652×10^{-5}
GCGTVD	6.666×10^{-5}	1,800	261.660	6.952×10^{-5}

Unsteady Viscous Flow about an Impulsively Started Cylinder

A classic problem of interest in fluid mechanics is the transient flow associated with the impulsively started uniform motion of a circular cylinder. A graphical depiction of this flow is given in Fig. 4.48. This flow was studied experimentally by Bouard and Countenceau [107], and Honji and Tenada [108], and numerically by Cebeci [109], Kawamura et al. [110], Chamberlain [111], Loc and Bouard [112], Giorgini and Pravia [113], and Rinaldo and Giorgini [114]. Although the geometry and conditions dictating such a flow are relatively simple, the resulting transient flowfield is extremely complex and has thus far defied any analytical solution. The flow demonstrates a marked Reynolds number dependence, as does the steady flow about a circular cylinder. Bouard and Countenceau [107] performed a detailed flow visualization study of impulsively started cylinder flows in the Reynolds number range of $40 < Re_D < 10^4$. Several interesting phenomena are immediately observable from their results. The most striking feature resulting from this flow is the transient development of a symmetric pair of vortices on the leeward side of the cylinder. These vortices are similar to the symmetric standing vortices observed in low Reynolds number steady cylinder flows. As time progresses, these vortices grow both in extent and strength until

a steady recirculating flow is established for low Reynolds numbers, or the classic higher Reynolds number time-periodic vortex shedding flow pattern develops.

Depending on Reynolds number, other additional, more subtle features develop in the flow in conjunction with the primary recirculating regions. An interesting phenomena occurs for a flow Reynolds number $1000 < Re_d < 10^4$. In addition to the primary vortices, smaller secondary recirculating regions develop near the separation point on the surface of the cylinder as shown in Fig. 4.49. These vortices are the result of the interaction between the primary vortices and the developing boundary layer on the cylinder. This interaction shows a strong Reynolds number dependence. Each of these flow phenomena presents challenging aspects for numerical flow simulation, and therefore this test case was chosen to allow a detailed examination of unsteady viscous flow predictions for complicated time-dependent flows.

This particular problem has many interesting aspects. It would appear that this effort is the first reported attempt to calculate this flow using time-marching compressible flow solvers. Previous calculations have all utilized an incompressible streamfunction-vorticity formulation. The application of the compressible flow solvers to a low Reynolds number and low Mach number flow will serve to test the behavior of each algorithm under less than ideal circumstances. It can be argued that even a high speed unsteady flow is likely to have regions dominated by low Reynolds number behavior, such as in a stagnation or separated flow zone.

In order to calculate this flow, a 361x99 numerical grid was generated and is pictured in Fig. 4.50. The cylinder diameter is arbitrarily chosen to be 2.0 meters. Again, the isolated cylinder is approximated by a cascade of cylinders with a large

periodic spacing (10 cylinder diameters) to minimize the aerodynamic interference between cylinders. The inlet plane is placed 5 cylinder diameters upstream of the cylinder, while the exit plane is placed 8 cylinder diameters downstream of the cylinder. The exit plane was placed somewhat farther away to avoid interference between the transient viscous wake and the downstream boundary. The average grid spacing at the cylinder surface is $0.005m$. Two cases were examined with different Reynolds numbers ($Re = 3000$, $Re = 9500$). Calculations were performed for each case and for each code such that the anticipated overall CPU time per code was consistent. Since some rather large execution times were anticipated for this test case, the largest possible time step is desirable during the course of this calculation. Since the GCGTVD scheme requires a rather large amount of CPU time, it was felt that the larger time steps permitted by the full linearization given in Equation(3.69) was perhaps the best approach available for this scheme, and therefore this approach was utilized in place of the diagonalized linearization given in Equation (3.70).

Reynolds Number = 3000

The first case examined was for a Reynolds number of 3000. Tables 4.8-4.9 lists the flow parameters used for this set of calculations. This particular set of data was chosen to satisfy the Reynolds number criteria, and to allow a reasonable number of time steps for the time step restricted explicit schemes. The time interval is expressed nondimensionally as $t = \bar{t}U_{max}/D$.

Previous calculations of this type have stressed the importance of intelligent specification of the initial conditions. Although the flow is ideally designed as a

Table 4.8: Impulsively started cylinder flow data, $Re=3000$

Parameter	Description	Value
D	Cylinder diameter	1.0m
u_{max}	Impulsive velocity	20.0m/s
ρ	Density	0.15kg/m ³
p	Pressure	971.817N/m ²
T	Temperature	22.5629deg.K
μ	Viscosity	0.001Ns/m ²
t_{tot}	Total time period	0.125s

Table 4.9: Impulsively started cylinder flow CPU time comparison, $Re=3000$

Method	Time Step	Iterations	CPU seconds	CPU seconds Grid Point / It.
Hopscotch	5.0x10(-6)	25,000	1759.00	1.968x10(-6)
Runge-Kutta	1.0x10(-5)	12,500	1799.00	5.033x10(-6)
AFLBI	2.5x10(-5)	5,000	2500.00	2.798x10(-5)
GCGTVD	2.5x10(-5)	5,000	3600.00	4.029x10(-5)

true impulsive start, this is physically unrealistic, and experimental evidence suggests that there is actually a finite time interval over which the acceleration of the cylinder takes place ($t = 0.02$). Loc and Bouard [112] utilized a fourth order compact streamfunction-vorticity scheme to calculate this flow. They proposed the use a steady state solution at a Reynolds number equal to 5 as the initial condition for the transient calculation. Unfortunately, several details concerning the exact implementation of this approach were unavailable and attempts to initiate the present calculations in this manner were unacceptable due to the overall extent of the viscous effects for such a low Reynolds number flow. Attempts to simulate a true impulsive start were also thwarted by oscillations generated in the numerical solution by the sudden start. Instead, the initial acceleration period of the cylinder as it approaches the impulsive velocity is approximately modelled according to the following formula:

$$u(t) = U_{max}(1.0 - e^{\frac{-5t}{t_1}}) \quad (4.8)$$

where, from the experimental evidence $t_1 = 0.02$. Each code utilized a moving grid fixed to the cylinder (in other words, the terms x_t, y_t in Eqs. 2.14-2.15 are nonzero). The grid speed was varied according to Eq. (4.8) until the impulsive velocity was reached, beyond which the grid velocity was constant.

The numerical boundary conditions for this flow were equivalent to the normal viscous cascade flow boundary conditions previously described. Since the overall flow is unsteady, nonreflective inlet and exit boundary conditions were used in place of the steady-state method of characteristic boundary conditions (see the Appendix for details). Since the cylinder actually moves through a stationary fluid, a non-zero velocity is imposed on the cylinder surface in place of the the no-slip condition.

The CPU and time step requirements for each code are listed in Table 4.9 for this case. Each code was operated at a realistic maximum constant time step based on stability considerations.

Experimental flow visualization photographs are given in Figs.4.51-4.53 for the $Re=3000$ case, for nondimensional times ranging from 0.5 to 2.5. The development of the secondary and primary vortices are immediately obvious.

A comparison of the predicted u velocity profiles along the downstream line of symmetry for the cylinder and experimental measurements are given for various times in Figs. 4.54, 4.55, 4.56, and 4.57 for the hopscotch, Runge-Kutta, AFLBI, and GCGTVD schemes, respectively. In each case, the unsteady velocities in the recirculating region are well predicted, as well as the extent of the primary recirculating region. An interesting observation in this case is that the maximum velocity in the recirculating region exceeds the cylinder velocity.

On a more qualitative level, predicted instantaneous streamline patterns for this flow are given at 5 different times in Figs. 4.58-4.62, 4.63-4.67, 4.68-4.72, and 4.73-4.77 for the hopscotch, Runge-Kutta, AFLBI, and GCGTVD algorithms, respectively. The flow visualization pictures presented by Loc and Bouard [112] are also given for comparison. The symmetric development of the primary vortices is clearly visible in each calculation. The development of the secondary vortices is also visible. Attempts to utilize smaller grids for the prediction of this flow were successful in predicting the primary recirculating region, but were unsuccessful in predicting these secondary vortices.

The similarity between the flow visualization and the simulations are obvious,

and serve to verify the accuracy of the numerical predictions.

Reynolds Number = 9500

The second case tested was for a Reynolds number of 9500. In this case, the freestream conditions were changed to produce the change in Reynolds number. Again, the 361x99 grid system pictured in Fig. 4.50 was used for this case. The change in freestream conditions resulted in an increase in the permissible time step, and therefore the larger Reynolds number case actually requires a somewhat smaller computational effort.

The computational parameters for the $Re=9500$ case are given in Table 4.10. The associated calculation time step data and CPU times are given in Table 4.11. Due to the success of the $Re=3000$ case, the calculation time step for the implicit schemes was increased to $CFL \approx 8$ for this case.

Experimental flow visualization photographs are given in Figs. 4.78-4.79 for the $Re=9500$ case. The development of the primary recirculating region is considerably different in this case. The primary vortex develops as a bulge near the upper and lower most part of the cylinder. This bulge then moves toward the centerline from either side, causing several secondary, and eventually tertiary vortices to form in its wake.

Again, predicted unsteady velocity profiles are compared with experimental data in Figs. 4.80-4.83. The agreement with the experimental measurements is again very good, especially for the implicit schemes. This is actually thought to be due to the more sophisticated damping schemes in the two implicit codes. Predicted instant-

Table 4.10: Impulsively started cylinder flow data, $Re=9500$

Parameter	Description	Value
D	Cylinder diameter	1.0m
u_{max}	Impulsive velocity	20.0m/s
ρ	Density	0.475kg/m ³
p	Pressure	3077.42N/m ²
T	Temperature	22.5629deg.K
μ	Viscosity	0.001Ns/m ²
t_{tot}	Total time period	0.1s

Table 4.11: Impulsively started cylinder flow CPU time comparison, $Re=9500$

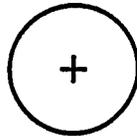
Method	Time Step	Iterations	CPU seconds	CPU seconds Grid Point / It.
Hopscotch	5.0x10(-6)	25,000	1759.00	1.968x10(-6)
Runge-Kutta	1.0x10(-5)	10,000	1799.00	5.033x10(-6)
AFLBI	4.0x10(-5)	2,500	2500.00	2.798x10(-5)
GCGTVD	4.0x10(-5)	2,500	3600.00	4.029x10(-5)

neous streamline patterns for this flow are given at 6 different times in Figs. 4.84-4.89, 4.90-4.95, 4.96-4.101, and 4.102-4.107 for the hopscotch, Runge-Kutta, AFLBI, and GCGTVD algorithms, respectively. The explicit schemes do not show strong evidence of the tertiary vortices, while the implicit schemes do. The experimental flow visualization provides some evidence of the tertiary vortex development.

This case has served to demonstrate the quality and fine detail possible with the application of computational fluid dynamics for the prediction of complex unsteady viscous flows.

Impulsively Started Cylinder Flow Description

Time $t < 0$, Cylinder Velocity = 0.0



Time $t > 0$, Cylinder Velocity = Constant

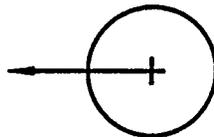
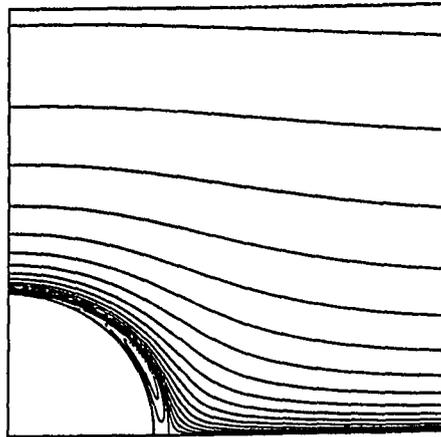
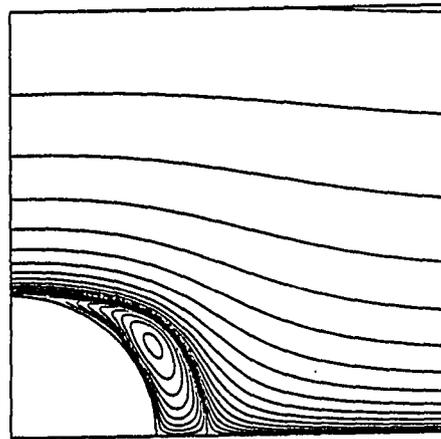


Figure 4.48: Impulsively started cylinder flow description

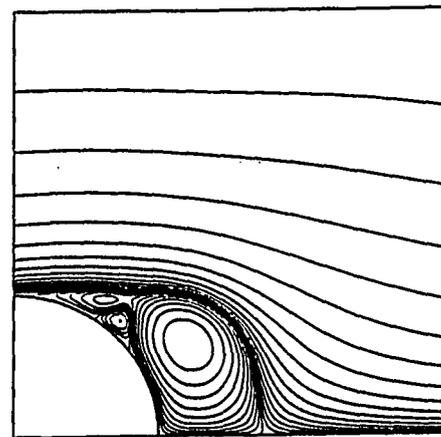
Increasing
Time



Initiation of
Primary Vortex



Growth of
Primary Vortex



Initiation of
Secondary Vortices

Figure 4.49: Transient vortex development of impulsively started cylinder flow

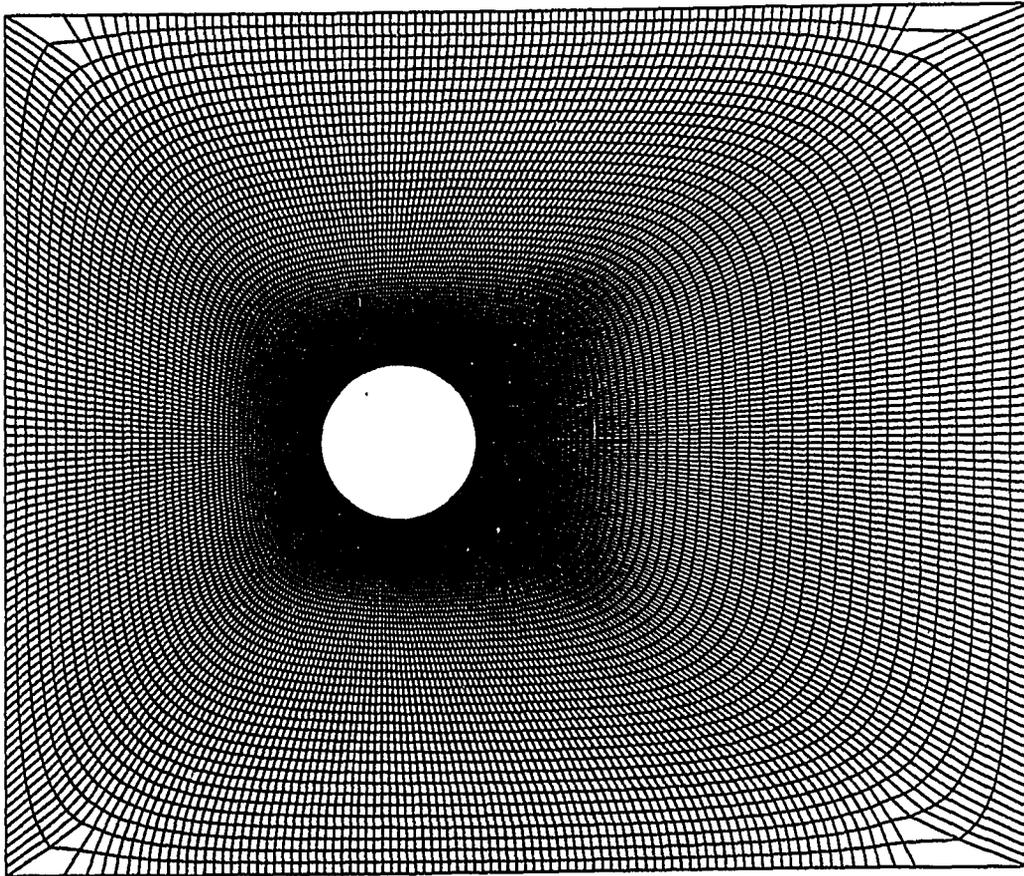


Figure 4.50: 361x99 grid for impulsively started cylinder flow



Figure 4.51: Experimental flow visualization for impulsively started cylinder ($Re=3000$), A: $Vt/D=0.5$, B: $Vt/D=1.0$



Figure 4.52: Experimental flow visualization for impulsively started cylinder ($Re=3000$), A: $Vt/D=1.5$, B: $Vt/D=2.0$



Figure 4.53: Experimental flow visualization for impulsively started cylinder ($Re=3000$), $Vt/D=2.5$

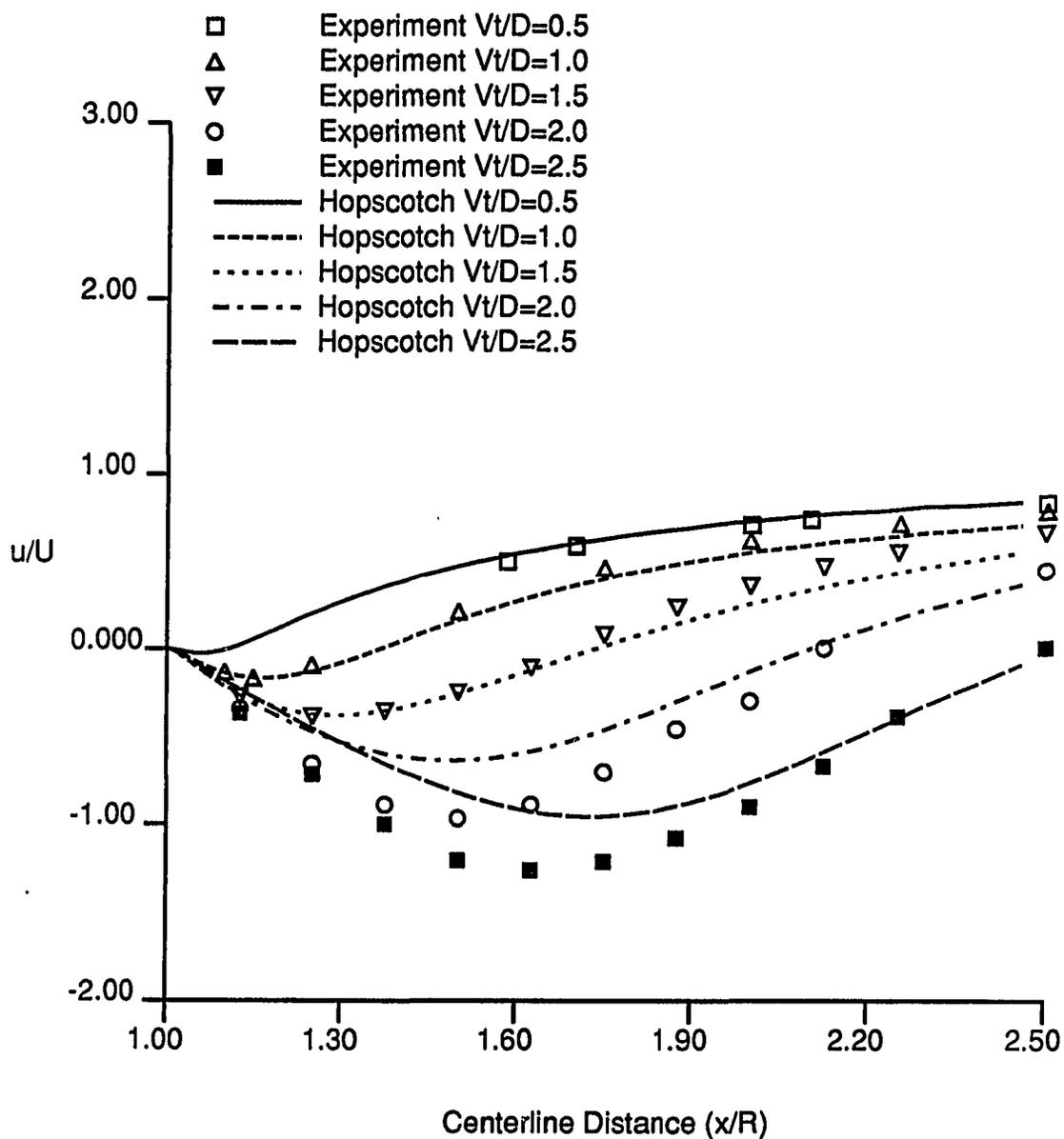
Hopscotch Impulsively Started Cylinder $Re=3000$ 

Figure 4.54: Comparison of predicted hopscotch and experimental unsteady velocity profiles for impulsively started cylinder flow ($Re=3000$)

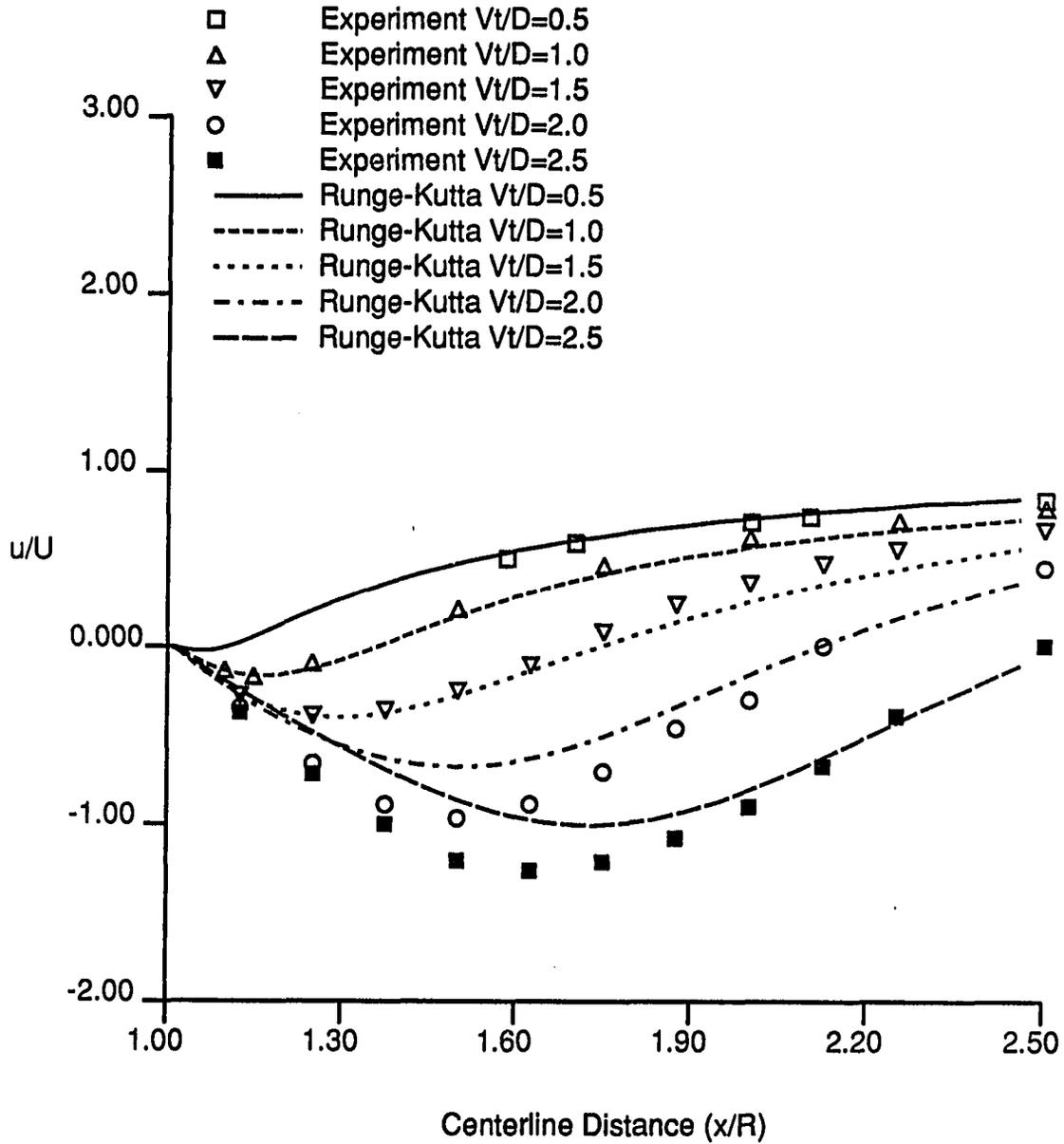
Runge-Kutta Impulsively Started Cylinder $Re=3000$ 

Figure 4.55: Comparison of predicted Runge-Kutta and experimental unsteady velocity profiles for impulsively started cylinder flow ($Re=3000$)

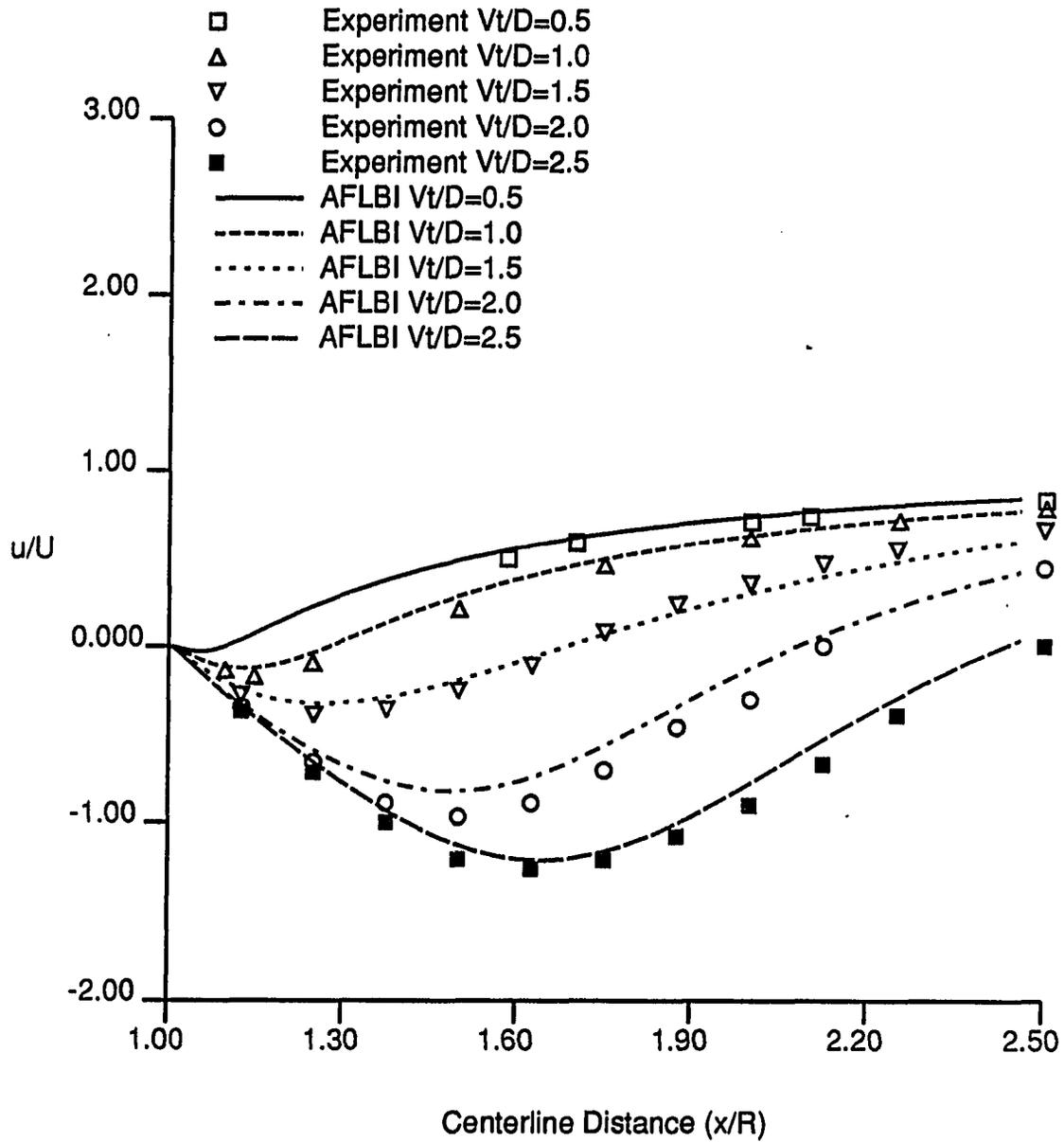
AFLBI Impulsively Started Cylinder $Re=3000$ 

Figure 4.56: Comparison of predicted AFLBI and experimental unsteady velocity profiles for impulsively started cylinder flow ($Re=3000$)

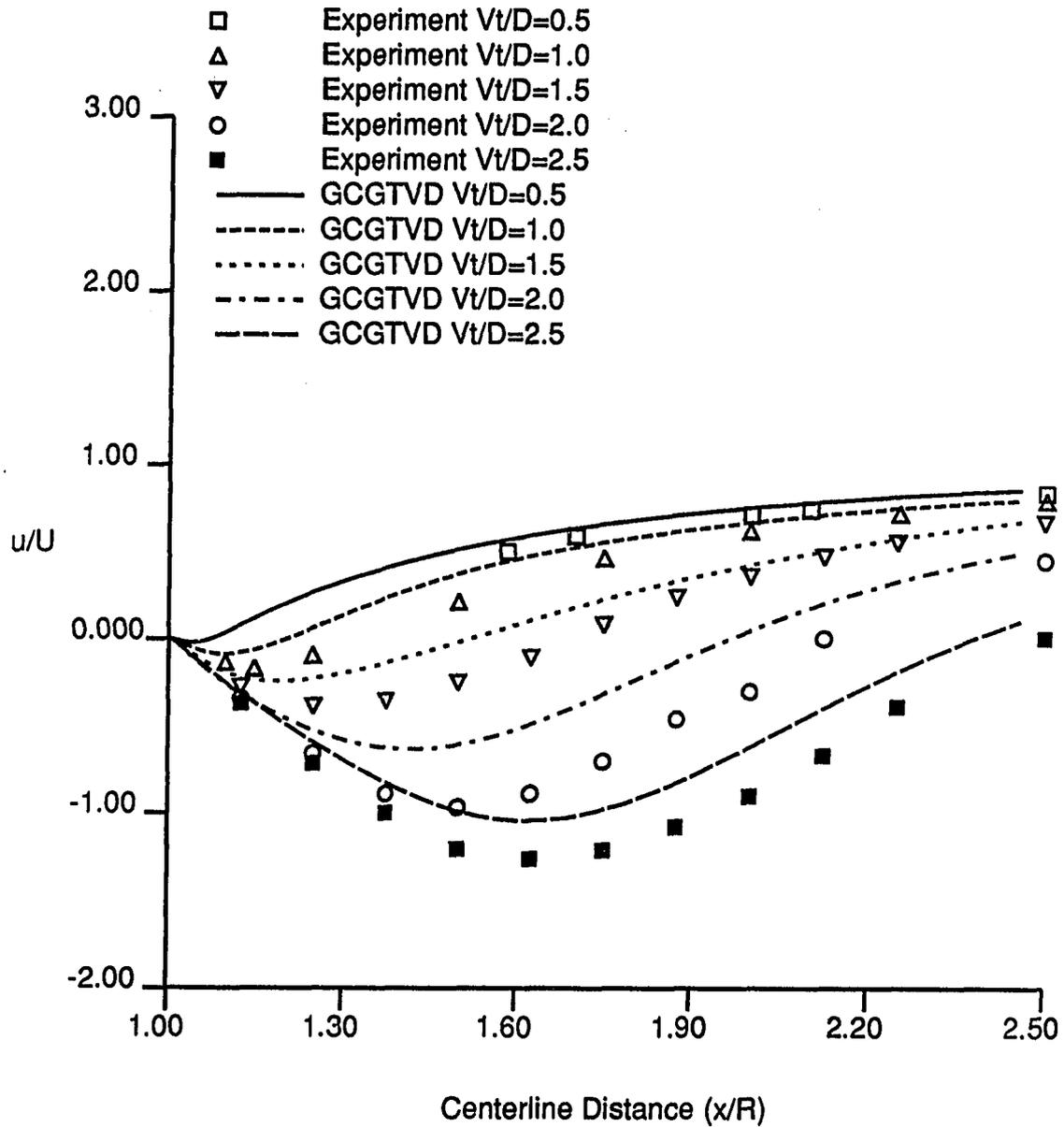
GCGTVD Impulsively Started Cylinder $Re=3000$ 

