

**PRACTICAL ASPECTS OF COMPUTATIONAL FLUID DYNAMICS
FOR TURBOMACHINERY**

A Dissertation

by

FORREST LAFON CARPENTER IV

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee, Paul G. A. Cizmas
Committee Members, Othon Rediniotis
Leland Carlson
Jean-Luc Guermond
Head of Department, Rodney D. W. Bowersox

August 2016

Major Subject: Aerospace Engineering

Copyright 2016 Forrest LaFon Carpenter IV

ABSTRACT

The following dissertation examines several aspects of numerical simulations for turbomachinery flows modeled with an in-house Reynolds-averaged Navier–Stokes solver. The impact of the turbulence model on the solution is also explored in this work. Additionally, the effects of different solution limiters, including both a new and modified limiter, are examined.

This dissertation also presents a new grid generator that was tailored for turbomachinery geometries. The grid generator uses a combination of structured grid blocks to discretize a single blade passage domain. Structured grid blocks can also be placed at the blade tip, allowing for the modelling of tip leakage flows.

A number of canonical cases were used to validate the additions and modifications to the flow solver. Among these cases were the inviscid flow through a convergent-divergent nozzle and a turbulent flat plate. It is shown that the new and modified limiters perform similarly to the existing limiter functions, and in some cases out-perform their predecessors.

The flow solver is further validated against two turbomachinery cases: an annular turbine vane and a transonic fan. Comparisons with experimental data are made in both cases. The effects of turbulent inlet conditions and the under-relaxation of the turbulence equations are examined for the turbine vane geometry. Two novel rubbing configurations are presented and examined in the turbulent transonic fan case. Additionally, a transonic fan case which includes the tip leakage flow is also presented and compared against the rubbing cases.

to my parents

ACKNOWLEDGEMENTS

Thank you to my advisor and committee chair, Dr. Paul Cizmas, for the wisdom and insight he has shared, and for the patience he has shown throughout my graduate studies. I would also like to thank my committee members, Dr. Othon Rediniotis, Dr. Leland Carlson, and Dr. Jean-Luc Guermond for sharing their time and knowledge with me throughout this undertaking.

Thank you to the Texas A&M High Performance Research Computing (HPRC) center for allowing the use of their systems for my research. A thank you also to the various members of the HPRC help desk for when things were not going quite my way.

I would also like to thank my fellow graduate students, Shalom Johnson, Thomas Brenner, David Liliedahl, Raymond Fontenot, Robert Brown, Neil Matula, and Brian Freno. They have helped and supported me throughout this endeavor too many times to count. A special thank you to Robert Brown, Raymond Fontenot, and Neil Matula for their assistance in the final preparation of this document.

Last but not least, I would like to thank my family. My parents, Forrest and Bren Carpenter, and my sister, Ashley Carpenter for all of their love and encouragement throughout this process. I would not be where I am today if not for them.

TABLE OF CONTENTS

	Page
ABSTRACT	ii
DEDICATION	iii
ACKNOWLEDGEMENTS	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	xi
LIST OF TABLES	xix
NOMENCLATURE	xxi
CHAPTER I INTRODUCTION	1
I.1. Statement of the Problem	1
I.2. Background	5
I.3. Original Contributions	10
I.4. Dissertation Outline	12
CHAPTER II PHYSICAL MODEL	13
II.1. Navier–Stokes Equations	13
II.2. Favre- and Reynolds–Averaged Navier–Stokes Equations	15
II.3. Turbulence Modeling	18
II.4. Equation Nondimensionalization	21
II.4.1. Nondimensional RANS Equations	22
II.4.2. Nondimensional $\kappa - \omega$ SST Equations	23
CHAPTER III NUMERICAL METHODS	25
III.1. Turbomachinery Grid Generation	25
III.1.1. Meridional Grid Generation	26
III.1.2. Blade Surface Definition	29
III.1.3. Blade-to-Blade Grids	29

	PAGE
III.1.4. Blade Tip Clearance Grids	31
III.1.5. Inlet and Outlet Grids	34
III.1.6. Near-Wall Grid Spacing	34
III.2. Flow Solver: UNS3D	36
III.2.1. Navier–Stokes Equations in the Rotat- ing Reference Frame	38
III.2.1.1. Semi-Discrete Form	41
III.2.2. Piecewise Linear Reconstruction	42
III.2.3. Turbulence Model Implementation	47
III.3. Boundary Conditions	50
III.3.1. Wall Boundaries	51
III.3.2. Subsonic Inlet	52
III.3.3. Subsonic Outlet	52
III.3.4. Turbulence Boundary Conditions	53
III.4. Parallelization	54
III.5. Turbomachinery Performance Analysis	55
 CHAPTER IV VERIFICATION AND VALIDATION RESULTS	 59
IV.1. Convergent-Divergent Nozzle	59
IV.1.1. Nozzle Definition and Grid Generation	60
IV.1.2. Case Overview	62
IV.1.3. Solution Post-Processing Methodology	64
IV.1.4. Subsonic Flow	65
IV.1.4.1. Convergence History	65
IV.1.4.2. Center-line Solution	67
IV.1.4.3. Solution Order of Accuracy	70
IV.1.5. Normal Shock Flow	72
IV.1.5.1. Convergence History	72
IV.1.5.2. Numerical “Shock Instabilities”	74
IV.1.5.3. Center-line Solution	79
IV.1.5.4. Solution Order of Accuracy	82
IV.2. NACA 0012	83
IV.2.1. Airfoil Definition and Grid Generation	83
IV.2.2. Boundary Conditions and Solution Strategy	85
IV.2.3. Convergence History	87

	PAGE
IV.2.4. Flow Field Comparison	91
IV.3. Turbulent Flat Plate	94
IV.3.1. Domain Definition and Boundary Conditions . .	94
IV.3.2. Computational Grids	94
IV.3.3. Solution Strategy	96
IV.3.4. Grid Convergence Study	98
IV.3.5. Turbulent Velocity Profiles	100
IV.3.6. Effect of Implicit Residual Smoothing	102
 CHAPTER V TURBINE VANE RESULTS	 107
V.1. Case Definition	107
V.2. Grid Generation	107
V.3. Flow Conditions	109
V.4. Solution Strategy	111
V.5. Convergence History	112
V.6. Grid Convergence Study	112
V.7. Comparison to Experiment	114
V.8. Medium Grid Solution	120
V.9. Impact of Under-relaxing the $\kappa - \omega$ Equations	125
V.10. Inlet Turbulence Boundary Conditions	129
 CHAPTER VI TRANSONIC FAN INVISCID ANALYSIS	 135
VI.1. Case Description: NASA Rotor 67	135
VI.2. Computational Grid Description	138
VI.3. Boundary Conditions	138
VI.4. Initial Operating Point Analysis	141
VI.5. Compressor Map Analyses	141
VI.5.1. Compressor Map Generation Procedure	142
VI.5.2. Residual History Comparison	143
VI.5.3. Rotor Operating Line	144
VI.5.4. Select Operating Points	145
VI.6. Effect of Higher-Order Limiters	150
 CHAPTER VII TRANSONIC FAN TURBULENT ANALYSES	 157

	PAGE
VII.1. Operating Line Generation Procedure	157
VII.2. Rubbing Configuration	159
VII.2.1. Computational Grid	159
VII.2.2. Boundary Conditions	163
VII.2.3. Compressor Map Speed Line Analyses	166
VII.2.3.1. Nonphysical Boundary Solution	166
VII.2.3.2. Operating Point Simulations	167
VII.2.3.3. Defining Stalled Flow	169
VII.2.3.4. Rotor Operating Line	170
VII.2.4. Relative Mach Number Contours	176
VII.3. Tip Leakage Flow Configuration	181
VII.3.1. Computational Grid	181
VII.3.2. Boundary Conditions	182
VII.3.3. Compressor Map Speed Line Analyses	184
VII.3.3.1. Effect of Gradient Method	184
VII.3.3.2. Operating Line Generation Pro- cedure Addendum	188
VII.3.3.3. Rotor Operating Line	189
VII.3.4. Relative Mach Contours	193
VII.3.5. Effect of Higher-Order Limiters	198
 CHAPTER VIII CONCLUSIONS	 205
 CHAPTER IX FUTURE WORK	 208
 REFERENCES	 209
 APPENDIX A: SUPPLEMENTAL TRANSONIC FAN RESULTS	 217
A.1. Inviscid Rotor 67	218
A.2. Turbulent Rotor 67: Rubbing Configuration	219
A.3. Turbulent Rotor 67: Clearance Flow Configuration	224
 APPENDIX B: GOVERNING EQUATION VELOCITY TRANSFORMATION	 229
B.1. Conservation of Mass	229
B.2. Conservation of Momentum	230

	PAGE
B.2.a. X-Momentum Equation	232
B.2.b. Y-Momentum Equation	233
B.2.c. Z-Momentum Equation	234
B.2.d. Final Integral Form	236
B.3. Conservation of Energy	237
 APPENDIX C: ANALYTIC NOZZLE SOLUTION	 241
C.1. Pure Subsonic Case	242
C.2. Normal Shock Case	243
 APPENDIX D: UNS3D INPUT FILES	 246
D.1. Convergent-Divergent Nozzle	248
D.2. Supersonic NACA 0012	250
D.3. Turbulent Flat Plate	253
D.4. Goldman Turbine Vane	256
D.5. NASA Rotor 67	259
D.5.a. Inviscid Analysis	259
D.5.a.i. Initial Operating Point	259
D.5.a.ii. Compressor Map Speed Line	262
D.5.b. Turbulent Analysis: Rubbing	263
D.5.b.i. Compressor Map Speed Line	266
D.5.c. Turbulent Analysis: Clearance	266
D.5.c.i. Compressor Map Speed Line	269
D.6. Ada HPC System Batch File Example for MPI	269
 APPENDIX E: TURBOMACHINERY GRID GENERATOR INPUT FILES	 271
E.1. Turbine Vane	271
E.1.a. Coarse Grid	271
E.1.b. Medium Grid	273
E.1.c. Fine Grid	276
E.2. Rotor 67	278
E.2.a. Inviscid Rotor 67	278
E.2.b. Turbulent Rotor 67: Rubbing	280
E.2.c. Turbulent Rotor 67: Clearance	283

APPENDIX F: TURBOMACHINERY GRID GENERATOR MANUAL	286
F.1. Introduction	286
F.2. Compiling and Running TGG	286
F.3. Input Files	287
F.3.a. Main Input File	287
F.3.a.i. &grid_size	287
F.3.a.ii. &grid_dims	289
F.3.a.iii. &grid_ctrl	291
F.3.a.iv. &cap_grid	292
F.3.a.v. &pad_grid	293
F.3.a.vi. &bounds	294
F.3.a.vii. &ctrls	294
F.3.a.viii. Post Namelist Input	296
F.3.b. Flow Path Definition	297
F.3.c. Blade Surface Definition	297
F.3.d. Freestream Profile Definition	299
F.4. Output Files	300
F.4.a. Unstructured Grid File	300
F.4.b. Ancillary Flow Solver Files	301
F.4.b.i. Annular Flow Initialization: XleXteRte	301
F.4.b.ii. Complete Structured-to-Unstructured Pointer: ijk.def	302
F.4.b.iii. Outlet Structured-to-Unstructured Pointer: outlet_ij.def	303
F.4.c. Unstructured Tecplot Data File	304
F.4.d. Structured Tecplot Data File	305
F.4.e. Tecplot Data Files For Debugging	306

LIST OF FIGURES

FIGURE		Page
I.1	Compressor secondary flows.	3
III.1	Defining the meridional grid with user prescribed boundaries.	27
III.2	Typical blade-to-blade meridional mesh showing the indicial coordinate system and spanwise arc length distribution vector, \vec{S}_k	28
III.3	O-4H multi-block blade-to-blade grid layout with block numbers.	30
III.4	Blade tip clearance grid (white shaded region) leading and trailing edge detail showing the block boundaries (red edges).	32
III.5	Typical blade tip clearance grid slice.	33
III.6	Velocity diagram for a point rotating about the inertial x-axis.	39
III.7	Linear reconstruction of a scalar variable, q	43
III.8	Stencils for the Venkatakrishnan ($---$) and MLP ($- \cdot \cdot -$) limiters in two dimensions; “ \circ ” denote primary grid vertices, “ \square ” are dual mesh vertices, “ \odot ” denote vertices used to compute variable extremum for a dual mesh vertex j	45
III.9	Flux quadrature points (\circ) for a boundary adjacent dual mesh volume ($---$).	51
III.10	Defining geometric quantities using an idealized extracted grid (black, solid) and the corresponding mesh duals (grey, dashed).	56
IV.1	Convergent-divergent nozzle profile.	60
IV.2	Medium nozzle grid and grid refinement detail.	61
IV.3	Subsonic convergent-divergent nozzle density residual histories.	66

FIGURE	Page
IV.4	Subsonic convergent-divergent nozzle center-line Mach number profiles for each grid. 68
IV.5	Subsonic convergent-divergent nozzle percent difference between solutions computed using the $-\epsilon_1$ and $-\epsilon_2$ variants of the Venkatakrishnan and MLP limiters. 69
IV.6	Subsonic convergent-divergent nozzle mass flow rate percent error variation with grid size. 71
IV.7	Normal shock convergent-divergent nozzle density residual histories. 74
IV.8	Normal shock convergent-divergent Venka- ϵ_1 density residual histories on the super-fine grid for increasing limiter constant values. 75
IV.9	Normal shock convergent-divergent nozzle Venka- ϵ_1 solutions on the super-fine grid for different values of the limiter constant. 77
IV.10	Normal shock convergent-divergent MLP- ϵ_1 density residual histories on the medium grid for increasing limit constant values. . 78
IV.11	Normal shock convergent-divergent MLP- ϵ_1 numerical “shock instability” map for different values of the limiter constant. 79
IV.12	Examination of the converged MLP- ϵ_1 normal shock convergent-divergent nozzle solutions. 80
IV.13	Normal shock convergent-divergent nozzle center-line Mach number profiles for each grid. 81
IV.14	NACA 0012 unstructured grid. 85
IV.15	Boundary conditions applied to the NACA 0012 flow domain. . . . 86
IV.16	Supersonic NACA 0012 residual convergence histories with the Venkatakrishnan limiters. 88

FIGURE	Page
IV.17	Supersonic NACA 0012 residual convergence histories with the MLP limiters. 89
IV.18	Normalized MLP- ϵ_1 pressure contours after 10,000 iterations. . . . 90
IV.19	Sampled normalized pressure histories compared against corresponding residual histories. 90
IV.20	NACA 0012 Mach number contours for each limiter solution. 92
IV.21	Flat plate computational domain. 95
IV.22	Coarse flat plate grid. 95
IV.23	Flat plate solution residual comparison. 99
IV.24	Computed surface distributions of y^+ for each flat plate grid. . . . 100
IV.25	Computed skin friction coefficient compared against experiment and theory. 101
IV.26	Fine flat plate grid extracted velocity profiles compared against experiment and empirical relations. 103
IV.27	Comparing fine grid residual histories for smoothed and not smoothed $\kappa - \omega$ equations. 104
IV.28	Profiles of κ and ω comparing the effect of using implicit residual smoothing (IRS) on the $\kappa - \omega$ equations at $Re_x = 5.018 \times 10^6$ 106
V.1	Typical turbine vane computational domain. 109
V.2	Turbine vane midspan blade-to-blade grid detail. 110
V.3	Turbine vane residual history plots for all three grids. 113
V.4	Computed mass flow rate variation with grid size. 114
V.5	Computed surface isentropic Mach numbers versus experiment. . . . 116

FIGURE	Page
V.6	Predicted and experimental spanwise profiles of flow angle, total pressure loss, and normalized static pressure. 118
V.7	Comparing predicted and measured normalized total pressure contours at 1/3 axial chords downstream of the vane trailing edge. 119
V.8	Predicted turbine vane Mach contours at midspan. 120
V.9	Predicted y^+ numbers along the turbine vane and hub surfaces. . . 121
V.10	End wall vortex locations given by a single iso-surface of normalized vorticity magnitude. 123
V.11	Velocity vectors tangential to the leading edge grid plane showing the end wall horseshoe vortices and the counter-rotating corner vortices. 124
V.12	Streamlines near the suction side surface showing liftoff of the passage vortex for: (a) current prediction and (b) prediction by Arnone et al. [1993]. 125
V.13	Turbine vane $\kappa - \omega$ residual history comparison for different under-relaxation factors. 127
V.14	Turbine vane flow residual history comparison for different under-relaxation factors. 128
V.15	Downstream spanwise profiles of flow angle and total pressure loss for different under-relaxation factors. 130
V.16	Turbine vane $\kappa - \omega$ residual histories for the different inlet turbulence conditions. 132
V.17	Turbine vane flow residual history comparison for the different inlet turbulence conditions. 133
V.18	Downstream spanwise profiles of flow angle and total pressure loss for different inlet turbulence conditions. 134

FIGURE	Page
VI.1 Rotor 67 blade and flow path geometry.	136
VI.2 Numerical measurement plane locations.	137
VI.3 Full inviscid Rotor 67 computational grid with a detail of the inlet grid.	139
VI.4 Tip blade-to-blade grid for the inviscid Rotor 67 case.	139
VI.5 Inviscid Rotor 67 initial operating point residual histories.	142
VI.6 Density residual histories for the inviscid Rotor 67 speed line.	144
VI.7 Predicted inviscid rotor performance at design wheel speed compared against experimental data.	146
VI.8 Predicted (left) and measured (right) relative Mach number contours at maximum flow, and at 70% span.	149
VI.9 Predicted (left) and measured (right) relative Mach number contours near peak efficiency, and at 70% span.	150
VI.10 Predicted (left) and measured (right) relative Mach number contours near stall, and at 70% span.	151
VI.11 Density residual histories for the initial inviscid Rotor 67 operating point using different higher-order limiters.	152
VI.12 Higher-order limiter solution comparison using downstream profiles of normalized static pressure and absolute flow angle.	154
VI.13 Predicted relative Mach number contours at 70% span for different higher-order limiters.	156
VII.1 Turbulent rubbing Rotor 67 tip layer blade-to-blade grid.	161
VII.2 First cell height variation across the rotor surface.	162

FIGURE	Page
VII.3 Domain decomposition breakdown for the turbulent rubbing Rotor 67 grid.	162
VII.4 Defining the allowable region for which rotating wall boundaries can rotate.	164
VII.5 Static temperature contours showing the nonphysical solution point for Case R67TR-b on the tip grid layer.	168
VII.6 Density and turbulent kinetic energy second order residual histories for both turbulent rubbing Rotor 67 cases.	169
VII.7 Iso-surfaces of axial velocity showing large regions of reversed flow indicating stall for Case R67TR-a.	170
VII.8 Predicted Rotor 67 performance for two turbulent rubbing configurations compared against experiment.	173
VII.9 Comparison of predicted and measured downstream hub static pressures for each operating point.	175
VII.10 Comparing spanwise profiles of predicted static pressure and absolute flow angle against measurement for both rubbing Rotor 67 configurations at “near peak efficiency”.	177
VII.11 Predicted and measured relative Mach number contours at 70% span for the “max flow” operating condition.	178
VII.12 Predicted and measured relative Mach number contours at 70% span for the “max flow” operating condition.	180
VII.13 Rotor 67 tip clearance grids.	183
VII.14 Turbulent Rotor 67 initial operating point residual histories for the flow and turbulence equations.	185
VII.15 Turbulent Rotor 67 initial operating point residual histories for different gradient evaluation methods.	186

FIGURE	Page
VII.16	Predicted Rotor 67 performance for two turbulent clearance flow cases compared against experiment and a turbulent rubbing configuration. 192
VII.17	Normalized hub static pressures measured at the downstream measurement location. 193
VII.18	Comparing spanwise profiles of predicted Rotor 67 static pressure and absolute flow angle against measurement at “near peak efficiency”. 194
VII.19	Predicted and measured relative Mach number contours at 70% span for the “max flow” operating condition. 196
VII.20	Predicted and measured relative Mach number contours at 70% span for the “near peak efficiency” operating condition. 197
VII.21	Comparison of the turbulent Rotor 67 residual convergence histories with higher-order limiter choice at the initial operating point. 199
VII.22	Comparison of the turbulent Rotor 67 adiabatic efficiency convergence histories with higher-order limiter choice at the initial operating point. 200
VII.23	Comparing spanwise profiles of static pressure and absolute flow angle, at both measurement planes, for turbulent Rotor 67 with different higher-order limiters. 202
VII.24	View looking downstream of relative velocity streamlines which show the separated region near the trailing edge and hub for Rotor 67 at the initial operating point. 203
VII.25	Predicted initial operating point turbulent relative Mach number contours comparing higher-order limiters at 70% span. 204
A.1	Predicted relative Mach number contours at 30% span for “max flow”. 221

FIGURE	Page
A.2	Predicted relative Mach number contours at 90% span for “max flow” 221
A.3	Predicted and measured relative Mach number contours at 30% span for “near peak efficiency”. 222
A.4	Predicted and measured relative Mach number contours at 90% span for “near peak efficiency”. 223
A.5	Predicted relative Mach number contours at 30% span for “max flow” 226
A.6	Predicted relative Mach number contours at 30% span for “max flow” 226
A.7	Predicted and measured relative Mach number contours at 30% span for “near peak efficiency”. 227
A.8	Predicted and measured relative Mach number contours at 90% span for “near peak efficiency”. 228
C.1	Defining the convergent-divergent nozzle. Normal shock quantities are given in maroon, location given by dashed line. 241
F.1	Blade-to-blade O-4H grid layout. 308
F.2	Defining the meridional grid. 309
F.3	Grid spacing around an airfoil section. 310
F.4	Spanwise grid spacing option definitions. 311
F.5	Defining the clearance O-H grids. 314

LIST OF TABLES

TABLE		Page
II.1	$\kappa - \omega$ (1) and $\kappa - \epsilon$ (2) model constants.	20
IV.1	Convergent-divergent grid dimensions	62
IV.2	Nozzle flow initial and boundary conditions.	62
IV.3	Subsonic convergent-divergent nozzle apparent solution orders. . . .	72
IV.4	Apparent order of the normal shock solutions using the finest three grids.	83
IV.5	NACA 0012 flow conditions.	86
IV.6	NACA 0012 drag coefficients	93
IV.7	Flat plate grid dimensions.	97
V.1	Turbine vane grid sizes and dimensions.	108
V.2	Turbine vane turbulence inlet boundary condition test matrix. . . .	131
VI.1	Inviscid Rotor 67 boundary conditions.	140
VI.2	Predicted and measured rotor performance at maximum flow, near peak efficiency, and near stall.	147
VI.3	Initial operating point rotor performance with respect to the limiter used.	153
VII.1	Turbulent rubbing Rotor 67 boundary conditions.	165
VII.2	Maximum mass flow rates for both turbulent rubbing Rotor 67 cases with comparison to experiment.	171
VII.3	Comparison between predicted and measured rotor perfor- mance at “near peak efficiency” for two turbulent rubbing Rotor 67 cases.	174

TABLE	Page
VII.4	Turbulent Rotor 67 boundary conditions. 184
VII.5	Turbulent Rotor 67 initial operating point performance variation with gradient calculation method. 187
VII.6	Predicted and measured equivalent maximum mass flow rates for turbulent Rotor 67. 190
VII.7	Comparing averaged turbulent Rotor 67 performance characteristic variations with higher-order limiter choice at the initial operating point. 201
A.1	Inviscid Rotor 67 compressor map speed line outlet pressure boundary condition and results summary. 218
A.2	Turbulent Rotor 67: Rubbing case R67TR-a compressor map speed line outlet boundary condition and results summary. . . 219
A.3	Turbulent Rotor 67: Rubbing case R67TR-b compressor map speed line outlet boundary condition and results summary. . . 220
A.4	Turbulent Rotor 67 compressor map speed line outlet boundary condition and results summary using the Dervieux limiter and the un-weighted LSQR gradient method. 224
A.5	Turbulent Rotor 67 compressor map speed line outlet boundary conditions and results summary using the Dervieux limiter and the Green-Gauss gradient method. 225
D.1	UNS3D input parameters for the higher order limiter and their options. 247
F.1	Turbomachinery grid generator debug Tecplot data file descriptions. 307

NOMENCLATURE

Roman Letters

c	Speed of sound
c_a	Axial chord length
c_p	Specific heat capacity at constant pressure
c_v	Specific heat capacity at constant volume
e	Internal energy per unit mass
E	Total energy per unit mass
\vec{g}	Gravitational force vector
h	Enthalpy
H	Total enthalpy
I	Rothalpy
$\bar{\bar{I}}$	Identity matrix
I_T	Turbulence intensity in percent
K	Turbulent kinetic energy
k	Thermal conductivity coefficient
k_T	Turbulent thermal conductivity coefficient

M	Mach number
p	Pressure
Pr	Prandtl number
Pr_T	Turbulent Prandtl number
\dot{q}	Heat source
r	Radius, $r = \sqrt{y^2 + z^2}$
Re	Reynolds number, $Re = \rho Ul/\mu$
R	Specific gas constant
t	Time
T	Temperature
\vec{U}	Absolute velocity vector
(u_1, u_2, u_3)	Absolute velocity vector components
\vec{W}	Relative velocity vector
(w_1, w_2, w_3)	Relative velocity vector components
(x, y, z)	Cartesian coordinates

Greek and Hebrew Letters

λ	Bulk viscosity
-----------	----------------

δ_{ij}	Kronecker delta
μ	Dynamic viscosity
μ_T	Dynamic eddy viscosity
\aleph	Generic scalar variable
ω	Rate of dissipation per unit turbulent kinetic energy
ρ	Density
$\bar{\bar{\tau}}$	Viscous stress tensor
θ	Circumferential angle, $\theta = \tan^{-1}(y/z)$
$\vec{\varphi}_R$	Relaxation factor vector
ω_x	Rotational rate about the x-axis [radians/sec]
γ	Ratio of specific heats, $\gamma = 1.4$ for air

Superscripts

(<i>h</i>)	Quantity located at hub
(<i>t</i>)	Quantity located at tip

Subscripts

1	Inlet
2	Outlet

<i>abs</i>	Absolute
<i>eq</i>	Equivalent flow condition
∞	Free stream
<i>le</i>	Leading edge
<i>rel</i>	Relative
<i>te</i>	Trailing edge
0	Total

Accents

$\tilde{\mathfrak{N}}$	Favre averaged variable
$\check{\mathfrak{N}}$	Nondimensional variable
$\bar{\mathfrak{N}}$	Reynolds averaged variable

Acronyms and Software

HPC High Performance Computing

MPI Message Passing Interface

RANS Favre- and Reynolds-Averaged Navier–Stokes

SST Shear Stress Transport

TGG Turbomachinery grid generator (software)

UNS3D Unsteady, three-dimensional Navier-Stokes solver (software)

CHAPTER I

INTRODUCTION

I.1. Statement of the Problem

The emphasis in current aero-engine design is in increasing the overall efficiency of the system. This includes improving the aerodynamic and structural characteristics of turbomachinery components to result in higher aerodynamic efficiencies and lower component weights. The net results of this effort are improved fuel burn characteristics and reduced operating costs. This work will focus on the aerodynamic characteristics of the flow through rotating and stationary blade rows.

Improving the aerodynamic characteristics of a turbomachine is no easy task as the flow is inherently unsteady, three-dimensional, and turbulent. Two of the three main components of a turbomachine, the compressor and the turbine, are made up of multiple rows of rotors and stators. In a multi-row environment, the relative motion of the blade rows results in a flow that can only be unsteady [Smith, 1955]. The different blade rows interact with one another through potential flow effects and primarily by the wakes shed by the blades and any other structure embedded in the flow [Parker & Watson, 1972]. The influence of the shed wakes is not limited to the stator or rotor just downstream. The wake can travel through the adjacent downstream row and effect the following row [Sanders & Fleeter, 2002; Key et al., 2010]. The effect of these multi-row interactions is to act as a source of unsteady aerodynamic forcing, which can lead to increased fatigue and even failure in the

affected turbomachinery components.

Multi-row effects can be ignored by examining a rotor (or stator) row in isolation. However, sources of flow unsteadiness and non-uniformity, known as secondary flows, can still be found. Secondary flows, which are naturally present in the multi-row environment, are defined by Smith [1955] as the difference between an idealized axisymmetric flow and the actual flow. Secondary flows include, but are not limited to, passage vortices, tip leakage flow, tip leakage vortex, and corner vortices [Lakshminarayana & Horlock, 1963; Langston, 2001]. An illustration of these secondary flows, originally from Lakshminarayana & Horlock [1963], is shown in Fig. I.1. As an example, in turbine cascades, secondary losses due to the passage and corner vortices, typically result in between 33% [Denton, 1993] to upwards of 50% [Langston, 2001] of the total aerodynamic losses. Aerodynamic losses due to skin friction, boundary layer growth, and separation along the surface of the blades account for nearly the rest.

Computation fluid dynamics (CFD) is the primary tool in use today for designing and optimizing turbomachinery components such that the effect of secondary and profiles losses are attenuated. The most obvious components to optimize are the blades themselves. Weingold et al. [1997], using Euler analysis, developed a bowed compressor stator which can either delay or suppress the formation of corner vortices, resulting in efficiency increases on the order of 1%. Lee & Kim [2000] used a Navier–Stokes solver to optimize the stacking line of a stator from a single axial compressor stage, resulting in a 1.1% increase in efficiency. Rubechini et al. [2012] were able to increase the efficiency of a 17-stage by 0.24% by optimizing the shape of all of

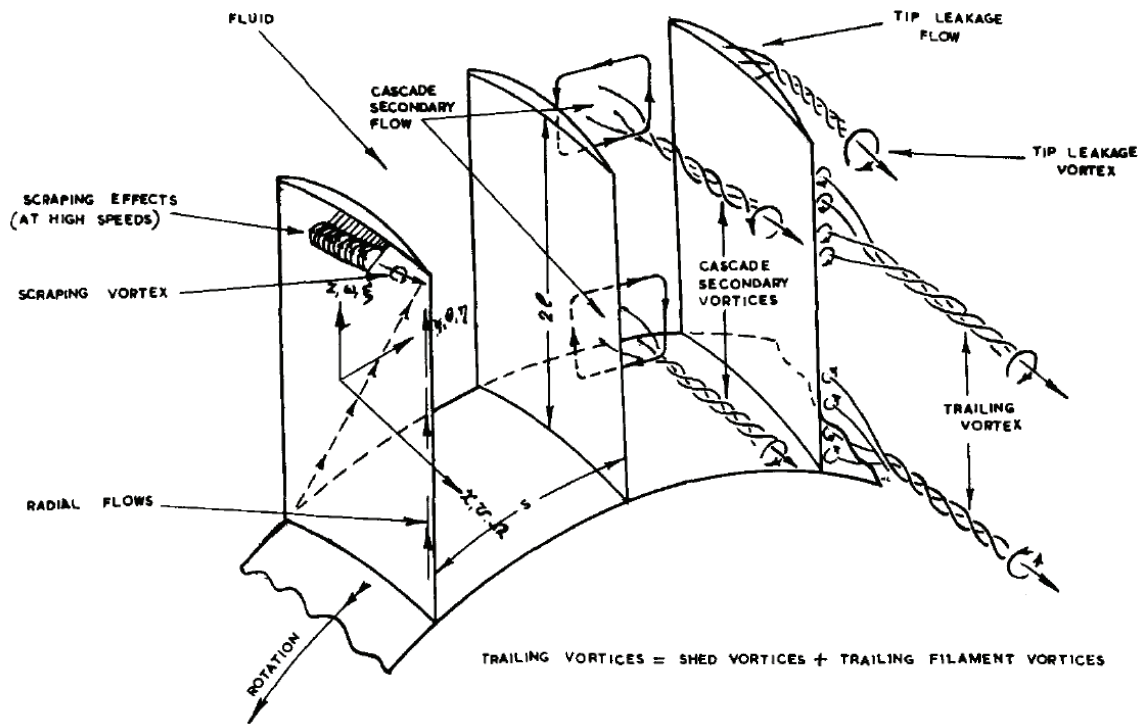


Figure I.1: Compressor secondary flows. Reproduced from Lakshminarayana & Horlock [1963].

the turbine's stators using Navier–Stokes analysis. Blade shapes are not the only component that can be optimized. Brennan et al. [2003] used CFD analysis to optimize the shape of a high pressure turbine's end walls for a predicted efficiency improvement of 0.4%.

Unfortunately, the wall clock time required to generate solutions for non-trivial geometries using CFD can be prohibitive to a designer, even with modern computers. Wall clock times can range from several days to on the order of a week or more

depending on the size of the mesh. This is undesirable when a large number of cases need to be run for a design process; for example, Brennan et al. [2003] needed upwards of 144 individual CFD simulations to optimize a turbine stage.

The use of reduced order models (ROM) based on proper orthogonal decomposition (POD) offers hope in reducing the overall computational time in a design study. POD based ROMs are used to reconstruct a solution from a set of optimal spatial and temporal modes extracted from a given ensemble of data, where the reconstructed solution can be at flow conditions other than what is defined by the ensemble. The ensemble of data can come from experimental data or numerical data from a full order model (FOM), such as a Navier–Stokes solver. POD has been previously applied to external flow design optimizations [LeGresley & Alonso, 2000], and to turbomachinery flow applications [Cizmas & Palacios, 2003; Brenner et al., 2013]. Cizmas et al. [2008] have shown that for a two-dimensional fluidized bed simulation that the POD based ROM was an order of magnitude quicker than the FOM used, and two orders of magnitude faster when acceleration techniques were introduced. This reduction in computational time means that a design point which may have taken several days to compute can now be done in hours.

As one would expect, the quality of the data ensemble will affect the accuracy of the ROM’s prediction. As such, this work is focused on improving the quality of results generated by an in-house Navier–Stokes solver for turbomachinery flows, known as UNS3D [Flitan & Cizmas, 2003]. Two turbomachinery flows are considered in this work. The first case is a turbine vane geometry tested by Goldman & McLallin [1977] and Goldman & Seasholtz [1982], referred to herein as the Goldman annular

cascade. The benefit of the vane geometry is that all of the modifications can be assessed without the additional complication of system rotation. To evaluate the solver for rotating turbomachinery cases, the solver was applied to the first rotor of a two-stage transonic fan developed at NASA Lewis, known as Rotor 67 [Urasek et al., 1979].

I.2. Background

Before the advent of computers made it possible to routinely solve the three-dimensional Euler or Navier–Stokes equations, engineers reduced the problem to two-dimensional domains on which simplified Euler equations could be solved. The earliest technique is known as the mean line method, in which the flow was computed between blade rows at the local mean radius. This method was developed by Howell [1945a,b] for compressors and for turbines by Ainley & Mathieson [1951]. The throughflow method was developed as an extension of the mean line method. Here, two-dimensional inviscid solutions were computed along streamlines in the hub-to-tip meridional plane concurrently with the inviscid flow on stream surfaces in the blade-to-blade direction [Wu, 1952]. As expected, the mean line and throughflow methods result in highly idealized flows.

A better approximation of the flow can be reached by solving the three-dimensional compressible Euler equations. The predictions made in locales for which boundary layer thicknesses are thin will be more accurate than those where thicker boundary layers or separation prevail [Cumpsty, 2004, p. 94]. Denton [1975] provided the early framework for finite volume based Euler solvers for both two- and three-dimensions.

For this work, Denton used a cell centered scheme with overlapping cells. This early work was not without its shortcomings. It suffered from a lack of entropy conservation and shock smearing, all of which Denton [1983] addressed through an improved implementation of the original method. Using a similar three-dimensional Euler solver, Chima & Strazisar [1983] were able to accurately predict shock structures, flow Mach numbers, and turning angles at different flow conditions for an experimentally tested transonic compressor rotor. It was found, as a consequence of neglecting viscous effects, that the Euler solver over-predicted the downstream total pressure and under-predicted the downstream relative Mach numbers [Chima & Strazisar, 1983].

Viscous effects must be included in the governing equations for the accurate prediction of secondary flows and the resultant losses. As a stopgap until solutions to the compressible Navier–Stokes equations became more feasible, Denton [1986] introduced viscous effects into a 3D Euler analysis by replacing the viscous terms of the momentum equations with an empirically determined distributed body force and neglecting the viscous terms found in the energy equation. Denton found that this method could provide a reasonable qualitative prediction of the viscous effects, but only if the necessary empirical input was chosen wisely.

Solutions of the Navier–Stokes equations, in particular the Reynolds-Averaged equations, provide the means to predict the complex flows found within turbomachines without the need for ad-hoc empirical input. Reynolds averaging allows the governing equations to be solved on reasonably sized grids by representing the turbulence in the form of the Reynolds-stress tensor [Blazek, 2005, p. 232]. The most

common approach to dealing with the Reynolds-stress tensor has been to apply the Boussinesq hypothesis, which introduces the eddy viscosity [Blazek, 2005, p. 235]. A multitude of turbulence models can then be chosen from to close the governing equations by determining the local value of the eddy viscosity.

Some of the earliest solutions to the three-dimensional Navier–Stokes equations for turbomachinery flows were done by Hah [1984], Hah & Leylek [1987], and Dawes [1988]. Hah [1984] used a finite volume scheme with higher order upwinding, and closed the governing equations using an algebraic Reynolds-stress model. The solver by Hah & Leylek [1987] wrote the governing equations using finite differences, utilizing a third-order accurate upwinding relaxation scheme and a two equation $\kappa - \epsilon$ turbulence model for closure. Both models required a pressure correction step to enforce mass conservation. Similar to Hah, Dawes [1988] wrote the governing equations using a finite volume formulation. However, Dawes [1988] differed by writing the equations in the relative frame of reference (rotor reference frame) and chose the mixing length Baldwin-Lomax turbulence model [Wilcox, 2010, p. 81].

Chima & Yokota [1990] and Chen et al. [1997] developed 3D Navier–Stokes solvers which were written in the relative frame of reference, but using velocity components in the absolute, or laboratory, reference frame. This approach allows for the steady-state solution of the governing equations to be found, since it is in a rotor relative reference frame, while simplifying the application of the boundary conditions. Both solvers used the thin-layer approximation for the Navier–Stokes equations with the Baldwin-Lomax turbulence model. The thin-layer approximation retains only the gradients that are in the wall normal direction, as they are assumed to domi-

nate [Blazek, 2005, p. 445]. Chima & Yokota [1990] used an explicit finite difference scheme to discretize the governing equations, while Chen et al. [1997] utilized an implicit finite volume scheme. Chima [1998b,a] improved upon the original solver by adding multi-block grid and multistage capabilities, and Chima & Liou [2003] saw to the addition of upwinding schemes and the addition of the $\kappa - \omega$ shear stress transport (SST) turbulence model by Menter [1993].

The flow within the Goldman annular cascade was predicted by Subramanian & Bozzola [1987] using a finite volume formulated 3D Navier–Stokes solver with closure by the Baldwin-Lomax turbulence model. The vane surface critical velocity ratios predicted by Subramanian & Bozzola were found to be a good match to the experimental values. Their solutions also showed the presence of the secondary flows spawned by the formation of a horseshoe vortex at the vane’s leading edge. Chima & Yokota [1990], using the previously discussed 3D Navier–Stokes solver, were also able to predict the development of a horseshoe vortex at vane leading edge near the endwalls. Their predicted vane surface pressures also showed excellent agreement with the measured values. However, Chima & Yokota found that the solver over-predicted losses. Arnone et al. [1993] also predicted the flow within the annular cascade using a 3D Navier–Stokes solver which employed implicit residual smoothing and the Baldwin-Lomax turbulence model. Arnone et al. were able to capture the development of the cascade’s secondary flows, and showed excellent qualitative agreement with the measure pressure distributions and downstream losses. Chima & Liou [2003] revisited the Goldman vane cascade with an improved version of the 1990 solver. Using newly implemented upwinding schemes and the Baldwin-Lomax

turbulence model, Chima & Liou found that upwinding the convective fluxes resulted in predictions of the downstream turning and wake profiles that match more closely to the measured values than the original central difference scheme.

Pierzga & Wood [1985] presented a detailed comparison between a 3D inviscid analysis, which included boundary layer effects by using an effective body, and the experimentally determined flow of Rotor 67. Overall, their analysis showed good agreement with the experimental data for all of the operating points simulated. Predicted relative Mach number contours at several spanwise locations were a qualitative match to the laser anemometry data for the majority of the operating points presented, indicating excellent shock capturing capability [Pierzga & Wood, 1985]. However, their analysis at the maximum flow (choked flow) operating point predicted the location of the shock on the upper span of the blade to be further downstream than what the experimental data showed. Subramanian & Bozzola [1987] showed that the correct shock location, at the max flow condition, could be predicted with the inclusion of viscous effects.

Additionally, Chima [1991], Hah & Reid [1992], Adamczyk et al. [1993], Jennions & Turner [1993], Arnone [1994], Arima et al. [1999], and Doi & Alonso [2002] have all simulated the flow within Rotor 67 using Navier–Stokes solvers. All of the authors found good agreement between their prediction and the experiment in terms of performance prediction, relative Mach number contours, and shock capturing across a range of operating points. Both Adamczyk et al. [1993] and Jennions & Turner [1993] examined the effect of the tip clearance height on the performance of the rotor, finding that increased tip clearances resulted in reduced performance due to higher

losses. Jennions & Turner found that simply doubling the height of the clearance could decrease the peak efficiency by approximately 1.2% and reduced the stall flow rate by roughly 12.5%. The literature showed that the maximum flow rate tended to be under-predicted by the viscous solvers. Reported flow rate differences ranged from as low as 0.48% [Arima et al., 1999] to as high as 1.32% [Arnone, 1994], with Chima [1991] and Jennions & Turner [1993] showing a 1.2% difference between prediction and experiment. Doi & Alonso [2002] suggested that reduced values in mass flow rate and total pressure may be due to a structural deformation, which naturally occurs due to rotational and aerodynamic forces, that is not modeled.

I.3. Original Contributions

This dissertation builds upon the previous work by Kim [2003] and Gargoloff [2007] with the goal of expanding the capabilities of the given computational tools, and developing new tools necessary for accurate computation of turbomachinery flows. In addition to improving the computational tools, the intention of this work was also to further build the knowledge base regarding the computation of turbomachinery flows.

A grid generation tool specifically designed for turbomachinery geometries was developed in collaboration with Matula [2014]. The grid generator uses a structured multi-block grid system to define the turbomachinery flow path and geometry, including the modelling of tip clearance flows. A turbomachinery solution post processor was also developed to work in conjunction with the grids generated by the new grid generation tool. The post processing tool has the capability to generate appropri-

ately averaged flow quantities at arbitrary axial locations, as well as extracting other extracting other pertinent flow information.

A radially varying outlet boundary condition was implemented to facilitate more accurate predictions of turbomachinery flows. The outlet static pressure was defined using the axisymmetric radial momentum equation to allow for a natural spanwise variation in pressure, as opposed to a uniform distribution. The turbulence model, the $\kappa-\omega$ SST model, was also updated in this work. In addition to updated the model constants to the most recent values, the definition of the eddy viscosity was modified to make it frame invariant. A modification to the Multi-directional Limiting Process (MLP) limiter from the literature was implemented to improve solution convergence when using said limiter. The modification to the MLP limiter was also applied to the Venkatakrisnan limiter. To the author’s knowledge, this is the first time the MLP limiter modification has been applied to the Venkatakrisnan limiter.

A pair of turbomachinery cases were considered to evaluate the accuracy of the improved flow solver. In the first case, a turbine vane was used to test the implementation of the new outlet boundary. In addition, the turbine vane geometry was used to examine what effect the inlet turbulence conditions and the under-relaxation of the turbulence equations have on the overall solution.

In the second case, the performance characteristics of a transonic fan operating at the design wheel speed were computed for both inviscid and turbulent flow conditions. Two novel rubbing configurations¹ and a realistic tip clearance flow configuration were considered for the turbulent simulations. Both the merits and

¹The rotor tip is allowed to “rub” against the outer endwall

disadvantages of the two rubbing configurations were explored. The effect of the choice of solution limiter on the overall solution was examined for both the inviscid and turbulent clearance flow cases.

I.4. Dissertation Outline

Chapter II presents the governing equations of the flow, and discusses Reynolds averaging and equation non-dimensionalization. The first part of Chapter III presents the new turbomachinery grid generator. Following the topic of grid generation in Chapter III, the flow solver, the discretized equations, and the boundary conditions are presented. In Chapter IV, the results from the validation of the solver improvements are presented. Inviscid results for a convergent-divergent nozzle, both purely subsonic and with a shock, and a NACA 0012 airfoil in supersonic freestream are presented to compare and contrast the effectiveness of the higher-order limiters tested. Turbulent flat plate results are then given to validate the improvements to the turbulence model. Chapter V compares the predicted turbulent flow through an annular turbine vane row against experimental data. Chapters VI and VII present inviscid and turbulent results, respectively, for a transonic fan with comparison to both measured and simulated data from the literature. The conclusions drawn from the numerical results presented herein, and recommendations for future work are given in Chapters VIII and IX, respectively.

CHAPTER II

PHYSICAL MODEL

This chapter opens with a statement of the equations which describe the motion of a viscous fluid. Next, averaging techniques used to approximate turbulent flows are presented. This is followed by a description of the turbulence model used to close the system of averaged equations. Finally the nondimensionalization of the governing equations is discussed.

II.1. Navier–Stokes Equations

Viscous fluid flow is described by three basic equations: conservation of mass, conservation of momentum, and conservation of energy. The conservation of momentum equations are also known as the Navier–Stokes equations. However this moniker is also used to group the three sets of conservation equations under one name. The term Navier–Stokes equations will be used henceforth in this dissertation to describe the full set of governing equations. The equations, assuming Einstein notation and in the absence of external forces, are shown below.

$$\begin{aligned}\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_i} &= 0 \\ \frac{\partial (\rho u_i)}{\partial t} + \frac{\partial (\rho u_j u_i)}{\partial x_j} &= \rho g_i - \frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \\ \frac{\partial (\rho E)}{\partial t} + \frac{\partial (\rho H u_i)}{\partial x_i} &= \frac{\partial}{\partial x_j} \left(k \frac{\partial T}{\partial x_j} + u_i \tau_{ij} \right) + \dot{q}\end{aligned}\tag{2.1}$$

The gravitational force, g_i , and the heat source, \dot{q} , are included in the definition of the Navier–Stokes equations for completeness, but they will be ignored herein.

For a Newtonian fluid the components of the viscous stress tensor are given by

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) + \lambda \frac{\partial u_k}{\partial x_k} \delta_{ij}. \quad (2.2)$$

The bulk viscosity, λ , is defined using Stokes' hypothesis, which is given by

$$\lambda + \frac{2}{3}\mu = 0. \quad (2.3)$$

Substituting (2.3) into (2.2) yields the below equation, which defines the form of the viscous stress tensor components used in this dissertation.

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \quad (2.4)$$

The total energy per unit mass and the total enthalpy are respectively given by the following formulae.

$$E = e + \frac{u_i u_i}{2} \quad (2.5)$$

$$H = h + \frac{u_i u_i}{2} \quad (2.6)$$

The working fluid is assumed to be a calorically perfect gas. The thermodynamic properties of a perfect gas are related by the equation of state, given by

$$p = \rho R T. \quad (2.7)$$

R in the above equation is the specific gas constant, and for dry air has a value of $287.16 \text{ J kg}^{-1}\text{K}^{-1}$. The enthalpy and internal energy per unit mass are defined as $h = c_p T$ and $e = c_v T$, respectively.

The Sutherland equation, given by (2.8), is used to define the dynamic viscosity as a function of temperature. The constants found in the Sutherland equation are, for air and in SI units, $C = 1.458 \times 10^{-6} \frac{\text{kg}}{\text{ms}\sqrt{\text{K}}}$ and $S = 110.4 \text{ K}$.

$$\mu = \frac{CT^{3/2}}{T + S} \quad (2.8)$$

Equation 2.9 defines the thermal conductivity coefficient, where Pr is the Prandtl number. The Prandtl number is assumed to be constant throughout the flow, and has the value $Pr = 0.72$ for air.

$$k = c_p \frac{\mu}{Pr} \quad (2.9)$$

II.2. Favre- and Reynolds–Averaged Navier–Stokes Equations

Most flows of interest to researchers and engineers are turbulent. Even with modern improvements in computing power, the direct numerical simulation of turbulent flows represents an intractable problem to all but idealized low Reynolds number cases. The enormity of the problem is the result of the requirement to sufficiently resolve the length and time scales, resulting in grid sizes that scale with $Re^{9/4}$ and CPU times with Re^3 [Blazek, 2005, p. 227]. An approximation of the turbulent flow is sought for the computation of flows with realistic Reynolds numbers.

Reynolds averaging is the method used to approximate incompressible turbulent flows. The averaging is accomplished by decomposing the state variables into mean and fluctuating components. The governing equations are then solved for the the mean quantities. The Reynolds decomposition of a general scalar variable is given by

$$\mathfrak{N} = \bar{\mathfrak{N}} + \mathfrak{N}' , \quad (2.10)$$

where $\bar{\aleph}$ is the mean component and \aleph' is the fluctuating component. The mean component is defined by a time average, as shown by (2.11). The period, T , over which the quantity is averaged is large compared to the time scale of the turbulence.

$$\bar{\aleph} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{t_0}^{t_0+T} \aleph dt \quad (2.11)$$

For compressible flows, Favre averaging is utilized for some of the variables instead of Reynolds averaging. Doing so removes additional terms that arise due to density fluctuations. Similarly to Reynolds averaging, the variables are split into a mean and a fluctuating component, as shown in (2.12). Now the averaged quantity, $\tilde{\aleph}$, is a density weighted average, as defined by (2.13).

$$\aleph = \tilde{\aleph} + \aleph'' \quad (2.12)$$

$$\tilde{\aleph} = \frac{1}{\bar{\rho}} \lim_{T \rightarrow \infty} \frac{1}{T} \int_{t_0}^{t_0+T} \rho \aleph dt \quad (2.13)$$

Applying Reynolds averaging to density and pressure, and Favre averaging to the remaining variables in (2.1) results in [Blazek, 2005, p. 234]

$$\begin{aligned} \frac{\partial \bar{\rho}}{\partial t} + \frac{\partial (\bar{\rho} \tilde{u}_i)}{\partial x_i} &= 0 \\ \frac{\partial (\bar{\rho} \tilde{u}_i)}{\partial t} + \frac{\partial (\bar{\rho} \tilde{u}_j \tilde{u}_i)}{\partial x_j} &= -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \left(\tilde{\tau}_{ij} - \widetilde{\bar{\rho} u_i'' u_j''} \right) \\ \frac{\partial (\bar{\rho} \tilde{E})}{\partial t} + \frac{\partial (\bar{\rho} \tilde{H} \tilde{u}_i)}{\partial x_i} &= \frac{\partial}{\partial x_j} \left(k \frac{\partial \tilde{T}}{\partial x_j} - \widetilde{\bar{\rho} u_j'' h''} + \tilde{u}_i \left(\tilde{\tau}_{ij} + \widetilde{\bar{\rho} u_i'' u_j''} \right) \right), \end{aligned} \quad (2.14)$$

which are known as the Favre- and Reynolds-Averaged Navier–Stokes equations.

Herein the equations will be referred to as the RANS equations for simplicity.

The viscous stress tensor in (2.14) is defined using (2.4) but with Favre-averaged velocity components. The definitions for the total energy and total enthalpy are now given, respectively, by [Wilcox, 2010, p. 249]

$$\begin{aligned}\tilde{E} &= \tilde{e} + \frac{1}{2}\tilde{u}_i\tilde{u}_i + \kappa \\ \tilde{H} &= \tilde{h} + \frac{1}{2}\tilde{u}_i\tilde{u}_i + \kappa,\end{aligned}\tag{2.15}$$

where $\kappa = \widetilde{u_i''u_i''}/2$ is the Favre-averaged turbulent kinetic energy.

Favre and Reynolds averaging introduces two additional terms to the governing equations. They are the Favre-averaged Reynolds stress tensor, $-\widetilde{\bar{\rho}u_i''u_j''}$, and the turbulent heat flux vector, $\widetilde{\bar{\rho}u_j''h''}$. This creates an underdetermined system of equations. As a result, approximations are introduced to close the equations.

The Boussinesq eddy-viscosity hypothesis is used to approximate the Favre-averaged Reynolds stress tensor. The Boussinesq approximation gives the Reynolds stress term to be

$$\tau_{ij}^F = -\widetilde{\bar{\rho}u_i''u_j''} = \mu_T \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} - \frac{2}{3} \frac{\partial \tilde{u}_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \bar{\rho} \kappa \delta_{ij},\tag{2.16}$$

where μ_T is the eddy viscosity. The eddy viscosity is computed by the turbulence model. The last term in (2.16) is required to provide the proper trace of τ_{ij}^F ; however, it is often neglected and will be done so here [Blazek, 2005, p. 236].

The typical approximation for the turbulent heat flux vector is based on the Reynolds analogy, and is given by

$$\widetilde{\bar{\rho}u_j''h''} = -k_T \frac{\partial \tilde{T}}{\partial x_j},\tag{2.17}$$

where k_T is the turbulent thermal conductivity coefficient. The turbulent thermal

conductivity coefficient is given by

$$k_T = c_p \frac{\mu_T}{Pr_T}, \quad (2.18)$$

where Pr_T is the turbulent Prandtl number. As with the Prandtl number, the turbulent Prandtl number is assumed to be constant over the flow field. The turbulent Prandtl number has a value of 0.9 for air.

Equation 2.14 can be rewritten, using (2.9) and (2.16)-(2.18), as

$$\begin{aligned} \frac{\partial \bar{\rho}}{\partial t} + \frac{\partial (\bar{\rho} \tilde{u}_i)}{\partial x_i} &= 0 \\ \frac{\partial (\bar{\rho} \tilde{u}_i)}{\partial t} + \frac{\partial (\bar{\rho} \tilde{u}_j \tilde{u}_i)}{\partial x_j} &= -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial \tilde{\tau}_{ij}}{\partial x_j} \\ \frac{\partial (\bar{\rho} \tilde{E})}{\partial t} + \frac{\partial (\bar{\rho} \tilde{H} \tilde{u}_i)}{\partial x_i} &= \frac{\partial}{\partial x_j} \left[c_p \left(\frac{\mu}{Pr} + \frac{\mu_T}{Pr_T} \right) \frac{\partial \tilde{T}}{\partial x_j} + \tilde{u}_i \tilde{\tau}_{ij} \right] \end{aligned} \quad (2.19)$$

with

$$\tilde{\tau}_{ij} = (\mu + \mu_T) \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} - \frac{2}{3} \frac{\partial \tilde{u}_k}{\partial x_k} \delta_{ij} \right). \quad (2.20)$$

Over-bars and tildes are dropped from the equations herein for convenience.

II.3. Turbulence Modeling

A turbulence model is needed to compute the eddy viscosity introduced into the RANS equations by the Boussinesq hypothesis. Two equation models provide a complete model in that the turbulence can be modeled without prior knowledge of the turbulent structure [Wilcox, 2010, pg. 122]. The most widely used two equation models are the $\kappa - \epsilon$ and $\kappa - \omega$ models. Each model is not without drawbacks, which are succinctly summarized by Menter [1993] and will not be repeated here.

The model chosen for this work is the two equation $\kappa - \omega$ Shear Stress Transport

(SST) eddy viscosity model by Menter [1993, 1994]. The SST model blends together the standard $\kappa - \epsilon$ and $\kappa - \omega$ models to utilize each model's strengths without suffering their weaknesses. The SST model transitions from the $\kappa - \omega$ model in the inner region of the boundary layer to the $\kappa - \epsilon$ model in the outer wake region. The SST model also improves prediction in regions of adverse pressure gradients by accounting for the transport of turbulent shear stress in the formulation of the eddy viscosity. Improvements to the original SST model by Hellsten [1998] and by Menter et al. [2003] have been used in this work.¹ The transport equations for κ and ω are given below.

$$\frac{\partial(\rho\kappa)}{\partial t} + \frac{\partial(\rho u_j \kappa)}{\partial x_j} = P_\kappa - \beta^* \rho \omega \kappa + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_\kappa \mu_T) \frac{\partial \kappa}{\partial x_j} \right] \quad (2.21)$$

$$\begin{aligned} \frac{\partial(\rho\omega)}{\partial t} + \frac{\partial(\rho u_j \omega)}{\partial x_j} = & \frac{\rho\alpha}{\mu_T} P_\kappa - \beta \rho \omega^2 + \frac{\partial}{\partial x_j} \left[(\mu + \sigma_\omega \mu_T) \frac{\partial \omega}{\partial x_j} \right] \\ & + 2\rho(1 - F_1) \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial \kappa}{\partial x_j} \frac{\partial \omega}{\partial x_j} \end{aligned} \quad (2.22)$$

From left to right, the terms on the right hand of the above equations represent production, dissipation, and diffusion. The extra term on the right hand side of (2.22) defines the cross diffusion. The production term in both equations is given by

$$P_\kappa = \tau_{ij}^F \frac{\partial u_i}{\partial x_j} = \left(\mu_T \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho \kappa \delta_{ij} \right) \frac{\partial u_i}{\partial x_j}. \quad (2.23)$$

¹A typographical error exists in [Menter et al., 2003] in the definition of the production term of the ω equation, which was corrected in [Menter, 2009].

The constants of the model are computed as a blended value of the $\kappa - \omega$ and $\kappa - \epsilon$ constants. The constants are computed using

$$\aleph = F_1 \aleph_1 + (1 - F_1) \aleph_2,$$

where \aleph_1 is any $\kappa - \omega$ constant, \aleph_2 is any $\kappa - \epsilon$ constant, and \aleph is the corresponding blended constant. The constants for the base models are given in Table II.1. The values given for α_1 and α_2 are taken from Menter et al. [2003] and differ slightly from the original definitions [Menter, 1993, 1994].

Table II.1: $\kappa - \omega$ (1) and $\kappa - \epsilon$ (2) model constants.

$\sigma_{K1} = 0.850$	$\sigma_{K2} = 1.000$
$\sigma_{\omega1} = 0.500$	$\sigma_{\omega2} = 0.856$
$\beta_1 = 0.075$	$\beta_2 = 0.0828$
$\alpha_1 = 5/9$	$\alpha_2 = 0.440$
$\beta^* = 0.09$	

The function F_1 is computed from

$$F_1 = \tanh(\arg_1^4) \tag{2.24}$$

with

$$\arg_1 = \min \left[\max \left(\frac{\sqrt{\kappa}}{0.09\omega y}, \frac{500\mu}{y^2\rho\omega} \right), \frac{4\rho\sigma_{\omega2}\kappa}{CD_{\kappa\omega}y} \right], \tag{2.25}$$

where y is the distance to the nearest wall and $CD_{\kappa\omega}$ is given by

$$CD_{\kappa\omega} = \max \left(2\rho\sigma_{\omega2} \frac{1}{\omega} \frac{\partial\kappa}{\partial x_j} \frac{\partial\omega}{\partial x_j}, 10^{-10} \right). \tag{2.26}$$

The definition of $CD_{\kappa\omega}$ uses the most recent value for the limiter by Menter et al. [2003].

The eddy viscosity for the SST model is given by [Hellsten, 1998; Menter et al., 2003]

$$\mu_T = \frac{a_1 \rho \kappa}{\max(a_1 \omega, |S_{ij}| F_2)}, \quad (2.27)$$

where $a_1 = 0.31$, and $|S_{ij}| = \sqrt{S_{ij} S_{ij}}$ is the magnitude of the strain rate tensor,

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right). \quad (2.28)$$

The magnitude of the strain rate tensor is used such that the formulation of the eddy viscosity is frame invariant [Hellsten, 1998]. The function F_2 is defined as

$$F_2 = \tanh(\arg_2) \quad (2.29)$$

with

$$\arg_2 = \max \left(2 \frac{\sqrt{\kappa}}{0.09 \omega y}, \frac{500 \mu}{y^2 \rho \omega} \right). \quad (2.30)$$

II.4. Equation Nondimensionalization

The governing equations presented so far have been written in terms of dimensional variables. As a result, the solutions of the equations will be dependent on the choice of units selected by the end user. These dimensional results may not be readily comparable to other results found in the literature. Nondimensionalizing the governing equations results in solutions that are in terms of useful and easily referenced nondimensional parameters such as Mach number or Reynolds number.

In addition to providing easily compared results, nondimensionalization also reduces the computational burden by shrinking the differences in magnitude between

the variables. For instance, at sea level, atmospheric pressure is five orders of magnitude larger than density and three orders of magnitude greater than temperature. With nondimensionalization, the magnitudes of the three variables of state become roughly the same.

The nondimensionalization used in this work is the same used by Kim [2003]. The dimensional variable definitions for the variables found in the Navier–Stokes equations are given below.

$$\begin{aligned}
\check{x}_i &= \frac{x_i}{l} & \check{t} &= \frac{tc_\infty}{l} & \check{u}_i &= \frac{u_i}{c_\infty} & \check{\rho} &= \frac{\rho}{\rho_\infty} \\
\check{p} &= \frac{p}{\rho_\infty c_\infty^2} & \check{E} &= \frac{E}{c_\infty^2} & \check{H} &= \frac{H}{c_\infty^2} & \check{T} &= \frac{T}{T_\infty} \\
& & \check{\mu} &= \frac{\mu}{\mu_\infty} & \check{\mu}_T &= \frac{\mu_T}{\mu_\infty} & &
\end{aligned} \tag{2.31}$$

II.4.1. Nondimensional RANS Equations

To arrive at the nondimensional form of the RANS equations, the dimensional variables in (2.19) and (2.20) are substituted for their nondimensional form using (2.31). The reference variables introduced by (2.31) are then collected on the right hand side of the equation. After cancellations and grouping of the remaining reference variables, the nondimensional RANS equations are given by

$$\begin{aligned}
\frac{\partial \check{\rho}}{\partial \check{t}} + \frac{\partial (\check{\rho} \check{u}_i)}{\partial \check{x}_i} &= 0 \\
\frac{\partial (\check{\rho} \check{u}_i)}{\partial \check{t}} + \frac{\partial (\check{\rho} \check{u}_j \check{u}_i)}{\partial \check{x}_j} &= -\frac{\partial \check{p}}{\partial \check{x}_i} + \frac{\partial \check{\tau}_{ij}}{\partial \check{x}_j} \\
\frac{\partial (\check{\rho} \check{E})}{\partial \check{t}} + \frac{\partial (\check{\rho} \check{H} \check{u}_i)}{\partial \check{x}_i} &= \frac{\partial}{\partial \check{x}_j} \left[\frac{\check{c}_p}{Re_\infty} \left(\frac{\check{\mu}}{Pr} + \frac{\check{\mu}_T}{Pr_T} \right) \frac{\partial \check{T}}{\partial \check{x}_j} + \check{u}_i \check{\tau}_{ij} \right],
\end{aligned} \tag{2.32}$$

where

$$\check{\tau}_{ij} = \frac{\check{\mu} + \check{\mu}_T}{Re_\infty} \left(\frac{\partial \check{u}_i}{\partial \check{x}_j} + \frac{\partial \check{u}_j}{\partial \check{x}_i} - \frac{2}{3} \frac{\partial \check{u}_k}{\partial \check{x}_k} \delta_{ij} \right). \tag{2.33}$$

The above nondimensional equations look nearly identical to the dimensional equations, except for the presence of a reference Reynolds number,

$$Re_\infty = \frac{\rho_\infty c_\infty l}{\mu_\infty}. \quad (2.34)$$

The nondimensional specific heat term in the energy equation is given by

$$\check{c}_p = \frac{1}{\gamma - 1},$$

and is derived by using the definitions $c_\infty = \sqrt{\gamma RT_\infty}$ and $c_p = \gamma R / (\gamma - 1)$.

II.4.2. Nondimensional $\kappa - \omega$ SST Equations

The nondimensionalization of the turbulence model equations is conducted in the same manner as that of the RANS equations. The nondimensional turbulence model variables are given by

$$\check{\kappa} = \frac{\kappa}{c_\infty^2} \quad \check{\omega} = \frac{\omega l}{c_\infty}. \quad (2.35)$$

Substituting (2.31) and (2.35) into (2.21)–(2.23) results in the nondimensional turbulence model equations:

$$\frac{\partial(\check{\rho}\check{\kappa})}{\partial\check{t}} + \frac{\partial(\check{\rho}\check{u}_j\check{\kappa})}{\partial\check{x}_j} = \check{P}_\kappa - \beta^*\check{\rho}\check{\omega}\check{\kappa} + \frac{1}{Re_\infty} \frac{\partial}{\partial\check{x}_j} \left[(\check{\mu} + \sigma_\kappa\check{\mu}_T) \frac{\partial\check{\kappa}}{\partial\check{x}_j} \right] \quad (2.36)$$

$$\begin{aligned} \frac{\partial(\check{\rho}\check{\omega})}{\partial\check{t}} + \frac{\partial(\check{\rho}\check{u}_j\check{\omega})}{\partial\check{x}_j} &= \frac{\check{\rho}\alpha Re_\infty}{\check{\mu}_T} \check{P}_\kappa - \beta\check{\rho}\check{\omega}^2 + \frac{1}{Re_\infty} \frac{\partial}{\partial\check{x}_j} \left[(\check{\mu} + \sigma_\omega\check{\mu}_T) \frac{\partial\check{\omega}}{\partial\check{x}_j} \right] \\ &+ 2\check{\rho}(1 - F_1) \sigma_{\omega 2} \frac{1}{\check{\omega}} \frac{\partial\check{\kappa}}{\partial\check{x}_j} \frac{\partial\check{\omega}}{\partial\check{x}_j}, \end{aligned} \quad (2.37)$$

with

$$\check{P}_\kappa = \left(\frac{\check{\mu}_T}{Re_\infty} \left(\frac{\partial \check{u}_i}{\partial \check{x}_j} + \frac{\partial \check{u}_j}{\partial \check{x}_i} - \frac{2}{3} \frac{\partial \check{u}_k}{\partial \check{x}_k} \delta_{ij} \right) - \frac{2}{3} \check{\rho} \check{\kappa} \delta_{ij} \right) \frac{\partial \check{u}_i}{\partial \check{x}_j}. \quad (2.38)$$

As with the RANS equations, the nondimensional equations are nearly identical to their dimensional form save for the presence of the reference Reynolds number, given by (2.34). All of the model constants are unchanged. The nondimensional eddy viscosity is given by

$$\frac{\check{\mu}_T}{Re_\infty} = \frac{a_1 \check{\rho} \check{\kappa}}{\max(a_1 \check{\omega}, |\check{S}_{ij}| F_2)}, \quad (2.39)$$

with $|\check{S}_{ij}| = \sqrt{\check{S}_{ij} \check{S}_{ij}}$ and $\check{S}_{ij} = \frac{1}{2} (\partial \check{u}_i / \partial \check{x}_j + \partial \check{u}_j / \partial \check{x}_i)$. The blending functions are still given by (2.24) and (2.29). The nondimensional blending function arguments are given below.

$$\arg_1 = \min \left[\max \left(\frac{\sqrt{\check{\kappa}}}{0.09 \check{\omega} \check{y}}, \frac{1}{Re_\infty} \frac{500 \check{\mu}}{\check{y}^2 \check{\rho} \check{\omega}} \right), \frac{4 \check{\rho} \sigma_\omega \check{\kappa}}{CD_{\kappa\omega} \check{y}} \right] \quad (2.40)$$

$$CD_{\kappa\omega} = \max \left(2 \check{\rho} \sigma_\omega \frac{1}{\check{\omega}} \frac{\partial \check{\kappa}}{\partial \check{x}_j} \frac{\partial \check{\omega}}{\partial \check{x}_j}, 10^{-10} \right) \quad (2.41)$$

$$\arg_2 = \max \left(2 \frac{\sqrt{\check{\kappa}}}{0.09 \check{\omega} \check{y}}, \frac{1}{Re_\infty} \frac{500 \check{\mu}}{\check{y}^2 \check{\rho} \check{\omega}} \right) \quad (2.42)$$

CHAPTER III

NUMERICAL METHODS

This chapter begins with a description of the mesh generation process used to construct grids suitable for turbomachinery simulations. Following this, the numerical methods used to solve the governing equations are described. The parallelization of the numerical methods are then briefly described. Finally, the averaging techniques used to compute the turbomachinery performance are described.

III.1. Turbomachinery Grid Generation

As will be shown later in this chapter, the governing equations are to be solved at discrete points within the domain of interest. A grid which accurately captures the geometry of the domain while maintaining an acceptable level of grid quality is therefore required. The purpose of this section is to detail how that grid is generated for annular turbomachinery domains.

An in-house turbomachinery grid generator, known as TGG, was developed to construct the computational grids to be used in this work. TGG is a multi-block grid generator which can be used for both rotors and stators. The grid generator can also generate clearance grids to accurately capture the region between blade tip and the adjacent casing. The steps of the grid generation process are:

1. Construct meridional grid
2. Intersect rotor/stator geometry with meridional grid

3. Generate blade-to-blade grids to define the blade passage
4. Optionally generate set of blade tip clearance grids
5. Optionally generate an inlet extension H-grid
6. Optionally generate an outlet extension H-grid
7. Construct grid relationships and output mesh

The following subsections describe the above steps in the grid generation process. A description of the grid generator's input file can be found in Appendix F.

III.1.1. Meridional Grid Generation

The meridional grid is a two dimensional grid in (x, r) , with index dimensions (I2D, K2D), that defines the flow path between the inner and outer walls. The meridional mesh also serves as the foundation for which the blade-to-blade grids are built upon. Blade-to-blade grids are constructed on each spanwise layer of the meridional mesh. To save computational time, the number of meridional layers should be much less than the desired number final grid layers.

The definitions of the flow path end walls are read as input to the grid generator. A description of the file format used to input the paths can be found in Appendix F. The streamwise boundaries of the meridional mesh domain are set by user defined points. These points are used to define edges, which are then intersected with the given end wall profiles, as shown in Fig. III.1. Cubic spline interpolation is used to determine the points of intersection.

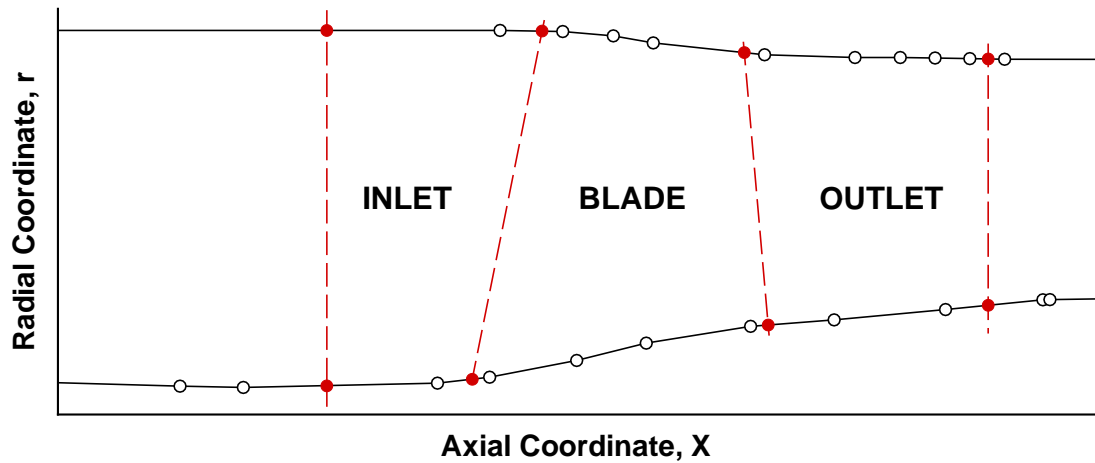


Figure III.1: Defining the meridional grid with user prescribed boundaries. “o” define endwall profiles, “—” are user defined grid boundaries, and “•” are intersected boundary points.

Once the boundary points have been intersected with the given profiles, the primary meridional mesh is then constructed in the region labeled “Blade” in Fig. III.1. First, the definitions of the inner and outer wall profiles are refined by adding equally spaced points between the intersected boundary nodes using a cubic spline. Cubic Bezier curves are used to connect the hub and tip wall nodes between the inlet and outlet boundaries. The cubic Bezier curves are defined such that the curves are perpendicular at the endwalls. A linear Bezier curve is used at the inlet and outlet boundaries.

The Bezier curves used to define the grid are computed with equally spaced points from the hub wall to the tip wall. The points can optionally be redistributed so that they are clustered near the walls. Clustering near the endwalls is recommended when constructing grids with viscous spacings.

Figure III.2 shows a typical meridional mesh for the “Blade” region. A vector of arc length distribution, $\vec{S}_k = [s_1, s_2, \dots, s_{I2D}]^T$, is computed for each grid layer of constant k-index. The arc length vectors are stored for later use in constructing the blade-to-blade grids. Meridional grids for the “Inlet” and “Outlet” regions in Fig. III.1 are only constructed later if the optional inlet and outlet grids are desired.

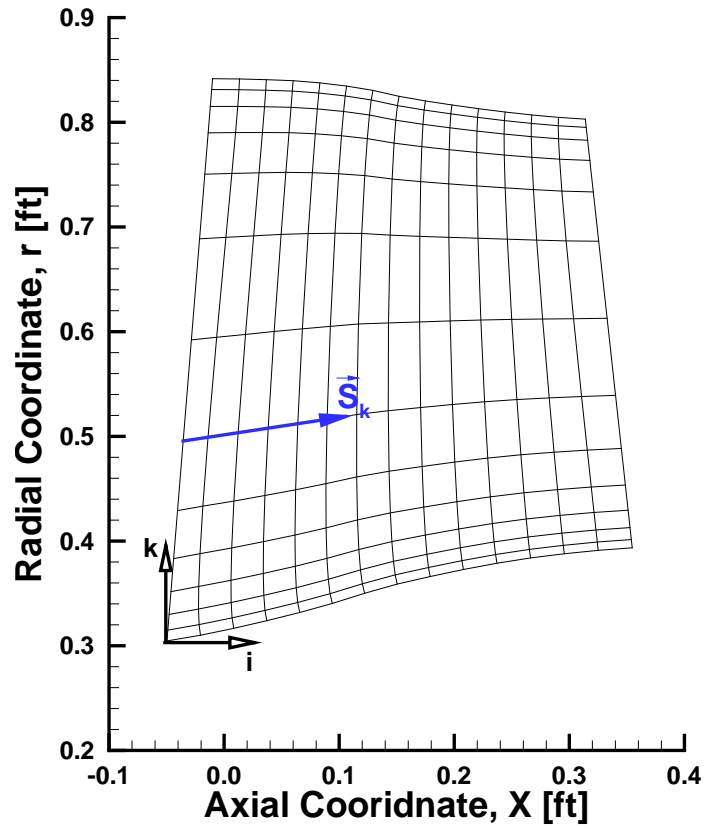


Figure III.2: Typical blade-to-blade meridional mesh showing the indicial coordinate system and spanwise arc length distribution vector, \vec{S}_k .

III.1.2. Blade Surface Definition

The coordinates defining the blade shape are given as input to the grid generator in cylindrical coordinates. The coordinates must fully define a number of spanwise sections of the blade, including the leading and trailing edges. The coordinates start at the trailing edge and wrap clockwise around the blade, with the trailing edge point being repeated. It is assumed that the sections are stacked from hub to tip. A description of the blade surface definition file can be found in Appendix F.

After input, the point distribution of each section is modified to have the desired number of points and grid spacing as set by the user. Then the blade surface is intersected with each layer of the meridional mesh to find the local blade cross section. Cubic spline interpolation is used to compute the intersection. The resulting cylindrical coordinates of the airfoil and the arc length of the meridional layer at each point of intersection are stored for later use.

III.1.3. Blade-to-Blade Grids

A multi-block configuration consisting of one O-grid and four H-grids (O-4H) was chosen for the blade-to-blade grids. Figure III.3 shows the layout of the grid blocks. Blocks four and five in Fig. III.3 are generated as a single block before being split in two.

The base three-dimensional blade-to-blade grids are built by generating two-dimensional grids at each spanwise layer of the meridional mesh and then stacking them together. As a result, a coordinate transformation is required for the construction of the two-dimensional grids. The reduced coordinates chosen for the two-

dimensional grids are $(s, \bar{r}\theta)$ coordinates. The variable s is the meridional arc length, \bar{r} is the average of the leading and trailing edge radii, and θ is the cylindrical coordinate. The meridional arc length was chosen to allow for an easy conversion back to cylindrical coordinates using the meridional mesh.

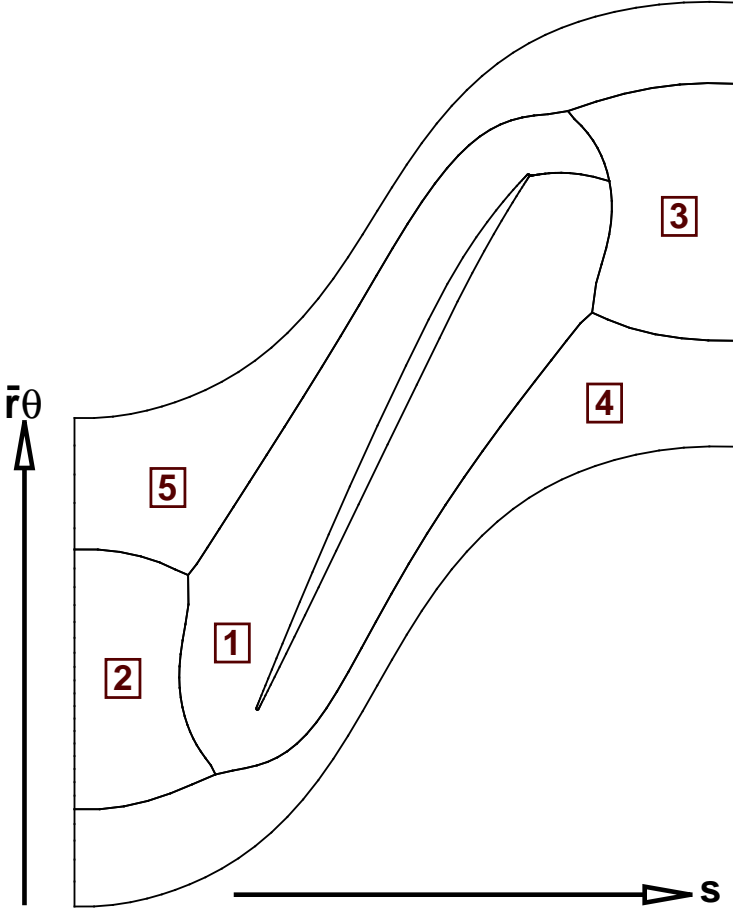


Figure III.3: O-4H multi-block blade-to-blade grid layout with block numbers.

The boundaries for each two-dimensional mesh are set by the boundaries of the meridional mesh at the current spanwise location, the intersected blade geom-

etry, and the user specified pitch. The initial H-grids are generated algebraically and a hyperbolic O-grid is constructed around the blade section. An iterative elliptic smoothing algorithm developed by Matula [2014] is used to ensure smooth transitions between the grid blocks and maintain an acceptable level of grid quality. The smoothing algorithm is based upon the previous work by Thompson et al. [1974] and Yamamoto & Engel [1997]. Each grid block is smoothed individually during each smoothing iteration. Internal block boundaries are handled by using the neighboring block's internal nodes as boundary conditions for the block being smoothed.

Once the grids have been smoothed, the meridional mesh is used to transform the coordinates back into cylindrical coordinates. Functions for (x, r) in terms of s are known from the definition of the meridional mesh. Using these functions with cubic spline interpolation allows for the calculation of the base blade-to-blade grid's (x, r) coordinates. The θ coordinate is computed by simply dividing by the local \bar{r} .

Up to this point, grids have only been constructed on the meridional layers. However, the actual grid will contain many more layers with a desired spanwise grid clustering. To accommodate this, cubic spline interpolation is used to generate the full set of grid layers using the base blade-to-blade grid as the background mesh.

III.1.4. Blade Tip Clearance Grids

Optionally, a set of grids can be generated to allow for the flow to be solved in the clearance region between the blade tip and the end wall. A multi-block grid configuration was chosen to allow for excellent matching with the surrounding blade-to-blade O-grid, while maintaining good grid quality. Figure III.4 shows a portion

of a single layer of the chosen O-H grid configuration near the leading and trailing edges.

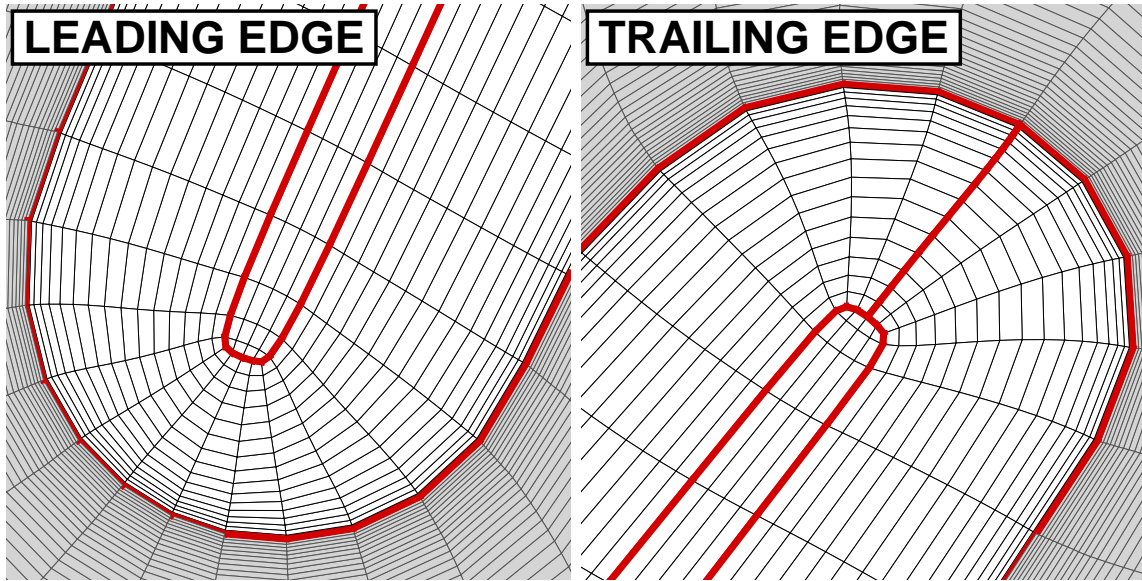


Figure III.4: Blade tip clearance grid (white shaded region) leading and trailing edge detail showing the block boundaries (red edges).

Unlike the blade-to-blade grids that were constructed on a subset of layers, clearance grid layers are built at each layer of the blade-to-blade grid in the clearance region. Spanwise clustering of the blade-to-blade grid layers is used to define the thickness of the clearance region and the distribution of the clearance grid layers. Figure III.5 shows an example of the spanwise clustering of the blade-to-blade and clearance grids. Consult Appendix F for details on specifying the grid distribution and clearance thickness.

The inner-most layer of the blade-to-blade O-grid serves as the boundary for

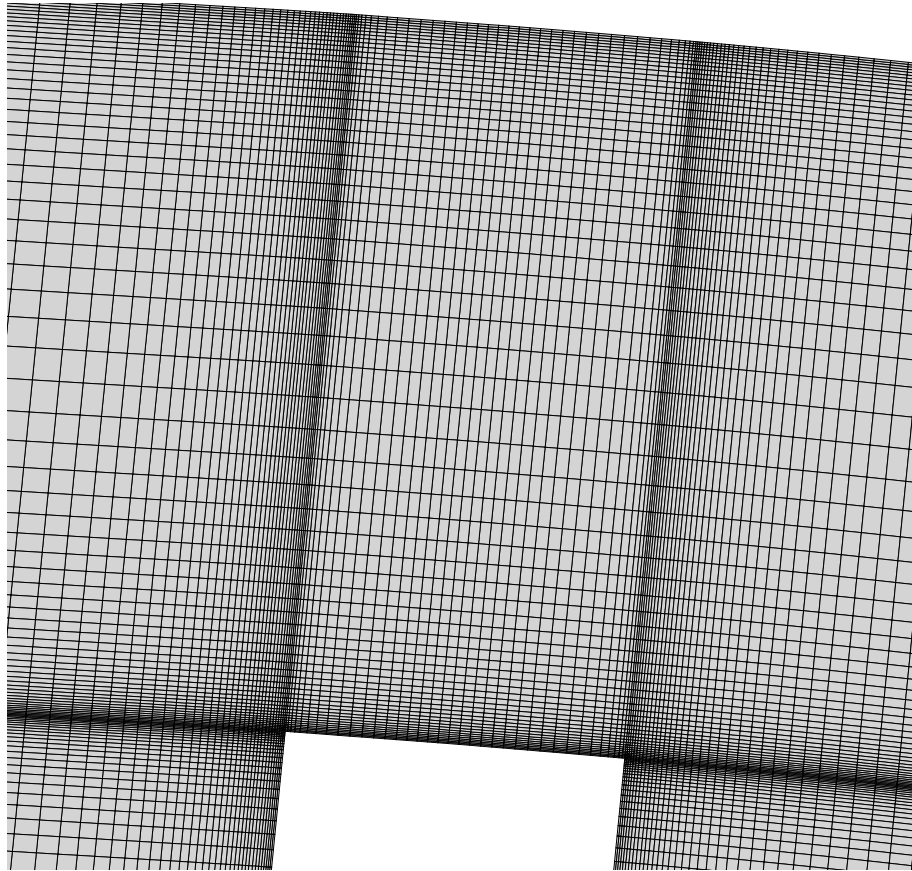


Figure III.5: Typical blade tip clearance grid slice.

each layer of the clearance grid. The three-dimensional coordinates of the boundary are reduced to a two-dimensional set of $(x, \bar{r}\theta)$ coordinates. The initial grids are generated algebraically, and then elliptically smoothed in the same manner as the blade-to-blade grids. Radial basis function interpolation is used to compute the radii of the points on each layer to complete define the three-dimensional coordinates of the grid.

III.1.5. Inlet and Outlet Grids

H-grids can be optionally appended to the blade-to-blade grids to move the inlet and outlet boundaries further away from the blade passage. While it is possible to extend the flow domain using only the blade-to-blade grids, using dedicated extension grids provides for a higher quality mesh in the blade passage region of the flow.

The (x, r) coordinates for the H-grids are found by generating auxiliary meridional grids in the regions labeled as “Inlet” and “Outlet” in Fig. III.1. The spanwise spacing of the grid layers is taken from the adjoining blade-to-blade grids. The streamwise spacing of the grid nodes is set to match closely with the spacing of the blade-to-blade grid to provide for a smooth transition between grids. A constant value of θ from the blade-to-blade grid is used throughout the grid.

III.1.6. Near-Wall Grid Spacing

The spacing of the grid next any wall is an important grid parameter that factors into the quality of the solution, especially viscous solutions. The user has control over the spacing adjacent to the endwalls, the primary surface of the blade, and the very tip of the blade if blade tip grids are present.

In general, the wall spacings are applied uniformly to the grid. Wall spacings are always applied uniformly to grid points that are adjacent to both endwalls and at the tip of the blade in the spanwise direction. The reader is directed to Appendix F for the description of the input parameters that are needed to apply these wall uniform spacings.

The user has two options when specifying the grid spacing adjacent to the pri-

mary blade surface. First, the user can uniformly apply the dimensional wall spacing across the entire blade surface. This first option serves as the default option in the grid generator. Optionally, the grid spacing next to the blade surface can be varied in both the chordwise and spanwise directions.

To generate a grid with varying wall grid spacings, the user specifies the nondimensional grid spacing parameter y^+ , which will be held constant. The dimensional grid spacing can then be computed from the definition of the y^+ number, and is given by

$$y_1 = \frac{\nu y^+}{u^*}.$$

The friction velocity is defined in the usual way as $u^* = \sqrt{\tau_w/\rho}$.

The laminar equation for the wall shear stress, $\tau_w = 0.332\rho U_\infty Re_x^{-1/2}$, was used for the calculation of the friction velocity. The length required by the Reynolds number is computed as the distance along either the pressure or suction side of the blade section. The wall spacing at the leading edge is computed as an average of the points on either side due to the discontinuity that occurs at $x = 0$. The laminar equation was chosen over the turbulent approximation due to the thinner nature of the laminar boundary layer.

From these equations it is apparent that something must be known about the flow for which the grid will be used to simulate. The user must choose an appropriate freestream density, kinematic viscosity, and velocity profile to completely define the above equations. Any number of points greater than or equal to two can be used to define the spanwise variation in freestream velocity. The information about the flow is kept in a separate input file, detailed in Appendix F. The units of the given

freestream conditions must be consistent with the units of the mesh.

III.2. Flow Solver: UNS3D

The equations of motion are solved using an in-house flow solver called UNS3D [Flitan & Cizmas, 2003], which stands for Unstructured, Unsteady Three-Dimensional. Previously, UNS3D has been used to solve turbomachinery flows [Kim, 2003; Brenner et al., 2013], aeroelastic flows [Gargoloff, 2007], and aeroacoustic problems in cavity flows [Liliedahl et al., 2011]. Most recently, UNS3D has been applied to solving aerothermodynamic problems in the hypersonic flow regime [Brown et al., 2014].

The governing equations are discretized using the finite volume method (FVM), requiring that the equations be expressed in integral form. A cell-vertex spatial discretization was used with a median-dual sub-mesh. As a result, the values of the volume averaged state vector are stored at the vertices of the primary mesh. Cell-vertex was chosen over cell-centered due to its flexibility with mixed element meshes, its smaller memory requirements, and for being the less computationally expensive of the two methods.

For a given dual mesh cell, the flux surface integral is approximated as a summation of numerical quadratures on its faces. The Godunov method is used to compute the inviscid fluxes [Godunov, 1959]. The local Riemann problem at each dual mesh interface is solved one of two ways. The first is Roe's approximate Riemann solver [Roe, 1981, 1986], with the Harten entropy fix [Harten & Hyman, 1983]. The second is a modified Roe scheme, known as RoeM, which uses additional functions based on the Mach number to prevent shock instabilities [soo Kim et al., 2003].

Unless otherwise stated, the chosen Riemann solver is the Roe with Harten entropy fix scheme.

Three different methods are available in UNS3D for the evaluation of the gradients at the mesh vertices. They are the Green-Gauss method [Kim, 2003], the least-squares method [Kim, 2003], and the least-squares with QR decomposition (LSQR) [Anderson & Bonhaus, 1994; Haselbacher & Blazek, 2000; Gargoloff, 2007]. The gradients are evaluated exactly for linearly varying functions on mixed grids by the least-squares method [Haselbacher & Blazek, 2000]. As such, and to avoid matrix inversion, the LSQR method was considered the default method, and used for all simulations herein unless otherwise noted.

Second-order spatial accuracy was obtained by using piecewise linear reconstruction [Barth & Jespersen, 1989]. Solution limiting is required when using second-order accuracy to prevent nonphysical oscillations in regions of large gradients, such as in the vicinity of shocks. The limiters used in this work were the Dervieux [1985], Venkatakrishnan [1995], and the Multi-dimensional Limiting Process (MLP) [Park et al., 2010; Park & Kim, 2012] limiters. A modification to the Venkatakrishnan limiter originally applied to the MLP limiter [Park & Kim, 2012] was also applied to the standard Venkatakrishnan limiter used in this work. The modification will be discussed in a later section of this chapter.

An explicit four stage Runge-Kutta method was used to integrate the discrete form of the governing equations. The scheme used is second-order accurate in time. Convergence acceleration techniques are also used to decrease the wall clock time for steady simulations. Both local time stepping [Jameson et al., 1981] and implicit

residual smoothing [Jameson et al., 1986] are used.

The above paragraphs are meant to serve as an overview of the methods used by the flow solver. Refer to Kim [2003] and Gargoloff [2007] for a complete description of the numerical methods mentioned. What follows is a description of the Navier–Stokes equations written in a rotating reference frame. The piecewise linear reconstruction is then presented so that the modification the Venkatakrishnan limiter constant can be properly introduced. Finally, the implementation of the turbulence model equations is given.

III.2.1. Navier–Stokes Equations in the Rotating Reference Frame

For rotational flows, it is convenient to write the governing equations in a reference frame which rotates with the domain of interest. Doing so allows the flow to be solved in a steady manner. Casting the equations in a rotating relative frame introduces two source terms to the momentum equations to account for centrifugal and Coriolis forces [Blazek, 2005, p.438].

The governing equations in the rotating, or relative, reference frame are written in terms of relative velocities, and as such, we wish to define the relationship between the relative and absolute velocities. Figure III.6 illustrates the velocity vectors associated with a point, p , in the rotating reference frame with the axis of rotation aligned with the inertial x -axis. The absolute velocity can be defined as the summation of the relative and transport velocities. The transport velocity defines the movement of the point about the axis of rotation, and is given by $\vec{V} = \vec{\omega} \times \vec{r}$. The relative velocity can then be defined as $\vec{W} = \vec{U} - \vec{\omega} \times \vec{r}$.

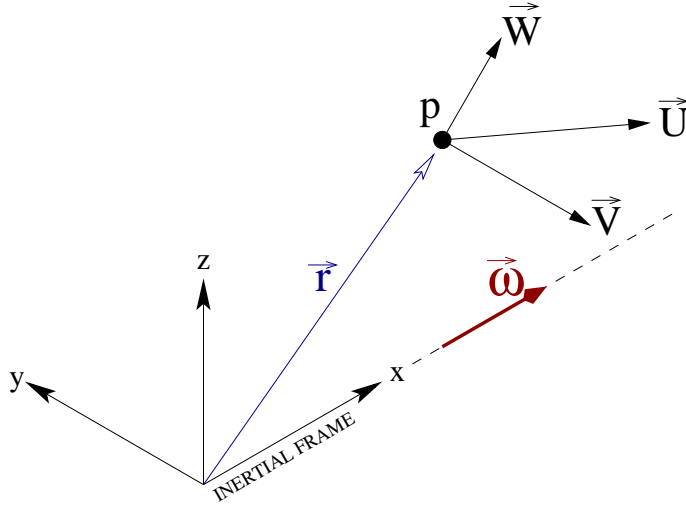


Figure III.6: Velocity diagram for a point rotating about the inertial x-axis.

For this work, the frame's rotation was confined about the inertial x-axis, such that $\vec{\omega} = \{\omega_x, 0, 0\}^T$. As a result, the transport velocity is defined as $\vec{V} = \{0, -z\omega_x, y\omega_x\}^T$, and the relative velocity is given by

$$\vec{W} = \begin{Bmatrix} u_1 \\ u_2 + z\omega_x \\ u_3 - y\omega_x \end{Bmatrix}. \quad (3.1)$$

The governing equations in the relative frame can be cast in terms of absolute velocity components [Chima & Yokota, 1990; Chen et al., 1997]. This is done by substituting (3.1) into the governing equations (see Appendix B). Equation 3.2 presents the integral form of the Navier–Stokes equations in the relative frame, but written

in terms of absolute velocities.

$$\frac{\partial}{\partial t} \int_{\Omega} \vec{Q} d\Omega + \oint_{\partial\Omega} (\vec{F}_c - \vec{F}_v) dS = \int_{\Omega} \vec{G} d\Omega \quad (3.2)$$

The conservative state vector in (3.2) is given by $\vec{Q} = \{\rho, \rho u_1, \rho u_2, \rho u_3, \rho E\}^T$. The convective flux vector, viscous flux vector, and source vector are given, respectively, by

$$\vec{F}_c = \begin{pmatrix} \rho W \\ \rho u_1 W + p n_x \\ \rho u_2 W + p n_y \\ \rho u_3 W + p n_z \\ \rho H W + p V \end{pmatrix} \quad \vec{F}_v = \begin{pmatrix} 0 \\ n_x \tau_{xx} + n_y \tau_{xy} + n_z \tau_{xz} \\ n_x \tau_{yx} + n_y \tau_{yy} + n_z \tau_{yz} \\ n_x \tau_{zx} + n_y \tau_{zy} + n_z \tau_{zz} \\ n_x \Theta_x + n_y \Theta_y + n_z \Theta_z \end{pmatrix} \quad \vec{G} = \begin{pmatrix} 0 \\ 0 \\ \rho u_3 \omega_x \\ -\rho u_2 \omega_x \\ 0 \end{pmatrix} \quad (3.3)$$

$W = \vec{W} \cdot \hat{n}$ is the contravariant velocity, and $V = \vec{V} \cdot \hat{n}$ is the normal component of the transport velocity. The components of the shear stress tensor are given by

$$\begin{aligned} \tau_{xx} &= \mu \frac{2}{3} \left(2 \frac{\partial u_1}{\partial x} - \frac{\partial u_2}{\partial y} - \frac{\partial u_3}{\partial z} \right) \\ \tau_{yy} &= \mu \frac{2}{3} \left(2 \frac{\partial u_2}{\partial y} - \frac{\partial u_1}{\partial x} - \frac{\partial u_3}{\partial z} \right) \\ \tau_{zz} &= \mu \frac{2}{3} \left(2 \frac{\partial u_3}{\partial z} - \frac{\partial u_1}{\partial x} - \frac{\partial u_2}{\partial y} \right) \\ \tau_{xy} &= \tau_{yx} = \mu \left(\frac{\partial u_1}{\partial y} + \frac{\partial u_2}{\partial x} \right) \\ \tau_{xz} &= \tau_{zx} = \mu \left(\frac{\partial u_1}{\partial z} + \frac{\partial u_3}{\partial x} \right) \\ \tau_{yz} &= \tau_{zy} = \mu \left(\frac{\partial u_2}{\partial z} + \frac{\partial u_3}{\partial y} \right). \end{aligned} \quad (3.4)$$

The viscous stress work and heat conduction terms in the viscous flux vector are

given by

$$\begin{aligned}
\Theta_x &= u_1\tau_{xx} + u_2\tau_{xy} + u_3\tau_{xz} + k\frac{\partial T}{\partial x} \\
\Theta_y &= u_1\tau_{yx} + u_2\tau_{yy} + u_3\tau_{yz} + k\frac{\partial T}{\partial y} \\
\Theta_z &= u_1\tau_{zx} + u_2\tau_{zy} + u_3\tau_{zz} + k\frac{\partial T}{\partial z},
\end{aligned} \tag{3.5}$$

where the thermal conductivity coefficient, k , was previously defined by (2.9), repeated below for convenience. For air, the Prandtl number is $Pr = 0.72$.

$$k = c_p \frac{\mu}{Pr} \tag{2.9}$$

III.2.1.1. Semi-Discrete Form

We wish to express (3.2) for each cell in the discretized domain. This done by introducing the numerical approximations for the volume and surface integrals found in the governing equations. The time derivative will remain exact for the time being. As a result, the equations obtained are known as the semi-discrete equations.

The vector of conservative variables is volume averaged and stored at the vertices of the mesh. For a given vertex, i , the volume averaged state vector is given by

$$\vec{q}_i \equiv \frac{1}{\Omega_i} \int_{\Omega} \vec{Q}_i d\Omega.$$

For this work, the mesh is assumed to remain fixed throughout the simulation. As a result, the time rate of change of the state vector for a given dual mesh cell may be written as

$$\frac{\partial}{\partial t} \int_{\Omega} \vec{Q}_i d\Omega = \Omega_i \frac{\partial \vec{q}_i}{\partial t}. \tag{3.6}$$

The source vector is assumed to be constant throughout a cell. Meaning the

volume integral of the source vector may be expressed as

$$\int_{\Omega} \vec{G}_i d\Omega = \Omega_i \vec{g}_i. \quad (3.7)$$

The surface integral in (3.2) is expressed as an approximation by summing the fluxes through each face of the cell. It is enough to assume a constant value for the flux at the midpoint of the face to remain second-order accurate. Therefore, the surface integral of the fluxes can be written as

$$\oint_{\partial\Omega} (\vec{F}_c - \vec{F}_v) dS \approx \sum_{j=1}^{N_F} (\vec{F}_c - \vec{F}_v)_j S_j, \quad (3.8)$$

where N_F is the number of faces for cell i , and S_j is the area of the current face.

Substituting (3.6)–(3.8) into (3.2) results in the semi-discrete form of the governing equations, given by (3.9). The bracketed terms in (3.9) are known collectively as the residual, \vec{R}_i .

$$\frac{\partial \vec{q}_i}{\partial t} = -\frac{1}{\Omega_i} \left[\sum_{j=1}^{N_F} (\vec{F}_c - \vec{F}_v)_j S_j - \Omega_i \vec{g}_i \right] = \frac{\vec{R}_i}{\Omega_i} \quad (3.9)$$

The discrete form of the equations are formed once the temporal discretization is accounted for. See Kim [2003] for a discussion on the temporal discretization.

III.2.2. Piecewise Linear Reconstruction

Second-order spatial accuracy is achieved by using piecewise linear reconstruction [Barth & Jespersen, 1989]. This method assumes that the solution variation within a cell is piecewise linear. The solution gradients at the mesh vertex are required for the reconstruction.

The reconstruction is used to define the left and right states needed to compute

the fluxes through a face. For a given volume averaged scalar variable, q , the left and right states are given by

$$\begin{aligned} q_L &= q_i + \frac{1}{2}\Phi_i(\nabla q_i \cdot \vec{r}_{ij}) \\ q_R &= q_j - \frac{1}{2}\Phi_j(\nabla q_j \cdot \vec{r}_{ij}) , \end{aligned} \tag{3.10}$$

where Φ is a limiter function, ∇q is the gradient of q , and \vec{r}_{ij} is the direction vector that points from i to j . Figure III.7 illustrates the piecewise linear reconstruction for a one dimensional case.

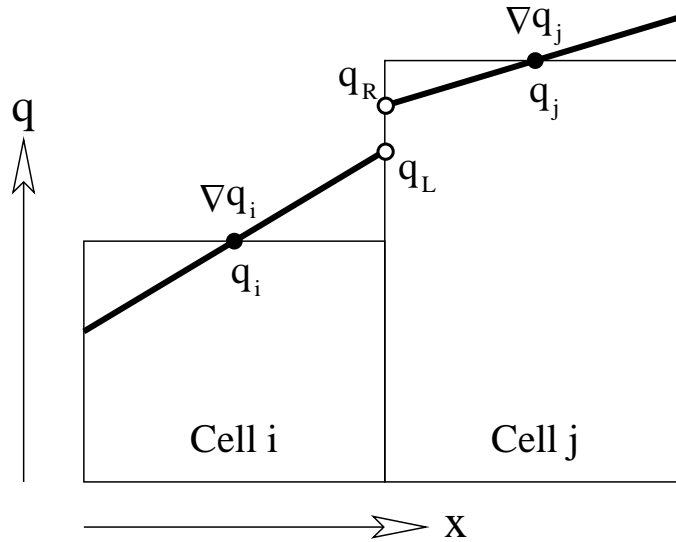


Figure III.7: Linear reconstruction of a scalar variable, q .

A limiter function is needed to prevent nonphysical oscillations in the solution in regions with large gradients. For example, this would include the area in the vicinity of a shock. As seen in (3.10), the limiter acts upon the gradient, functioning as a slope limiter. In regions of large discontinuities, the limiter tends toward a value of

zero. A value of zero results in a monotonicity preserving first-order accurate constant reconstruction.

The Dervieux [1985], Venkatakrishnan [1995], and MLP [Park et al., 2010; Park & Kim, 2012] limiter functions were used in this work. The MLP limiter used was the MLP-u2 limiter [Park & Kim, 2012]. The MLP-u2 and Venkatakrishnan limiters are discussed below. Refer to Gargoloff [2007] for a description of the implementation of the Dervieux limiter.

At the vertex i , the Venkatakrishnan limiter is given by $\Phi_i^V = \min \{ \phi_{i1}^V, \dots, \phi_{iN_v}^V \}$; where ϕ_{ij}^V is given by (3.11), and N_v is the number of vertices connected by edge to i . Figure III.8 (left half) defines the limiter's stencil. In (3.11) q_i^{max} and q_i^{min} are the local maximum and minimum values, respectively, of the variable. They are defined using the vertices connected to and including vertex i . The direction vector, \vec{r}_{ij} , points to the dual mesh boundary as shown in Fig. III.8. The Venkatakrishnan function, φ , is given by (3.12).

$$\phi_{ij}^{(V)} = \min \begin{cases} \varphi \left(\frac{q_i^{max} - q_i}{\nabla q_i \cdot \vec{r}_{ij}} \right) & \text{if } \nabla q_i \cdot \vec{r}_{ij} > 0 \\ \varphi \left(\frac{q_i^{min} - q_i}{\nabla q_i \cdot \vec{r}_{ij}} \right) & \text{if } \nabla q_i \cdot \vec{r}_{ij} < 0 \\ 1 & \text{otherwise} \end{cases} \quad (3.11)$$

$$\varphi \left(\frac{a}{b} \right) = \frac{1}{b} \left[\frac{(a + \epsilon^2) b + 2b^2 a}{a^2 + 2b^2 + ba + \epsilon^2} \right] \quad (3.12)$$

At the vertex i , the MLP limiter is defined as $\Phi_i^M = \min \{ \phi_{i1}^M, \dots, \phi_{iN_c}^M \}$. Here, ϕ_{ij}^M is given by (3.13), and N_c is the number of dual mesh vertices associated with the vertex i . Figure III.8 (right half) defines the stencil for the MLP limiter. q_j^{max}

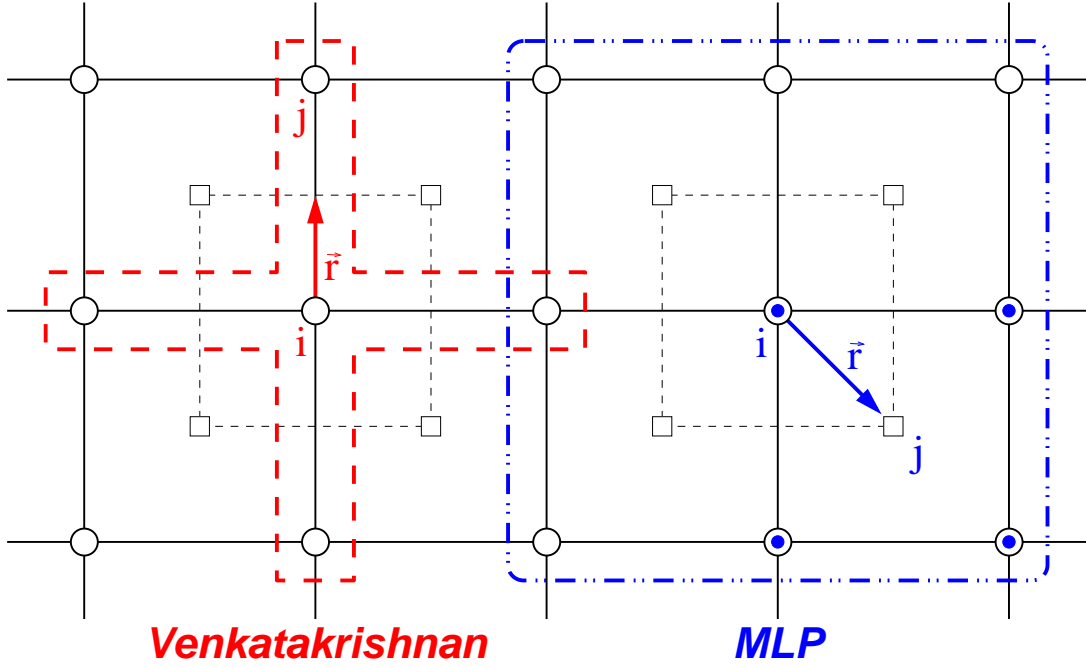


Figure III.8: Stencils for the Venkatakrishnan (---) and MLP (- · - ·) limiters in two dimensions; “o” denote primary grid vertices, “□” are dual mesh vertices, “⊙” denote vertices used to compute variable extremum for a dual mesh vertex j .

and q_j^{min} are defined as the maximum and minimum values of the variable from the vertices associated with the dual mesh vertex j , as shown in Fig. III.8. Equation 3.12 is used to define φ in (3.13), making the limiter the MLP-u2 limiter [Park & Kim, 2012].

$$\phi_{ij}^{(M)} = \begin{cases} \varphi \left(\max \left\{ \frac{q_j^{max} - q_i}{\nabla q_i \cdot \vec{r}_{ij}}, \frac{q_j^{min} - q_i}{\nabla q_i \cdot \vec{r}_{ij}} \right\} \right) & \text{if } \nabla q_i \cdot \vec{r}_{ij} \neq 0 \\ 1 & \text{otherwise} \end{cases} \quad (3.13)$$

The ϵ^2 term in (3.12) is used to control the amount of limiting applied. Venkatakr-

ishnan [1995] gives the definition to be

$$\epsilon^2 = (C\Delta x)^3, \quad (3.14)$$

where C is a constant, and Δx is a measure of the local spatial discretization. For very large values of C , the limiter tends towards being unlimited, and a value of zero corresponds to no limiting. In this work C has a default value of 5.0, and $(\Delta x)^3$ is represented by the volume of the local dual mesh cell, Ω_i .

A potential issue with the basic definition of ϵ is that it is based solely on the geometry. Large variations in the cell volumes included in the limiter’s stencil could prevent the detection of smooth flow regions. This would cause the limiter to become active in a region where it is not needed, which in turn would negatively impact convergence.

Wang [2000] suggested making ϵ a function of q only. Park & Kim [2012] posit that ϵ should maintain a connection to the spatial accuracy, and introduced a new definition for ϵ based upon the local spatial discretization and flow variation. Their new definition was used with the MLP-u2 limiter to create the MLP-u2 (new) limiter [Park & Kim, 2012].

The ϵ definition by Park & Kim was implemented for both the Venkatakrishnan and MLP-u2 limiters in UNS3D as a user defined option. Herein limiters using this new ϵ definition are denoted with the suffix “- ϵ_2 ”, and “- ϵ_1 ” is used when the original definition is used by the limiter.¹ The Park & Kim definition is given by

$$\epsilon^2 = \frac{K_1}{1 + \theta} \Delta q_i,$$

¹The MLP- ϵ_1 and MLP- ϵ_2 limiters correspond to the MLP-u2 and MLP-u2 (new) limiters found in Park & Kim [2012].

where $\Delta q_i = q_i^{max} - q_i^{min}$ is the maximum variation of the local flow. For the Venkatakrisnan limiter, q_i^{max} and q_i^{min} are the maximum and minimum values of the variable from the vertices connected to and including vertex i . For MLP, the extremum are taken from the vertices enclosed by the stencil in the right half of Fig. III.8. The parameter θ is the ratio of the local flow variation to the local measure of spatial discretization, and is defined as

$$\theta = \frac{\Delta q_i}{K_2 \Delta x^{3/2}} = \frac{\Delta q_i}{K_2 \sqrt{\Omega_i}}.$$

The constants K_1 and K_2 were assigned a value of 5.0 in accordance to Park & Kim [2012].

III.2.3. Turbulence Model Implementation

The turbulence equations share a similar form to those of the Navier–Stokes equations. As such, they can be written in integral form similarly to (3.2). Equations 2.21 and 2.22 can therefore be written as

$$\frac{\partial}{\partial t} \int_{\Omega} \vec{Q}_T d\Omega + \oint_{\partial\Omega} (\vec{\Gamma}_c - \vec{\Gamma}_v) dS = \int_{\Omega} \vec{\Pi} d\Omega. \quad (3.15)$$

The conservative turbulent state vector is defined as $\vec{Q}_T = \{\rho\kappa, \rho\omega\}^T$. The vectors defining the convective and viscous-like fluxes are given, respectively, by

$$\vec{\Gamma}_c = \begin{Bmatrix} \rho\kappa W \\ \rho\omega W \end{Bmatrix} \quad \vec{\Gamma}_v = \begin{Bmatrix} n_x \tau_x^\kappa + n_y \tau_y^\kappa + n_z \tau_z^\kappa \\ n_x \tau_x^\omega + n_y \tau_y^\omega + n_z \tau_z^\omega \end{Bmatrix}, \quad (3.16)$$

where, again, W is the contravariant velocity as defined on page 40. The “shear stress” components given in (3.16) are defined as

$$\tau_{\aleph}^\kappa = (\mu + \sigma_\kappa \mu_T) \frac{\partial \kappa}{\partial \aleph} \quad \tau_{\aleph}^\omega = (\mu + \sigma_\omega \mu_T) \frac{\partial \omega}{\partial \aleph},$$

where x , y , or z can be substituted for \aleph . The vector of source terms is given by

$$\vec{\Pi} = \left\{ \begin{array}{c} P_\kappa - D_\kappa \\ \rho\alpha P_\kappa/\mu_T - D_\omega + CD \end{array} \right\}, \quad (3.17)$$

where P_κ is the production term defined by (2.23). In (3.17), D_κ and D_ω define the dissipation of κ and ω , respectively, and are defined by (3.18) and (3.19). The cross diffusion term, CD , is given by (3.20), and the blending function, F_1 , was defined by (2.24).

$$D_\kappa = \beta^* \rho \omega \kappa \quad (3.18)$$

$$D_\omega = \beta \rho \omega^2 \quad (3.19)$$

$$CD = \frac{2\rho\sigma_\omega}{\omega} \nabla_\kappa \nabla_\omega (1 - F_1) \quad (3.20)$$

It is evident from (3.16) that the turbulence equations have been written in the rotational reference frame. Therefore, it is prudent to mention that the production term, P_κ , as well as the definition for the eddy viscosity are invariant. The model constants were defined in Table II.1 on page 20.

A limiter is applied to the production of turbulent kinetic energy to prevent spikes in the eddy viscosity, and to prevent turbulence build up in stagnation regions [Menter, 1993; Menter et al., 2003]. Equation 3.21 defines the production limiter. The value of the constant coefficient multiplying the dissipation of k in (3.21) was suggested by Menter et al. [2003].

$$\tilde{P}_\kappa = \min(P_\kappa, 10 \cdot D_\kappa) \quad (3.21)$$

The numerical methods used to solve the Navier–Stokes equations were applied to (3.15) as the two sets of equations share the same form. Menter [1993] showed the

solution of the turbulence equations to be nearly independent of the scheme and its accuracy used to solve the equations. As a result, the convective fluxes are computed using Roe's approximate Riemann solver with a first-order constant reconstruction. The viscous fluxes are computed using averaged gradients.

The source terms can dominate the turbulence equations, leading to the equations becoming stiff. One way to alleviate this is to treat the source terms implicitly. To do this, we start with the discrete form of (3.15). For a given cell, i , and at time n , the discrete form of the turbulence equations are given by

$$\Delta \vec{q}_{T,i}^n = -\frac{\Delta t_i}{\Omega_i} \left[\sum_{j=1}^{N_F} \left(\vec{\Gamma}_c^n - \vec{\Gamma}_v^n \right)_j S_j - \vec{\pi}_i^n \Omega_i \right].$$

Here $\vec{q}_{T,i}$ and $\vec{\pi}_i$ are, respectively, the volume averaged turbulent state vector and the volume averaged vector of source terms. Evaluating the source term at the next time step yields

$$\Delta \vec{q}_{T,i}^n = -\frac{\Delta t_i}{\Omega_i} \left[\sum_{j=1}^{N_F} \left(\vec{\Gamma}_c^n - \vec{\Gamma}_v^n \right)_j S_j - \vec{\pi}_i^{n+1} \Omega_i \right]. \quad (3.22)$$

Using a Taylor series, the source term at the next time step can be approximated as

$$\vec{\pi}_i^{n+1} \approx \vec{\pi}_i^n + \Delta \vec{q}_{T,i}^n \left. \frac{\partial \vec{\pi}}{\partial \vec{q}_T} \right|_i^n. \quad (3.23)$$

Menter [1993] suggests the linearized form of the derivatives found in the series expansion be given by

$$\frac{\partial \vec{\pi}}{\partial \vec{q}_T} \approx \begin{bmatrix} -D_\kappa/\kappa & 0 \\ 0 & -(|CD| + 2D_\omega)/\omega \end{bmatrix}. \quad (3.24)$$

By substituting (3.23) into (3.22), we arrive at

$$\Delta \vec{q}_{T,i}^n = -\frac{\Delta t_i}{\Omega_i} \left[\sum_{j=1}^N \left(\vec{\Gamma}_c^n - \vec{\Gamma}_v^n \right)_j S_j - \vec{\pi}_i^n \Omega_i \right] + \Delta t_i \Delta \vec{q}_{T,i}^n \left. \frac{\partial \vec{\pi}}{\partial \vec{q}_T} \right|_i^n.$$

Defining the residual, \vec{R}_i , as the quantity inside the brackets of the above equation and collecting like terms on the left hand side gives

$$\left(\bar{\mathbf{I}} - \Delta t_i \left. \frac{\partial \vec{\pi}}{\partial \vec{q}_T} \right|_i \right) \Delta \vec{q}_{T,i}^n = - \frac{\Delta t_i}{\Omega_i} \vec{R}_i^n.$$

Multiplying both sides by the inverse of the matrix from the left hand side of the above equation yields the final discrete form of turbulence equations, given by

$$\vec{q}_{T,i}^{n+1} = \vec{q}_{T,i}^n - \frac{\Delta t_i}{\Omega_i} \bar{\Upsilon}_i^n \vec{R}_i^n, \quad (3.25)$$

where the matrix $\bar{\Upsilon}_i^n$ is defined as

$$\bar{\Upsilon}_i^n = \begin{bmatrix} (1 + \Delta t D_\kappa / \kappa)^{-1} & 0 \\ 0 & (1 + \Delta t (|CD| + 2D_\omega) / \omega)^{-1} \end{bmatrix}.$$

III.3. Boundary Conditions

The boundary conditions used to set the flow conditions are discussed in this section. The boundary conditions needed for turbomachinery simulations include wall, inlet, outlet, and periodic boundary conditions. The wall, inlet, and outlet boundary conditions are discussed herein. Refer to Kim [2003] for a discussion on the periodic boundaries. Also included at the end of this section are the boundary conditions needed by the turbulence model.

The boundary conditions in UNS3D are primarily applied weakly as boundary fluxes [Whitaker, 1993]. This is opposed to directly applying the boundary conditions to the boundary nodes. Intermediate values of the state variables are computed at quadrature points on the boundary faces. The computed intermediate values are used as one side of the Riemann problem in the flux calculation for the adjacent

boundary node. A first-order constant reconstruction is used when computing the boundary fluxes. Figure III.9 shows the quadrature points, open circles, for a dual mesh cell adjacent to a boundary in two dimensions.

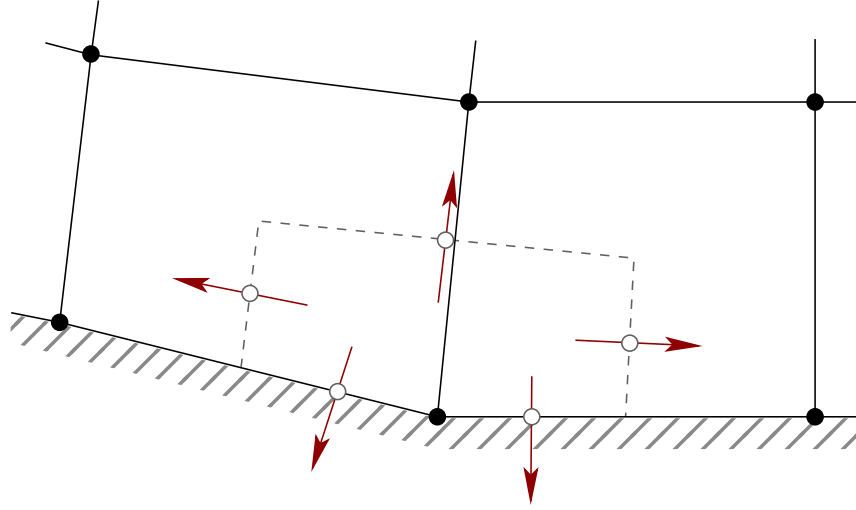


Figure III.9: Flux quadrature points (\circ) for a boundary adjacent dual mesh volume (---).

III.3.1. Wall Boundaries

At the wall boundaries the contravariant velocity, W , is set to zero. This no penetration condition results in the boundary flux for the continuity equation to vanish, and is the only necessary condition for inviscid flows. For the no-slip condition, the absolute velocity components are set to zero for stationary walls and to $\vec{V} = [0, -z\omega_x, y\omega_x]^T$ for rotating walls. Unlike any other boundary conditions, the no-slip condition is strongly enforced at the boundary; meaning the wall velocity is

applied directly to the boundary vertices. As a result, the momentum equations are no longer required to be solved at the wall.

For the no-slip boundaries, the only contributions to the energy flux are $p_w (\vec{V} \cdot \hat{n})$ and $k\nabla T_w$. Assuming an adiabatic wall condition results in $\nabla T_w \cdot \hat{n} = 0$. The contribution to the energy flux from the wall boundary vanishes entirely for an adiabatic, stationary wall.

III.3.2. Subsonic Inlet

Only subsonic inflow is considered in this work. The subsonic inlet requires one condition be set by the interior of the flow and four conditions from upstream. Here the outgoing Riemann invariant from the interior is used in conjunction with the total pressure, total temperature, and two flow angles being specified upstream. Refer to Blazek [2005, p. 287] for a description of how the intermediate states are computed.

III.3.3. Subsonic Outlet

The subsonic outlet condition is defined using four states from the interior and one imposed external state. Entropy, tangential velocity, speed of sound, and total pressure are taken from the interior. The imposed downstream condition is the static pressure, p_b . Refer to Kim [2003] for a description of how the intermediate states are computed at the boundary for a constant external pressure.

Imposing a constant value for the external static pressure, while logical for external flows, is not necessarily prudent for internal flows in which the flow is turned. For

the internal flow case the static pressure is determined by integrating the axisymmetric radial momentum equation [Chima & Yokota, 1990], given by

$$\frac{dp}{dr} = \frac{\rho v_\theta^2}{r}, \quad (3.26)$$

where $v_\theta = (u_2 z - u_3 y) / r$. The integration requires that the pressure be specified at one end of the profile. The pressure was imposed at the hub in this work.

Circumferentially averaged values of density and tangential velocity were used for the integration due to the axisymmetric nature of (3.26). The radii were taken from the vertices along one of the periodic faces. The integration was approximated using the trapezoidal rule. The resultant pressure profile was used as the basis to find the pressure at each outlet boundary face quadrature point through linear interpolation.

III.3.4. Turbulence Boundary Conditions

Boundary conditions at the inlet, outlet, and walls must be specified for the turbulence model. The turbulence boundary conditions at the inlet and outlet are applied weakly in the same manner as the flow boundary conditions. The wall turbulence boundary conditions are applied directly to the wall vertices.

The freestream turbulence intensity and turbulent length scale are specified by the user at the inlet. The freestream turbulence intensity is given by [Blazek, 2005, p. 246]

$$I_T|_\infty = \frac{\sqrt{\frac{2}{3}\kappa_\infty}}{\|\vec{U}_\infty\|_2}, \quad (3.27)$$

and the freestream turbulent length scale is given by [Menter, 1993]

$$\ell_T|_\infty = \frac{\sqrt{\kappa_\infty}}{0.09\omega_\infty}. \quad (3.28)$$

Equations 3.27 and 3.28 are rearranged to define the freestream conditions for κ and ω , which are what is imposed by the turbulence inlet condition within the flow solver.

The resultant definitions for the freestream conditions are given in (3.29).

$$\kappa_\infty = \frac{3}{2} \left(\left\| \vec{U}_\infty \right\|_2 I_T|_\infty \right)^2 \quad \omega_\infty = \frac{\sqrt{\kappa_\infty}}{0.09 \ell_T|_\infty} \quad (3.29)$$

The boundary conditions at the wall are given by [Menter, 1993]

$$\kappa_w = 0 \quad \omega_w = \frac{60\mu_w}{0.075\rho_w (\Delta y)^2}, \quad (3.30)$$

where Δy is the distance from the wall to the nearest interior point. A zero gradient for both turbulence variables is specified at the outlet.

III.4. Parallelization

The flow solver was previously parallelized to decrease the wall clock times required to achieve convergence [Kim, 2003]. The parallelization uses the Message Passing Interface (MPI) to facilitate communication between processors. The solution domain is decomposed into multiple blocks, one for each processor, and with each block overlapping the adjacent block by a single cell. Unlike the previous work by Kim [2003], which decomposed the domain in a single direction, the current work uses a multi-directional domain decomposition strategy [Brown, 2016]. Refer to Brown for a detailed description of the parallel solution strategy.

III.5. Turbomachinery Performance Analysis

This section discusses the post-processing steps taken to compute the thermodynamic performance parameters of a turbomachine row. This includes the different averaging techniques used and the calculation of equivalent values, efficiencies, and thermodynamic property ratios. A separate program from the flow solver was written and used for the solution post-processing.

The post-processing program extracts the solution from cuts upstream and downstream of the blade row. The cuts are made on planes of constant x , and their locations are user specified. The extracted properties of the flow on these grids are averaged to produce both spanwise profiles and overall averages.

Dual mesh faces are constructed to compute the geometric information required for the averaging. Figure III.10 shows a portion of an idealized extracted grid, black solid grid, with the corresponding dual mesh in grey. The normal vector at a given (j, k) location, as shown in Fig. III.10, is defined as $\hat{n} = \vec{S}_{jk}/S_{jk}$, where the area based vector is given by [Blazek, 2005, p. 82]

$$\vec{S}_{jk} = \frac{1}{2} \begin{bmatrix} \Delta z_A \Delta y_B - \Delta y_A \Delta z_B \\ \Delta x_A \Delta z_B - \Delta z_A \Delta x_B \\ \Delta y_A \Delta x_B - \Delta x_A \Delta y_B \end{bmatrix},$$

and $\Delta \xi_A = \xi_4 - \xi_2$ and $\Delta \xi_B = \xi_3 - \xi_1$ for $\xi = x, y, z$. The numbered indices correspond to the numbering shown in Fig. III.10 for the dual mesh face. The area of the face is given by $S_{jk} = \left\| \vec{S}_{jk} \right\|_2$.

Velocities, flow angles, and the intensive properties of the flow are averaged using an area average. The area average is defined as

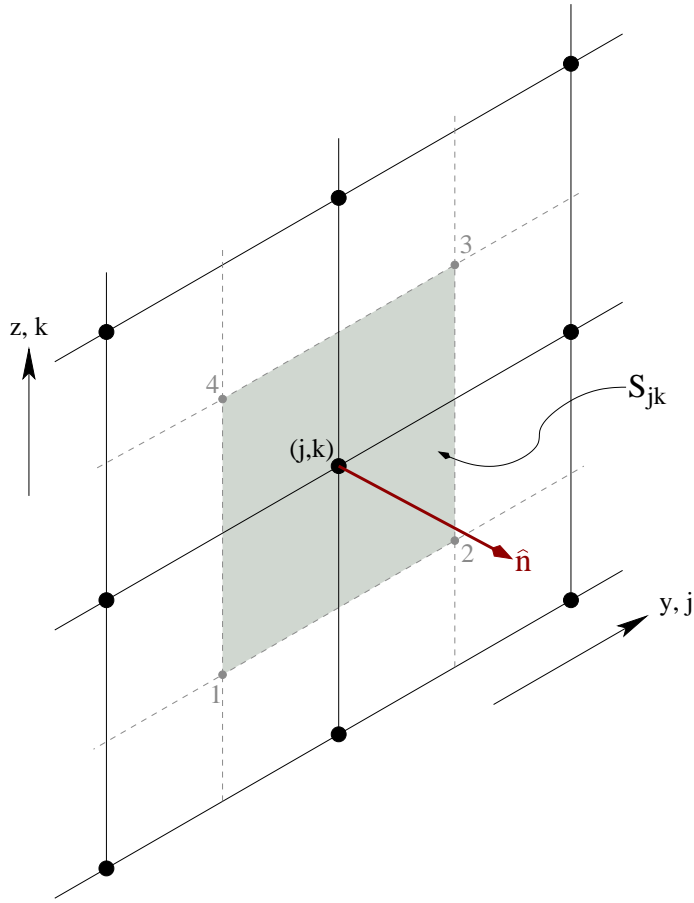


Figure III.10: Defining geometric quantities using an idealized extracted grid (black, solid) and the corresponding mesh duals (grey, dashed).

$$\bar{\varphi} = \frac{1}{S} \int \varphi dS,$$

where φ represents any variable of interest. Although total temperature is an intensive property, it is appropriate to compute its average using a mass average [Cumpsty & Horlock, 2006]. The mass averaged total temperature is given by

$$\bar{T}_0 = \frac{1}{\dot{m}} \int T_0 d\dot{m}.$$

Based on the recommendation of Cumpsty & Horlock [2006], the total pressure was work averaged as opposed to mass averaged. Work averaging the total pressure results in an average value that produces the correct work input or output for an ideal compressor. The work averaged total pressure is defined as [Cumpsty & Horlock, 2006]

$$\bar{p}_0 = \left[\frac{\int T_0 d\dot{m}}{\int \left(T_0 / p_0^{\frac{\gamma-1}{\gamma}} \right) d\dot{m}} \right]^{\frac{\gamma}{\gamma-1}}.$$

The thermodynamic performance parameters can be computed once average values are obtained at the upstream and downstream planes. The total pressure ratio is defined as

$$\pi^* = \frac{\bar{p}_{0,2}}{\bar{p}_{0,1}},$$

where the subscripts “1” and “2” refer, respectively, to the upstream and downstream locations. The adiabatic efficiency is defined as [Howell, 1945b]

$$\eta_{ad} = \frac{(\pi^*)^{(\gamma-1)/\gamma} - 1}{\frac{\bar{T}_{0,2}}{\bar{T}_{0,1}} - 1},$$

and the polytropic efficiency is given by [Howell, 1945b]

$$\eta_{pc} = \frac{\gamma - 1}{\gamma} \frac{\ln(\pi^*)}{\ln\left(\frac{\bar{T}_{0,2}}{\bar{T}_{0,1}}\right)}.$$

The computed mass flow rate and the specified wheel speed have been corrected to standard sea level conditions. The correction is done so that the results can be readily compared against those found in literature. The sea level equivalent mass flow rate and wheel speed are given, respectively, by [Hill & Peterson, 1992, p. 296]

$$\dot{m}_{EQ} = \frac{\dot{m}\sqrt{\theta}}{\delta} \quad N_{EQ} = \frac{N}{\sqrt{\theta}}, \quad (3.31)$$

where $\theta = T_{0,1}/T_{std}$ and $\delta = p_{0,1}/p_{std}$. Here $T_{std} = 288.3$ K and $p_{std} = 101325$ Pa. The total conditions are defined as the total conditions specified in the input file to the flow solver.

CHAPTER IV

VERIFICATION AND VALIDATION RESULTS

This chapter presents the results from the verification and validation tests that were conducted to assess the modifications made to UNS3D. The results from the higher order limiter verification are presented first. The results for a turbulent flat plate geometry are then presented.

A convergent-divergent nozzle and an isolated airfoil in a supersonic freestream were used to test the modification of the Venkatakrisnan and MLP limiters found within UNS3D. Both flows were assumed to be inviscid. The two geometries are also used to compare the effectiveness of the original limiters found in UNS3D.

IV.1. Convergent-Divergent Nozzle

This section presents the numerical results for two inviscid convergent-divergent nozzle flows. The two flows examined are: (1) purely subsonic flow and (2) subsonic-supersonic-subsonic flow with a normal shock. These are excellent test cases as analytic results are available (see Appendix C) for comparison.

These tests served two purposes. The first was to verify that using the Park & Kim [2012] definition for ϵ^2 in the Venkatakrisnan and MLP limiters did not adversely affect the solution. The second purpose of these simulations was to quantify the apparent order of accuracy for the given limiters.

IV.1.1. Nozzle Definition and Grid Generation

The variation in nozzle area was the same as that used by Liou [1987]. Equation 4.1 defines the nozzle area with respect to the streamwise coordinate. Figure IV.1 shows the resultant nozzle shape. The length of the nozzle was 10 in. (0.254 m) and the throat was located at $x = 5$ in. (0.127 m). The cross sectional area of the throat was 1 in.² (6.4516×10^{-4} m²).

$$\frac{A(x)}{A_{\text{throat}}} = \begin{cases} \frac{7}{4} - \frac{3}{4} \cos \left[\left(2\frac{x}{l} - 1 \right) \pi \right] & x \leq l/2 \\ \frac{5}{4} - \frac{1}{4} \cos \left[\left(2\frac{x}{l} - 1 \right) \pi \right] & x > l/2 \end{cases}, \quad 0 \leq x \leq l \quad (4.1)$$

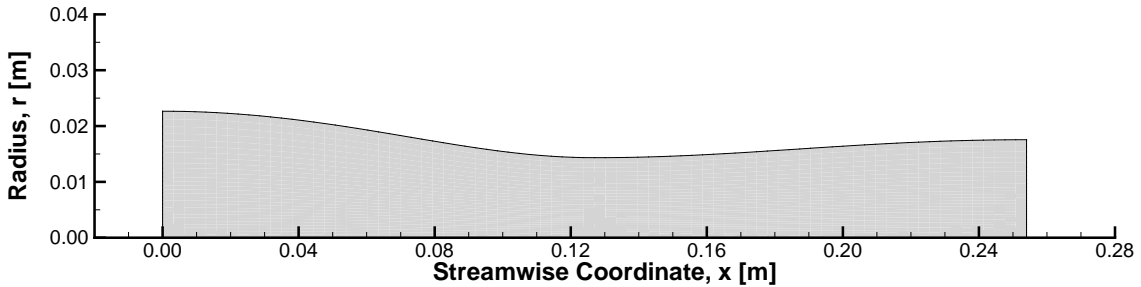


Figure IV.1: Convergent-divergent nozzle profile.

Four computational grids of increasing refinement were constructed for the given geometry. The grids are constructed entirely of hexahedral elements with equally spaced grid layers in the streamwise direction. Each grid layer consists of an H-grid in the interior, which is surrounded by O-grid. The finest mesh, called the “super-fine” mesh herein, was constructed first. Lower refinement grids were constructed by removing every other point along each edge of the next finest grid. Figure IV.2

illustrates the general configuration of the grid, as well as showing a detail of the grid refinement.

A nondimensional grid spacing parameter, h , was defined for each of the four grids as $h = \Delta x / \Delta x_{SF}$. Δx is the uniform distance between grid layers, and Δx_{SF} is the distance between layers for the super-fine mesh. Table IV.1 details the number of nodes and the nondimensional grid spacing parameter for each grid used in this study.

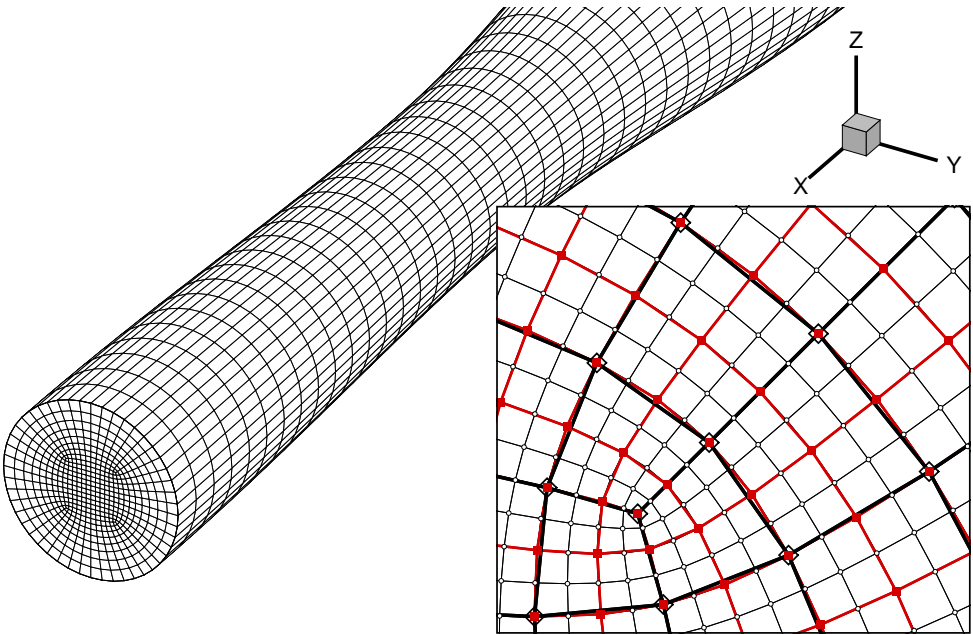


Figure IV.2: Medium nozzle grid and grid refinement detail.

Table IV.1: Convergent-divergent grid dimensions

Grid Name	Total Nodes	h
Coarse	3,045	8
Medium	22,673	4
Fine	175,041	2
Super-fine	1,375,745	1

IV.1.2. Case Overview

Both the subsonic and normal shock cases utilized the same reference and reservoir conditions. The outlet static pressure was varied to set the type of flow within the nozzle. The parameters used to define the flow for both cases can be found in Table IV.2.

Table IV.2: Nozzle flow initial and boundary conditions.

Reference Conditions	Reservoir Conditions
$p_\infty = 101325.0 \text{ Pa}$	$p_0 = 104190.585 \text{ Pa}$
$T_\infty = 288.3 \text{ K}$	$T_0 = 290.435 \text{ K}$
$M_\infty = 0.2$ –	
Exit Pressure Conditions	
Subsonic case: $p_e/p_0 = 0.89$	
Normal shock case: $p_e/p_0 = 0.835$	

Six cases based on desired solution accuracy and limiter selection were defined. Each case was run on every grid and for both flow conditions. The six cases were defined as:

1. First-order accurate¹
2. Second-order accurate using the Dervieux limiter
3. Second-order accurate using the Venkatakrishnan limiter with the original ϵ^2 formulation (Venka- ϵ_1)
4. Second-order accurate using the Venkatakrishnan limiter with the flow based ϵ^2 formulation (Venka- ϵ_2)
5. Second-order accurate using the MLP limiter with the original ϵ^2 formulation (MLP- ϵ_1)
6. Second-order accurate using the MLP limiter with the flow based ϵ^2 formulation (MLP- ϵ_2)

¹First-order accuracy was obtained using a constant construction at the cell faces as opposed to piecewise linear for second-order accuracy.

IV.1.3. Solution Post-Processing Methodology

This section defines the methods used to post-process the results obtained from the convergent-divergent nozzle simulations. The method used to compare two solutions is presented first. Then the method used to compute the solution order of accuracy is given.

The difference between two solutions computed on the same mesh was quantified by computing the root mean square (RMS) of the absolute percent difference at each value location. The absolute percent difference was defined as

$$\Delta = \frac{|\aleph_1 - \aleph_2|}{\aleph_1} \quad (4.2)$$

where \aleph is the solution variable of choice and the subscript gives the solution number. The RMS of the solution difference was computed once the absolute percent difference between solutions was computed at every point using

$$\Delta_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_{i=1}^N \Delta_i^2} \quad (4.3)$$

where N is the number of data points in the set.

The apparent order of accuracy was computed for each of the cases given in the previous section. The apparent order was calculated using [Celik et al., 2008]

$$p = \frac{|\ln(\varepsilon_{32}/\varepsilon_{21})|}{\ln(r_{21})} \quad (4.4)$$

where $r_{21} = h_2/h_1$ and $\varepsilon_{ji} = \aleph_j - \aleph_i$ for the quantity of interest, \aleph . The subscripts “1”, “2”, “3” denote the level of grid refinement, with “1” being for the most refined mesh. For this work, the subscript “1” corresponded to the super-fine mesh, “2” to the fine mesh, and “3” to the medium mesh.

The quantity of interest used to compute the apparent solution orders was the absolute percent error of mass flow rate. The absolute percent error was defined by (4.2) with \aleph_1 being the analytic value and \aleph_2 being the computed value for the given grid and case.

IV.1.4. Subsonic Flow

This section presents the results of the six aforementioned cases for the subsonic flow through a convergent-divergent nozzle. The convergence histories of each case are presented first. The solution along the center-line of the nozzle is then presented. Finally, the computed order of accuracy of each solution is given.

IV.1.4.1. Convergence History

Each subsonic case was initialized using the reference flow conditions given in Table IV.2. The solution was then computed using the desired solver options until the residuals dropped five orders in magnitude. Figure IV.3 shows the density convergence histories for each case and grid. The convergence of the other variables were found to be quantitatively similar to density and were therefore not shown.

The convergence of the modified MLP and the new Venkatakrishnan limiter solutions mirrored the convergence of their predecessors for the majority of the simulations on each grid. Differences between the residual convergence of the $-\epsilon_1$ and $-\epsilon_2$ solutions became apparent as the solution neared the convergence criterion. On the medium grid (Fig. IV.3b), the Venka- ϵ_1 solution experienced numerical fluctuations near the end of the simulation while the Venka- ϵ_2 solution convergence

remained smooth throughout. The MLP- ϵ_1 limiter solutions displayed stalled convergence trends on both the fine and super-fine grids, whereas the MLP- ϵ_2 solutions converged cleanly in both cases.

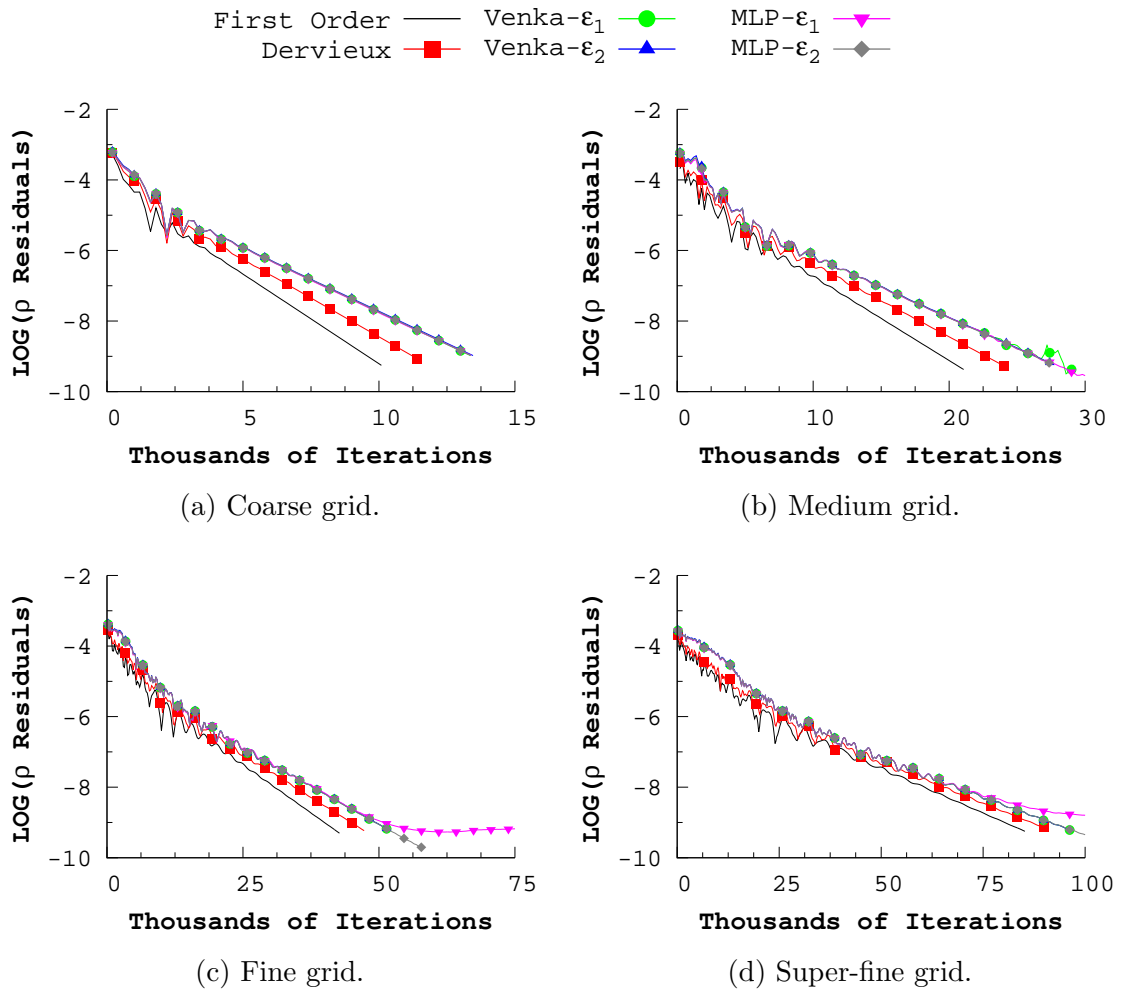


Figure IV.3: Subsonic convergent-divergent nozzle density residual histories.

IV.1.4.2. Center-line Solution

The computed Mach numbers along the center-line were compared against the analytic solution for each case and grid, as shown in Fig. IV.4. The predicted stream-wise variations in Mach number showed good qualitative agreement with the analytic solution, with the agreement growing stronger as the grid was refined. The best quantitative agreement between prediction and the analytic solution was found using the Venkatakrishnan and MLP limiters. No discernible differences were seen between the $-\epsilon_1$ and $-\epsilon_2$ limiter solutions on any grid. The Dervieux limiter gave better results than when the solver was run first-order, but not as good as the other limiters tested.

The differences between the $-\epsilon_1$ and $-\epsilon_2$ solutions were computed using (4.2) for the Mach number and the three thermodynamic state variables along the center-line. The RMS of the differences were then computed using (4.3) to quantify the difference in solutions for a given grid. Figure IV.5 illustrates the variation in solution differences for both the Venkatakrishnan and MLP limiters.

The largest differences occurred on the coarse grid, and were on the order of $10^{-1}\%$ for both limiters. The computed solution differences decreased as the grid size was increased. The rate at which the differences decreased was, on average, consistent with second-order, as illustrated in Fig. IV.5.

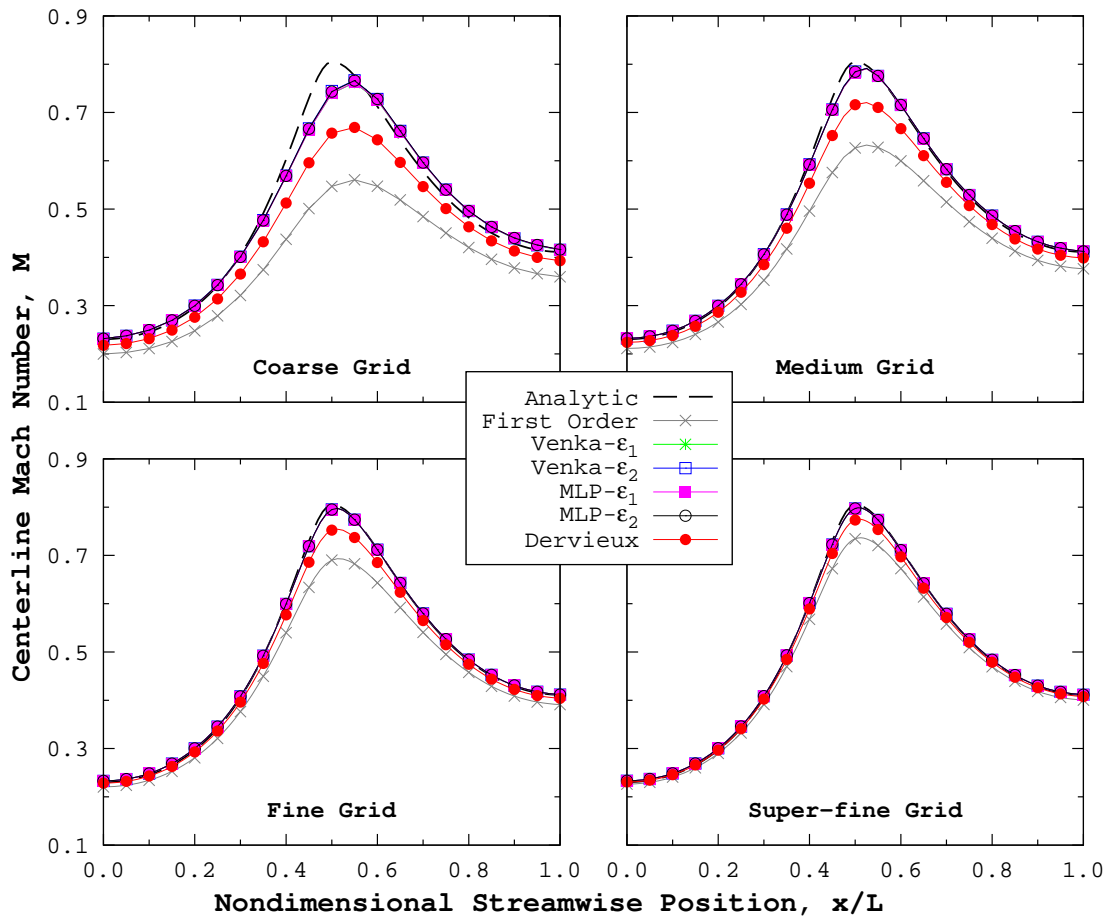


Figure IV.4: Subsonic convergent-divergent nozzle center-line Mach number profiles for each grid.