Figure 4.57: Comparison of predicted GCGTVD and experimental unsteady velocity profiles for impulsively started cylinder flow ($Re=3000$)

Hopscotch Impulsively Started Cylinder $Re=3000$ $Vt/D=0.5$

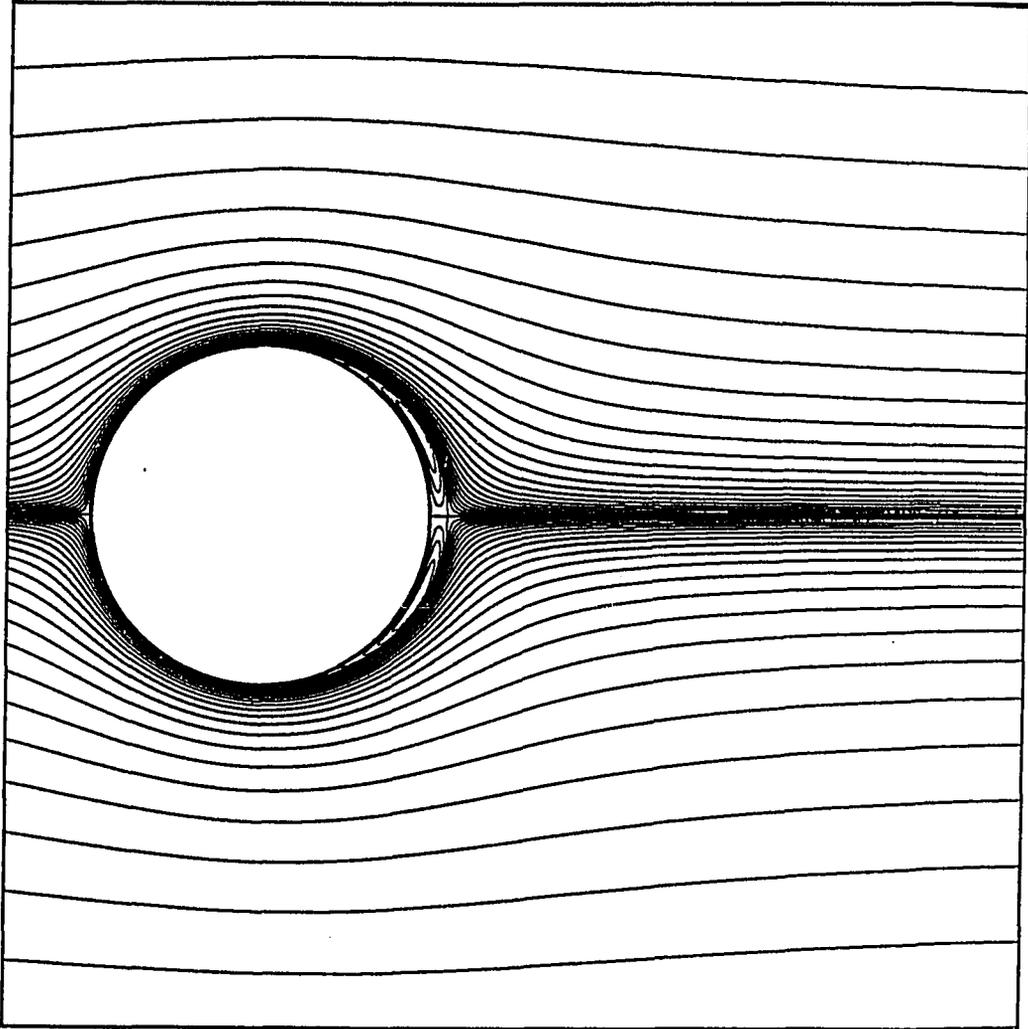


Figure 4.58: Hopscotch predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=0.5$)

Hopscotch Impulsively Started Cylinder $Re=3000$ $Vt/D=1.0$

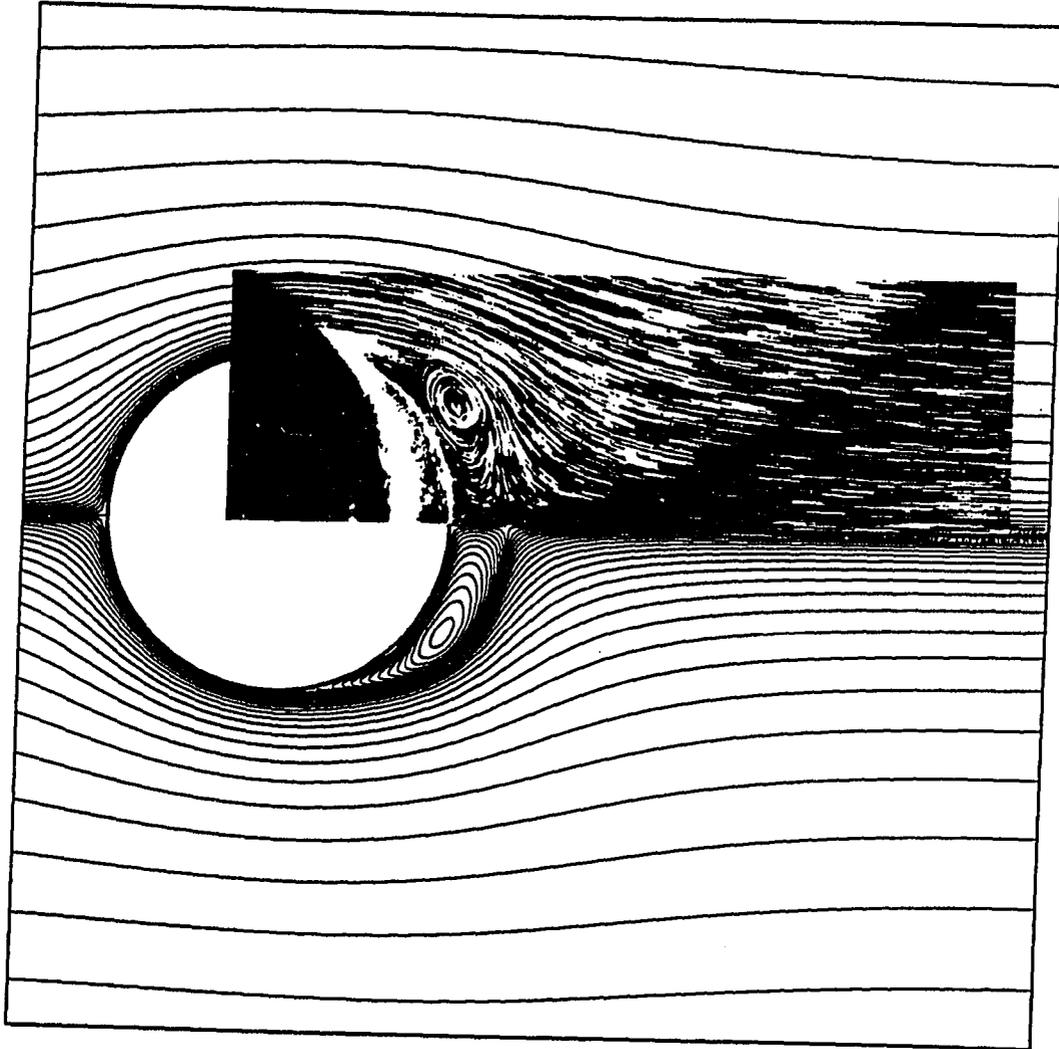


Figure 4.59: Hopscotch predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=1.0$)

Hopscotch Impulsively Started Cylinder $Re=3000$ $Vt/D=1.5$

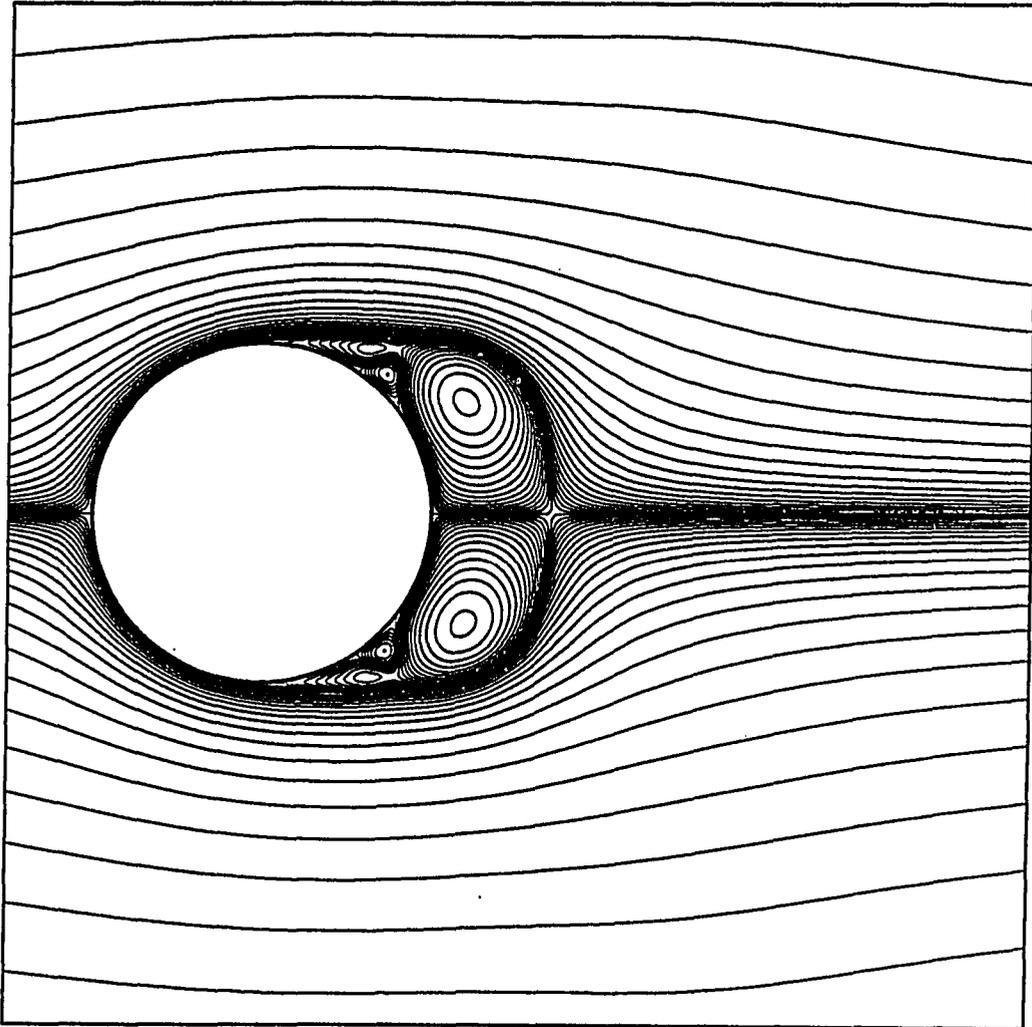


Figure 4.60: Hopscotch predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=1.5$)

Hopscotch Impulsively Started Cylinder $Re=3000$ $Vt/D=2.0$

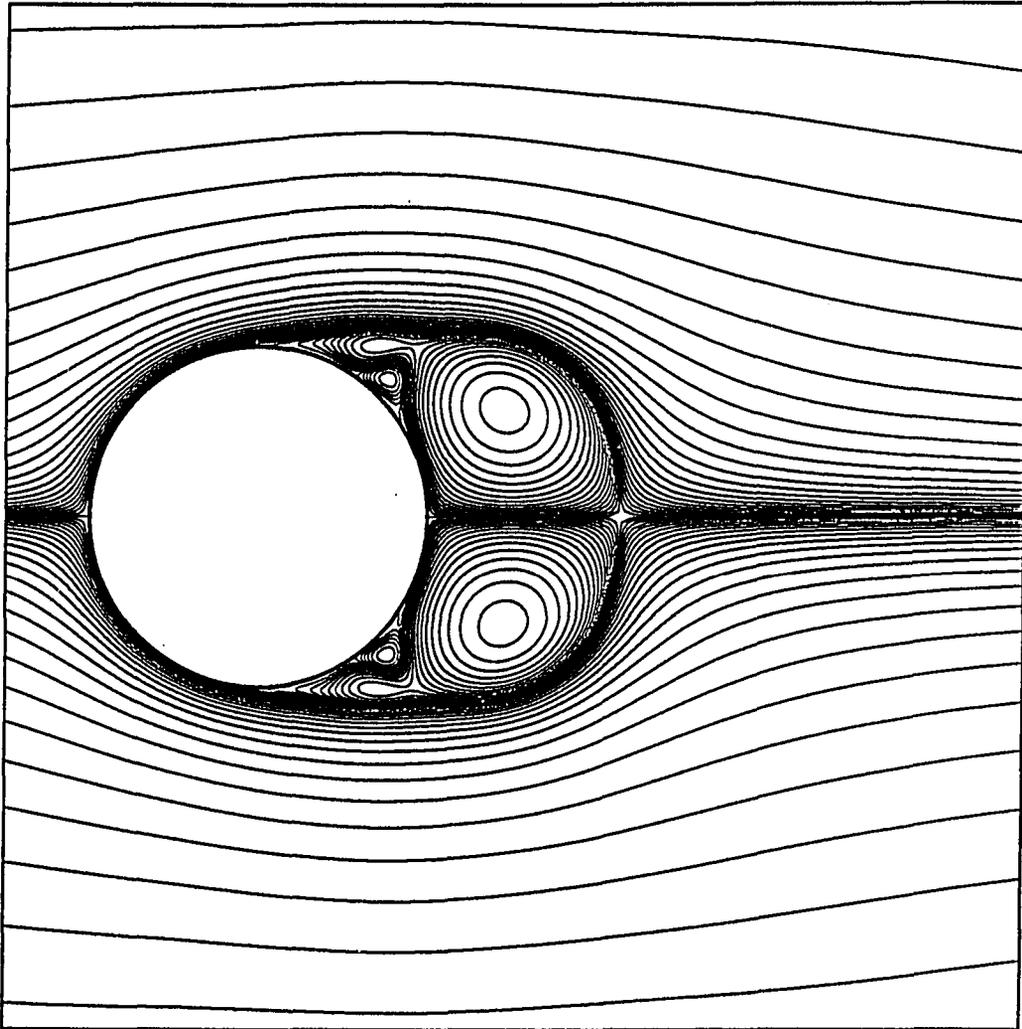


Figure 4.61: Hopscotch predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=2.0$)

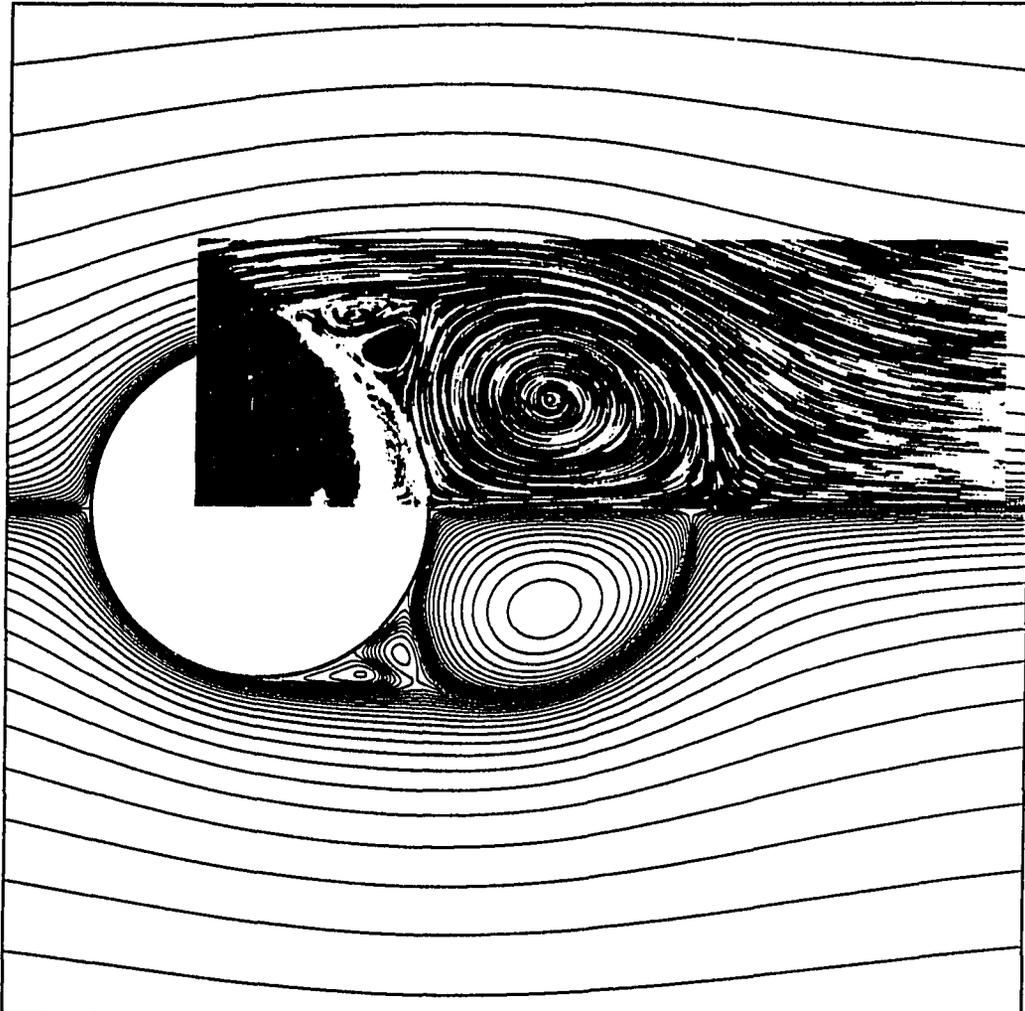
Hopscotch Impulsively Started Cylinder $Re=3000$ $Vt/D=2.5$ 

Figure 4.62: Hopscotch predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=2.5$)

Runge-Kutta Impulsively Started Cylinder $Re=3000$ $Vt/D=0.5$

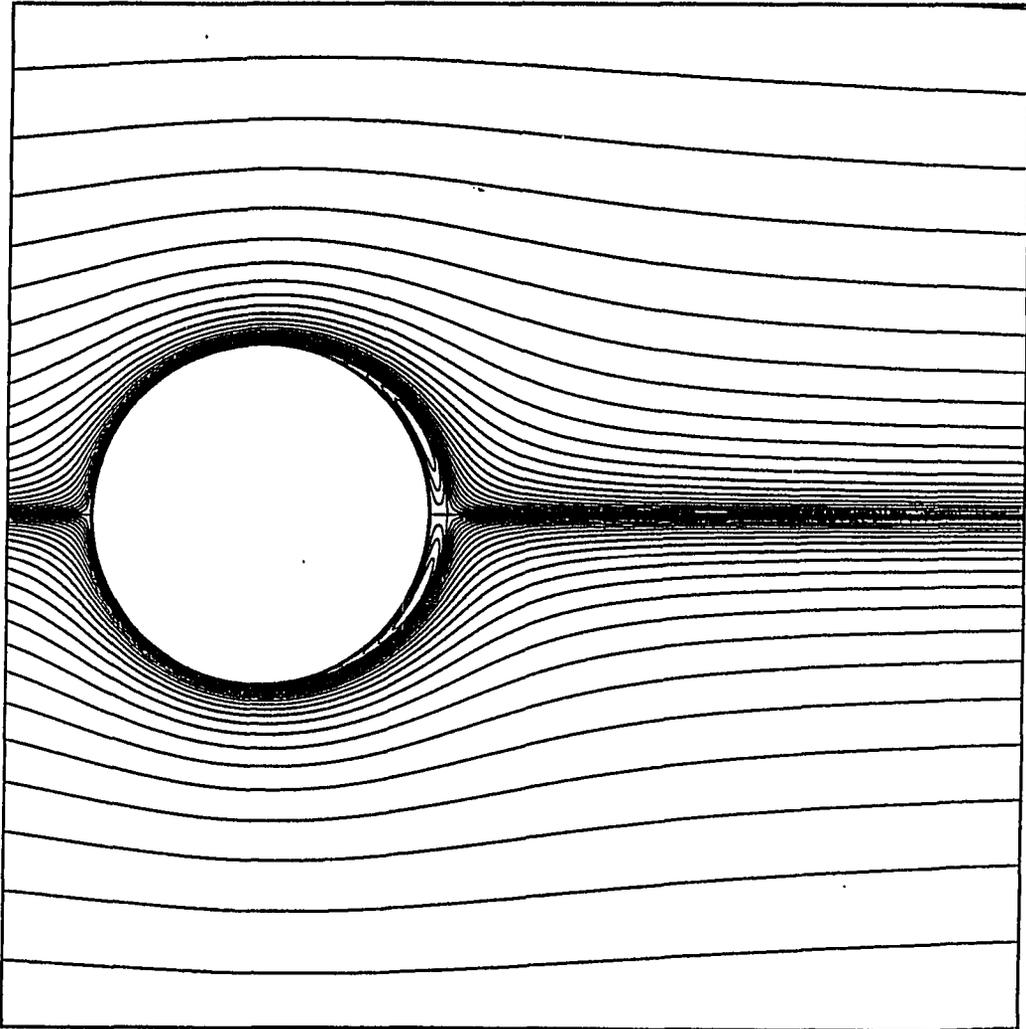


Figure 4.63: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=0.5$)

Runge-Kutta Impulsively Started Cylinder $Re=3000$ $Vt/D=1.0$

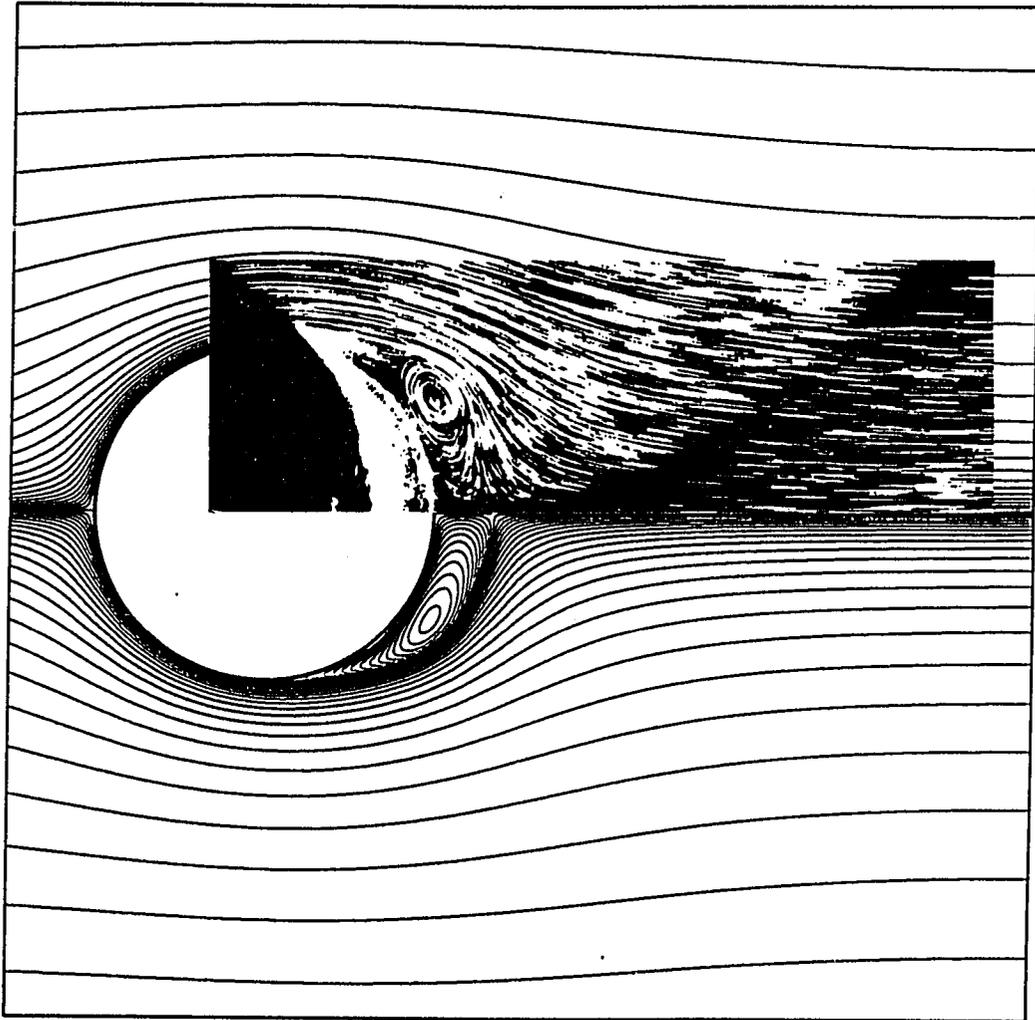


Figure 4.64: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=1.0$)

Runge-Kutta Impulsively Started Cylinder $Re=3000$ $Vt/D=1.5$

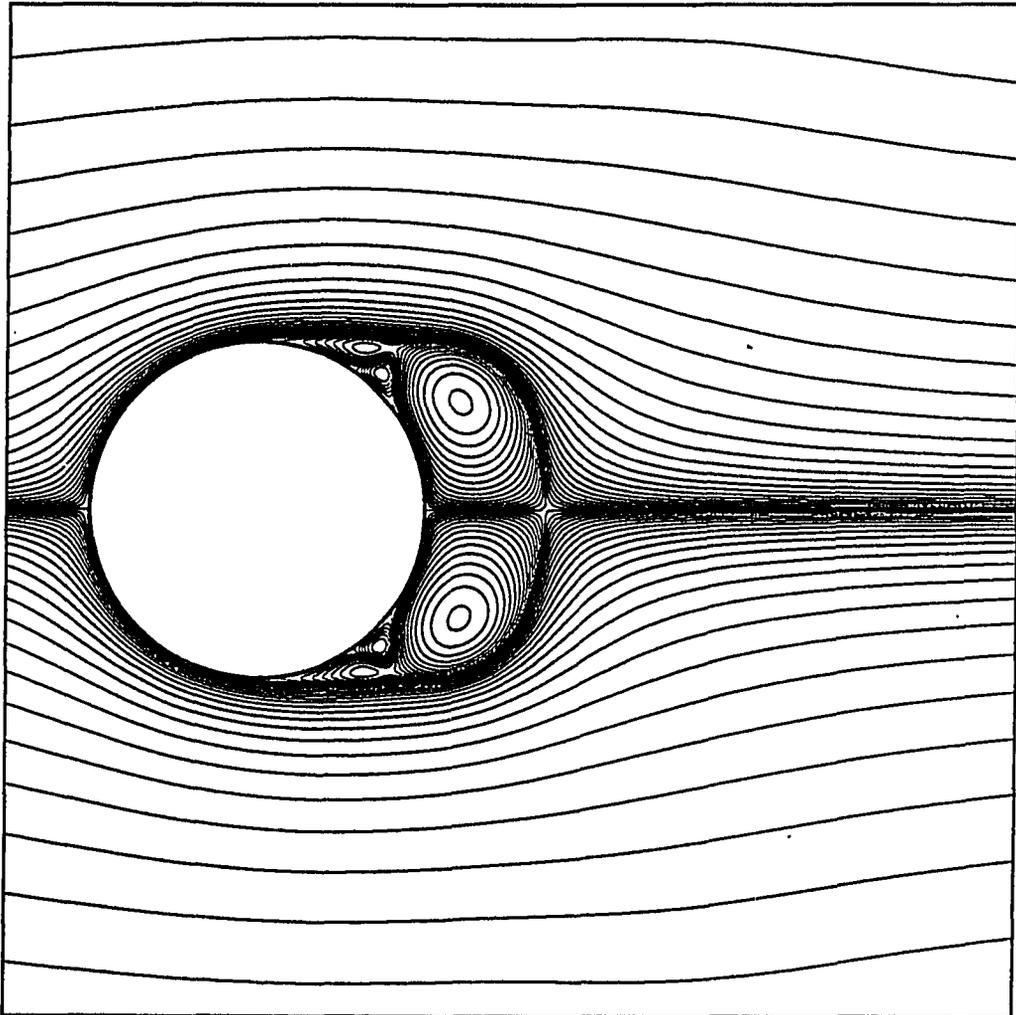


Figure 4.65: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=1.5$)