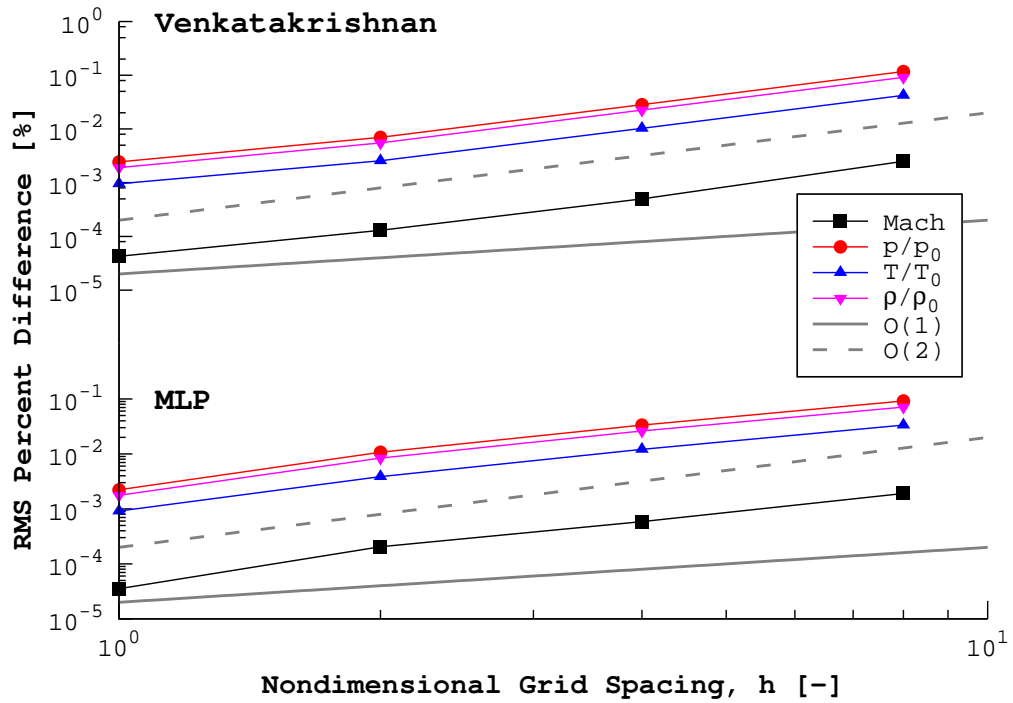


Figure IV.5: Subsonic convergent-divergent nozzle percent difference between solutions computed using the $-\epsilon_1$ and $-\epsilon_2$ variants of the Venkatakrishnan (top) and MLP (bottom) limiters.

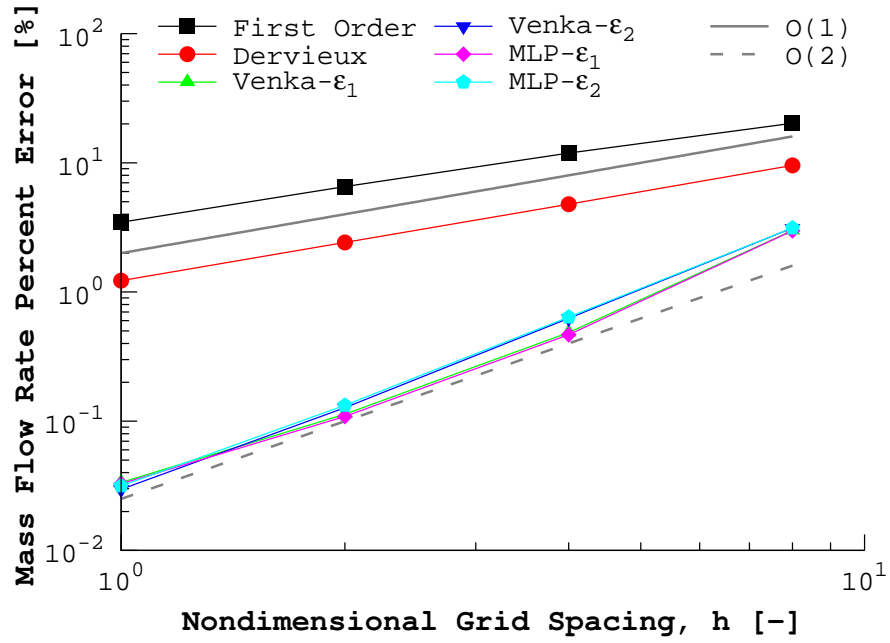
IV.1.4.3. Solution Order of Accuracy

Computed values of the mass flow rate at the nozzle throat and exit were used to determine the apparent solution order of accuracy. An absolute percent error between the computed and analytic values of mass flow rate was computed using the method outlined in Section IV.1.3. The analytic mass flow rate for the given flow conditions was $\dot{m}_a = 0.1538$ kg/s.

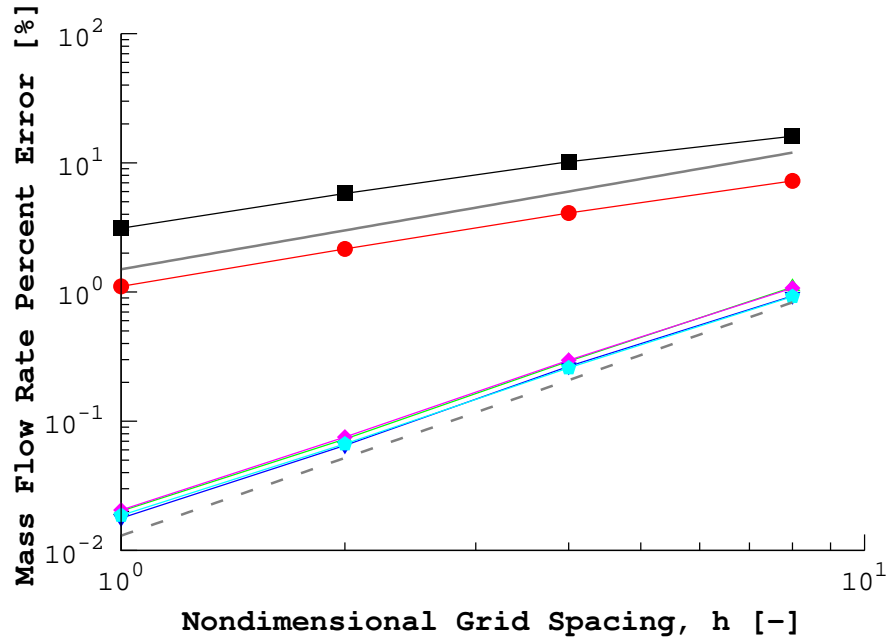
The mass flow rate error decreased with an increase in grid size for all six cases at both streamwise locations, as illustrated in Fig. IV.6. Both the first-order solution and the second-order Dervieux limiter solution exhibited a first-order reduction in the mass flow rate error at both streamwise locations. The rate at which the error decreased for the solutions computed using each versions of the remaining limiters was found to be consistent with second-order.

Table IV.3 presents the apparent solution orders, computed using (4.4), for all six subsonic nozzle cases. Better than second-order accuracy was obtained when using either versions of the Venkatakrishnan and MLP limiters. Using the $-\epsilon_2$ versions of the Venkatakrishnan and MLP limiters resulted in higher apparent orders than when using their predecessors. Using the Dervieux limiter resulted in an apparent solution order consistent with first-order accuracy.

The apparent solution orders were lower at the nozzle exit than at the throat for all cases examined, as evident in Table IV.3. The lower solution orders at the nozzle exit are due to the proximity of the outlet boundary condition. The outlet boundary condition, as with all of the other boundary conditions, is implemented in a first-order manner. As a result, the order of the solution near any boundary will



(a) Nozzle Throat



(b) Nozzle Exit

Figure IV.6: Subsonic convergent-divergent nozzle mass flow rate percent error variation with grid size. “O(1)” and “O(2)” represent first- and second-order convergence rates, respectively.

be reduced towards first-order.

Table IV.3: Subsonic convergent-divergent nozzle apparent solution orders.

	First-Order	Dervieux	Venka- ϵ_1	Venka- ϵ_2	MLP- ϵ_1	MLP- ϵ_2
Nozzle Throat	0.80	0.98	2.22	2.36	2.24	2.32
Nozzle Exit	0.72	0.88	2.06	2.09	2.01	2.00
Average Value	0.76	0.93	2.14	2.22	2.12	2.16

IV.1.5. Normal Shock Flow

The section presents the results of the six aforementioned cases for the flow through a convergent-divergent nozzle with a normal shock present. The convergence histories of each case are presented first. Secondly, some numerical “shock instabilities” that were observed are discussed. The solutions along the center-line of nozzle are compared. Finally, the apparent orders of accuracy for each case are presented.

IV.1.5.1. Convergence History

The converged subsonic flow results were used as the initial solution for their respective normal shock flow cases. The flow solver was run in an identical fashion to the subsonic cases, with only the outlet pressure being modified according to Table IV.2. The “five order of magnitude” convergence criterion was applied to the

cases herein.

The residual convergence for the solutions which utilized the $-\epsilon_2$ limiter variants were qualitatively identical to that of the $-\epsilon_1$ limiter variant residual histories, as shown in Fig. IV.7. Only the density residuals are shown in Fig. IV.7 due the similarity of the other variable residuals. The residual convergence for solutions using either MLP limiter were seen to stall on the three finest grids. Convergence stall was also exhibited by both Venkatakrisshnan limiter solutions on the super-fine grid. In all cases in which residual convergence stall occurred, the $-\epsilon_2$ limiters resulted in marginally lower residual values. Solutions using the Dervieux limiter were seen to converge on all grids.

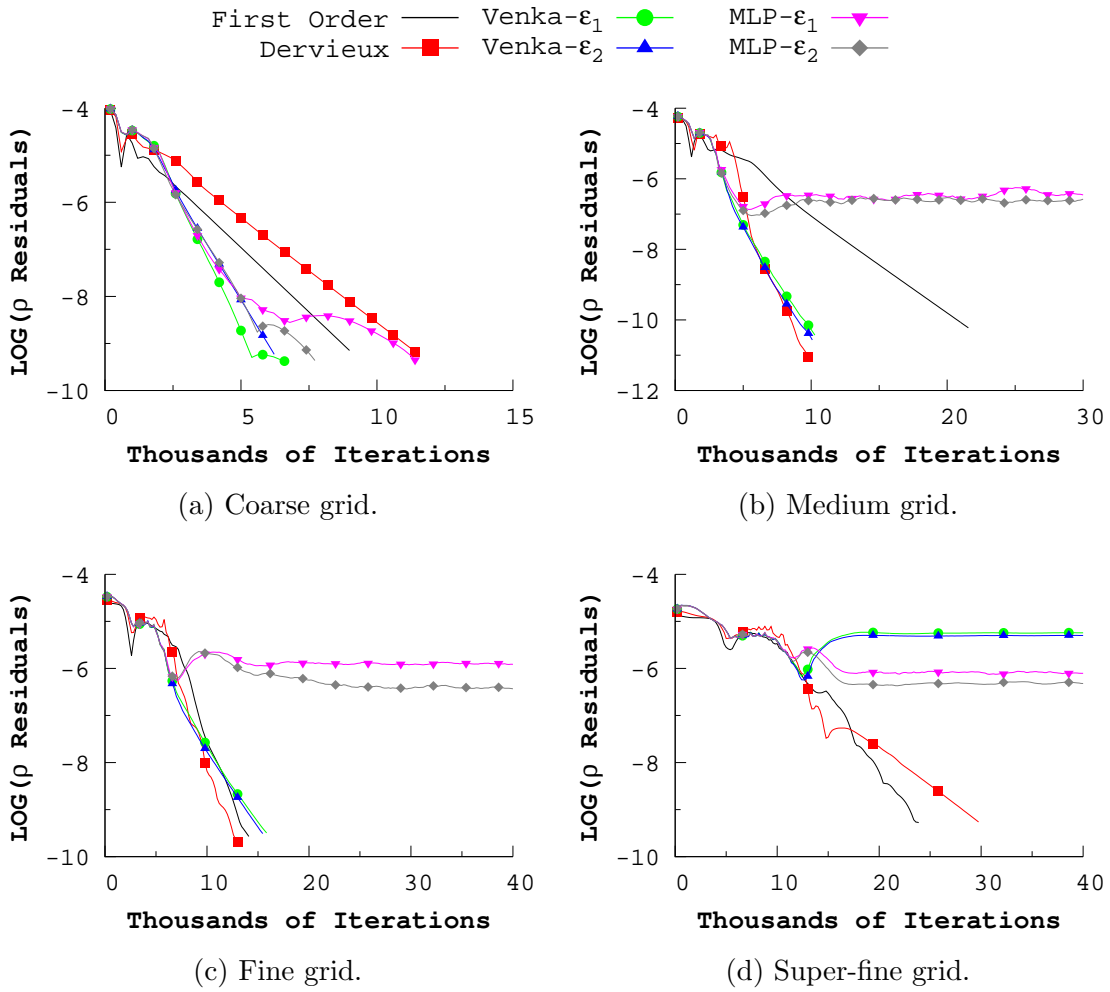


Figure IV.7: Normal shock convergent-divergent nozzle density residual histories.

IV.1.5.2. Numerical “Shock Instabilities”

The Venkatakrishnan constant, defined as C in (3.14), was increased for the Venka- ϵ_1 solution on the super-fine grid in an effort to obtain solution convergence. The value of C was increased from its default value of five until the solution residuals converged. It was found that the solution would converge for $C \geq 175$, as shown in Fig. IV.8. It should be noted that $C = 175$ is not necessarily the minimum value of

C which will result in residual convergence, but only the first value for which residual convergence was observed.

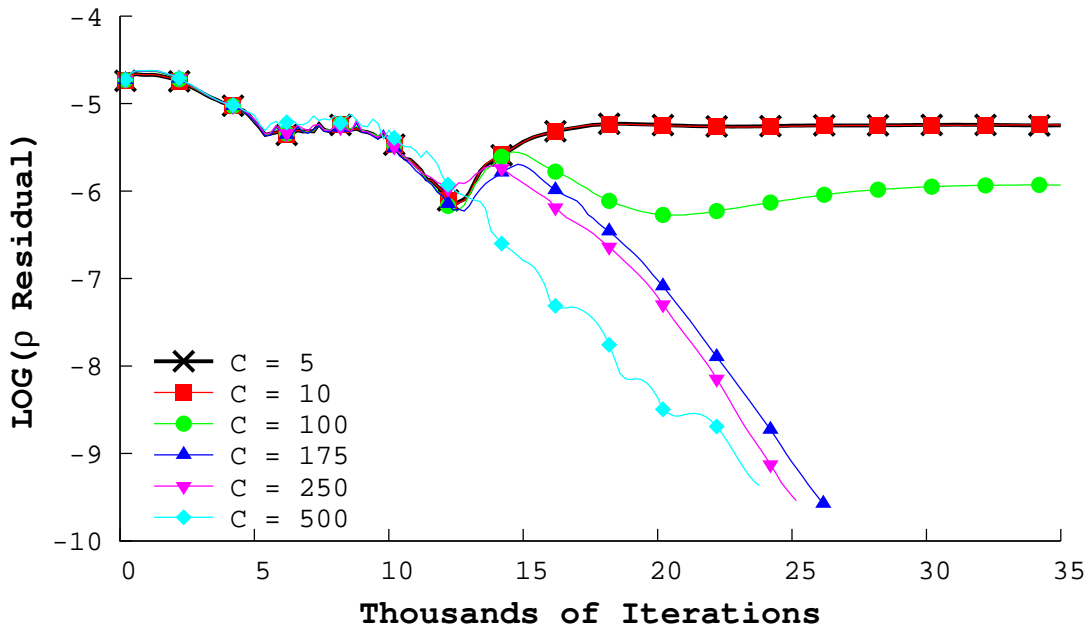


Figure IV.8: Normal shock convergent-divergent Venka- ϵ_1 density residual histories on the super-fine grid for increasing limiter constant values.

Contours of the Mach number were extracted along the $z = 0$ plane for solutions which used three different values of C , as shown in Fig. IV.9a. For the default value of $C = 5$, the contours revealed a smooth solution leading up to the normal shock and a “streaky” solution downstream of the shock. The smoothness of the downstream solution was improved as C was increased until the contours became completely smooth at $C = 175$.

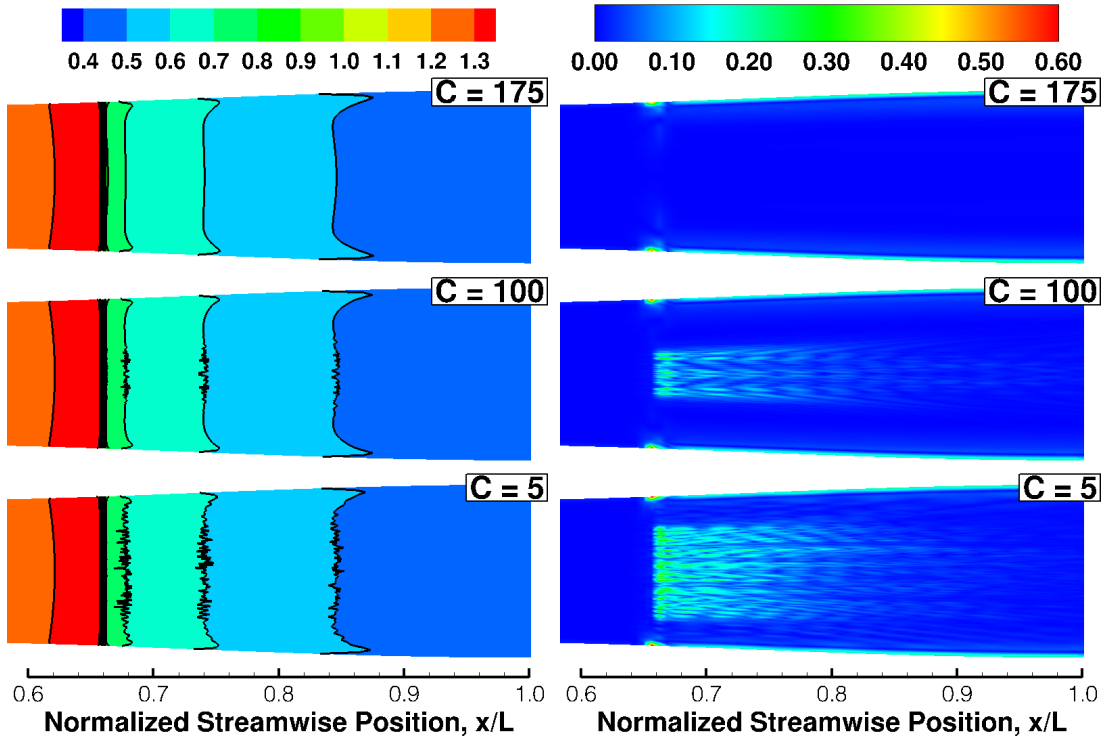
The solution was further examined by extracting contours of maximum-normalized

vorticity magnitude along the same plane, as illustrated in Fig. IV.9b. Large amounts of vorticity was seen to be emanating from and streaking downstream of the normal shock for the case which used the default C value. These streaks of vorticity have been previously used to characterize a numerical shock instability by Kitamura et al. [2009], referred to herein as the “streaking vorticity shock instability”².

It was found that increasing C had the effect of reducing the amount of vorticity that was generated by the normal shock, as shown in Fig. IV.9b. The converged solution, $C = 175$, did not exhibit an vorticity streaks downstream of the shock. From this result, it was concluded that the stall residual convergence observed for the Venka- ϵ_1 solution on the super-fine grid was the result of the “streaking vorticity shock instability”, and that increasing C was a viable method of eliminating the instability.

The “streaking vorticity shock instability” was also found in the MLP- ϵ_1 and MLP- ϵ_2 solutions on the medium, fine, and super-fine grids. Starting with the medium grid, simulations with the MLP- ϵ_1 limiter were computed with increasing values of the limiter constant, C . A value of 300 was required for the solution residuals to converge. As the limiter constant was increased the character of the stalled residual convergence changed, as shown in Fig. IV.10. The pattern of the stalled residual history went from being flat to being sinusoidal, which indicated that a different cause for the stalled converged was likely.

²Referred to as the “Stage 2” shock instability by Kitamura et al. [2009].



(a) Mach number.

(b) Maximum-normalized vorticity magnitude.

Figure IV.9: Normal shock convergent-divergent nozzle Venka- ϵ_1 solutions on the super-fine grid for different values of the limiter constant.

No streaking vorticity was seen to be emanating from the normal shock for the cases with the sinusoidal residual histories. Instead, it was found that the position of the shock was periodically changing. The shock position was seen to move downstream one grid layer before returning to its original position. The retreat of the shock to its original position occurred at the minimum point of the residual oscillation. From this result, it was concluded that the stalled, flat residual history corresponded to the “streaking vorticity shock instability” and the sinusoidal residual

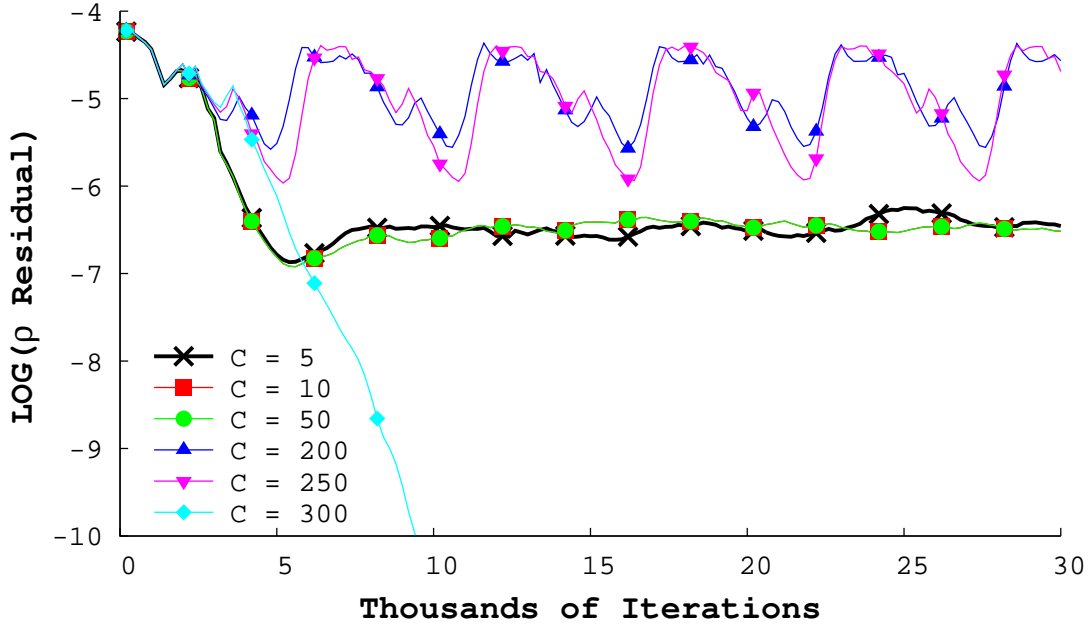


Figure IV.10: Normal shock convergent-divergent MLP- ϵ_1 density residual histories on the medium grid for increasing limit constant values.

history corresponded to a numerical “moving shock instability”.

Similarly with the medium grid solution, the limiter constant was increased for the MLP- ϵ_1 solutions on the fine and super-fine grids until convergence was achieved. Figure IV.11 shows the resultant unstable and stable points for each grid, where a stable point was defined as a point for which the residuals converged. The “moving shock instability” was observed on the fine grid for large values of C , but not on the super-fine grid. A very large value of the limiter constant, $C = 900$, was required for convergence on the fine grid.

The use of large limiter constants with the MLP- ϵ_1 limiter to achieve convergence was not without consequence. The converged solutions on the medium and fine grids

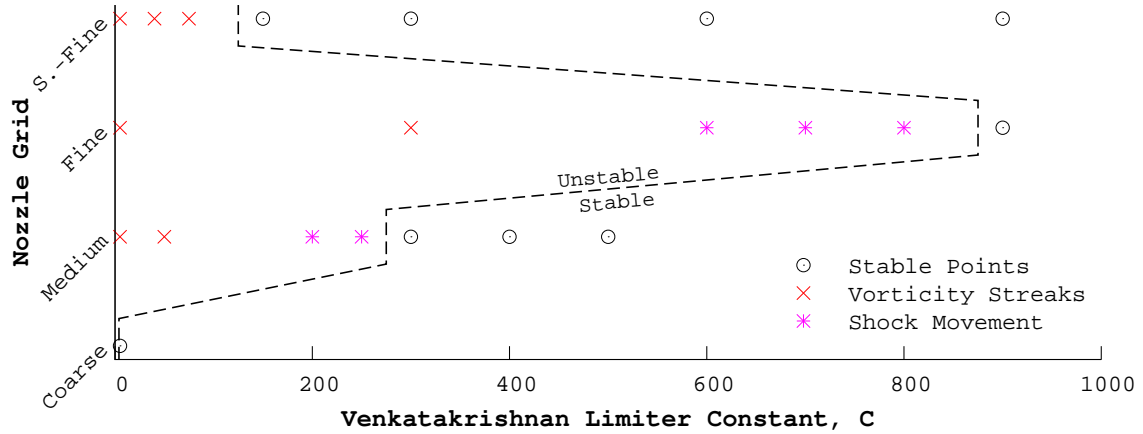


Figure IV.11: Normal shock convergent-divergent MLP- ϵ_1 numerical “shock instability” map for different values of the limiter constant. Convergence was achieved at “stable” points.

exhibited a large amount of overshoot at the front of the normal shock, as shown in Fig. IV.12a. The position of the normal shock was also shifted downstream, as far as one cell length in the case of the medium grid. Examining the contours of the MLP- ϵ_1 limiter function, as defined by Φ_i^M in Section III.2.2, showed that the large limiter values resulted in unlimited solutions on the medium and fine grids. The MLP- ϵ_1 limiter function for density can be seen in Fig. IV.12b for the finest grids, illustrating the unlimited nature of the solutions. The MLP- ϵ_1 solution on the super-fine grid remained limited due to a higher grid density.

IV.1.5.3. Center-line Solution

Figure IV.13 compares the streamwise variation of the Mach number computed on each grid for the six stated cases against the analytic solution. Solutions shown

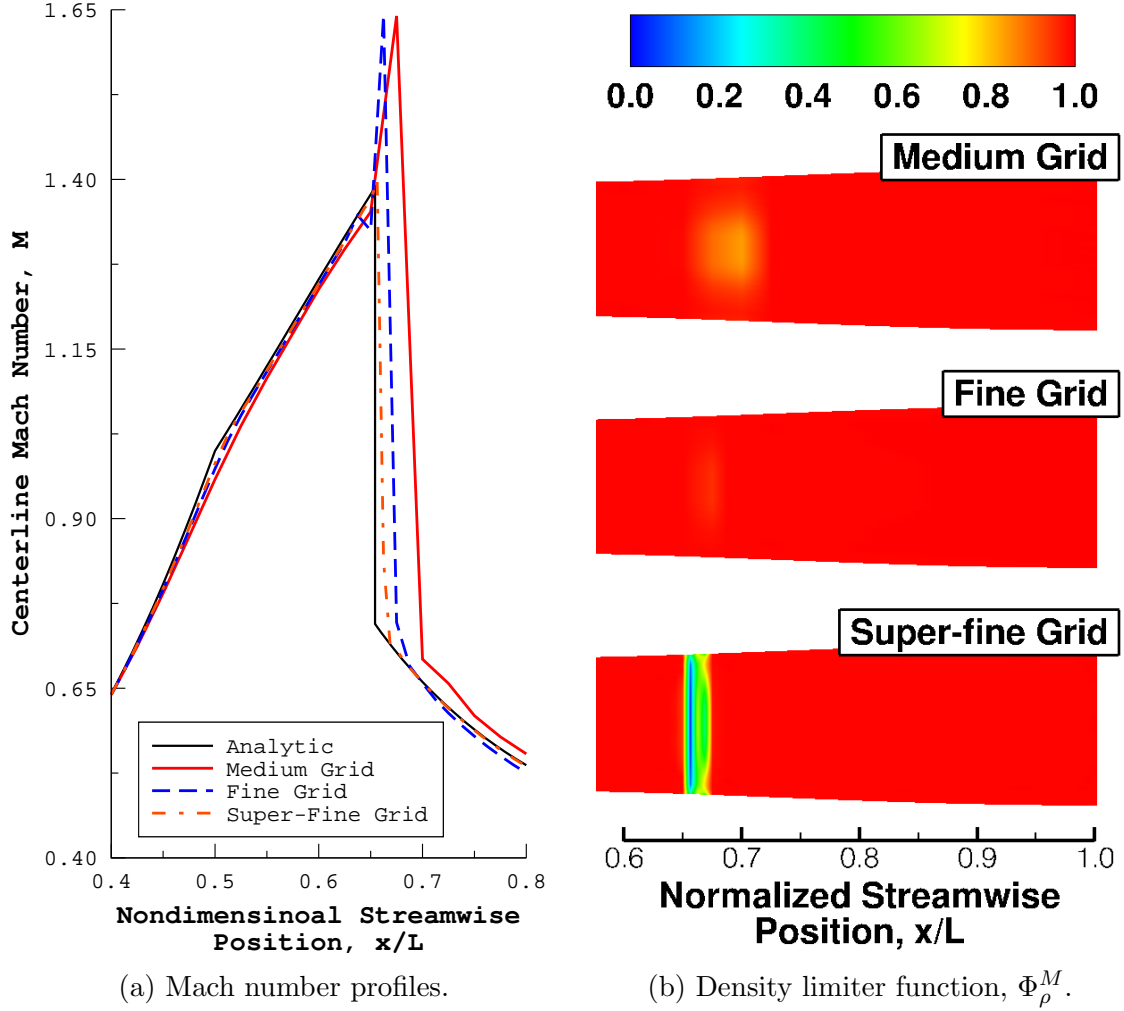


Figure IV.12: Examination of the converged MLP- ϵ_1 normal shock convergent-divergent nozzle solutions. $\Phi_\rho^M = 0$ and 1 correspond to full and no limiting, respectively.

for the Venka- ϵ_1 and MLP- ϵ_1 limiters were computed using the default value of C to prevent solution unlimiting. The qualitative agreement between prediction and the analytic solution improved as the grid size was increased.

No noticeable difference was observed between solutions computed using the Venka- ϵ_1 and Venka- ϵ_2 limiters. The story was also the same for both versions of the MLP limiter except on the Fine mesh. On the fine mesh the MLP- ϵ_1 and MLP- ϵ_2 solutions bracketed the analytic solution downstream of the shock. Even though they showed good agreement with the analytic solution, the Dervieux limiter solutions exhibited both over- and under-shoot in the vicinity of the normal shock.

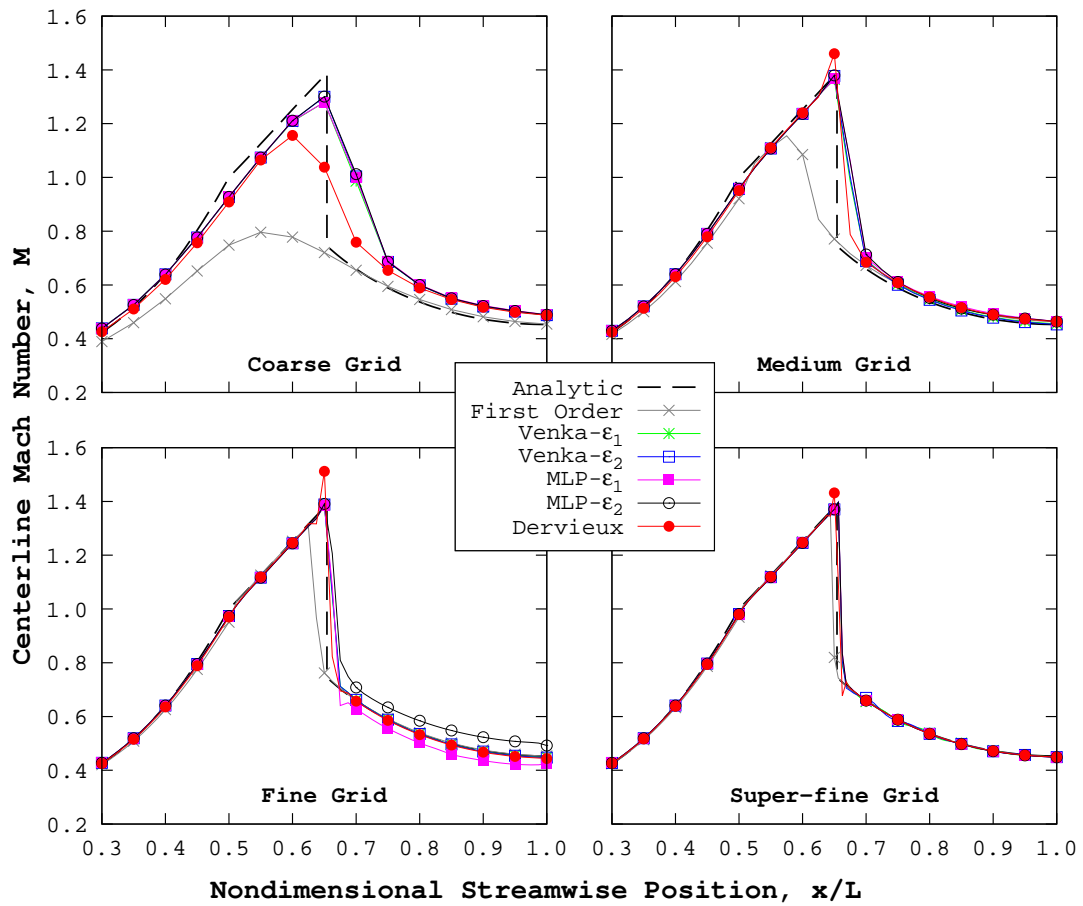


Figure IV.13: Normal shock convergent-divergent nozzle center-line Mach number profiles for each grid.

IV.1.5.4. Solution Order of Accuracy

Apparent solution orders were computed for the normal shock cases in the same manner as was done for the purely subsonic cases in Section IV.1.4.3. Values of the mass flow rate at the nozzle throat and exit were used from the three finest grids. Two sets of solution orders were computed for the Venka- ϵ_1 and MLP- ϵ_1 limiter solutions. In the first, only solutions which used the default limiter constant value ($C = 5$) were considered. For the second set, the solution orders were computed from the solutions with converged residual histories ($C > 5$).

The computed apparent solution orders are shown in Table IV.4. All of the limiters tested resulted in apparent solution orders consistent with second-order accuracy, regardless of whether the residuals converged or not. Solution orders that were computed using mass flow rate values from stalled convergence cases were lower than when all three cases converged, even when solution un-limiting occurred. The Venka- ϵ_2 limiter resulted in the highest solution order even with the solution convergence stalling on the super-fine grid.

Table IV.4: Apparent order of the normal shock solutions using the finest three grids. Values in parenthesis were computed using mass flow rates from stalled convergence solutions.

	First-Order	Dervieux	Venka- ϵ_1	Venka- ϵ_2	MLP- ϵ_1	MLP- ϵ_2
Nozzle Throat	1.05	1.29	2.19 (2.13)	(2.12)	2.20 (2.14)	(2.15)
Nozzle Exit	–	2.01	1.66 (1.30)	(1.76)	1.36 (1.06)	(1.16)
Average Value	1.05	1.65	1.92 (1.71)	(1.94)	1.78 (1.60)	(1.66)

IV.2. NACA 0012

This section presents the inviscid results for a NACA 0012 airfoil simulated at supersonic freestream conditions. The freestream Mach number was 2, and the airfoil was set at 0° angle-of-attack. This particular case was chosen as it was shown by Venkatakrishnan [1995] to exhibit stalled convergence of the residuals. The purpose of these simulations was to further examine the effectiveness of the Park & Kim modification to the Venkatakrishnan and MLP limiters.

IV.2.1. Airfoil Definition and Grid Generation

A modified thickness profile was used to define the airfoil shape. The thickness profile is given by [Fransson & Verdon, 1993, p. 880]

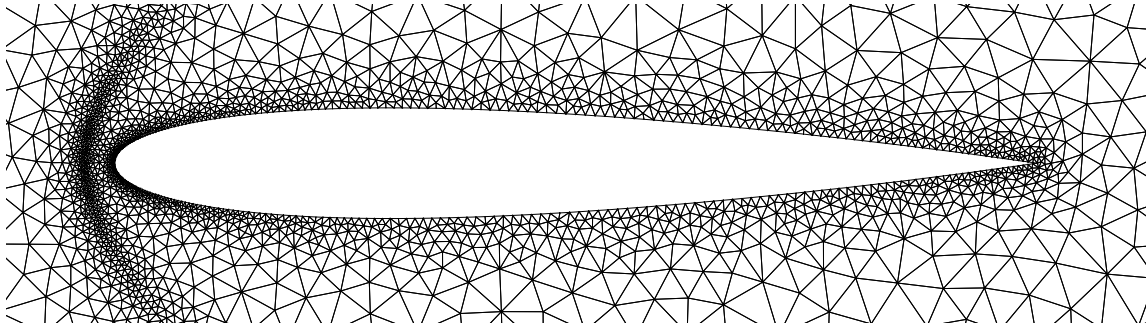
$$\pm y_t = \frac{t}{0.2} (0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4) \quad (4.5)$$

where t is the maximum thickness as a percentage of the chord, and $0 \leq x \leq 1$. The modification of the standard NACA equation is the coefficient of the quartic term. The modification used creates a sharp trailing edge, as opposed to the blunt trailing edge that results from the standard equation. For reference, the standard quartic coefficient is 0.1015 [Abbott & Doenhoff, 1959, p. 113].

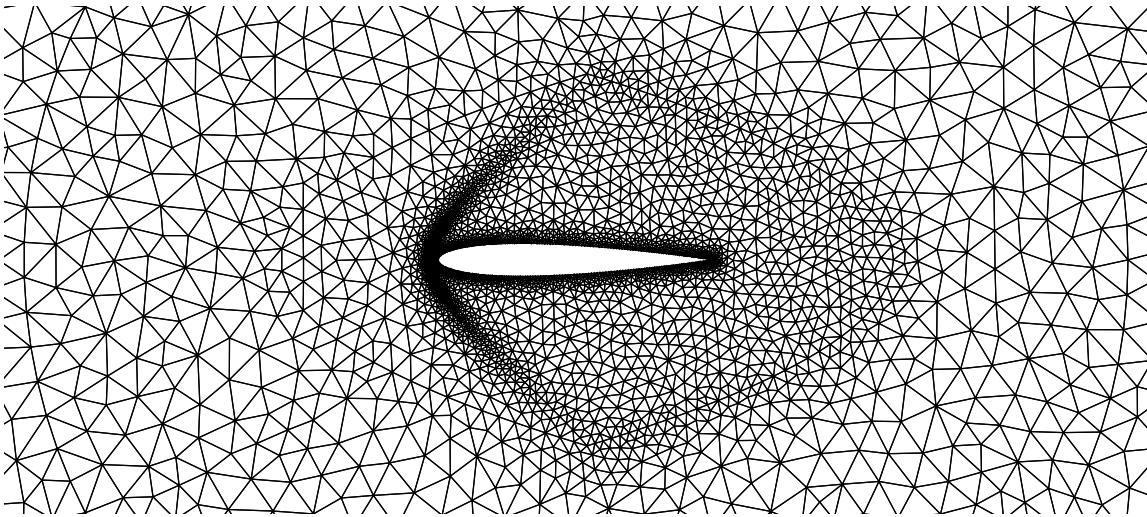
An unstructured grid was constructed using the commercially available grid generation software called Pointwise [Pointwise, 2011]. The grid was constructed by first generating a two-dimensional grid of unstructured triangles and then extruding this layer a set distance. The resultant three-dimensional grid is made up of six node prismatic elements.

The two-dimensional domain boundaries were placed fifty chords away from the airfoil in all directions. Three hundred points were used to define the airfoil, with nodes clustered at the leading and trailing edges. as illustrated in Fig. IV.14a. The freestream domain was subdivided primarily by concentric circles about the centroid of the airfoil. Points were clustered in the anticipated location of the bow by subdividing the inner-most circular region using a Bezier curve, as illustrated in Fig. IV.14b. The location of the shock clustering was refined by simulating the flow, find the location of the shock, and manually moving the grid points to match the shock location.

The unstructured two-dimensional grid was extruded once in the z -direction to create a three-dimensional grid that was one cell thick. The thickness of the cell was defined to be 10% of the chord. The resultant three-dimensional grid contained 10,586 nodes.



(a) Airfoil detail.



(b) Shock clustering detail.

Figure IV.14: NACA 0012 unstructured grid.

IV.2.2. Boundary Conditions and Solution Strategy

Figure IV.15 illustrates the different boundary condition types, and their locations, that were applied to each two-dimensional layer. A symmetry boundary condition was applied to the grid faces on the z_{min} and z_{max} planes.

The freestream conditions were set using standard air at sea level. The desired freestream Mach number for this case was $M_\infty = 2$. The inlet total pressure and

temperature were computed isentropically from the freestream pressure, temperature, and Mach number. The outlet static pressure was defined to be the freestream pressure. The inlet and outlet boundary conditions are summarized in Table IV.5.

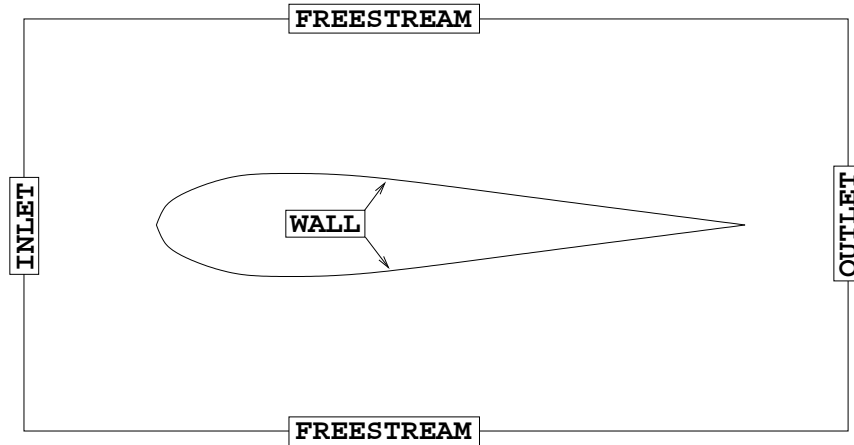


Figure IV.15: Boundary conditions applied to the NACA 0012 flow domain. Symmetry boundaries are applied to the out of plane sides.

Table IV.5: NACA 0012 flow conditions.

Freestream Conditions	Inlet Conditions	Outlet Conditions
$p_\infty = 101325.0 \text{ Pa}$	$p_0 = 792812.3 \text{ Pa}$	$p_e = 101325.0 \text{ Pa}$
$T_\infty = 288.3 \text{ K}$	$T_0 = 518.9 \text{ K}$	
$M_\infty = 2.0$		

Solutions were computed with second-order accuracy using both variants of the

Venkatakrishnan and MLP limiters³. Appendix D, starting on p. 250, contains a full accounting of the flow solver options used for this case. Each simulation was started from a uniform flow field and run for 10,000 iterations. The initial flow field was set using the freestream conditions defined in Table IV.5.

IV.2.3. Convergence History

The solution residuals for each limiter solution pair were compared to examine the effect of the Park & Kim limiter constant modification on the solution. It was found for the Venkatakrishnan limiter that there was virtually no difference in the residual histories, as shown by Fig. IV.16. Although there is no improvement to the convergence of the residuals, it is important to note that there was not a negative impact seen when using the Venka- ϵ_2 implementation.

The story is a bit different for the MLP limiters. The residuals were seen to drop approximately four orders of magnitude before stalling when using the MLP- ϵ_1 implementation with the default constant, $C = 5$. However, a clear improvement was seen when using the MLP- ϵ_2 limiter, as shown by Fig. IV.17. The MLP- ϵ_2 limiter allowed the solution residuals to converge nearly eight orders of magnitude without stalling.

Both MLP solutions were re-computed with snapshots of the flow being extracted every 25 iterations in an effort to determine the cause of residuals stalling. The pressure was normalized by the freestream pressure ($p_{ref} = 101325$ Pa), and was sampled at a point, seen in Fig. IV.18, just upstream of the leading edge.

³Refer back to Section IV.1, page 63 for a description of the limiters.

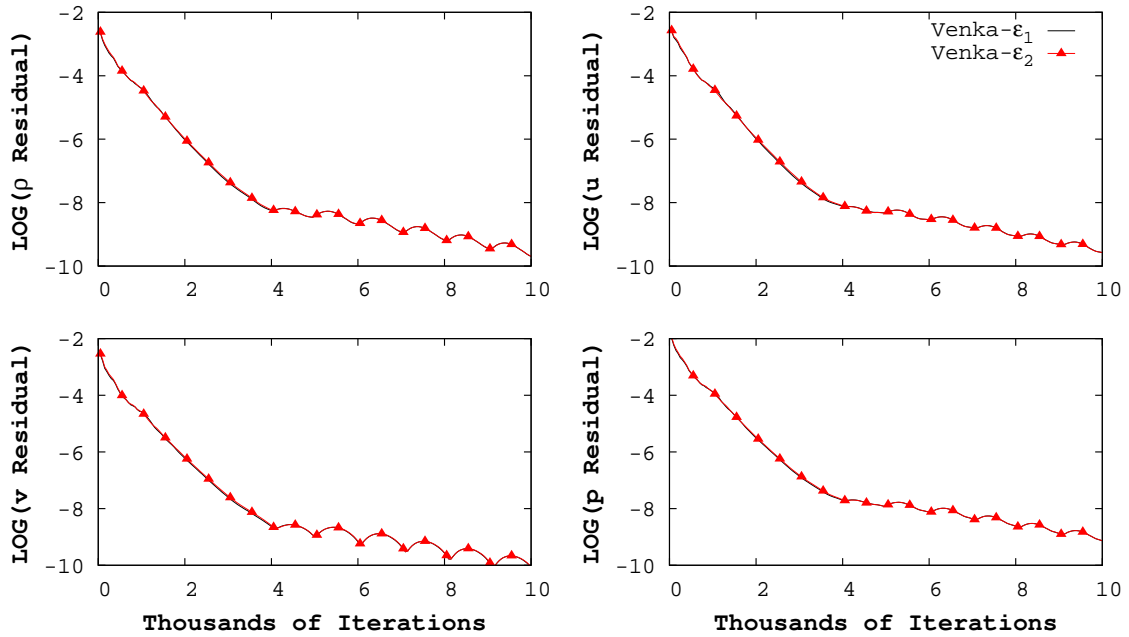


Figure IV.16: Supersonic NACA 0012 residual convergence histories with the Venkatakrishnan limiters.

It was found from examining animations of the solution contours that the position of the bow shock computed using the MLP- ϵ_1 limiter was not stable, exhibiting the numerical “moving shock instability” seen in Section IV.1.5.2. The shock solution determined by the MLP- ϵ_2 limiter proved to be stable with no visible movement of the shock. The sampled pressure histories from both solutions corroborate this conclusion.

Figure IV.19 shows the extracted normalized pressure histories for both the MLP- ϵ_1 and MLP- ϵ_2 limiters, and compares them to their respective residual histories. The figure shows that the pressure at the sample point has a periodic oscillatory behavior when using the MLP- ϵ_1 limiter with the default value for C . In addition to

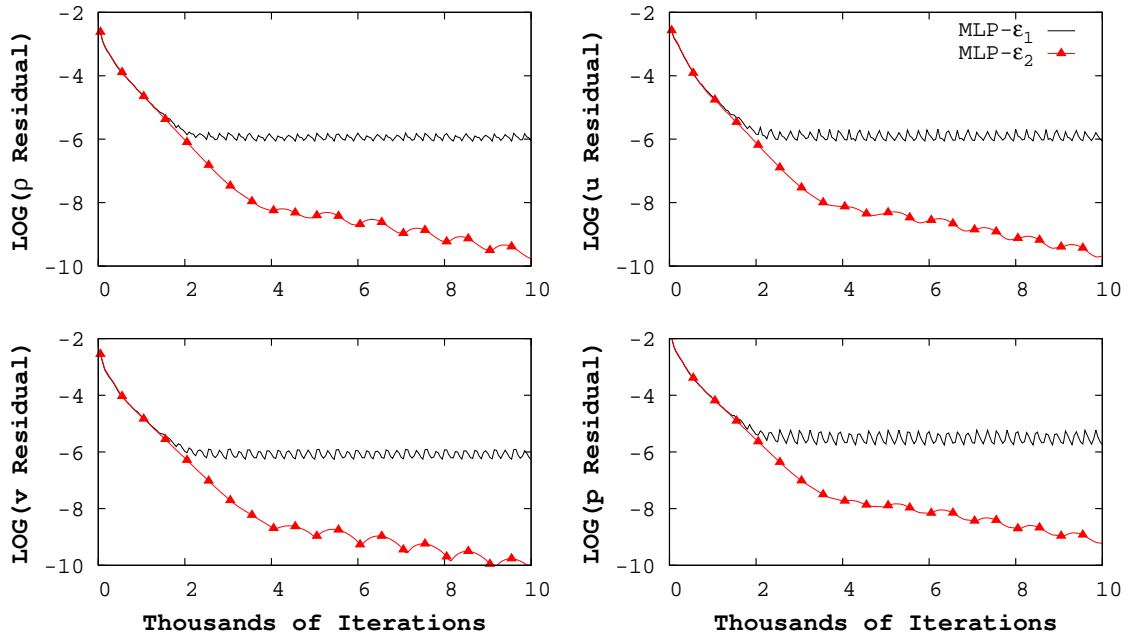


Figure IV.17: Supersonic NACA 0012 residual convergence histories with the MLP limiters.

indicating shock movement, the oscillatory behavior of the pressure prevented local solution convergence all around the airfoil. The lack of local convergence caused the solution residuals to stall.

It was found that convergence could be achieved with the MLP- ϵ_1 limiter by increasing the limiter constant to $C = 6$ from the default value of five, as shown in Fig. IV.19. The shock instability was seen to be present early on in the solution before being suppressed by the limiter function. Figure IV.19 also shows that evolution of the stable MLP- ϵ_2 solution is much smoother than the MLP- ϵ_1 solutions, as no shock instabilities seem to be present at any stage of the solution calculation.

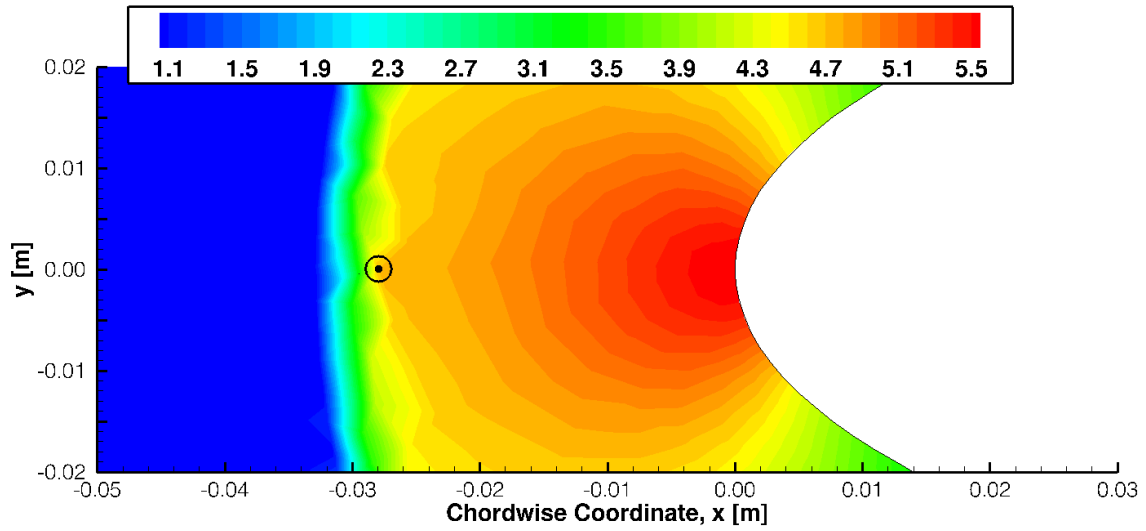


Figure IV.18: Normalized MLP- ϵ_1 pressure contours after 10,000 iterations. The numerical probe is located at the location indicated by “ \odot ”.

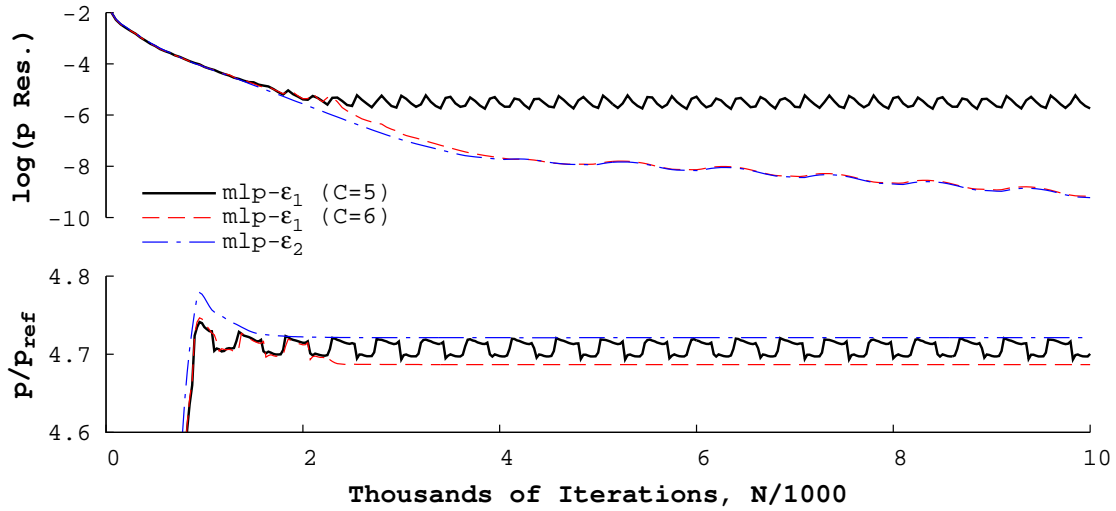


Figure IV.19: Sampled normalized pressure histories compared against their corresponding residual histories. The sampling point is shown in Fig. IV.18.

IV.2.4. Flow Field Comparison

All four solutions were found to be very similar both qualitatively and quantitatively. Figure IV.20 shows the Mach number contours for each case. The figure shows few differences between the solutions generated with the different limiters. The noticeable differences occur just upstream of the shock in the form of small “pockets” of higher Mach. However, the difference in the Mach number between these “pockets” and the surrounding flow is very small, roughly $\Delta M \approx 0.001$.

The drag coefficient, C_D , was used to quantitatively compare the four solutions. The drag coefficient was defined in the typical manner as $C_D = D/(qA)$. The drag, D , is due to pressure forces only, and was computed by the flow solver. For the given freestream conditions, the dynamic pressure was $q = 283.5842$ kPa. The reference area was taken as the wall surface area from the computational grid, the value being $A = 0.1$ m². The computed drag coefficients are given in Table IV.6.

The table shows little variation in the computed drag coefficient between solution methods. The largest absolute percent difference was found to be 0.296% between the Venka- ϵ_2 and MLP- ϵ_1 solutions. The absolute percent differences between the limiter options was found to be even smaller. The difference between the Venkatakrishnan limiters was found to be 0.05%, and the difference between the MLP limiters was 0.0992%.

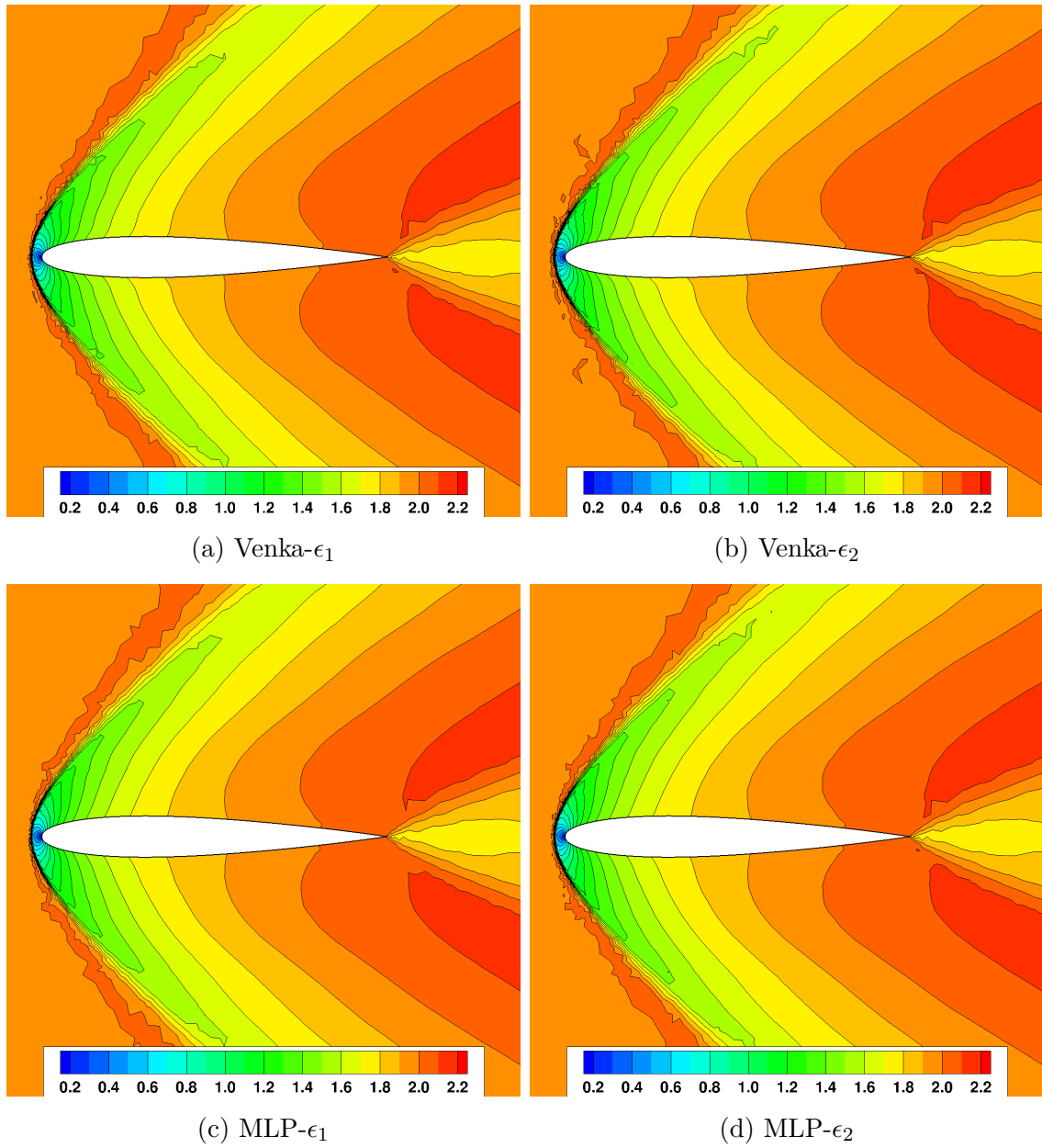


Figure IV.20: NACA 0012 Mach number contours for each limiter solution.

Table IV.6: NACA 0012 drag coefficients.

	Venka- ϵ_1	Venka- ϵ_2	MLP- ϵ_1	MLP- ϵ_2
C_D :	0.092462	0.092416	0.092689	0.092597

IV.3. Turbulent Flat Plate

This section presents the results for turbulent flow over a zero pressure gradient flat plate. Updates to the turbulence model had been made as a part of this work. This case was chosen to test the implementation of these updates due to its simplicity and its wide use in the literature. The particulars for this case were taken from NASA Langley’s *Turbulence Modeling Resource* website [NASA, 2015].

IV.3.1. Domain Definition and Boundary Conditions

The computational domain for the flat plate geometry is given in Fig. IV.21. The length, L , of the flat plate was 2 meters, and the height of the domain was $L/2$. The domain inlet is positioned a distance of $L/6$ upstream of the plate’s leading edge. The width, W , held constant for all four grids at $L/200$. The boundary condition types that were applied to the domain are also indicated in the figure.

The boundary conditions for this case were set such that the freestream Mach number was $M_\infty = 0.2$. To that end, the inlet total conditions were $p_0 = 104190.471$ Pa and $T_0 = 290.455$ K. Standard conditions at sea level were assumed at the outlet. The turbulent length scale and intensity were also set at the inlet, respectively, to $\ell_T|_\infty = 0.1$ m and $I_T|_\infty = 1\%$. Based on these turbulence inlet conditions, the freestream viscosity ratio was $\mu_T/\mu_\infty = 3.3$.

IV.3.2. Computational Grids

The four grids used for this test case were downloaded from the NASA Langley website [NASA, 2013]. Each downloaded grid was a two-dimensional, structured grid.

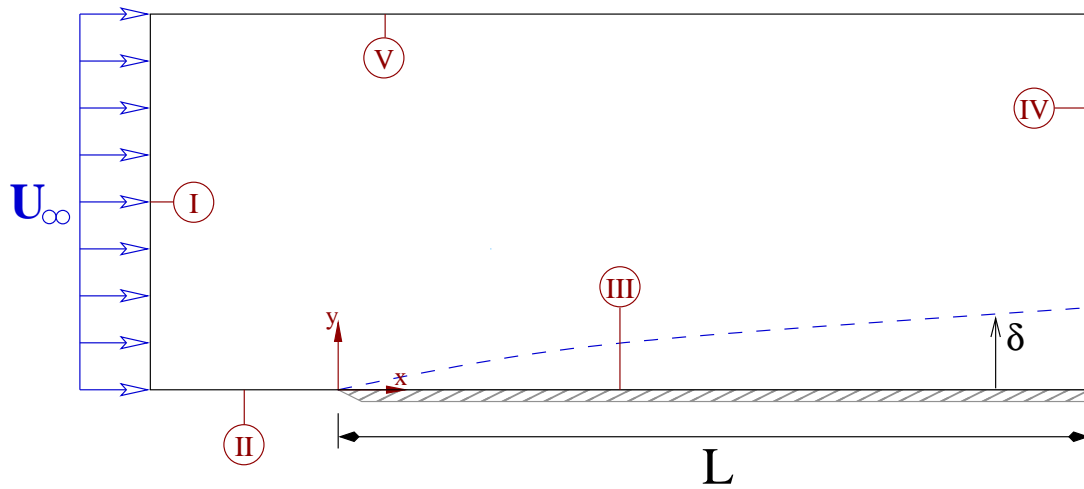


Figure IV.21: Flat plate computational domain. Given boundary conditions are: inlet (I), symmetry (II), viscous wall (III), outlet (IV), and freestream (V).

The grid points were clustered near the leading edge of the plate and in the normal direction to the plate surface. Figure IV.22 illustrates the grid point clustering on the coarsest of the four grids.

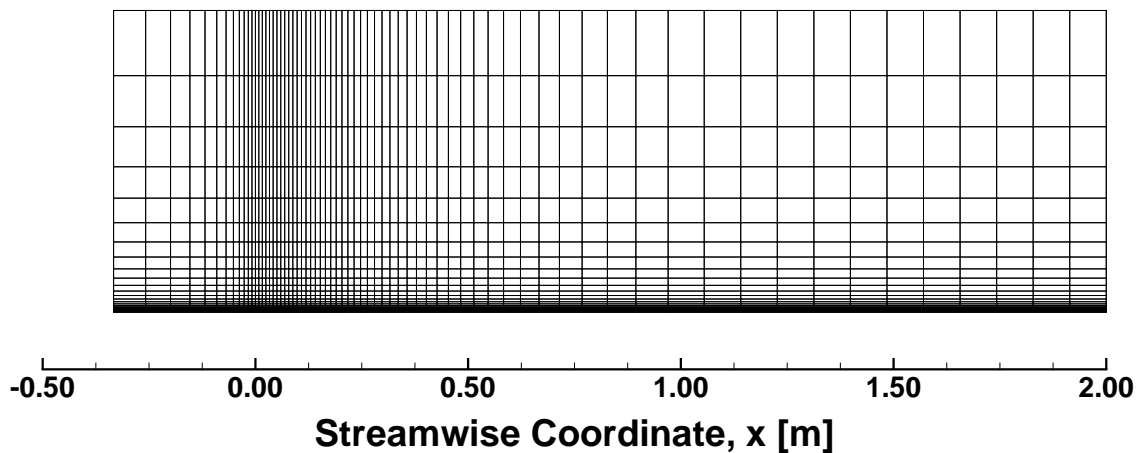


Figure IV.22: Coarse flat plate grid.

The requisite three-dimensional grids for the flow solver were created by extruding the downloaded two-dimensional grids once by $\Delta z = W$ to form a once cell thick grid. A FORTRAN program was written to take the given two-dimensional structured grids and “generate” corresponding three-dimensional unstructured grids.

The sizes of each grid and the height of the first cell adjacent to the plate are presented in Table IV.7. The distance to the first node away from the wall is presented nondimensionally as y^+ numbers, where

$$y^+ = \frac{u_* y}{\nu}. \quad (4.6)$$

The friction velocity needed in (4.6) is given by

$$u_* = \sqrt{\frac{\tau_w}{\rho}}. \quad (4.7)$$

The values for the y^+ number in the table were estimated using Schlichting’s approximations for τ_w [Schlichting, 1968, p. 598] evaluated at the midpoint of the plate, $x = 1$ m.

IV.3.3. *Solution Strategy*

The solution strategy for the turbulent flat plate case was to initialize the flow field using a laminar flow solution before activating the turbulence model. Once the laminar flow was set, κ and ω were uniformly initialized throughout the field using the specified inlet conditions. One thousand first-order simulations were completed first to allow the turbulent solution to settle before switching to second-order accurate. UNS3D was simulated using second-order accuracy, with the basic Venkatakrisnan limiter, until the flow residuals either dropped five orders in magnitude or began to

Table IV.7: Flat plate grid dimensions.

Grid	I _{MAX}	J _{MAX}	K _{MAX}	N _P ^a	y ^{+b}
Coarse	69	49	2	57	0.85
Medium	137	97	2	113	0.42
Fine	273	193	2	225	0.21
Super-fine	545	385	2	449	0.10

^a Number of points on the flat plate in the x-direction

^b Estimated value at $x = L/2$, $Re_x = 4.659 \times 10^6$

level off. Implicit residual smoothing was used for the Navier–Stokes equations, but not for the $\kappa - \omega$ equations. The reasoning behind the residual smoothing choices will be explored at the end of this section. Consult Appendix D, p. 253, for a sample input file for this case showing all the code options used.

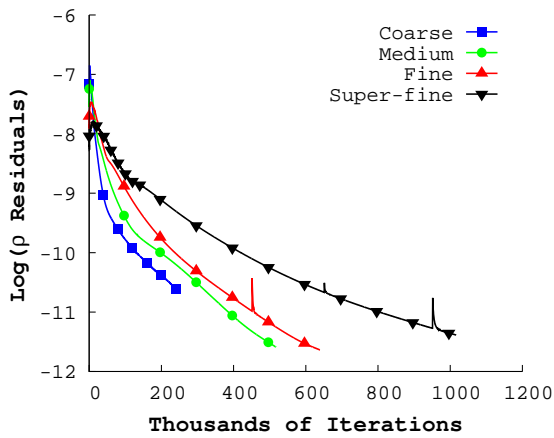
The numerical results to be presented in this section are compared against the experimental data generated by Wieghardt & Tillmann [1951]. The plotted experimental data points were digitized from the paper by Yoder & Georgiadis [1999]. In addition to plotting experimental data for the velocity profiles, the empirical equation for the logarithmic overlap region and the laminar sublayer are plotted. The equation of the logarithmic overlap region is given by $u^+ = \ln(y^+)/0.41 + 5$ [White, 2006, p. 416], and the laminar sublayer is given by $u^+ = y^+$ [White, 2006, p. 419]. In both equations $u^+ = u/u_*$, where u_* was previously defined by (4.7).

IV.3.4. Grid Convergence Study

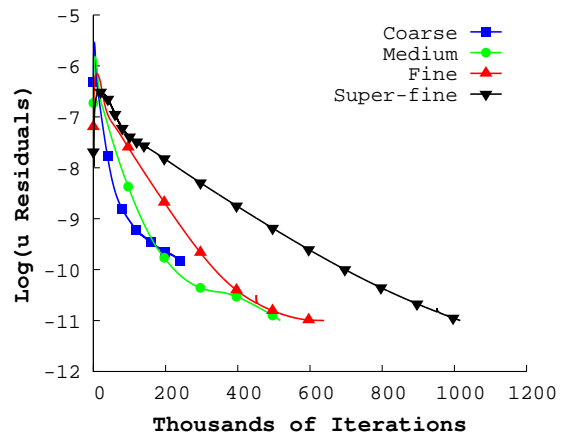
The overall solutions from the four grids were compared against one another before examining the velocity profiles. This was done to choose one of the solutions to examine in more detail. The convergence histories of the solution residuals have been plotted in Figure IV.23. A plot of the w -velocity residual was omitted as the flow is two-dimensional, and the variations in w are very small. The plot shows that the residuals for all of the variables converged steadily, with the overall slope of the convergence decreasing as the grid size was increased. Around one million iterations were required for the super-fine grid to achieve a five order magnitude drop in the flow residuals. One interesting point of fact is that the initial value for the ω residuals is positive. However, the value quickly becomes negative within roughly 1,000 iterations for all four grids.

It was found from looking at the computed distribution of the y^+ number that the estimates found in Table IV.7 were conservative estimates. The estimated values were found to be approximately 20% higher than the computed values at $x = 1$ m. The computed distributions of y^+ are compared against the estimated values in Fig. IV.24. Arithmetic averages of the y^+ number computed for each grid are also given in the figure.

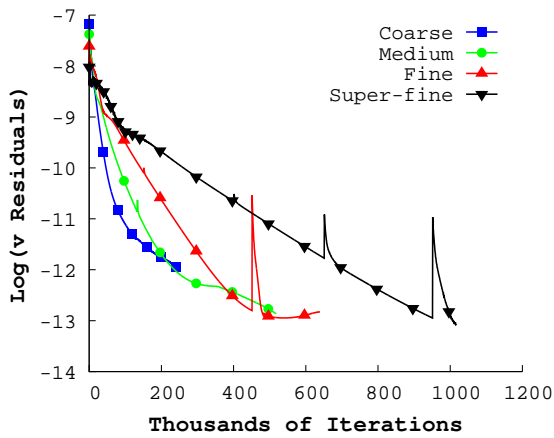
It was found that the reason for the large difference between estimated and computed y^+ numbers was due to the wall shear stress estimates, given by Schlichting's power law formulation, were higher than the computed values. Figure IV.25 shows that Schlichting's estimation of the skin friction coefficient is also a conservative estimate in comparison to the experimental data of Wieghardt & Tillmann [1951]. As



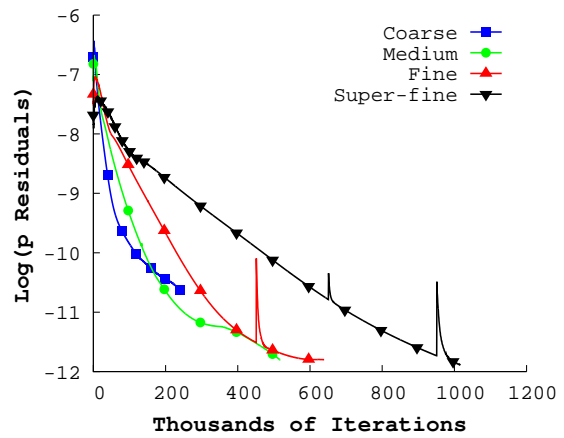
(a) Density residuals.



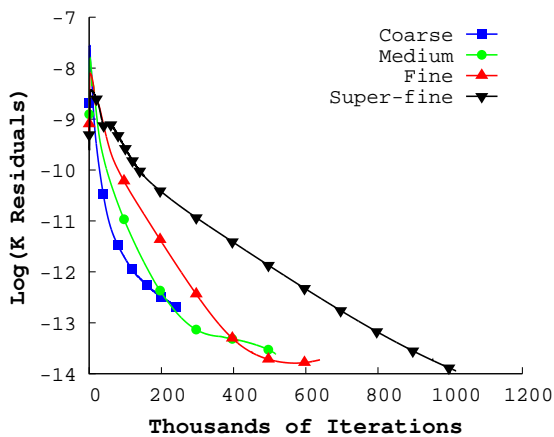
(b) U-velocity residuals.



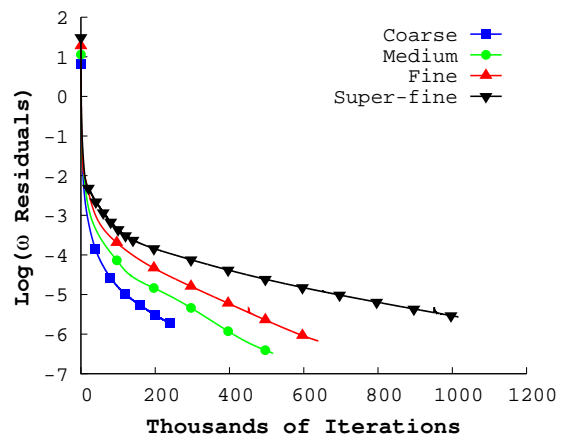
(c) V-velocity residuals.



(d) Pressure residuals.



(e) κ residuals.



(f) ω residuals.

Figure IV.23: Flat plate solution residual comparison.

the figure shows, the computed values of the skin friction coefficient tend towards the experimental data as the grid is refined.