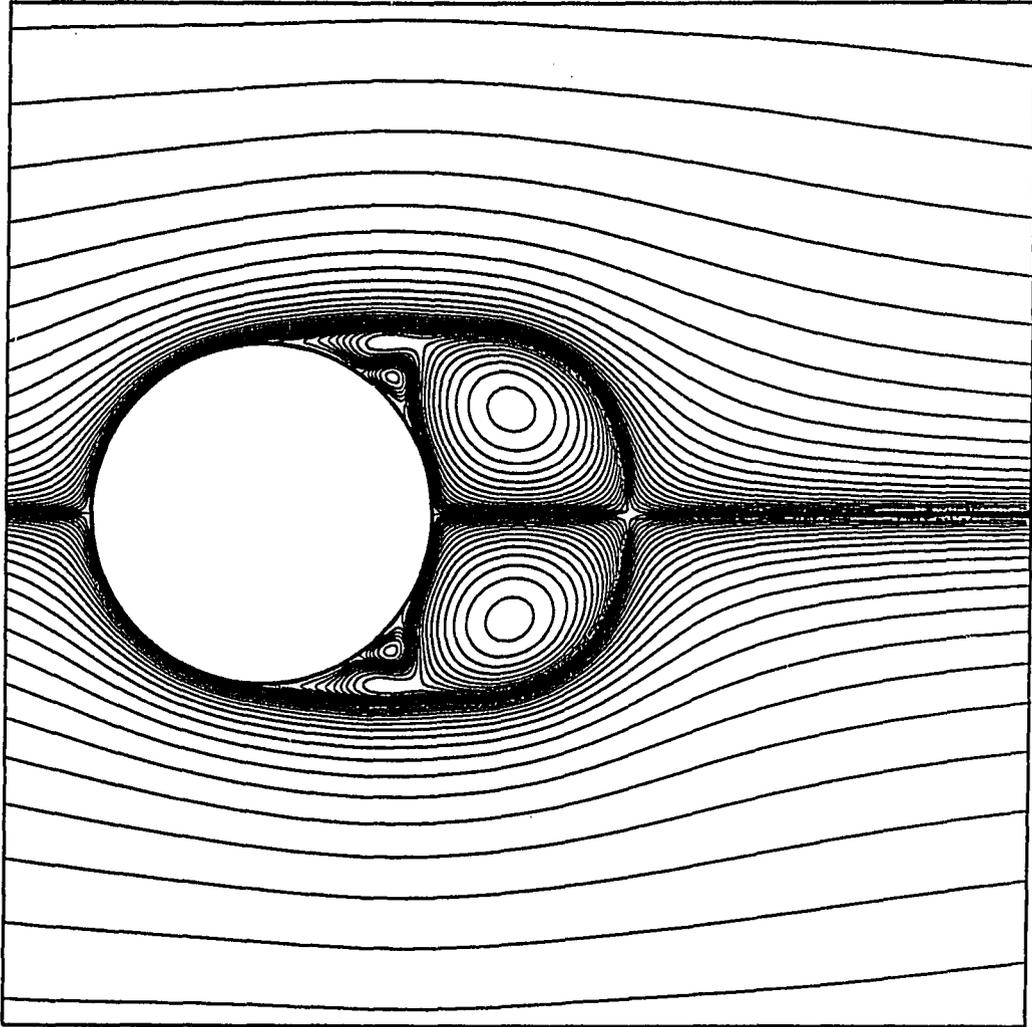
Runge-Kutta Impulsively Started Cylinder $Re=3000$ $Vt/D=2.0$ 

Figure 4.66: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=2.0$)

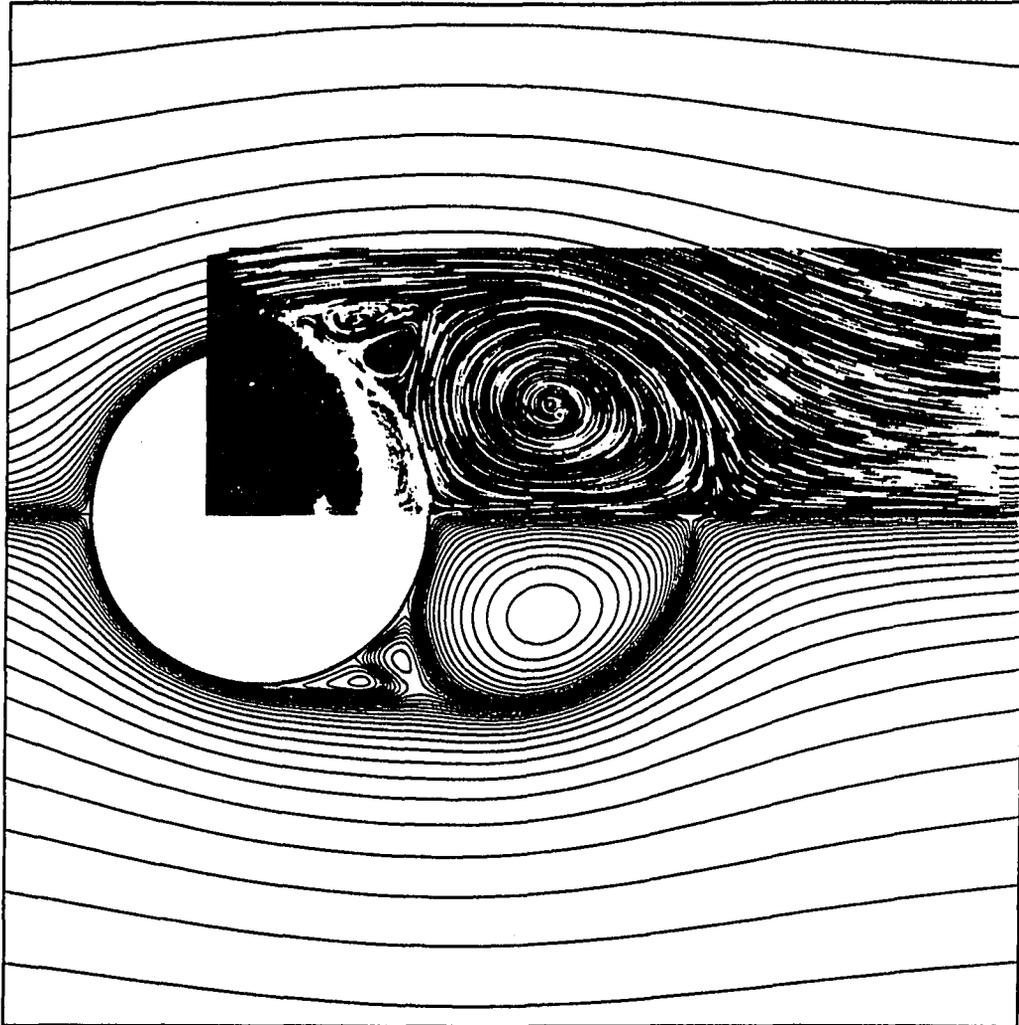
Runge-Kutta Impulsively Started Cylinder $Re=3000$ $Vt/D=2.5$ 

Figure 4.67: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=2.5$)

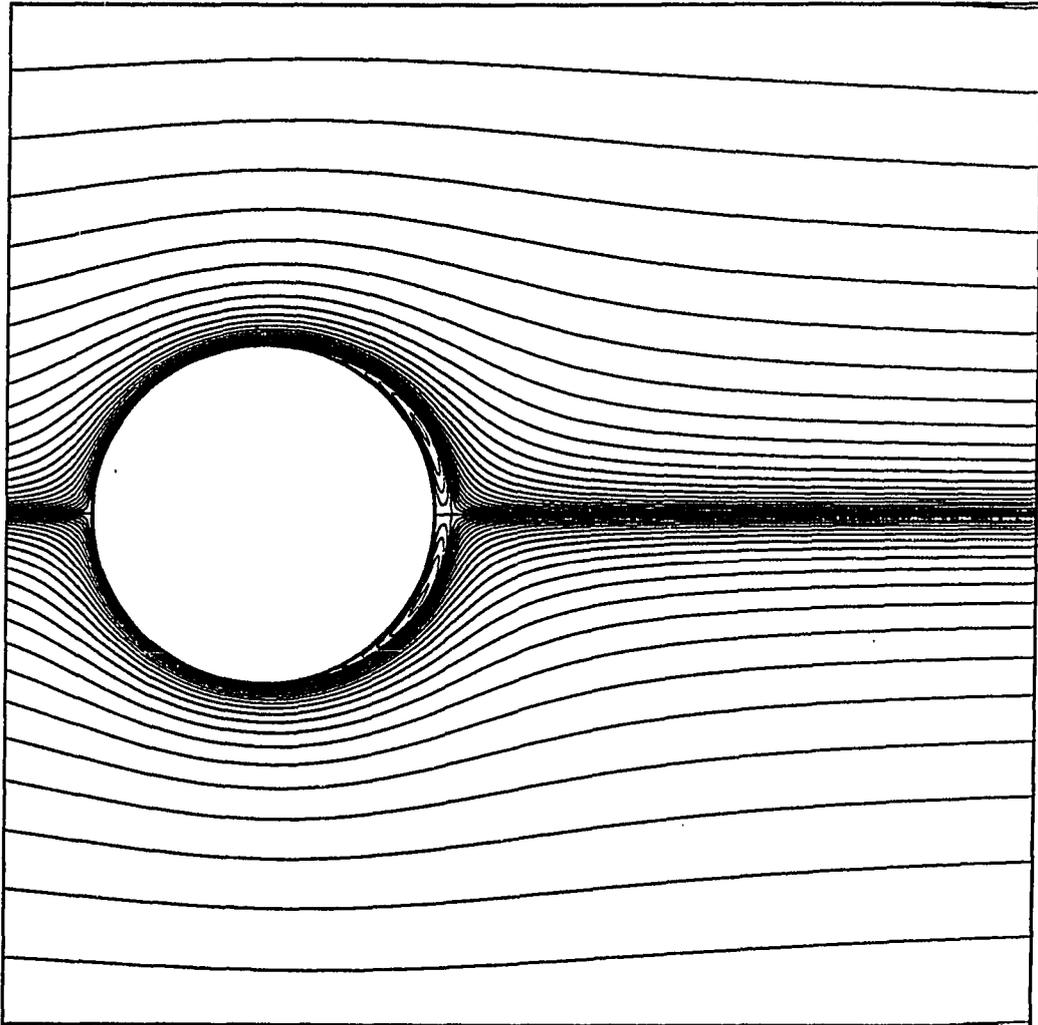
AFLBI Impulsively Started Cylinder $Re=3000$ $V/D=0.5$ 

Figure 4.68: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=0.5$)

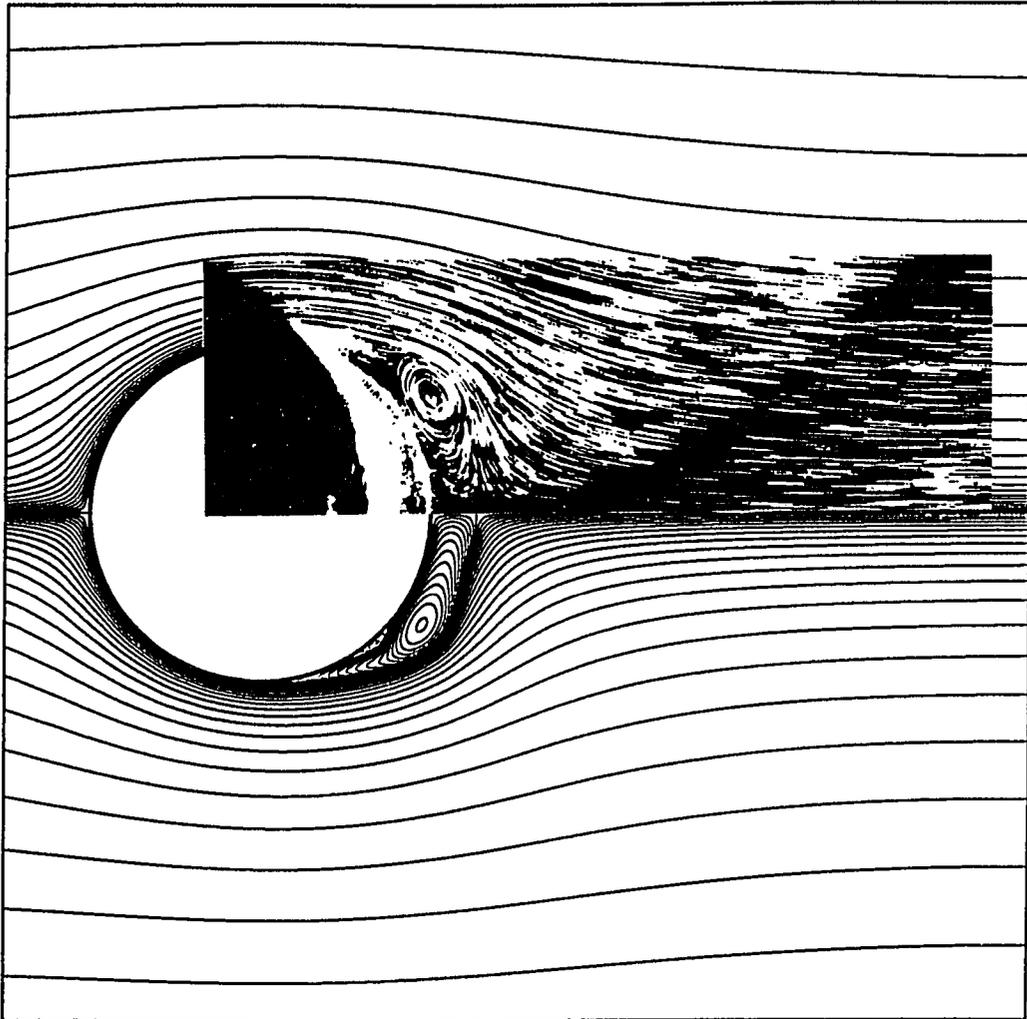
AFLBI Impulsively Started Cylinder $Re=3000$ $Vt/D=1.0$ 

Figure 4.69: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=1.0$)

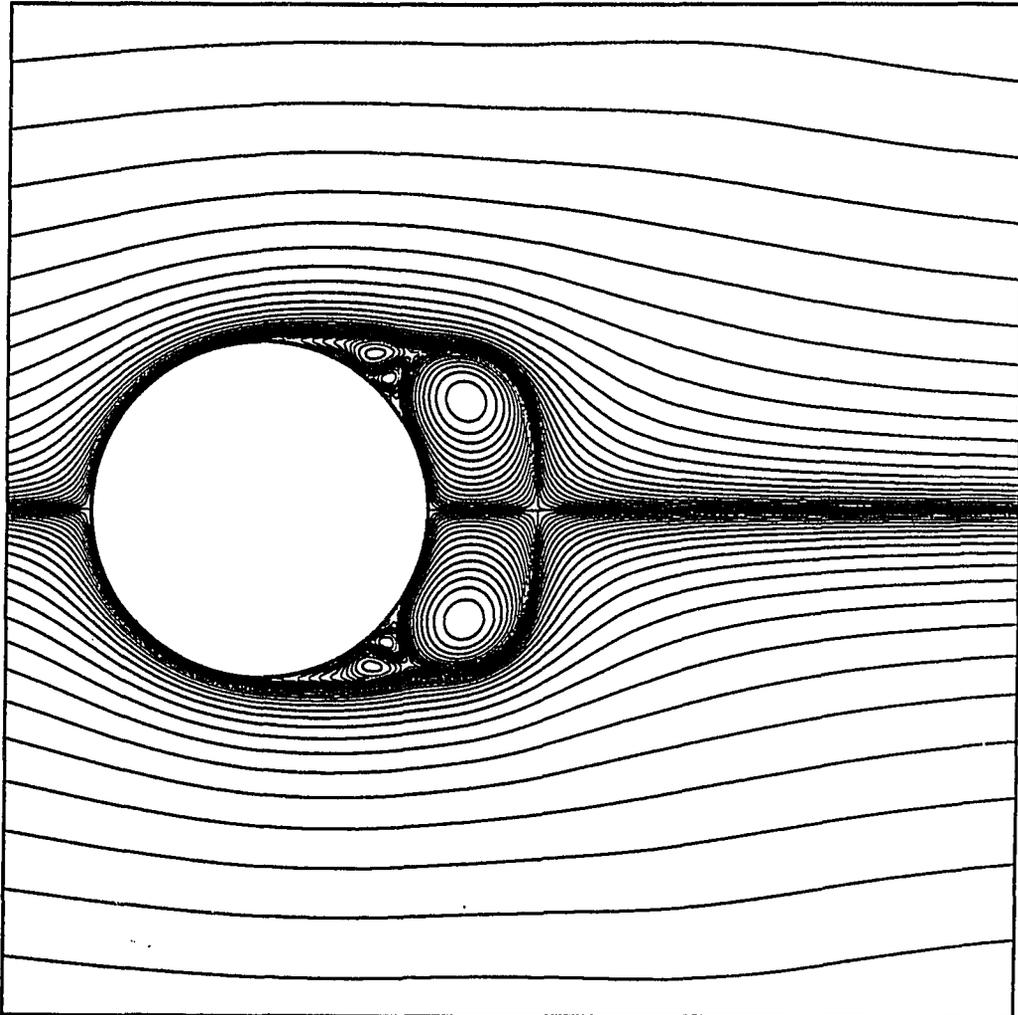
AFLBI Impulsively Started Cylinder $Re=3000$ $Vt/D=1.5$ 

Figure 4.70: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=1.5$)

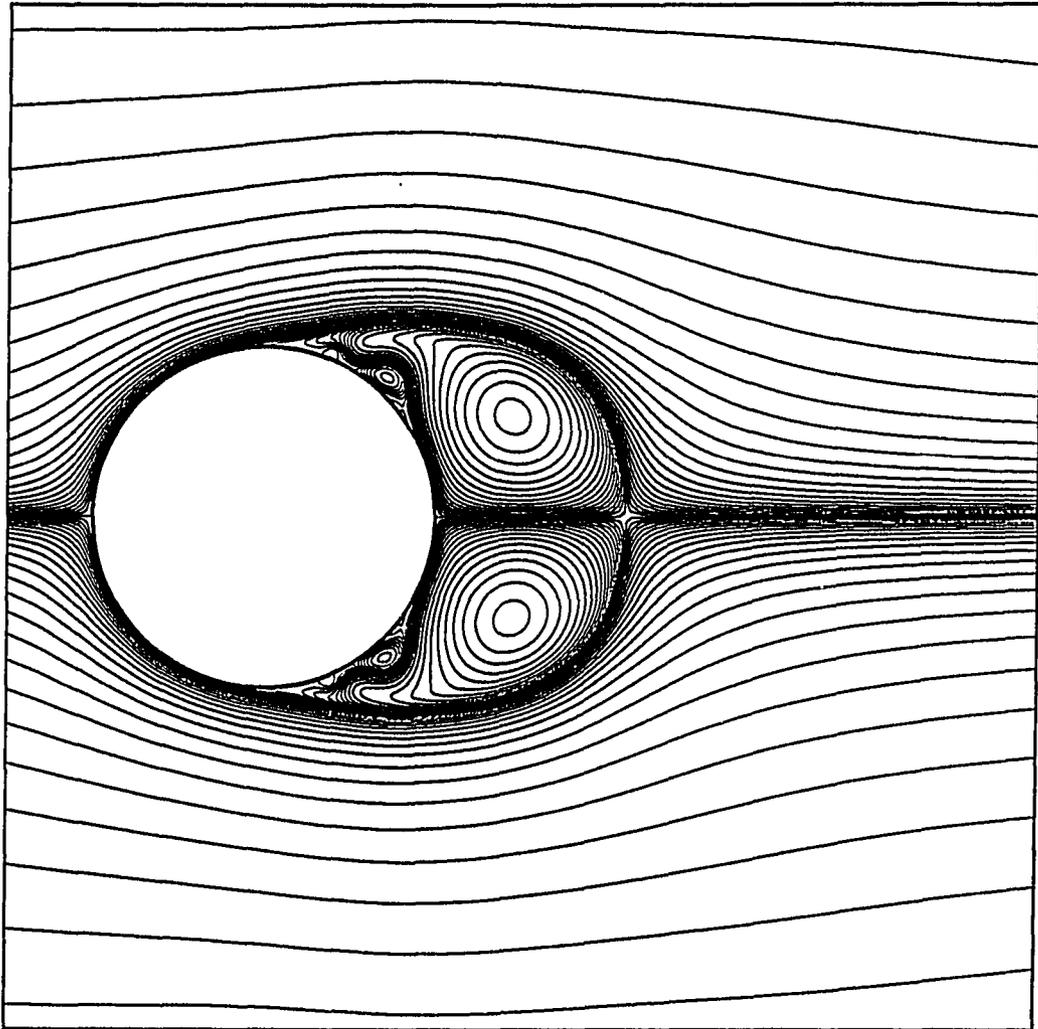
AFLBI Impulsively Started Cylinder $Re=3000$ $Vt/D=2.0$ 

Figure 4.71: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=2.0$)

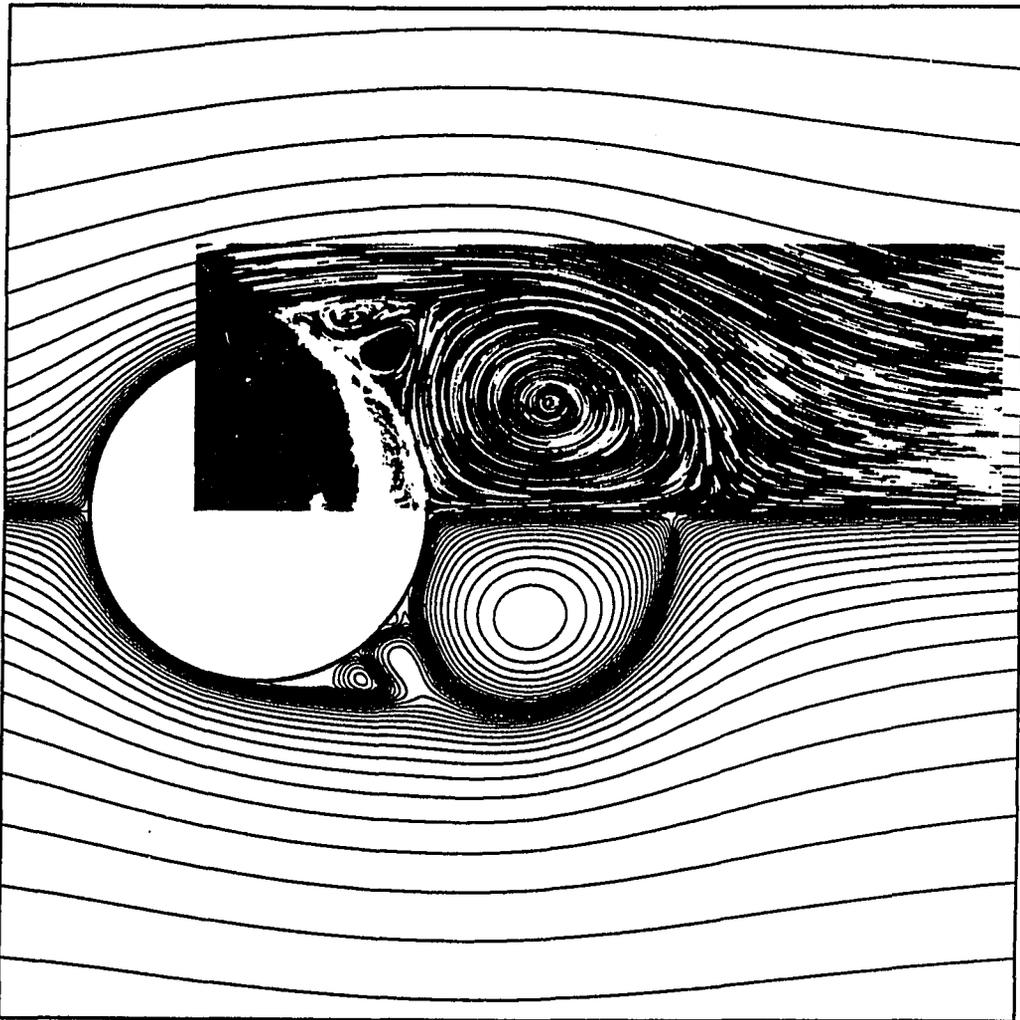
AFLBI Impulsively Started Cylinder $Re=3000$ $Vt/D=2.5$ 

Figure 4.72: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=2.5$)

GCGTVD Impulsively Started Cylinder $Re=3000$ $Vt/D=0.5$

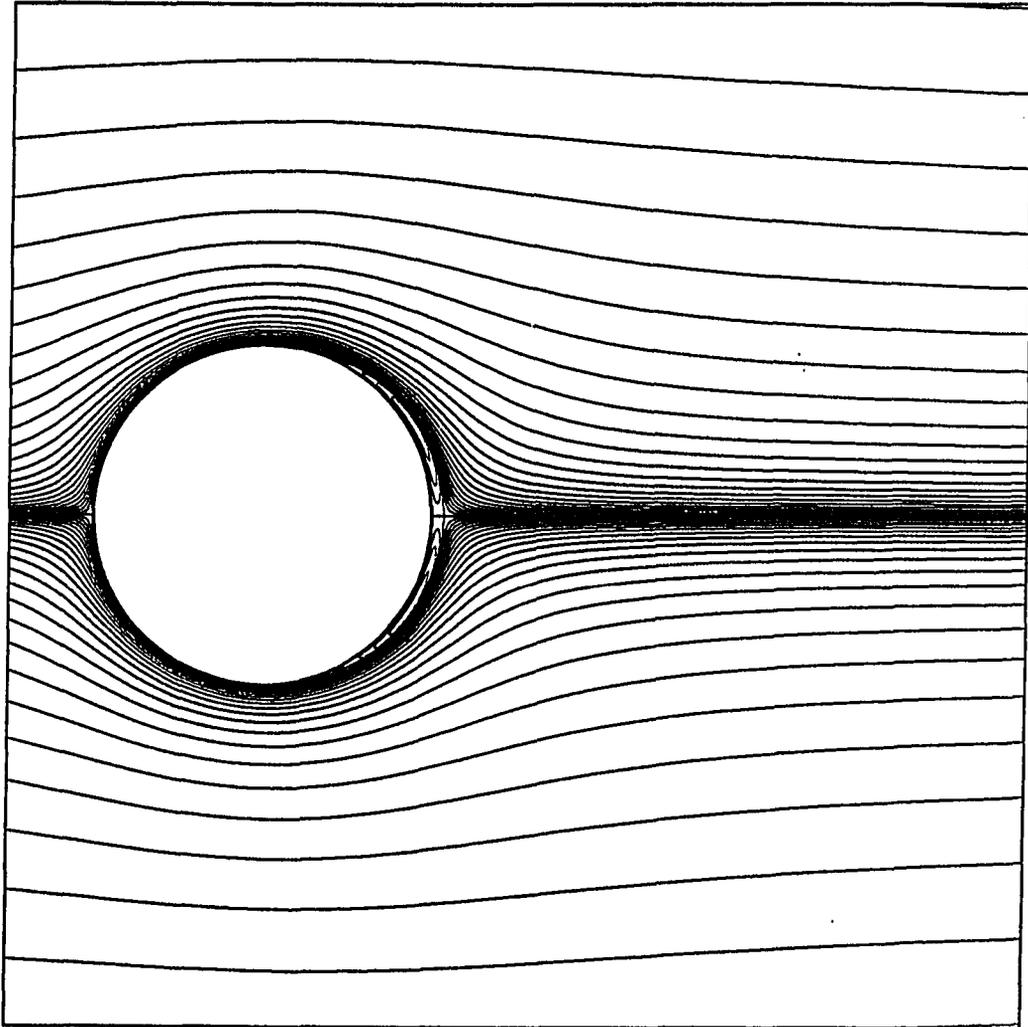


Figure 4.73: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=0.5$)

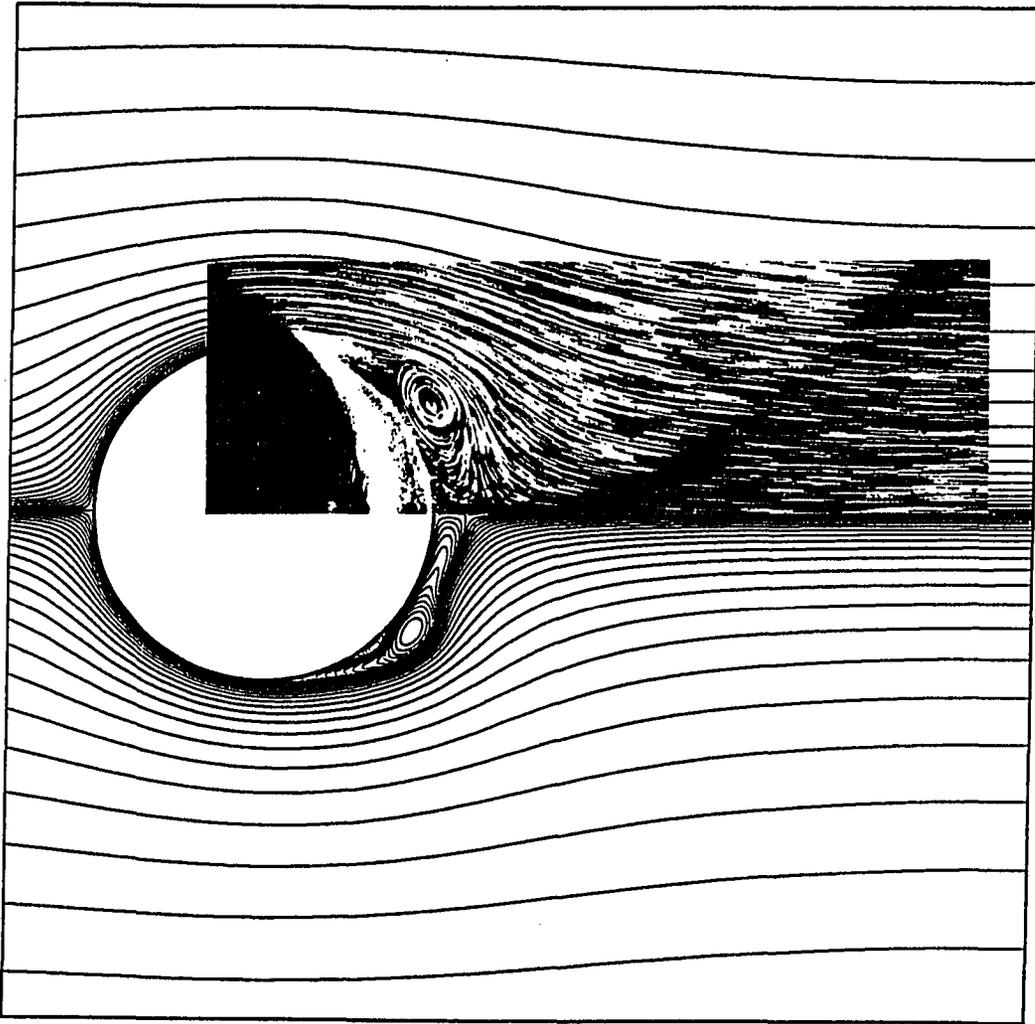
GCGTVD Impulsively Started Cylinder $Re=3000$ $Vt/D=1.0$ 

Figure 4.74: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=1.0$)

GCGTVD Impulsively Started Cylinder $Re=3000$ $Vt/D=1.5$

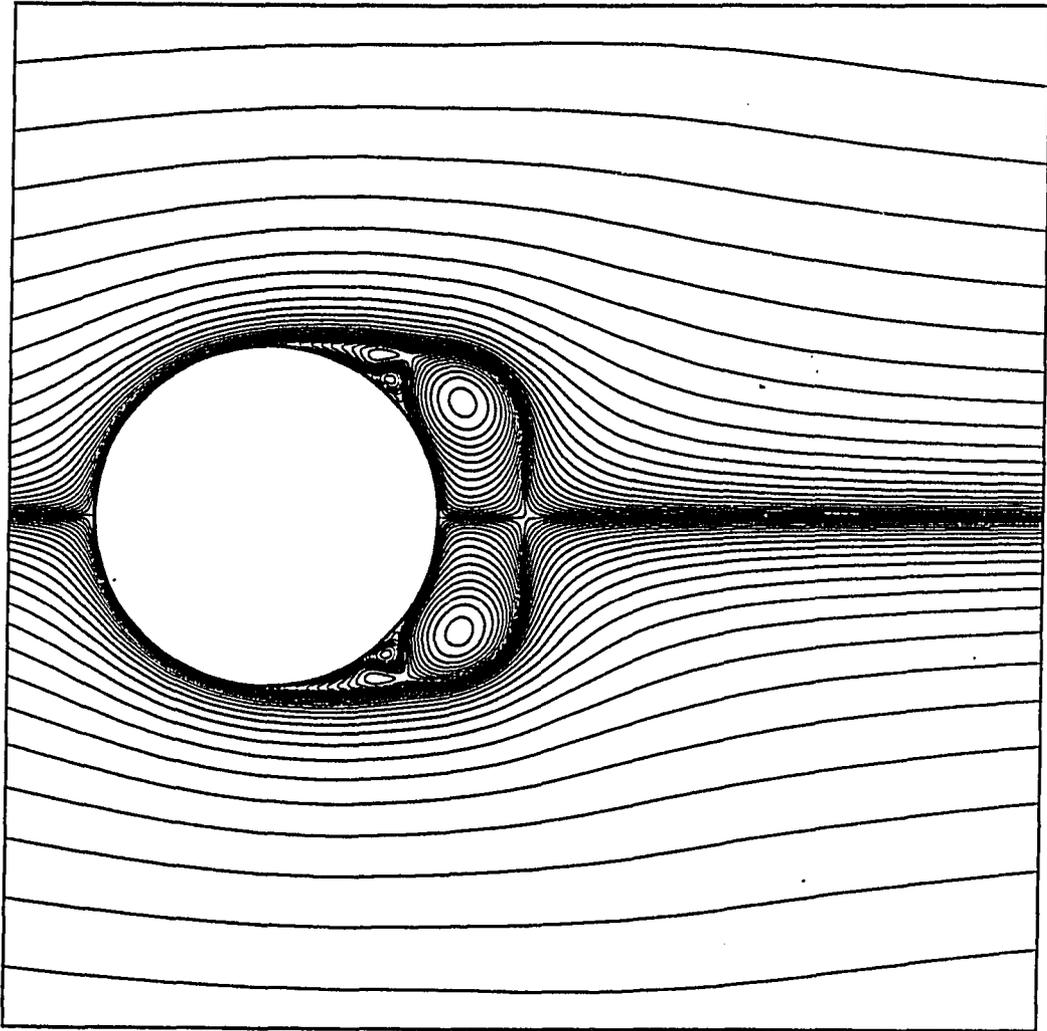


Figure 4.75: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=1.5$)

GCGTVD Impulsively Started Cylinder $Re=3000$ $Vt/D=2.0$

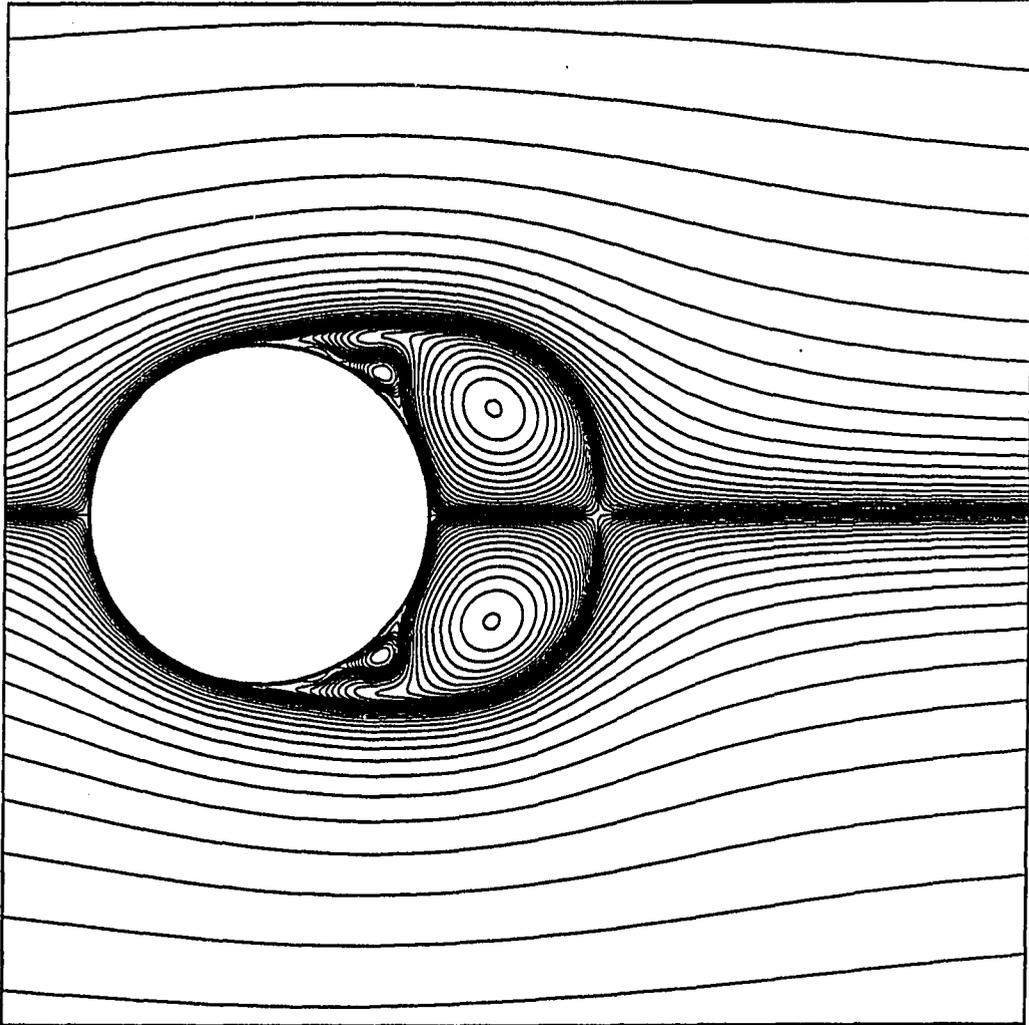


Figure 4.76: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=2.0$)

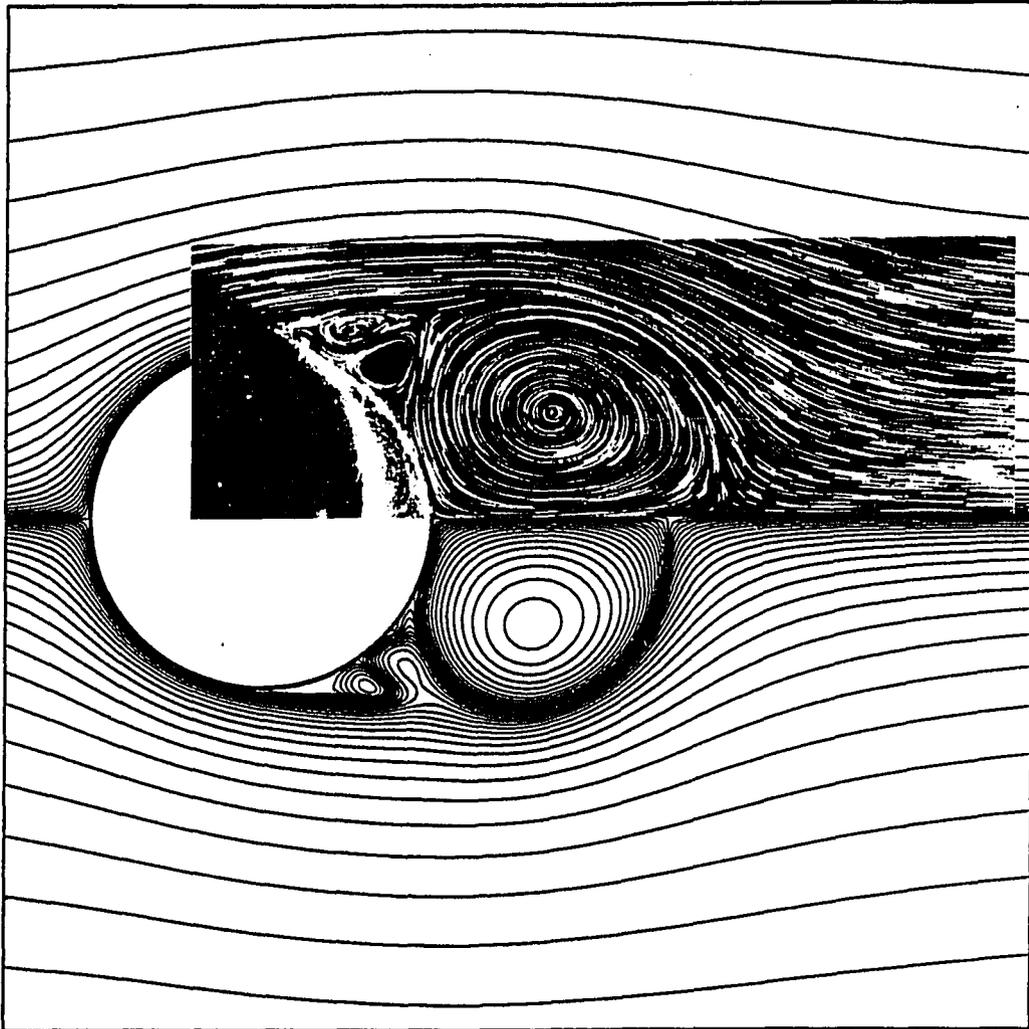
GCGTVD Impulsively Started Cylinder $Re=3000$ $Vt/D=2.5$ 

Figure 4.77: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=3000$, $t=2.5$)

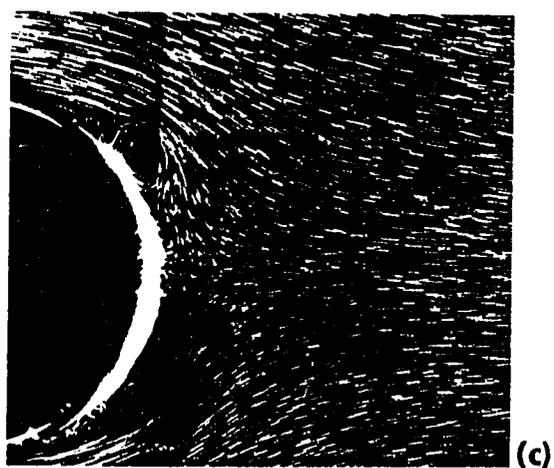
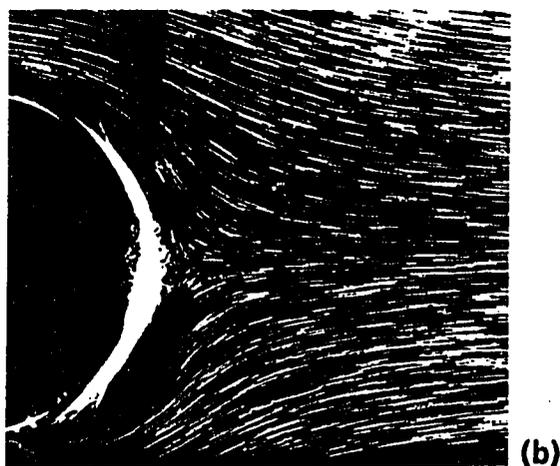
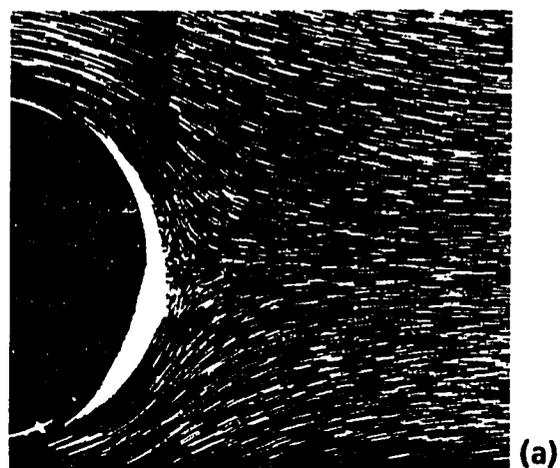


Figure 4.78: Experimental flow visualization for impulsively started cylinder ($Re=9500$), A: $Vt/D=0.6$, B: $Vt/D=0.8$, C: $Vt/D=1.0$

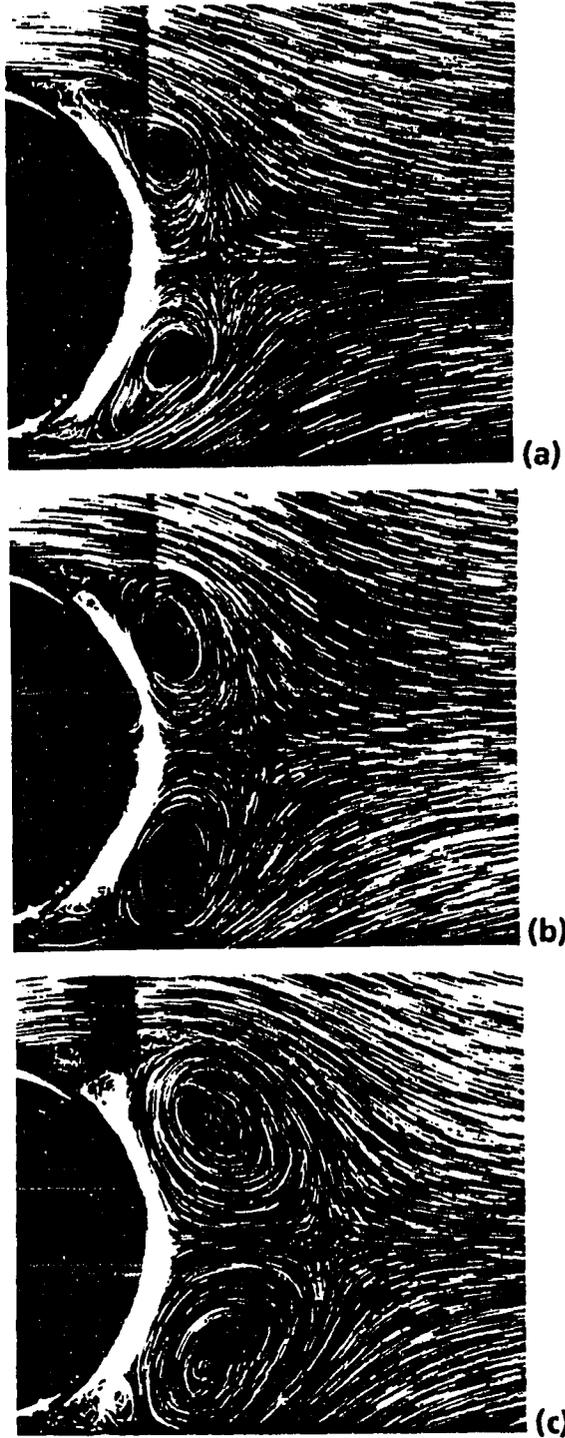


Figure 4.79: Experimental flow visualization for impulsively started cylinder ($Re=9500$), A: $Vt/D=1.4$, B: $Vt/D=1.6$, C: $Vt/D=2.0$

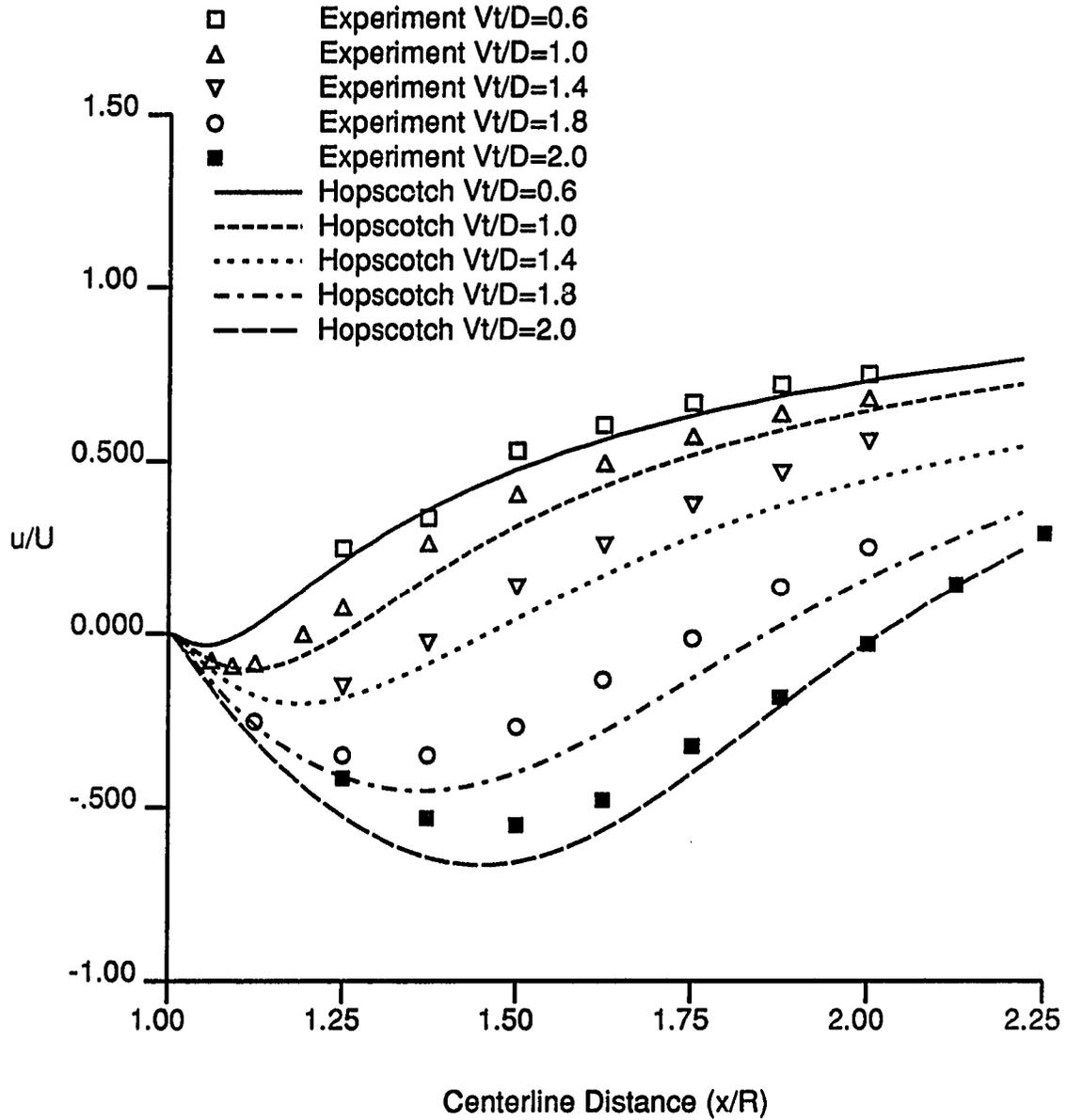
Hopscotch Impulsively Started Cylinder $Re=9500$ 

Figure 4.80: Comparison of predicted hopscotch and experimental unsteady velocity profiles for impulsively started cylinder flow ($Re=9500$)

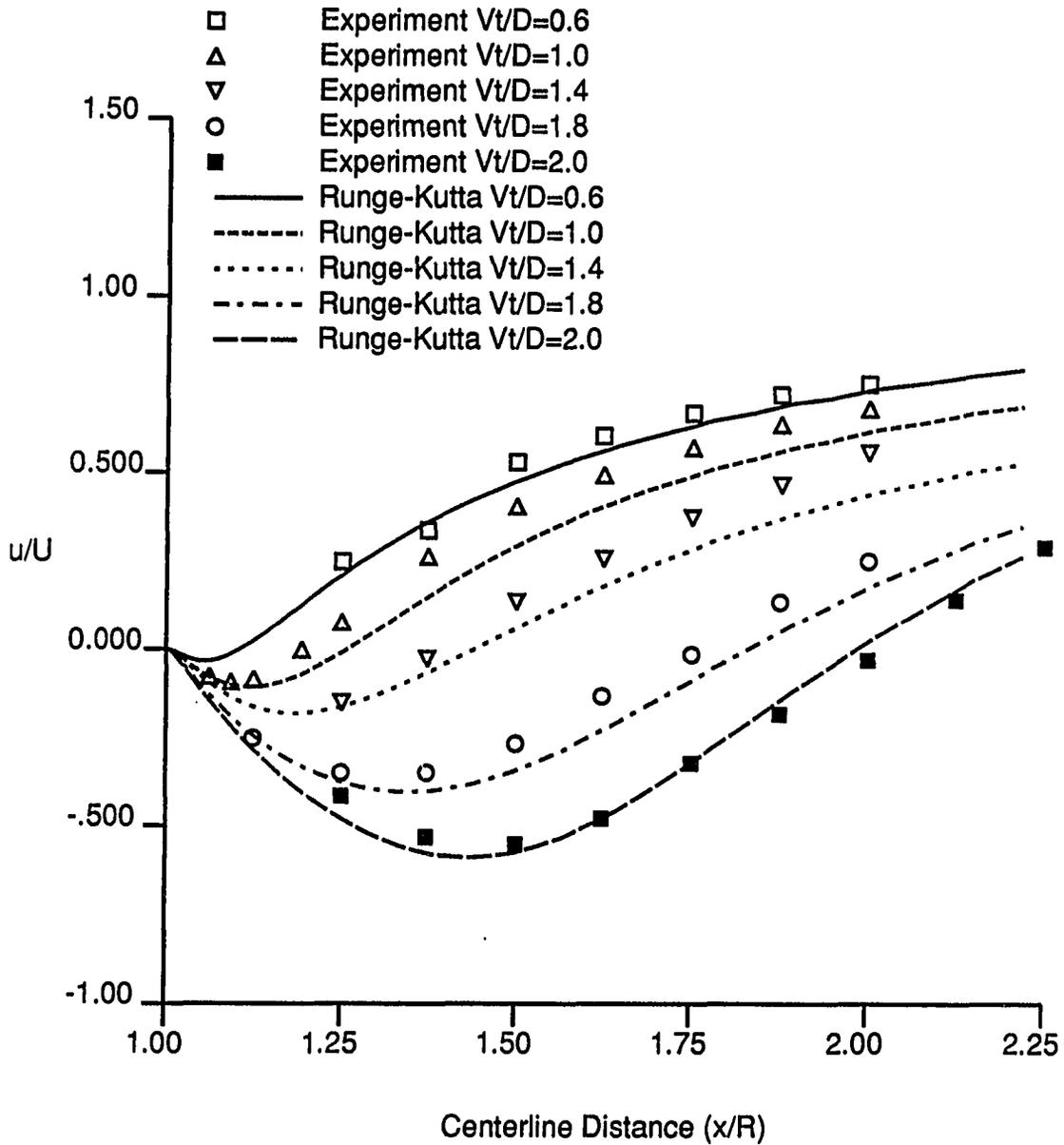
Runge-Kutta Impulsively Started Cylinder $Re=9500$ 

Figure 4.81: Comparison of predicted Runge-Kutta and experimental unsteady velocity profiles for impulsively started cylinder flow ($Re=9500$)

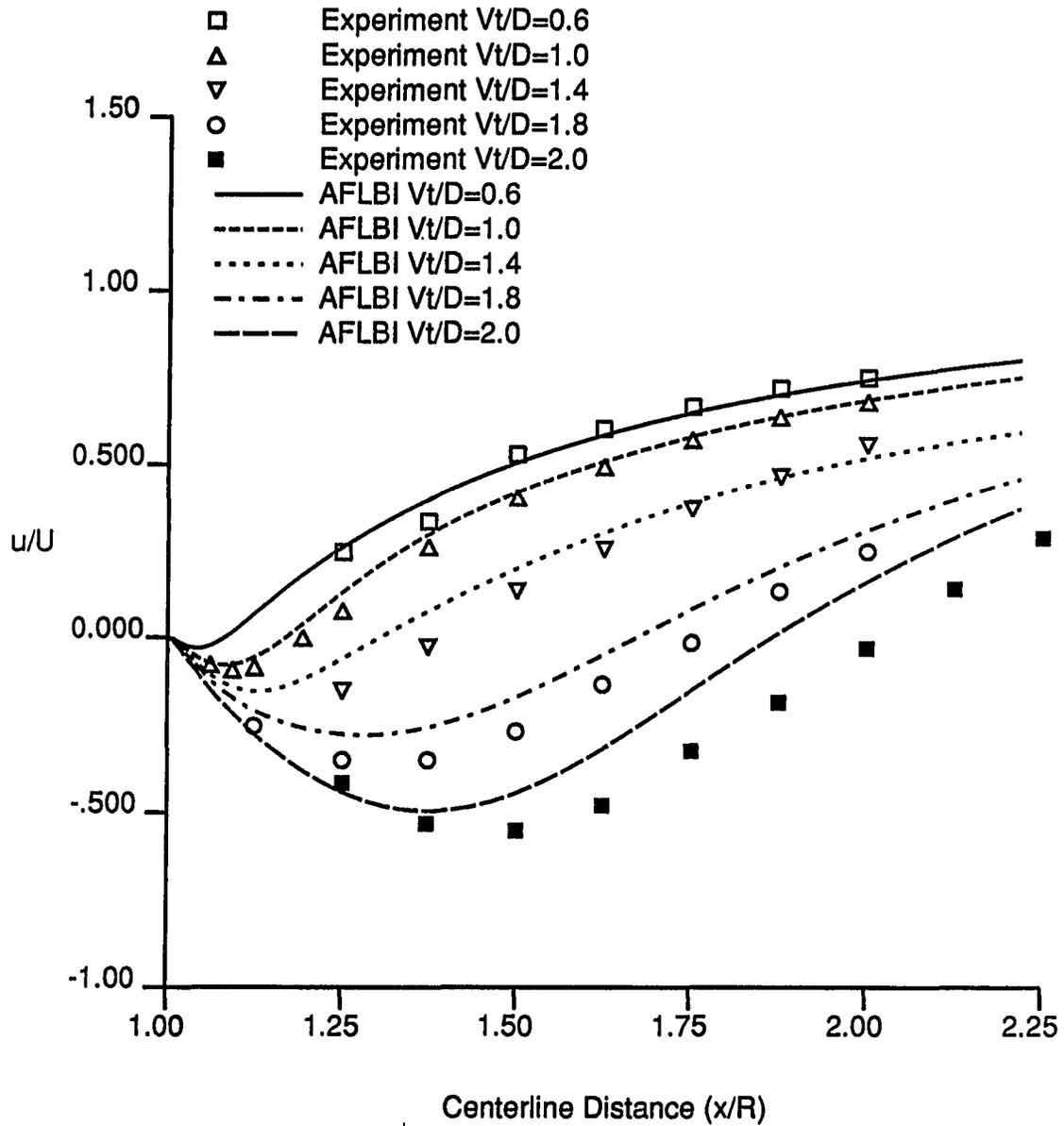
AFLBI Impulsively Started Cylinder $Re=9500$ 

Figure 4.82: Comparison of predicted AFLBI and experimental unsteady velocity profiles for impulsively started cylinder flow ($Re=9500$)

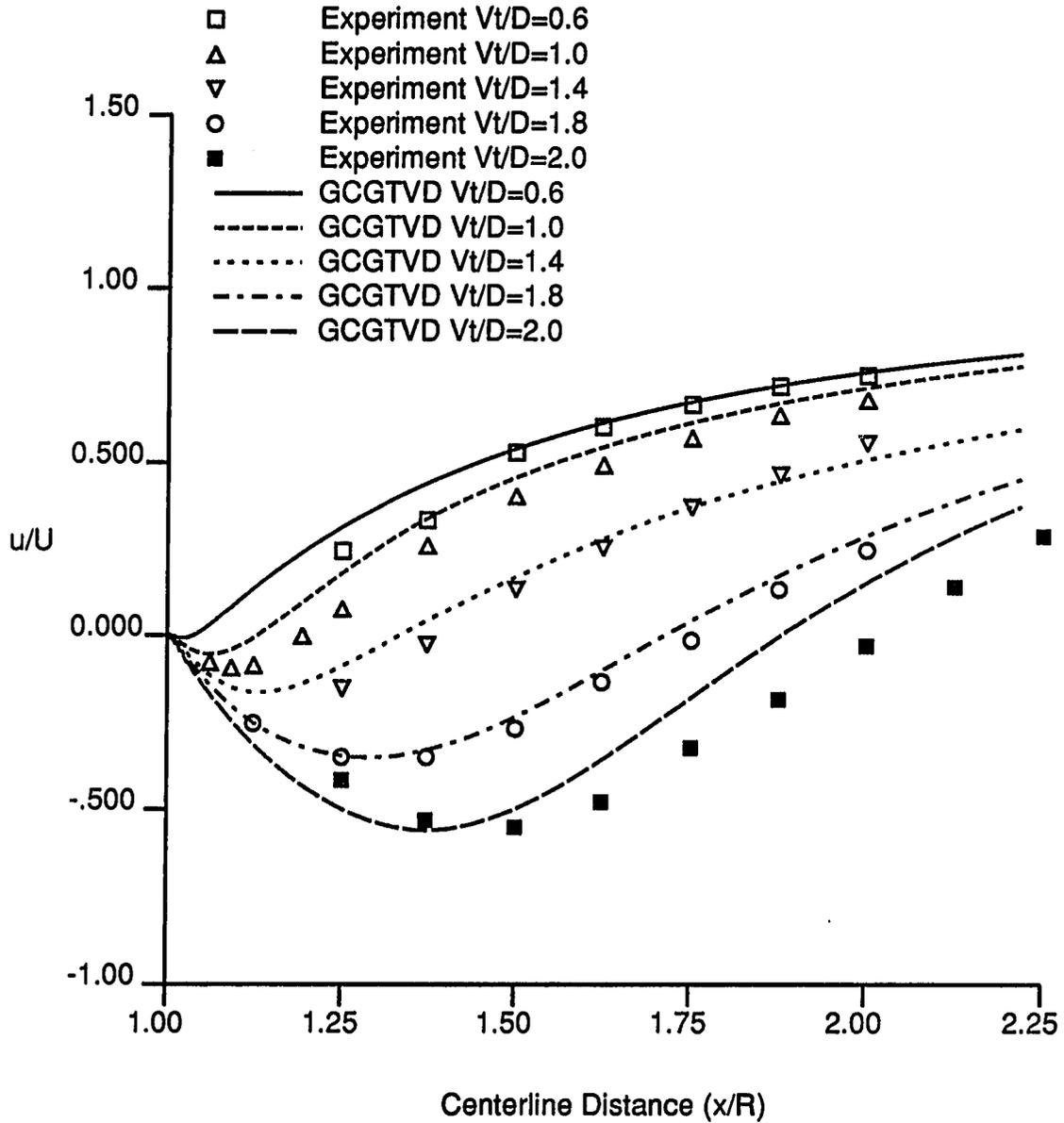
GCGTVD Impulsively Started Cylinder $Re=9500$ 