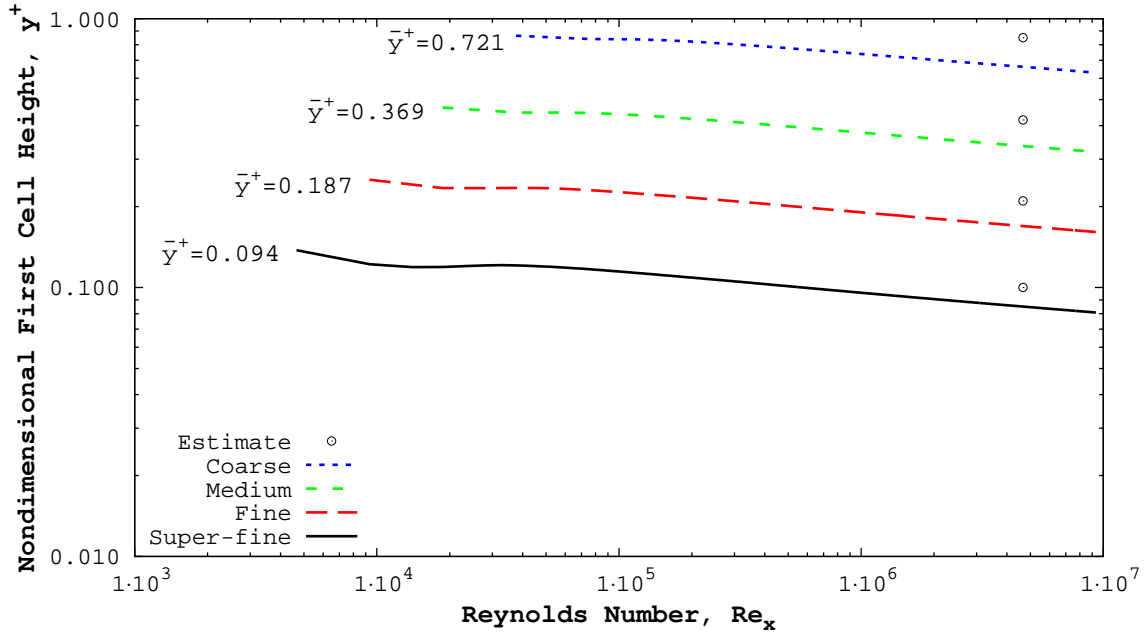


Figure IV.24: Computed surface distributions of y^+ for each flat plate grid. The arithmetic mean of a given data set is given to the left of the curve. Values for the estimates are given in Table IV.7.

IV.3.5. Turbulent Velocity Profiles

The fine grid solution, as opposed to the super-fine grid solution, was used to compare the computed velocity profiles against experimental data and empirical relations. Two factors motivated the selection of the fine grid solution. The first was the similarity of the two solutions, as illustrated by Fig. IV.25. The second was wall

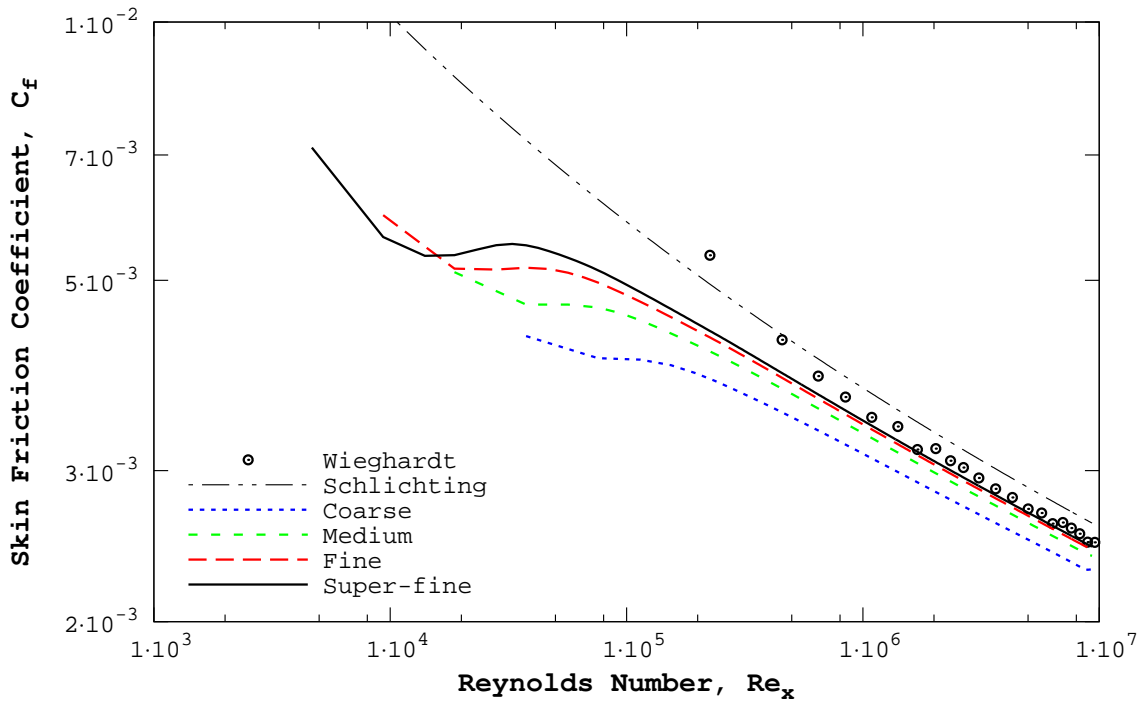


Figure IV.25: Computed skin friction coefficient compared against experiment [Wieghardt & Tillmann, 1951] and theory [Schlichting, 1968, p. 598]. Plotted on a log-log scale.

clock time. The super-fine case required approximately 1.5 times more iterations to achieve the same level of convergence as the fine grid case.

Velocity profiles were extracted at four streamwise locations. The velocity was extracted directly from the grid points to avoid interpolation. As a result, the Reynolds numbers of the extracted data points do not exactly match the experimental Reynolds numbers. The largest difference in Reynolds numbers occurs at the lowest value at 1.9% difference.

Figure IV.26 shows the extracted velocity profiles compared against experiment

and empirical relations. Overall, it was found that UNS3D matched the experimental data closely. The largest difference between the numerical and experimental results were observed at the lowest Reynolds number examined. Here, the computed velocities in the outer region of the profile are roughly 10% higher than what was seen in the experiment. However, Yoder & Georgiadis [1999] saw a similar over-prediction in their predictions at this location as well.

IV.3.6. Effect of Implicit Residual Smoothing

It was stated earlier that implicit residual smoothing (IRS) was not used with the $\kappa - \omega$ equations, even though it was being employed with the Navier–Stokes equations. This was not the case prior to the turbulence model updates that were made in this work. Previously, the same number of smoothing iterations were applied to both the Navier–Stokes and $\kappa - \omega$ equations.

Neither the κ or ω equations showed good convergence trends with the previous version, in which the $\kappa - \omega$ equations were smoothed. This is exemplified by the top two plots in Fig. IV.27. In the plots, the convergence for both turbulence equations is stalled, and worse yet, the sign of the residual for the ω equation remains positive throughout the entirety of the simulation. On the other hand, the figure shows that the turbulence equations residuals dropped several orders of magnitude when IRS is turned off for the $\kappa - \omega$ equations.

The bottom two plots in Fig. IV.27 show that the convergence of the Navier–Stokes equations was hindered by the stalled convergence of the turbulence equations. The residuals for the flow equations were seen to drop roughly 3 orders of magnitude

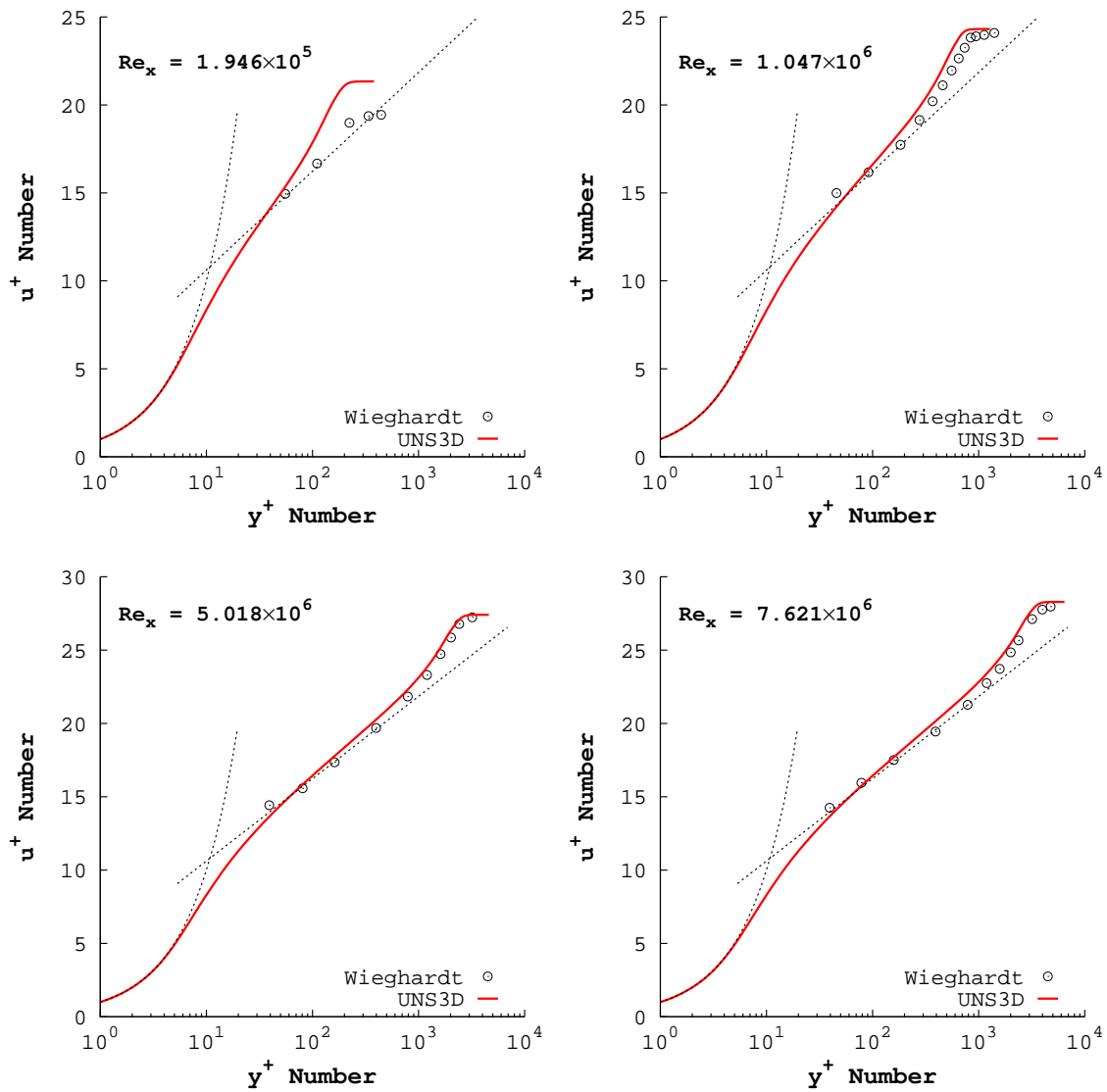


Figure IV.26: Fine flat plate grid extracted velocity profiles compared against experiment [Wiegardt & Tillmann, 1951] and empirical relations [White, 2006, p. 416]. The Reynolds numbers given are from the computation.

before convergence stalled. As with the turbulence residuals, turning off IRS for the turbulence equations improved the convergence of the flow equations. The stalled

convergence previously observed was replaced by convergence of the variables. While residuals for only two variables are shown in the figure, the trends shown are exhibited by the remaining variables.

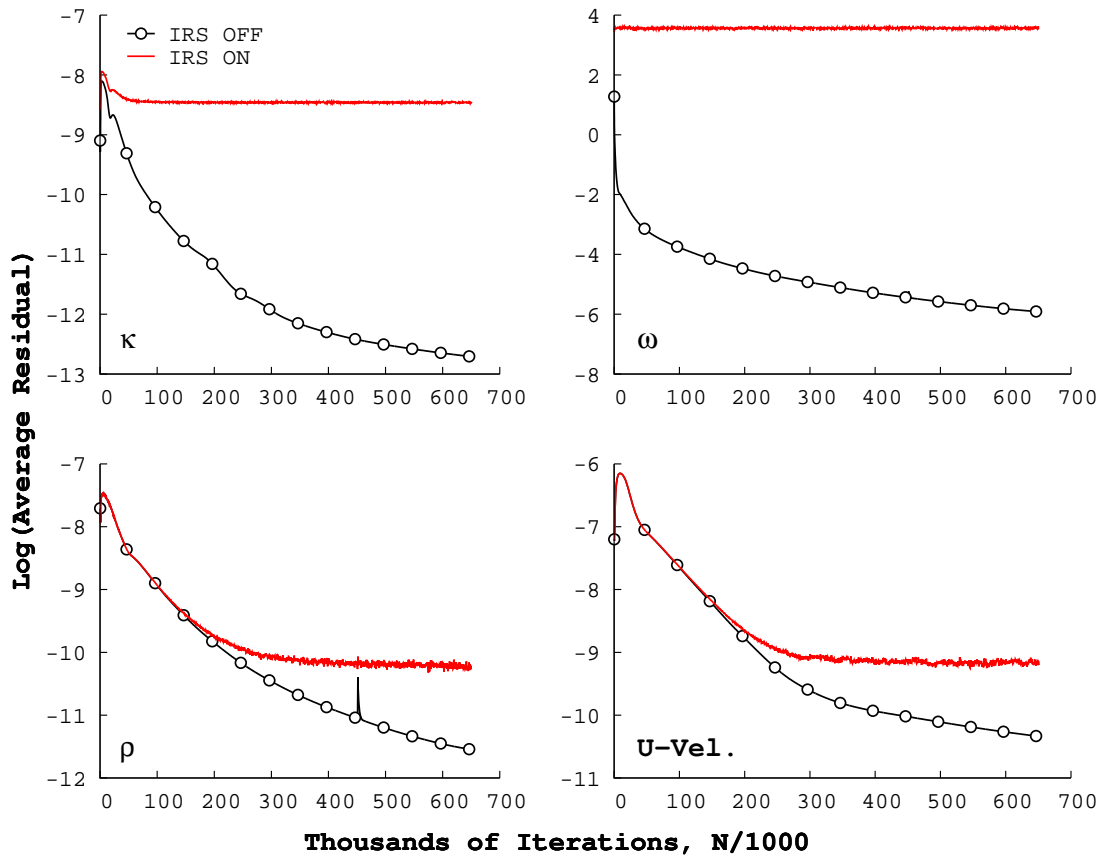


Figure IV.27: Comparing fine grid residual histories for smoothed and not smoothed $\kappa - \omega$ equations.

To get a better understanding of the problem at hand, the turbulence equation variables were extracted from the flow field, such that profiles normal to the wall could be constructed. The variables were extracted at $Re_x = 5.018 \times 10^6$ in the same

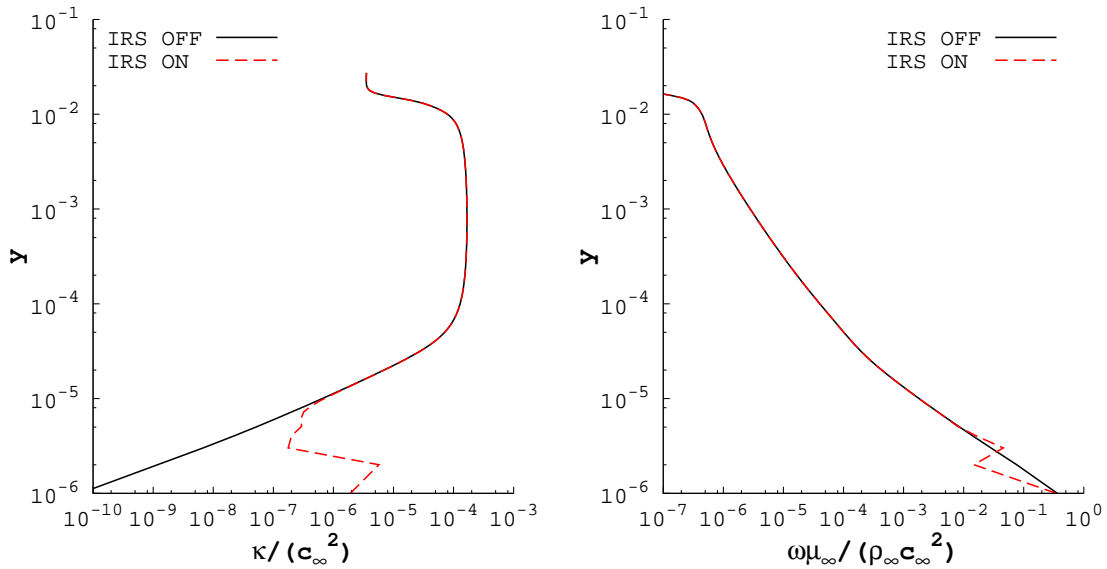
manner as the velocity field was. The profiles of the turbulence variables for both the case of using IRS and without using IRS on the $\kappa - \omega$ equations are shown in Fig. IV.28.

What was found, and what the figure shows, is that the turbulence solution for both variables was distorted in the near wall region when smoothing was used on the turbulence equations. For the fine grid, the near wall region included the first ten grid points above the wall. In contrast, the non-smoothed $\kappa - \omega$ equations produced a result that smoothly approached the value of the boundary condition at the wall. Although the figure only shows this phenomenon at a single point in the flow, examination of the solution contours revealed that this poor solution feature was prevalent throughout the near wall region of the domain, appearing as streaks in the solution contours.

The distortion in the IRS case is most likely being caused by conflict between the turbulence equation boundary conditions and the smoothing. The turbulence boundary conditions are strongly enforced at all boundaries, meaning that the value at the boundary is explicitly defined. To enforce this during the integration step, the values of the residual⁴ at these nodes are set to zero, i.e. $\vec{R}_i = 0$. The implicit residual smoothing algorithm, as implemented in UNS3D, does not differentiate between interior and boundary nodes. As a result, the residual that was once set to zero now becomes non-zero, and will be updated during the integration to the next iteration.

At present, the solution to this problem is to simply avoid implicit residual smoothing for the turbulence equations. However, it is possible that smoothing the

⁴Refer back to p. 50 for the definition of the turbulence equation residual.



(a) Normalized Turbulent Kinetic Energy. (b) Normalized Specific Dissipation Rate.

Figure IV.28: Profiles of κ and ω , extracted from the fine grid, comparing the effect of using implicit residual smoothing (IRS) on the $\kappa - \omega$ equations at $Re_x = 5.018 \times 10^6$.

turbulence residuals could be beneficial in regions where flow separation dominates. A more complete solution to this problem would be implementing selected smoothing, in which boundary nodes where the boundary condition has been strongly applied are skipped in the smoothing algorithm. This would ultimately entail much more coding and thorough testing than time has allowed for, thus it will be left for future work.

CHAPTER V

TURBINE VANE RESULTS

This chapter presents the results for the flow through an annular row of turbine vanes. This case was used to evaluate UNS3D with a simple turbomachinery flow. First, a set of baseline results are presented in the form of a grid convergence study. Then the effect of applying different amounts of under-relaxation to the turbulence equations is explored. Finally, a parametric study of the turbulent inlet boundary conditions are presented.

V.1. Case Definition

The vane geometry is representative of a core turbine vane and is the same geometry used in experiments by Goldman & McLallin [1977] and Goldman & Seasholtz [1982]. The annular cascade has 36 untwisted, constant profile vanes. The vanes are 38.1 mm in height and have an axial chord of 38.23 mm. The diameter at the vane tip was 508 mm, and the hub-to-tip ratio was given by $D_{hub}/D_{tip} = 0.85$. The radii of both the hub and tip endwalls were defined as being constant.

V.2. Grid Generation

The computational grids used in these tests were constructed using the turbomachinery grid generator presented in Chapter III (p. 25). Periodicity was used to simplify the domain from 36 vanes to a single vane. The inlet and outlet planes were located 5.0 and 5.5 axial chords away, respectively, from the vane. The geometrical

coordinates of the vane and the endwalls were taken from the test case material supplied with the NASA turbomachinery flow solver SWIFT [Chima, 2003].

Three grids were constructed for these tests. Viscous wall spacing was evenly applied to the grid adjacent to each wall boundary. Consecutively finer grids were constructed by doubling the number of grid points and halving the grid spacing of the previous grid. The relevant grid sizes for each grid are given in Table V.1. The distances given in the table are given as inner variables, where the wall shear stress has been estimated using the formula by Schlichting [1968, p. 598]. An example of a typical turbine vane grid can be seen in Fig. V.1, and Fig. V.2 shows a close up a single grid layer near the midspan of the vane. The input files used to construct the grids can be found in Appendix E, starting on p. 271.

Table V.1: Turbine vane grid sizes and dimensions.

Grid	I_{MAX}^a	K_{MAX}^b	Total Nodes	y^{+c}	z^{+d}
Coarse	111	50	271,400	1.80	2.30
Medium	221	100	2,134,200	0.90	1.15
Fine	443	200	17,315,600	0.45	0.58

^a Number of points, per layer, defining the vane ^b Number of layers in the spanwise direction ^c Distance to first node off of the vane at the “mid-chord” (estimate) ^d Distance from endwalls to first node at the vane leading edge (estimate)

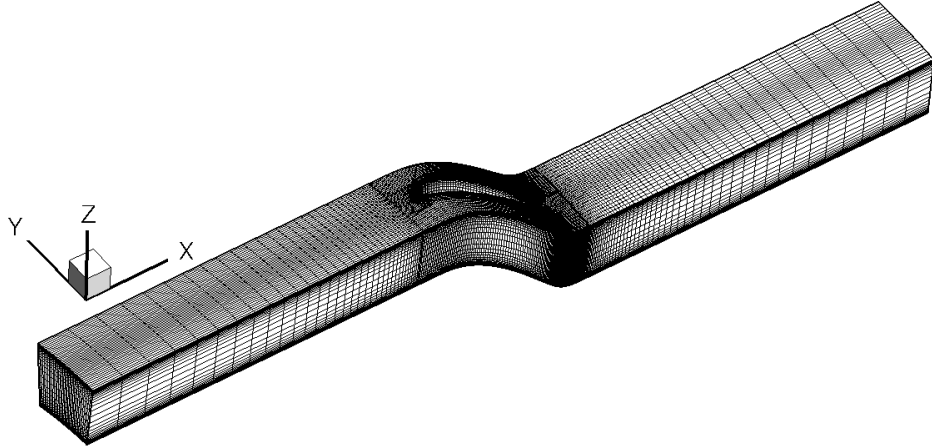


Figure V.1: Typical turbine vane computational domain.

V.3. Flow Conditions

The simulated flow conditions were derived from the conditions observed in the experiments by Goldman & Seasholtz [1982]. The freestream Mach number was used to set the total conditions at the domain inlet. For the desired freestream Mach number, $M_\infty = 0.212$, the inlet total pressure and temperature were determined to be $p_0 = 104548.744$ Pa and $T_0 = 290.891$ K. The static conditions at the inlet were assumed to be the standard values at sea level. The inlet turbulent length scale was defined as 1/1000 of the midspan pitch, or $\ell_T|_\infty = 3.8 \times 10^{-5}$ m. The inlet turbulence intensity was set as $I_T|_\infty = 5\%$.

In the experiment, the ratio of the outlet static pressure at the hub to the inlet total pressure was held fixed at $p_{e,hub}/p_{0,in} = 0.65$ [Goldman & Seasholtz, 1982]. However, for the simulations, the outlet pressure was set using the experimentally determined value for the mass flow rate ($\dot{m}_{exp} = 4.828$ kg/s). Starting with the

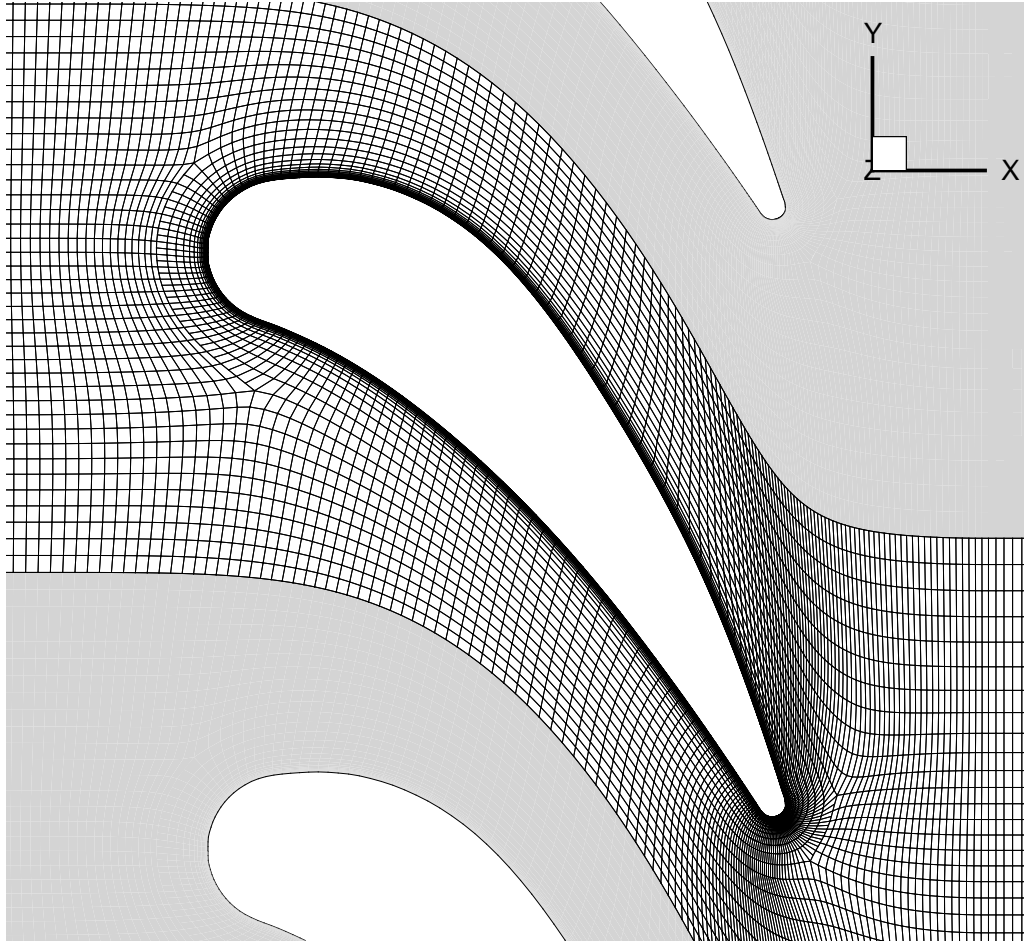


Figure V.2: Turbine vane midspan blade-to-blade grid detail (medium grid). Every other point in the j -direction has been skipped for better figure clarity.

experimentally determined value, the outlet hub static pressure was adjusted until the computed mass flow rate matched experiment. The coarse grid was used to determine this adjusted back pressure due to its relatively small size. The outlet hub static pressure determined on the coarse grid was then used as the boundary condition for both the medium and fine grids. The resultant hub static pressure at the outlet was found to be $p_{e,hub} = 65583.427$ Pa. The resultant ratio of pressures

was then $p_{e,hub}/p_{0,in} = 0.627$.

V.4. Solution Strategy

The solutions on the three grids were computed steady and second-order accurate using the Dervieux limiter. For these cases, the RoeM flux scheme was used as opposed to the basic Roe scheme. The residuals for the $\kappa - \omega$ equations were under-relaxed on all three grids. An under-relaxation value of 1/2 was applied equally to both equations on all three grids. For a detailed look at the solver options utilized in the case, the reader should consult the input file found in Appendix D (p. 256).

The solution was started from constant freestream conditions on both the coarse and medium grids. The fields were initialized with constant pressure, density, and velocity magnitude. The initial direction of the velocity was defined by the chord line of the vane. Starting with this initial field, the solver simulated first-order accurate inviscid for 500 iterations, followed by another 500 first-order laminar iterations. Finally, the turbulence model was activated, with a uniform initial turbulence field defined by the inlet conditions. Four thousand first-order turbulent iterations were computed before switching to second-order accurate. The solver was run until the reduction in the Navier–Stokes equation residuals approached or reached five orders of magnitude.

The solution on the fine grid was started using the solution from a coarser grid due to the large number of nodes contained in the fine grid. The medium grid solution was interpolated onto the fine grid using the inverse distance weighting method. Two thousand first-order turbulent iterations were run before switching

the solver to second-order accurate. The solver was run until the residuals began to level-off, which occurred after roughly a 4.5 order of magnitude decrease in the residuals.

V.5. Convergence History

The residual histories for both the Navier–Stokes equations and the $\kappa - \omega$ equations, for each grid, can be seen in Fig. V.3. The Navier–Stokes equations were seen to converge well on all three grids. On the other hand, the turbulence equations did not exhibit as smooth of a convergent trend as the flow equations. As can be seen in the Fig. V.3, the ω residuals exhibited quite a bit of “static” compared to all of the other residuals and even experienced stalled convergence on the fine grid.

The turbulence model itself is most likely the primary culprit behind the poor convergence of the $\kappa - \omega$ equations. Turbulence models based on the Boussinesq hypothesis suffer when there is a large amount of streamline curvature, boundary layer separation and attachment, and in regions where secondary flows exist [Blazek, 2005, p. 237]. Upon further examination of the geometry used for the current case (see Fig. V.2, p. 110) one could expect one or more of these flow features to be present. At a minimum, large streamline curvature and secondary flows should be expected.

V.6. Grid Convergence Study

Figure V.4 shows the variation in mass flow rate with grid size. With the computed mass flow rate on the smallest mesh matching the experimental value, it

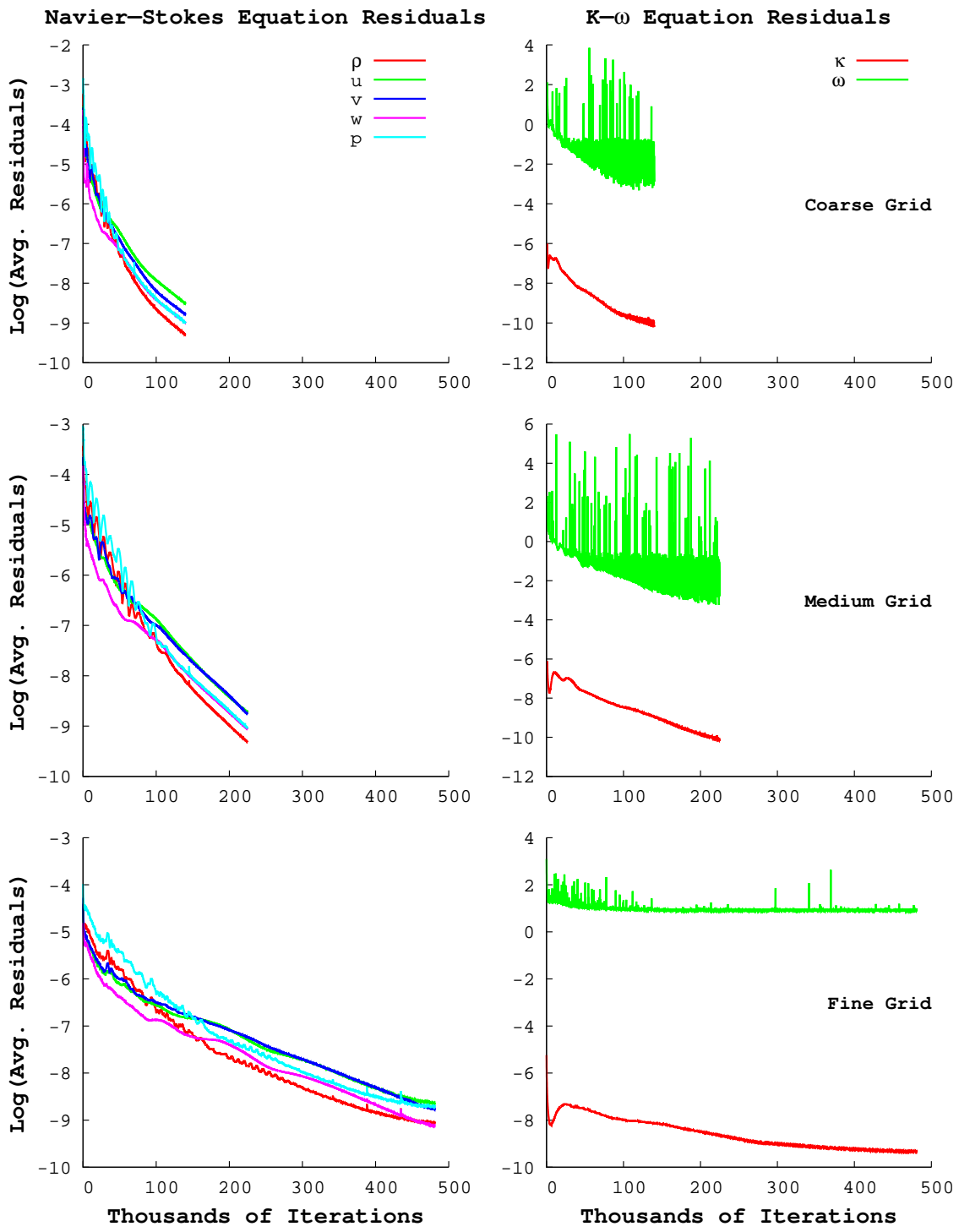


Figure V.3: Turbine vane residual history plots for all three grids.

was found that the computed mass flow rates were approximately 0.4% and 0.6% higher than the experimental value on the medium and fine grids, respectively. It can be seen in the figure that as the grid size increases, the change in computed mass flow rate decreases. As a result, a grid independent solution should be found on a grid with higher resolution than the finest used in this study.

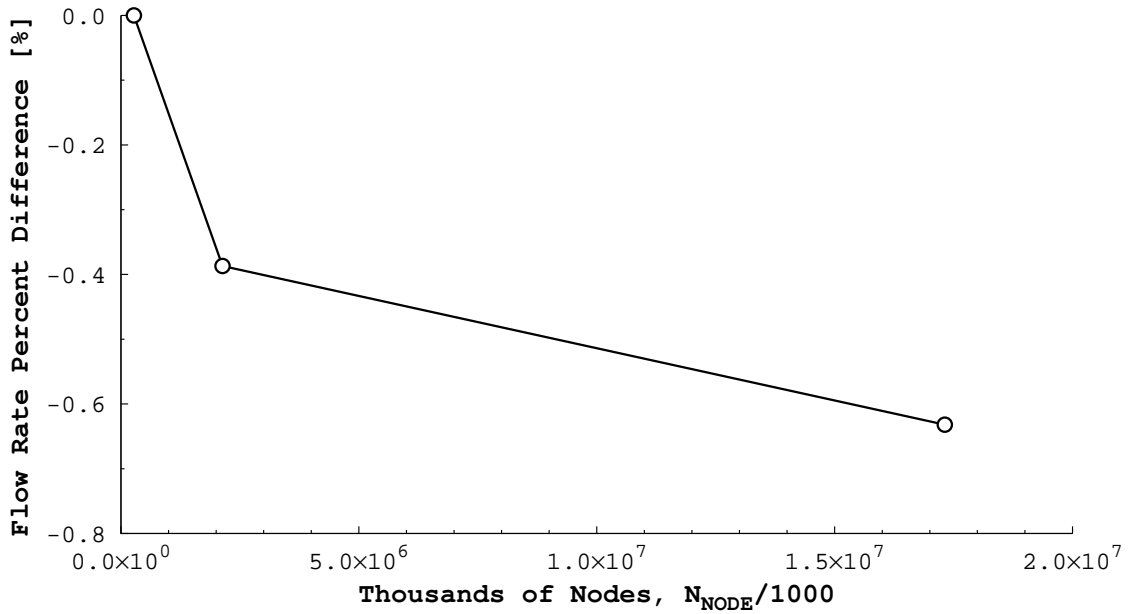


Figure V.4: Computed mass flow rate variation with grid size. Percent difference given by $\% \text{Diff} = 1 - \dot{m}/\dot{m}_{exp}$, where $\dot{m}_{exp} = 4.828 \text{ kg/s}$ is the experimental value.

V.7. Comparison to Experiment

Isentropic Mach number distributions were computed at three spanwise stations. Their locations corresponded to measurement locations found in the experiment such

that a direct comparison with the experimental data could be made. The isentropic Mach number was computed by solving

$$\frac{p_{0_1}}{p} = \left(1 + \frac{\gamma - 1}{2} M_{is}^2\right)^{\gamma/(\gamma-1)}$$

for M_{is} . Here, p_{0_1} is the average inlet total pressure, and p is the static pressure on the surface of the vane.

The computed isentropic Mach number distributions are compared against the experiment in Fig. V.5. The computed profiles for all three grids show good qualitative agreement with the experimental data. Very little difference was seen between the three solutions along the suction side of the vane at any spanwise location. The largest variations in the computed solution occurred on the pressure side of the vane for $x/c < 0.6$. In this region, the medium and fine solutions were similar, and both had slightly larger values of M_{is} than the coarse solution. Beyond $x/c \approx 0.6$, the curves for the three solutions collapsed onto one another.

Further comparison against the experiment was made by examining the flow downstream of the vane. The computed solutions were extracted at a plane of constant x and circumferentially averaged along each radial layer. The location of the sampling corresponded to the measurement plane from the experiments and was 1/3 axial chords downstream of the vane. The circumferentially averaged solution was used to generate spanwise profiles of flow angle, normalized static pressure, and total pressure loss. The flow angle was defined as the angle between the axial and tangential velocities, and the static pressure was normalized by the specified inlet total pressure. The comparison between numerical and experimental results can be seen in Fig. V.6.

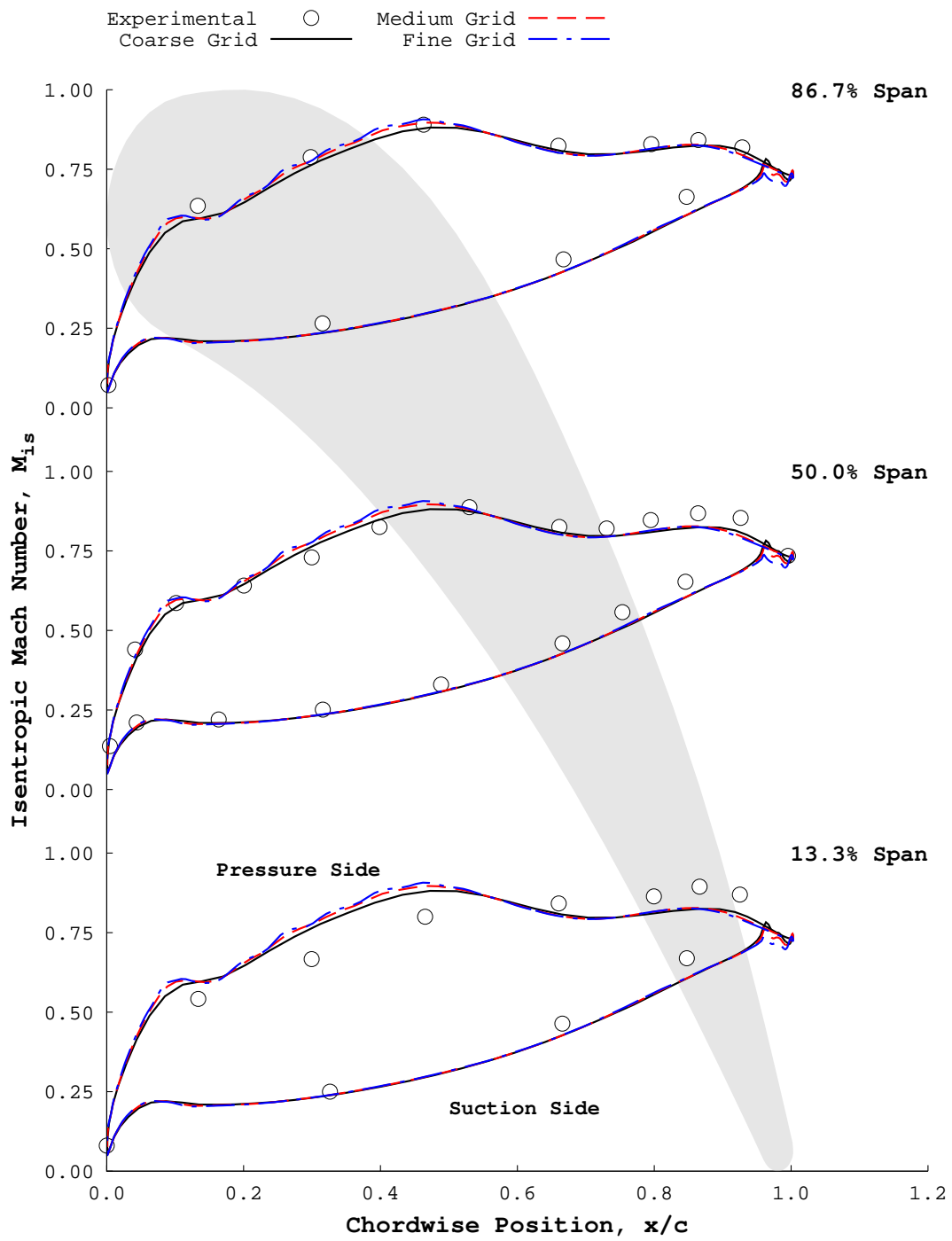


Figure V.5: Computed surface isentropic Mach numbers versus experiment [Goldman & Seasholtz, 1982]. Spanwise locations are measured from the hub.

There was very little qualitative difference between the spanwise profiles constructed from each grid, as shown in Fig. V.6. The computed flow angles were found to be in good agreement with the experimental data. The predicted total pressure losses were found to show good qualitative agreement with the experimental data. The total pressure loss profile generated on the coarse mesh showed the largest losses. Predicted losses tended towards the experimental data as the grid size was increased.

Qualitatively, the predicted profiles of static pressure are in agreement with the experimental data. The discrepancy in the magnitude of the static pressure is the result of matching the experimental mass flow through the throttling of the outlet static pressure. A lower outlet static pressure was required to match the experimental mass flow rate, resulting in the lower values of p/p_{0_1} seen in Fig. V.6. However, the predicted shape of the static pressure profile lends credence to the proper implementation of the new outlet radial variation boundary condition.

One final comparison with experiment was done by comparing the total pressure in the wake of the vane. Following Goldman & McLallin [1977], normalized total pressure contours were extracted on a planar slice 1/3 axial chords downstream of the vane's trailing edge. The specified inlet total pressure was used to normalize the predicted total pressure. The medium grid was selected for this comparison due to its relatively small size, with respect to the fine mesh, and the close similarity between the medium and fine results.

Figure V.7 shows the comparison between the predicted total pressure and the measured total pressure in the wake of the vane. The predicted wake profile showed excellent qualitative agreement with the experimental data. In comparing the con-

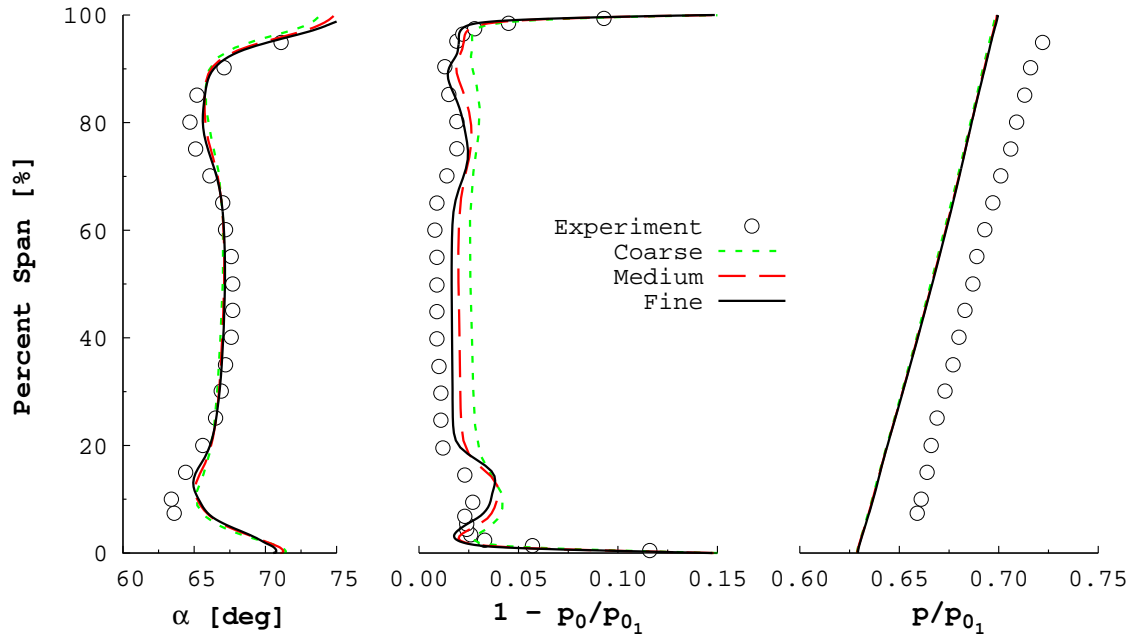
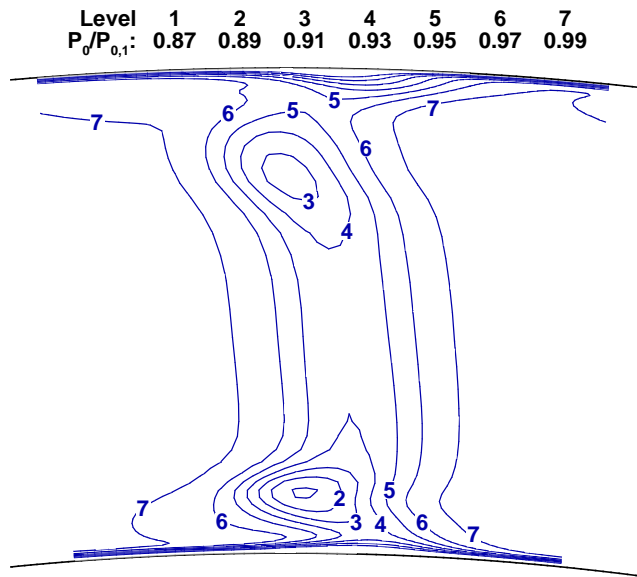
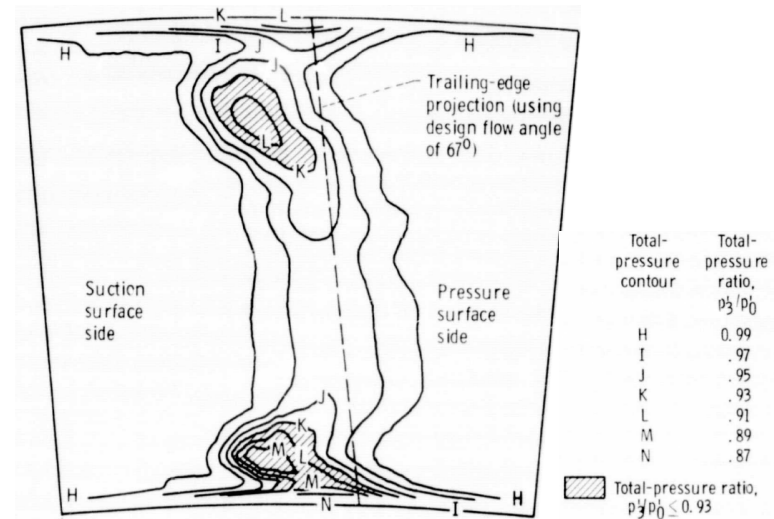


Figure V.6: Predicted and experimental [Goldman & Seasholtz, 1982] spanwise profiles of flow angle (α), total pressure loss ($1 - p_0/p_{01}$), and normalized static pressure (p/p_{01}).

four plots, it was seen that UNS3D under-predicts the total pressure in the bottom 10% of the span. The low total pressure region ($p_0/p_{01} \leq 0.93$) seen in the experiment (shaded region of Fig. V.7b) was more compact, whereas the predicted low pressure region is more circumferentially elongated. This results in the “bubble” of over-predicted total pressure loss seen previously in Fig. V.6.



(a) UN3D



(b) Goldman & McLallin [1977]

Figure V.7: Comparing predicted (left) and measured [Goldman & McLallin, 1977] (right) normalized total pressure contours at $1/3$ axial chords downstream of the vane trailing edge. The same contour levels are used by both plots.

V.8. Medium Grid Solution

Contours of the predicted Mach number at the midspan are shown in Fig. V.8. The contours in the figure were taken from the medium grid. The predicted Mach contours were found to be in excellent agreement with the flow predicted by Chima & Yokota [1990] and by Chima [2003].

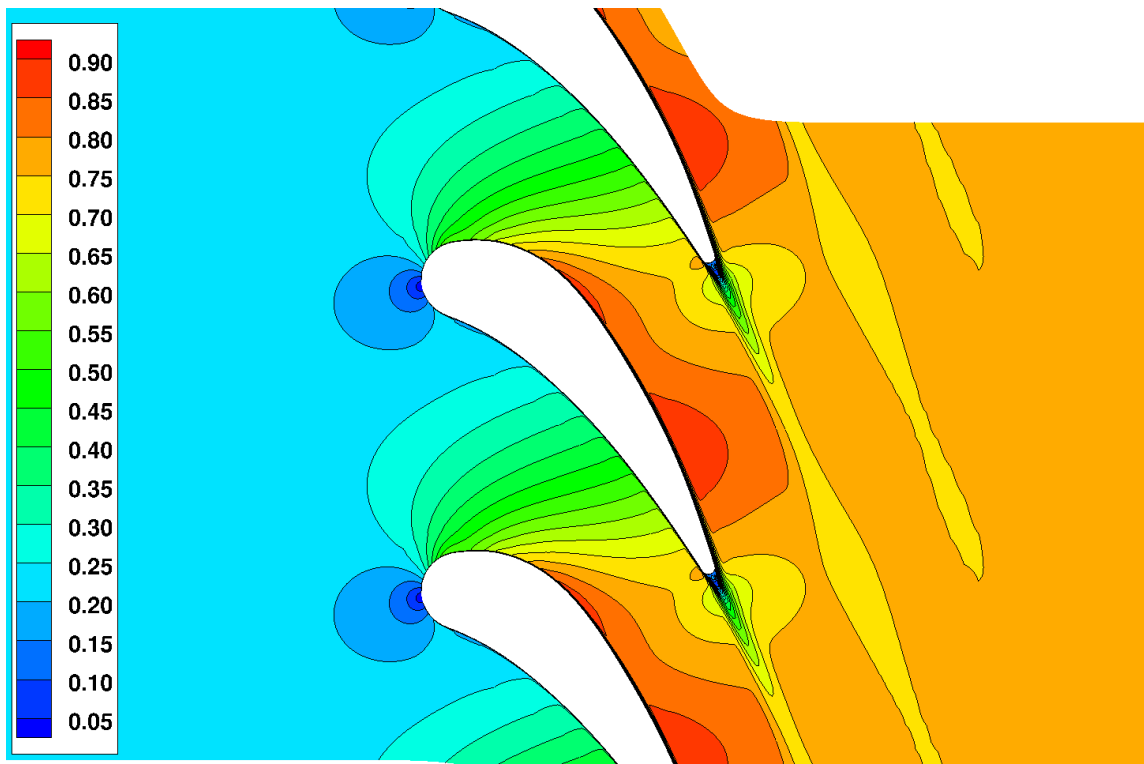


Figure V.8: Predicted turbine vane Mach contours at midspan (medium grid).

The estimated values of y^+ found in Table V.1 were compared against the values predicted by UNS3D for the medium grid. Contours of the predicted y^+ number along both sides of the vane and the hub casing are shown in Fig. V.9. It was found

that the estimated values of y^+ were greatly under-estimated.

The average y^+ numbers predicted by UNS3D on the hub and tip casings, at the leading edge, were 1.73 and 1.88, respectively. Whereas the estimated value was 1.15, upwards of a 60% difference between estimation and prediction. The story was the same at the midspan of the blade. Here the average predicted value, $\bar{y}^+ = 1.7383$, was nearly 100% larger than the estimation.

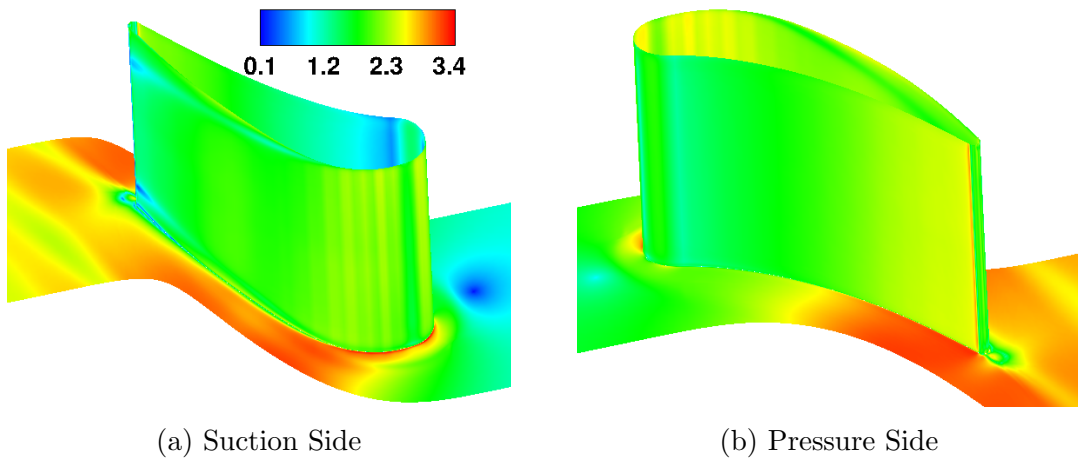


Figure V.9: Predicted y^+ numbers along the turbine vane and hub surfaces.

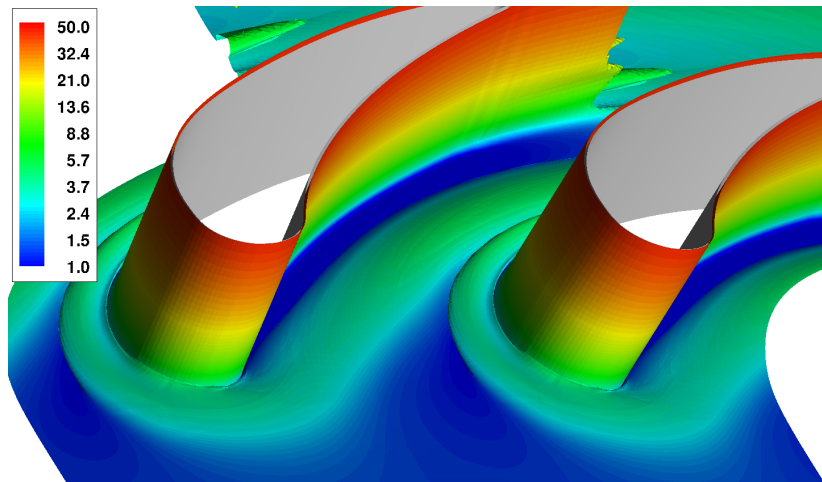
The cause for this under-estimation of y^+ is undoubtedly due to the fact that the flow is being accelerated by the turbine vane. As a result of the flow acceleration, the edge velocity, in this case the inlet freestream, used to originally estimate the y^+ number was too low. A more appropriate edge velocity to use for the estimation would be the average outlet velocity.

An iso-surface of normalized vorticity magnitude was used to identify the location of the vortices near the end walls. Figure V.10 shows the vortex locations using

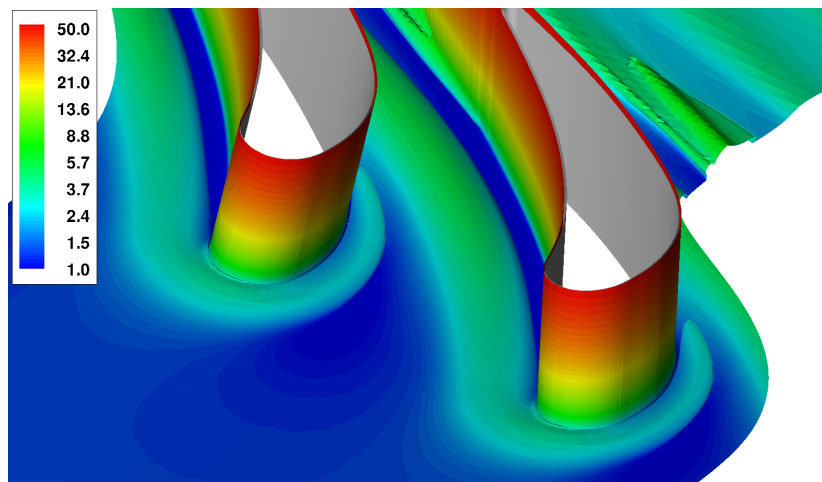
a normalized vorticity magnitude of $1.5 \cdot 10^{-2}$, and has been colored by percent span to show depth. The figure shows the presence of a horseshoe vortex that formed near the leading edge of the vane with both legs of the vortex being drawn into the passage. The pressure side leg of the horseshoe vortex ultimately becomes the passage vortex [Langston, 2001]. The figure shows that the vortices extended between 5-10% of the vane's span away from the end walls into the freestream.

The flow was extracted on a grid plane of constant i at the leading edge of the vane to further examine the vortical structures near both the end walls and the leading edge. The velocity vectors tangent to the grid plane near the hub and the tip end walls are shown in Fig. V.11. The figure shows a region of re-circulation upstream of the vane near both end walls, indicating the presence of the horseshoe vortices. Also shown in the figure details are the counter-rotating corner vortices that form at the junctions between the vane's leading edge and the end walls. The predicted velocity field was found to be qualitatively similar to the one computed by Arnone et al. [1993].

The liftoff of the passage vortex was visualized by examining the streamlines immediately adjacent to the suction side of the vane. The streamlines were computed as surface streamlines with the surface being defined as the $j = 2$ grid surface to mimic an oil flow visualization technique. The resultant streamline are shown in Fig. V.12a, and are compared against streamlines numerically predicted by Arnone et al. [1993] in Fig. V.12b. The figures show that the result by UNS3D are similar to those found in the literature. The distance along the span that the streamlines were displaced near the tip was larger for UNS3D by approximately 10% of the span



(a) Hub End Wall



(b) Tip End Wall

Figure V.10: End wall vortex locations given by a single iso-surface of normalized vorticity magnitude. Contours are percent span from the nearest end wall.

in comparison to Arnone et al.. The reason for this difference is likely the size of the inlet boundary layer. Arnone et al. [1993] allowed for the presence of a boundary layer at the domain inlet, whereas UNS3D assumed uniform flow across the entirety of the inlet.

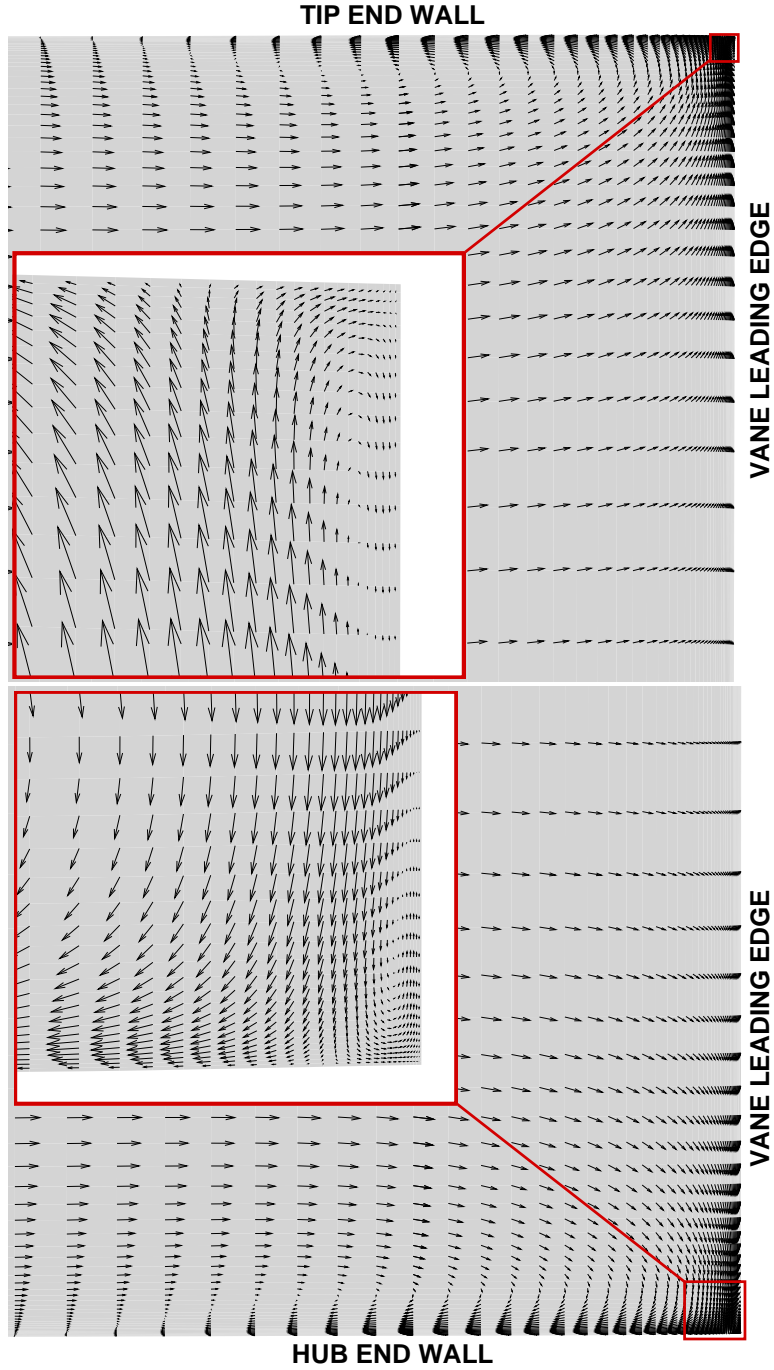


Figure V.11: Velocity vectors tangential to the leading edge grid plane showing the end wall horseshoe vortices and the counter-rotating corner vortices.

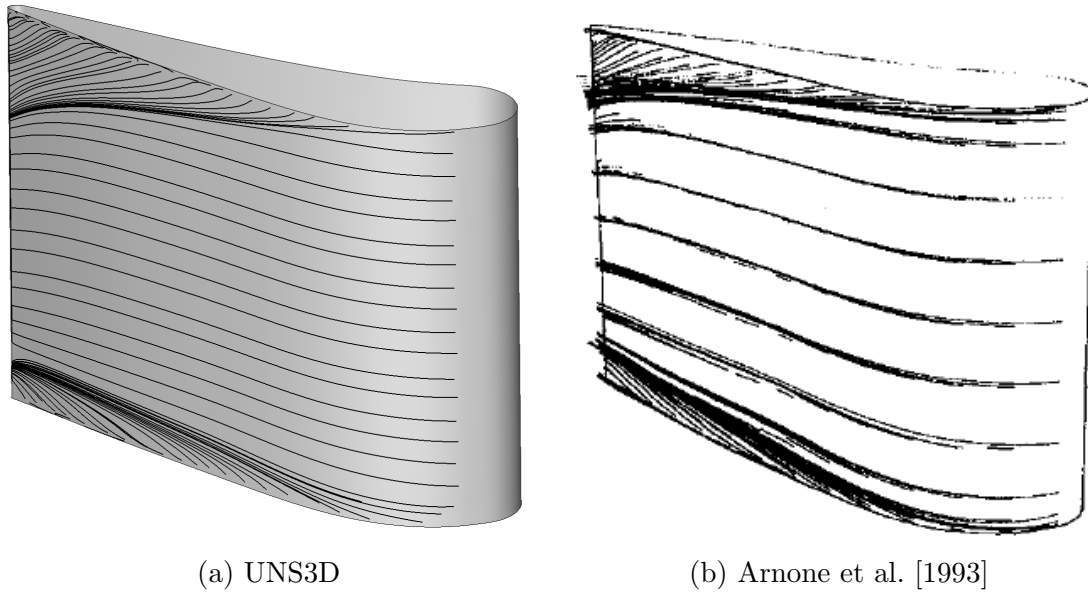


Figure V.12: Streamlines near the suction side surface showing liftoff of the passage vortex for: (a) current prediction and (b) prediction by Arnone et al. [1993].

V.9. Impact of Under-relaxing the $\kappa - \omega$ Equations

The purpose of this section is to examine what effect under-relaxing the turbulence equations has on solution convergence, both for the $\kappa - \omega$ and Navier–Stokes equations and on the solution itself. An under-relaxation factor of 1/2 was applied to each turbulence equation in the results previously presented in this chapter. Results for three additional under-relaxation factors are presented herein.

For completeness, the relaxation parameter is a modifier that acts upon the time step. It can be used to either over-relax (increase Δt) or under-relax (decrease Δt) a given equation. The relaxation parameter, φ_R , can be applied individually to each component of the discrete equation. The relaxed discrete form of the flow

and turbulence equations, written for cell i at time step n , is written as $\Delta\vec{q} = [\varphi_{R_1} R_1, \dots, \varphi_{R_7} R_7]^T \frac{\Delta t}{\Omega}$. The sixth and seventh components of the preceding relaxed discrete equation correspond, respectively, to the κ and ω equations.

Solutions were computed using the medium grid in the same manner as the previously presented results. The only difference being the choice of relaxation parameter for the turbulence equations. The new under-relaxation parameters tested were 1/100, 1/10, and 1. Each value was applied equally to both the κ and ω equations. Because under-relaxation was applied equally and only to the turbulence equations, the subscripts “6” and “7” are dropped herein in favor of φ_R .

It was found that using a greater amount of under-relaxation ($\varphi_R \rightarrow 0$) resulted in poorer convergence trends for the $\kappa-\omega$ equations. To illustrate this, the turbulence equation residual histories for each value of φ_R have been plotted in Fig. V.13. The κ equation residuals stalled for $\varphi_R \leq 1/10$, and showed good convergent trends for $\varphi_R = 1/2$ and $\varphi_R = 1$. Only the greatest amount of under-relaxation ($\varphi_R = 1/100$) was able to suppress the oscillatory nature of the ω residuals. However, the ω residual remained positive, and this under-relaxation parameter resulted in the worst convergence for the κ equation.

It was also shown that the convergence of the flow equations was hindered by the poor $\kappa-\omega$ convergence. This is best illustrated by Fig. V.14, where the residuals for the flow equations are plotted for each value of φ_R tested. The residuals for both the $\varphi_R = 1/2$ and $\varphi_R = 1$ cases roughly averaged a 5.5 order of magnitude drop. Neither case showed signs of the residual leveling off, and the $\varphi_R = 1/2$ case exhibited the best overall convergence. The residuals for the $\varphi_R = 1/10$ case appeared to level off

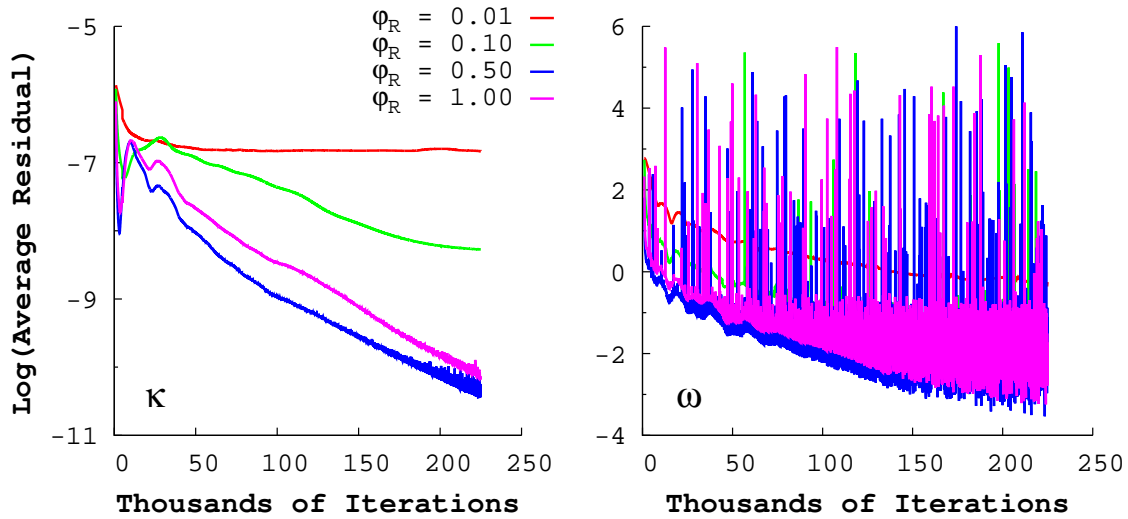


Figure V.13: Turbine vane $\kappa - \omega$ residual history comparison for different under-relaxation factors.

after dropping approximately 4.2 orders of magnitude, on average.

The largest amount of under-relaxation caused the flow equation residuals to converge in a “bumpy” fashion, as shown in Fig. V.14. These flow residuals indicate that the solver is having difficulty arriving at a steady solution, likely caused by additional separation in the flow not found in the other cases. Even so, the residuals for this case dropped by roughly 3.5 orders of magnitude, on average.

As before, spanwise profiles of flow angle and total pressure loss were extracted 1/3 axial chords downstream of the vane. The computed profiles are shown in Fig. V.15, where the solid black lines correspond to the previously presented results.

It was found that the spanwise profiles for both the $\varphi_R = 1/2$ and $\varphi_R = 1$ cases were nearly indistinguishable. The spanwise profiles for $\varphi_R = 1/10$ were very similar overall to the lower under-relaxation cases. The primary difference was found in the

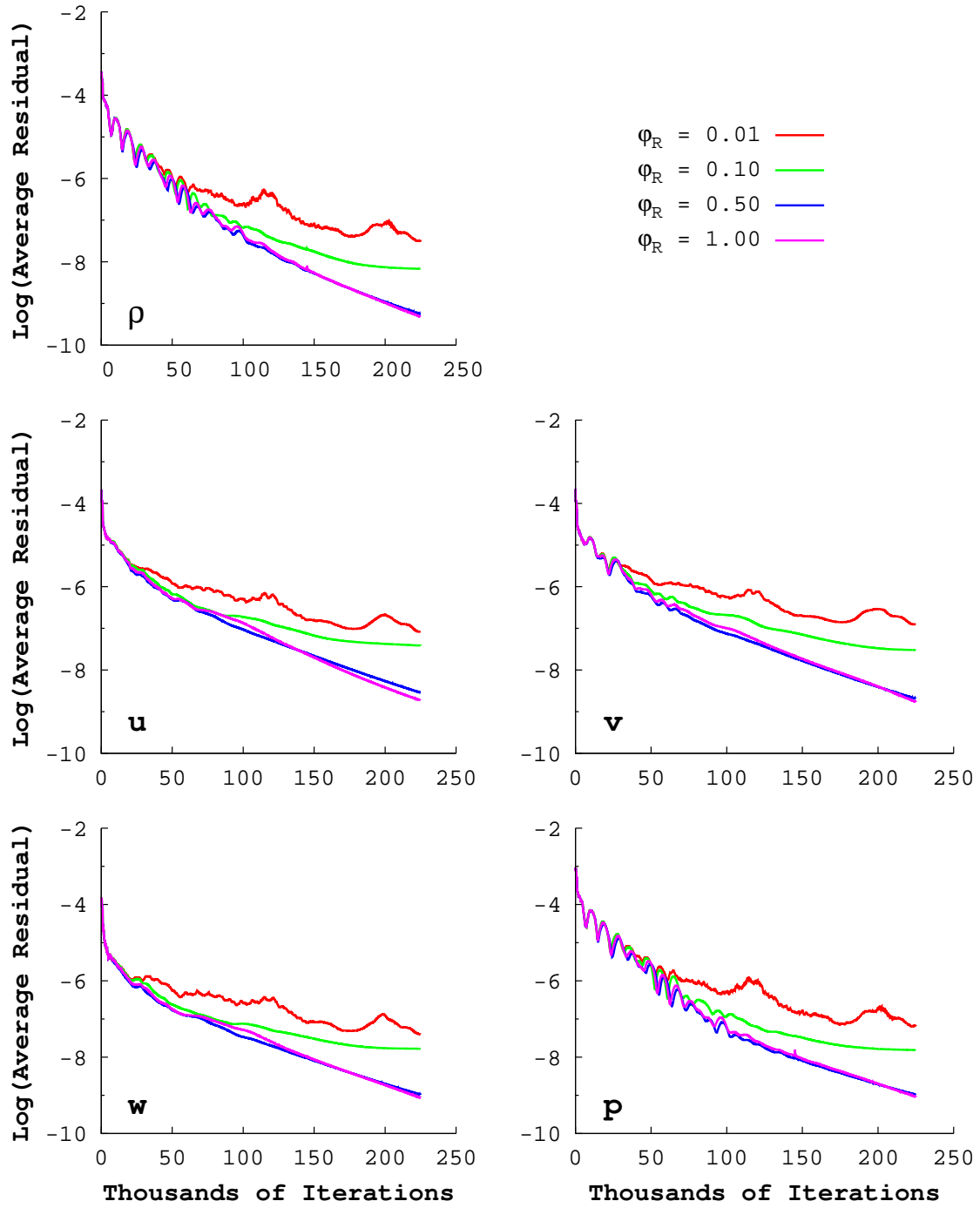


Figure V.14: Turbine vane flow residual history comparison for different under-relaxation factors.

lower 20%, where the location of the highest total pressure loss was shifted slightly upwards.

In the case of $\varphi_R = 1/100$, large differences were seen in the profiles for both the flow angle and the total pressure loss in comparison to the other cases. The differences were the greatest in the regions near the hub and tip walls. However, the predicted flow characteristics towards the middle of the vane were quantitatively similar to the other under-relaxation parameter results.

The most intriguing part of the spanwise profiles is in the first 20% of the span for the $\varphi_R = 1/100$ case. Here it was seen that both flow parameters, most notably for total pressure loss, had non-smooth spanwise profiles. This is the result of not enough dissipation being added into the flow by the turbulence model, resulting from a reduced time step size from the under-relaxation procedure. The small effective time step becomes prohibitive to the $\kappa - \omega$ equations, preventing them from reaching a useful or “converged” solution.

As with any turbomachinery case, secondary flows and separation are always present, making these flows inherently unsteady. Using too much under-relaxation removed the turbulence model’s ability to dampen out some of the unsteadiness so that an average, steady solution could be found. The “bumpy” residuals in Fig. V.14 and the poor prediction in the near-endwall regions of the wake are evidence of this.