Figure 4.83: Comparison of predicted GCGTVD and experimental unsteady velocity profiles for impulsively started cylinder flow ($Re=9500$)

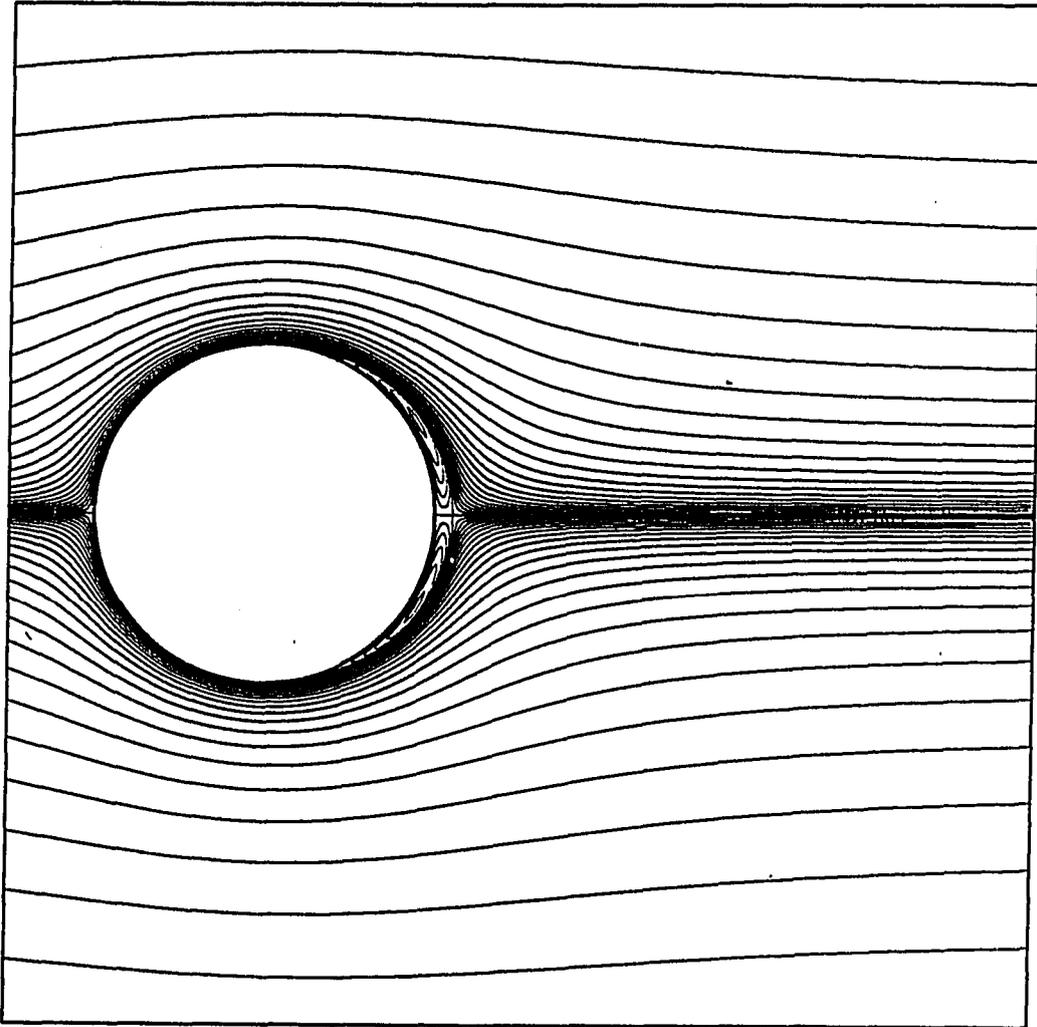
Hopscotch Impulsively Started Cylinder $Re=9500$ $Vt/D=0.6$ 

Figure 4.84: Hopscotch predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=0.6$)

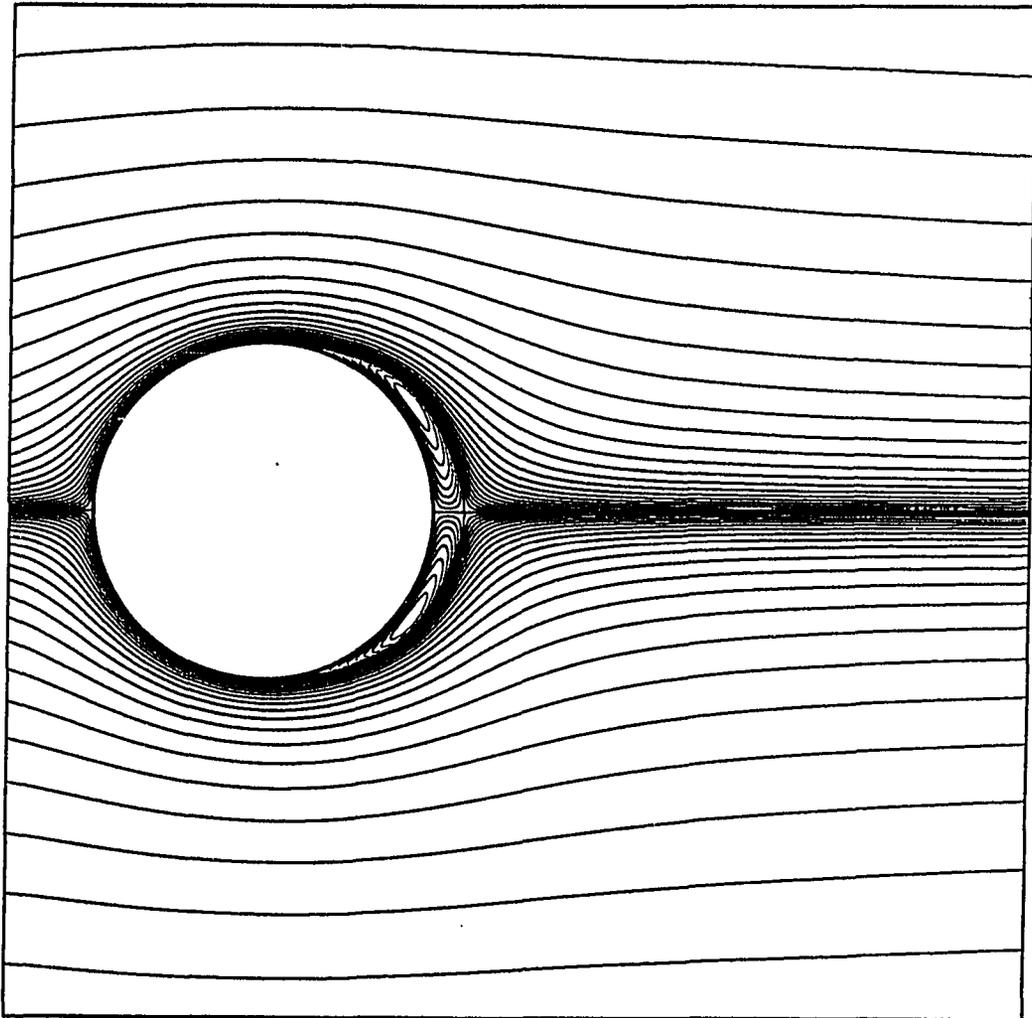
Hopscotch Impulsively Started Cylinder $Re=9500$ $Vt/D=0.8$ 

Figure 4.85: Hopscotch predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=0.8$)

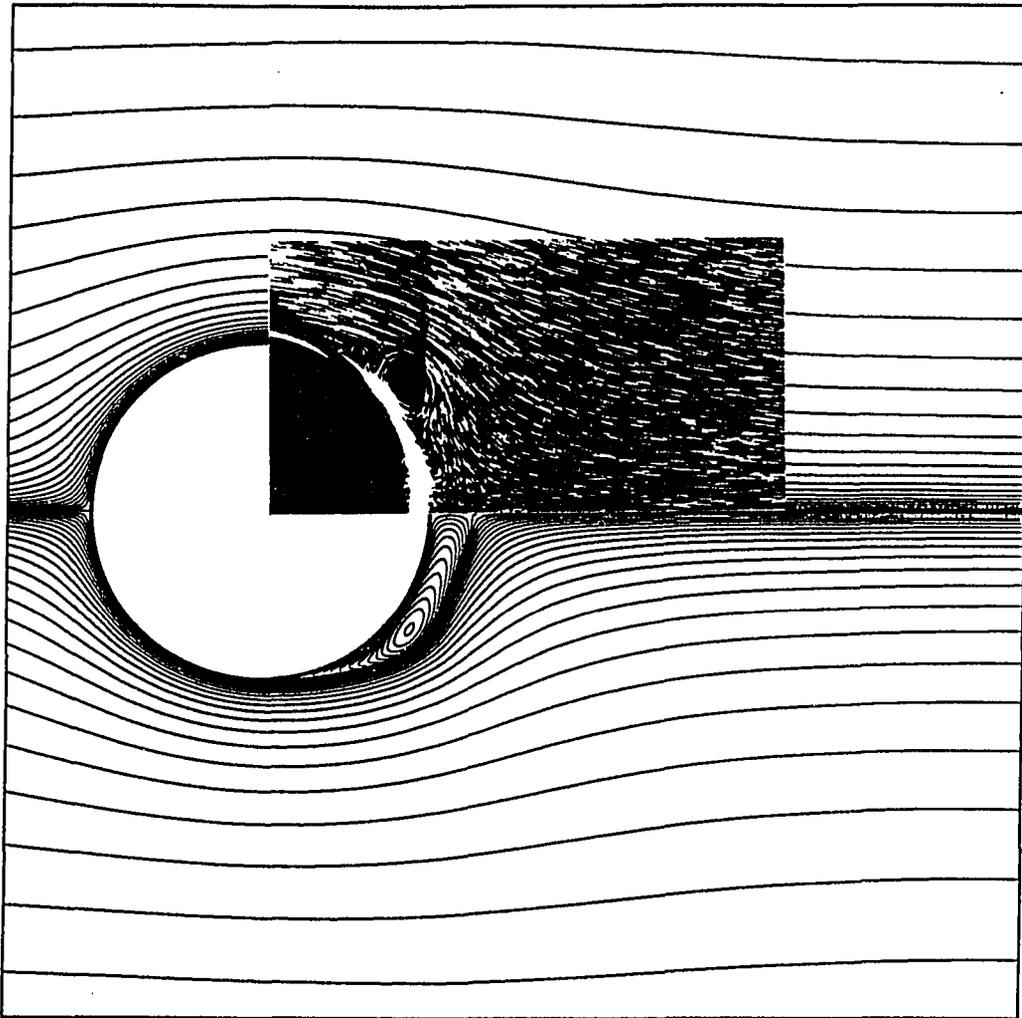
Hopscotch Impulsively Started Cylinder $Re=9500$ $Vt/D=1.0$ 

Figure 4.86: Hopscotch predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=1.0$)

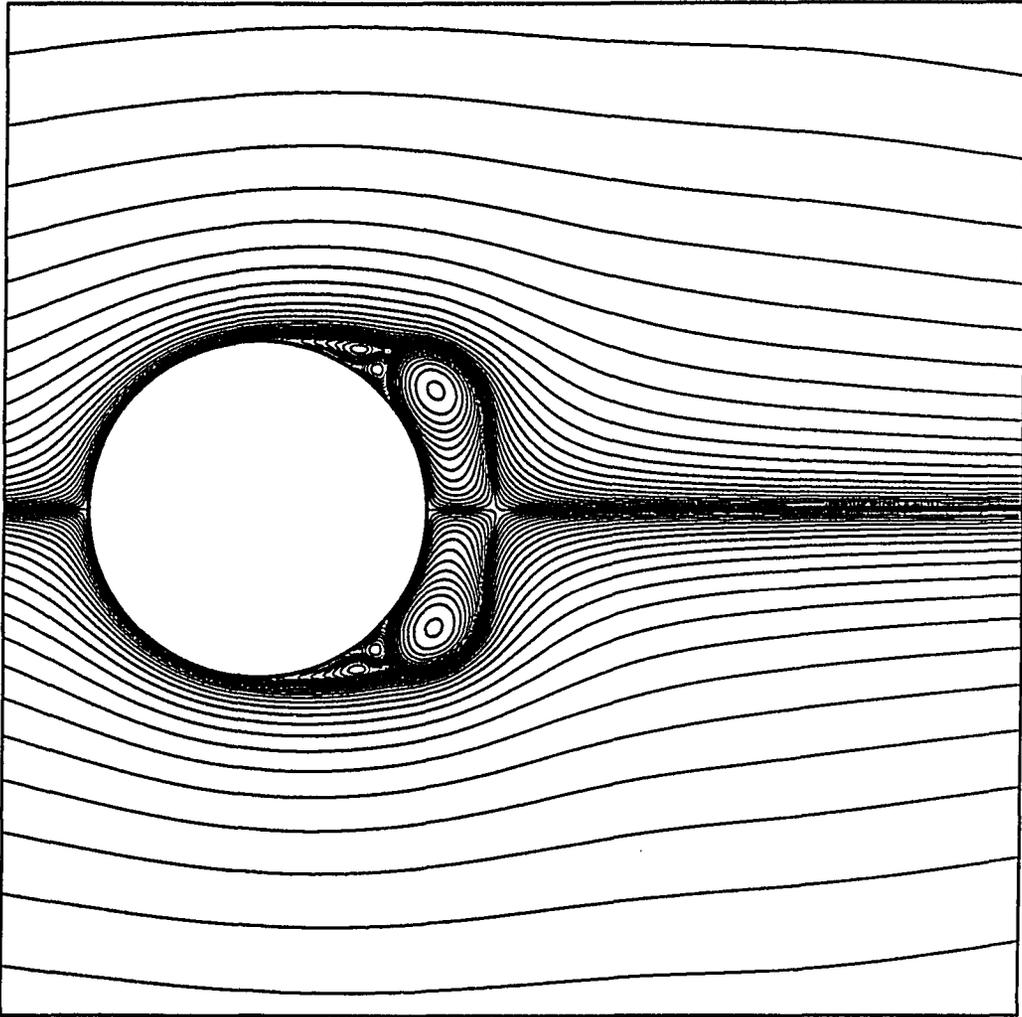
Hopscotch Impulsively Started Cylinder $Re=9500$ $Vt/D=1.4$ 

Figure 4.87: Hopscotch predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=1.4$)

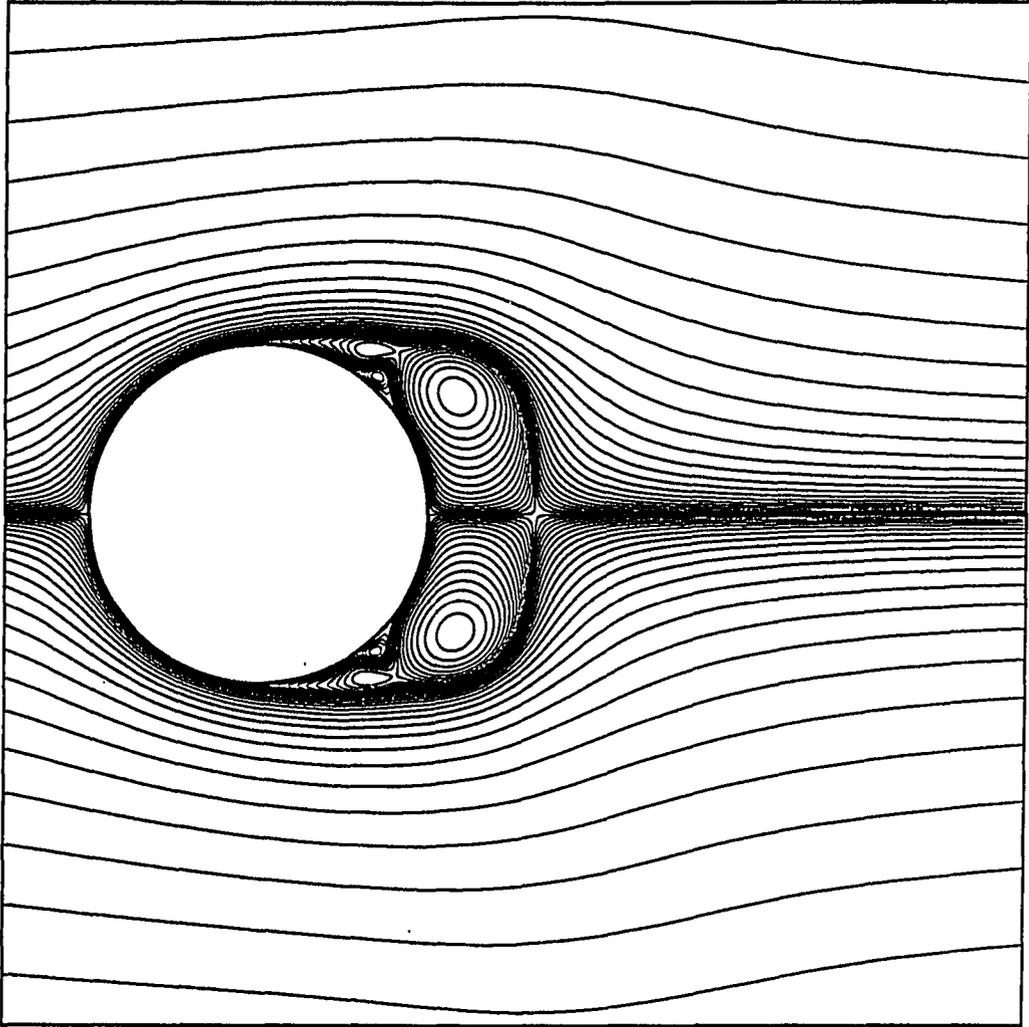
Hopscotch Impulsively Started Cylinder $Re=9500$ $Vt/D=1.6$ 

Figure 4.88: Hopscotch predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=1.6$)

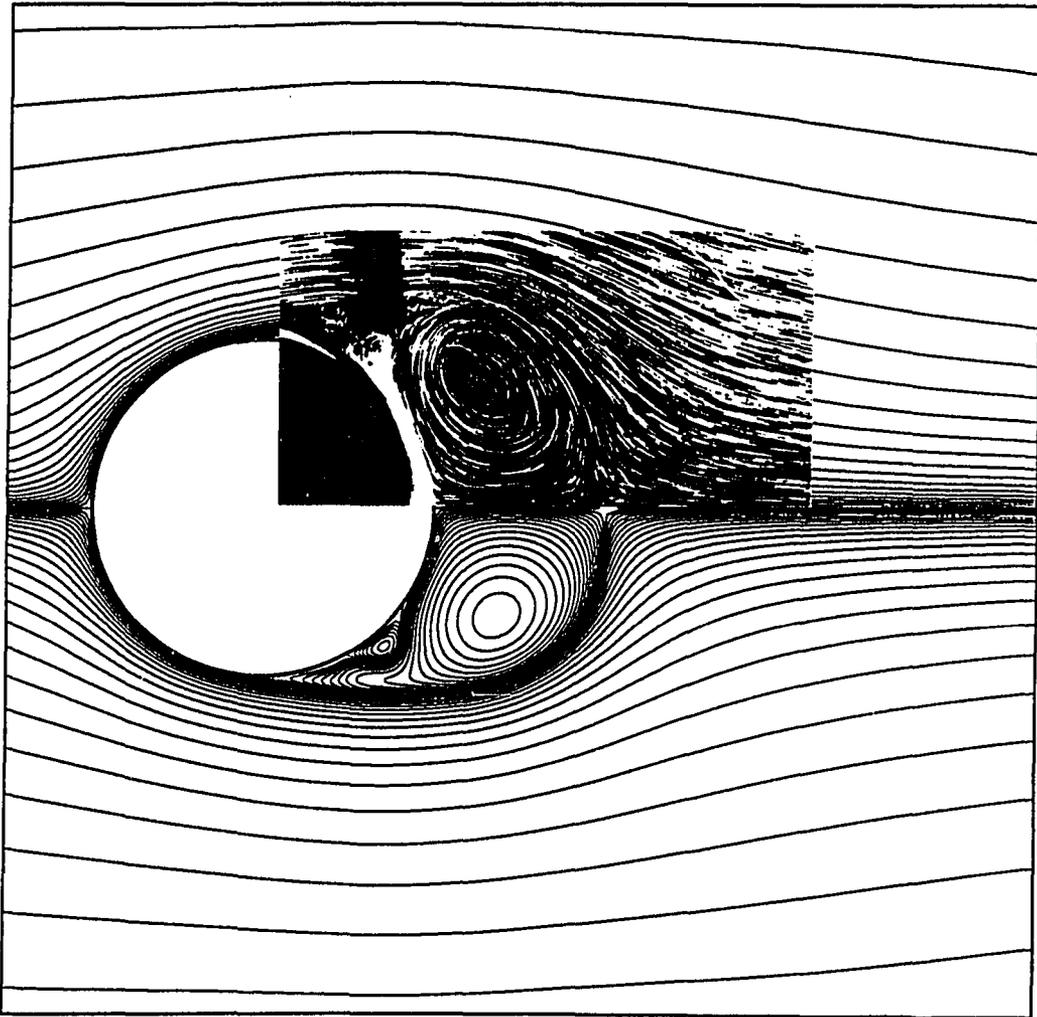
Hopscotch Impulsively Started Cylinder $Re=9500$ $Vt/D=2.0$ 

Figure 4.89: Hopscotch predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=2.0$)

Runge-Kutta Impulsively Started Cylinder $Re=9500$ $Vt/D=0.6$

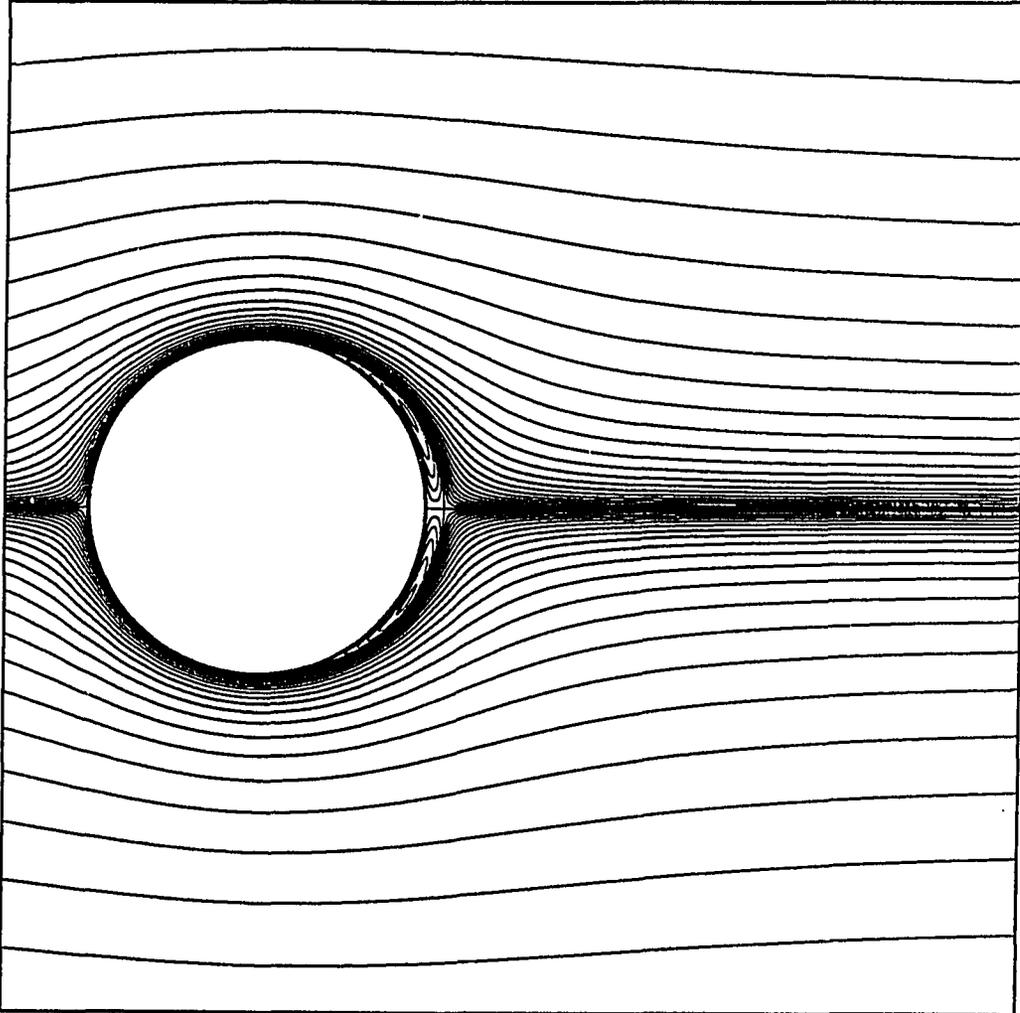


Figure 4.90: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=0.6$)

Runge-Kutta Impulsively Started Cylinder $Re=9500$ $Vt/D=0.8$

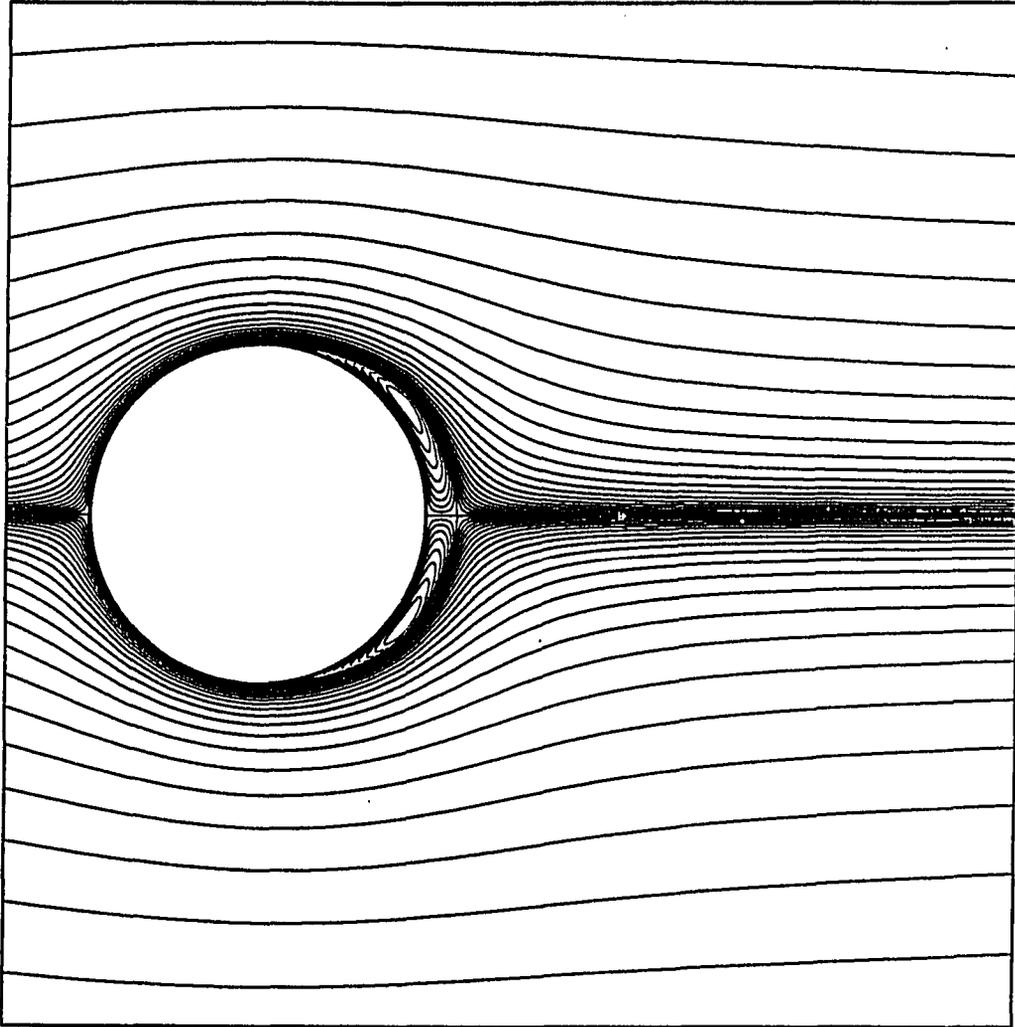


Figure 4.91: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=0.8$)

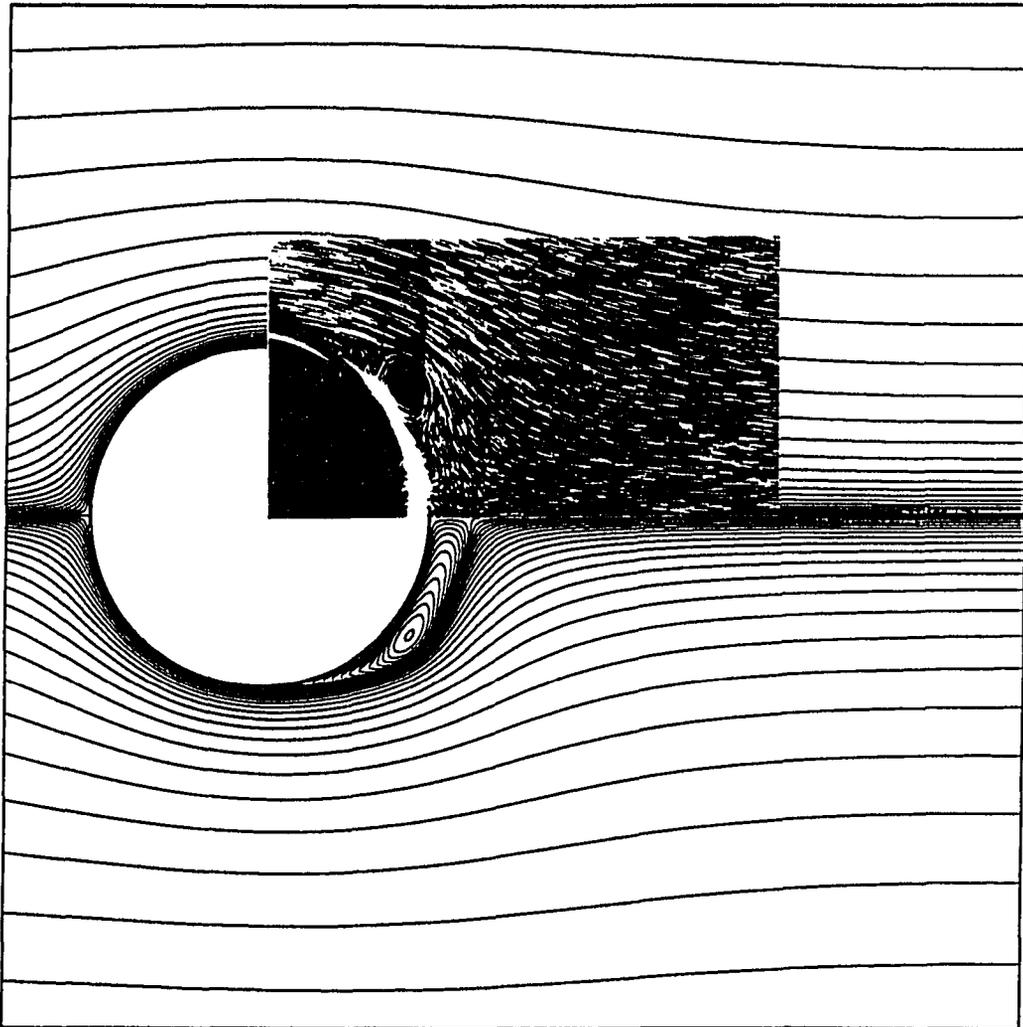
Runge-Kutta Impulsively Started Cylinder $Re=9500$ $Vt/D=1.0$ 

Figure 4.92: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=1.0$)

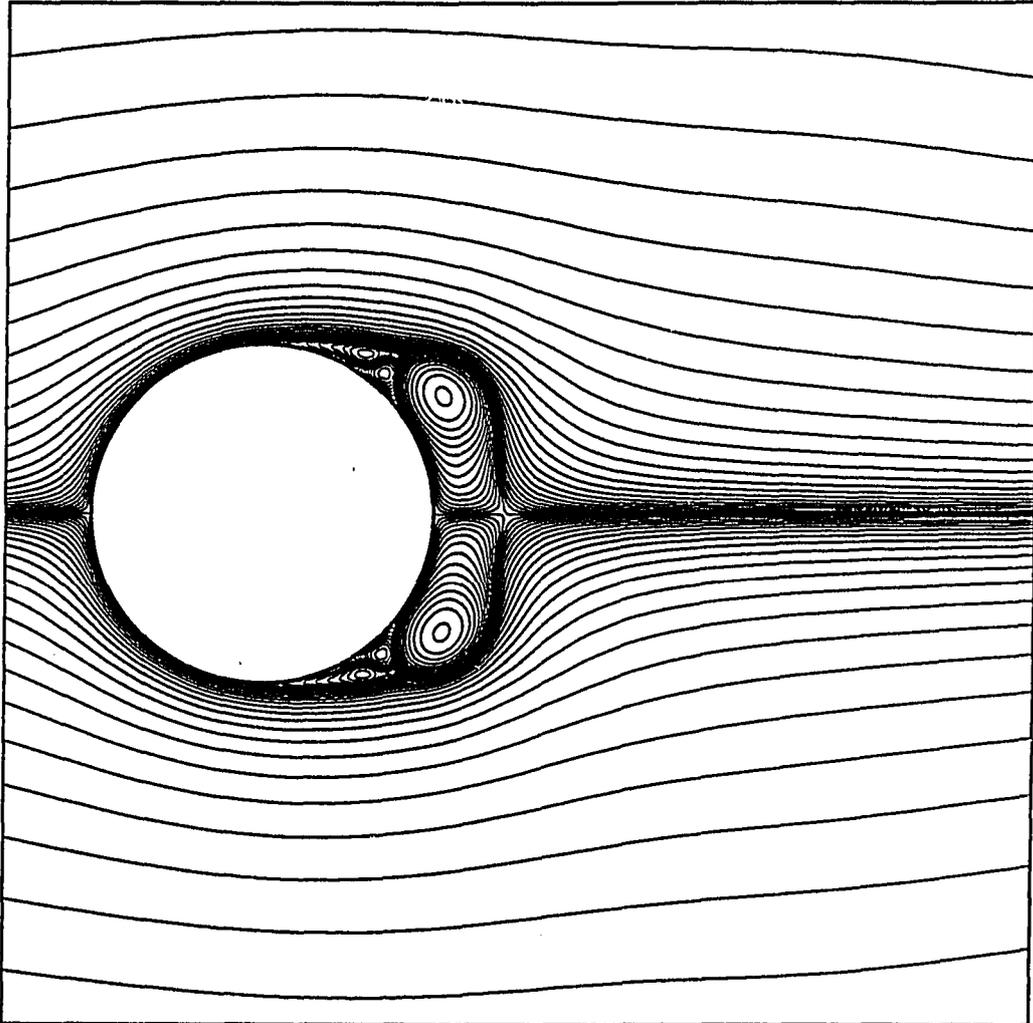
Runge-Kutta Impulsively Started Cylinder $Re=9500$ $Vt/D=1.4$ 

Figure 4.93: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=1.4$)

Runge-Kutta Impulsively Started Cylinder $Re=9500$ $Vt/D=1.6$

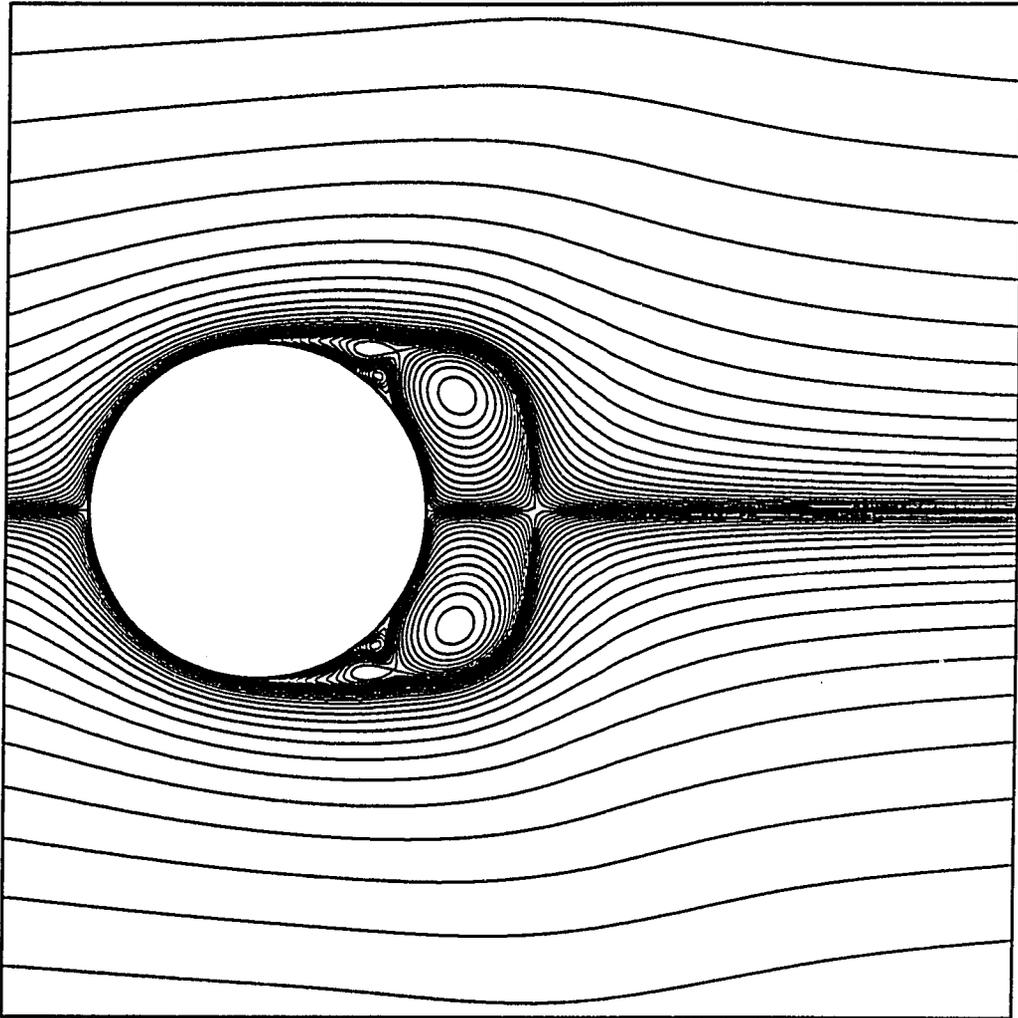


Figure 4.94: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=1.6$)

Runge-Kutta Impulsively Started Cylinder $Re=9500$ $Vt/D=2.0$

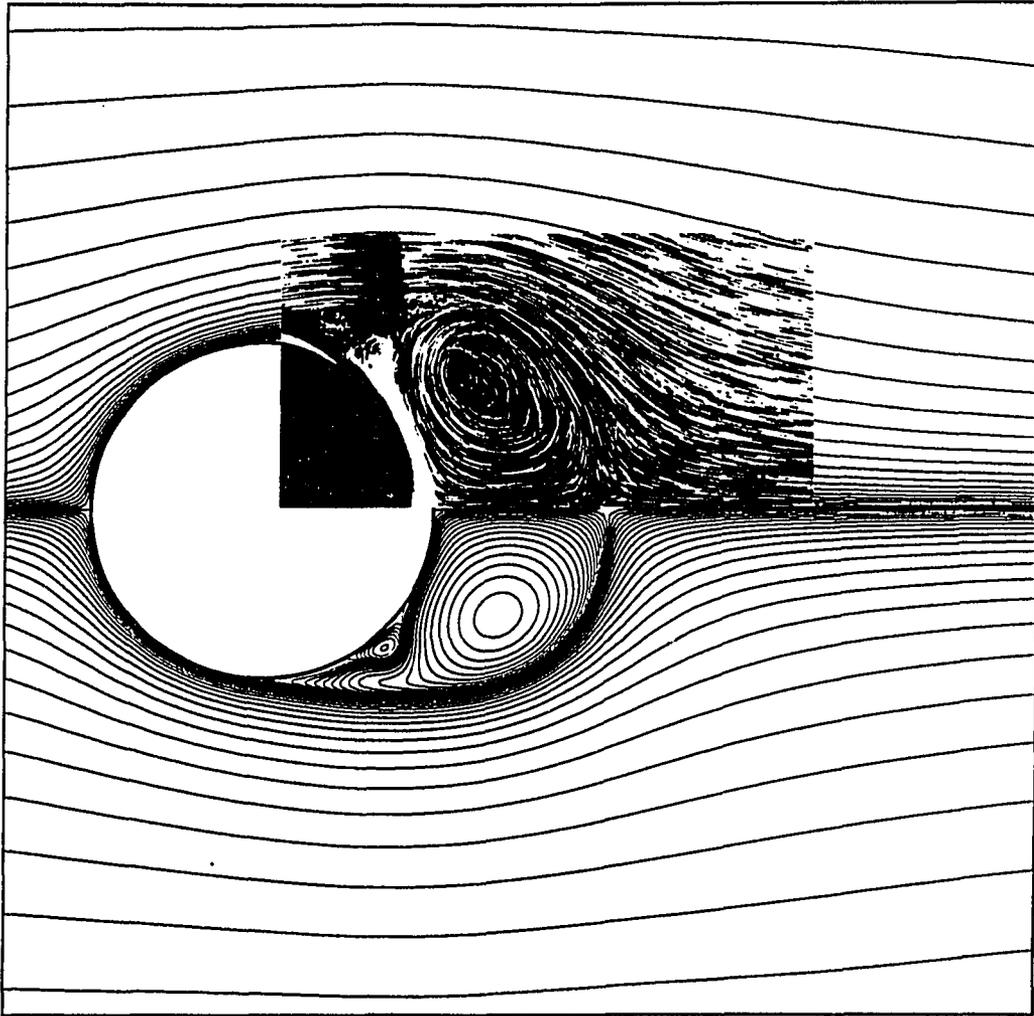


Figure 4.95: Runge-Kutta predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=2.0$)

AFLBI Impulsively Started Cylinder $Re=9500$ $V/D=0.6$

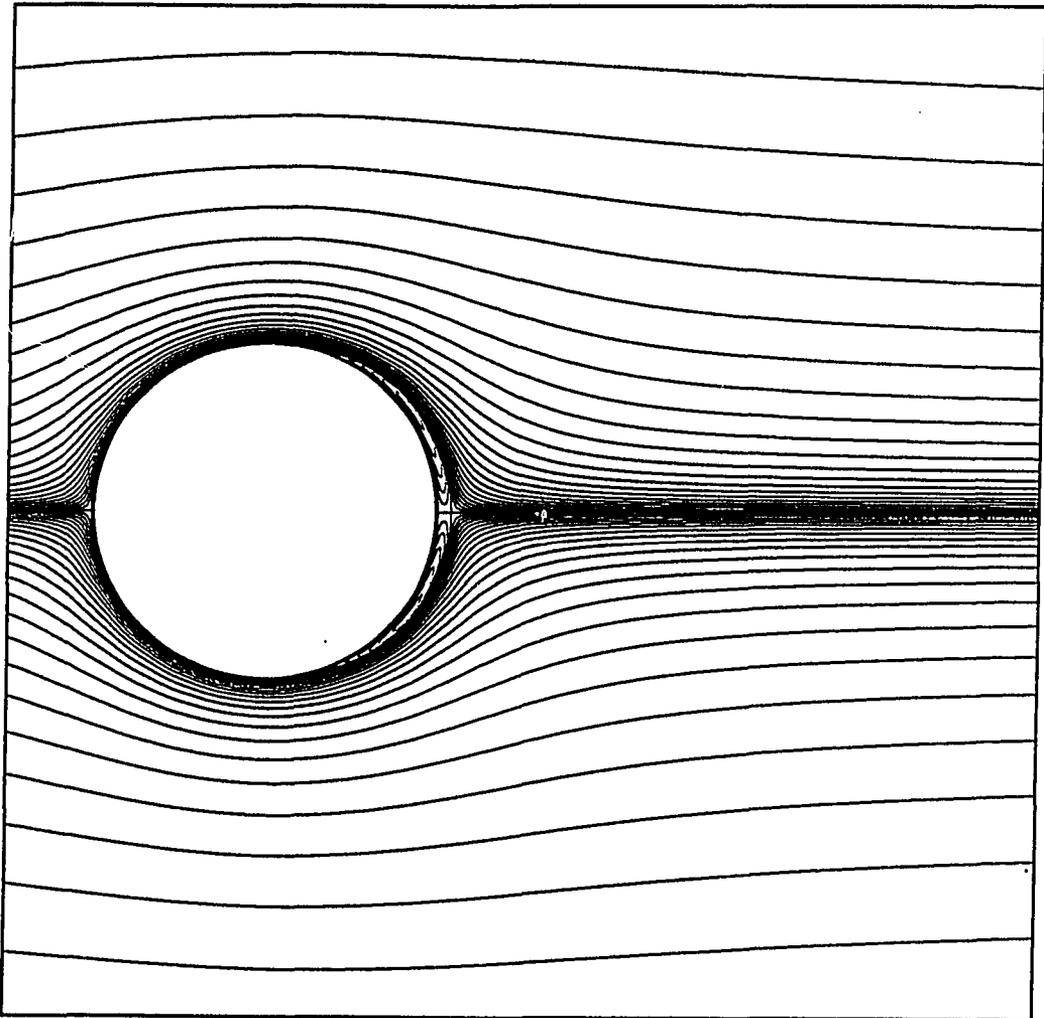


Figure 4.96: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=0.6$)

AFLBI Impulsively Started Cylinder $Re=9500$ $Vt/D=0.8$

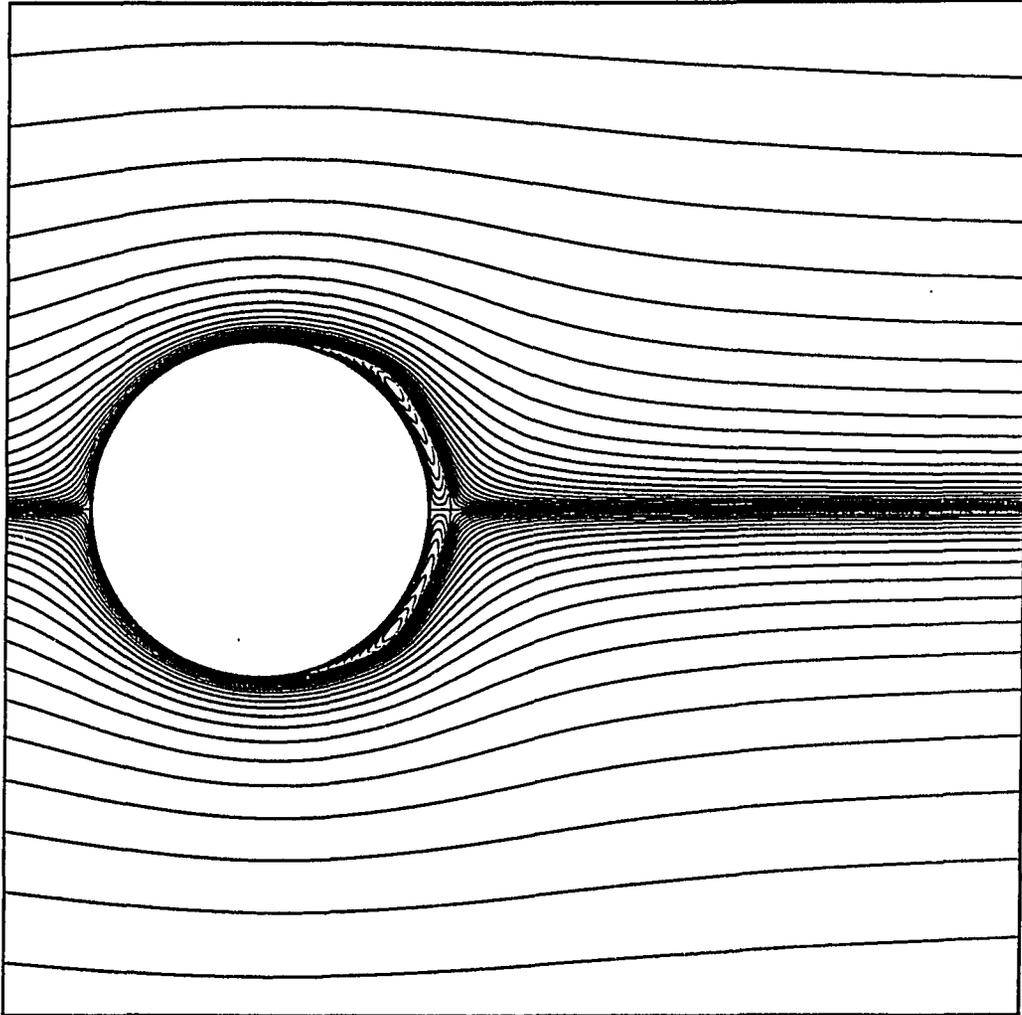


Figure 4.97: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=0.8$)

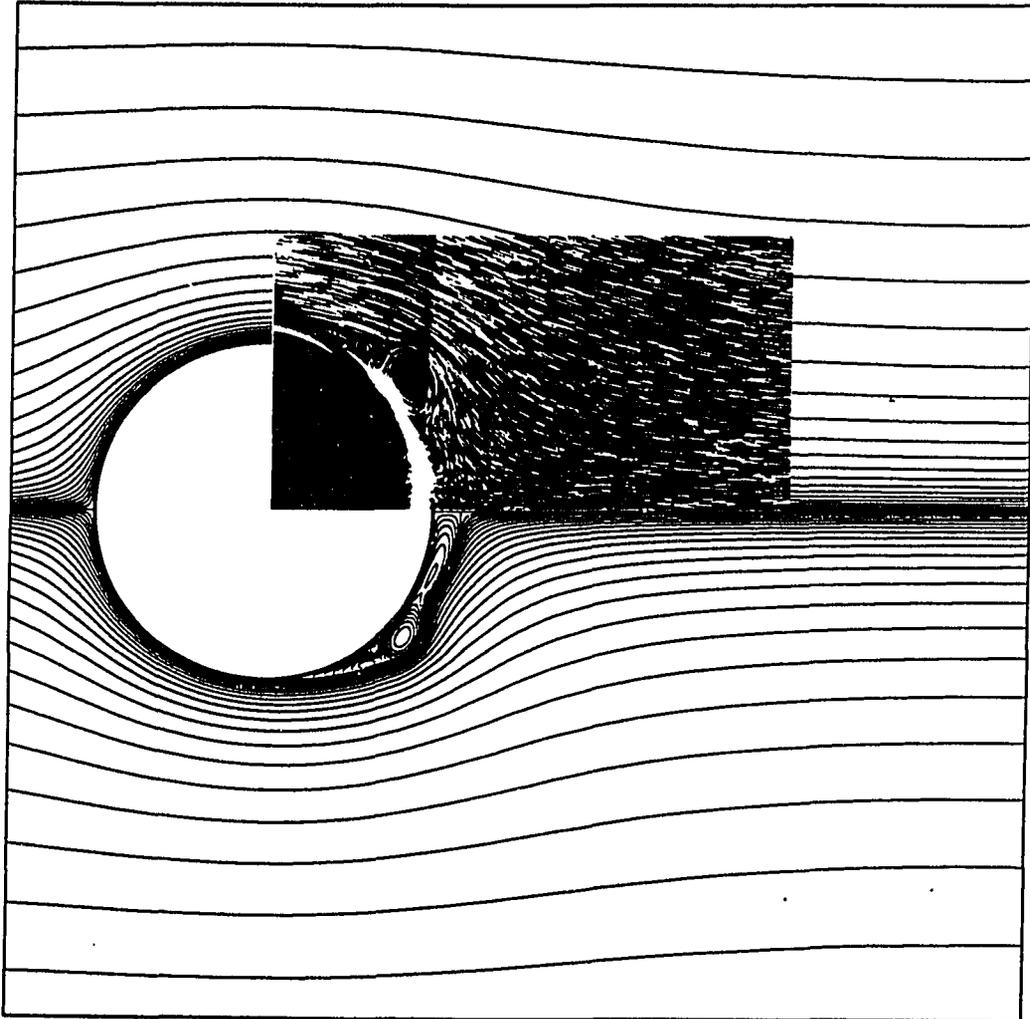
AFLBI Impulsively Started Cylinder $Re=9500$ $Vt/D=1.0$ 

Figure 4.98: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=1.0$)

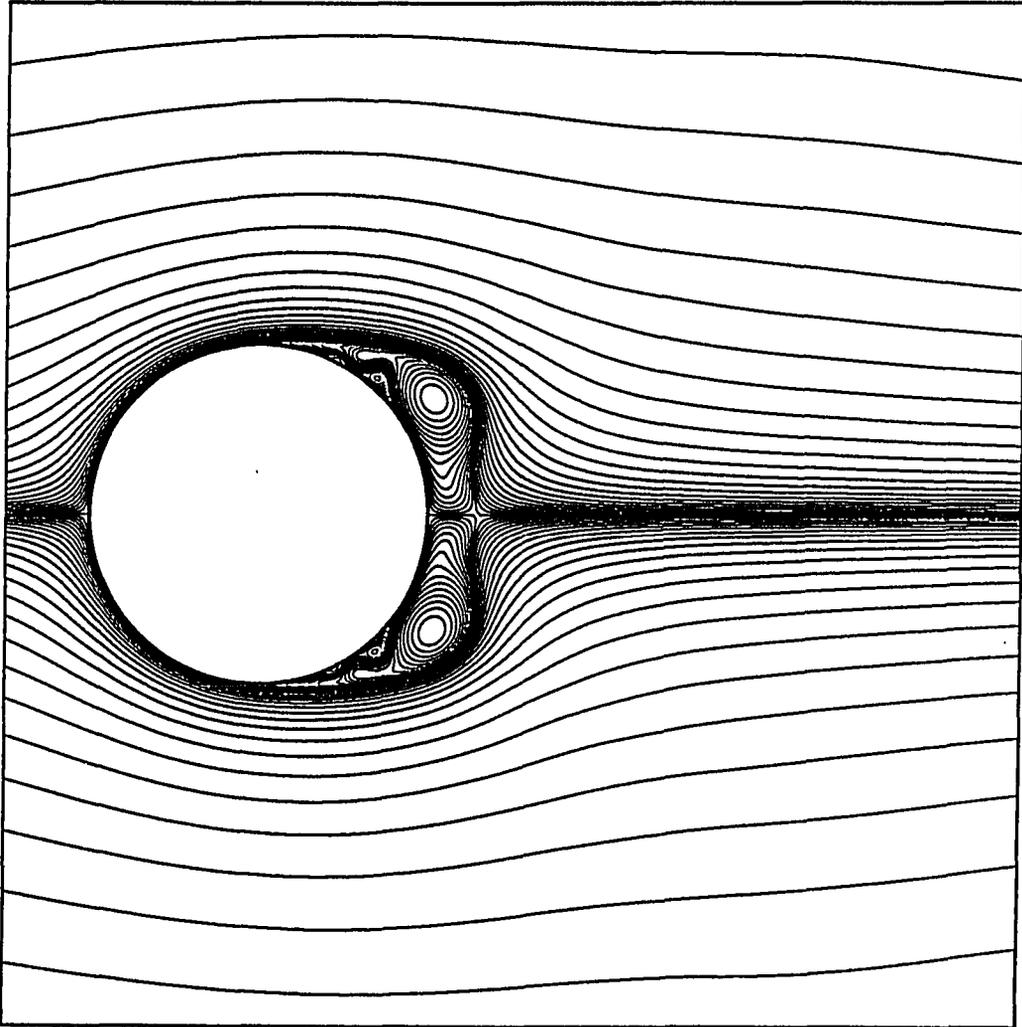
AFLBI Impulsively Started Cylinder $Re=9500$ $Vt/D=1.4$ 

Figure 4.99: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=1.4$)

AFLBI Impulsively Started Cylinder $Re=9500$ $Vt/D=1.6$

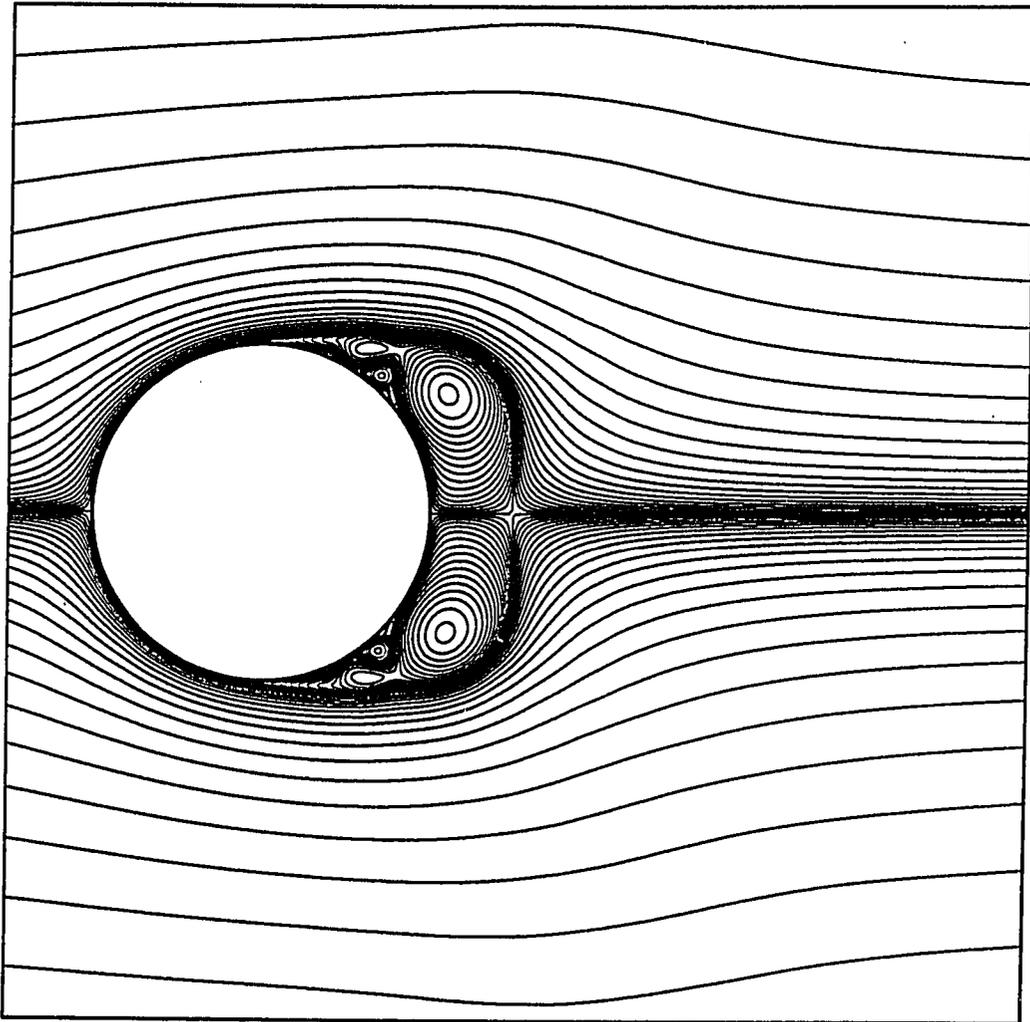


Figure 4.100: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=1.6$)