V.10. Inlet Turbulence Boundary Conditions

The purpose of this section is to show what effect the turbulence inlet boundary conditions have on the solution. Both the turbulence intensity and length scale were

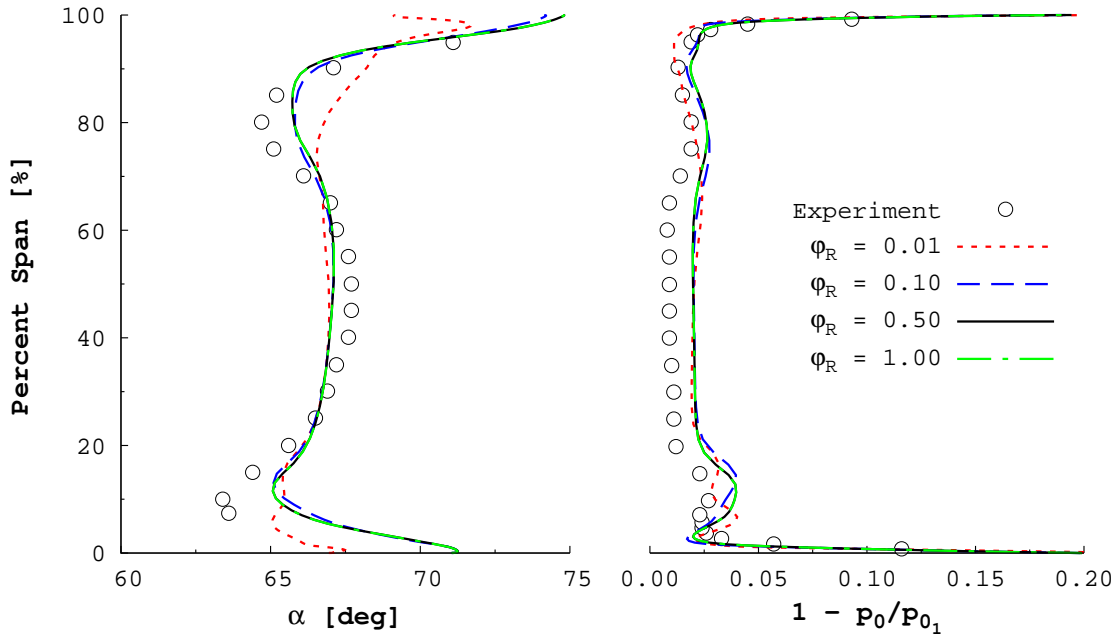


Figure V.15: Downstream spanwise profiles of flow angle (α) and total pressure loss ($1 - p_0/p_{01}$) for different under-relaxation factors. Experimental data by Goldman & Seasholtz [1982].

varied in this study. The effect on the solution will be gauged by comparing the residual histories of the equations and by looking at the solution downstream of the vane.

The medium grid was again used for these numerical experiments. The flow conditions were the same as those used in the grid convergence study in the main part of this chapter. The original turbulence equation under-relaxation factor of $1/2$ was used for these tests. The parameters being varied here are the inlet turbulent intensity, $I_T|_{\infty}$, and the inlet turbulence length scale, $\ell_T|_{\infty}$. Table V.2 presents the turbulent inlet condition combinations that were tested and gives the name assigned

to the specific case. These names found in the table will be used for case identification herein.

Table V.2: Turbine vane turbulence inlet boundary condition test matrix.

$I_T _\infty$ \ $\ell_T _\infty$ [m]	1%	5%	10%
3.8×10^{-5}	GT01-a	GT05-a [†]	GT10-a
3.5×10^{-4}	GT01-b	GT05-b	GT10-b

[†] Baseline case previously presented in main part of current chapter.

The residual histories for the κ - ω equations are shown in Fig. V.16. It was found that there was very little difference in the convergence of the turbulence equations for the different inlet conditions tested. The five new cases followed the same convergent trends of the baseline case (Case GT05-a).

The comparison of the flow equation convergence histories yielded similar results to that of the turbulence equation convergence. The different inlet conditions did not negatively impact the convergence of the flow equations, as shown in Fig. V.17. It was found that convergence was slightly improved near the end of the simulation for the cases which had both the largest two intensities and length scales (Cases GT05-b and GT10-b).

Spanwise profiles of flow angle and total pressure loss were generated from a planar cut 1/3 axial chords downstream of the vane for each case given in Table V.2. The predicted spanwise profiles were compared against experimental data [Goldman & Seasholtz, 1982], as shown in Fig. V.18. The predicted profiles were found to be

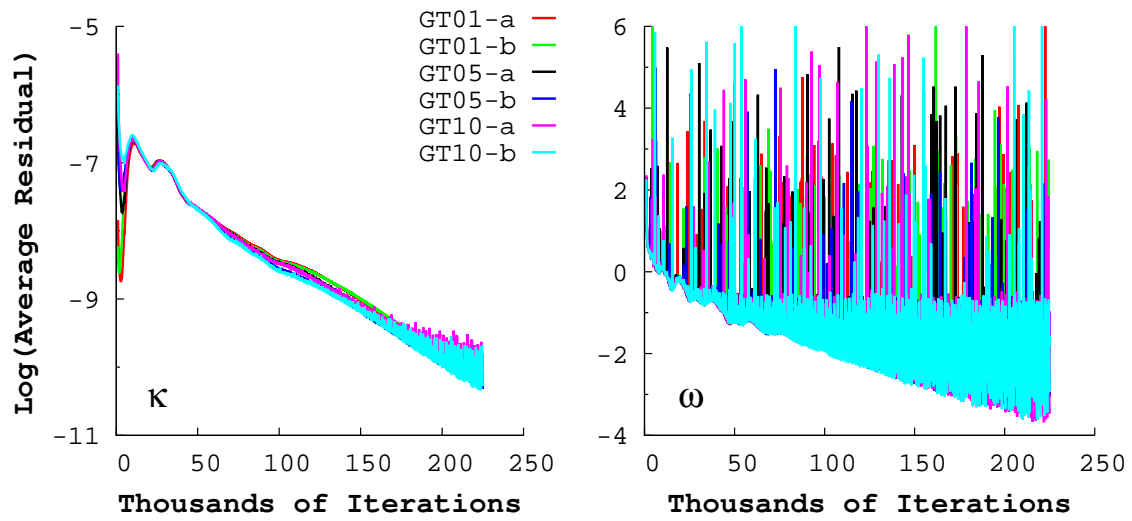


Figure V.16: Turbine vane $\kappa - \omega$ residual histories for the different inlet turbulence conditions.

very similar, both qualitatively and quantitatively, regardless of the turbulence inlet conditions used.

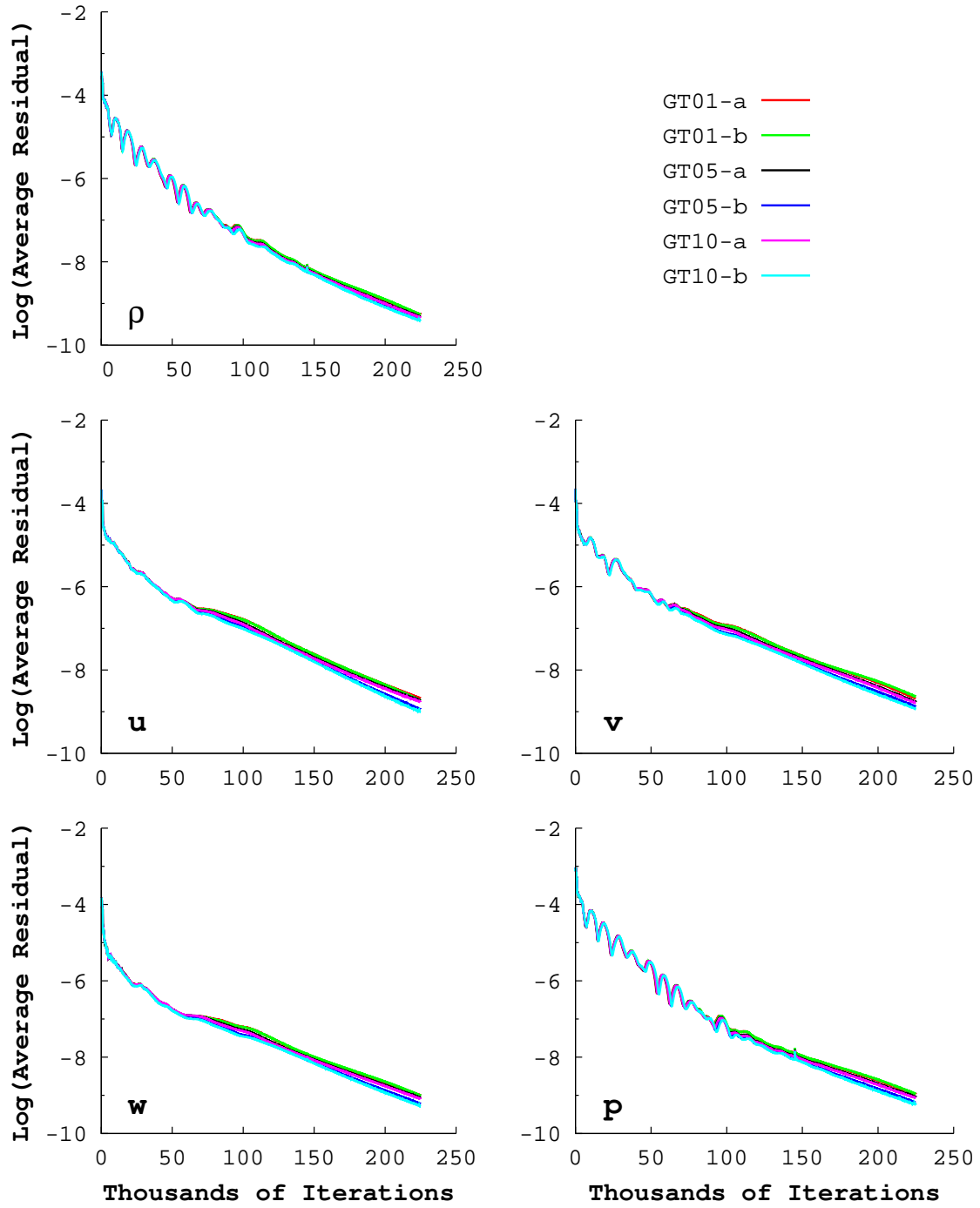


Figure V.17: Turbine vane flow residual history comparison for the different inlet turbulence conditions.

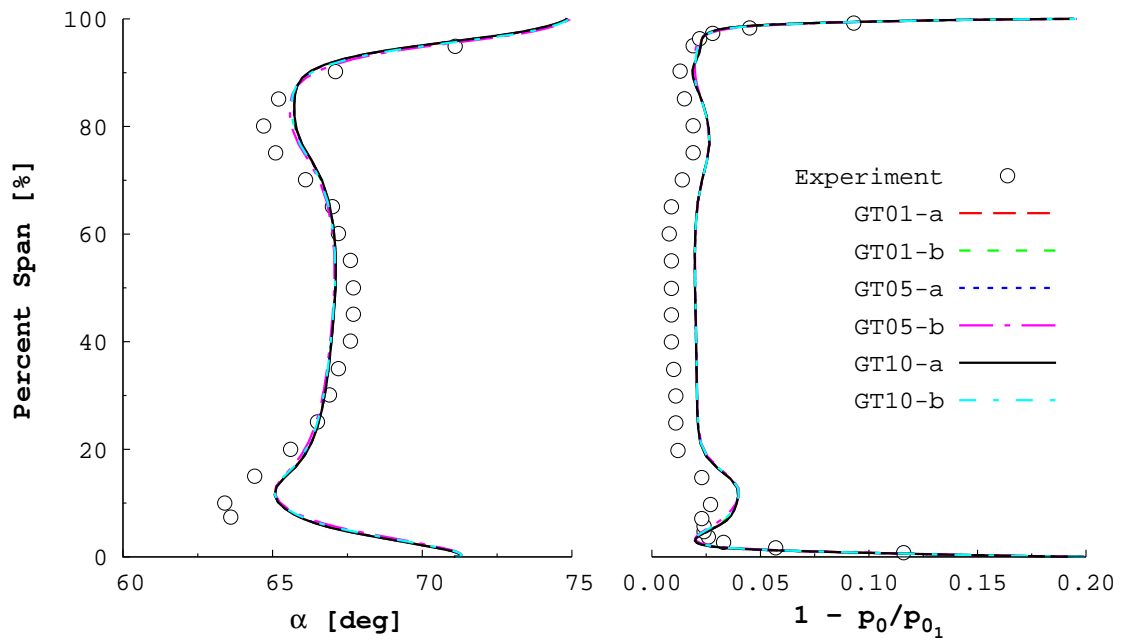


Figure V.18: Downstream spanwise profiles of flow angle (α) and total pressure loss ($1 - p_0/p_{01}$) for different inlet turbulence conditions. Experimental data by Goldman & Seasholtz [1982].

CHAPTER VI

TRANSONIC FAN INVISCID ANALYSIS

This chapter contains results from the inviscid analysis of a transonic fan rotor that was developed and tested at NASA Lewis [Urasek et al., 1979]. A complete description of the rotor is first presented. Following this, the computational grid and the boundary condition definitions are described. The calculation of a compressor map speed line, at design wheel speed, is the presented. Finally, a study on the impact of the choice of higher-order limiter on a single operating point solution is presented.

VI.1. Case Description: NASA Rotor 67

Rotor 67 is the first-stage rotor of a two-stage fan developed at NASA Lewis [Urasek et al., 1979]. The rotor has a low aspect-ratio of 1.53, and is made up of twenty-two multiple-circular-arc blades. The hub and tip profiles of the rotor blade are shown in Fig. VI.1a, and a portion of the full annular blade row can be seen in Fig. VI.1b. The axial chords for the hub and the tip blade profiles are 9.104 cm and 4.029 cm, respectively. The flow paths for the inner and outer walls are shown in Fig. VI.1c.

The NASA Rotor 67 geometry was chosen as the test case due to its wide use in literature and the availability of geometry. The experimental data of Pierzga & Wood [1985] and Wood et al. [1990] was used for comparison herein. The geometrical coordinates for both the rotor and flow path were taken from the SWIFT test case package [Chima, 2003].

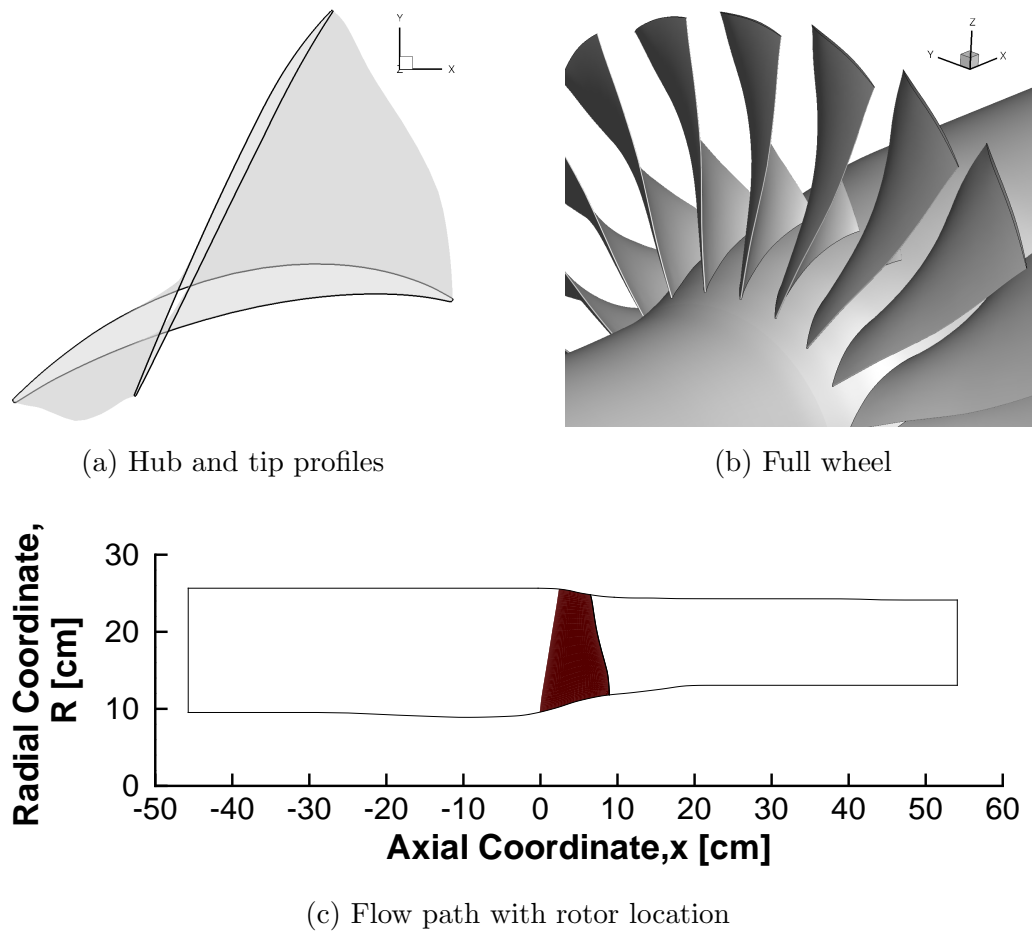


Figure VI.1: Rotor 67 blade and flow path geometry.

The design point for the rotor was a wheel speed of 16,043 RPM, a pressure ratio of 1.63, and a mass flow rate of 33.25 kg/s. For the design tip speed of 428.9 m/s, the relative tip inlet Mach number was 1.38. This relative tip velocity corresponds to a Reynolds number, based on axial chord, of $Re_{c_a} = 1.32 \times 10^6$. The reported maximum mass flow rates, which occur when the flow becomes choked, from the experiments were 34.92 kg/s [Pierzga & Wood, 1985] and 34.96 kg/s [Wood et al., 1990]. The flow rates found in both the experimental and predicted compressor map

plots are normalized by their respective maximum flow rates.

In order to compute the rotor performance, the solution was extracted on planes of constant x just upstream and downstream of the rotor. The location of the upstream and downstream measurement planes were $x_p = -2.473$ cm and $x_p = 11.011$ cm, respectively, and are the same as those used in the experiment by Wood et al. [1990]. Figure VI.2 shows the locations of the measurement planes in relation to the rotor. The same measurement plane locations are used for every case presented herein.

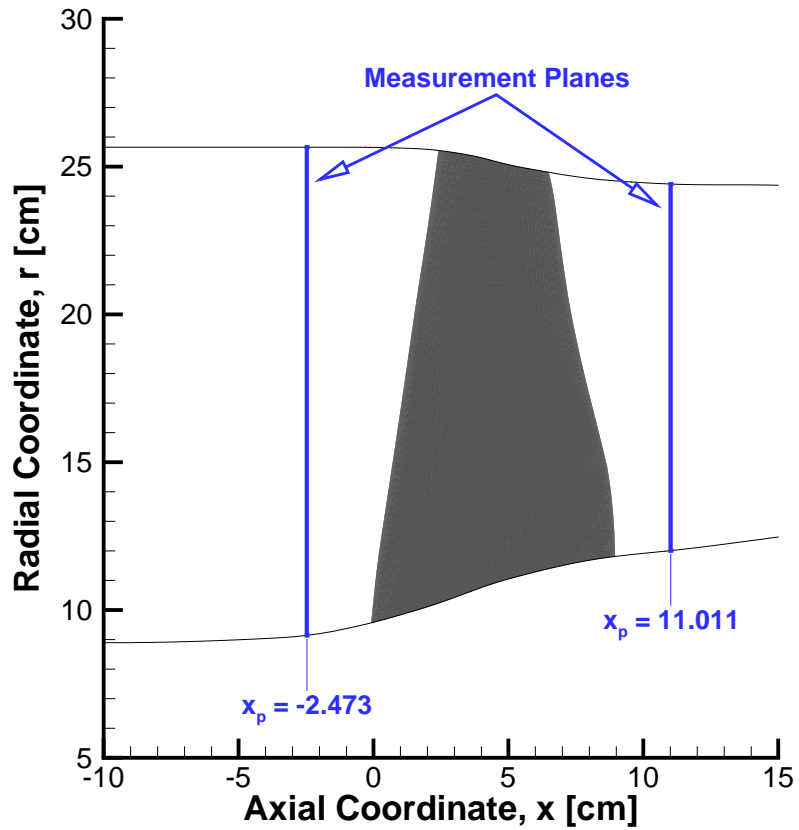


Figure VI.2: Numerical measurement plane locations.

VI.2. Computational Grid Description

The computational grid for the inviscid Rotor 67 case was constructed using the turbomachinery grid generator presented in Section III.1 (p. 25). The flow was assumed to be periodic, allowing for a single blade passage to be modeled. The rotor blade was modeled using 131 points distributed along the surface of the blade in the i -direction on 36 spanwise layers. This resulted in a grid with 299,844 unique nodes.

Uniform spacing was applied at each wall boundary. The grid spacing along the surface of the blade was set such that $y_1/c_a^{(h)} = 1.675 \times 10^{-3}$, and the grid spacing normal to the endwalls was set as $z_1/c_a^{(h)} = 9.378 \times 10^{-3}$. Here, $c_a^{(h)}$ is the hub axial chord. The grid generator input file in Appendix E (p. 278) gives a complete description for all of the grid dimensions used. The complete computational domain can be seen in Fig. VI.3, and Fig. VI.4 shows the details of the tip blade-to-blade grid layer.

VI.3. Boundary Conditions

Three boundary conditions types, excluding periodicity, were applied to grid: wall, inlet, and outlet. All of the walls were defined as rotating walls. The inlet boundary was set five axial hub chords upstream of the hub leading edge. The outlet was placed five axial hub chords downstream of the hub trailing edge. The large distances to the inlet and outlet boundaries were chosen to minimize their effect on the near-rotor solution as they are not non-reflecting boundaries.

Standard conditions at sea level were assumed for pressure and temperature at the domain inlet. The total conditions were computed isentropically for an inlet Mach

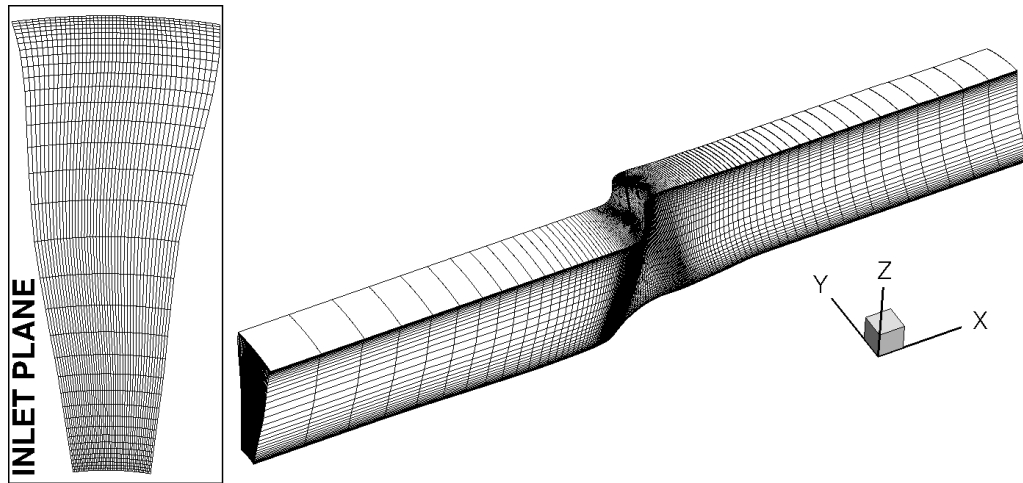


Figure VI.3: Full inviscid Rotor 67 computational grid with a detail of the inlet grid. All “j” grid lines have been omitted for clarity.

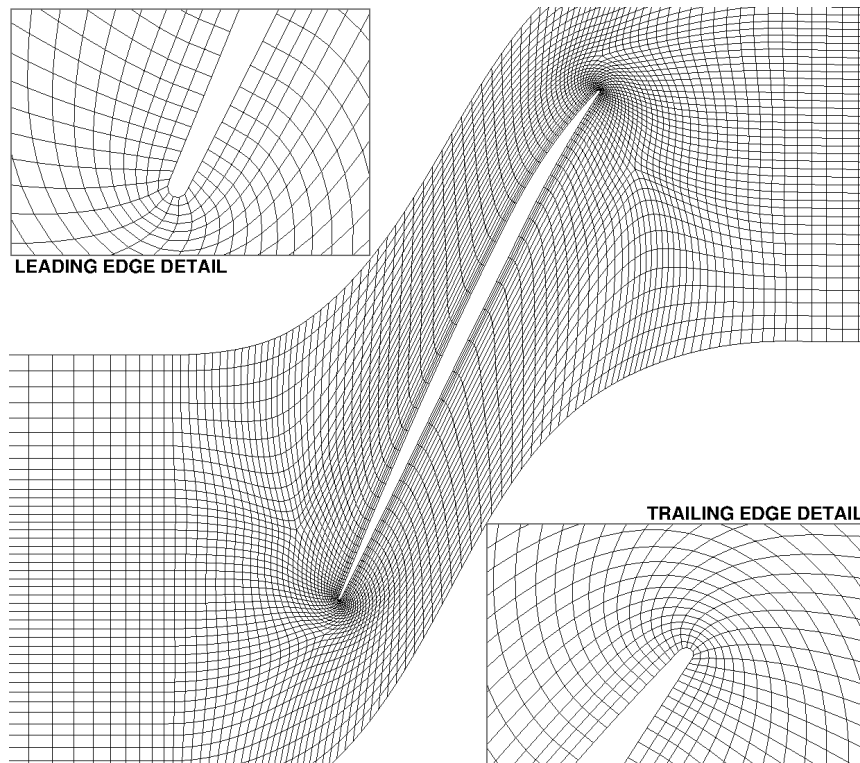


Figure VI.4: Tip blade-to-blade grid for the inviscid Rotor 67 case.

number of 0.55 and are given in Table VI.1. The outlet conditions were specified by setting the static pressure at the hub. The outlet static pressure was throttled to simulate different operating points on a speedline. The range of outlet static pressure used can be found in Table VI.1.

The flow conditions, including the wheel speed, from the literature are given at standard conditions, i.e. $p_{0_1} = 101325$ Pa and $T_{0_1} = 288.3$ K. As such, the reported wheel speed must be adjusted to give an equivalent wheel speed if the same flow conditions from the literature are to be matched. The equivalent wheel speed is a function of the nominal wheel speed and the total conditions, as shown previously in (3.31). The computed equivalent wheel speed and the equivalent pressure and temperature ratios (δ and θ , respectively) are shown in Table VI.1 for this case.

Table VI.1: Inviscid Rotor 67 boundary conditions.

Freestream Conditions	Inlet Conditions	Outlet Conditions
$p_\infty = 101325.0$ Pa	$p_{0_1} = 124452.598$ Pa	$0.85 \leq p_{hub}/p_{0_1} \leq 1.115$
$T_\infty = 288.3$ K	$T_{0_1} = 305.742$ K	
$M_\infty = 0.55$ –		
Equivalent Conditions		
$\delta = 1.2283$ –	$\theta = 1.0605$ –	$N_{EQ} = 16520.0$ RPM

VI.4. Initial Operating Point Analysis

The outlet condition for the initial operating point was set at $p_{hub}/p_{0_1} = 1.0$. Uniform freestream pressure, temperature, and velocity magnitude were used to initialize the flow field. The initial relative velocity vector was aligned to the local rotor chord. To start the flow, the solver was run first-order with the back pressure being applied uniformly across the outlet until the residuals dropped five orders of magnitude. At this point, the solver was then switched to second-order accurate and run until the residual variation per iteration was negligible. The outlet static pressure was determined using simple radial equilibrium for the second-order accurate simulations. The higher-order limiter used was the Dervieux limiter. The second-order input file for this operating point can be found in Appendix D (p. 259).

Figure VI.5 shows the residual histories for both the first-order and second-order simulations of the initial operating point. The second-order accurate simulation flow residuals dropped on average 3.3 orders of magnitude before their converged stalled. However, solution convergence could be, and was, declared once the change in the integral quantities of the flow (e.g. flow rate, pressure ratio, etc.) became negligible [Venkatakrisnan, 1995]. A total of 60,000 iterations was found to be sufficient to achieve this convergence criterion for the initial operating point.

VI.5. Compressor Map Analyses

This section discusses the analyses of a single compressor map speed line at design wheel speed. First, the procedure used to construct the speed line is presented. Following this, the solution residual histories for the speed line operating points are

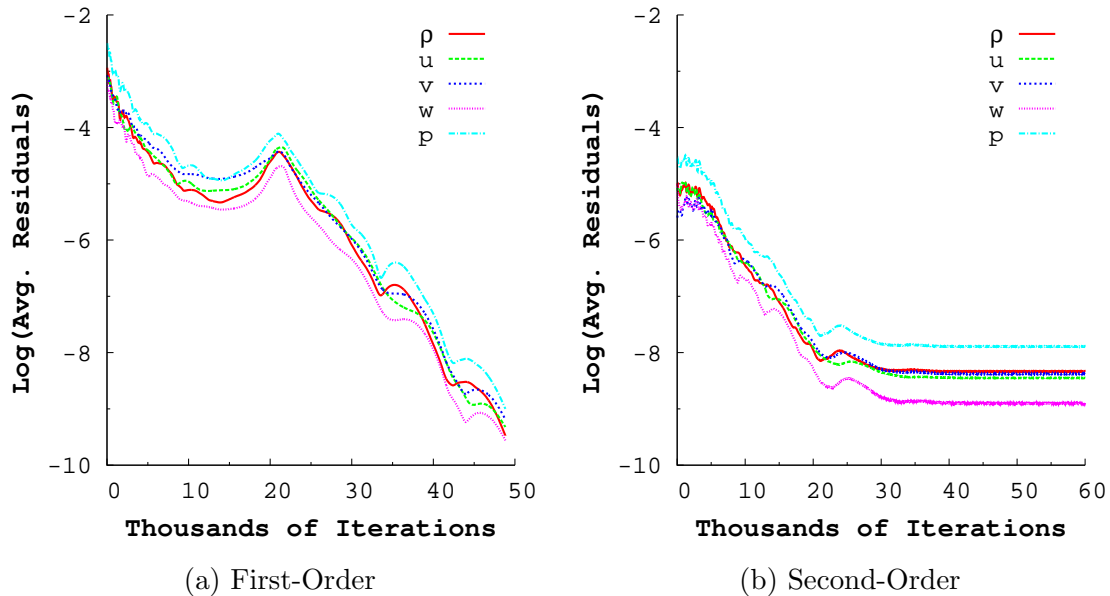


Figure VI.5: Inviscid Rotor 67 initial operating point residual histories.

discussed. Then the overall performance of the inviscid rotor is presented. Finally, a more detailed examination of three operating points is given.

VI.5.1. Compressor Map Generation Procedure

The initial operating point's second-order solution was used as the initial simulation field for the other operating points on the speed line. of the other points along the speed line. The second-order solution for an operating point with a slightly higher, or lower, outlet pressure was started directly from the initial solution. The same convergence criterion used for the initial operating point simulation was applied to the remaining operating point simulations. The closest converged operating point solution would be used as the starting point as the desired outlet pressures

moved further away from the initial value. In all, fourteen operating points were simulated to construct the inviscid speed line. The outlet boundary conditions used to construct the speed line can be found in Table A.1, on p. 218.

VI.5.2. Residual History Comparison

The density residual histories for all fourteen operating points are shown in Fig. VI.6. The residual history for the other variables were qualitatively similar to the density residuals and will not be presented here. Stalled convergence was explicitly seen for all but two operating points, $p_{hub}/p_{0_1} = 1.095$ and 1.100 .

The convergence of the residuals worsened as the back pressure was increased. As a result, the number of iterations required to meet the convergence criterion was increased by a factor of 3 in the worst performing cases. The convergence trend observed with the higher back pressure operating points was unsurprising. Increasing the back pressure pushes the solution towards the stall point, where the flow naturally becomes more unsteady and less periodic.

Operating points with a lower back pressure than the initial operating point had convergence rates similar to the initial point. The number of iterations required to achieve convergence was only slightly more than what was needed by the initial point. Convergence was also seen to stall after a smaller decrease in residual magnitude than for the initial point.

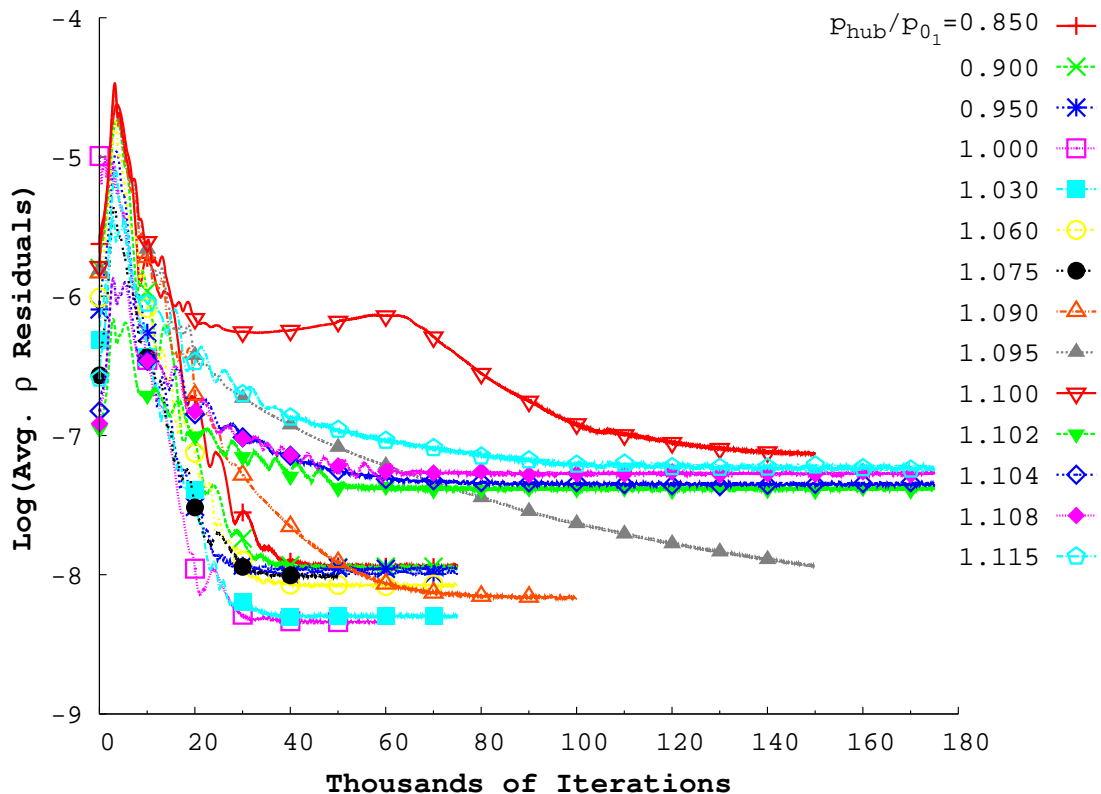


Figure VI.6: Density residual histories for the inviscid Rotor 67 speed line.

VI.5.3. Rotor Operating Line

This section presents the performance of the inviscid rotor at design wheel speed at all of the computed operating points. The evaluation of the equivalent mass flow rate is discussed first. Lastly, the compressor map speed line is presented.

Equivalent mass flow rate values were computed at the measurement planes (see Fig. VI.2) for each operating point. The ratio of outlet to inlet flow rate varied less than 0.1% for all of the operating points. The maximum predicted equivalent flow rate was $\dot{m}_{max} = 35.0478$ kg/s. The predicted maximum equivalent flow rate

agreed very well with the reported experimental maximum flow rates. On average, the predicted value was 0.3% higher than the experimental rates.

The computed performance parameters for each operating point were combined together to form a compressor map speed line. Figure VI.7 shows the predicted inviscid speed line compared against the experimental data by Pierzga & Wood [1985] and Wood et al. [1990]¹. Following Wood et al. [1990], the mass flow rate in Fig. VI.7 has been normalized by the maximum flow rate. In general, the inviscid analysis resulted in an over-prediction of total pressure ratio and adiabatic efficiency for all values of normalized flow rate. However, this was not unexpected as viscous effects and losses were not accounted for by this analysis.

VI.5.4. Select Operating Points

This section provides further analysis for the three operating points labeled in Fig. VI.7: “maximum flow”, “near peak efficiency”, and “near stall”. The “maximum flow”, “near peak efficiency”, and “near stall” operating points corresponded to the normalized outlet boundary conditions $p_{hub}/p_{0_1} = 0.85, 1.09, \text{ and } 1.108$, respectively. The inviscid analyses of the three operating points were compared against the experimental data compiled by Pierzga & Wood [1985] and Wood et al. [1990]. Table VI.2 summarizes the performance characteristics of each operating point and gives comparison to the experimental data.

¹The data plotted in Fig. VI.7 are also tabulated in Table A.1 on p. 218.

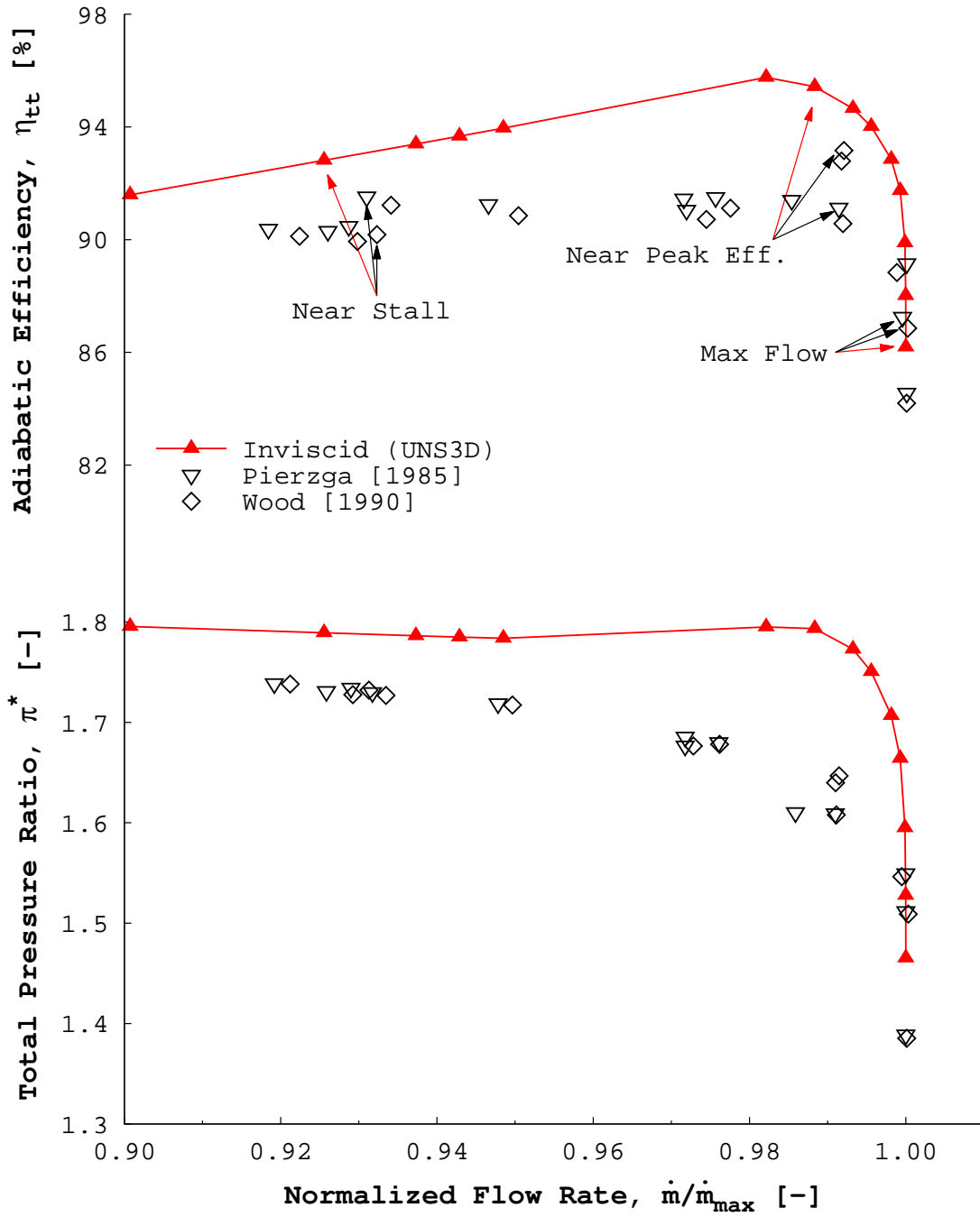


Figure VI.7: Predicted inviscid rotor performance at design wheel speed compared against experimental data [Pierzga & Wood, 1985; Wood et al., 1990].

Table VI.2: Predicted and measured rotor performance at maximum flow, near peak efficiency, and near stall.

	Maximum Flow			Near Peak Efficiency			Near Stall		
	\dot{m}/\dot{m}_{max}	π^*	η_{tt}	\dot{m}/\dot{m}_{max}	π^*	η_{tt}	\dot{m}/\dot{m}_{max}	π^*	η_{tt}
UNS3D [†]	1.000	1.465	0.8619	0.988	1.793	0.9543	0.926	1.789	0.9282
Pierzga & Wood [1985] [‡]	1.000	1.550	0.8722	0.991	1.608	0.9111	0.931	1.730	0.9149
Wood et al. [1990] [‡]	1.000	1.509	0.8686	0.992	1.647	0.9316	0.932	1.731	0.9017

[†] $\dot{m}_{max} = 35.05$ kg/s [‡] $\dot{m}_{max} = 34.92$ kg/s [‡] $\dot{m}_{max} = 34.96$ kg/s

Table VI.2 confirms that the inviscid analysis, in general, results in an over-prediction of total pressure ratio and adiabatic efficiency. The exception, in this case, lies at the maximum flow operating point where the analysis resulted in an under-prediction of both performance parameters. The under-prediction can be explained by the experimenter's choice of operating point to label as the "maximum flow" operating point. Figure VI.7 shows experimental data points which lie below the indicated maximum flow points. Had any of these operating points been chosen by the experimenters, then the analysis would then have provided an over-prediction for this operating point.

Contours of the relative Mach number were extracted at 70% of the span² for the three operating points. The predicted flow fields were compared against laser anemometry data collected and processed by Pierzga & Wood. The experimental data includes the locations of the bow and passage shocks that were indicated by the measured data. The experimental shock locations are shown as the thick dashed lines in the following figures.

Figure VI.8 compares the predicted and measured flow fields for the maximum flow operating point. The analysis provided a good prediction of the upstream flow field in comparison to the measured data. However, at this operating point, the predicted flow between the rotor blades was qualitatively different than the experimentally measured passage flow. The analysis showed that the flow was accelerated through the entire length of the blade passage until meeting a passage shock, located

²Percent span is measured from the hub.

at the trailing edge. While a passage shock was present in the real flow, its location was much further upstream than what was predicted in the analysis.

The inviscid flow predicted by UNS3D at maximum flow was found to be qualitatively similar to the inviscid flows predicted by Pierzga & Wood [1985] and Subramanian & Bozzola [1987] for the same operating point. Both authors predicted the same location of the passage shock as the current work. Subramanian & Bozzola [1987] went on to show that the experimentally measured flow field could be predicted with the inclusion of viscous effects.



Figure VI.8: Predicted (left) and measured [Pierzga & Wood, 1985] (right) relative Mach number contours at maximum flow, and at 70% span.

Figures VI.9 and VI.10 compare the predicted inviscid and experimentally measured flow fields at the near peak efficiency and near stall operating points, respectively. Both figures indicate good qualitative agreement between the inviscid analysis and experiment for these operating points. The analysis resulted in lower relative

Mach numbers leaving the rotor passage. As a result, higher static pressure increases across the rotor were achieved by the simulations.

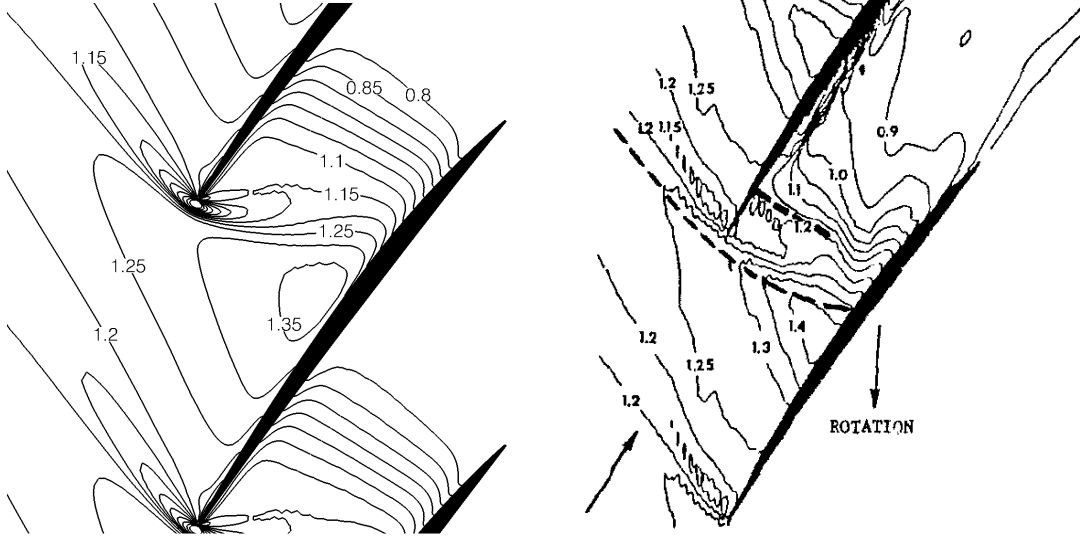


Figure VI.9: Predicted (left) and measured [Pierzga & Wood, 1985] (right) relative Mach number contours near peak efficiency, and at 70% span.

VI.6. Effect of Higher-Order Limiters

This section presents the effect of the choice of higher-order limiter on the inviscid solution. In the previous analyses, the Dervieux limiter was used exclusively. For this study, both versions of the Venkatakrishnan and MLP limiters, as defined in Section III.2.2, were used for a single operating point simulation. The initial operating point ($p_{hub}/p_{0_1} = 1$) was chosen such that all of the simulations were started from the original converged first-order solution.

The higher-order limiter simulations were run until the solutions were suffi-

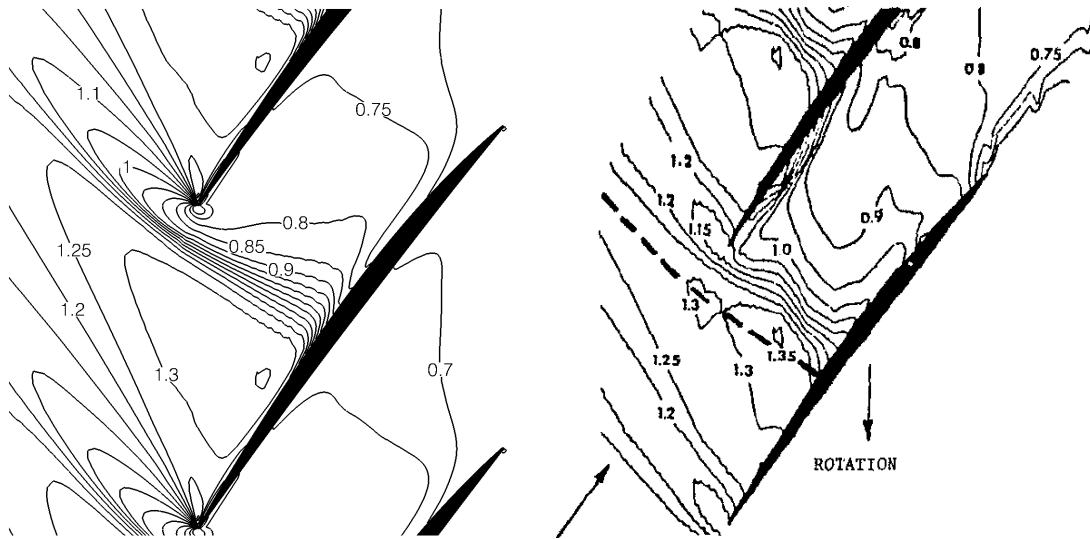


Figure VI.10: Predicted (left) and measured [Pierzga & Wood, 1985] (right) relative Mach number contours near stall, and at 70% span.

ciently converged using the previous convergence criterion. Figure VI.11 shows the convergence of the density residuals of the Venkatakrishnan and MLP limiters compared against the previous Dervieux limiter result. It was found that the standard Venkatakrishnan (Venka- ϵ_1) and MLP (MLP- ϵ_1) limiters resulted in the worst reduction in residual magnitude. Using the ϵ^2 modification with both limiters allowed for a slight improvement in convergence level. The modified Venkatakrishnan limiter (Venka- ϵ_2) performed the best with the modification, reaching a density residual magnitude reduction of 3.4. For comparison, the Dervieux limiter also reached a density residual magnitude reduction of 3.4.

The solutions for the higher-order limiter cases were extracted at the measurement planes to compute equivalent mass flow rates, total pressure ratios, and adiabatic efficiencies for each case. The computed rotor performance parameters are

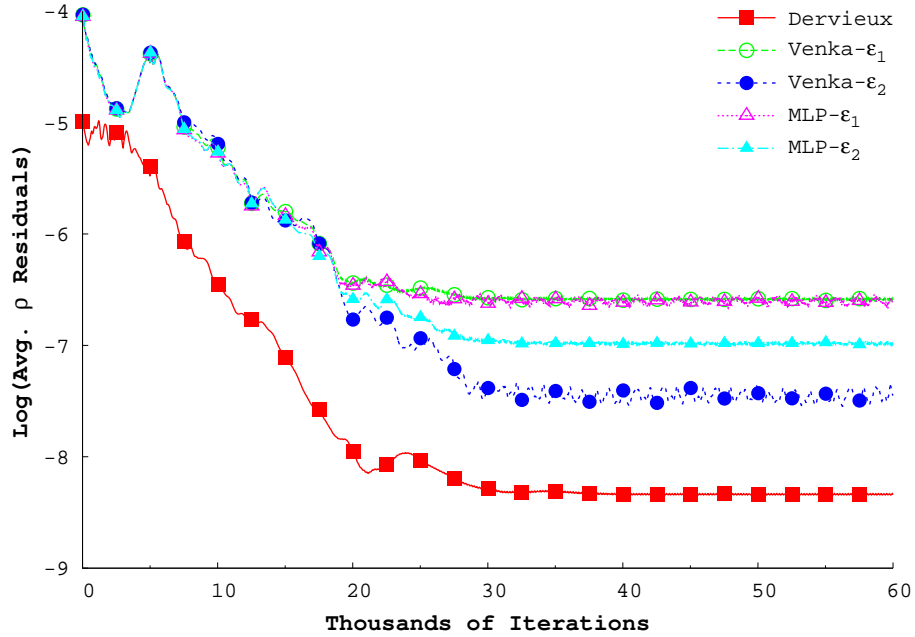


Figure VI.11: Density residual histories for the initial inviscid Rotor 67 operating point using different higher-order limiters.

collected in Table VI.3, where they are compared against the original Dervieux solution. The table shows that both versions of the Venkatakrishnan and MLP limiters resulted in slightly higher values for all of the computed performance parameters. The maximum absolute percent differences for mass flow rate, total pressure ratio, and adiabatic efficiency were 0.24%, 0.42%, and 0.35%, respectively. The Dervieux limiter, however, resulted in a solution with a ratio of outlet to inlet mass flow rate slightly closer to unity than all of the others. From the table, one could state that the solutions were nearly identical.

The extracted solutions at the downstream measurement plane for each case were circumferentially averaged at each grid plane to construct spanwise profiles

Table VI.3: Initial operating point rotor performance with respect to the limiter used.

Limiter	\dot{m}_{eq} [kg/s]	$\dot{m}_{out}/\dot{m}_{in}$	π^*	η_{tt}
Dervieux	35.023	0.9992	1.664	0.9174
Venka- ϵ_1	35.106	0.9976	1.669	0.9205
Venka- ϵ_2	35.105	0.9977	1.669	0.9206
MLP- ϵ_1	35.061	0.9986	1.671	0.9197
MLP- ϵ_2	35.062	0.9985	1.670	0.9194

of static pressure and absolute flow angles. Both downstream profiles are shown in Fig. VI.12, where the static pressures have been normalized by the specified inlet total pressure. The downstream pressure profiles were nearly quantitatively identical. The downstream radial profiles of absolute flow angle were qualitatively similar. In comparison to the Dervieux limiter solution, the Venkatakrishnan and MLP limiters produced slightly more turning in the upper 60% of the blade's span and slightly lower turning in the bottom 40%. The largest observed difference of 1.8° took place at 100% span, between the Dervieux and Venka- ϵ_1 limiters.

Contours of the relative Mach number were extracted at 70% span for all five limiter cases. The contour plots are collected in Fig. VI.13, where the Dervieux limiter case was the one used in the previous compressor map speed line analysis. The computed contours showed that all five limiters produced nearly identical flows. The most notable difference was that in both versions of the Venkatakrishnan limiter,

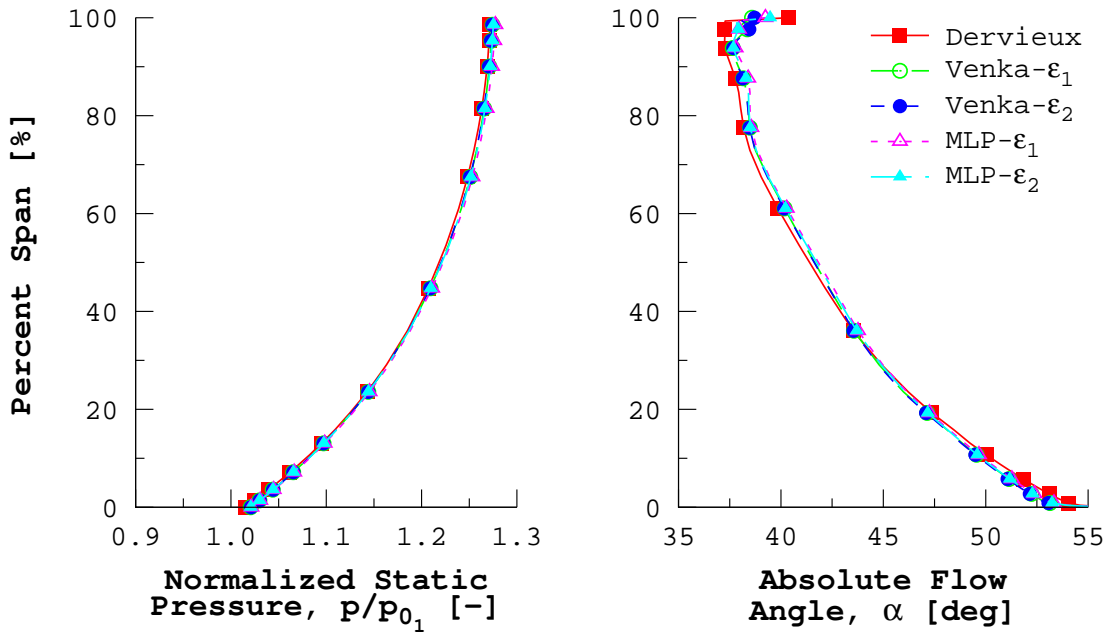


Figure VI.12: Higher-order limiter solution comparison using downstream profiles of normalized static pressure and absolute flow angle.

there was small bubble of faster flow present just off the blade's surface at about 50% of the axial chord.

A more subtle difference between Dervieux and the other limiters was found in the Mach contours around the passage shock near the trailing edge. With the Dervieux solution, the relative Mach contours were more tightly packed in the region near the suction side, and the contour lines appear more "serrated" in the approximate region where the shock front was located. On the other hand, the relative Mach contours for both the Venkatakrishnan and MLP appear much cleaner with less "serration" in the contours and a more uniform gradient across the shock region. These solution differences in the vicinity of the passage shock are the result of the

limiters themselves and explain the slight differences in downstream flow angle that were observed.

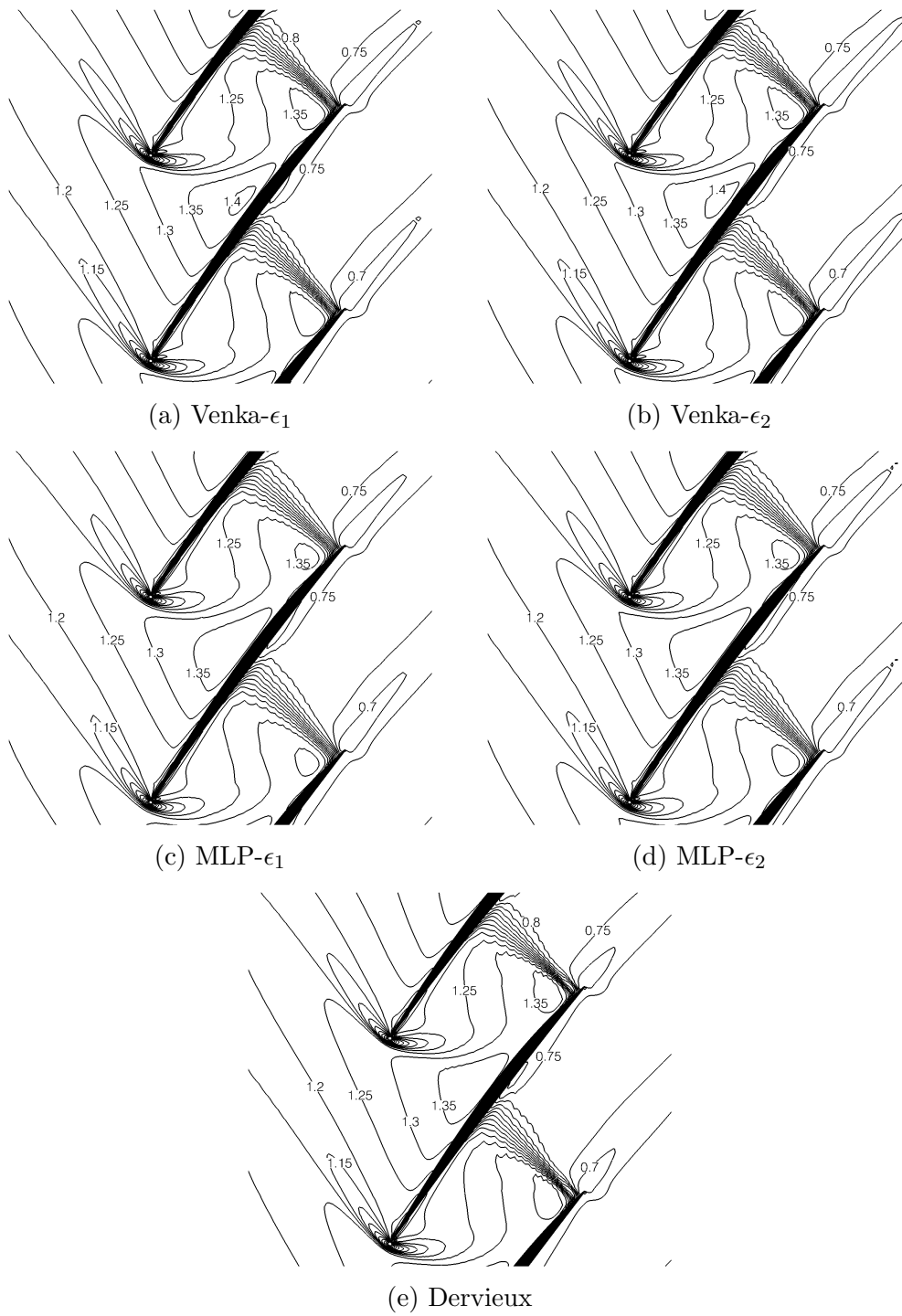


Figure VI.13: Predicted relative Mach number contours at 70% span for different higher-order limiters.

CHAPTER VII

TRANSONIC FAN TURBULENT ANALYSES

The chapter presents the results of the turbulent analyses for two transonic fan rotor configurations. In the first configuration, the rotor is allowed to “rub” against the outer casing, as was done for the analysis in the previous chapter. The flow around the tip of the rotor was included in the second configuration via the use of tip clearance grids. The NASA Rotor 67 geometry was once again used for these analyses. The details of the rotor geometry are not repeated in this chapter, and as such, the reader is referred back to Section VI.1 (p. 135) for a detailed description.

The first section of this chapter describes the methodology used to compute the operating lines for both rotor configurations. The rubbing rotor configuration, in which two wall boundary definitions are explored, is then presented. The last section presents the realistic rotor configuration which includes the tip leakage flow. Comparisons with experimental data are given for both geometry configurations.

VII.1. Operating Line Generation Procedure

This section describes the procedure used to generate the rotor’s operating line. The outlined solution strategy was used for each rotor configuration found in this chapter. Any deviations from the given procedure for a specific case will be highlighted in the appropriate sections.

The simulation of the operating line began with the calculation of a single operating point, referred to herein as the “initial operating point”. The flow solution

was initialized for this operating point using a previously computed inviscid solution. Nearest node interpolation was used to transfer the inviscid solution onto the turbulent mesh. The turbulence field was uniformly initialized from the inlet turbulence boundary conditions.

The same solver options, unless otherwise noted, were used throughout the simulations found in this chapter. Gradients were evaluated via least-squares with QR decomposition (LSQR), and the Dervieux limiter was used with the second-order simulations. Outlet conditions were set using the method described in Section III.3.3. Both turbulence equations were under-relaxed by a factor of 0.75 and no relaxation was applied to the flow equations.

At the initial operating point, the flow solver was first simulated with first-order accuracy for 100,000 iterations. The first-order simulation allowed for the formation of boundary layers and to give both the flow and turbulence solutions the opportunity to settle. The flow solver was then simulated using second-order accuracy until the convergence criteria had been satisfied.

The convergence criteria for the calculation of all operating points was based on the integral properties of the flow: mass flow rate, total pressure ratio, and adiabatic efficiency. Convergence was declared once the absolute change in each integral quantity had become sufficiently small. A value of 0.01% was used as the convergence threshold value.

The initial operating point was used as the basis to compute the remaining operating points found on the operating line. Each new operating point was initialized from the closest converged operating point solution. Each operating point was

simulated with second-order accuracy in the same manner as the initial operating point.

The outlet static pressure was varied to achieve the different operating points. The pressure was lowered from the value at the initial operating until the mass flow rate no longer varied, indicating choked flow. On the other side of the initial operating point, the pressure was raised until the rotor solution exhibited “stall”.

VII.2. Rubbing Configuration

This section presents the solutions for two rubbing Rotor 67 cases. The two cases differed in the implementation of the viscous wall boundary in the domain. The computational grid that was used for both cases, and the boundary conditions that were applied are discussed first. Secondly, the compressor map speed lines at design wheel speed for both cases are presented. Finally, a more detailed analysis of the solution at select operating points is given.

VII.2.1. Computational Grid

This section covers the generation of the computational grid used for both rubbing cases. The in-house grid generator described in Section III.1 was used to build the grid, and the input file used can be found in Appendix E, starting on p. 280. The general configuration and size of the grid is first presented. Finally the domain decomposition used to prepare the grid for parallel simulations is discussed.

A single blade passage was constructed using the O-4H configuration. Additional H-grids were added to move the inlet and outlet boundary planes five axial hub chords

away from the rotor. For a given grid layer, 167 grid points were used to define the local airfoil section, with 173 being used to define the passage between two rotors. The tip layer blade-to-blade grid can be seen Fig. VII.1a. Grid points were clustered at both the leading and trailing edges as shown in Fig. VII.1b–c. One hundred thirty five grid layers were used to define the span of the rotor. In all, the grid contained 2,474,550 unique nodes.

The distance to the first node away from the rotor surface was varied in both the streamwise and spanwise directions. The variation in cell height was achieved using the method previously described in Section III.1.6. The upstream velocity profile required by the method was taken from the performance data compiled by Urasek et al. [1979].¹ The minimum first cell height was computed to be 4.8×10^{-5} cm at the leading edge of the tip profile. The difference in cell heights for a given grid layer are evident in Fig. VII.1b–c, and the overall variation in first cell height for this grid is shown in Fig. VII.2.

Constant viscous spacing was applied to both the hub and tip endwalls. The viscous spacing along the hub endwall was specified as $z_1^{(h)} = 1.83 \times 10^{-4}$ cm. The tip endwall spacing was set as half of the hub spacing, $z_1^{(t)} = 9.144 \times 10^{-5}$ cm.

The computational grid was split using domain decomposition in for parallel computations. The grid was split into 40 blocks by first equally subdividing the domain five times along the spanwise grid layers. Then each of the five spanwise blocks were split into two blocks according a pitch-normalized angle. Finally the grid blocks were split equally four times in the axial direction. Figure VII.3 shows

¹The relative velocity magnitudes were taken from Reading 1393, found in Table IX(c).

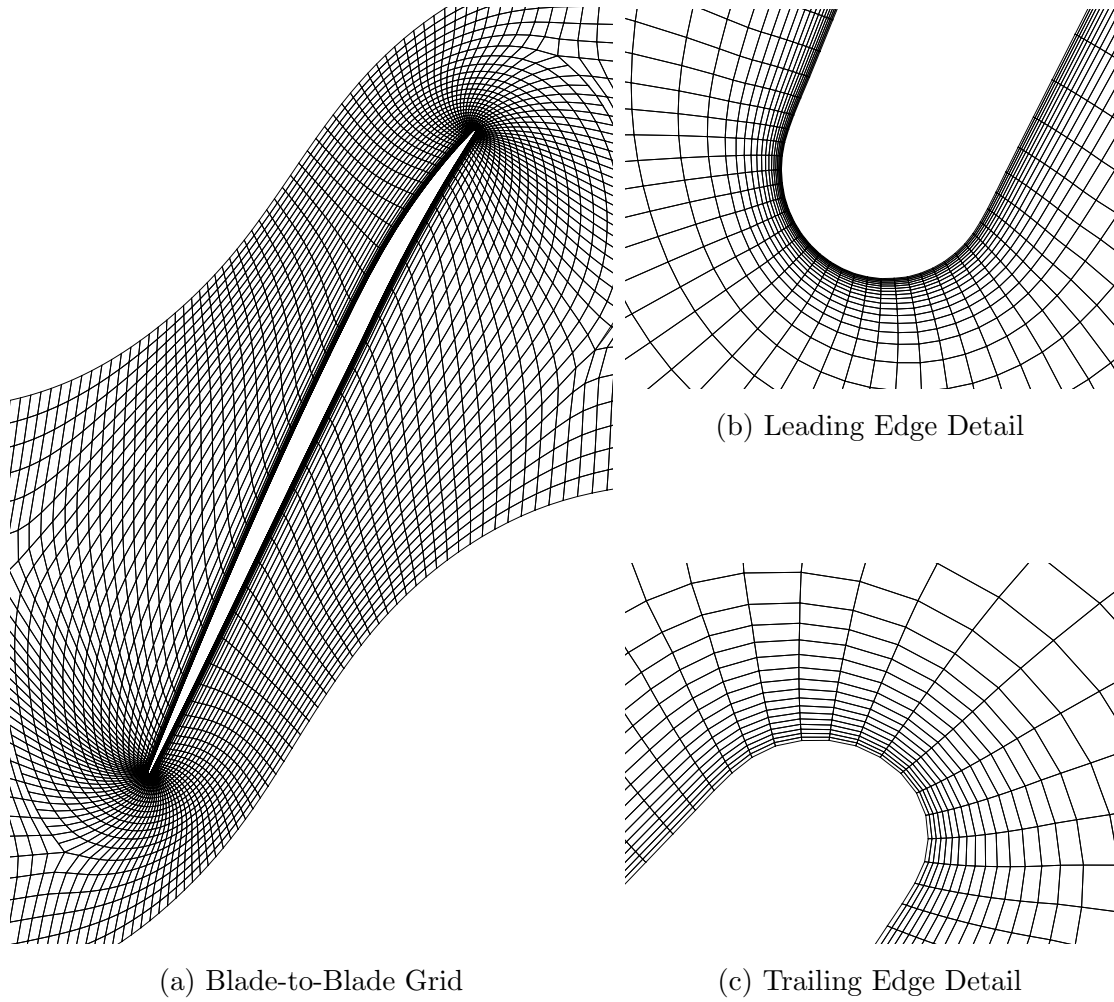


Figure VII.1: Turbulent rubbing Rotor 67 tip layer blade-to-blade grid. Every other j^{th} grid point has been removed for clarity. Leading and trailing edge details are shown at the same magnification level.

the resultant domain decomposition that was used. On average, the decomposed grid blocks contained approximately 75,000 nodes, or roughly 3% of the original grid nodes.

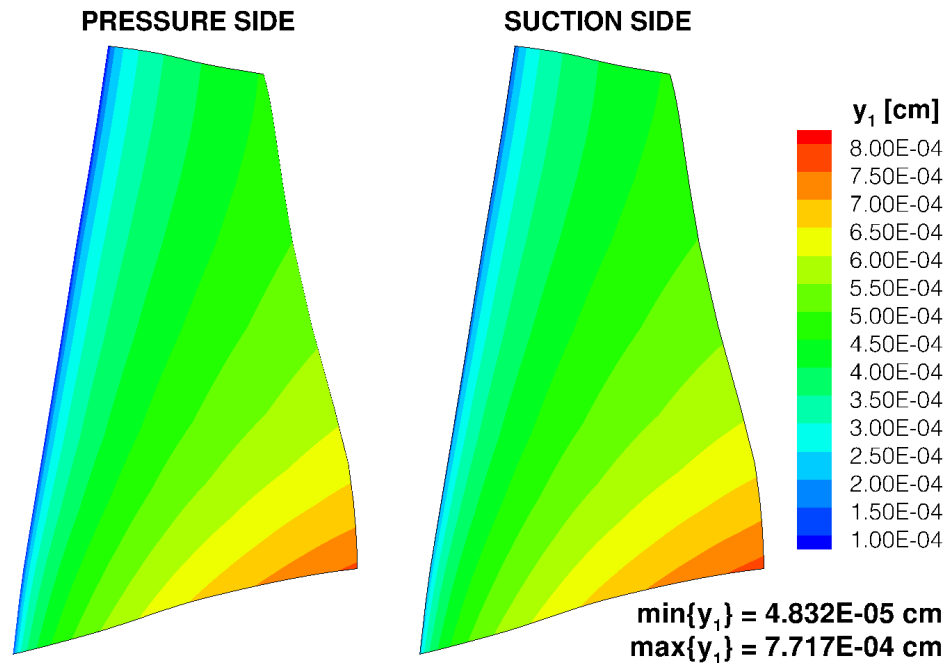


Figure VII.2: First cell height variation across the rotor surface.

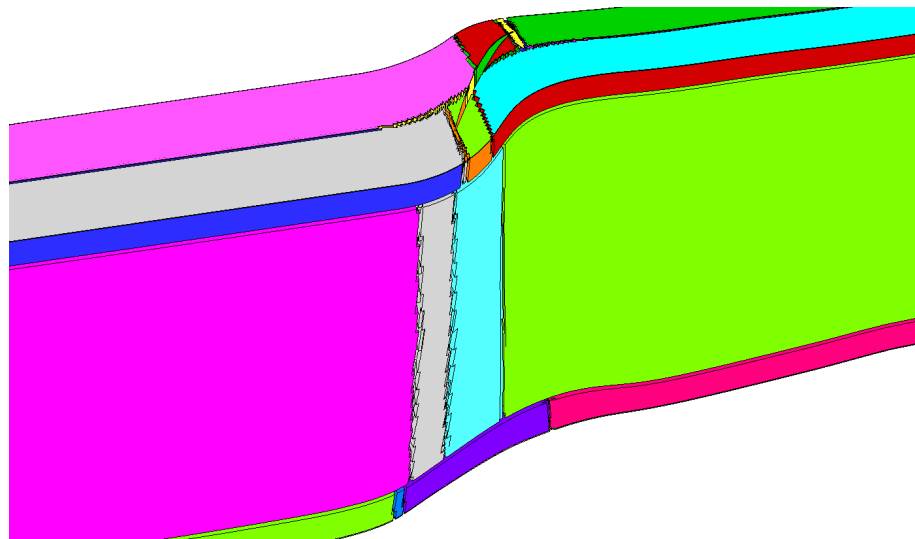


Figure VII.3: Domain decomposition breakdown for the turbulent rubbing Rotor 67 grid. Each color represents a separate processor block.

VII.2.2. Boundary Conditions

This section presents the boundary conditions that were applied to the two turbulent rubbing rotor cases. The three boundary conditions that were applied to the grid were: inlet, outlet, and rotating viscous wall (applied to all walls). As was previously stated, the inlet and outlet boundaries were moved five axial hub chords away from the rotor to reduce the impact of any reflected waves on the solution.

The option exists in UNS3D to allow only a certain portion of the rotating walls to actually rotate with the given wheel speed. A wall boundary node is allowed to rotate if its axial location, x_b , satisfies $xr_{le} \leq x_b \leq xr_{te}$, where xr_{le} and xr_{te} are defined in Fig. VII.4. The rotational speed of the walls is set to zero when the axial location of the wall boundary node falls outside of the given range, as illustrated by the dark grey regions in Fig. VII.4.

The two turbulent rubbing cases are differentiated by how the wall boundary conditions were utilized. In the first case, known as R67TR-a, $xr_{le} = -10^{30}$ cm and $xr_{te} = 10^{30}$ cm; that is to say, that all of the rotating walls were allowed to rotate. For the second case, called R67TR-b, $xr_{le} = -1.374$ cm and $xr_{te} = 9.365$ cm. The values chosen correspond to the location of the rotating hub in the experiments by Wood et al. [1990]. The resultant surfaces that are allowed to rotate are shown in Fig. VII.4 as the light grey surfaces.

A third option for the wall boundary conditions would be to specify that the tip endwall be completely fixed, and apply rotating wall boundary conditions to the blade and hub endwall. Aside from still allowing the rotor to rub against the tip endwall, this configuration is the most representative of the actual machine. However,

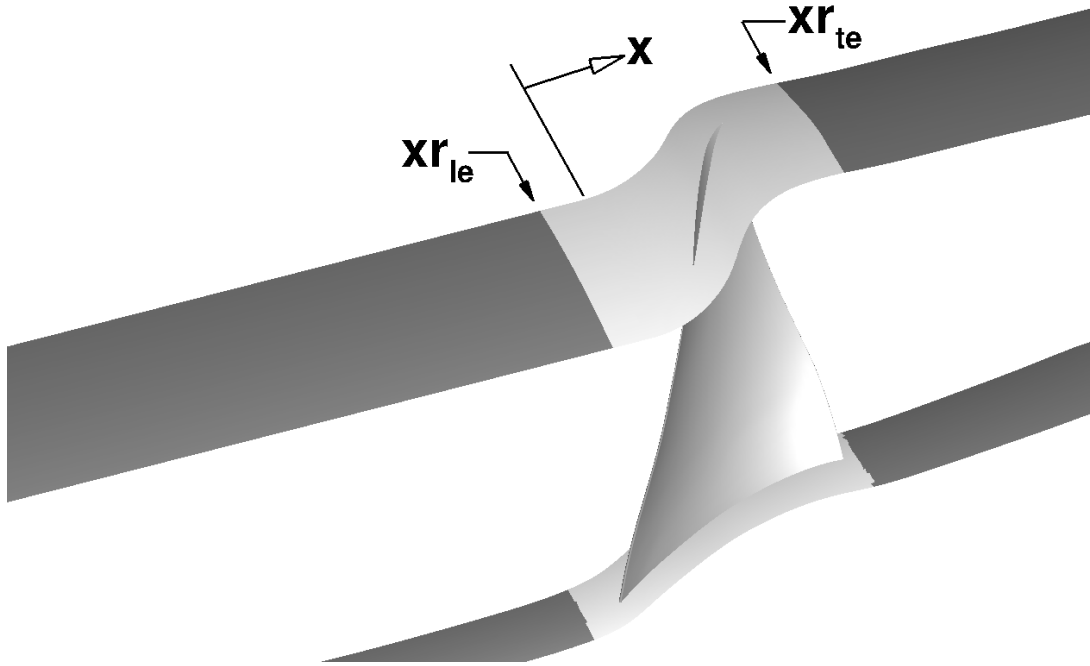


Figure VII.4: Defining the allowable region (light grey surfaces) for which rotating wall boundaries can rotate. The figure shows the rotation boundaries for Case R67TR-b.

using this wall boundary configuration in UNS3D resulted in solution divergence very early in all attempted simulations with this configuration. The location at which the solution diverged was always found to be at the intersection of the blade tip and the endwall. No suitable explanation was found for the cause of the divergence. The wall boundary configurations of the cases described within this section were derived as a direct result of the divergence issues encountered.