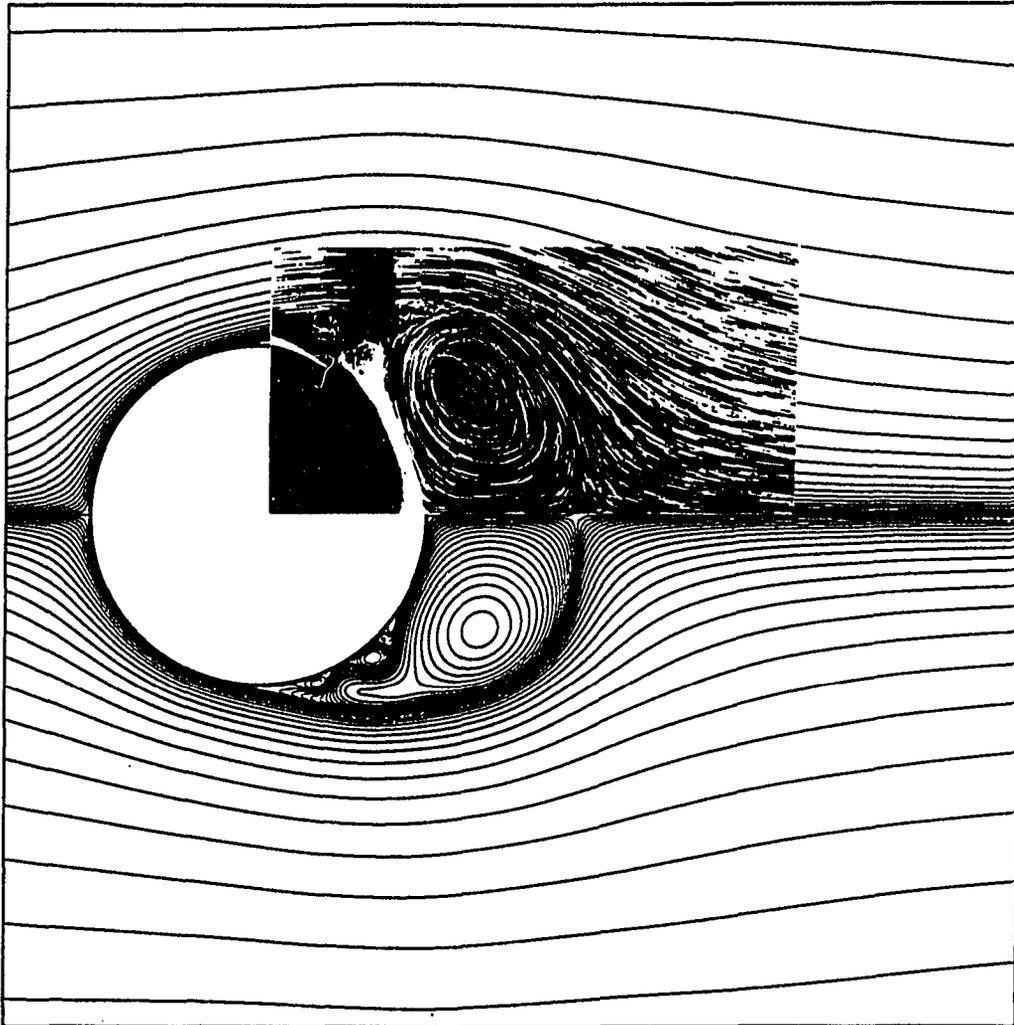
AFLBI Impulsively Started Cylinder $Re=9500$ $Vt/D=2.0$ 

Figure 4.101: AFLBI predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=2.0$)

GCGTVD Impulsively Started Cylinder $Re=9500$ $Vt/D=0.6$

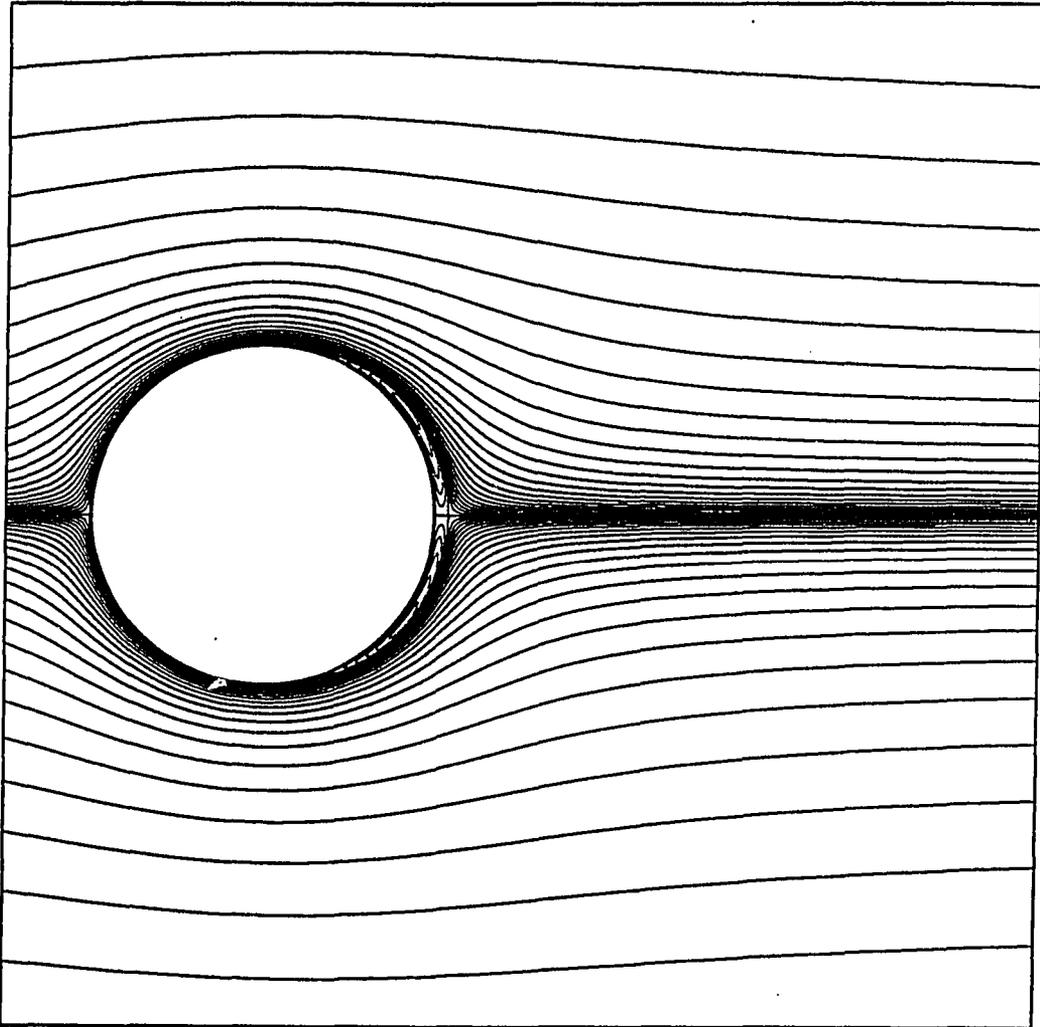


Figure 4.102: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=0.6$)

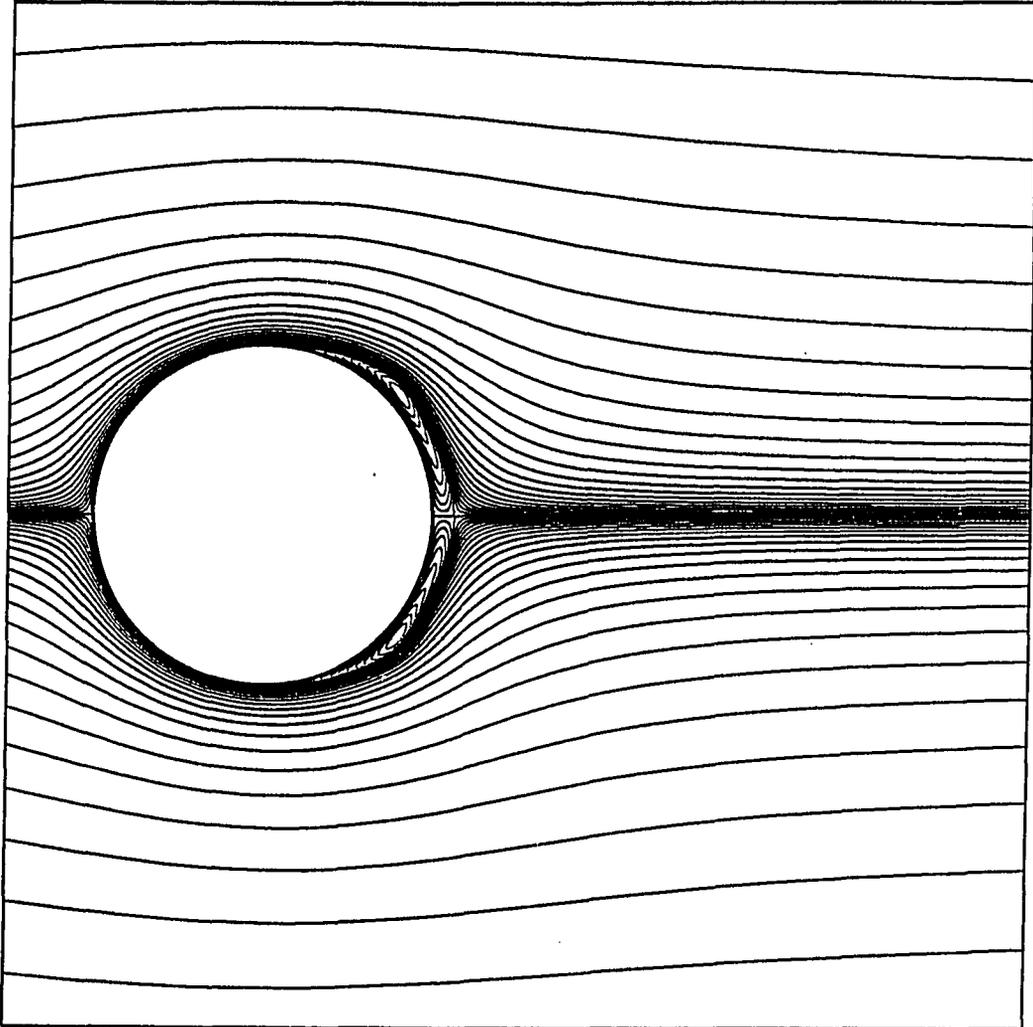
GCGTVD Impulsively Started Cylinder $Re=9500$ $Vt/D=0.8$ 

Figure 4.103: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=0.8$)

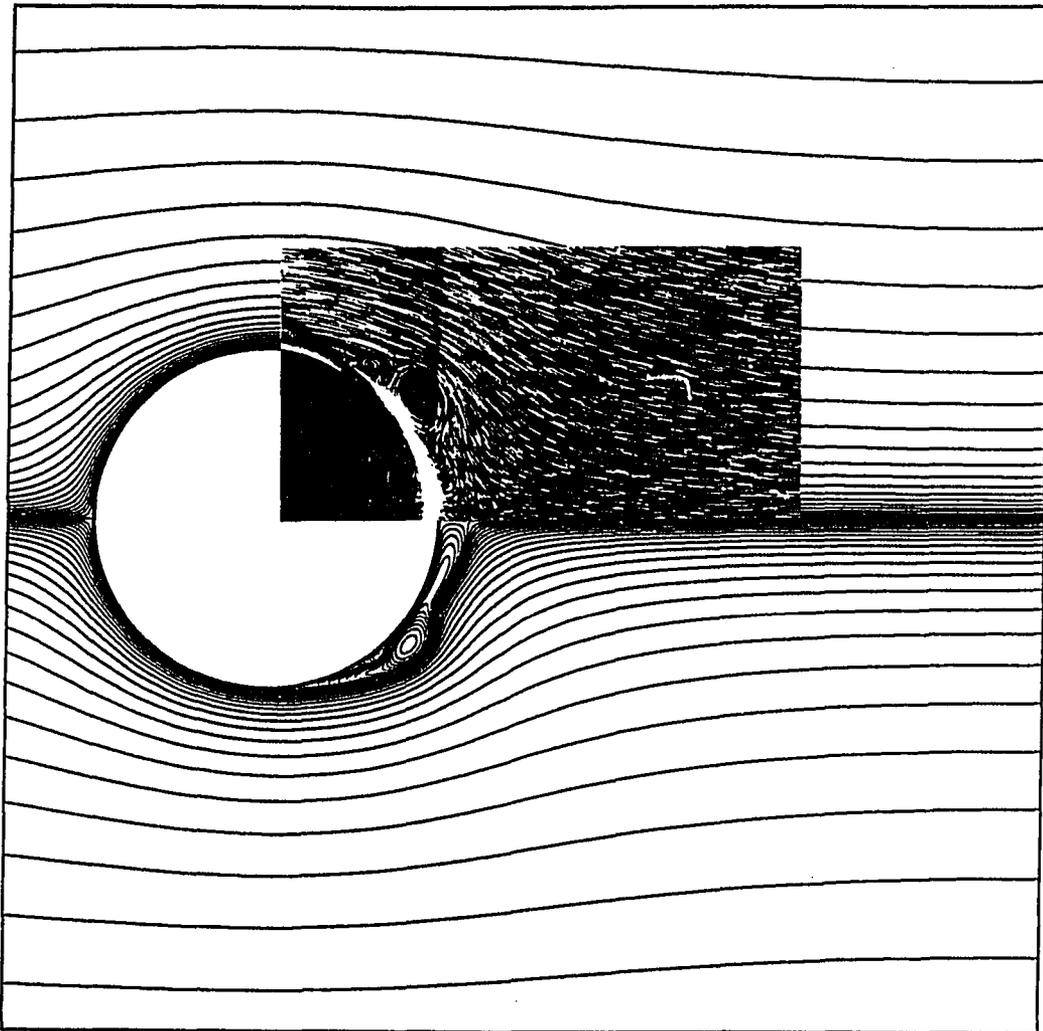
GCGTVD Impulsively Started Cylinder $Re=9500$ $Vt/D=1.0$ 

Figure 4.104: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=1.0$)

GCGTVD Impulsively Started Cylinder $Re=9500$ $Vt/D=1.4$

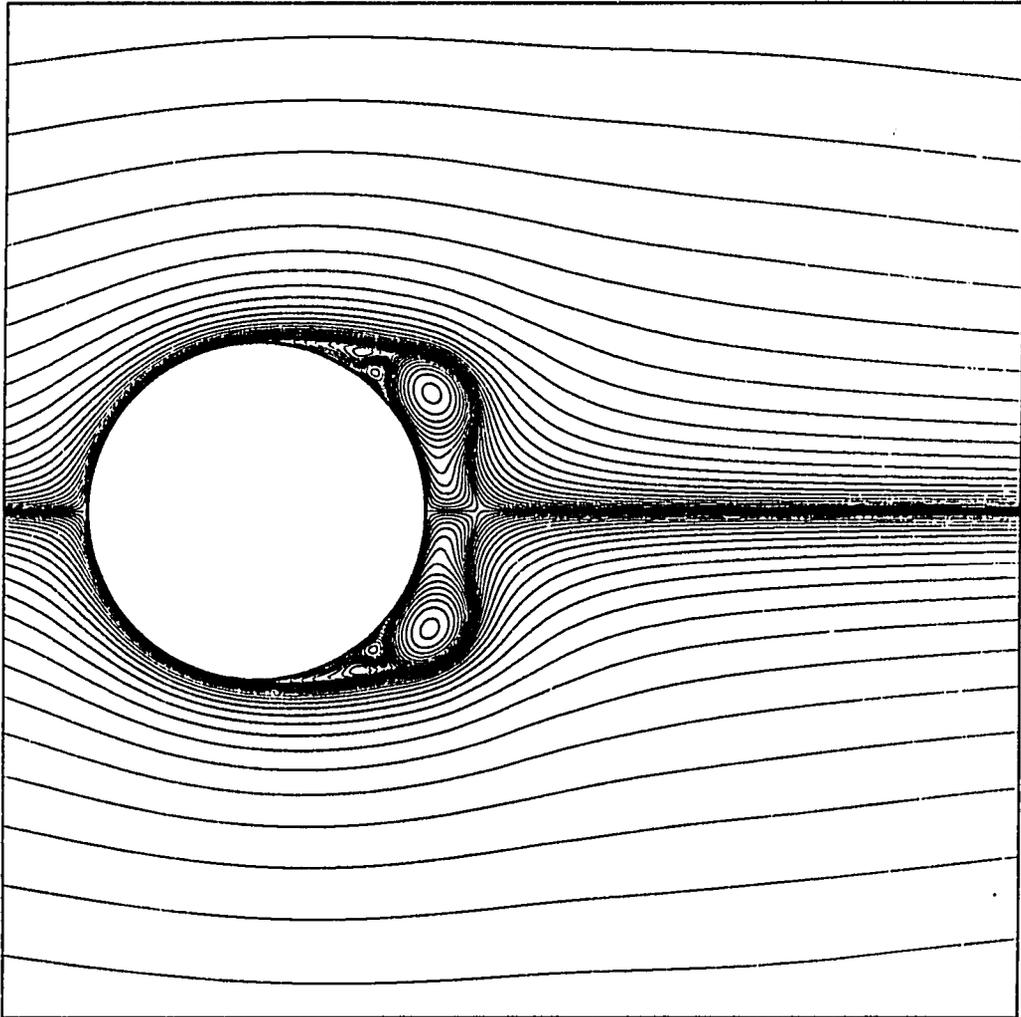


Figure 4.105: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=1.4$)

GCGTVD Impulsively Started Cylinder $Re=9500$ $Vt/D=1.6$

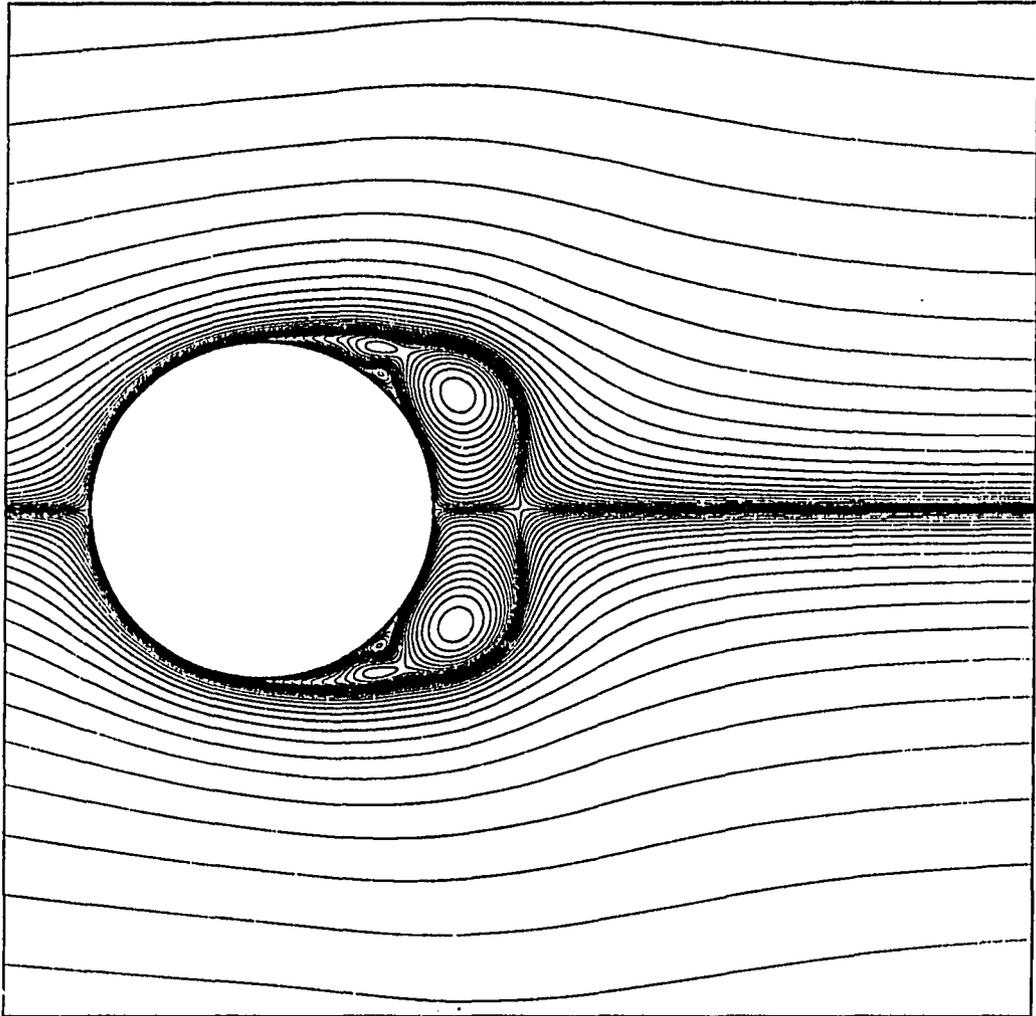


Figure 4.106: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=1.6$)

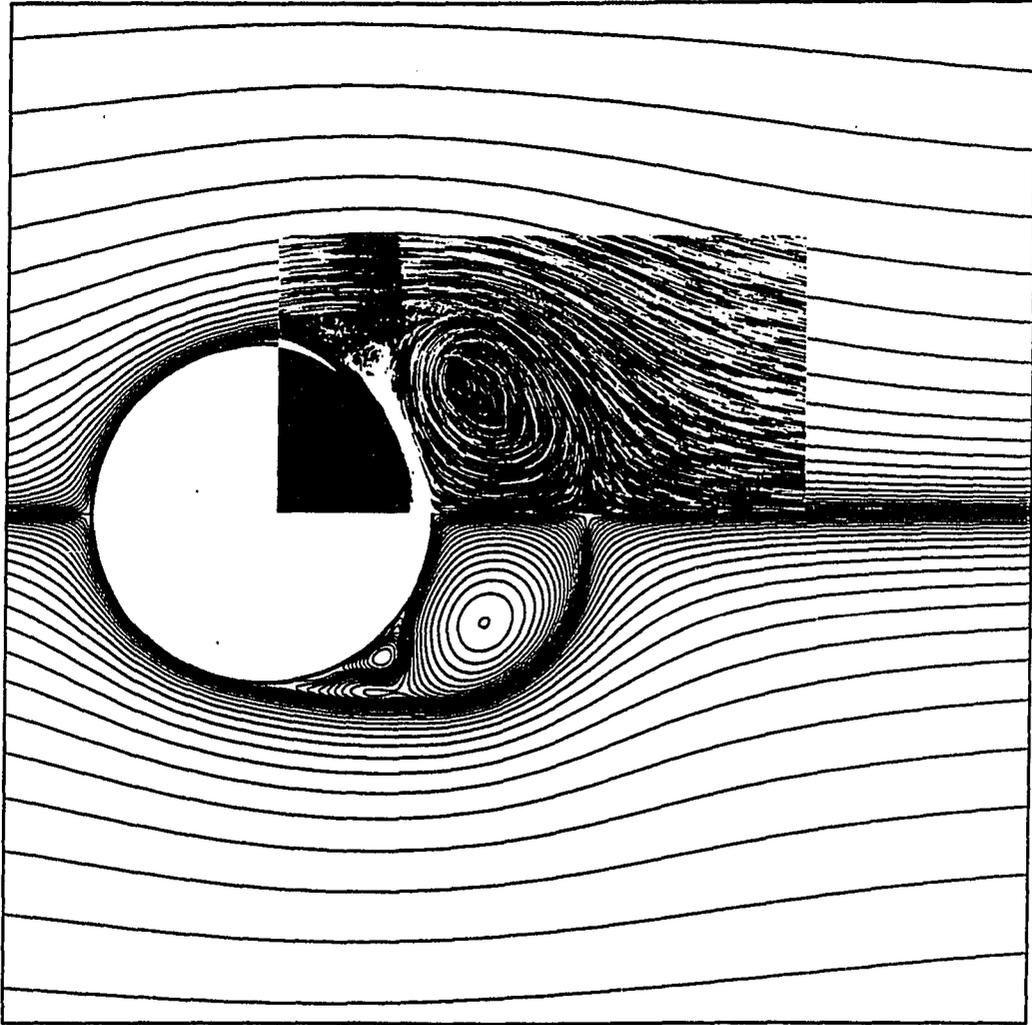
GCGTVD Impulsively Started Cylinder $Re=9500$ $Vt/D=2.0$ 

Figure 4.107: GCGTVD predicted streamline pattern for impulsively started cylinder flow. ($Re=9500$, $t=2.0$)

CHAPTER 5. CONCLUDING REMARKS

Conclusions

Results from four separate finite-difference, time-marching numerical algorithms for the time-dependent Navier-Stokes equations have been compared for steady and unsteady viscous and inviscid compressible flows. A new implicit, nonoscillatory, upwind biased finite volume algorithm for the Navier-Stokes equations was developed utilizing a relaxation approach based on a generalized conjugate gradient routine and was included in the comparisons described above. Each scheme was applied to a series of spatially periodic flows and was found to be both stable and accurate for a variety of flow conditions.

For many of the cases tested, little discernable difference was observed in the results from the four algorithms, unless a significant shock wave was involved, in which case the upwind algorithm was clearly superior. In terms of computational effort, unfortunately, the GCGTVD scheme was consistently the most expensive, followed in order by the AFLBI, Runge-Kutta, and hopscotch algorithms.

The algorithms were developed following an extensive study of a one-dimensional shock tube problem involving 44 different algorithms. The four two-dimensional time-marching algorithms were initially tested for inviscid critical and supercritical flow

over an isolated cylinder. The TVD-based code displayed a superior shock-capturing ability for the supercritical flow case. Each code gave acceptable results for the inviscid flow over two turbine cascades. The grid dependence of the solutions was examined for the prediction of the flow in a low stagger turbine cascade. All of the codes demonstrated the capability of predicting a turbine cascade operating with a leading edge separation. Predictions from the codes demonstrated good agreement with analytical results for both steady and unsteady boundary-layer type flows. The boundary-layer calculation suggested that the implicit formulations were advantageous due to the ability to reach extremely low convergence levels.

The complex transient vortical flow pattern resulting from the impulsive motion of a circular cylinder was also successfully calculated and showed good agreement with both flow visualization and experimental data.

Ultimately, it is very difficult to select an optimum algorithm. Each code has features that make it desirable and other features that make it undesirable. The hopscotch code is very fast, yet suffers from a time step limitation which could inhibit calculations for fine-grid viscous flows. The Runge-Kutta scheme has desirable time-accuracy, but also suffers from the time-step restriction. The AFLBI code, like most implicit codes, requires a significant amount of memory to run, but permits increased time steps. The GCGTVD algorithm is very expensive, but provides outstanding shock resolution. The suitability of any algorithm for a particular flow problem is then best based on the experience of the user and the particular application to be examined.

Recommendations for Future Research

Due to the magnitude and number of calculations performed in this study, there were several areas of this research which warranted further attention and improvement. Unfortunately, a lack of time or money often prohibited further investigation in some areas.

It is believed that further refinement of the conjugate gradient solution procedure should yield improved CPU times for the upwind algorithm. Specifically, execution on multiple processor computer architectures and testing of preconditioned conjugate gradient iterations are areas which show promise. In particular, preconditioning the system by using a strongly implicit (SIP) reduction procedure has shown some promise.

The interpolation scheme in the TVD flux formulation appears to be adequate; however, for highly stretched grids, an interpolation scheme based on the variable grid spacing in the physical plane may be more accurate, and could enhance the accuracy of the algorithm. A small improvement in CPU time might be afforded by using a simple algebraic average for the midpoint variables in the upwind flux construction rather than the more expensive Roe [15] averaging procedure. There may also be some advantage in basing the interpolation on primitive variables rather than characteristic variables, so long as proper limiting is applied. In addition, recent schemes based on nonoscillatory flux constructions have been found to maintain overall second order accuracy without the TVD type limiting scheme [69]. Such schemes may be additionally useful for viscous flow calculations by minimizing the addition of numerical damping.

The steady implicit calculations often suffered from poor convergence in viscous dominated flows due to the lack of viscous influence in the implicit coefficient matrix structure. This might be improved by a more accurate linearization of the implicit viscous fluxes. However, another approach may be to "fortify" the equations with a rapid boundary-layer solution as a means to accelerate the convergence for viscous flows. The use of the fortified Navier-Stokes system is relatively new, and could show some promise for future complex viscous flow calculations.

All of the calculations and comparisons performed here were for laminar flows. The extension to turbulent flow calculations is nontrivial and there are many uncertainties which accompany the introduction of a turbulence model in any algorithm. Specifically, the interaction of a turbulence model and the damping algorithms in central-difference schemes and the limiters in upwind difference schemes are likely to be areas for which uncertainties may arise.

BIBLIOGRAPHY

- [1] Tenpas, P. W., and Pletcher, R. H. "Solution of the Navier-Stokes Equations for Subsonic Flows Using a Coupled Space-Marching Method" AIAA Paper 87-1173, 1987.
- [2] Lewis, J. P., Hall, E. J., and Delaney, R. A. "Numerical Prediction of Turbine Vane-Blade Interaction" AIAA Paper 87-1279, 1987.
- [3] Rai, M. M. "Unsteady Three-Dimensional Navier-Stokes Simulations of Turbine Rotor-Stator Interaction" AIAA Paper 87-2058, 1987.
- [4] Jorgenson, P. C. E., and Chima, R. V. "An Explicit Runge-Kutta Method for Unsteady Rotor-Stator Interaction" AIAA Paper 88-0413.
- [5] Erdos, J. I., and Alzner, E. "Numerical Solution of Periodic Transonic Flow through a Fan Stage" AIAA Journal , 15, 1559-1568, November, 1977.
- [6] Mabey, D. G. "A Review of some Recent Research on Time-Dependent Aerodynamics" Aeronautical Journal, 88, 23-37, February, 1984.
- [7] Brines, G. L. "Design Approach for an Optimum Prop-Fan Propulsion System" Presented at the 1985 Beijing International Gas Turbine Symposium and Exposition, Beijing, Peoples Republic of China, September 1-7, 1985.
- [8] De Ruyck, J., and Hirsch, C. "Instantaneous Flowfield Measurements of Stalled Regions on an Oscillating Airfoil" AIAA Paper 84-1565, 1984.
- [9] Chi, R. M., and Srinivasan, A. V. "Some Recent Advances in the Understanding and Prediction of Turbomachine Subsonic Stall Flutter" Presented at the International Gas Turbine Conference, Amsterdam, Netherlands, June 3-7, 1984.
- [10] Covert, E. E., and Lorber, P. F. "Unsteady Turbulent Boundary Layers in Adverse Pressure Gradients" AIAA Journal , 22, 22-28, 1984.

- [11] Owen, F. K. "Application of Laser Velocimetry to Unsteady Flows in Large Scale, High Speed Tunnels" AIAA Paper 84-25230, 1984. ICIASF '83 Record, 1983.
- [12] McCroskey, W. J., Kutler, P., and Bridgeman, J. O. "Status and Prospects of Computational Fluid Dynamics for Unsteady Transonic Viscous Flows" NASA TM 86018, October, 1984.
- [13] Hindman, R. G. "On Shock Capturing Methods and Why They Work" AIAA Paper 88-0622, 1988.
- [14] Lax, P. D., "Weak Solutions of Nonlinear Hyperbolic Equations and Their Numerical Computation" Communications on Pure and Applied Mathematics, 7, 159-193, 1954.
- [15] Roe, P. L. "Characteristic-Based Schemes for the Euler Equations" Annual Review of Fluid Mechanics , 18, 337-365, 1986.
- [16] Steger, J. L., and Warming, R. F. "Flux Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite-Difference Methods" Journal of Computational Physics, 40, 263-293, 1981.
- [17] Van Leer, B. "Flux Vector Splitting for the Euler Equations" Lecture Notes in Physics, 170, 507-512, 1982.
- [18] Roe, P. L. "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes" Journal of Computational Physics, 43, 357-372, 1981.
- [19] Yee, H. C., and Harten, A. "Implicit TVD Schemes for Hyperbolic Systems of Conservation Laws in Curvilinear Coordinates" AIAA Journal, 25, No. 2, 266-274, 1987
- [20] Chakravarthy, S. R., and Osher, S. "Numerical Experiments with the Osher Upwind Scheme for the Euler Equations" AIAA Journal, 21, No. 9, 1241-1248, 1983.
- [21] Chakravarthy, S. R. "Application of a New Class of High Accuracy TVD Schemes to the Navier-Stokes Equations" AIAA Paper 85-0165, January, 1985.
- [22] Chakravarthy, S. R. "The Versatility and Reliability of Euler Solvers Based on High Accuracy TVD Formulations" AIAA Paper 86-0243, January, 1986.
- [23] Tong, S. S. "Time-Split Inflow and Outflow Boundary Condition Treatment" ASME Paper 85-GT-165, 1985.