The freestream and inlet flow conditions used for both cases were identical to those used previously in the inviscid simulations. The same was true of the equivalent wheel speed. These conditions have been repeated in Table VII.1 for completeness.

The total range of normalized static pressure used in the simulation of both cases is also given in Table VII.1. The maximum normalized static pressure that was used in either of the two cases was 1.04, which approximately 8% less than the maximum outlet static pressure achieved with the inviscid simulations.

The turbulence conditions within the computational domain were set by the turbulence intensity and length scale at the inlet. Using the results of the turbine vane study as a guide, the turbulence intensity was set to 3%. The turbulent length scale was set based on the distance between the blade tip and the outer endwall, $\ell_T|_\infty = 1.04$ mm. Using the prescribed inlet turbulent quantities, the ratio of viscosities at the inlet was computed to be $(\mu_T/\mu)_\infty = 44.7$.

Table VII.1: Turbulent rubbing Rotor 67 boundary conditions.

Freestream Conditions	Inlet Conditions	Outlet Conditions
$p_\infty = 101325.0$ Pa	$p_{0_1} = 124452.598$ Pa	$0.864 \leq p_2^{(h)}/p_{0_1} \leq 1.04$
$T_\infty = 288.3$ K	$T_{0_1} = 305.742$ K	
$M_\infty = 0.55$ –		
Inlet Turbulence Conditions		
$\ell_T _\infty = 1.04$ mm		$I_T _\infty = 3\%$
Equivalent Conditions		
$\delta = 1.2283$ –	$\theta = 1.0605$ –	$N_{eq} = 16520.0$ RPM

VII.2.3. Compressor Map Speed Line Analyses

This section presents the results of a compressor map speed line analysis for both rubbing configurations. A nonphysical boundary solution that was observed for case R67TR-b is presented first. The calculation of the operating points and a definition for “stall” are then discussed. Finally the performance of the rotor across the range of operating points is presented for both cases.

Speed lines for both viscous wall boundary configurations were computed using the procedures presented in Section VII.1 with two exceptions. The flow at the initial operating point for case R67TR-b was initialized using a uniform flow field with the relative velocity aligned to the local chord. Also, the outlet static pressure was assumed to be constant across the span during the first-order simulation of R67TR-b. Sample input files for the flow solver can be found for both rotor configurations in Appendix D, starting on p. 263.

VII.2.3.1. Nonphysical Boundary Solution

A peculiarity observed for Case R67TR-b was that at a single point in the flow the solution became nonphysical. The nonphysical point was located $0.94c_a^{(t)}$ ahead of the leading edge at the periodic boundaries of the tip grid layer, as shown in Fig. VII.5a. The axial location of this point also corresponded to the point to $x = xr_{le}$. What made this nonphysical point peculiar was that its affect on the solution was limited to the nodes immediately adjacent, and did serve as a detriment to the overall solution. Figure VII.5b shows that at most five nodes on the tip grid layer were affected by the bad point. The nonphysical solution was seen to permeate

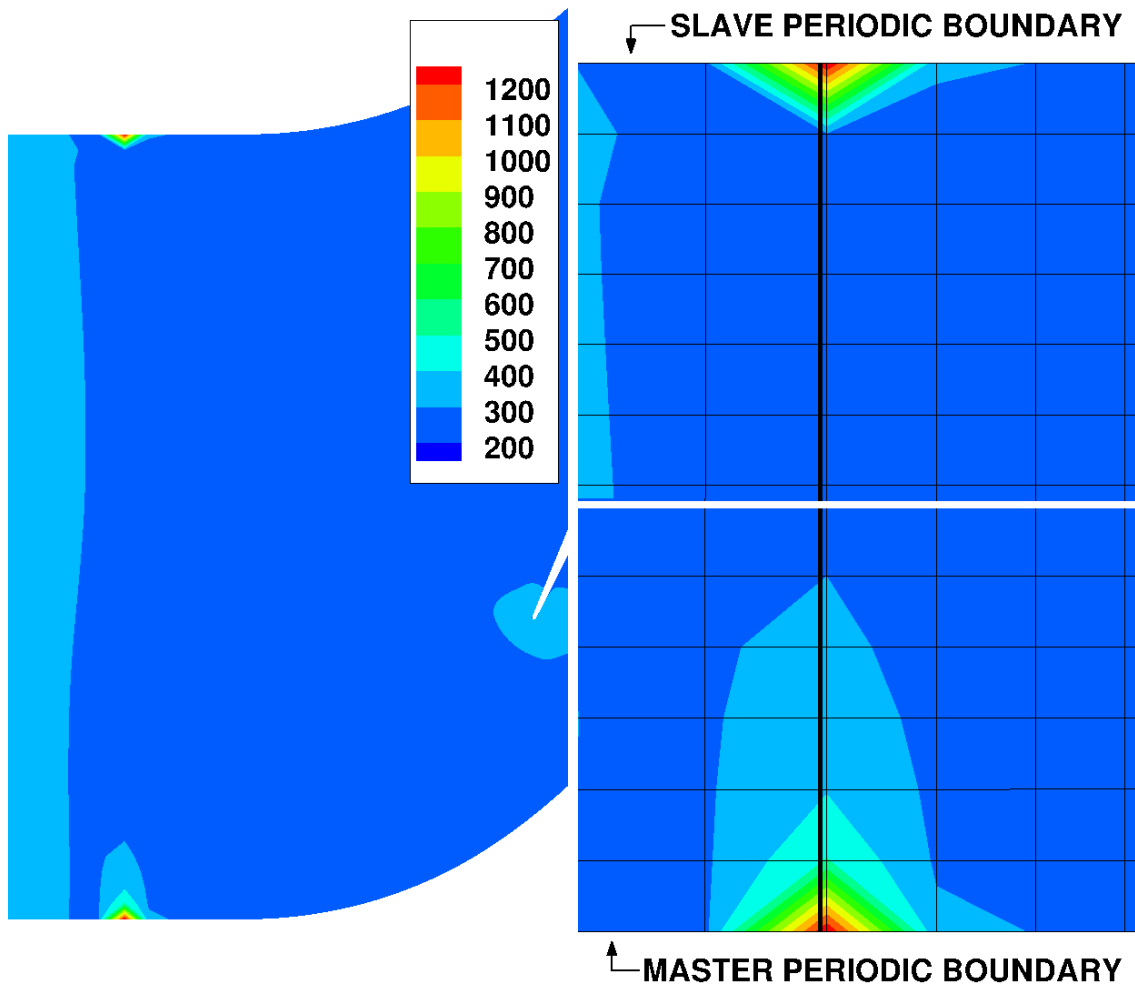
up to 15 points in the spanwise direction away from the wall, but again, the affect of the nonphysical solution on the surrounding nodes was constrained as observed on the tip grid layer.

The viscous rotating wall boundary condition was called into question due to the location of the point being at $x = xr_{te}$. However, a thorough inspection of the boundary condition yielded no answers as the implementation of the boundary condition was verified. Ultimately, the existence of the nonphysical point was deemed to be “OK” because it was not affecting the solution anywhere else in the domain, and because a realistic overall solution was obtained in spite of it. The nonphysical point would be found in all other operating point solutions as well, but again, no affect of the point on the rest of the predicted flow was evident in any case.

VII.2.3.2. Operating Point Simulations

This section describes the calculation of the operating points that were used to define the speed line for each configuration. Ten operating points for case R67TR-a and eight for case R67TR-b were simulated. The boundary conditions for each operating point are given in Table A.2 and Table A.3 for cases R67TR-a and R67TR-b, respectively.

The density and turbulent kinetic energy residual histories at four operating points for each case are shown in Fig. VII.6. The four cases, in order of increasing $p_2^{(h)}$, correspond to: maximum flow, an intermediate point, the final point before stall, and the first stall point. The figure shows that only 1 to 2 orders of magnitude decrease in the density residuals was achieved in either case for the non-stall operating points.



(a) Nonphysical Point Locations

(b) Grid Details

Figure VII.5: Static temperature contours showing the nonphysical solution point for Case R67TR-b on the tip grid layer. Thick vertical lines in (b) are at $x = xr_{le}$.

The remaining four flow residuals exhibited trends similar to the density residual. The decrease in magnitude of the solution residuals, for any variable, decreased as the back pressure was increased until stall occurred. Once stall occurred the residuals would show a divergent trend and increase multiple orders of magnitude before decreasing once more, as shown in Fig. VII.6.

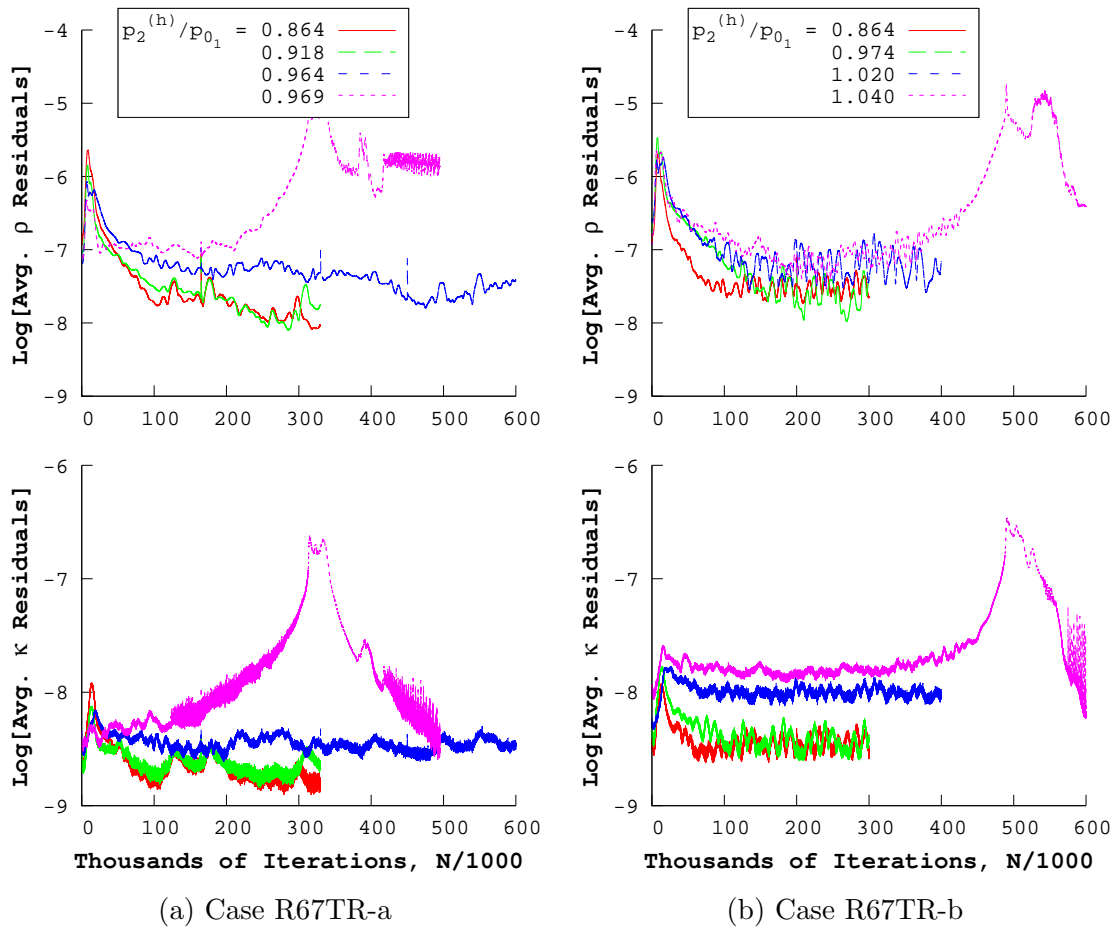


Figure VII.6: Density and turbulent kinetic energy second order residual histories for both turbulent rubbing Rotor 67 cases.

VII.2.3.3. Defining Stalled Flow

The stalled rotor flow was characterized by large regions of separated and reversed flow in the upper 15% span of the rotor. The region of reversed flow that occurred during stall for Case R67TR-a is illustrated in Fig. VII.7 using iso-surfaces of negative axial velocity. Downstream of this reversed flow region, the axial velocity was an order of magnitude smaller than the relative velocity in the y-direction. As

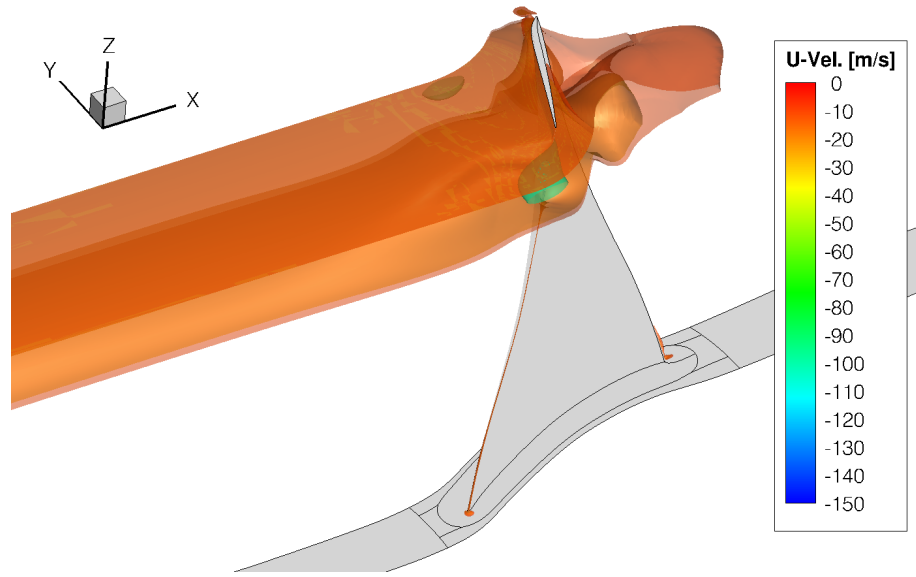


Figure VII.7: Iso-surfaces of axial velocity showing large regions of reversed flow indicating stall for Case R67TR-a.

a result, the flow direction tended to be more tangential than axial. Mass flow rate served as an excellent indicator of stall due to these poor flow conditions. During stall, the mass flow rate would drop quickly to be on the order of half of the maximum flow rate, $\dot{m}_{stall}/\dot{m}_{max} \approx 0.5$. Three of the ten operating points for R67TR-a resulted in stall, while one for R67TR-b stalled.

VII.2.3.4. Rotor Operating Line

This section presents the overall performance characteristics predicted by the turbulent simulation of the two rubbing configurations. The maximum mass flow rates for each grid configuration are first discussed. The compressor map speed line for both cases is then presented. Finally, the spanwise variation of the flow at “near peak efficiency” is discussed.

The predicted maximum equivalent mass flow rates computed at the upstream measurement plane are summarized in Table VII.2. For the two cases, the predicted mass flow rates were lower than the measured values. For case R67TR-a, in which all the walls were rotated, the maximum predicted mass flow rate was 3.2% lower than experiment. The predicted maximum flow rate for case R67TR-b, where only a small portion of the walls rotated, was closer to the experimental values with a difference of 2.1%. The differences in both cases were larger than the largest difference found in the literature, which was 1.32% [Arnone, 1994].

Table VII.2: Maximum mass flow rates for both turbulent rubbing Rotor 67 cases with comparison to experiment [Pierzga & Wood, 1985; Wood et al., 1990].

R67TR-a	R67TR-b	Pierzga & Wood	Wood et al.
33.834 kg/s	34.215 kg/s	34.92 kg/s	34.96 kg/s

R67TR-a had a lower maximum flow rate compared to R67TR-b due to the inclusion of the rotating endwalls. With the wall boundary conditions written using absolute velocities, the absolute velocity of the wall is equal to local wall speed. As a result, flow is entrained in the direction of the rotation and causes a deficit in the axial velocity near the endwalls. Slower axial flow near the end walls means less mass flow rate in these regions, which reduces the overall mass flow rate of the machine. This effect is most notable at the tip endwall where the rotational speed results in an absolute Mach number of 1.4 upstream of the rotor.

The flow was extracted and averaged at the measurement planes for each op-

erating point and case to construct the speed lines shown in Fig. VII.8. The mass flow rates for each data set have been normalized by the maximum value to provide better qualitative comparison between data sets. The operating points that experienced stall have been excluded from the plots as cases that fell well outside the range of the experimental data.

It was found that for the points simulated, the operating range of the rotor was approximately half of the range measured in the experiments. The simulation of additional points could increase the operating envelope of the rotor. This was not done as finding the “near peak efficiency” point was the desired goal. However, the difference in $p_2^{(h)}$ between the stalled and not stalled points was relatively small. This would indicate that only one or two additional operating points could be feasibly added to increase the predicted operating range.

The predicted adiabatic efficiency, η_{tt} , of the rotor showed good agreement for the operating points simulated as shown in Fig. VII.8. The “near peak efficiency” point was defined as the point at which maximum efficiency occurred. The adiabatic efficiencies at “near peak efficiency” are summarized in Table VII.3 for both cases. The predicted efficiencies for both cases were within 1.8% of the measured values. The predicted “near peak efficiency” points occurred at normalized mass flow rates that were 0.8% lower than the measured point.

The predicted total pressure ratios, π^* , were found to qualitatively match the trend of the experimental data, as shown in Fig. VII.8. Values predicted at “near peak efficiency” for each rubbing case are summarized in Table VII.3. In general, rubbing case R67TR-b quantitatively matched the experimental data much closely

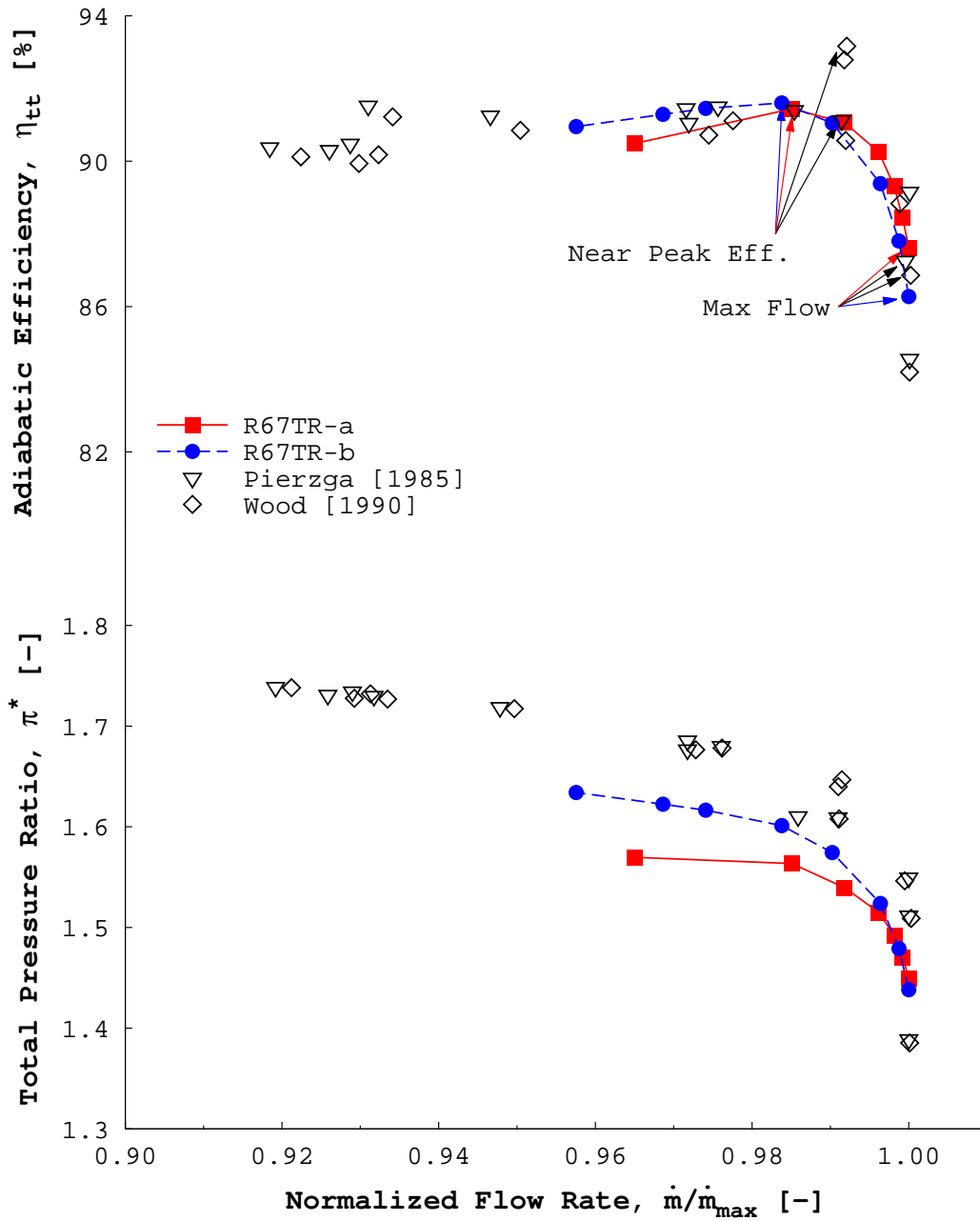


Figure VII.8: Predicted Rotor 67 performance for two turbulent rubbing configurations compared against experiment [Pierzga & Wood, 1985; Wood et al., 1990].

than R67TR-a. At “near peak efficiency” R67TR-b was 0.43% and 2.8% less than the values reported by Pierzga & Wood [1985] and Wood et al. [1990], respectively. The differences for case R67TR-a were much higher with the predicted pressure ranging between 7-10% lower than the measured values. The differences between the predicted and measured pressure ratio increased as the back pressure increased for case R67TR-a.

Table VII.3: Comparison between predicted and measured rotor performance at “near peak efficiency” for two turbulent rubbing Rotor 67 cases.

Data Set	\dot{m}/\dot{m}_{max}	π^*	η_{tt}
R67TR-a	0.985	1.492	0.9144
R67TR-b	0.984	1.601	0.9160
Pierzga & Wood [1985]	0.991	1.608	0.9111
Wood et al. [1990]	0.992	1.647	0.9316

Predicted values of static pressure on the hub endwall are shown plotted against measured values at the downstream measurement location in Fig. VII.9. The trends shown in Fig. VII.9 are qualitatively similar to those seen in the plot of total pressure ratio in Fig. VII.8. Due to the rotation of the endwalls, much lower static pressures were predicted with case R67TR-a than in the experiment. Case R67TR-b was a better quantitative match with the experimental data. For case R67TR-b, at “near peak efficiency”, the predicted normalized hub pressure was $p^{(h)}/p_{0_1} = 1.007$, which was within 0.5% of the experimental data. The predicted pressure at the hub for

case R67TR-a was 2% lower than the measured values.

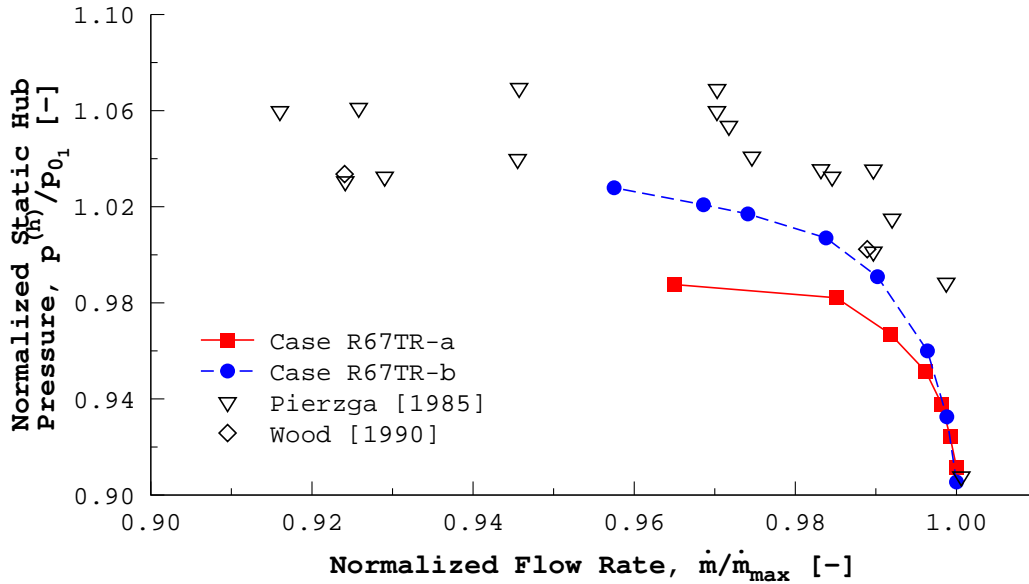


Figure VII.9: Comparison of predicted and measured [Pierzga & Wood, 1985; Wood et al., 1990] downstream hub static pressures for each operating point. Predicted pressures are at the downstream measurement location.

At “near peak efficiency”, the predicted spanwise profiles of static pressure at the upstream and downstream measurement locations qualitatively agreed with the experimental data, as shown in Fig. VII.10. Both grid configurations resulted in nearly identical upstream pressure profiles. Downstream of the rotor, the static pressure was under-predicted by case R67TR-a, in which all of the walls rotated with speed N_{eq} . On the other hand, case r67TR-b was a better quantitative match to the measured data. The downstream profile for R67TR-a was seen to “bend” slightly away from the measured data near the tip. This is due to the influence

of the wall’s rotation on the radial equilibrium of pressure on the outlet, as it is a function of the tangential velocity. For R67TR-a, the tangential velocity was higher than what would be expected due to entrainment by the endwalls.

Figure VII.10 shows that for the majority of the span, the predicted absolute flow angles were an excellent qualitative match to the measured data. The major deviation from measurement occurred with case R67TR-a near the endwalls. This was not unexpected due to the expected influence of the rotating endwalls on the near wall flow. The plot of absolute flow angle in Fig. VII.10 illustrates very well the reason for the lower R67TR-a mass flow rates. The high amount of turning over a much larger region of the span shown in the figure near the end walls for R67TR-a illustrates the axial velocity deficit near the endwalls that was previously discussed.

VII.2.4. Relative Mach Number Contours

In this section, contours of relative Mach number for both rubbing cases are compared against the measured flow at the “max flow” and “near peak efficiency” operating points. Only a single spanwise location, 70% span, is presented for brevity. Flow contours at additional spanwise locations for each operating point can be found in Appendix A, starting on p. 221.

For the “max flow” operating point, the predicted relative Mach number contours for both cases, shown in Fig. VII.11, qualitatively agreed with the measured flow contours. The relative Mach number just upstream of the rotor leading edge and at the inlet of the rotor passage was slightly under-predicted in both cases by approximately $M_{rel} = 0.05$. The passage shock was located in a more realistic posi-

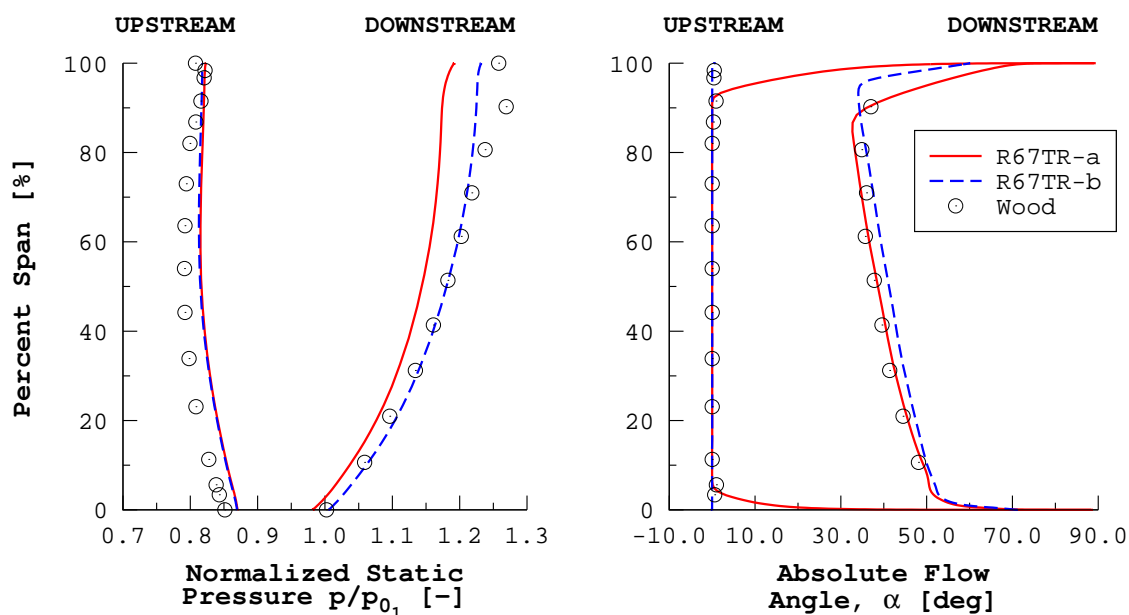


Figure VII.10: Comparing spanwise profiles of predicted static pressure and absolute flow angle against measurement [Wood et al., 1990] for both rubbing Rotor 67 configurations at “near peak efficiency”.

tion than what was predicted by the inviscid simulation of the “max flow” operating point. However, the passage shock still appears to be at bit further downstream than what was observed in the experiment.

Very few noticeable differences were found between the two predicted flow fields at “max flow”. The main difference between the two solutions occurred within the passage between rotors, upstream of the shock. The flow for R67TR-b maintained more of its speed through the passage near the suction side boundary layer, indicated by a larger region encompassed by the $M_{rel} = 1.25$ contour level.

At the “near peak efficiency” point, the predicted flow contours were the best qualitative match to experiment in the region upstream of the rotor leading edge.

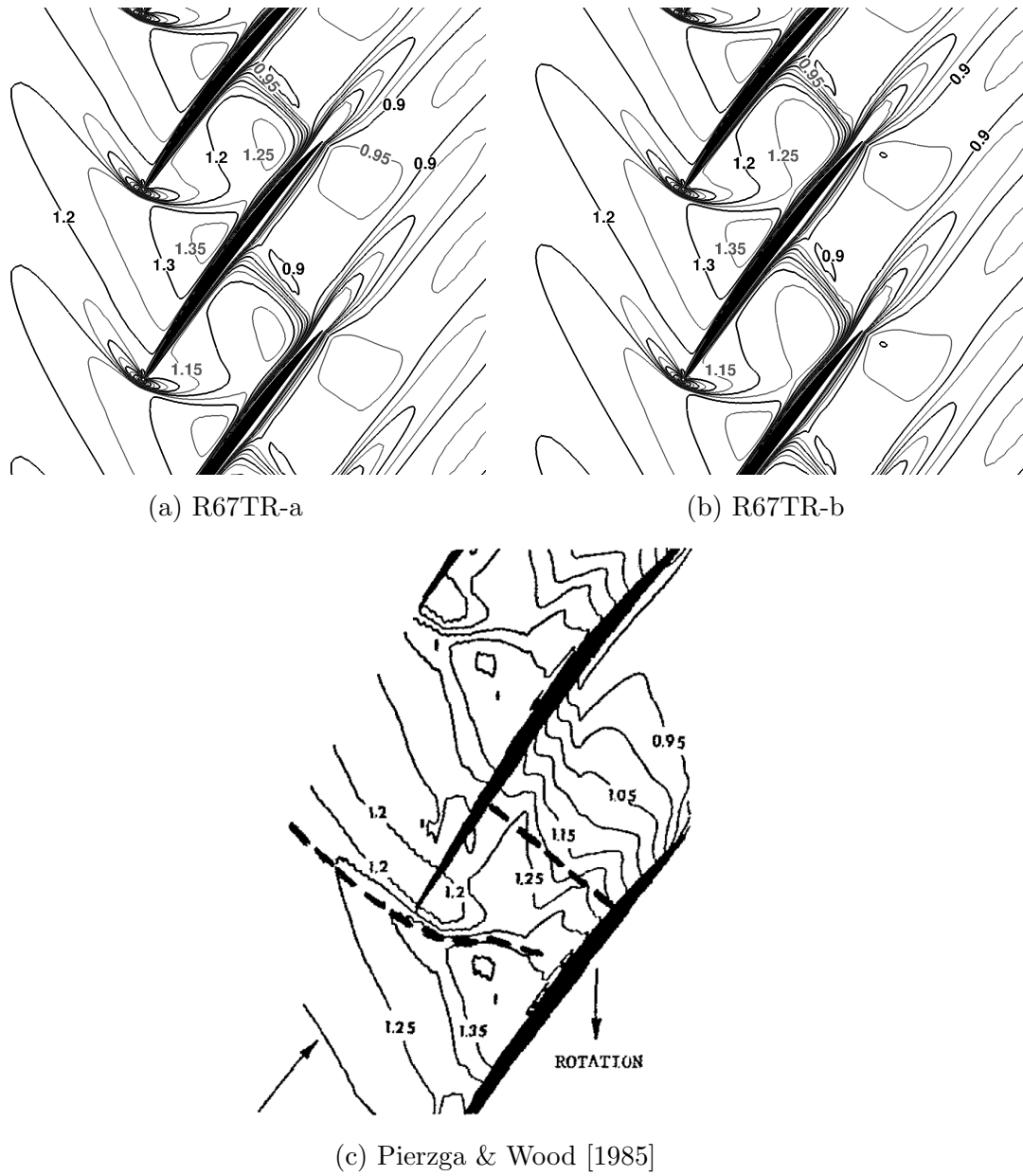


Figure VII.11: Predicted and measured [Pierzga & Wood, 1985] relative Mach number contours ($\Delta M_{rel} = 0.05$) at 70% span for the "max flow" operating condition.

Figure VII.12 compares the predicted flow fields for each rubbing case with the measured flow. As with the “max flow” point, the flow appeared slightly slower upstream of the rotor.

The biggest difference between both predicted flows and the experiment occurred at the inlet of the blade passage. Both predicted flow fields had decelerated flow at the passage inlet. Whereas in the experiment, the passage shock decelerated the flow at approximately 25% of the chordwise distance into the passage, as shown by the short, bold, dashed line in Fig. VII.12c. In both simulations, the sonic line was also much closer to the passage inlet than what was observed in experiment. By visual comparison, the predicted “near peak efficiency” flow contours more closely resembled the measured “near stall” contours, shown in Fig. VII.12d.

By comparing the two predicted flow fields, it was observed that the flow was decelerated more as it passed through the blade passage in case R67TR-b. Figures VII.12a–b reaffirmed the higher static pressure rise that was shown in Fig. VII.10. The sonic line in R67TR-b was also found further downstream in the passage than in R67TR-a, in which the sonic line sat roughly at the passage inlet. The upstream flow field for both cases were nearly identical.

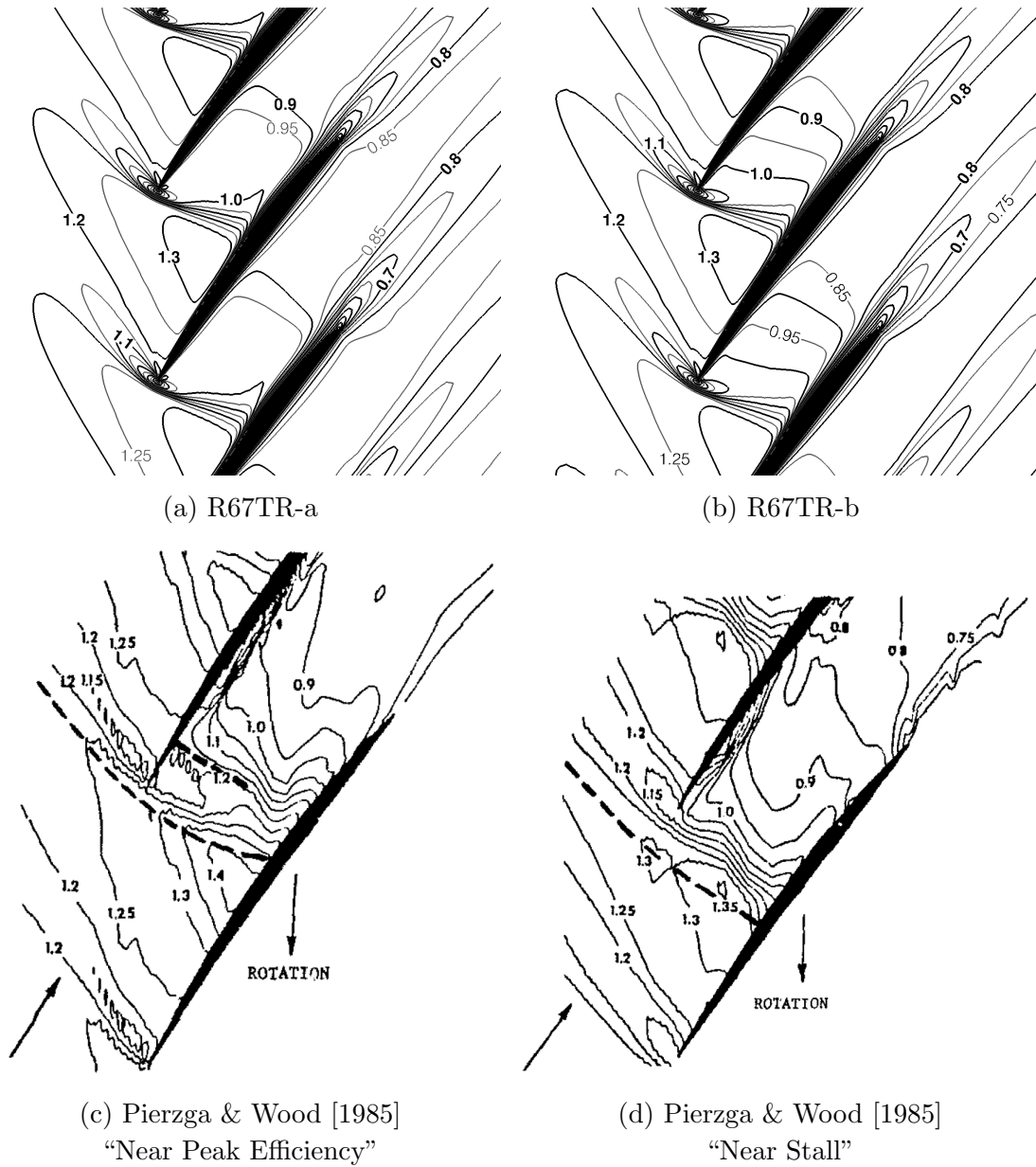


Figure VII.12: Predicted and measured [Pierzga & Wood, 1985] relative Mach number contours ($\Delta M_{rel} = 0.05$) at 70% span for the “near peak efficiency” operating condition.

VII.3. Tip Leakage Flow Configuration

This section presents the analyses of the realistic Rotor 67 configuration in which a clearance gap exists between the rotor tip and endwall. First, the computational grid and boundary conditions are presented. Following this, the primary speed line analyses of the rotor are discussed, and the flow specific operating points are examined. The effect of the choice of higher order limiter is then finally explored.

VII.3.1. Computational Grid

This section details the computational grid used for the numerical studies found herein. The size and dimensions of the blade-to-blade and tip clearance grids are presented first. The domain decomposition done in preparation of parallel simulations is then presented. The grid generator input file for the current grid can be found in Appendix E, starting on p. 283.

The computational grid used in Section VII.2 served as the basis for the current grid. The blade-to-blade grid blocks for the current grid shared all of the grid dimensions of its predecessor except the number of spanwise grid layers. The number of spanwise layers was increased from 135 to 175 layers to account for the inclusion of the tip clearance grids. Refer back to Section VII.2.1 for any details on the blade-to-blade grids.

The tip clearance region was discretized using an O-H grid configuration, as shown in Fig. VII.13. The clearance gap was defined as 1 mm [Wood et al., 1990]. Forty spanwise grid layers were used to define the tip clearance region, leaving 135 spanwise layers to define the rotor. The viscous spacing in the direction normal to

the rotor tip was $z_1^{(r)} = 1.22 \times 10^{-4}$ cm. The viscous spacing at the endwall remained unchanged from the old mesh with a value of $z_1^{(t)} = 9.144 \times 10^{-5}$ cm.

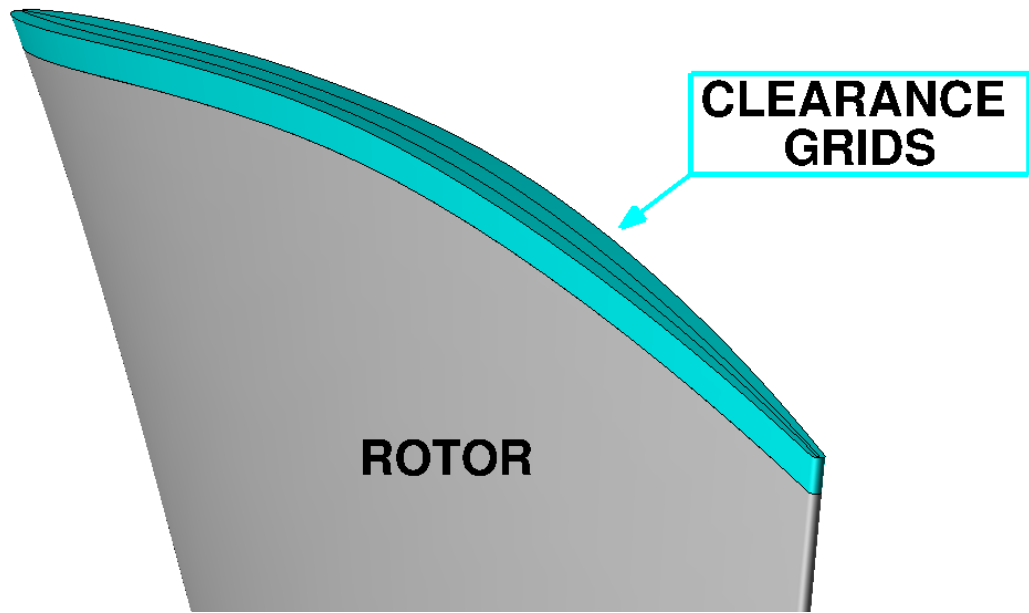
The clearance O-grid had an **IMAX** of 167 and was determined by the dimension of the blade passage O-grid. The **JMAX** dimension of the clearance O-grid was 25, and the **JMAX** dimension of the clearance H-grid was 9. The **IMAX** dimension for the clearance H-grid was computed as a function of the previous three dimensions. The resultant set of clearance grids had 191,000 unique nodes. The size of the complete turbulent Rotor 67 grid was 3,387,830 unique nodes.

The computational grid was split into forty grid blocks for parallel computations in the same manner detailed at the end of Section VII.2.1. The grid was split first five times along the spanwise direction. Then each of the five resultant grid blocks were split twice in the periodic direction. Finally, each grid blocks were split four times in the axial direction. On average, each of the final grid blocks contained approximately 101,000 nodes, or roughly 3% of the original grid nodes.

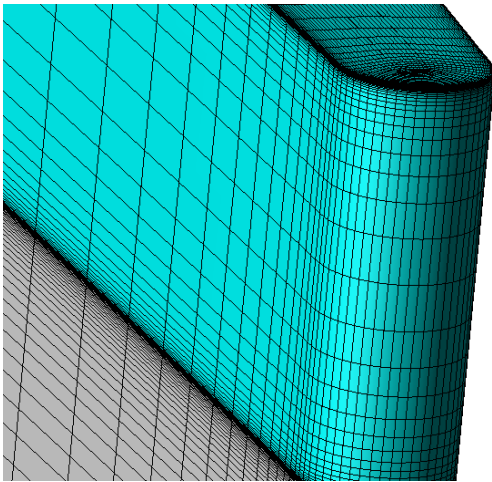
VII.3.2. Boundary Conditions

This section presents the application of the boundary conditions to the computational grid and the specific values used with the flow solver. Rotating viscous wall boundaries were applied to the hub endwall and rotor surfaces. The tip endwall was defined as a fixed viscous wall boundary. Inlet and outlet boundaries were applied in the same manner as with the rubbing cases.

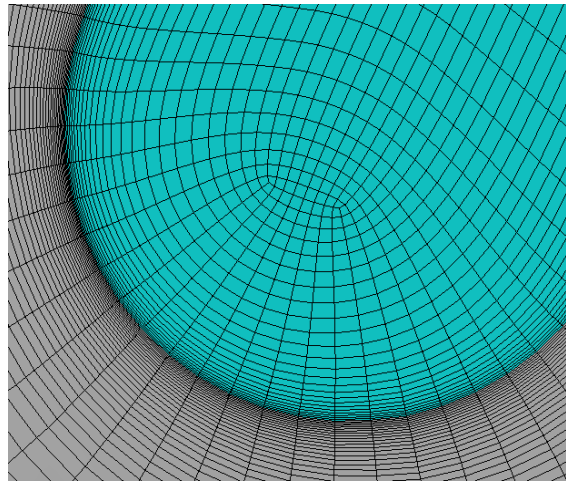
The freestream, inlet, and outlet boundary conditions are summarized in Table VII.4. The same freestream and inlet total conditions that were used for the



(a) General Configuration



(b) Spanwise Layer Distribution



(c) Leading Edge Detail

Figure VII.13: Rotor 67 tip clearance grids.

preceding Rotor 67 cases were once again applied to the current case. The range of outlet pressures used in the current study differed slightly from previous studies with a lower minimum $p_2^{(h)}$ being used.

Table VII.4: Turbulent Rotor 67 boundary conditions.

Freestream Conditions	Inlet Conditions	Outlet Conditions
$p_\infty = 101325.0 \text{ Pa}$	$p_{0_1} = 124452.598 \text{ Pa}$	$0.849 \leq p_2^{(h)}/p_{0_1} \leq 1.02$
$T_\infty = 288.3 \text{ K}$	$T_{0_1} = 305.742 \text{ K}$	
$M_\infty = 0.55 \text{ -}$		
Inlet Turbulence Conditions		
$\ell_T _\infty = 1.04 \text{ mm}$		$I_T _\infty = 3\%$
Equivalent Conditions		
$\delta = 1.2283 \text{ -}$	$\theta = 1.0605 \text{ -}$	$N_{eq} = 16520.0 \text{ RPM}$

VII.3.3. Compressor Map Speed Line Analyses

This section presents the analyses of the compressor map speed line at the design wheel speed for the realistic Rotor 67 configuration. First, the effect of the gradient method on the solution is presented. An addendum to the operating line calculation procedure previously outlined in Section VII.1 is given. Finally, the predicted performance of the rotor is given and discussed.

VII.3.3.1. Effect of Gradient Method

This section presents the effect of the choice of gradient evaluation method on the solution. The initial operating point was used in this analysis because each separate simulation could be run identically with the exception of the gradient method. With exception to the gradient method, the initial operating point was computed using

the procedure outline in Section VII.1.

The initial operating point simulation using the LSQR gradient method exhibited poor convergence trends, as illustrated in Fig. VII.14. As similar poor convergence trends had been observed when using either the Venkatakrisnan or MLP limiter options, the initial operating point was simulated using the other gradient calculation methods available: least-squares (LS) and Green-Gauss (GG). These additional simulations were computed exactly as the LSQR simulation had been, with the only difference in the input being the choice of gradient method.

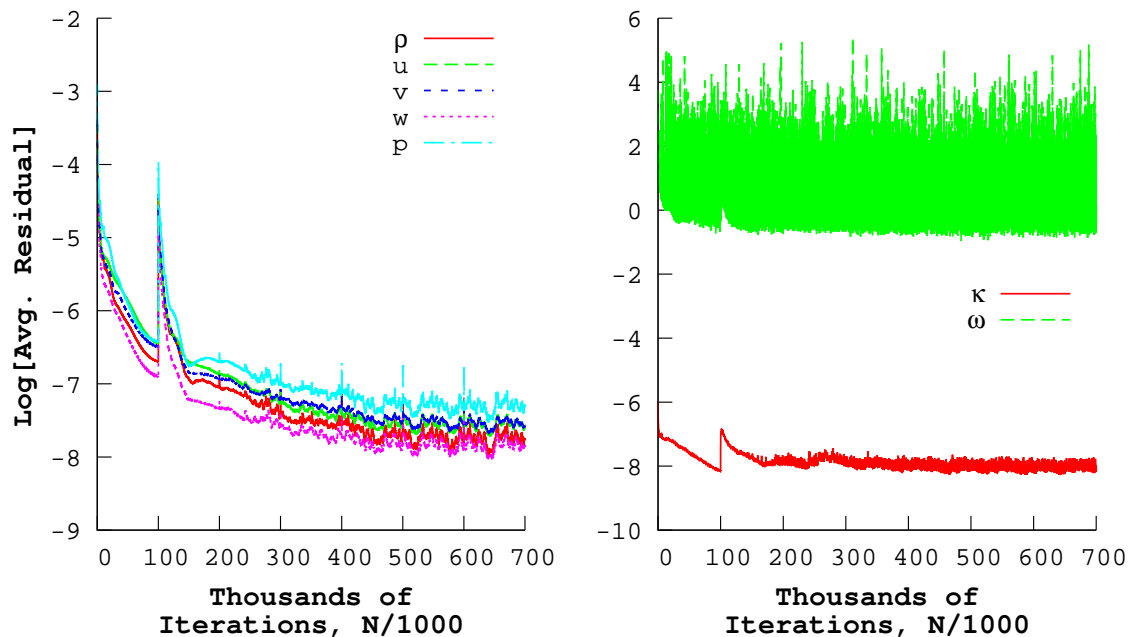


Figure VII.14: Turbulent Rotor 67 initial operating point residual histories for the flow (left) and turbulence (right) equations.

Figure VII.15 compares the density and turbulence residuals from the three separate gradient method simulations. The density residual was only shown because

the other flow residuals shared the same trend as the density residual. The solution residuals showed convergent trends when the gradient method was anything other than LSQR. In addition to the convergent trends, the residual behaviour was more smooth and organized than what was seen with LSQR. This fact was especially true of the ω residual convergence. The little spikes shown with the least-squares and Green-Gauss residual curves were the result of restarting the simulation, and the residuals were seen to quickly drop to their pre-restart values.

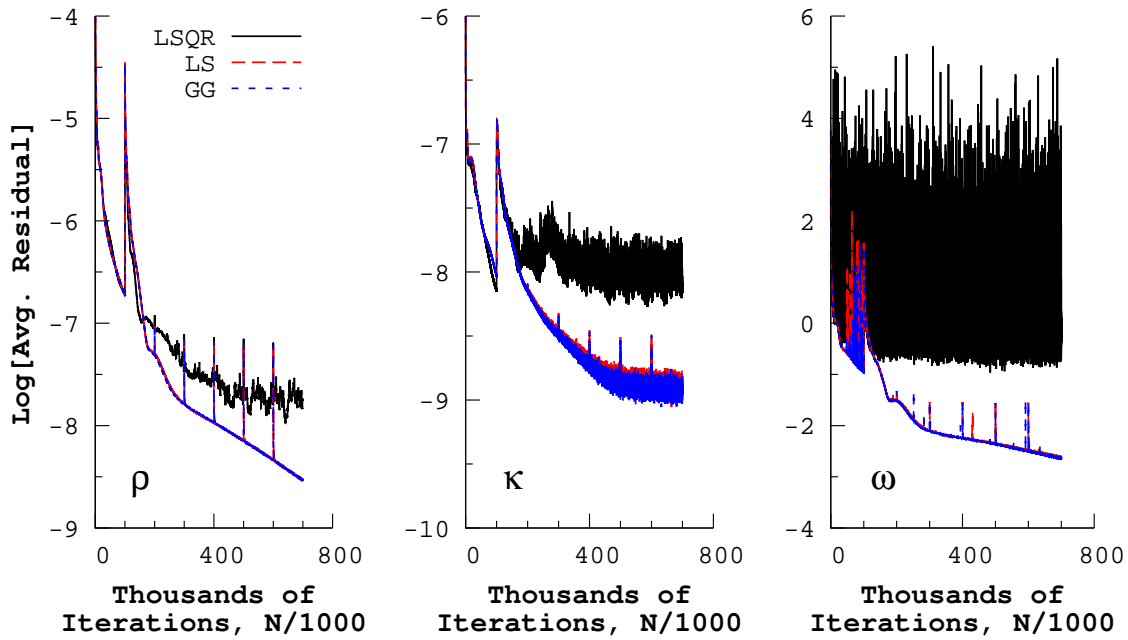


Figure VII.15: Turbulent Rotor 67 initial operating point residual histories for different gradient evaluation methods.

The solution was also found to change appreciably when the gradient evaluation method was changed. Table VII.5 presents the rotor performance from the flow at the measurement planes for each gradient evaluation method. The predicted performance

parameters from the least-squares and Green-Gauss solutions were nearly identical; on the order of a percent of a percent difference in values. By comparison, the LSQR solution under-predicted the mass flow rate, total pressure ratio, and adiabatic efficiency by 1.5%, 1.6%, and 0.1%, respectively.

Table VII.5: Turbulent Rotor 67 initial operating point performance variation with gradient calculation method.

Gradient Method	Mass Flow Rate, \dot{m}_{eq} [kg/s]	Total Pressure Ratio, π^* [-]	Adiabatic Efficiency, η_{tt} [%]
Least-Squares w/QR	33.813	1.517	88.896
Least-Squares	34.320	1.542	88.994
Green-Gauss	34.323	1.542	89.019

From the differences observed with the convergence and the solution, it was expected that there was a coding bug hidden within the LSQR subroutines. Inspection of the relevant subroutines turned up no implementation issues. What was found, however, was that the LSQR formulation that was implemented by Gargoloff [2007] was an un-weighted LSQR formulation [Anderson & Bonhaus, 1994; Haselbacher & Blazek, 2000].

It is known that the un-weighted LSQR method results in the inaccurate approximation of the gradients in regions where the flow varies non-linearly [Anderson & Bonhaus, 1994; Mavriplis, 2003]. Mavriplis [2003] showed that the un-weighted LSQR largely under-predicted the gradients for non-linearly varying functions in re-

gions of the mesh which contained high aspect ratio elements with large amounts of surface curvature. Such regions are common along any curved boundary with viscous spacing.

Large inaccuracies in the gradient evaluation can result in the degradation of the overall solution [Mavriplis, 2003]. The degradation occurs because accurate values of the gradients are needed to compute the viscous fluxes and the source terms of the turbulence equations. Mavriplis [2003] showed that using a LSQR formulation weighted by the inverse distance to the adjacent nodes greatly improved the accuracy of the gradient calculation. The basic least-squares gradient method implemented by Kim [2003], and used above, used the inverse-distance weighting.

VII.3.3.2. Operating Line Generation Procedure Addendum

With exception to the gradient evaluation method, the operating points were computed in the manner described in Section VII.1. Two gradient evaluation methods were used to compute the remaining operating points. The first was the un-weighted LSQR method. The un-weighted method was used because it was not a known issue at the time, and it was only after all of the data had been collected that the gradient evaluation issue came to light. The second gradient evaluation method use was the Green-Gauss method. It was chosen because it was the first gradient method implemented in UNS3D, and as such was perceived to be the most “fleshed out” of the gradient methods. Time restrictions ultimately prevented any additional operating point simulations using the basic least-squares implementation.

The LSQR and Green-Gauss results are labeled in all plots herein as R67TC-

LSQR and R67TC-GG, respectively. The outlet boundary conditions and a summary of the results at each operating point for the LSQR simulations can be found in Table A.4. Table A.5 contains the same information for the Green-Gauss operating line simulations. A sample input file for a LSQR simulation can be found in Appendix D, starting on p. 266.

VII.3.3.3. Rotor Operating Line

The predicted performance characteristics for the rotor, using both un-weighted LSQR and Green-Gauss gradient methods, are presented herein. First, the mass flow rates at the “max flow” operating point are first given. The compressor map speed line, at design wheel speed, is then presented and discussed. Finally, the flow downstream of the rotor at the “near peak efficiency” operating point is examined. The predicted flows are compared against experimental data and against the better of the two previously discussed rubbing configurations, case R67TR-b.

The “max flow” operating point was defined as the operating point for which the lowest outlet static pressure was used to simulate the flow. As a result, the largest predicted mass flow rate occurs at this point. The “near peak efficiency” operating point was defined a little differently for the LSQR and Green-Gauss gradient evaluation cases. For LSQR, the “near peak efficiency” point was chosen as the operating point with the highest adiabatic efficiency. For Green-Gauss, the operating point just to right of the point at which maximum adiabatic efficiency occurred was chosen. The reason for the difference in definition was because the LSQR simulations resulted in an attenuated outlet pressure range, as will be shown later in this section,

that fell below measured values for “near peak efficiency”.

Table VII.6 gives the predicted maximum mass flow rates for the two clearance flow cases. Also given are the measured maximum flow rates and the predicted value from case R67TR-b for comparison. The maximum flow rate predicted with LSQR was 0.8% smaller than the flow rate predicted when Green-Gauss was used, and was 2.1% less than the average measured value. The difference between the Green-Gauss result and the averaged measured value was 1.33%, which was comparable with the value obtained by Arnone [1994].

Table VII.6: Predicted and measured [Pierzga & Wood, 1985; Wood et al., 1990] equivalent maximum mass flow rates (kg/s) for turbulent Rotor 67.

Tip Clearance Flow		Rubbing Flow [†]	Pierzga & Wood	Wood et al.
LSQR	Green-Gauss			
34.215	34.481	34.215	34.92	34.96

[†] Rubbing configuration R67TR-b

The compressor map speed lines for the two clearance flow cases can be seen in Fig. VII.16. The mass flow rates for each speed line have been normalized by their respective maximum. The predicted curves of total pressure ratio and adiabatic efficiency for both clearance flow cases showed qualitative agreement with the measured curves. The curves generated using Green-Gauss gradient evaluation were a much better quantitative match to measurement than the LSQR result. The Green-Gauss result was also a very close match to the values predicted by the rubbing configura-

tion, which used un-weighted LSQR gradient evaluation.

It was found that the operating ranges, in terms of normalized mass flow rate, were smaller than that predicted by the rubbing configuration. This was true for both Green-Gauss and LSQR. The range predicted by the Green-Gauss speed line was slightly smaller than that predicted by LSQR. However, this result should be taken with a grain of salt as no effort was undertaken to find the true stall point due to time constraints. The simulation of additional operating points at smaller $\Delta p_2^{(h)}$ increments should extend the range of both speed lines further to the left of the plot.

Figure VII.17 compares the predicted hub endwall static pressure at the downstream measurement plane against experimental data for both gradient method solutions. Using LSQR resulted in static pressures that were smaller overall than when using Green-Gauss. The static pressures predicted with Green-Gauss were a better quantitative match to experiment, as shown in Fig. VII.17. The Green-Gauss results also matched the predicted hub pressures from the best rubbing case (R67TR-b).

Spanwise profiles of static pressure and absolute flow angle were constructed by circumferentially averaging the flow at both measurement planes. The resultant spanwise profiles for both gradient method solutions at “near peak efficiency” are presented in Figure VII.18, where they are compared against measurement and the rubbing configuration, case R67TR-b. Both clearance flow solutions showed excellent qualitative agreement with the measured profiles, for both pressure and flow angle. Quantitatively, the static pressure profiles from the Green-Gauss gradient evaluation case were a closer match to measurement than LSQR.

The only discernible difference in the flow angle occurred downstream near the

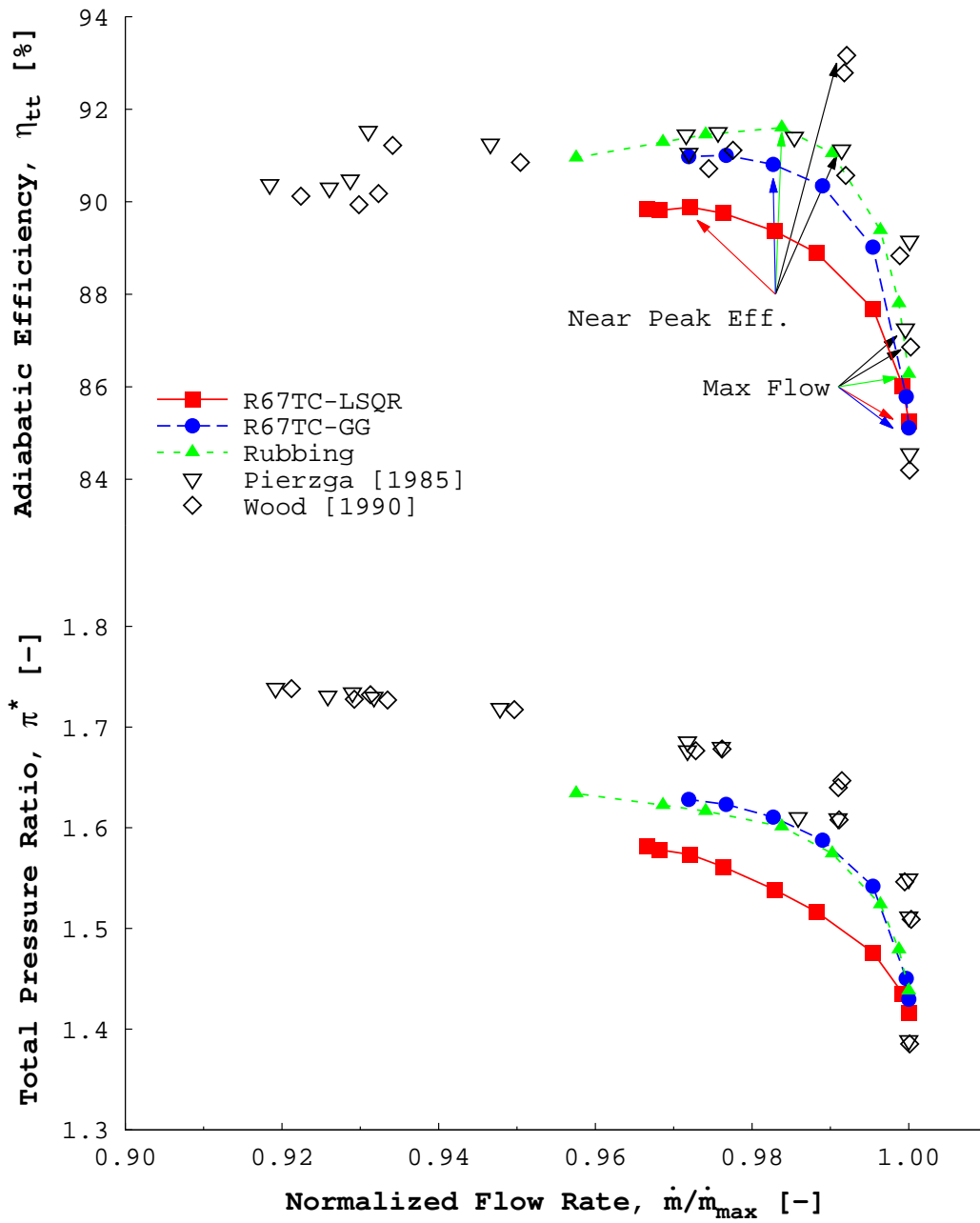


Figure VII.16: Predicted Rotor 67 performance for two turbulent clearance flow cases compared against experiment [Pierzga & Wood, 1985; Wood et al., 1990] and a turbulent rubbing configuration (R67TR-b).

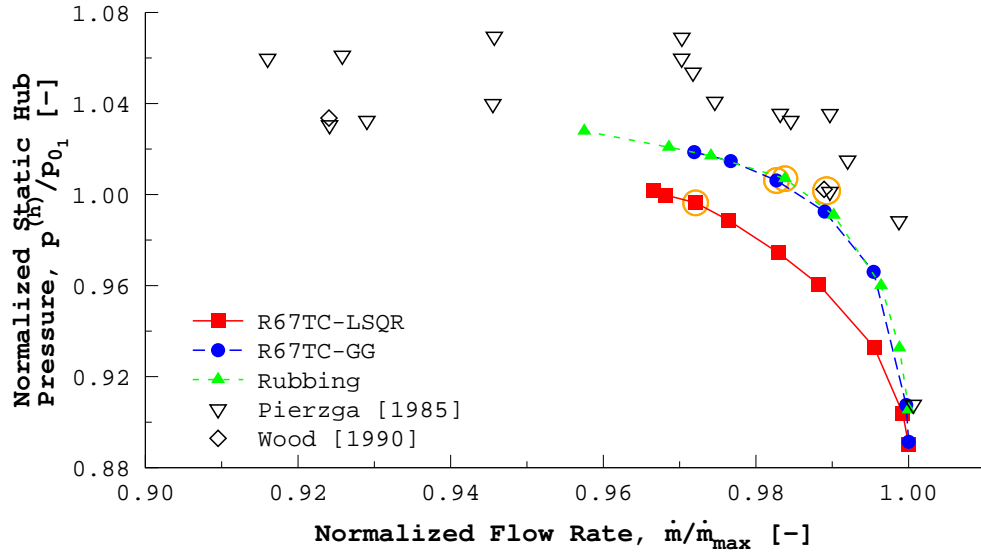


Figure VII.17: Normalized hub static pressures measured at the downstream measurement location. Circled data points are at “near peak efficiency”.

rotor tip. The flow angles predicted when using Green-Gauss were closer to the measured values than LSQR. Gradients play a much larger role in this region of the flow due to the tip leakage flow and the vortices that are generated as a result. It is not a surprise then that Green-Gauss outperformed LSQR in this region.

VII.3.4. Relative Mach Contours

Contours of the relative Mach number at a given spanwise station for the “max flow” and “near peak efficiency” operating points are presented for both clearance flow cases. The predicted flows at 70% span are compared against predicted contours from the best rubbing configuration, case R67TR-b, and against the measured flow. Relative Mach number contours at 30% and 90% span can be found in Appendix A,

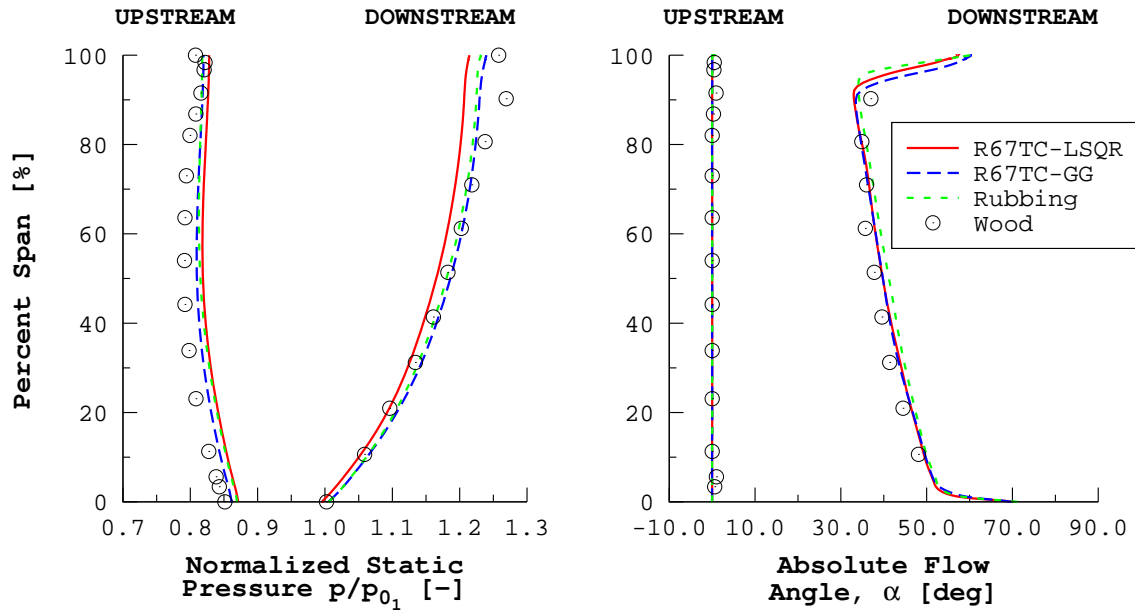


Figure VII.18: Comparing spanwise profiles of predicted Rotor 67 static pressure and absolute flow angle against measurement [Wood et al., 1990] at “near peak efficiency”.

starting on p. 226.

Figure VII.19 shows the predicted flow fields at the “max flow” operating point, and compares them against the previous rubbing result and experiment. The two predicted flow fields were qualitatively similar to the measured flow field. The biggest difference between experiment and prediction was the location of the passage shock. Both predicted flow fields showed the passage shock sitting just upstream of the trailing edge, whereas in the experiment, the shock sat further upstream towards mid-chord.

The primary difference between the un-weighted LSQR and Green-Gauss gradient evaluation cases was found in the passage between rotor blades. The passage

shock predicted by the Green-Gauss simulation was more compact and stronger than what was found in the LSQR solution. This difference was attributed to the LSQR gradients not being computed accurately in the region of the shock.

In comparing both clearance flow solutions to the rubbing flow solution it was observed that the LSQR clearance flow best matched the rubbing flow. This was to be expected as both solutions were computed using un-weighted LSQR. Both clearance flows exhibited less separation along the aft portion of the rotor, as shown by Figs. VII.19a–c.

At “near peak efficiency” it was observed that both the LSQR and green-Gauss solutions were a good qualitative match to the measured flow field, as shown in Fig. VII.20abd. The Green-Gauss solution was the best match to the measured flow field. Again, the Green-Gauss solution resulted in a more compact passage shock, whereas the was more diffuse with LSQR. The relative Mach number contours at the additional spanwise locations given in Appendix A showed that Green-Gauss gradient evaluation was superior to the un-weighted LSQR in predicting the flow field across the rotor’s span at “near peak efficiency”.

The two clearance flow solutions were also qualitatively similar to the rubbing configuration solution, as shown in Figs. VII.20a–c. The LSQR predicted passage flow was a better qualitative match to the rubbing case than the Green-Gauss solution. In both the LSQR and rubbing cases, the flow was decelerated less through the passage than what was predicted by Green-Gauss. The amount of separation that was apparent along the aft portion the rotor was less for the Green-Gauss case than the LSQR and rubbing cases.

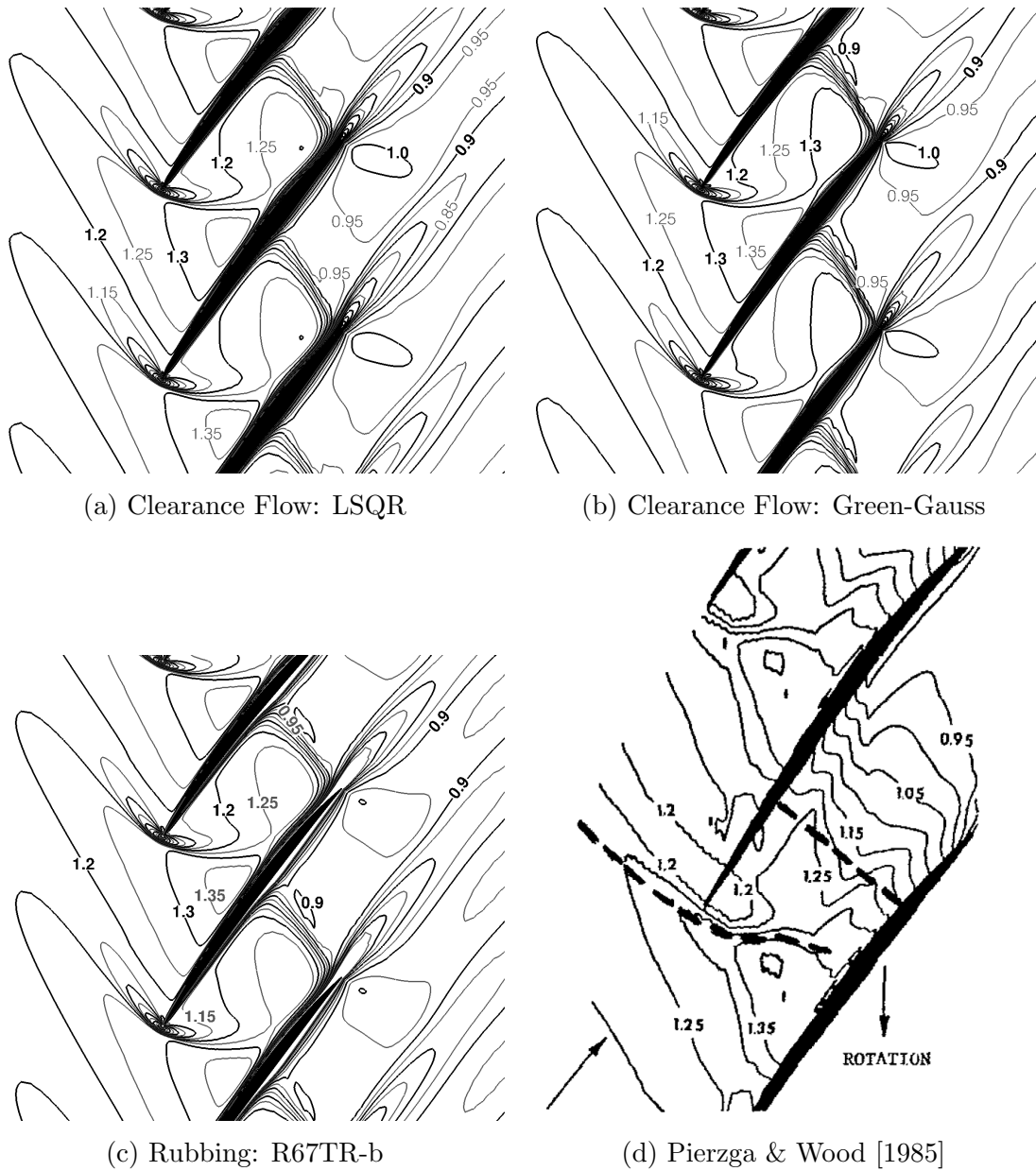


Figure VII.19: Predicted and measured [Pierzga & Wood, 1985] relative Mach number contours ($\Delta M_{rel} = 0.05$) at 70% span for the “max flow” operating condition.