- [24] Hedstrom, G. W. "Nonreflecting Boundary Conditions for Nonlinear Hyperbolic Systems" Journal of Computational Physics, 30, 222-237, 1979.
- [25] Zucrow, M. J., and Hoffman, J. D. Gas Dynamics. Volume I., John Wiley and Sons Inc., New York, New York, 1976.
- [26] Zucrow, M. J., and Hoffman, J. D. Gas Dynamics Volume 2: Multidimensional Flow John Wiley and Sons Inc., New York, New York, 1977.
- [27] Stokes, G. G. Trans. Camb. Phil. Soc., 9, Pt. II, 8-106, 1851.
- [28] Rosenhead, L. Laminar Boundary Layers Oxford University Press, London, England, 1963.
- [29] Mirels, H. "Boundary Layer Behind Shock or Thin Expansion Wave Moving into Stationary Fluid" NACA TN 3712, 1956.
- [30] Lighthill, M. J. "The Response of Laminar Skin Friction and Heat Transfer to Fluctuations in the Stream Velocity" Proceedings of the Royal Society, 224A, 1-23, 1954.
- [31] Chow, L. J., and Goorjian, P. M. "Implicit Unsteady Transonic Airfoil Calculations at Supersonic Freestreams" AIAA Paper 82-0934, 1982.
- [32] Caspar, J. R., and Verdon, J. M. "Numerical Treatment of Unsteady Subsonic Flow Past an Oscillating Cascade" AIAA Paper 80-1428, 1980.
- [33] Caspar, J. R. "A Model Problem Study of Transonic Potential Flow Procedures" AIAA Paper 80-0337, 1980.
- [34] Krammer, P. "Computation of Unsteady Blade Forces in Turbomachines by Means of Potential Flow Theory and by Simulating Viscous Wakes" ASME Paper 82-GT-198, 1982.
- [35] Chipman, R., and Jameson, A. "Fully Conservative Numerical Solutions for Unsteady Irrotational Transonic Flow about Airfoils" AIAA Paper 79-1555, 1979.
- [36] Malone, J. B., and Sankar, N. L. "Numerical Simulation of 2-D Unsteady Transonic Flows Using the Full-Potential Equation" AIAA Paper 83-0233, 1983.
- [37] Steger, J. L., and Cardonna, F. X. "A Conservative Implicit Finite Difference Algorithm for the Unsteady Transonic Full Potential Equation" NASA TM 81211, 1980.

- [38] Shankar, V, Ide, H., Gorski, J., and Osher, S. "A Fast, Time-Accurate, Unsteady Full Potential Scheme" AIAA Journal , 25, 230-238, January, 1987.
- [39] Sankar, L. N., Malone, J. B., and Schuster, D. "Full Potential and Euler Solutions for the Unsteady Transonic Flow Past a Fighter Wing" AIAA Paper 85-4061, 1985.
- [40] Jameson, A., and Venkatakrishnan, V. "Transonic Flows about Oscillating Airfoils Using the Euler Equations" AIAA Paper 85-1514, 1985.
- [41] Allmaras, S. R., and Giles, M. B. "A Second Order Flux Split Scheme for the Unsteady 2-D Euler Equations on Arbitrary Meshes" AIAA Paper 87-1119, 1987.
- [42] Belk, D. M., and Whitfield, D. L. "Time-Accurate Euler Equation Solutions on Dynamic Blocked Grids" AIAA Paper 87-1127, 1987.
- [43] Anderson, W. K., Thomas, J. L., and Rumsey, C. L. "Extension and Application of Flux-Vector Splitting to Unsteady Calculations on Dynamic Meshes" AIAA Paper 87-1152, 1987.
- [44] Kwon, O. K., and Delaney, R. A. "Solution Procedure for Unsteady Two-Dimensional Boundary Layers" AIAA Paper 86-1108, 1986.
- [45] Cebeci, T. "Calculation of Unsteady Two-Dimensional Laminar and Turbulent Boundary Layers with Fluctuations in External Velocity" Proc. Royal Soc. London, A, 355, 225-238, 1977.
- [46] Rizzetta, D. P., and Borland, C. J. "Unsteady Transonic Flow over Wings Including Viscous-Inviscid Interaction" AIAA Journal, 21, No. 3, 363-371, 1983.
- [47] Weinberg, B. C., McDonald, H., and Shamroth, S. J. "A Solution Procedure for Two- and Three-Dimensional Unsteady Viscous Flows" AIAA Paper 85-0298, January, 1985.
- [48] Newsome, R. W. "Numerical Simulation of Near-Steady and Unsteady, Subcritical Inlet Flow" AIAA Journal , 22, No. 10, 1375-1379, 1984.
- [49] Osswald, G. A., Ghia, K. N., and Ghia, U. "Study of Incompressible Separated Flow Using an Implicit Time-Dependent Technique" AIAA Paper 87-1139, 1987.

- [50] Ramakrishnan, S. V., and Rubin, S. G. "Numerical Solution of Unsteady Compressible Reduced Navier-Stokes Equations" AIAA Paper 86-0205, January, 1986.
- [51] Ramakrishnan, S. V., and Rubin, S. G. "Transient Flow Past a Finite Flat Plate at Incidence" AIAA Paper 87-0192, January, 1987.
- [52] Ramakrishnan, S. V., and Rubin, S. G. "Time-Consistent Pressure Relaxation Procedure for Compressible Reduced Navier-Stokes Equations" AIAA Journal, 25, No. 7, 905-913, July, 1987.
- [53] Davis, S., Cho, Y.-C., and McLachlan, B. G. "Experimental and Computational Studies of Unsteady Viscous Flows" AIAA Paper 88-3521-CP, 1988.
- [54] Gustafson, K., and Leben, R. R. "Robust Multigrid Computation and Visualization of Separation and Vortex Evolution in Aerodynamic Flows" AIAA Paper 88-3604-CP, 1988.
- [55] Simpson, B., and Whitfield, D. L. "A Flux Difference Split Algorithm for Unsteady Thin Layer Navier-Stokes Solutions" AIAA Paper 89-1995-CP, 1989.
- [56] Meakin, R. L., and Suhs, N. E. "Unsteady Aerodynamic Simulation of Multiple Bodies in Relative Motion" AIAA Paper 89-1996-CP, 1989.
- [57] Lerat, A. "Implicit Methods of Second Order Accuracy for the Euler Equations" AIAA Journal , 23, No. 1, 33-40, 1985.
- [58] Anderson, D. A., Tannehill, J. C., and Pletcher, R. H. Computational Fluid Mechanics and Heat Transfer McGraw-Hill Book Co., New York, 1984.
- [59] White, F. M. Viscous Fluid Flow McGraw-Hill Inc., New York, New York, 1974.
- [60] Vinokur, M. "Conservation Equations of Gas Dynamics in Curvilinear Coordinate Systems" Journal of Computational Physics, 14, 105-125, 1974.
- [61] Sod, G. A. "A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws" Journal of Computational Physics, 22, 1-31, 1978.
- [62] Owczarek, J. A. Fundamentals of Gas Dynamics International Textbook Company, Scranton, Pennsylvania, 1973.
- [63] Delaney, R. A. "Time-Marching Analysis of Steady Transonic Flow in Turbomachinery Cascades Using the Hopscotch Method" J. of Engineering for Power, 105, 272-279, April 1983.

- [64] Jameson, A., Schmidt, W., and Turkel, E. "Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes" AIAA Paper 81-1259, 1981.
- [65] Beam, R. M., and Warming, R. F. "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation-Law Form" Journal of Computational Physics, 22, 87-110, 1976.
- [66] Pulliam, T. H. "Artificial Dissipation Models for the Euler Equations" AIAA Journal, 24, No. 12, 1931-1940, December 1984.
- [67] Chakravarthy, S. R. "Relaxation Methods for Unfactored Implicit Upwind Schemes" AIAA Paper 84-0165, 1984.
- [68] Harten, A. "High Resolution Schemes for Hyperbolic Conservation Laws" Journal of Computational Physics, 49, , 357-393, 1983.
- [69] Harten, A., and Osher, S. "Uniformly High-Order Accurate Nonoscillatory Schemes. I" SIAM Journal of Numerical Analysis, 24, No. 2, 279-309, April 1987.
- [70] Yee, H. C. "Linearized Form of Implicit TVD Schemes for the Multidimensional Euler and Navier-Stokes Equations" Computers and Mathematics with Applications, 12A, Nos. 4/5, 413-432, 1986.
- [71] Anderson, W. K., Thomas, J. L., and Van Leer, B. "Comparison of Finite Volume Flux Vector Splittings for the Euler Equations" AIAA Journal , 24, 1453-1460, 1986.
- [72] Coakley, T. J. "Implicit Upwind Methods for the Compressible Navier-Stokes Equations" AIAA Journal , 23, No. 3, 374-380, 1985.
- [73] Brackbill, J. B., and Saltzman, J. S. "Adaptive Zoning for Singular Problems in Two Dimensions" Journal of Computational Physics, 46, 342-368, 1982.
- [74] Thompson, J. F., "Elliptic Grid Generation" In Numerical Grid Generation, (Edited by Joe Thompson), Elsevier Science Publishing Co., 1982.
- [75] Gordon, P. "Nonsymmetric Difference Equations" J. Soc. Appl. Math., 13, No. 3, 667-673, September 1965.
- [76] Scala, M., and Gordon, P. "Solution of the Time-Dependent Navier-Stokes Equations for the Flow around a Circular Cylinder" AIAA Journal , 6, No. 5, 815-822, May 1968.

- [77] Gourlay, A. R. "Hopscotch: a Fast Second-Order Partial Differential Equation Solver" J. Inst. Math. Appl., 6, 375-390, 1970.
- [78] Gourlay, A. R., and Morris, J. L. "Hopscotch Difference Method for Nonlinear Hyperbolic Systems" IBM J. of Res. and Dev., 16, No. 4, 349-353, July 1972.
- [79] Gottlieb, D., and Gustaffson, B. "On the Navier-Stokes Equation with Constant Total Temperature" NASA CR-132664, 1975.
- [80] Holst, T. L. "The Relative Merits of Several Numerical Techniques for Solving the Compressible Navier-Stokes Equations" NASA CP-2001, Advances in Engineering Science, 4, 1467-1481, 1976.
- [81] Rudy, D. H., Morris, D. J., Blanchard, D. K., Cooke, C. H., and Rubin, S. G. "An Investigation of Several Numerical Procedures for Time-Asymptotic Compressible Navier-Stokes Solutions" NASA SP-347, Aerodynamic Analyses Requiring Advanced Computers, Part 1, 437-468, 1975.
- [82] Gourlay, A. R., "Recent Developments of the Hopscotch Method" In Advances in Computer Methods for Partial Differential Equations II, R. Vichnevetsky, Editor, 1-6, 1977.
- [83] Greenberg, J. B. "Some Semi-Implicit Hopscotch Type Methods for the Time-Dependent Compressible Navier-Stokes Equations" AIAA Paper 81-1022, 1981.
- [84] Shieh, C. F., and Delaney, R. A. "An Accurate and Efficient Euler Solver for Three-Dimensional Turbomachinery Flows" Journal of Turbomachinery, 109, 346-353, July 1987.
- [85] Kwon, O. K., and Delaney, R. A. "A Navier-Stokes Solution Procedure for Analysis of Steady Two-Dimensional Transonic Nozzle Flows" AIAA Paper 85-1597, 1985.
- [86] Kwon, O. K. "Navier-Stokes Solution for Steady Two-Dimensional Transonic Cascade Flows" ASME Paper 87-GT-54, 1987.
- [87] Turkel, E., and Van Leer, B. "Flux Vector Splitting and Runge-Kutta Methods for the Euler Equations" AIAA Journal, 24, 556-570, 1986.
- [88] Jameson, A., and Baker, T. J. "Solution of the Euler Equations for Complex Configurations" AIAA Paper 83-1929, 1983.

- [89] Subramanian, S. V., and Bozzola, R. "Computation of Transonic Flows in Turbomachines Using the Runge-Kutta Integration Scheme" AIAA Journal , 25, 71-72, January, 1987.
- [90] Swanson, R. C., and Turkel, E. "A Multistage Time-Stepping Scheme for the Navier-Stokes Equations" ICASE Report No. 84-62, February, 1985.
- [91] Morinishi, K., and Satofuka, N. "A Numerical Study of Viscous Transonic Flows Using RRK Scheme" AIAA Paper 87-0426, 1987.
- [92] Merkle, C. L., and Tsai, Y. L. "Application of Runge-Kutta Schemes to Incompressible Flows" AIAA Paper 86-0553, 1986.
- [93] Mitchell, A. R., and Griffiths, D. F. The Finite-Difference Method in Partial Differential Equations John Wiley and Sons Publishing, New York, New York, 1980.
- [94] Schafer, O., Fruhauf, H. H., Bauer, B., and Guggolz, M. "Application of Navier-Stokes Analysis to Flows through Plane Cascades" ASME Paper 85-GT-56, 1985.
- [95] Caughey, D. A., and Turkel, E. "Effects of Numerical Dissipation on Finite-Volume Solutions of Compressible Flow Problems" AIAA Paper 88-0621, 1988.
- [96] Chakravarthy, S. R. "Euler Equations-Implicit Schemes and Boundary Conditions" AIAA Journal, 21, No. 5, 699-706, 1983.
- [97] Yee, H. C., Beam, R. M., and Warming, R. F. "Boundary Approximations for Implicit Schemes for One-Dimensional Inviscid Equations of Gasdynamics" AIAA Journal, 20, No. 9, 1203-1211, 1982.
- [98] Godunov, S. K., "A Finite-Difference Method for the Numerical Computation of Discontinuous Solutions of the Equations of Fluid Dynamics" Mat. Sb., 47, 357-393, 1959.
- [99] Chakravarthy, S. R. "A New Class of High Accuracy TVD Schemes for Hyperbolic Conservation Laws" AIAA Paper 85-0363, 1985.
- [100] Chakravarthy, S. R. "Essentially Non-Oscillatory Shock-Capturing Schemes of Arbitrarily-High Accuracy" AIAA Paper 86-0339, January, 1986.
- [101] Hestenes, M. R., and Stiefel, E. "Methods of Conjugate Gradients for Solving Linear Systems" Journal of Res. Nat. Bur. Standards, 49, 409-436, 1952.

- [102] Concus, P., and Golub, G., "A Generalized Conjugate Gradient Method for Nonsymmetric Systems of Linear Equations" Lecture Notes in Economic and Mathematical Systems, 134, 56-65, 1976.
- [103] Elman, H. C., "Preconditioned Conjugate Gradient Methods for Nonsymmetric Systems of Linear Equations" In Advances in Computer Methods for Partial Differential Equations-IV, 409-417, 1981.
- [104] Nicholson, J. H., Forest, A. E., and Oldfield, M. L. G. "Heat Transfer Optimized Turbine Rotor Blades-An Experimental Study Using Transient Technique" ASME Paper 82-GT-04, 1982.
- [105] Jay, R. unpublished data for the high turning turbine blade, Allison Gas Turbine Division, General Motors Corporation, Indianapolis, Indiana, 1982.
- [106] Blasius, H., Z. Angew. Math. Phys., 56, pp. 1-37, (English translation, NACA Technical Memorandum 1256).
- [107] Bouard, R., and Coutanceau, M., "The Early Stage of Development of the Wake Behind an Impulsively Started Cylinder for $40 < Re < 10^4$ " Journal of Fluid Mechanics, 101, No. 3, 583-607, 1980.
- [108] Honji, H., and Taneda, S. Journal of Physics Society of Japan, 27, 1668-1677, 1969.
- [109] Cebeci, T., "Heat Transfer from a Circular Cylinder Impulsively Started from Rest" Numerical Heat Transfer, 1, No. 4, 557-567, 1978.
- [110] Kawamura, T., Hongo, B., and Kuwahara, K. "Computation of High Reynolds Number Flow around a Circular Cylinder with Surface Roughness" AIAA Paper 84-0340, 1984.
- [111] Chamberlain, R. R. "Unsteady Flow Phenomena in the Near Wake of a Circular Cylinder" AIAA Paper 87-0371, 1987.
- [112] Loc, T. P., and Bouard, R., "Numerical Solution of the Early Stage of the Unsteady Viscous Flow around a Circular Cylinder: A Comparison with Experimental Visualization and Measurements" Journal of Fluid Mechanics, 160, 93-117, 1985.
- [113] Giorgini A., and Pravia, J. R. "Evolution of Secondary and Tertiary Vortex Formations in the Wake of an Impulsively Started Circular Cylinder: The Symmetric Wake Case for $Re=3000$." In Proceedings of the International Symposium

on Refined Flow Modelling and Turbulence Measurements, University of Iowa Press, D15-1-D15-10, 1985.

- [114] Rinaldo A., and Giorgini, A. "A Mixed Algorithm for the Calculation of Rapidly Varying Flows: The Impulsively Started Circular Cylinder" International Journal for Numerical Methods in Fluids, 4, 949-969, 1984.
- [115] Cline, M. C. "NAP: A Computer Program for the Computation of Two-Dimensional Time-Dependent Inviscid Nozzle Flow" Los Alamos National Laboratory Report LA-5894, Jan. 1977.

**APPENDIX . EXPLICIT METHOD OF CHARACTERISTICS
BOUNDARY ALGORITHM**

The explicit reference plane method of characteristics boundary algorithm employs the numerical technique described by Cline [115]. The formulation is based on an isentropic form of the governing equations in characteristic form. Since the algorithm is based on the concepts of characteristic theory, some of the details of the derivation are omitted; however, there are many texts [25] which more completely describe the theory.

There are several methods for deriving the characteristic form of the Navier-Stokes equations. In this study, the following treatment is applied. Beginning with the Navier-Stokes equations in transformed coordinates, the energy equation is replaced with an isentropic relation:

$$p/\rho^\gamma = \text{constant} \quad (A.1)$$

and the derivatives in the remaining continuity and momentum equations are expanded to yield the nonconservative form. All pressure terms are eliminated using the isentropic relation. It is then possible to form a general linear combination of the three equations as:

$$\sigma_1(\text{continuity}) + \sigma_2(\xi - \text{momentum}) + \sigma_3(\eta - \text{momentum}) = 0 \quad (A.2)$$

Proper choice of the multiplying factors (σ_{1-3}) reduces the number of independent dimensions in the original equations from three to two. This is equivalent to rewriting the equations along the characteristic surface. The compatibility equations which are valid along each of the characteristic surfaces are derived by explicitly solving for the values of σ_{1-3} and expanding the linear combination described above. Once these equations are established, the actual boundary algorithm proceeds using a reference plane method of characteristics solution technique. This involves solving a finite difference form of the compatibility relations along a characteristic surface involving one set of grid lines, while treating the derivatives along the opposite set of grid lines as source terms.

The characteristic surfaces and compatibility equations may be expressed in the following reference plane format for a calculation at a $\xi = \text{constant}$ boundary point:

Along the stream surface:

$$d\xi = U_\xi dt \quad (A.3)$$

the reference plane compatibility equation is:

$$x_\eta d_L u + y_\eta d_L v = (x_\eta \psi'_2 + y_\eta \psi'_3) dt \quad (A.4)$$

and the derivative operator d_L is defined as:

$$\frac{d_L}{dt} = \frac{\partial}{\partial t} + U_\xi \frac{\partial}{\partial \xi} \quad (A.5)$$

Along the wave surface:

$$d\xi = [U_\xi \mp a\sqrt{x_\eta^2 + y_\eta^2}] dt \quad (A.6)$$

the reference plane compatibility equation is:

$$d_L p \mp \frac{a\rho y_\eta}{\sqrt{\alpha}} d_L u \pm \frac{a\rho x_\eta}{\sqrt{\alpha}} d_L v = [\psi_1^* \mp \frac{a\rho y_\eta}{\sqrt{\alpha}} \psi_2^* \pm \frac{a\rho x_\eta}{\sqrt{\alpha}} \psi_2^*] dt \quad (A.7)$$

and the derivative operator d_L is defined as:

$$\frac{d_L}{dt} = \frac{\partial}{\partial t} + (U_\xi \mp a\sqrt{\alpha}) \frac{\partial}{\partial \xi} \quad (\text{A.8})$$

The source terms ψ are defined as:

$$\begin{aligned} \psi'_2 &= -(U_\eta u_\eta - \frac{y_\xi p_\eta}{\rho}) \\ \psi'_3 &= -(U_\eta v_\eta + \frac{x_\xi p_\eta}{\rho}) \\ \psi_1^* &= -(U_\eta p_\eta - a^2 \rho y_\xi u_\eta + a^2 \rho x_\xi v_\eta) \\ \psi_2^* &= -(U_\eta u_\eta - \frac{y_\xi p_\eta}{\rho}) \\ \psi_3^* &= -(U_\eta v_\eta + \frac{x_\xi p_\eta}{\rho}) \end{aligned} \quad (\text{A.9})$$

Similar relations exist for a constant η boundary. For convenience here, and in the time-marching programs, the viscous derivative terms have been neglected. It is possible to include the viscous terms as source terms by modifying the ψ'_{2-3} and ψ^*_{1-3} , terms above; however, little difference in the computed results is expected for the grids used in this study.

In order to demonstrate the solution procedure for the compatibility equations, it is useful to illustrate the characteristic paths graphically as shown in Fig. A.1 for a subsonic normal inlet and Fig. A.2 for a subsonic normal exit. Here the characteristic paths at the inlet and exit boundaries for subsonic normal flow are outlined in three-dimensional space where the third spatial direction represents time. Only the characteristics emanating from within the computational domain can be used in the boundary point calculation since they indicate the correct propagation of information in the physical domain.

Inlet Boundary Point Characteristic Surfaces

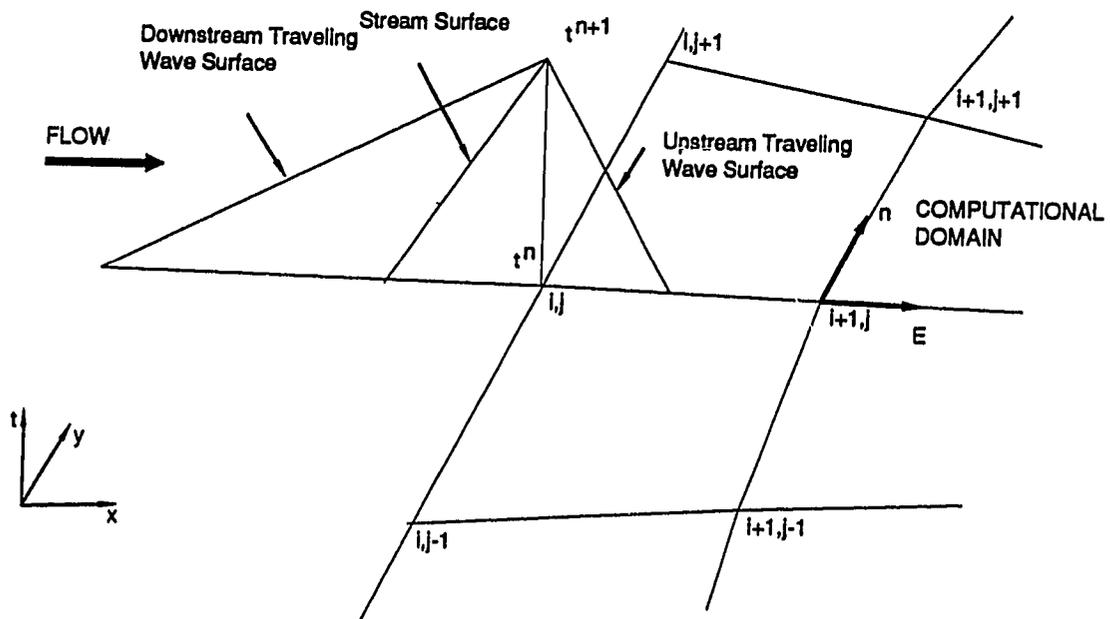


Figure A.1: Characteristic paths for two-dimensional subsonic normal inlet flow calculations

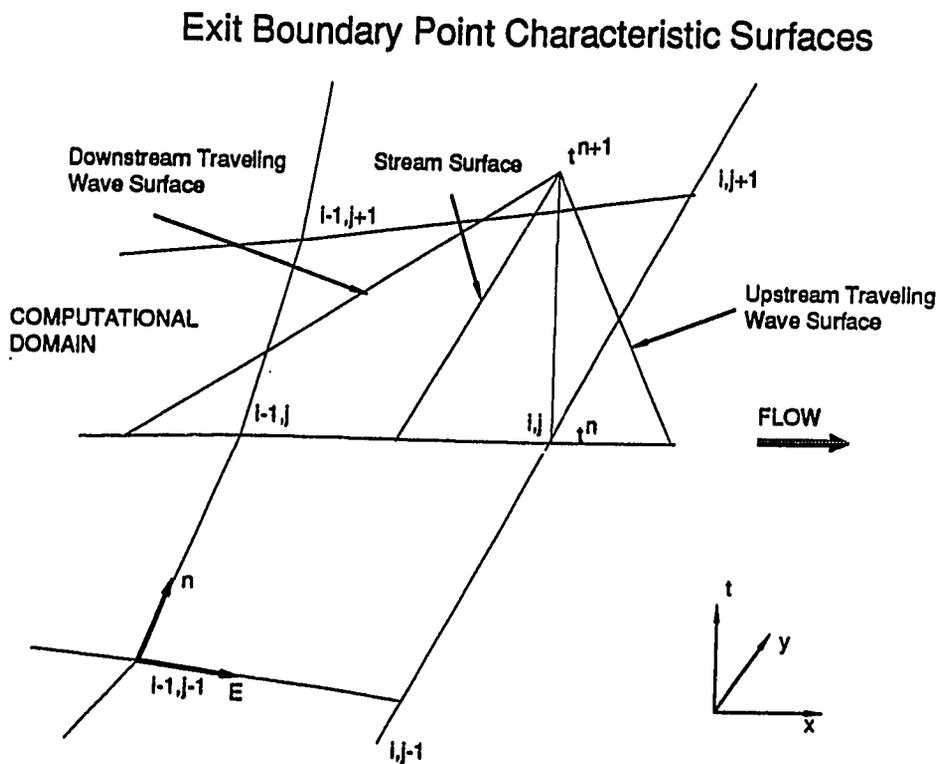


Figure A.2: Characteristic paths for two-dimensional subsonic normal exit flow calculations

Since there are basically five unknowns at each boundary (ρ, u, v, T, p) , additional conditions are required to supplement the characteristic relations valid at the inlet or exit boundary. Different techniques are applied for steady and unsteady flow predictions. For steady flows, the inlet boundary is assumed to have a subsonic normal velocity, and therefore there is only one upstream traveling characteristic relation which may be used. The remaining flow variables are determined by specifying the inlet total pressure and total temperature, and the inlet flow angle. The equation of state then closes the system. The exit boundary is also assumed to have a subsonic normal velocity component, and therefore there are two downstream traveling characteristic relations which may be used. The remaining flow variables are determined by specifying the exit static pressure and extrapolating the total temperature from the interior. Again the equation of state closes the system.

The numerical approximation of the compatibility equations shown above proceeds by writing the compatibility equations in finite difference form along the characteristic surface. Therefore, a term such as $d_j p$ in the wave compatibility equation at the inlet is approximated by $p_{i,j}^{n+1} - p_{base}$, where p_{base} is determined through interpolation of the variables $p_{i,j}^n$ and $p_{i+1,j}^n$. A linear interpolation is used based on the known location of the base of the characteristic surface from:

$$\Delta\xi = \Delta t [U_\xi \mp a \sqrt{x_\eta^2 + y_\eta^2}]_{i,j}^n \quad (A.10)$$

(see Fig. A.1). All non-characteristic surface derivatives are approximated by central differences using data from the n time level. In combination with the remaining boundary specification equations, all of the flow variables can be determined at the new time level. The overall calculation scheme is explicit, and therefore the usual

CFL time step stability criteria must be observed. The truncation error of the scheme behaves as $[O(\Delta t, \Delta x^2)]$, although differencing along the characteristic surface does not necessarily guarantee overall second order spatial accuracy for steady flows.

For unsteady flows, a slightly different concept is applied. The supplemental specification of flow variables at the inlet and exit (i.e., specifying the exit static pressure) can lead to nonphysical reflections of traveling waves in the numerical solution and obscure the true physics of an unsteady flow. In order to avoid the introduction of nonphysical reflections of traveling waves, so-called nonreflecting boundary conditions must be used. Erdos and Alzner [5], Tong [23], and Hedstrom [24] have all derived various boundary conditions which minimize the effects of artificial reflection, and a similar approach is taken here. Following the concept applied by Erdos and Alzner [5] for the prediction of an inviscid rotor-stator interaction problem, a one-dimensional approach is utilized for the flow near an inflow or outflow boundary, and Riemann invariants are specified instead of the flow variables themselves. This approximately replaces the characteristic paths emanating from outside the computational domain, and allows traveling waves to pass through the computational boundary without reflection. Again, the flow normal to both the inlet and exit planes is assumed to be subsonic.

For downstream traveling waves, the Riemann invariant is expressed as:

$$\frac{2a}{(\gamma - 1)} + u = C_1 \quad (A.11)$$

and for upstream traveling waves, the Riemann invariant is:

$$\frac{2a}{(\gamma - 1)} - u = C_2 \quad (A.12)$$

Thus, at the inlet boundary for unsteady flow calculations, the single upstream-running characteristic relation is supplemented by a specification of the inlet flow angle, the inlet entropy, the equation of state, and a specification of the Riemann invariant C_1 . For exit boundary point calculations for unsteady flows, the two downstream-running characteristic relations are supplemented by extrapolation of total enthalpy, the equation of state, and a specification of the Riemann invariant C_2 .