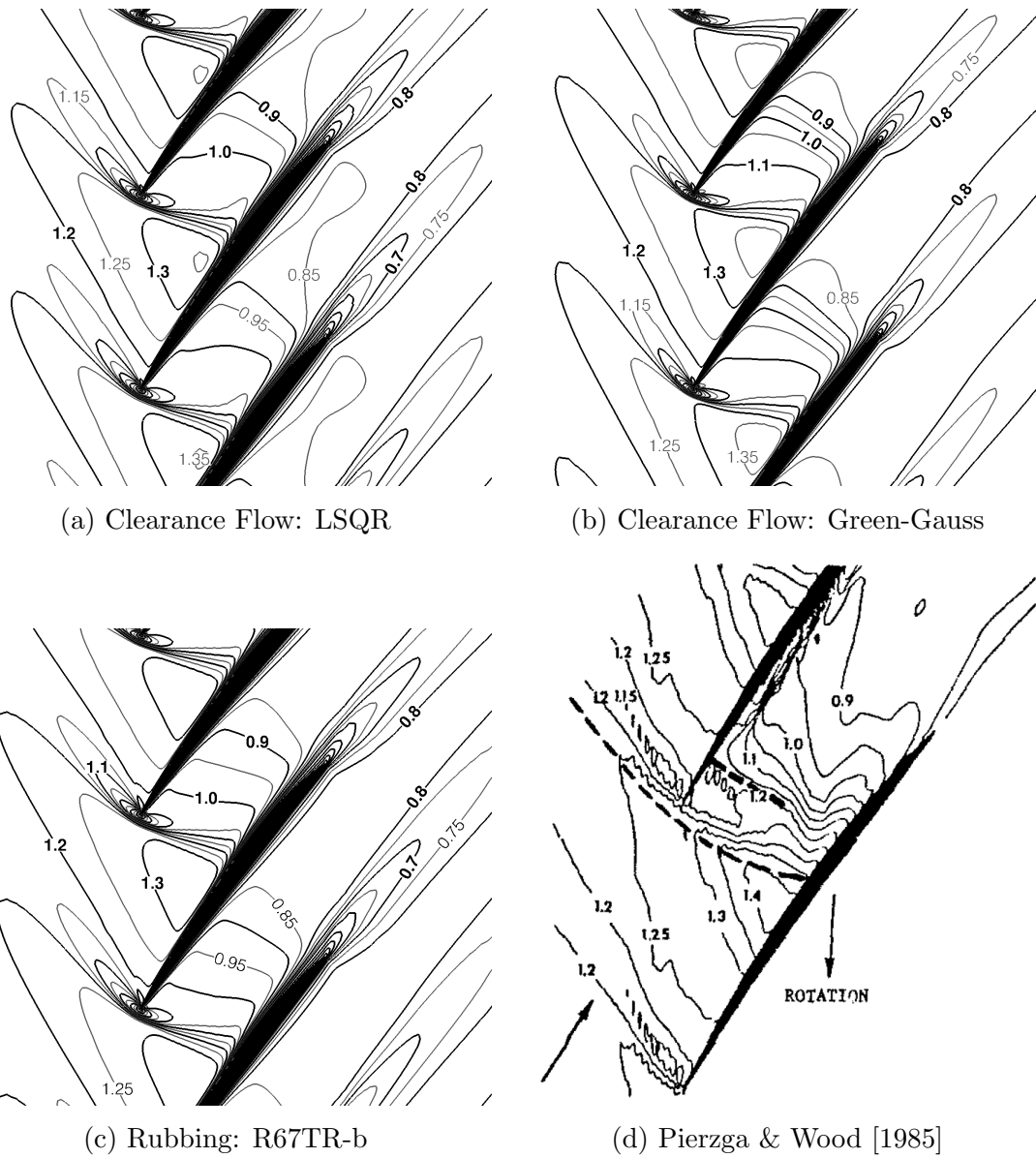


Figure VII.20: Predicted and measured [Pierzga & Wood, 1985] relative Mach number contours ($\Delta M_{rel} = 0.05$) at 70% span for the “near peak efficiency” operating condition.

VII.3.5. *Effect of Higher-Order Limiters*

This section examines the effect of the higher-order limiter on the predicted performance of the rotor at the initial operating point. The additional limiters examined were the Venka- ϵ_1 , Venka- ϵ_2 , and MLP- ϵ_2 limiters. The MLP- ϵ_1 limiter was not tested because the inviscid test of the limiters revealed little difference between Venka- ϵ_1 and MLP- ϵ_1 .

The initial operating point simulations for the additional limiters were run in the same manner as was described in Section VII.3.3.1. The un-weighted LSQR gradient method was used exclusively for the limiter tests as they were completed before the gradient method survey in Section VII.3.3.1 was completed. The Dervieux solution, computed using Green-Gauss gradients, is used herein as the reference solution because it was the closest overall match to the measured rotor.

Figure VII.21 shows the solution convergence for the different limiter simulations. Only the density residual was shown as the convergence trends of the other flow variables were qualitatively similar. Both Venkatakrishnan limiters and the MLP limiter, MLP- ϵ_2 , showed poor convergence trends for every variable. Severe residual convergence stall was observed, limiting the order of magnitude reduction for the residuals to less than one for the Venkatakrishnan limiters and about one for MLP- ϵ_2 .

In addition to poor convergence of the variable residuals, both Venkatakrishnan limiters exhibited poor convergence of the integral properties of interest. This was especially true of the adiabatic efficiency, as shown in Fig. VII.22. The MLP- ϵ_2 and Dervieux limiter solutions did not exhibit this poor overall convergence behavior.

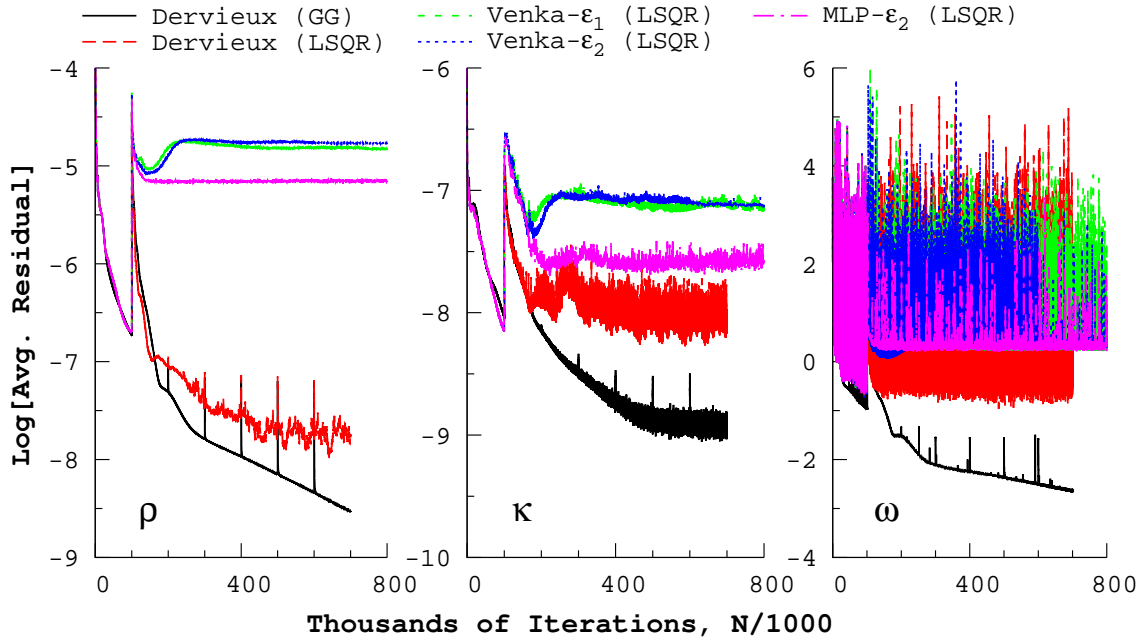


Figure VII.21: Comparison of the turbulent Rotor 67 residual convergence histories with higher-order limiter choice at the initial operating point.

Average values of the rotor performance characteristics were computed due to the lack of convergence observed with both Venkatakrishnan limiters. Error bars were computed for each average using the standard error of the values used. Table VII.7 presents both the average and the error bars associated with each performance variable for the limiters tested. The last five data points were used to average both Venkatakrishnan data sets, and three were used for the other cases.

Both Venkatakrishnan limiters and MLP- ϵ_2 resulted in lower average values for mass flow rate and total pressure ratio than the Dervieux limiter solution. This was true with both Green-Gauss and LSQR. The difference between the limiters tested in this section and the Green-Gauss/Dervieux solution was 2% for the mass flow

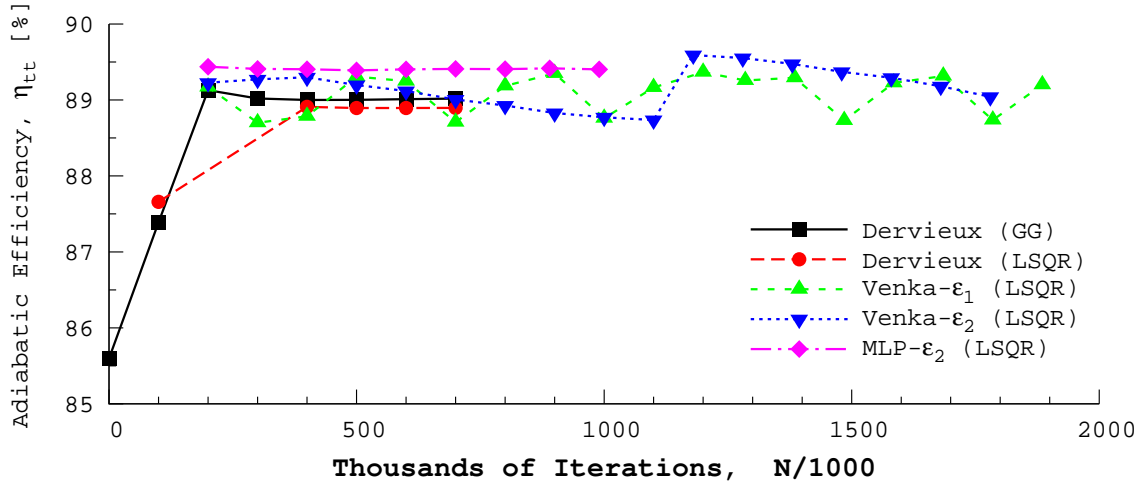


Figure VII.22: Comparison of the turbulent Rotor 67 adiabatic efficiency convergence histories with higher-order limiter choice at the initial operating point.

rates and 4% for total pressure ratio. The computed average values for adiabatic efficiency were within 0.5% of the reference solution, despite a lack of convergence by the Venkatakrishnan solutions.

The spanwise profiles of static pressure and absolute flow angle at both measurement planes for both Venkatakrishnan limiters and MLP- ϵ_2 were found to be qualitatively similar to the reference solution, as shown in Fig. VII.23. The spanwise variation of the flow upstream of the rotors were nearly identical for the limiters tested here and with the LSQR/Dervieux solution. The downstream profiles for both Venkatakrishnan limiters and MLP- ϵ_2 were quantitatively similar to one another; however, they resulted in under-predicted values of both static pressure and flow angle. The most noticeable feature observed with the limiters examined in this section was an inflection point in the downstream profile of absolute flow angle at around 5% span from the hub.

Table VII.7: Comparing averaged turbulent Rotor 67 performance characteristic variations with higher-order limiter choice at the initial operating point.

Higher-Order Limiter	Mass Flow Rate, \dot{m}_{eq} [kg/s]	Total Pressure Ratio, π^* [-]	Adiabatic Efficiency, η_{tt} [%]
Dervieux [†]	$34.3221 \pm 6 \times 10^{-4}$	$1.5417 \pm 2 \times 10^{-4}$	$89.01 \pm 4 \times 10^{-3}$
Dervieux [‡]	$33.8134 \pm 6 \times 10^{-4}$	$1.5163 \pm 9 \times 10^{-5}$	$88.89 \pm 4 \times 10^{-4}$
Venka- ϵ_1 [‡]	$33.6746 \pm 7 \times 10^{-4}$	$1.4820 \pm 2 \times 10^{-4}$	$89.04 \pm 1 \times 10^{-1}$
Venka- ϵ_2 [‡]	$33.6744 \pm 7 \times 10^{-4}$	$1.4819 \pm 1 \times 10^{-4}$	$89.27 \pm 8 \times 10^{-2}$
MLP- ϵ_2 [‡]	$33.6816 \pm 8 \times 10^{-4}$	$1.4838 \pm 4 \times 10^{-5}$	$89.41 \pm 3 \times 10^{-3}$

[†] Gradients evaluated using Green-Gauss [‡] Gradients evaluated using un-weighted LSQR

The inflection point in the spanwise distribution of the downstream flow angle was the result of a large separated region of flow on the last third of the rotor's suction side. The large separated region, shown in Fig. VII.24a, resulted in a vortex which traveled up along the rotor's trailing edge until it lifted off and entered the wake flow. Both the Venka- ϵ_2 and MLP- ϵ_2 limiters resulted in a qualitatively similar flow field as shown in Fig. VII.24a for Venka- ϵ_1 .

The Dervieux solution, Fig. VII.24b, resulted in a compact region of separation at the trailing edge near the hub, resulting in lower flow deviation. The choice of gradient evaluation did not result in a qualitatively different flow field as the one shown in Fig. VII.24b. The trailing edge separation region near the hub has reported as being observed in numerical simulations [Chima, 1991; Jennions & Turner, 1993; Arima et al., 1999]. Chima [1991] also predicted the spanwise traveling trailing edge

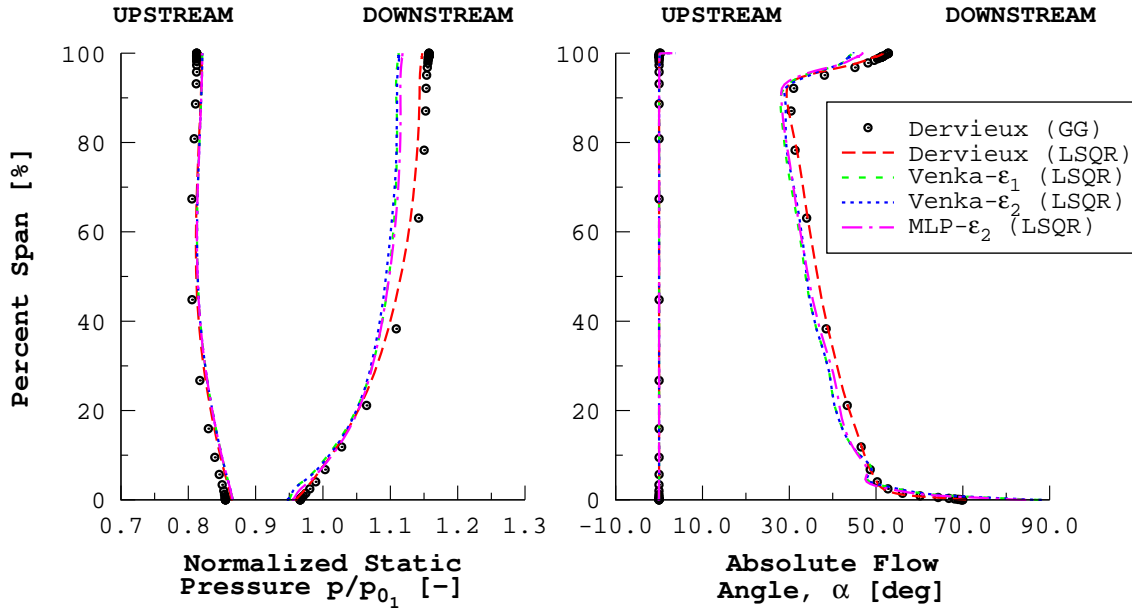
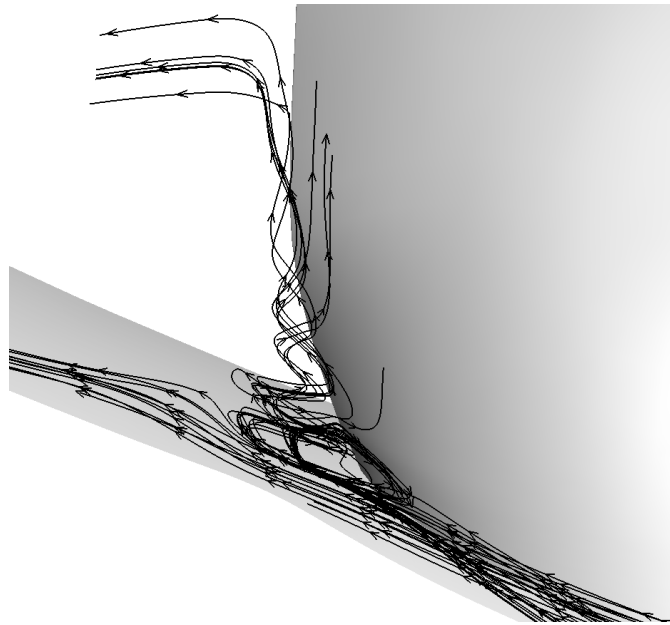


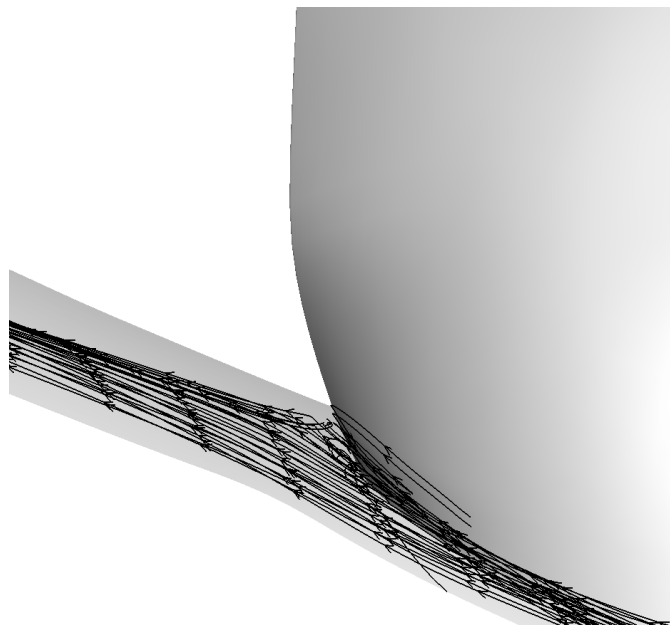
Figure VII.23: Comparing spanwise profiles of static pressure and absolute flow angle, at both measurement planes, for turbulent Rotor 67 with different higher-order limiters. Solution is at the initial operating point.

vortex, however the predicted separation region was not as immense as what was observed for Venka- ϵ_1 , Venka- ϵ_2 , and MLP- ϵ_2 solutions.

Examining contours of relative Mach number at 70% span showed that the predicted Venka- ϵ_1 , Venka- ϵ_2 , and MLP- ϵ_2 flow structures were qualitatively identical. All of the Venkatakrishnan based limiter solutions exhibited a larger separated region on the suction side of the rotor than either of the Dervieux solutions, as shown in Fig. VII.25. Although not shown, the Venkatakrishnan based limiter solutions showed more separation along the entire span, similar to what is shown in Fig. VII.25. The higher amount of separation along the span accounts for the lower downstream pressure and absolute flow angle that was observed with these cases.



(a) Venka- ϵ_1 (LSQR)



(b) Dervieux (Green-Gauss)

Figure VII.24: View looking downstream of relative velocity streamlines which show the separated region near the trailing edge and hub for Rotor 67 at the initial operating point.

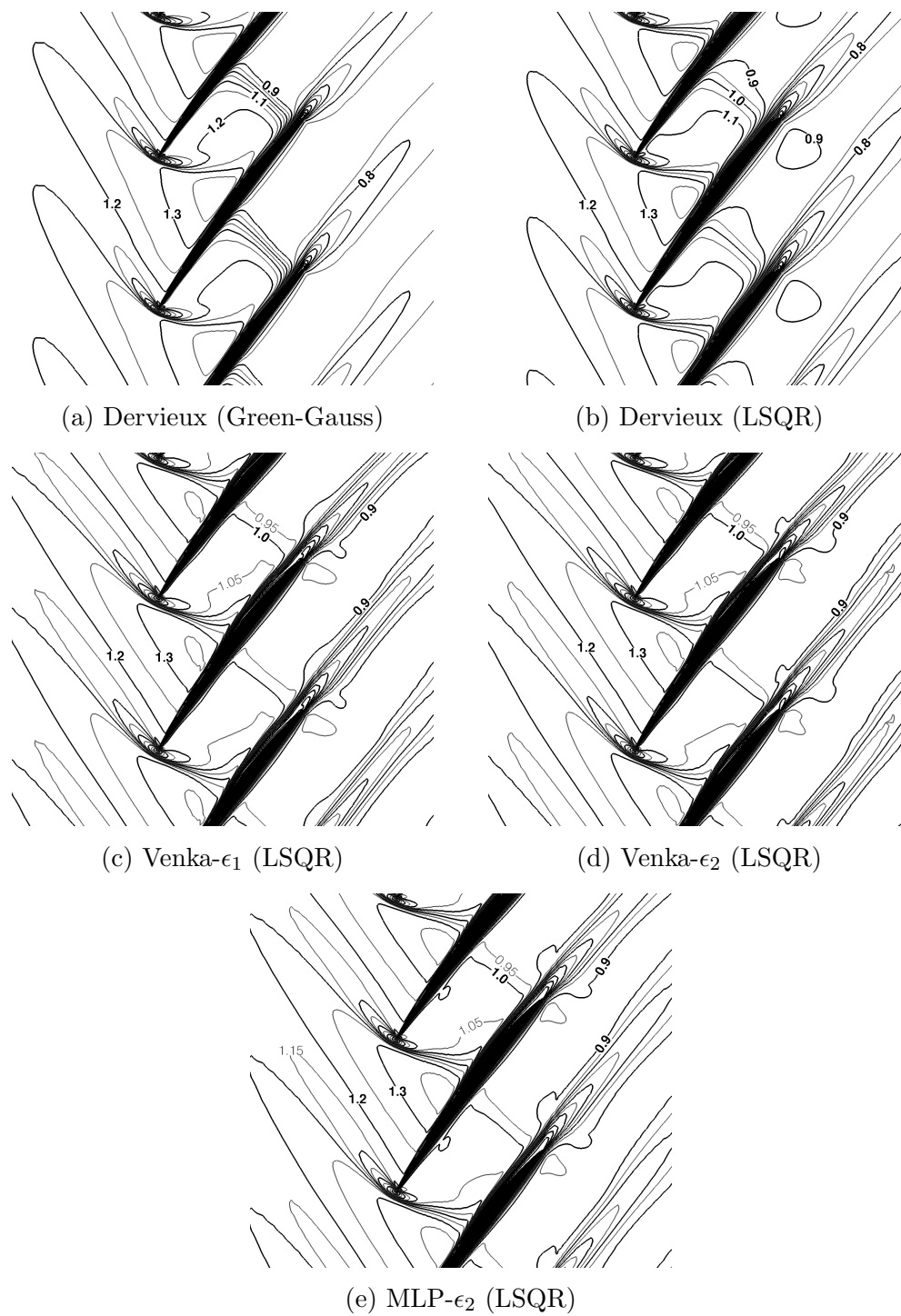


Figure VII.25: Predicted initial operating point turbulent relative Mach number contours ($\Delta M_{rel} = 0.05$) comparing higher-order limiters at 70% span.

CHAPTER VIII

CONCLUSIONS

This chapter presents the conclusions taken from the numerical results presented in the preceding chapters. This dissertation offered a new Venkatakrishnan solution limiter based upon the ϵ^2 definition by Park & Kim [2012]. In addition, the MLP limiter was modified to utilize the newer ϵ^2 definition. The usefulness of the new and modified limiters were evaluated using both canonical and turbomachinery test cases. Enhancements to the turbulence model and outlet boundary condition were also completed as a part of this work. The following conclusions were drawn from the numerical simulations found in this work:

1. The new Venkatakrishnan limiter was found to offer improvement over the base version of the limiter for cases in which convergence was disrupted. For cases in which no convergence issues were observed, the new Venkatakrishnan limiter performed in a manner very similar to the base version. Solutions computed using both limiter versions were found to agree quantitatively.
2. The modified MLP limiter also showed improvement over the base version of the limiter in some cases. The modified limiter behaved similarly to the base version when poor solution convergence was not an issue. Solutions computed using both limiter versions were found to agree quantitatively.
3. Application of implicit residual smoothing to the turbulence model equations prevented the convergence of the model variables, κ and ω . The convergence

of the mass, momentum, and energy equations was observed to stall due to the non-convergence of the turbulence model. The cause was found to be inaccurate turbulence boundary conditions that resulted from the smoothing of the boundary nodes. The near-term solution was to avoid smoothing the $\kappa - \omega$ equations.

4. Excessive under-relaxation of the turbulence equations was found to prevent the evolution of the turbulence field. As a result, the quality of the flow solution was degraded, and the convergence of all equations hindered. Values of under-relaxation greater or equal to 1/2 were found to be usable.
5. The turbulent flow through an annular turbine vane row was simulated using the flow solver. The predicted vane surface pressures and the predicted flow in the wake of the vane showed excellent agreement with the experimentally measured flow. The flow solver was also able to capture the salient features of the flow associated with turbomachinery flows. These features included: horseshoe vortices that form at the leading edge of the vane near the endwalls, the passage vortices which are born from the leading edge horseshoe vortices, and the lift-off of said passage vortices.
6. The inviscid prediction of a transonic fan rotor showed good agreement with measured data. As was expected, the inviscid solutions over-predicted the total pressure ratios and the adiabatic efficiencies; however, the predicted mass flow rates were in excellent quantitative agreement with experiment.
7. The turbulent flow through a transonic fan rotor with no tip leakage flow

was computed using the flow solver for two boundary condition configurations. All viscous endwalls were assumed to rotate with the rotor in the first, and only a segment of both endwalls, which bounded the rotor, were assumed to rotate in the second. Both cases showed qualitative agreement with experiment. However, both cases showed serious flaws. The case in which all walls rotated resulted in the largest under-prediction of all performance parameters. The second case suffered from an unexplained numerical issue which resulted in a non-physical point in the flow at all operating points. As such, neither case was found to be useful.

8. The turbulent flow through a transonic fan rotor with the tip leakage flow modeled was successfully computed. The best agreement with experiment was found using the Green-Gauss gradient method with the Dervieux limiter. The un-weighted least squares with QR (LSQR) was found to be unsuitable for simulating rotating geometries.

CHAPTER IX

FUTURE WORK

The goal of this sought to improve the accuracy of the numerical simulation of turbomachinery flows. Based upon the results of this effort, the following work should be done to further this goal:

1. Investigate in detail the hypothesis that the solution limiters were responsible for instigating the numerical “shock instabilities” that were found for the convergent-divergent nozzle flows with normal shocks. Studies exploring different grid configurations and the use of gradient methods other than un-weighted LSQR should be included.
2. Develop an improved implicit smoothing algorithm which does not smooth boundary nodes for equations which have hard set values. The dampening of high frequency numerical errors that is associated with implicit residual smoothing could prove useful for further improving the convergence of the turbulence equations. Also, the impact on the flow must be addressed as many of the flow equation boundary conditions for viscous flows are rigidly enforced.
3. Develop a weighted LSQR gradient method suitable for turbomachinery flows. The new algorithm would overcome the observed shortcomings of the un-weighted LSQR gradient method.
4. Further investigate the accuracy of the flow solver for turbomachinery aeromechanical flows.

REFERENCES

- Abbott, I. H., & Doenhoff, A. E. V. (1959). *Theory of Wing Sections*. New York: Dover Publications.
- Adamczyk, J. J., Celestina, M. L., & Grietzer, E. M. (1993). The role of tip clearance in high-speed fan stall. *Journal of Turbomachinery*, *115*, 28–38. doi:10.1115/1.2929212.
- Ainley, D. G., & Mathieson, G. C. R. (1951). *A Method of Performance Estimation for Axial-Flow Turbines*. Aeronautical Research Council Reports and Memoranda 2974 Ministry of Supply London: Her Majesty's Stationary Office.
- Anderson, J. D. (1995). *Computational Fluid Dynamics: The Basics with Applications*. McGraw-Hill.
- Anderson, W. K. A., & Bonhaus, D. L. (1994). An implicit upwind algorithm for computing turbulent flows on unstructured grids. *Computers & Fluids*, *23*, 1–21. doi:10.1016/0045-7930(94)90023-x.
- Arima, T., Sonoda, T., Shirotori, M., Tamura, A., & Kikuchi, K. (1999). A numerical investigation of transonic axial compressor rotor flow using a low-reynolds number $k-\epsilon$ turbulence model. *Journal of Turbomachinery*, *121*, 44–58. doi:10.1115/1.2841233.
- Arnone, A. (1994). Viscous analysis of three-dimensional rotor flow using a multigrid method. *Journal of Turbomachinery*, *116*, 435–445. doi:10.1115/1.2929430.
- Arnone, A., Liou, M.-S., & Povinelli, L. A. (1993). Multigrid calculation of three-dimensional viscous cascade flows. *Journal of Propulsion and Power*, *9*, 605–614. doi:10.2514/3.23664.
- Barth, T. J., & Jespersen, D. C. (1989). The design and application of upwind schemes on unstructured meshes. In *27th Aerospace Sciences Meeting AIAA-89-0366*. AIAA. doi:10.2514/6.1989-366.
- Blazek, J. (2005). *Computational Fluid Dynamics: Principles and Applications*. (2nd ed.). Elsevier.
- Brennan, G., Harvey, N. W., Rose, M. G., Fomison, N., & Taylor, M. D. (2003). Improving the efficiency of the trent 500-hp turbine using nonaxisymmetric end walls—part i: Turbine design. *Journal of Turbomachinery*, *125*, 497–504. doi:10.1115/1.1450766.

- Brenner, T. A., Carpenter, F. L., Freno, B. A., & Cizmas, P. G. A. (2013). A reduced-order model for turbomachinery flows using proper orthogonal decomposition. In *ASME Turbo Expo 2013: Turbine Technical Conference and Exposition* GT2013-94914. ASME. doi:10.1115/gt2013-94914.
- Brown, R. L. (2016). *Numerical Simulations of Hypersonic Aerothermoelastic Phenomena*. Dissertation Texas A&M University.
- Brown, R. L., Das, K., Cizmas, P. G. A., & Whitcomb, J. D. (2014). Numerical investigation of actively cooled structures in hypersonic flow. *Journal of Aircraft*, 51, 1522–1531. doi:10.2514/1.c032394.
- Celik, I. B., Ghia, U., Roache, P. J., Freitas, C. J., Coleman, H., & Raad, P. E. (2008). Procedure for estimation and reporting of uncertainty due to discretization in CFD applications. *Journal of Fluids Engineering*, 130. doi:10.1115/1.2960953.
- Chen, J. P., Ghosh, A. R., Sreenivas, K., & Whitfield, D. L. (1997). Comparison of computations using navier–stokes equations in rotating and fixed coordinates for flow through turbomachinery. In *35th Aerospace Sciences Meeting and Exhibit* AIAA-97-0878. AIAA. doi:10.2514/6.1997-878.
- Chima, R. V. (1991). *Viscous Three-Dimensional Calculations of Transonic Fan Performance*. Technical Memorandum 103800 NASA Cleveland, Ohio.
- Chima, R. V. (1998a). Calculation of multistage turbomachinery using steady characteristic boundary conditions. In *36th AIAA Aerospace Sciences Meeting and Exhibit* 98-0968. Reno, Nevada: AIAA. doi:10.2514/6.1998-968.
- Chima, R. V. (1998b). Calculation of tip clearance effects in a transonic compressor rotor. *Journal of Turbomachinery*, 120, 131–140. doi:10.1115/1.2841374.
- Chima, R. V. (2003). *SWIFT - Multiblock Analysis Code for Turbomachinery: User's Manual and Documentation (Version 300)*. NASA Glenn Research Center Cleveland, OH 44135.
- Chima, R. V., & Liou, M.-S. (2003). *Comparison of the AUSM⁺ and H-CUSP Schemes for Turbomachinery Application*. Technical Memorandum 2003-212457 NASA Glenn Research Center Cleveland, Ohio.
- Chima, R. V., & Strazisar, A. J. (1983). Comparison of two- and three-dimensional flow computations with laser anemometer measurements in a transonic compressor rotor. *Journal of Engineering for Power*, 105, 596–603. doi:10.1115/1.3227459.
- Chima, R. V., & Yokota, J. W. (1990). Numerical analysis of three-dimensional viscous internal flows. *AIAA Journal*, 28, 798–806. doi:10.2514/3.25122.

- Cizmas, P. G. A. (2014). *UNS3D-POD User's Manual: Release 5.1*. Aerospace Numerical Simulation Laboratory at Texas A&M University.
- Cizmas, P. G. A., & Palacios, A. (2003). Proper orthogonal decomposition of turbine rotor-stator interaction. *Journal of Propulsion and Power*, *19*, 268–281. doi:10.2514/2.6108.
- Cizmas, P. G. A., Richardson, B. R., Brenner, T. A., O'Brien, T. J., & Breault, R. W. (2008). Acceleration techniques for reduced-order models based on proper orthogonal decomposition. *Journal of Computational Physics*, *227*, 7791–7812. doi:10.1016/j.jcp.2008.04.036.
- Cumpsty, N. A. (2004). *Compressor Aerodynamics*. Malabar, Florida: Krieger.
- Cumpsty, N. A., & Horlock, J. H. (2006). Averaging nonuniform flow for a purpose. *Journal of Turbomachinery*, *128*, 120–129. doi:10.1115/1.2098807.
- Dawes, W. N. (1988). Development of a 3d navier stokes solver for application to all types of turbomachinery. In *Gas Turbine and Aeroengine Congress 88-GT-70*. Amsterdam, The Netherlands: ASME. doi:10.1115/88-gt-70.
- Denton, J. D. (1975). *A Time Marching Method for Two- and Three-Dimensional Blade to Blade Flows*. Aeronautical Research Council Reports and Memoranda 3775 Ministry of Defence London: Her Majesty's Stationary Office.
- Denton, J. D. (1983). An improved time-marching method for turbomachinery flow calculation. *Journal of Engineering for Power*, *105*, 514–521. doi:10.1115/1.3227444.
- Denton, J. D. (1986). The use of a distributed body force to simulate viscous effects in 3d flow calculations. In *International Gas Turbine Conference and Exhibit 86-GT-144*. Dusseldorf, West Germany: ASME. doi:10.1115/86-gt-144.
- Denton, J. D. (1993). Loss mechanisms in turbomachines. *Journal of Turbomachinery*, *115*, 621–656. doi:10.1115/93-GT-435.
- Dervieux, A. (1985). Steady euler simulations using unstructured meshes. *Computational Fluid Dynamics, VKI Lecture Series 1985-04*, *1*.
- Doi, H., & Alonso, J. J. (2002). Fluid/structure coupled aeroelastic computations for transonic flows in turbomachinery. In *ASME Turbo Expo 2002 GT-2002-30313*. Amsterdam, The Netherlands: ASME. doi:10.1115/GT2002-30313.
- Flitan, H. C., & Cizmas, P. G. A. (2003). Analysis of unsteady aerothermodynamics effects in a turbine combustor. In K. C. Hall, R. E. Kielb, & J. P. Thomas (Eds.), *Unsteady Aerodynamics and Aeroelasticity of Turbomachines* (pp. 551–556). Springer. doi:10.1007/1-4020-4605-7_40.

- Fransson, T. H., & Verdon, J. M. (1993). Panel discussion on standard configurations for unsteady flow through vibrating axial-flow turbomachine-cascades. In H. M. Atassi (Ed.), *Unsteady Aerodynamics, Aeroacoustics, and Aeroelasticity of Turbomachines and Propellers*. Springer Science & Business Media. doi:10.1007/978-1-4613-9341-2_44.
- Gargoloff, J. I. (2007). *A Numerical Method for Fully Nonlinear Aeroelastic Analysis*. Dissertation Texas A&M University.
- Godunov, S. K. (1959). A difference scheme for numerical computation discontinuous solution of hydrodynamic equations. *Math. Sbornik*, *47*, 271–306. Translated US Joint Publications Research Service, JPRS 7226, 1969.
- Goldman, L. J., & McLallin, K. L. (1977). *Effect of Endwall Cooling on Secondary Flows in Turbine Stator Vanes*. Technical Memorandum X-73568 NASA.
- Goldman, L. J., & Seasholtz, R. G. (1982). *Laser Anemometer Measurements in an Annular Cascade of Core Turbine Vanes and Comparison With Theory*. Technical Paper 2018 NASA.
- Hah, C. (1984). A navier–stokes analysis of three-dimensional turbulent flows inside turbine blade rows at design and off-design conditions. *Journal of Engineering for Gas Turbines and Power*, *106*, 420–429. doi:10.1115/1.3239582.
- Hah, C., & Leylek, J. H. (1987). Numerical solution of three-dimensional turbulent flows for modern gas turbine components. In *Gas Turbine Conference and Exhibition 87-GT-84*. Anaheim, California: ASME. doi:10.1115/87-GT-84.
- Hah, C., & Reid, L. (1992). A viscous flow study of shock-boundary layer interaction, radial transport, and wake development in a transonic compressor. *Journal of Turbomachinery*, *114*, 538–547. doi:10.1115/1.2929177.
- Harten, A., & Hyman, J. M. (1983). Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *Journal of Computational Physics*, *50*, 235–269. doi:10.1016/0021-9991(83)90066-9.
- Haselbacher, A., & Blazek, J. (2000). Accurate and efficient discretization of navier–stokes equations on mixed grids. *AIAA Journal*, *38*, 2094–2102. doi:10.2514/2.871.
- Hellsten, A. (1998). Some improvements in menter’s $k - \omega$ sst turbulence model. AIAA-98-2554. AIAA. doi:10.2514/6.1998-2554.
- Hill, P. G., & Peterson, C. R. (1992). *Mechanics and Thermodynamics of Propulsion*. (2nd ed.). Addison-Wesley.

- Howell, A. R. (1945a). Design of axial compressors. *Proceedings of the Institution of Mechanical Engineers*, 153, 452–462.
- Howell, A. R. (1945b). Fluid dynamics of axial compressors. *Proceedings of the Institution of Mechanical Engineers*, 153, 441–452.
- Jameson, A., Baker, T. J., & Weatherill, N. P. (1986). Calculation of inviscid transonic flow over a complete aircraft. In *AIAA 24th Aerospace Sciences Meeting* AIAA-86-0103. AIAA. doi:10.2514/6.1986-103.
- Jameson, A., Schmidt, W., & Turkel, E. (1981). Numerical solutions of the euler equations by finite volume methods using runge-kutta time-stepping schemes. In *AIAA 14th Fluid and Plasma Dynamics Conference* AIAA-81-1259. AIAA. doi:10.2514/6.1981-1259.
- Jennions, I. K., & Turner, M. G. (1993). Three-dimensional navier–stokes computations of transonic fan flow using an explicit flow solver and an implicit k- ϵ solver. *Journal of Turbomachinery*, 115, 261–272. doi:10.1115/1.2929232.
- Key, N. L., Lawless, P. B., & Fleeter, S. (2010). Rotor wake variability in a multistage compressor. *Journal of Propulsion and Power*, 26, 344–352. doi:10.2514/1.45715.
- Kim, K. (2003). *Three-Dimensional Hybrid Grid Generator and Unstructured Flow Solver for Compressors and Turbines*. Dissertation Texas A&M University.
- soo Kim, S., Kim, C., Rho, O.-H., & Hong, S. K. (2003). Cures for the shock instability: Development of a shock-stable roe scheme. *Journal of Computational Physics*, 185, 342–374. doi:10.1016/s0021-9991(02)00037-2.
- Kitamura, K., Roe, P., & Ismail, F. (2009). Evaluation of euler fluxes for hypersonic flow computations. *AIAA Journal*, 47, 44–53. doi:10.2514/1.33735.
- Lakshminarayana, B., & Horlock, J. H. (1963). Review: Secondary flows and losses in cascades and axial-flow turbomachines. *International Journal of Mechanical Sciences*, 5, 287–307. doi:10.1016/0020-7403(63)90055-9.
- Langston, L. S. (2001). Secondary flows in axial turbines – a review. *Annals of the New York Academy of Sciences*, 934, 11–26. doi:10.1111/j.1749-6632.2001.tb05839.x.
- Lee, S.-Y., & Kim, K.-Y. (2000). Design optimization of axial flow compressor blades with three-dimensional navier-stokes solver. *KSME International Journal*, 14, 1005–1012.
- LeGresley, P. A., & Alonso, J. J. (2000). Airfoil design optimization using reduced order models based on proper orthogonal decomposition. In *FLUIDS 2000 Conference and Exhibit* 2000-2545. AIAA. doi:10.2514/6.2000-2545.

- Liliedahl, D. N., Carpenter, F. L., & Cizmas, P. G. A. (2011). Prediction of aeroacoustic resonance in cavities of hole-pattern stator seals. *Journal of Engineering for Gas Turbines and Power*, 133. doi:10.1115/1.4002038.
- Liou, M.-S. (1987). A generalized procedure for constructing an upwind-based tvd scheme. In *AIAA 25th Aerospace Sciences Meeting AIAA-87-0355*. AIAA. doi:10.2514/6.1987-355.
- Matula, N. R. (2014). Private Communication.
- Mavriplis, D. J. (2003). Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. In *16th AIAA Computational Fluid Dynamics Conference 2003-3986*. AIAA. doi:10.2514/6.2003-3986.
- Menter, F. R. (1993). Zonal two equation $k - \omega$ turbulence models for aerodynamic flows. In *23rd Fluid Dynamics, Plasmadynamics, and Lasers Conference AIAA-93-2906*. AIAA. doi:10.2514/6.1993-2906.
- Menter, F. R. (1994). Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32, 1598–1605. doi:10.2514/3.12149.
- Menter, F. R. (2009). Review of the shear-stress transport turbulence model experience from an industrial perspective. *International Journal of Computational Fluid Dynamics*, 23, 305–316. doi:10.1080/10618560902773387.
- Menter, F. R., Kuntz, M., & Langtry, R. (2003). Ten years of industrial experience with the sst turbulence model. In K. Hanjalic, Y. Nagano, & M. Tummers (Eds.), *Turbulence, Heat and Mass Transfer 4* (pp. 625–632). Begell House, Inc.
- NASA (2013). Grids - 2d zero pressure gradient flat plate verification case. URL: http://turbmodels.larc.nasa.gov/flatplate_grids.html.
- NASA (2015). 2d zero pressure gradient flat plate verification case. URL: <http://turbmodels.larc.nasa.gov/flatplate.html>.
- Park, J. S., & Kim, C. (2012). Multi-dimensional limiting process for finite volume methods on unstructured grids. *Computers & Fluids*, 65, 8–24. doi:10.1016/j.compfluid.2012.04.015.
- Park, J. S., Yoon, S.-H., & Kim, C. (2010). Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids. *Journal of Computational Physics*, 229, 788–812. doi:10.1016/j.jcp.2009.10.011.
- Parker, R., & Watson, J. F. (1972). Interaction effects between blade rows in turbomachines. *Proceedings of the Institution of Mechanical Engineers*, 186, 331–340. doi:10.1243/pime_proc_1972_186_036_02.

- Pierzga, M. J., & Wood, J. R. (1985). Investigation of the three-dimensional flow field within a transonic fan rotor: Experiment and analysis. *J. Eng. Gas Turbines Power*, *107*, 436–448. doi:10.1115/1.3239744.
- Pointwise (2011). *Pointwise User Manual*. Pointwise, Inc. Fort Worth, TX.
- Roe, P. L. (1981). Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, *43*, 357–372. doi:10.1016/0021-9991(81)90128-5.
- Roe, P. L. (1986). Characteristic based schemes for the euler equations. *Annual Review of Fluid Mechanics*, *18*, 337–365. doi:10.1146/annurev.fluid.18.1.337.
- Rubechini, F., Schneider, A., Arnone, A., Cecchi, S., & Malavasi, F. (2012). A redesign strategy to improve the efficiency of a 17-stage steam turbine. *Journal of Turbomachinery*, *134*. doi:10.1115/1.4003082.
- Sanders, A. J., & Fleeter, S. (2002). Rotor blade-to-blade wake variability and effect on downstream vane response. *Journal of Propulsion and Power*, *18*, 456–464. doi:10.2514/2.5956.
- Schlichting, H. (1968). *Boundary-Layer Theory*. (Sixth ed.). McGraw-Hill.
- Smith, L. H. (1955). Secondary flow in axial-flow turbomachinery. *Transactions of the ASME*, *77*, 1065–1076.
- Staff (1953). *Equations, Tables, and Charts for Compressible Flow*. Technical Report 1135 NACA.
- Subramanian, S. V., & Bozzola, R. (1987). Numerical simulation of three-dimensional flow fields in turbomachinery blade rows using the compressible navier–stokes equations. In *19th Fluid Dynamics, Plasma Dynamics and Lasers Conference* 87-1314. AIAA. doi:10.2514/6.1987-1314.
- Thompson, J. F., Thames, F. C., & Mastin, C. W. (1974). Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies. *Journal of Computational Physics*, *15*, 299–319. doi:10.1016/0021-9991(74)90114-4.
- Ursek, D. C., Gorrell, W. T., & Cunnan, W. S. (1979). *Performance of a Two-Stage Fan Having Low-Aspect-Ratio, First-Stage Rotor Blading*. Technical Paper 1493 NASA.
- Venkatakrishnan, V. (1995). Convergence to steady state solutions of the euler equations on unstructured grids with limiters. *Journal of Computational Physics*, *118*, 120–130. doi:10.1006/jcph.1995.1084.

- Wang, Z. J. (2000). A fast nested multi-grid viscous flow solver for adaptive cartesian/quad grids. *International Journal for Numerical Methods in Fluids*, 33, 657–680.
- Weingold, H. D., Neubert, R. J., Behlke, R. F., & Potter, G. E. (1997). Bowed stators: An example of CFD applied to improve multistage compressor efficiency. *Journal of Turbomachinery*, 119, 161–168. doi:10.1115/1.2841086.
- Whitaker, D. L. (1993). Three-dimensional unstructured grid euler computations using a fully-implicit, upwind methods. In *11th AIAA Computational Fluid Dynamics Conference AIAA-93-3337*. doi:10.2514/6.1993-3337.
- White, F. M. (2006). *Viscous Fluid Flow*. (3rd ed.). McGraw-Hill.
- Wiegardt, K., & Tillmann, W. (1951). *On the Turbulent Friction Layer for Rising Pressure*. Technical Memorandum 1314 NACA.
- Wilcox, D. C. (2010). *Turbulence Modeling for CFD*. (3rd ed.). DCW Industries.
- Wood, J. R., Strazisar, T., & Hathaway, M. (1990). Test case e/co-2: Single transonic fan rotor. In L. Fottner (Ed.), *Test Cases for Computation of Internal Flows in Aero Engine Components* 275 (pp. 165–213). AGARD.
- Wu, C.-H. (1952). *A General Theory of Three-Dimensional Flow in Subsonic and Supersonic Turbomachines of Axial-, Radial-, and Mixed-Flow Types*. Technical Note 2604 NACA.
- Yamamoto, K., & Engel, K. (1997). Multi-block grid generation using an elliptic differential equation. In *35th Aerospace Sciences Meeting and Exhibit AIAA-97-0201*. doi:10.2514/6.1997-201.
- Yoder, D. A., & Georgiadis, N. J. (1999). Implementation and validation of the chien $k-\epsilon$ turbulence model in the WIND navier–stokes code. In *37th AIAA Aerospace Sciences Meeting and Exhibit AIAA-99-0745*. doi:10.2514/6.1999-745.

APPENDIX A

SUPPLEMENTAL TRANSONIC FAN RESULTS

This appendix contains tables and figures that supplement the data presented in Chapters VI and VII for the NASA transonic fan rotor known as Rotor 67. See Section VI.1 for a detailed description of the rotor.

The purpose of this appendix is to tabulate the data used to generate the compressor map speedlines presented in the main body, and to provide the reader with additional data that could be used for future comparison. The supplemental data is grouped based upon the case it belongs to, and is ordered according to its introduction in the main body.

Notes

- The “Case ID” given in the speed line tables corresponds to the directory name in which the corresponding data is stored.
- All given pressures have been normalized by $p_{0_1} = 124452.598$ Pa.
- Percent span was measured from the hub.

A.1. Inviscid Rotor 67

Table A.1: Inviscid Rotor 67 compressor map speed line outlet pressure boundary condition and results summary.

Case ID	Input	Result		
	$p_2^{(h)}/p_{01}$	\dot{m}_{eq} [kg/s]	π^*	η_{tt}
R009 [‡]	0.850	35.0478	1.46532	0.86189
R008	0.900	35.0474	1.52765	0.88015
R007	0.950	35.0442	1.59497	0.89885
R001 [‡]	1.000	35.0228	1.66414	0.81737
R002	1.030	34.9826	1.70679	0.92846
R003	1.060	34.8929	1.75053	0.94016
R015	1.075	34.8108	1.77285	0.94650
R004 [‡]	1.090	34.6394	1.79344	0.95429
R005	1.095	34.4215	1.79501	0.95759
R006	1.100	33.2429	1.78375	0.93955
R010	1.102	33.0458	1.78491	0.93671
R011	1.104	32.8505	1.78622	0.93395
R012 [‡]	1.108	32.4388	1.78919	0.92815
R013	1.115	31.5688	1.79549	0.91588

[‡] “Max flow” operating point [‡] Initial operating point [‡] “Near peak efficiency” operating point [‡] “Near stall” operating point

A.2. Turbulent Rotor 67: Rubbing Configuration

- R67TR-a: All viscous walls rotate with speed N_{eq} .
- R67TR-b: Only the viscous wall boundary faces between xr_{le} and xr_{te} rotate with speed N_{eq} .

Table A.2: Turbulent Rotor 67: Rubbing case R67TR-a compressor map speed line outlet boundary condition and results summary.

Case ID	Input	Result			
	$p_2^{(h)}/p_{01}$	$p_{m2}^{(h)}/p_{01}^\dagger$	\dot{m}_{eq} [kg/s]	π^*	η_{tt}
P004/R001 [‡]	0.8643	0.9116	33.8344	1.4497	0.8761
P002/R001	0.8819	0.9244	33.8085	1.4702	0.8843
P001/R004 [‡]	0.8999	0.9376	33.7723	1.4920	0.8930
P003/R001	0.9179	0.9514	33.7060	1.5149	0.9024
P005/R001	0.9363	0.9670	33.5581	1.5395	0.9106
P006/R001 [‡]	0.9550	0.9821	33.3290	1.5635	0.9144
P009/R001	0.9642	0.9876	32.6508	1.5697	0.9049
P010/R001 [‡]	0.9690	0.9232	16.7300	1.5888	0.8821
P007/R001 [‡]	0.9741	0.9207	18.9436	1.5976	0.8875
P008/R001 [‡]	0.9936	0.9349	18.1848	1.6282	0.8928

[†] Hub static pressure at downstream measurement plane

[‡] “Max flow” operating point [‡] Initial operating point

[‡] “Near peak efficiency” operating point [‡] Stalled flow encountered

Table A.3: Turbulent Rotor 67: Rubbing case R67TR-b compressor map speed line outlet boundary condition and results summary.

Case ID	Input	Result			
	$p_2^{(h)}/p_{01}$	$p_{m2}^{(h)}/p_{01}^\dagger$	\dot{m}_{eq} [kg/s]	π^*	η_{tt}
P004/R001 [‡]	0.8643	0.9054	34.2151	1.4382	0.8627
P001/R002 [‡]	0.8999	0.9326	34.1732	1.4791	0.8781
P005/R001	0.9363	0.9600	34.0915	1.5238	0.8938
P007/R001	0.9741	0.9909	33.8809	1.5744	0.9105
P008/R001 [‡]	0.9936	1.0070	33.6605	1.6012	0.9160
P009/R001	1.0050	1.0170	33.3286	1.6165	0.9145
P010/R001	1.0100	1.0208	33.1420	1.6225	0.9129
P011/R001	1.0200	1.0279	32.7625	1.6340	0.9095
P012/R001 [‡]	1.0400	1.0412	16.2432	1.7027	0.8973

[†] Hub static pressure at downstream measurement plane

[‡] “Max flow” operating point [‡] Initial operating point

[‡] “Near peak efficiency” operating point [‡] Stalled flow encountered

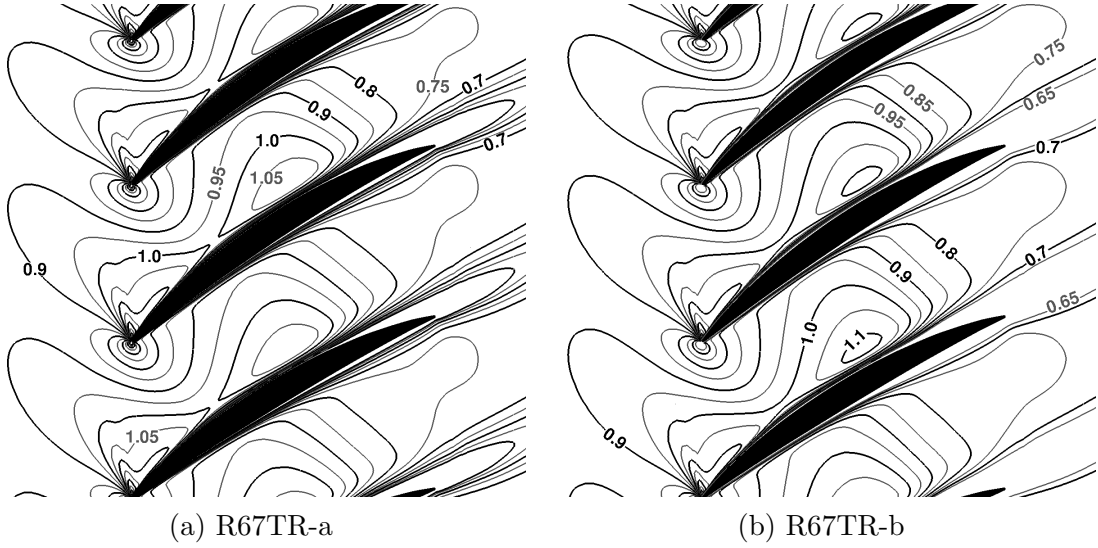


Figure A.1: Predicted relative Mach number contours ($\Delta M_{rel} = 0.05$) at 30% span for "max flow".

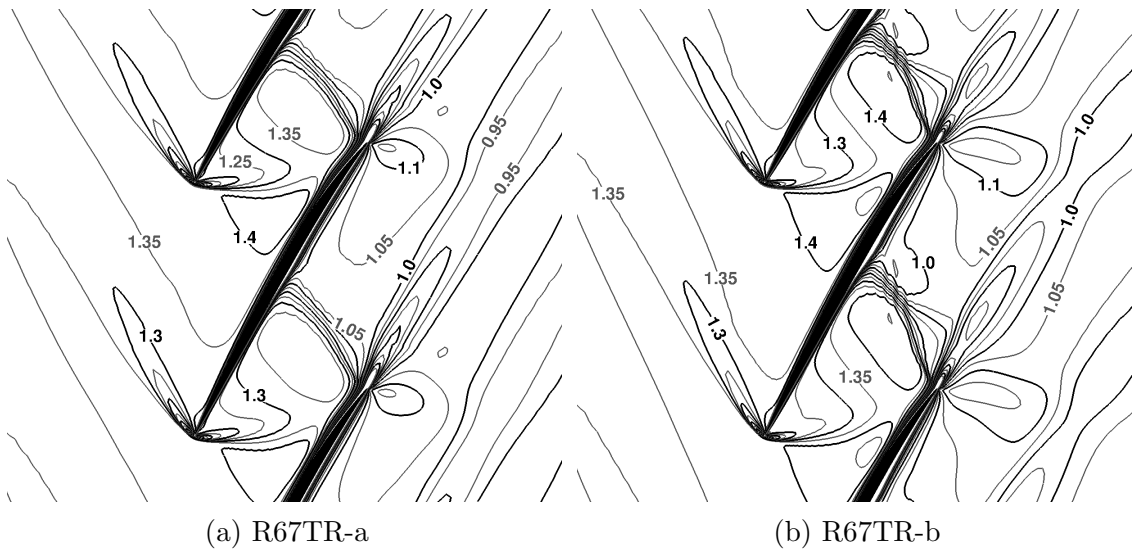


Figure A.2: Predicted relative Mach number contours ($\Delta M_{rel} = 0.05$) at 90% span for "max flow".

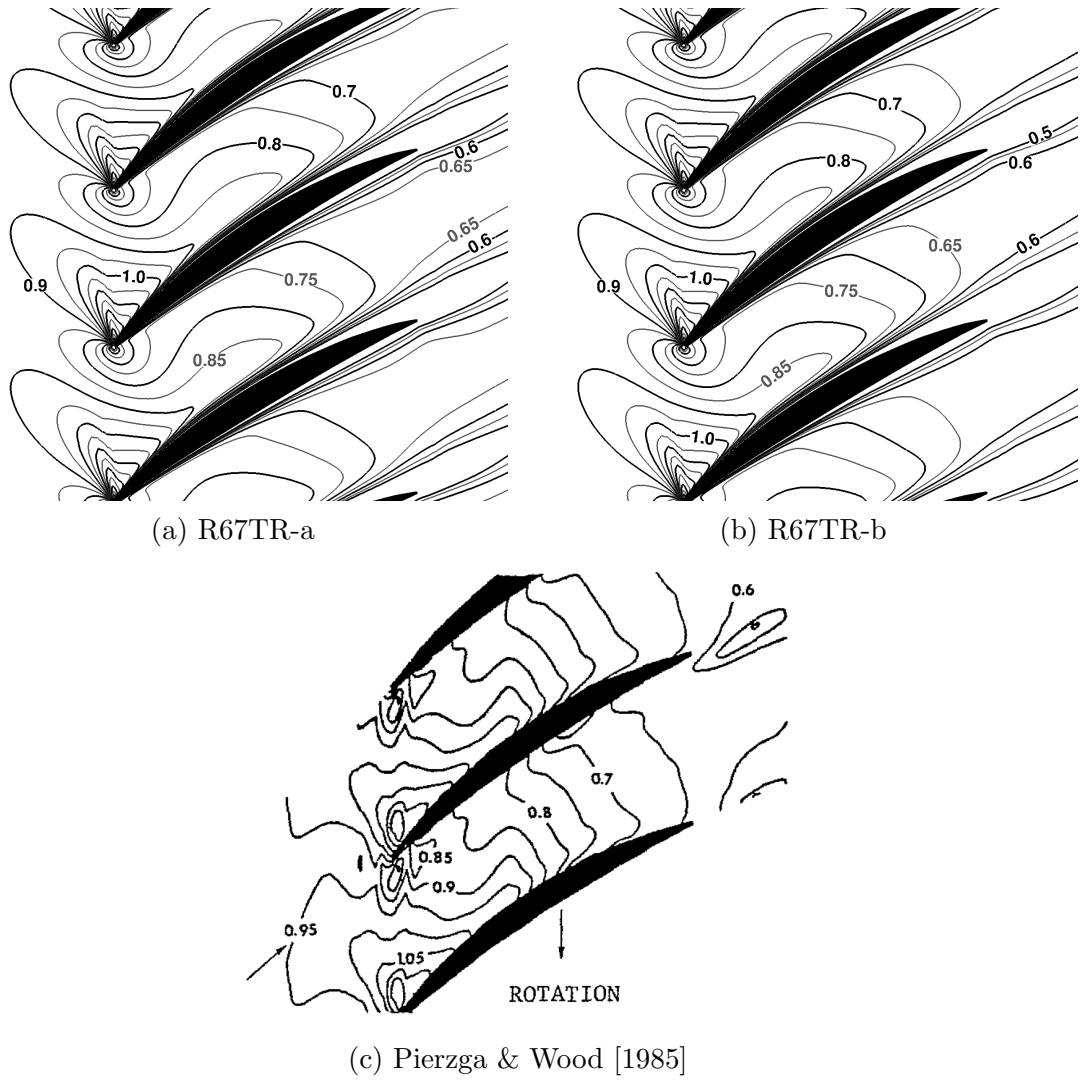


Figure A.3: Predicted and measured [Pierzga & Wood, 1985] relative Mach number contours ($\Delta M_{rel} = 0.05$) at 30% span for “near peak efficiency”.

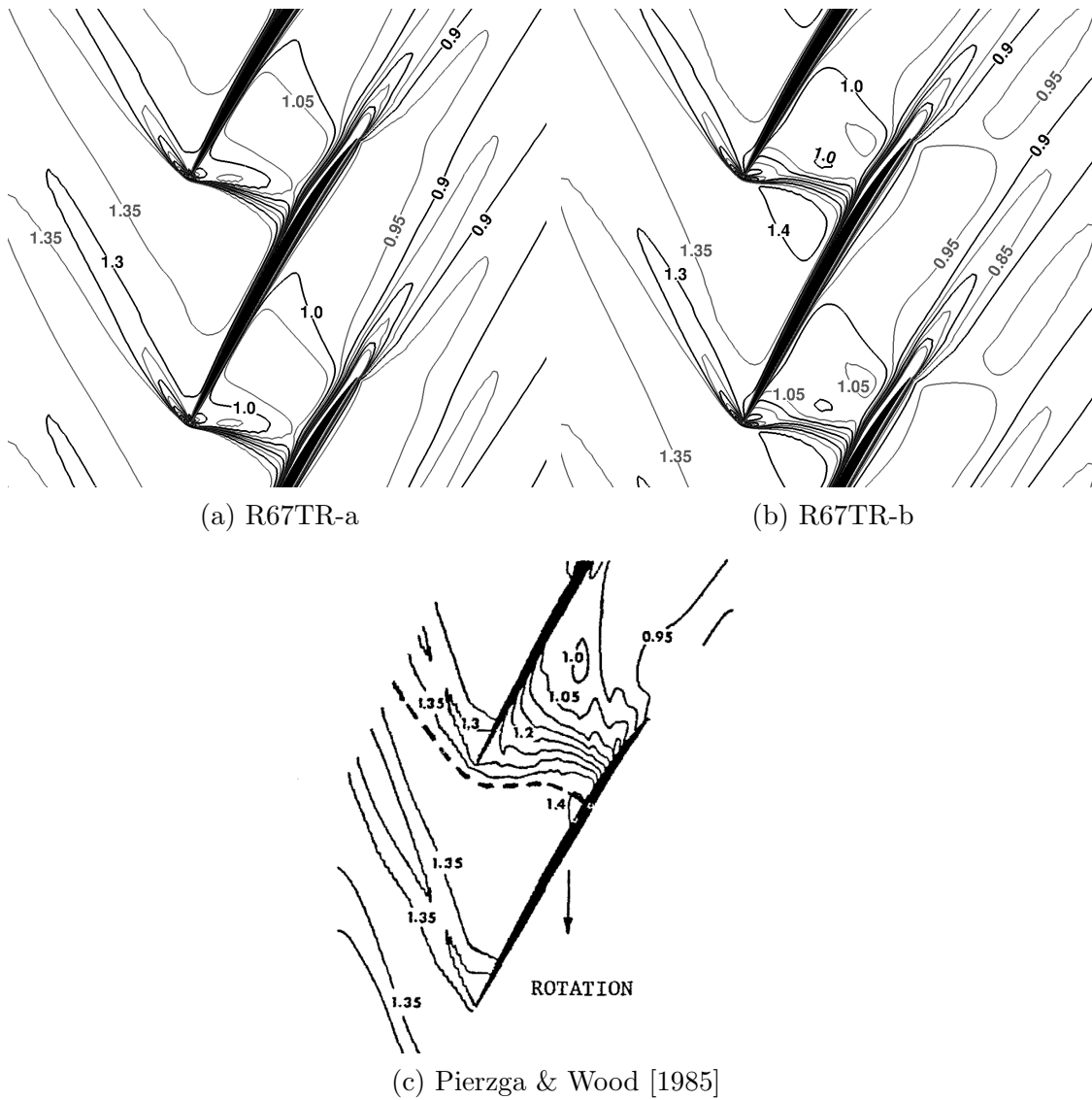


Figure A.4: Predicted and measured [Pierzga & Wood, 1985] relative Mach number contours ($\Delta M_{rel} = 0.05$) at 90% span for “near peak efficiency”.

A.3. Turbulent Rotor 67: Clearance Flow Configuration

Table A.4: Turbulent Rotor 67 compressor map speed line outlet boundary condition and results summary using the Dervieux limiter and the un-weighted LSQR gradient method.

Case ID	Input	Result			
	$p_2^{(h)}/p_{01}$	$p_{m2}^{(h)}/p_{01}^\dagger$	\dot{m}_{eq} [kg/s]	π^*	η_{tt}
P006/R001 [‡]	0.8493	0.8900	34.2146	1.4164	0.8524
P002/R001	0.8643	0.9039	34.1885	1.4354	0.8602
P005/R001	0.8999	0.9330	34.0596	1.4759	0.8777
P001/R001 [‡]	0.9363	0.9604	33.8125	1.5165	0.8890
P007/R001	0.9552	0.9744	33.6308	1.5383	0.8936
P003/R001	0.9741	0.9888	33.4061	1.5611	0.8975
P008/R001 [‡]	0.9839	0.9964	33.2609	1.5734	0.8988
P009/R001	0.9883	0.9998	33.1252	1.5781	0.8981
P010/R001	0.9910	1.0019	33.0726	1.5816	0.8983
P004/R001 [‡]	0.9936	0.9770	29.4456	1.5087	0.8520

[†] Hub static pressure at downstream measurement plane

[‡] “Max flow” operating point [‡] Initial operating point

[‡] “Near peak efficiency” operating point [‡] Stalled flow encountered

Table A.5: Turbulent Rotor 67 compressor map speed line outlet boundary conditions and results summary using the Dervieux limiter and the Green-Gauss gradient method.

Case ID	Input	Result			
	$p_2^{(h)}/p_{0_1}$	$p_{m_2}^{(h)}/p_{0_1}^\dagger$	\dot{m}_{eq} [kg/s]	π^*	η_{tt}
P006/R006 [‡]	0.8493	0.8914	34.4807	1.4298	0.8511
P002/R006	0.8643	0.9076	34.4696	1.4503	0.8579
P001/R006 [‡]	0.9363	0.9661	34.3229	1.5419	0.8902
P003/R006	0.9741	0.9925	34.1015	1.5877	0.9034
P004/R006 ^b	0.9936	1.0062	33.8839	1.6106	0.9081
P011/R006	1.0050	1.0147	33.6769	1.6232	0.9100
P012/R006	1.0100	1.0186	33.5121	1.6281	0.9098
P013/R006 [#]	1.0200	0.9929	29.2152	1.5498	0.8643

[†] Hub static pressure at downstream measurement plane

[‡] “Max flow” operating point [‡] Initial operating point

^b “Near peak efficiency” operating point [#] Stalled flow encountered

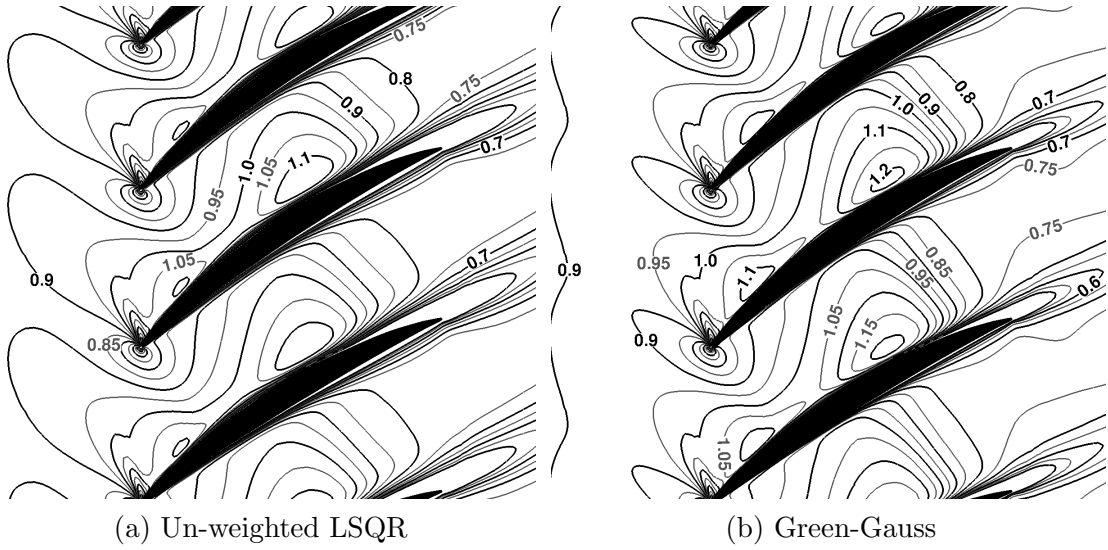


Figure A.5: Predicted relative Mach number contours ($\Delta M_{rel} = 0.05$) at 30% span for “max flow”.

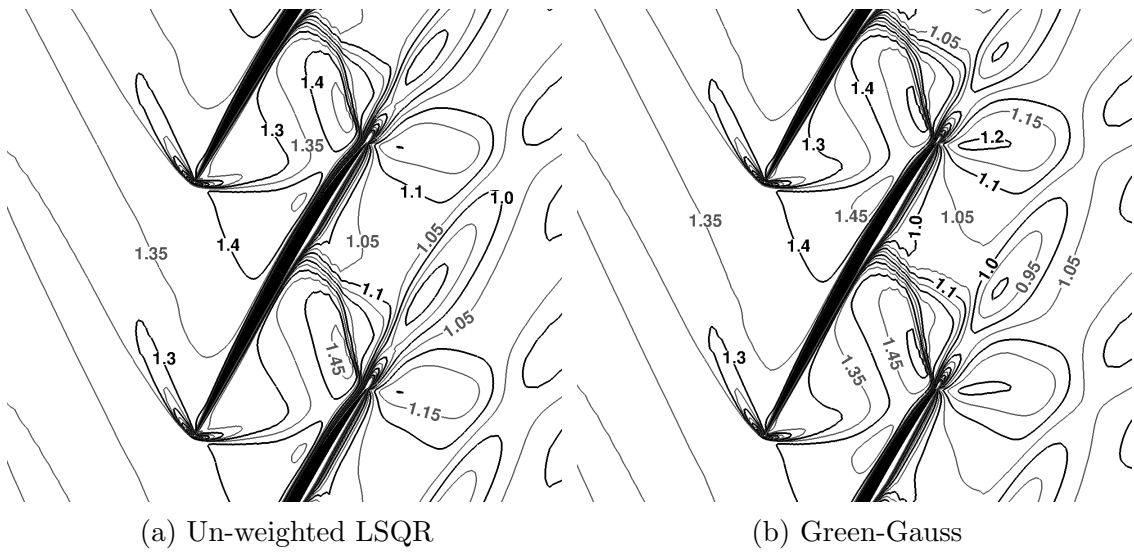


Figure A.6: Predicted relative Mach number contours ($\Delta M_{rel} = 0.05$) at 90% span for “max flow”.

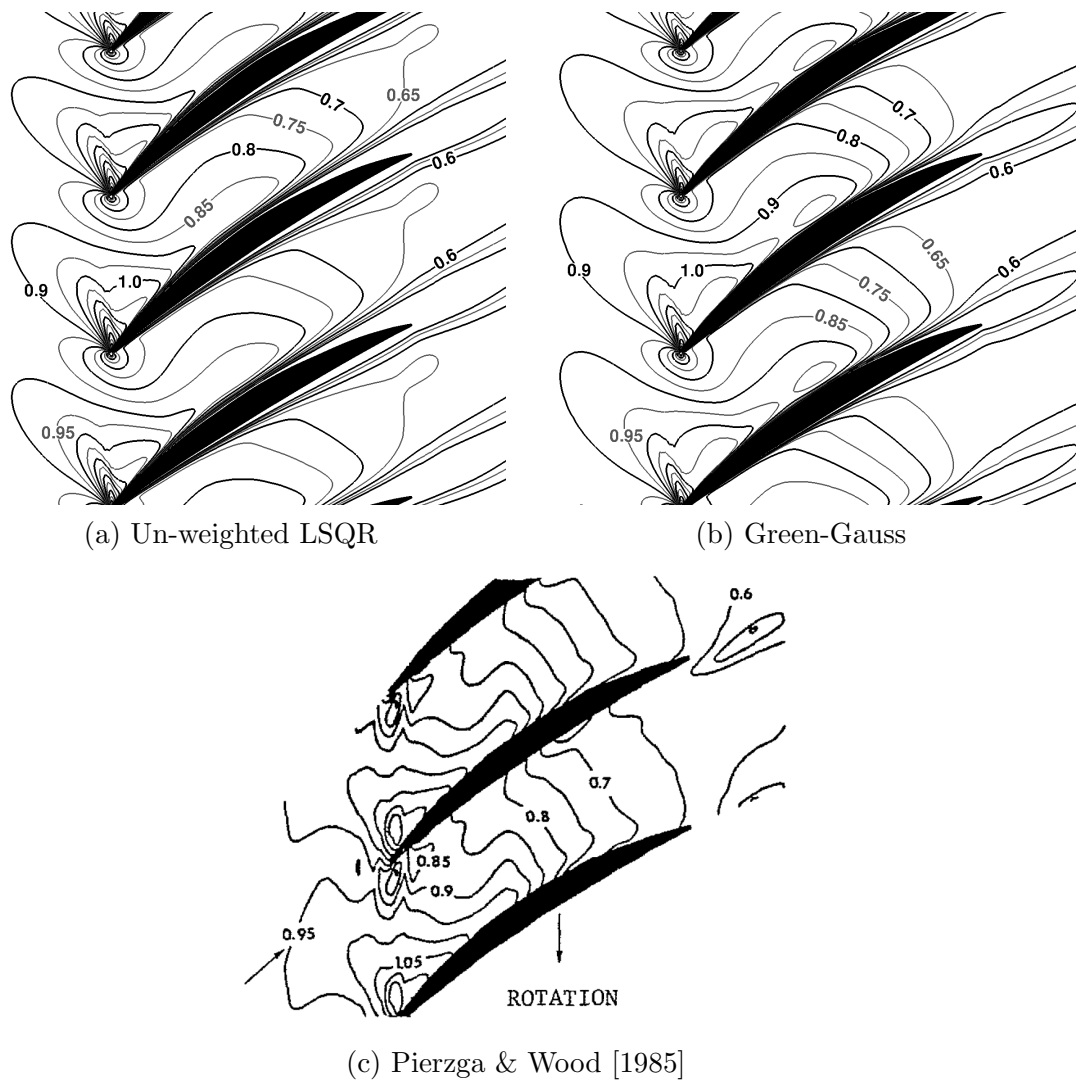


Figure A.7: Predicted and measured [Pierzga & Wood, 1985] relative Mach number contours ($\Delta M_{rel} = 0.05$) at 30% span for “near peak efficiency”.

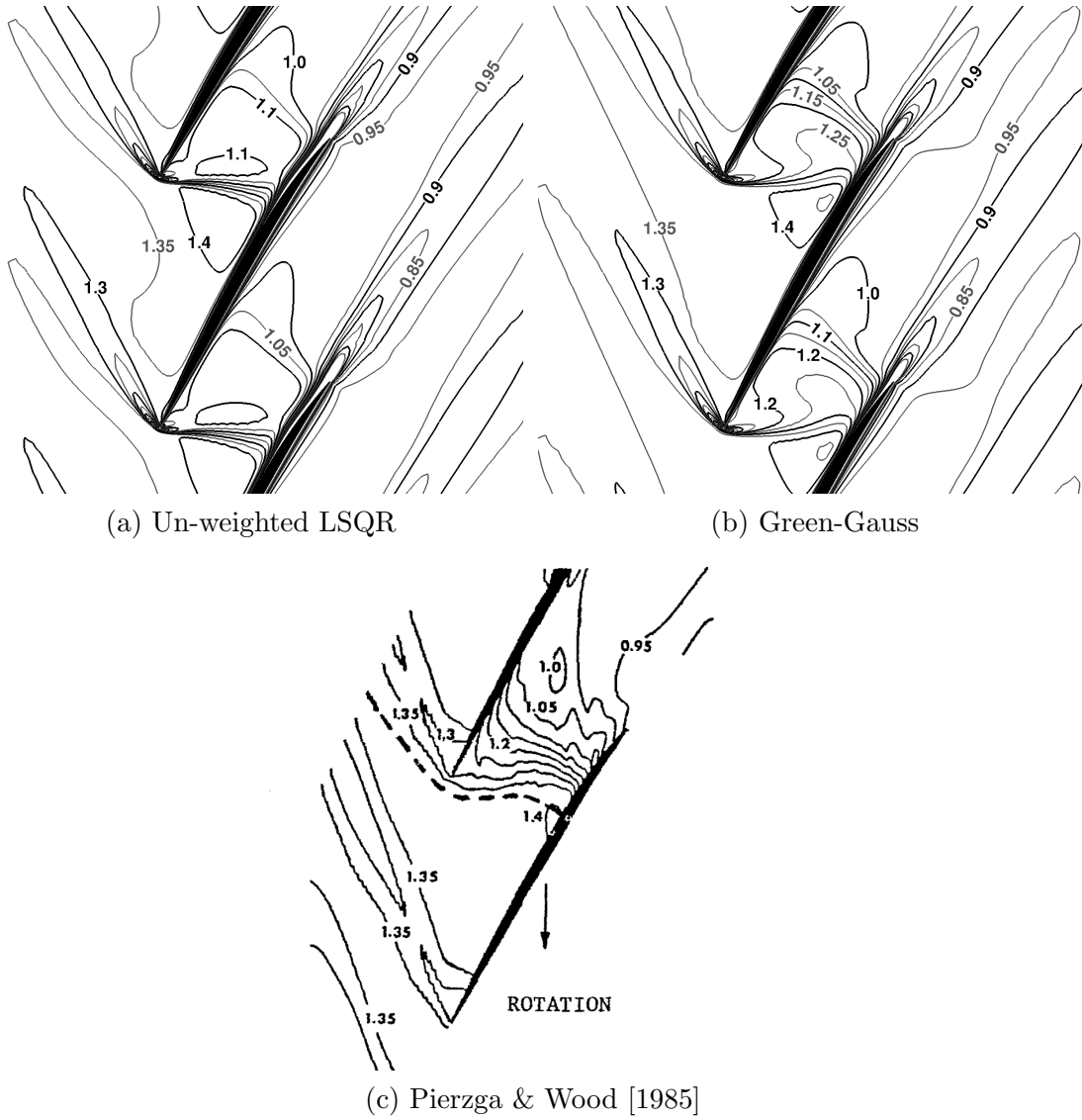


Figure A.8: Predicted and measured [Pierzga & Wood, 1985] relative Mach number contours ($\Delta M_{rel} = 0.05$) at 90% span for “near peak efficiency”.

APPENDIX B

GOVERNING EQUATION VELOCITY TRANSFORMATION

The derivation of the Navier-Stokes equations written in a rotating reference frame but cast in terms of absolute velocities is presented herein. The absolute velocity is defined as

$$\vec{U} = \vec{W} + \vec{V}, \quad (\text{B.1})$$

where \vec{W} is the relative velocity, and \vec{V} is the transport velocity. The transport velocity, defined in Eq. B.2, for a given point is given as the cross product between the rotation vector of the frame and a point. Only rotation about the x-axis, $\vec{\omega} = [\omega_x, 0, 0]^T$, is considered. The relative velocities are then expressed in terms of absolute velocity components by rearranging Eq. B.1 and using Eq. B.2, as shown by Eq. B.3.

$$\vec{V} = \vec{\omega} \times \vec{r} = \begin{vmatrix} i & j & k \\ \omega_x & 0 & 0 \\ x & y & z \end{vmatrix} = \begin{bmatrix} 0 \\ -z\omega_x \\ y\omega_x \end{bmatrix} \quad (\text{B.2})$$

$$\vec{W} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 + z\omega_x \\ u_3 - y\omega_x \end{bmatrix} \quad (\text{B.3})$$

B.1. Conservation of Mass

The integral form of the continuity equation in the rotating reference frame is given by

$$\frac{\partial}{\partial t} \int_{\Omega} \rho d\Omega + \oint_{\partial\Omega} \rho (\vec{W} \cdot \hat{n}) dS = 0. \quad (\text{B.4})$$

Applying the divergence theorem results in the differential form of the equation. That is

$$\begin{aligned} \int_{\Omega} \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{W}) \right] d\Omega &= 0 \\ \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{W}) &= 0 \\ \frac{\partial \rho}{\partial t} + \frac{\partial(\rho w_1)}{\partial x} + \frac{\partial(\rho w_2)}{\partial y} + \frac{\partial(\rho w_3)}{\partial z} &= 0. \end{aligned}$$

Substituting equation B.3 yields into the above differential equation yields

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_1)}{\partial x} + \frac{\partial}{\partial y} [\rho(u_2 + z\omega_x)] + \frac{\partial}{\partial z} [\rho(u_3 - y\omega_x)] &= 0 \\ \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_1)}{\partial x} + \frac{\partial(\rho u_2)}{\partial y} + \frac{\partial(\rho u_3)}{\partial z} &= \frac{\partial(\rho y\omega_x)}{\partial z} - \frac{\partial(\rho z\omega_x)}{\partial y}. \end{aligned} \quad (\text{B.5})$$

Equation B.5 is the final form of the continuity equation in the rotating reference frame cast in terms of absolute velocity. Transforming B.5 to integral form results in the starting equation, B.4.

B.2. Conservation of Momentum

The integral form of the momentum conservation equations in the rotating reference frame is given by

$$\begin{aligned} \frac{\partial}{\partial t} \int_{\Omega} \rho \vec{W} d\Omega + \oint_{\partial\Omega} \rho \vec{W} (\vec{W} \cdot \hat{n}) dS &= \int_{\Omega} \rho (\vec{f}_e + \vec{f}_{cor} + \vec{f}_{cen}) d\Omega - \oint_{\partial\Omega} p (\bar{\bar{I}} \cdot \hat{n}) dS \\ &+ \oint_{\partial\Omega} (\bar{\bar{\tau}} \cdot \hat{n}) dS \end{aligned} \quad (\text{B.6})$$

where $\bar{\bar{I}}$ is the identity matrix and $\bar{\bar{\tau}}$ is the stress tensor, which is given by

$$\bar{\bar{\tau}} = \begin{bmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \tau_{zz} \end{bmatrix}.$$

The components of the viscous stress tensor in the rotating reference frame are defined by

$$\begin{aligned}
\tau_{xx} &= \mu \frac{2}{3} \left(2 \frac{\partial w_1}{\partial x} - \frac{\partial w_2}{\partial y} - \frac{\partial w_3}{\partial z} \right) \\
\tau_{yy} &= \mu \frac{2}{3} \left(2 \frac{\partial w_2}{\partial y} - \frac{\partial w_1}{\partial x} - \frac{\partial w_3}{\partial z} \right) \\
\tau_{zz} &= \mu \frac{2}{3} \left(2 \frac{\partial w_3}{\partial z} - \frac{\partial w_1}{\partial x} - \frac{\partial w_2}{\partial y} \right) \\
\tau_{xy} = \tau_{yx} &= \mu \left(\frac{\partial w_1}{\partial y} + \frac{\partial w_2}{\partial x} \right) \\
\tau_{xz} = \tau_{zx} &= \mu \left(\frac{\partial w_1}{\partial z} + \frac{\partial w_3}{\partial x} \right) \\
\tau_{yz} = \tau_{zy} &= \mu \left(\frac{\partial w_2}{\partial z} + \frac{\partial w_3}{\partial y} \right)
\end{aligned} \tag{B.7}$$

where μ is the dynamic viscosity. The external forces, \vec{f}_e , are assumed to be zero for the derivation. The Coriolis force per unit mass is given by

$$\vec{f}_{cor} = -2 \left(\vec{\omega} \times \vec{W} \right) = -2 \begin{vmatrix} i & j & k \\ \omega_x & 0 & 0 \\ w_1 & w_2 & w_3 \end{vmatrix} = \begin{bmatrix} 0 \\ 2w_3\omega_x \\ -2w_2\omega_x \end{bmatrix}.$$

The centrifugal force per unit mass is given by

$$\vec{f}_{cen} = -\vec{\omega} \times (\vec{\omega} \times \vec{r}) = - \begin{vmatrix} i & j & k \\ \omega_x & 0 & 0 \\ 0 & -z\omega_x & y\omega_x \end{vmatrix} = \begin{bmatrix} 0 \\ y\omega_x^2 \\ z\omega_x^2 \end{bmatrix}.$$

Combining the forces acting on the system into a single term results in

$$\vec{f}_r = \begin{bmatrix} 0 \\ y\omega_x^2 + 2w_3\omega_x \\ z\omega_x^2 - 2w_2\omega_x \end{bmatrix} \tag{B.8}$$

Using B.8 with B.6 and applying the divergence theorem yields the differential form of the momentum conservation equations, i.e.,

$$\begin{aligned} \int_{\Omega} \frac{\partial(\rho\vec{W})}{\partial t} d\Omega + \int_{\Omega} \nabla \cdot (\rho\vec{W} \otimes \vec{W}) d\Omega &= \int_{\Omega} \rho\vec{f}_r d\Omega - \int_{\Omega} \nabla \cdot p\bar{\mathbf{I}} d\Omega \\ &= \int_{\Omega} \nabla \cdot \bar{\tau} d\Omega \\ \int_{\Omega} \left[\frac{\partial(\rho\vec{W})}{\partial t} + \nabla \cdot (\rho\vec{W} \otimes \vec{W} + p\bar{\mathbf{I}} - \bar{\tau}) - \rho\vec{f}_r \right] d\Omega &= 0 \\ \frac{\partial(\rho\vec{W})}{\partial t} + \nabla \cdot (\rho\vec{W} \otimes \vec{W} + p\bar{\mathbf{I}} - \bar{\tau}) &= \rho\vec{f}_r. \end{aligned}$$

Substituting the components of B.3 into B.7 results in the shear stress tensor components written in terms of absolute velocities. The form of the transformed equations is identical to the original form only with w_i replaced by u_i , where $i = 1, 2, 3$. This is proven through the differentiation of the transport velocity, i.e.,

$$\begin{aligned} \frac{\partial\vec{V}}{\partial x} &= \frac{\partial v_2}{\partial y} = \frac{\partial v_3}{\partial z} = 0 \\ \frac{\partial v_2}{\partial z} + \frac{\partial v_3}{\partial y} &= \omega_x - \omega_x = 0. \end{aligned}$$

B.2.a. X-Momentum Equation

The differential form of the x-momentum equation in the rotating reference frame is given by

$$\frac{\partial(\rho w_1)}{\partial t} + \frac{\partial(\rho w_1 w_1)}{\partial x} + \frac{\partial(\rho w_1 w_2)}{\partial y} + \frac{\partial(\rho w_1 w_3)}{\partial z} + \frac{\partial p}{\partial x} - \tau_1 = 0$$

where $\tau_1 = (\nabla \cdot \bar{\tau})\hat{i}$. Equation B.3 is used with each term to cast the equation in terms of absolute velocity. Applying the velocity transformation gives

$$\begin{aligned} \frac{\partial(\rho u_1)}{\partial t} + \frac{\partial(\rho u_1 u_1)}{\partial x} + \frac{\partial(\rho u_1(u_2 + z\omega_x))}{\partial y} + \frac{\partial(\rho u_1(u_3 - y\omega_x))}{\partial z} + \frac{\partial p}{\partial x} - \tau_1 &= 0 \\ \frac{\partial(\rho u_1)}{\partial t} + \frac{\partial(\rho u_1 w_1)}{\partial x} + \frac{\partial(\rho u_1 w_2)}{\partial y} + \frac{\partial(\rho u_1 w_3)}{\partial z} + \frac{\partial p}{\partial x} - \tau_1 &= 0. \quad (\text{B.9}) \end{aligned}$$

B.2.b. *Y-Momentum Equation*

The differential form of the y-momentum equation in the rotating reference frame is given by

$$\frac{\partial(\rho w_2)}{\partial t} + \frac{\partial(\rho w_2 w_1)}{\partial x} + \frac{\partial(\rho w_2 w_2)}{\partial y} + \frac{\partial(\rho w_2 w_3)}{\partial z} + \frac{\partial p}{\partial y} - \tau_2 = \rho (y\omega_x^2 + 2w_3\omega_x)$$

where $\tau_2 = (\nabla \cdot \bar{\tau})\hat{j}$. Applying B.3 term by term in the differential equation gives

$$\frac{\partial(\rho w_2)}{\partial t} = \frac{\partial(\rho u_2)}{\partial t} + \frac{\partial(\rho z\omega_x)}{\partial t},$$

$$\begin{aligned} \frac{\partial(\rho w_2 w_1)}{\partial x} &= \frac{\partial(\rho(u_2 + z\omega_x)u_1)}{\partial x} \\ &= \frac{\partial(\rho u_2 u_1)}{\partial x} + \frac{\partial(\rho u_1 z\omega_x)}{\partial x} \\ &= \frac{\partial(\rho u_2 u_1)}{\partial x} + z\omega_x \frac{\partial(\rho u_1)}{\partial x}, \end{aligned}$$

$$\begin{aligned} \frac{\partial(\rho w_2 w_2)}{\partial y} &= \frac{\partial(\rho(u_2 + z\omega_x)^2)}{\partial y} \\ &= \frac{\partial(\rho(u_2 u_2 + 2z\omega_x u_2 + (z\omega_x)^2))}{\partial y} \\ &= \frac{\partial(\rho u_2 u_2)}{\partial y} + 2z\omega_x \frac{\partial(\rho u_2)}{\partial y} + (z\omega_x)^2 \frac{\partial \rho}{\partial y}, \end{aligned}$$

$$\begin{aligned} \frac{\partial(\rho w_2 w_3)}{\partial z} &= \frac{\partial(\rho(u_2 + z\omega_x)(u_3 - y\omega_x))}{\partial z} \\ &= \frac{\partial(\rho(u_2 u_3 + u_3 z\omega_x - u_2 y\omega_x - yz\omega_x^2))}{\partial z} \\ &= \frac{\partial(\rho u_2 u_3)}{\partial z} + \frac{\partial(\rho u_3 z\omega_x)}{\partial z} - \frac{\partial(\rho u_2 y\omega_x)}{\partial z} - \frac{\partial(\rho yz\omega_x^2)}{\partial z} \\ &= \frac{\partial(\rho u_2 u_3)}{\partial z} + z\omega_x \frac{\partial(\rho u_3)}{\partial z} + \rho u_3 \omega_x - y\omega_x \frac{\partial(\rho u_2)}{\partial z} - \rho y\omega_x^2 \\ &\quad - z\omega_x \frac{\partial(\rho y\omega_x)}{\partial z}, \end{aligned}$$

and

$$\begin{aligned} \rho (y\omega_x^2 + 2w_3\omega_x) &= \rho y\omega_x^2 + 2\rho\omega_x(u_3 - y\omega_x) \\ &= \rho y\omega_x^2 + 2\rho\omega_x u_3 - 2\rho y\omega_x^2 \\ &= 2\rho\omega_x u_3 - \rho y\omega_x^2. \end{aligned}$$

Rewriting the differential equation with the transformed terms and noting term cancellations gives

$$\begin{aligned} \frac{\partial(\rho u_2)}{\partial t} + z\omega_x \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_2 u_1)}{\partial x} + z\omega_x \frac{\partial(\rho u_1)}{\partial x} + \frac{\partial(\rho u_2 u_2)}{\partial y} + 2z\omega_x \frac{\partial(\rho u_2)}{\partial y} \\ + (z\omega_x)^2 \frac{\partial \rho}{\partial y} + \frac{\partial(\rho u_2 u_3)}{\partial z} + z\omega_x \frac{\partial(\rho u_3)}{\partial z} + \cancel{\rho u_3 \omega_x} - y\omega_x \frac{\partial(\rho u_2)}{\partial z} - \cancel{\rho y \omega_x^2} \\ - z\omega_x \frac{\partial(\rho y \omega_x)}{\partial z} + \frac{\partial p}{\partial y} - \tau_2 = \cancel{2\rho u_3 \omega_x} - \cancel{\rho y \omega_x^2}. \end{aligned}$$

Intelligently grouping the $z\omega_x$ terms together yields

$$\begin{aligned} \frac{\partial(\rho u_2)}{\partial t} + \frac{\partial(\rho u_2 u_1)}{\partial x} + \frac{\partial(\rho u_2 u_2)}{\partial y} + \frac{\partial(\rho u_2 u_3)}{\partial z} + \frac{\partial p}{\partial y} - \tau_2 + z\omega_x \left[\frac{\partial \rho}{\partial t} \right. \\ \left. + \frac{\partial(\rho u_1)}{\partial x} + \frac{\partial(\rho u_2)}{\partial y} + \frac{\partial(\rho u_3)}{\partial z} \right] + z\omega_x \frac{\partial(\rho u_2)}{\partial y} - y\omega_x \frac{\partial(\rho u_2)}{\partial z} + (z\omega_x)^2 \frac{\partial \rho}{\partial y} \\ - z\omega_x \frac{\partial(\rho y \omega_x)}{\partial z} = \rho u_3 \omega_x \end{aligned}$$

where the bracketed terms are the left hand side of the continuity equation. Substituting B.5 and observing the cancellations due to the substitution results in

$$\begin{aligned} \frac{\partial(\rho u_2)}{\partial t} + \frac{\partial(\rho u_2 u_1)}{\partial x} + \frac{\partial(\rho u_2 u_2)}{\partial y} + \frac{\partial(\rho u_2 u_3)}{\partial z} + \frac{\partial p}{\partial y} + z\omega_x \left[\frac{\partial(\rho y \omega_x)}{\partial z} - z\omega_x \frac{\partial \rho}{\partial y} \right] \\ + z\omega_x \frac{\partial(\rho u_2)}{\partial y} - y\omega_x \frac{\partial(\rho u_2)}{\partial z} + \cancel{(z\omega_x)^2 \frac{\partial \rho}{\partial y}} - \cancel{z\omega_x \frac{\partial(\rho y \omega_x)}{\partial z}} = \rho u_3 \omega_x. \end{aligned}$$

Combining the remaining $\partial/\partial y$ and $\partial/\partial z$ derivatives and using B.3 yields the final form of the equation, i.e.,

$$\frac{\partial(\rho u_2)}{\partial t} + \frac{\partial(\rho u_2 w_1)}{\partial x} + \frac{\partial(\rho u_2 w_2)}{\partial y} + \frac{\partial(\rho u_2 w_3)}{\partial z} + \frac{\partial p}{\partial y} - \tau_2 = \rho u_3 \omega_x. \quad (\text{B.10})$$

B.2.c. Z-Momentum Equation

The differential form of the z-momentum equation in the rotating reference frame is

$$\frac{\partial(\rho w_3)}{\partial t} + \frac{\partial(\rho w_3 w_1)}{\partial x} + \frac{\partial(\rho w_3 w_2)}{\partial y} + \frac{\partial(\rho w_3 w_3)}{\partial z} + \frac{\partial p}{\partial z} - \tau_3 = \rho (z\omega_x^2 - 2\rho w_2 \omega_x)$$

where $\tau_3 = (\nabla \cdot \bar{\tau})\hat{k}$. Applying B.3 term by term to the given differential equation gives

$$\frac{\partial(\rho w_3)}{\partial t} = \frac{\partial(\rho u_3)}{\partial t} + \frac{\partial(\rho y \omega_x)}{\partial t},$$

$$\begin{aligned} \frac{\partial(\rho w_3 w_1)}{\partial x} &= \frac{\partial(\rho(u_3 - y\omega_x)u_1)}{\partial x} \\ &= \frac{\partial(\rho u_3 u_1)}{\partial x} - \frac{\partial(\rho u_1 y \omega_x)}{\partial x} \\ &= \frac{\partial(\rho u_3 u_1)}{\partial x} - y\omega_x \frac{\partial(\rho u_1)}{\partial x}, \end{aligned}$$

$$\begin{aligned} \frac{\partial(\rho w_3 w_2)}{\partial y} &= \frac{\partial(\rho(u_3 - y\omega_x)(u_2 + z\omega_x))}{\partial y} \\ &= \frac{\partial(\rho(u_3 u_2 + u_3 z \omega_x - u_2 y \omega_x - y z \omega_x^2))}{\partial y} \\ &= \frac{\partial(\rho u_3 u_2)}{\partial y} + \frac{\partial(\rho u_3 z \omega_x)}{\partial y} - \frac{\partial(\rho u_2 y \omega_x)}{\partial y} - \frac{\partial(\rho y z \omega_x^2)}{\partial y} \\ &= \frac{\partial(\rho u_3 u_2)}{\partial y} + z\omega_x \frac{\partial(\rho u_3)}{\partial y} - y\omega_x \frac{\partial(\rho u_2)}{\partial y} - \rho u_2 \omega_x - y\omega_x \frac{\partial(\rho z \omega_x)}{\partial y} - \rho z \omega_x^2, \end{aligned}$$

and

$$\begin{aligned} \frac{\partial(\rho w_3 w_3)}{\partial z} &= \frac{\partial(\rho(u_3 - y\omega_x)^2)}{\partial z} \\ &= \frac{\partial(\rho(u_3 u_3 - 2u_3 y \omega_x + (y\omega_x)^2))}{\partial z} \\ &= \frac{\partial(\rho u_3 u_3)}{\partial z} - 2y\omega_x \frac{\partial(\rho u_3)}{\partial z} + (y\omega_x)^2 \frac{\partial \rho}{\partial z}. \end{aligned}$$

Rewriting the differential equation with the transformed terms and noting term cancellation gives

$$\begin{aligned} \frac{\partial(\rho u_3)}{\partial t} - y\omega_x \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_3 u_1)}{\partial x} - y\omega_x \frac{\partial(\rho u_1)}{\partial x} + \frac{\partial(\rho u_3 u_2)}{\partial y} + z\omega_x \frac{\partial(\rho u_3)}{\partial y} \\ - y\omega_x \frac{\partial(\rho u_2)}{\partial y} - \cancel{\rho u_2 \omega_x} - y\omega_x \frac{\partial(\rho z \omega_x)}{\partial y} - \cancel{\rho z \omega_x^2} + \frac{\partial(\rho u_3 u_3)}{\partial z} - 2y\omega_x \frac{\partial(\rho u_3)}{\partial z} \\ + (y\omega_x)^2 \frac{\partial \rho}{\partial z} + \frac{\partial p}{\partial z} - \tau_3 = \cancel{2\rho u_2 \omega_x} - \cancel{\rho z \omega_x^2}. \end{aligned}$$

Rearranging the differential equation and intelligently grouping the $y\omega_x$ terms gives

$$\begin{aligned} \frac{\partial(\rho u_3)}{\partial t} + \frac{\partial(\rho u_3 u_1)}{\partial x} + \frac{\partial(\rho u_3 u_2)}{\partial y} + \frac{\partial(\rho u_3 u_3)}{\partial z} + \frac{\partial p}{\partial z} - \tau_3 - y\omega_x \left[\frac{\partial \rho}{\partial t} \right. \\ \left. + \frac{\partial(\rho u_1)}{\partial x} + \frac{\partial(\rho u_2)}{\partial y} + \frac{\partial(\rho u_3)}{\partial z} \right] + z\omega_x \frac{\partial(\rho u_3)}{\partial y} - y\omega_x \frac{\partial(\rho z\omega_x)}{\partial y} \\ - y\omega_x \frac{\partial(\rho u_3)}{\partial z} + (y\omega_x)^2 \frac{\partial \rho}{\partial z} = -\rho u_2 \omega_x \end{aligned}$$

where the terms in brackets are the left hand side of the continuity equation. Substituting B.5 and observing the resultant cancellations yields

$$\begin{aligned} \frac{\partial(\rho u_3)}{\partial t} + \frac{\partial(\rho u_3 u_1)}{\partial x} + \frac{\partial(\rho u_3 u_2)}{\partial y} + \frac{\partial(\rho u_3 u_3)}{\partial z} + \frac{\partial p}{\partial z} - \tau_3 - y\omega_x \left[\frac{\partial(\rho y\omega_x)}{\partial z} - \frac{\partial(\rho z\omega_x)}{\partial y} \right] \\ + z\omega_x \frac{\partial(\rho u_3)}{\partial y} - y\omega_x \frac{\partial(\rho z\omega_x)}{\partial y} - y\omega_x \frac{\partial(\rho u_3)}{\partial z} + (y\omega_x)^2 \frac{\partial \rho}{\partial z} = -\rho u_2 \omega_x. \end{aligned}$$

Combining the remaining $\partial/\partial y$ and $\partial/\partial z$ terms and using B.3 results in the final form of the equation, i.e.,

$$\frac{\partial(\rho u_3)}{\partial t} + \frac{\partial(\rho u_3 w_1)}{\partial x} + \frac{\partial(\rho u_3 w_2)}{\partial y} + \frac{\partial(\rho u_3 w_3)}{\partial z} + \frac{\partial p}{\partial z} - \tau_3 = -\rho u_2 \omega_x. \quad (\text{B.11})$$

B.2.d. Final Integral Form

The integral form of the momentum conservation equations is obtained by applying the divergence theorem to the vector form of the derived equations. Combining B.9–B.11 results in

$$\frac{\partial(\rho \vec{U})}{\partial t} + \nabla \cdot (\rho \vec{U} \otimes \vec{W} + p \vec{\mathbb{I}} - \vec{\tau}) = \rho \vec{f}_a \quad (\text{B.12})$$

where $\vec{f}_a = [0, 0, u_3\omega_x, -u_2\omega_x, 0]^T$. Applying the divergence theorem to B.12 results in the integral form of the equation, i.e.,

$$\begin{aligned} \frac{\partial}{\partial t} \int_{\Omega} \rho \vec{U} d\Omega + \oint_{\partial\Omega} \rho \vec{U} (\vec{W} \cdot \hat{n}) dS = \int_{\Omega} \rho \vec{f}_a d\Omega - \oint_{\partial\Omega} p (\vec{\mathbb{I}} \cdot \hat{n}) dS \\ + \oint_{\partial\Omega} (\vec{\tau} \cdot \hat{n}) dS. \end{aligned} \quad (\text{B.13})$$

B.3. Conservation of Energy

The integral form of the energy conservation equation in the rotating reference frame is given by

$$\frac{\partial}{\partial t} \int_{\Omega} \rho E_r d\Omega + \oint_{\partial\Omega} \rho I \vec{W} \cdot \hat{n} dS = \oint_{\partial\Omega} \left(k \nabla T + \bar{\tau} \vec{W} \right) \cdot \hat{n} dS, \quad (\text{B.14})$$

where E_r is the relative total energy and I is rothalpy. The relative total energy is defined as [Blazek, 2005, p. 438]

$$E_r = e + \frac{1}{2} |\vec{W}|^2 - \frac{1}{2} |\vec{V}|^2, \quad (\text{B.15})$$

and the rothalpy is given by [Blazek, 2005, p. 439]

$$I = h + \frac{1}{2} |\vec{W}|^2 - \frac{1}{2} |\vec{V}|^2. \quad (\text{B.16})$$

Applying the divergence theorem to B.14 results in the differential form of the equation,

$$\frac{\partial (\rho E_r)}{\partial t} + \nabla \cdot (\rho I \vec{W}) - \nabla \cdot (k \nabla T + \bar{\tau} \vec{W}) = 0. \quad (\text{B.17})$$

Focus is first applied to the divergence of the product of shear stress and relative velocity. Applying the chain rule, we obtain

$$\nabla \cdot (\bar{\tau} \vec{W}) = \nabla \bar{\tau} \cdot \vec{W} + \bar{\tau} \cdot \nabla \vec{W}.$$

The second term on the right hand side of the above equation is equal to zero due to the conservation of mass. Substituting Eq. B.3, and carrying out the algebra gives

$$\begin{aligned} \nabla \cdot (\bar{\tau} \vec{W}) &= \nabla \bar{\tau} \cdot \vec{U} - \nabla \bar{\tau} \cdot \vec{V} \\ &= \left(\nabla \bar{\tau} \cdot \vec{U} + \bar{\tau} \cdot \nabla \vec{U} \right) - \nabla \bar{\tau} \cdot \vec{V} \\ &= \nabla \cdot (\bar{\tau} \vec{U}) + z \omega_x \tau_2 - y \omega_x \tau_3. \end{aligned}$$

Here, as before, $\tau_2 = (\nabla \bar{\tau}) \cdot \hat{j}$ and $\tau_3 = (\nabla \bar{\tau}) \cdot \hat{k}$. Substituting the above result back into Eq. B.17 gives

$$\frac{\partial (\rho E_r)}{\partial t} + \nabla \cdot (\rho I \vec{W}) - \nabla \cdot (k \nabla T + \bar{\tau} \vec{U}) - z \omega_x \tau_2 + y \omega_x \tau_3 = 0. \quad (\text{B.18})$$

Next, the time derivative of the relative total energy is transformed. Substituting Eq. B.3 into the definition of the relative total energy, Eq. B.15, gives

$$\begin{aligned}
E_r &= e + \frac{1}{2} [\vec{W} \cdot \vec{W} - \vec{V} \cdot \vec{V}] \\
&= e + \frac{1}{2} [(\vec{U} - \vec{V}) \cdot (\vec{U} - \vec{V}) - \vec{V} \cdot \vec{V}] \\
&= e + \frac{1}{2} [\vec{U} \cdot \vec{U} + \vec{V} \cdot \vec{V} - 2\vec{U} \cdot \vec{V} - \vec{V} \cdot \vec{V}] \\
&= E - \vec{U} \cdot \vec{V}.
\end{aligned}$$

Using the above result, the time derivative of the relative total energy can be written as

$$\begin{aligned}
\frac{\partial(\rho E_r)}{\partial t} &= \frac{\partial}{\partial t} [\rho (E - \vec{U} \cdot \vec{V})] \\
&= \frac{\partial(\rho E)}{\partial t} + \frac{\partial}{\partial t} [\rho (u_2 z \omega_x - u_3 y \omega_x)] \\
&= \frac{\partial(\rho E)}{\partial t} + z \omega_x \frac{\partial(\rho u_2)}{\partial t} - y \omega_x \frac{\partial(\rho u_3)}{\partial t}.
\end{aligned}$$

Finally, the velocity in the convective flux term is transformed. The rothalpy can be written in terms of absolute velocities as

$$\begin{aligned}
I &= h + \frac{1}{2} (\vec{W} \cdot \vec{W} - \vec{V} \cdot \vec{V}) \\
&= H - \vec{U} \cdot \vec{V} \\
&= H + u_2 z \omega_x - u_3 y \omega_x
\end{aligned}$$

Applying the above result with Eq. B.3 to each component of the convective flux term gives

$$\begin{aligned}
\frac{\partial(\rho I w_1)}{\partial x} &= \frac{\partial}{\partial x} [\rho (H + u_2 z \omega_x - u_3 y \omega_x) u_1] \\
&= \frac{\partial(\rho H u_1)}{\partial x} + z \omega_x \frac{\partial(\rho u_1 u_2)}{\partial x} - y \omega_x \frac{\partial(\rho u_1 u_3)}{\partial x},
\end{aligned}$$

$$\begin{aligned}
\frac{\partial(\rho I w_2)}{\partial y} &= \frac{\partial}{\partial y} [\rho I u_2 + \rho I z \omega_x] \\
&= \frac{\partial}{\partial y} [\rho (H + u_2 z \omega_x - u_3 y \omega_x) u_2 + \rho (H + u_2 z \omega_x - u_3 y \omega_x) z \omega_x] \\
&= \frac{\partial}{\partial y} [\rho H (u_2 + z \omega_x)] + z \omega_x \frac{\partial}{\partial y} [\rho u_2 (u_2 + z \omega_x)] - y \omega_x \frac{\partial}{\partial y} [\rho u_3 (u_2 + z \omega_x)] \\
&\quad - \rho u_3 u_2 \omega_x - \rho u_3 z \omega_x^2 \\
&= \frac{\partial(\rho H w_2)}{\partial y} + z \omega_x \frac{\partial(\rho u_2 w_2)}{\partial y} - y \omega_x \frac{\partial(\rho u_3 w_2)}{\partial y} - \rho u_3 u_2 \omega_x - \rho u_3 z \omega_x^2,
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial(\rho I w_3)}{\partial z} &= \frac{\partial}{\partial z} [\rho I u_3 - \rho I y \omega_x] \\
&= \frac{\partial}{\partial z} [\rho (H + u_2 z \omega_x - u_3 y \omega_x) u_3 - \rho (H + u_2 z \omega_x - u_3 y \omega_x) y \omega_x] \\
&= \frac{\partial}{\partial z} [\rho H (u_3 - y \omega_x)] + z \omega_x \frac{\partial}{\partial z} [\rho u_2 (u_3 - y \omega_x)] - y \omega_x \frac{\partial}{\partial z} [\rho u_3 (u_3 - y \omega_x)] \\
&\quad + \rho u_2 u_3 \omega_x - \rho u_2 y \omega_x^2 \\
&= \frac{\partial(\rho H w_3)}{\partial z} + z \omega_x \frac{\partial(\rho u_2 w_3)}{\partial z} - y \omega_x \frac{\partial(\rho u_3 w_3)}{\partial z} + \rho u_2 u_3 \omega_x - \rho u_2 y \omega_x^2.
\end{aligned}$$

Substituting the transformed time and convective derivatives into Eq. B.18 and combining like terms gives

$$\begin{aligned}
&\frac{\partial(\rho E)}{\partial t} + \frac{\partial(\rho H w_1)}{\partial x} + \frac{\partial(\rho H w_2)}{\partial y} + \frac{\partial(\rho H w_3)}{\partial z} - \nabla \cdot (k \nabla T + \bar{\tau} \vec{U}) + z \omega_x \left[\frac{\partial(\rho u_2)}{\partial t} \right. \\
&\quad \left. + \frac{\partial(\rho u_2 w_1)}{\partial x} + \frac{\partial(\rho u_2 w_2)}{\partial y} + \frac{\partial(u_2 w_3)}{\partial z} - \tau_2 - \rho u_3 \omega_x \right] - y \omega_x \left[\frac{\partial(\rho u_3)}{\partial t} \right. \\
&\quad \left. + \frac{\partial(\rho u_3 w_1)}{\partial x} + \frac{\partial(\rho u_3 w_2)}{\partial y} + \frac{\partial(\rho u_3 w_3)}{\partial z} - \tau_3 + \rho u_2 \omega_x \right] = 0.
\end{aligned}$$

It is observed that the second and third set of bracketed terms in the above equation are the partial forms of the y-momentum and z-momentum equations, respectively. Substituting Eq. B.10 and Eq. B.11 into the above equation results in

$$\frac{\partial(\rho E)}{\partial t} + \nabla \cdot (\rho H \vec{W}) - \nabla \cdot (k \nabla T + \bar{\tau} \vec{U}) + \nabla \cdot (p \vec{V}) = 0. \quad (\text{B.19})$$

The integral form of the equation can be recovered by applying the divergence the-

orem, giving

$$\frac{\partial}{\partial t} \int_{\Omega} \rho E d\Omega + \oint_{\partial\Omega} (\rho H \vec{W} + p \vec{V}) \cdot \hat{n} dS = \oint_{\partial\Omega} (k \nabla T + \bar{\tau} \vec{U}) \cdot \hat{n} dS. \quad (\text{B.20})$$

APPENDIX C

ANALYTIC NOZZLE SOLUTION

This appendix presents the analytic solution for a convergent-divergent nozzle. The methodologies for both a purely subsonic flow and for a flow in which a normal shock sits in the diffuser are given. Anderson [1995, see Ch. 7], was used as the source for this material. Consult Fig. C.1 for the definitions of the variables used herein.

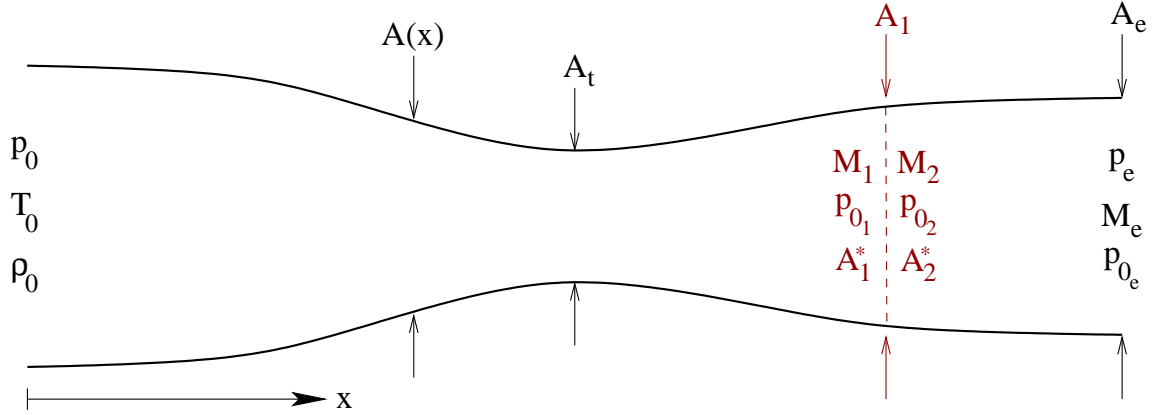


Figure C.1: Defining the convergent-divergent nozzle. Normal shock quantities are given in maroon, location given by dashed line.

It is assumed that the flow in the nozzle is steady, isentropic, and quasi-one-dimensional. Using these assumptions the Navier–Stokes equations can be simplified and analytically solved. Equation (C.1), which relates Mach number to only nozzle area, is the result [Anderson, 1995, p. 286]. Equations (C.2)–(C.4) are then used to find pressure, density, and temperature once the Mach number is known. A^* in (C.1)

is the critical area at which the flow is sonic, that is to say $M = 1$ when $A/A^* = 1$.

$$\left(\frac{A}{A^*}\right)^2 = \frac{1}{M^2} \left[\frac{2}{\gamma+1} \left(1 + \frac{\gamma-1}{2} M^2 \right) \right]^{(\gamma+1)/(\gamma-1)}. \quad (\text{C.1})$$

$$\frac{p}{p_0} = \left(1 + \frac{\gamma-1}{2} M^2 \right)^{-\gamma/(\gamma-1)} \quad (\text{C.2})$$

$$\frac{\rho}{\rho_0} = \left(1 + \frac{\gamma-1}{2} M^2 \right)^{-1/(\gamma-1)} \quad (\text{C.3})$$

$$\frac{T}{T_0} = \left(1 + \frac{\gamma-1}{2} M^2 \right)^{-1} \quad (\text{C.4})$$

The flow within the nozzle is determined by the choice of p_e/p_0 . Ratios near unity result in the flow being purely subsonic. Reducing the exit pressure results in the flow becoming choked, with sonic flow at the throat and supersonic flow downstream. In the choked case, the value of p_e/p_0 will determine whether the flow downstream of the throat will be purely supersonic, or if a normal shock will be formed to facilitate the necessary pressure rise to obtain the desired exit pressure.

C.1. Pure Subsonic Case

This section details the steps required to find the analytic nozzle solution for a purely subsonic case. For the flow to be purely subsonic, the critical area must be less than the throat area, A_t . In this case, the critical area is determined from the specified p_e/p_0 . The steps for finding the critical area and the analytic solution are as follows:

1. Compute the exit Mach number, M_e , using Eq. C.2, i.e.

$$\frac{p_e}{p_0} = \left(1 + \frac{\gamma-1}{2} M_e^2 \right)^{-\gamma/(\gamma-1)}$$

2. Compute the critical area using the known exit conditions and Eq. C.1, i.e.

$$\left(\frac{A_e}{A^*}\right)^2 = \frac{1}{M_e^2} \left[\frac{2}{\gamma+1} \left(1 + \frac{\gamma-1}{2} M_e^2 \right) \right]^{(\gamma+1)/(\gamma-1)}.$$

3. Once the critical area is known, Eq. C.1–C.4 can be used directly to compute the variation of the Mach number and the thermodynamic properties across the nozzle.

C.2. Normal Shock Case

This section details the steps necessary to find the analytic solution for the case in which the specified p_e/p_0 results in a normal shock sitting in the diffuser. For this case we know the following:

- The shock location, found at $A(x_s) = A_1$, is decided by the value of p_e/p_0 .
- The flow from the inlet to just upstream of the shock is isentropic, i.e. $p_{0_1} = p_0$.
- The critical area upstream of the shock is constant and equal to the throat area, $A_1^* = A_t$.
- The subsonic region downstream of the shock is isentropic such that $p_{0_e} = p_{0_2}$.
- The critical area downstream of the shock is similar to the critical area for the pure subsonic case as it is a reference value that must be computed.
- Total temperature and mass flow rate are constant through a normal shock. For the nozzle, the mass flow is given by [Anderson, 1995, p.359]

$$\dot{m} = \frac{p_0 A^*}{\sqrt{T_0}} \sqrt{\frac{\gamma}{R} \left(\frac{2}{\gamma + 1} \right)^{(\gamma+1)/(\gamma-1)}} \quad (\text{C.5})$$

Finding the location of the shock is first step in computing the analytic solution. The steps necessary to compute the shock location are:

1. Compute the exit Mach number using the fact that mass flow rate and total temperature are constant across a normal shock. Equating (C.5) for just upstream and downstream of the shock gives $p_{0_1} A_1^* = p_{0_2} A_2^*$. Using this equality,

we can write

$$\frac{p_e}{p_{0_2}} \frac{A_e}{A_2^*} = \frac{p_e}{p_{0_1}} \frac{A_e}{A_1^*} = \frac{p_e}{p_{0_1}} \frac{A_e}{A_t},$$

where the far right hand side is known from the given nozzle pressure ratio and the geometry. The exit Mach number can then be found after multiplying (C.1) and (C.2) together to replace the left hand side of the above equation, i.e. [Anderson, 1995, p.360]

$$\frac{p_e}{p_{0_2}} \frac{A_e}{A_2^*} \equiv \frac{1}{M_e} \left(\frac{2}{\gamma + 1} \right)^{(\gamma+1)/2(\gamma-1)} \left(1 + \frac{\gamma - 1}{2} M_e^2 \right)^{-1/2} = \frac{p_e}{p_{0_1}} \frac{A_e}{A_t}$$

2. Once the exit Mach number is known, the total pressure ratio across the normal shock can be computed using

$$\frac{p_{0,2}}{p_{0,1}} = \frac{p_{0,2}}{p_e} \frac{p_e}{p_{0,1}},$$

where $p_e/p_{0,2}$ is given by Eq. C.2 with $M = M_e$.

3. Using the total pressure ratio, the Mach number just upstream of the shock, M_1 , is computed using the normal shock relation [Staff, 1953]

$$\frac{p_{0,2}}{p_{0,1}} = \left[\frac{(\gamma + 1) M_1^2}{(\gamma - 1) M_1^2 + 2} \right]^{\gamma/(\gamma-1)} \left[\frac{\gamma + 1}{2\gamma M_1^2 - (\gamma - 1)} \right]^{1(\gamma-1)} \quad (\text{C.6})$$

4. The location of the shock, given by A_1 , can be directly computed using (C.1) with $M = M_1$ and $A^* = A_t$.

Computing the analytic solution from the inlet to just upstream of the normal shock is straightforward once the location of the shock is known. The Mach number in the nozzle is given by (C.1), recalling that $A_1^* = A_t$. The thermodynamic properties are then given by (C.2)–(C.4).

The critical area for the subsonic region downstream of the normal shock must be computed before computing the analytic solution for this region. The critical area is computed in the same manner as was used for the purely subsonic case.

Equation (C.1) is used with the already computed exit Mach number to find the critical area, namely

$$\left(\frac{A_e}{A_2^*}\right)^2 = \frac{1}{M_e^2} \left[\frac{2}{\gamma+1} \left(1 + \frac{\gamma-1}{2} M_e^2\right) \right]^{(\gamma+1)/(\gamma-1)}.$$

With the critical area known, the Mach number and thermodynamic properties can then be computed using (C.1)–(C.4).

At this point we have an analytic solution that is based on two sets of total conditions: p_0 upstream of the shock, and p_{0_e} downstream of the shock. What we desire is a solution based solely on the inlet total conditions. To achieve this, (C.2) is multiplied by p_{0_2}/p_{0_1} , and (C.3) is multiplied by ρ_{0_2}/ρ_{0_1} . By equating mass flow rates, the total pressure ratio can be expressed as

$$\frac{p_{0_2}}{p_{0_1}} = \frac{A_1^*}{A_2^*} = \frac{A_t}{A_2^*}.$$

Equation (C.3) is used to define the ratio of total densities:

$$\frac{\rho_{0_2}}{\rho_{0_1}} = \frac{\rho_2 \left(1 + \frac{\gamma-1}{2} M_2^2\right)^{1/(\gamma-1)}}{\rho_1 \left(1 + \frac{\gamma-1}{2} M_1^2\right)^{1/(\gamma-1)}}$$

The ratio of static densities in the above equation can be replaced using a normal shock relation [Staff, 1953] to yield

$$\frac{\rho_{0_2}}{\rho_{0_1}} = \frac{(\gamma+1) M_1^2}{(\gamma-1) M_1^2 + 2} \frac{\left(1 + \frac{\gamma-1}{2} M_2^2\right)^{1/(\gamma-1)}}{\left(1 + \frac{\gamma-1}{2} M_1^2\right)^{1/(\gamma-1)}}.$$

APPENDIX D

UNS3D INPUT FILES

The purpose of this appendix is to provide future users of UNS3D with a good starting point for simulations of the same or similar geometry found in this work. One input file is given per case, and any input parameters that were varied in this work will be highlighted. The UNS3D User's Manual [Cizmas, 2014] should be consulted for the definitions of any input parameters found herein. Any input parameter not specified is assumed to be assigned the default value by UNS3D.

One new addition to UNS3D that may or may not have been added to the user's manual at the time of this reading is the input parameter `typlim_opt`. This parameter controls which definition of ϵ^2 is used for the Venkatakrisshnan based limiters. The default value for the input parameter is 0. The acceptable values for `typlim_opt` for the limiters are summarized in Table D.1.

A sample batch file for an MPI job submission on the Ada HPC System at Texas A&M is also given at the end of this appendix. Consult the supercomputing center's website¹ for detailed information on submitting jobs to the desired HPC system.

¹sc.tamu.edu

Table D.1: UNS3D input parameters for the higher order limiter and their options.

typlim	Name	typlim_opt	Option Description
		0	Original ϵ^2 definition by Venkatakrisnan [1995] (Venka- ϵ_1)
2	Venkatakrisnan	1	New ϵ^2 definition by Wang [2000]
		2	New ϵ^2 definition by Park & Kim [2012] (Venka- ϵ_2)
3	Dervieux	0	N/A
		0	Original ϵ^2 definition by Venkatakrisnan [1995] (MLP- ϵ_1)
4	MLP	1	New ϵ^2 definition by Park & Kim [2012] (MLP- ϵ_2)

D.1. Convergent-Divergent Nozzle

The input file given below is for the **purely subsonic** inviscid flow through a convergent-divergent nozzle. The case is second-order accurate using the Venkatakrishnan limiter, specifically Venka- ϵ_1 . Refer to Table D.1 for the appropriate input parameters when changing the limiter type. The only necessary change to the boundary conditions to generate a flow with a normal shock is to decrease the back pressure (`pback`). For the normal shock case described in Section IV.1 (p. 59) use `pback = 78142.9d0`.

```
&cardf
  title          = "Inviscid Convergent-Divergent Nozzle"
  tecplot_name   = "plt/cdnozzle.plt"
  dump_tecplot   = T
/
&cardg
  gridfile       = "mesh/vol.mesh"
  c2nfile        = "mesh/c2n.def"
  l2gfile        = "mesh/loc2glob.dat_001"
/
&cardh
  filedin        = "out/none.out"
  filedout       = "out/run001.out"
/
&cardi
  fileturin      = "out/none.tur"
  fileturout     = "out/none.tur"
/
&cardk
/
&card0
  inbc           = 0
  ioutbc         = 1
  ipex           = 0
  igeom          = 1
  ispet          = 0
  iwall          = 3
  deltafhat      = 0.001
/
&card1
  istep          = 0
```



```

    iturm          = 0
    npseudotimesteps = 30000
    mtime          = 1
/
&card2
    invis         = T
    lamin         = T
    iorder        = 2
    irhsm         = 4
    lsgg          = 2
    mstg          = 4
    typlim        = 2
    typlim_opt    = 0
    fluxtype      = 1
/
&card2a
    imp           = F
/
&card3
    cfl           = 1.0d0
    epss          = 0.5
/
&card4
    u0            = 0.2d0
/
&card5
    ptot          = 104190.585d0
    ttot          = 290.435d0
    pback         = 92729.6d0
/
&card6
    pref          = 101325.0d0
    tref          = 288.15d0
    scale         = 1.0d0
/
&card7
/
&card8
/
&card9
    intev_freq    = 1000
    res_freq      = 200
    itersave      = 1000
    intev_freq_pt = 1000
    res_freq_pt   = 200

```

```

/
&card10
/
&cardforced
/
&cardrom
/
&vortex
/
&cardprecon
/

```

D.2. Supersonic NACA 0012

The input file given below was used to compute the supersonic flow around a NACA 0012 airfoil section. This particular simulation used the MLP- ϵ_1 limiter with the default value of the constant (`venka_c = 5.0d0`). Refer to Table D.1 when setting up simulations with different limiters and limiter options.

The major difference between this input file and all of the other input files in this appendix is the selection of the inlet boundary condition. The supersonic inlet condition (`inbc = 5`) is used here instead of the standard subsonic inlet condition.

```

&cardf
  title      = "Supersonic NACA 0012"
  tecplot_name = "plt/naca.plt"
  dump_tecplot = T
  dump_yplus  = F
  rsdfile     = "txt/001.dat"
/
&cardg
  gridfile   = "mesh/vol.mesh"
  c2nfile    = "mesh/c2n.def"
  l2gfile    = "mesh/loc2glob.dat_002"
/
&cardh
  filedin    = "out/run001.out"
  filedout   = "out/run001.out"
/
&cardi
  fileturin  = "out/none.tur"

```

```

    fileturout = "out/none.tur"
/
&cardk
/
&card0
    noblade      = 1
    inbc         = 5 ! supersonic inlet bc
    ioutbc       = 1
    ipex         = 0
    igeom        = 1
    ispet        = 0
    leak_outlet  = F
    iwall        = 3
    deltafhat    = 0.001
/
&card1
    istep        = 0
    iturm        = 0
    npseudotimesteps = 10000
    mtime        = 1
    iramp0       = 0
    iramp        = 1
    timestep     = 0
    iramp0_leak  = 0
    iramp_leak   = 1
/
&card2
    invis        = T
    lamin        = T
    iorder       = 2
    irhsm        = 4
    lsgg         = 2
    mstg         = 4
    typlim       = 4
    typlim_opt   = 0
    venka_c      = 5.0d0 ! default value
    fluxtype     = 1
/
&card2a
    imp          = .false.
/
&card3
    cfl          = 0.8d0
    epss         = 0.5
/

```

```

&card4
  u0          = 2.0d0
/
&card5
  ptot       = 792812.3d0
  ttot       = 518.9d0
  pback      = 101325.0d0
/
&card6
  pref       = 101325.0d0
  tref       = 288.15d0
  scale      = 1.0d0
/
&card7
/
&card8
/
&card9
  intev_freq = 500
  res_freq   = 50
  itersave   = 1000
  intev_freq_pt = 500
  res_freq_pt = 50
  ires       = 1
  MonitorMaxMach = F
  MaxMachThreshold = 2.5
  reset_iter_counter = T
  q_corrctn_limit = 1.0d-14
  q_corrctn_ratio = 14
/
&card10
/
&cardforced
/
&cardrom
/
&vortex
/
&cardprecon
/

```

D.3. Turbulent Flat Plate

The input file given below was used to compute the subsonic, turbulent flow over a zero pressure gradient flat plate. This particular input file is for the second-order accurate portion of the simulation. The input file can be modified for the first-order start simulation by changing the value of `iorder` from 2 to 1, and by changing `npseudotimesteps` to 1000.

```
&cardf
  title          = "Turbulent Flat Plate"
  tecplot_name   = "plt/fp.plt"
  dump_tecplot   = T
  dump_yplus     = F
  dump_v3        = F
  rsdfile        = "txt/fp.002.dat"
  case_name      = "yp/wc2"
  relative_V     = F
/
&cardg
  gridfile       = "mesh/vol.mesh"
  c2nfile        = "mesh/c2n.def"
  l2gfile        = "mesh/loc2glob.dat_008"
/
&cardh
  filedin        = "out/fp001.out"
  filedout       = "out/fp002.out"
/
&cardi
  fileturin      = "out/fp001.tur"
  fileturout     = "out/fp002.tur"
/
&cardk
/
&card0
  noblade        = 1
  inbc           = 0
  ioutbc         = 1
  ipex           = 0
  igeom          = 1
  ispet          = 0
  leak_outlet    = F
  iwall          = 3
  deltafhat      = 0.001
```

```

/
&card1
  istep          = 1
  iturm          = 1
  npseudotimesteps = 150000
  mtime         = 1
  iramp0        = 0
  iramp         = 1
  timestep      = 0
  iramp0_leak   = 0
  iramp_leak    = 1
/
&card2
  invis         = F
  lamin        = F
  iorder       = 2
  irhsm        = 4
  lsgg        = 2
  mstg         = 4
  typlim       = 2
  typlim_opt   = 0
  fluxtype     = 1
/
&card2a
  imp          = .false.
/
&card3
  cfl          = 1.0d0
  epss         = 0.5
/
&card4
  u0           = 0.2d0
  tintens      = 0.01d0
  tlength      = 0.1d0
/
&card5
  ptot         = 104190.471d0
  ttot         = 290.455d0
  pback        = 101325.0d0
/
&card6
  pref         = 101325.0d0
  tref         = 288.15d0
  scale        = 1.0d0
/

```

```
&card7
/
&card8
/
&card9
  debug          = F
  echo           = F
  intev_freq     = 5000
  res_freq       = 500
  itersave       = 10000
  intev_freq_pt  = 5000
  res_freq_pt    = 500
  ires           = 1
  MonitorMaxMach = .true.
  MaxMachThreshold = 2.5
  reset_iter_counter = F
  Force_l        = .true.
  q_corrctn_limit = 1.0d-9
/
&card10
  resrlx(6) = 1.0d0
  resrlx(7) = 1.0d0
/
&cardforced
/
&cardrom
/
&vortex
/
&cardprecon
/
```

D.4. Goldman Turbine Vane

The input file given in this section was used to compute the baseline flow for the results presented in Chapter V. This particular input file is the final restart input file needed to achieve convergence on the medium grid. As such, this input file is step up for steady, second-order, turbulent flow. Consult the text of Chapter V for a description of the solution strategy, which included both inviscid and laminar simulations for the coarse and medium grids.

```
&cardf
  title          = "Rotor 67: Turbulent, IPEX=0"
  tecplot_name   = "plt/006.plt"
  dump_tecplot   = T
  dump_yplus     = T
  rsdfile        = "txt/006.dat"
  case_name      = "yp/LR67"
  relative_V     = F
/
&cardg
  gridfile       = "mesh/vol.mesh"
  c2nfile        = "mesh/c2n.def"
  XleXteRte     = "mesh/XleXteRte"
  l2gfile        = "mesh/loc2glob.dat_040"
  ijkfile        = "mesh/outlet_ij.def"
/
&cardh
  filedin        = "out/r005.out"
  filedout       = "out/r006.out"
/
&cardi
  fileturin      = "out/sst005.out"
  fileturout     = "out/sst006.out"
/
&cardk
/
&card0
  noblade        = 36
  inbc           = 0
  ioutbc         = 1
  ipex           = 1
  igeom          = 0
  ispet          = 0
```



```

    leak_outlet = F
    iwall       = 3
    deltafhat   = 0.001
/
&card1
    istep       = 1
    iturm       = 1
    npseudotimesteps = 80000
    mtime       = 1
    iramp0      = 0
    iramp       = 1
    timestep    = 0
    iramp0_leak = 0
    iramp_leak  = 1
/
&card2
    invis       = F
    lamin       = F
    iorder      = 2
    irhsm       = 4
    lsgg        = 0
    mstg        = 4
    typlim      = 3
    fluxtype    = 3
    preconditioned = F
/
&card2a
    imp         = .false.
/
&card3
    cfl         = 0.3d0
    epss        = 0.5
    dtimedim    = 1.0d0
/
&card4
    u0          = 0.212d0
    tintens     = 0.05d0
    tlength     = 3.8d-5
/
&card5
    ptot        = 104548.744d0
    ttot        = 290.891d0
    pback       = 65583.427d0
    twall       = 288.000d0
/

```

```

&card6
  pref      = 101325.0d0
  tref      =   288.3d0
  scale     =    1.0d0
/
&card7
/
&card8
/
&card9
  debug      = F
  echo       = F
  intev_freq = 250
  res_freq   = 50
  itersave   = 10000
  intev_freq_pt = 250
  res_freq_pt = 50
  ires       = 1
  MonitorMaxMach = .true.
  MaxMachThreshold = 2.5d0
  reset_iter_counter = F
  Force_l    = .true.
  q_corrctn_limit = 1.0d-9
  q_corrctn_ratio = 10
/
&card10
  resrlx(1) = 1.0d0
  resrlx(2) = 1.0d0
  resrlx(3) = 1.0d0
  resrlx(4) = 1.0d0
  resrlx(5) = 1.0d0
  resrlx(6) = 0.5d0
  resrlx(7) = 0.5d0
/
&cardforced
  def_type = "none"
  cascade = F
/
&cardrom
/
&vortex
/
&cardprecon
  eps_minur = 0.5d0
/

```

D.5. NASA Rotor 67

This section includes samples of the input files used for the simulations presented in Chapters VI and VII. The input files given are for one specific simulation. Details on how to modify the input file for other operating points or numerical tests are given at the end of the input file.

D.5.a. *Inviscid Analysis*

This section first presents a typical input file for the second-order, inviscid simulation of Rotor 67. Lastly, the outlet pressure boundary conditions and a summary of the pertinent results are tabulated for the compressor map speed line operating points.

The Dervieux limiter was used for all but the cases associated with the higher-order limiter tests. Roe with the Harten entropy fix was used to compute the convective fluxes. Refer back to Section VI.4 (p. 141) for a more detailed description of the solution strategy.

D.5.a.i. Initial Operating Point

The input file given in this section was used to compute the second-order solution for the initial operating point presented in Section VI.4. This particular input file is the first of two, and requires the first-order solution restart file (`IOL.out`). The second input file restarts the solution computed by the given input file. The only differences between the input files, other than modifying the restart file parameters, are:

- Increasing `npseudotimesteps` from 500 to 55000
- Increasing `cf1` from 0.6d0 to 0.75d0
- Changing `reset_iter_counter` from T to F

The same input file can be used to compute the first-order solution by setting: `iorder` to 1, `ipex` to 0, and `istep` to 0.

```
&cardf
title      = "Rotor 67: Inviscid, IPEX=0"
tecplot_name = "plt/02.plt"
dump_tecplot = T
dump_yplus  = F
rsdfile     = "txt/004.dat"
case_name   = "yp/r67"
relative_V  = T
/
&cardg
gridfile    = "mesh/vol.mesh"
c2nfile     = "mesh/c2n.def"
XleXteRte  = "mesh/XleXteRte"
l2gfile     = "mesh/loc2glob.dat_012"
ijkfile     = "mesh/outlet_ij.def"
/
&cardh
filedin     = "out/IOL.out"
filedout    = "out/r004.out"
/
&cardi
fileturin   = "out/none.out"
fileturout  = "out/none.out"
/
&cardk
/
&cardO
noblade     = 22
inbc        = 0
ioutbc      = 1
ipex        = 1
igeom       = 0
ispet       = 0
leak_outlet = F
iwall       = 3
deltafhat   = 0.001
/
&card1
istep       = 1
iturm       = 0
npseudotimesteps = 5000
mtime       = 1
```

```

iramp0          = 0
iramp           = 1
timestep        = 0
iramp0_leak     = 0
iramp_leak      = 1
/
&card2
invis           = T
lamin           = T
iorder          = 2
irhsm           = 4
lsgg            = 2
mstg            = 4
typlim         = 3
fluxtype       = 1
/
&card2a
imp             = F
/
&card3
cfl             = 0.6d0
epss            = 0.5
dtimedim       = 1.0d0
/
&card4
u0              = 0.59d0
/
&card5
ptot            = 124452.598d0
ttot            = 305.742d0
pback           = 124452.598d0
/
&card6
pref            = 101325.0d0
tref            = 288.3d0
scale           = 1.0d0
/
&card7
/
&card8
omegax          = 16520.0d0
/
&card9
debug           = F
echo            = F

```

```

intev_freq      = 250
res_freq        = 25
itersave        = 500
intev_freq_pt   = 250
res_freq_pt     = 25
ires            = 1
MonitorMaxMach  = T
MaxMachThreshold = 2.5d0
reset_iter_counter = T
Force_l         = T
q_corrctn_limit = 1.0d-9
q_corrctn_ratio = 10
/
&card10
/
&cardforced
def_type = "none"
cascade = F
/
&cardrom
/
&vortex
init_vortex = F
/
&cardprecon
eps_minur = 0.1d0
/

```

D.5.a.ii. Compressor Map Speed Line

The primary modification to the input file to compute a different operating point was to change the outlet static pressure in the second-order input file given in the previous sub-section. The solution for each operating point would be restarted from the closest completed simulation, so the correct restart file (`filedin`) would also need to be supplied in the input file. Table A.1 (p. 218) lists the outlet static pressure (at the hub) used for each operating point simulated. The static pressures have been normalized by the inlet total pressure, $p_{0_1} = 124452.598$ Pa.

D.5.b. Turbulent Analysis: Rubbing

This section contains the input file for the second-order accurate, turbulent simulation of Rotor 67 with no tip clearance flow. The input file for the initial operating point for case R67TR-b is presented first, and is followed by instructions on how to set case R67TR-a and generate speed lines for both cases. Refer to Section VII.2 for a description of the cases and solution strategies. Assume code default values for variables not provided in the input file.

```
&cardf
title          = "Rotor 67: Turbulent/Rubbing/Initial Point"
tecplot_name   = "plt/001.plt"
dump_tecplot  = T
dump_yplus    = T
dump_v3       = F
rsdfile       = "txt/001.dat"
case_name     = "yp/R67"
relative_V    = T
/
&cardg
gridfile      = "mesh/vol.mesh"
c2nfile      = "mesh/c2n.def"
XleXteRte    = "mesh/XleXteRte"
l2gfile      = "mesh/loc2glob.dat_040"
ijkfile      = "mesh/outlet_ij.def"
/
&cardh
filedin       = "../F002/out/r001.out"
filedout     = "out/r001.out"
/
&cardi
fileturin    = "../F002/out/sst001.out"
fileturout   = "out/sst001.out"
/
&cardk
/
&cardO
noblade      = 22
inbc         = 0
ioutbc       = 1
ipex         = 1
igeom        = 0
```

```

ispet          = 0
leak_outlet    = F
iwall          = 3
deltafhat      = 0.001
/
&card1
istep          = 1
iturn          = 1
npseudotimesteps = 200000
mtime         = 1
iramp0        = 0
iramp         = 1
timestep      = 0
iramp0_leak   = 0
iramp_leak    = 1
/
&card2
invis         = F
lamin         = F
iorder        = 2
irhsm         = 4
lsgg          = 2
mstg          = 4
typlim        = 3
typlim_opt    = 0
fluxtype      = 1
/
&card2a
imp           = .false.
/
&card3
cfl           = 0.6d0
epss          = 0.5
dtimedim      = 1.0d0
/
&card4
u0            = 0.55d0
tintens       = 0.03d0
tlength       = 1.04d-3
/
&card5
ptot          = 124452.598d0
ttot          = 305.742d0
pback         = 112000.0d0
/

```



```

&card6
pref      = 101325.0d0
tref      = 288.3d0
scale     = 1.0d0
/
&card7
/
&card8
omegax    = 16520.0d0
xrle      = -1.374d-2
xrte      = 9.365d-2
/
&card9
debug      = F
echo      = F
intev_freq = 200
res_freq   = 10
itersave   = 5000
intev_freq_pt = 200
res_freq_pt = 10
ires       = 1
MonitorMaxMach = T
MaxMachThreshold = 2.5d0
reset_iter_counter = T
Force_l    = T
q_corrctn_limit = 1.0d-9
q_corrctn_ratio = 10
/
&card10
resrlx(1) = 1.0d0
resrlx(2) = 1.0d0
resrlx(3) = 1.0d0
resrlx(4) = 1.0d0
resrlx(5) = 1.0d0
resrlx(6) = 0.75d0
resrlx(7) = 0.75d0
/
&cardforced
def_type = "none"
cascade = F
/
&cardrom
/
&vortex
init_vortex = F

```

```

/
&cardprecon
eps_minur = 0.4d0
/

```

The same input file from above can be used with rubbing case R67TR-a by removing the input variables `xr1e` and `xrte`, from `&card8`, so that the default values are used by UNS3D.

D.5.b.i. Compressor Map Speed Line

The compressor map speed lines for each case were generated by modifying the back pressure in the above input file. Consult Table A.2 to find the hub outlet back pressures used for case R67TR-a. Table A.3 provides the boundary condition schedule for case R67TR-b.

D.5.c. Turbulent Analysis: Clearance

This section contains the input file for the second-order accurate, turbulent simulation of Rotor 67 with a modeled tip clearance. The input file for the initial operating point is given first. Additional details on computing the flow using different gradients, higher-order limiters are given after the input file is presented. Finally, instructions on computing the speed line are given. Assume default values for all variables not explicitly defined.

```

&cardf
title          = "Rotor 67: Turbulent/Initial Point"
tecplot_name   = "plt/002.plt"
dump_tecplot   = T
dump_yplus     = T
dump_v3        = F
rsdfile        = "txt/002.dat"
case_name      = "yp/R67"
relative_V     = T
/
&cardg
gridfile       = "mesh/vol.mesh"

```

```

c2nfile      = "mesh/c2n.def"
XleXteRte   = "mesh/XleXteRte"
l2gfile      = "mesh/loc2glob.dat_040"
ijkfile      = "mesh/outlet_ij.def"
/
&cardh
filedin      = "out/r001.out"
filedout     = "out/r002.out"
/
&cardi
fileturin    = "out/sst001.out"
fileturout   = "out/sst002.out"
/
&cardk
/
&card0
noblade      = 22
inbc         = 0
ioutbc       = 1
ipex         = 1
igeom        = 0
ispet        = 0
leak_outlet  = F
iwall        = 3
deltafhat    = 0.001
/
&card1
istep        = 1
iturm        = 1
npseudotimesteps = 100000
mtime        = 1
iramp0       = 0
iramp        = 1
timestep     = 0
iramp0_leak  = 0
iramp_leak   = 1
/
&card2
invis        = F
lamin        = F
iorder       = 2
irhsm        = 4
lsgg         = 2 ! LSQR
mstg         = 4
typlim       = 3 ! Dervieux

```

```

typlim_opt = 0
fluxtype   = 1
/
&card2a
imp        = .false.
/
&card3
cfl        = 0.6d0
epss       = 0.5
dtimedim   = 1.0d0
/
&card4
u0         = 0.55d0
tintens    = 0.03d0
tlength    = 1.04d-3
/
&card5
ptot       = 124452.598d0
ttot       = 305.742d0
pback      = 116525.0d0
/
&card6
pref       = 101325.0d0
tref       = 288.3d0
scale      = 1.0d0
/
&card7
/
&card8
omegax     = 16520.0d0
xrle       = -1.374d-2
xrte       = 9.365d-2
/
&card9
debug      = F
echo       = F
intev_freq = 200
res_freq   = 10
itersave   = 5000
intev_freq_pt = 200
res_freq_pt = 10
ires       = 1
MonitorMaxMach = T
MaxMachThreshold = 2.5d0
reset_iter_counter = F

```

```

Force_l           = T
q_corrctn_limit  = 1.0d-9
q_corrctn_ratio  = 10
/
&card10
resrlx(1) = 1.0d0
resrlx(2) = 1.0d0
resrlx(3) = 1.0d0
resrlx(4) = 1.0d0
resrlx(5) = 1.0d0
resrlx(6) = 0.75d0
resrlx(7) = 0.75d0
/
&cardforced
def_type = "none"
cascade = F
/
&cardrom
/
&vortex
init_vortex = F
/
&cardprecon
eps_minur = 0.4d0
/

```

Green-Gauss or basic least-squares may be used for gradient evaluation by switching `lsgg` to 0 or 1, respectively. Consult Table D.1 on how to change the input parameters `typlim` and `typlim_opt` to use the different limiters.

D.5.c.i. Compressor Map Speed Line

The above input file can be used to compute the two speed lines that were presented in Section VII.3. The gradient method is changed through the parameter `lsgg`, as mentioned just above. The additional operating points are found using the outlet boundary schedules tabulated in Tables A.4 and A.5.

D.6. Ada HPC System Batch File Example for MPI

```
#BSUB -J R67_SC07_P1R001 # job name
```

```
#BSUB -L /bin/bash          # always here
#BSUB -W 65:00              # sets wallclock limit (65 hours)
#BSUB -n 40                 # assigns 8 cpus/cores for execution.
#BSUB -R "span[ptile=20]"   # assigns 8 cores per node.
#BSUB -M 5000               # sets to 5,000MB the per process
#BSUB -R "rusage[mem=1000]" #
#BSUB -R "select[nxt]"     # node-type
#BSUB -o stdout1.%J        # directs the job's standard output to
                           # stdout1.jobid

cd $LS_SUBCWD # change to directory where job was submitted
module load intel          # loads the INTEL software tool chain
mpirun -np 40 ./uns3d input.007 > $LS_SUBCWD/uns3d.input07 # execute
```

APPENDIX E

TURBOMACHINERY GRID GENERATOR INPUT FILES

This appendix contains the namelist format input files for the turbomachinery grid generator known as TGG, which was presented in Chapter III. The description of every variable and the necessary ancillary files for the grid generator was given in Appendix F. The units (e.g. feet or meters) for each case will be indicated in the preamble of each subsequent subsection.

E.1. Turbine Vane

This section contains the TGG input files for the three turbine vane grids discussed in Chapter V. Refer to Appendix F for the description of the variables given. Any variable not given is assumed to have the code assigned default value. The file names for the coarse, medium, and fine grids are, respectively, `GT00.nml`, `GT01.nml`, and `GT02.nml`. The geometry and the grid spacings given in each input file are in feet.

E.1.a. Coarse Grid

```
&GRID_SIZE
  IMAX1    = 111,
  JMAX1    = 21,
  KMAX     = 50,
  KBLH     = 2,
  KBLT     = 15,
  KBLB     = 15,
  IMAX2    = 10,
  JMAX2    = 19,
  IMAX3    = 14,
  JMAX3    = 9,
  JMAX4    = 23,
  JOFFREEZE = 12,
  I2D      = 15,
  K2D      = 5,
```

```

/
&GRID_DIMS
S1MAX      = 0.01d0
DSMIN      = 2.4d-5
DSMAX      = 4.0d-3
USE_DSMAX  = F,
DSRA       = 0.452d0
DSLE       = 2.0d-3
DSTE       = 8.0d-5
DSHUB      = 3.6d-5
DSTIP      = 3.6d-5
DELTAH     = 1.15d-3
DELTAT     = 6.0d-4
DELTAB     = 6.0d-4
/
&GRID_CTRL
KCLUSS     = 1,
B1_EPS1    = -6.0d0,
B1_EPS2    = 0.0d0,
B1_ALPHA   = 1.0d0,
OMEGA      = 0.5d0,
ITER       = 500
PMAG       = 0.7d0,
QMAG       = 0.8d0,
SINGDIST   = 2.0d-3, 2.0d-3
IROT       = 108, 108
/
&CAP_GRID
HUBCAP     = F,
TIPCAP     = F,
dscap      = 3.28084d-3,
dscap      = 2.5d-3,
dsclus     = 4.0625d-6
jc1max     = 11,
jc2max     = 35,
kcmax      = 70,
iadj_le    = 25,
iadj_te    = 25,
citer      = 1000,
citerl     = 1,
avg_corner = 0.5d0,
omegax     = 0.3d0,
rext       = 5.0d0,
sext       = 8.0d0,
/

```



```

&PAD_GRID
  inletH   = T
  outletH  = T
  ilmax    = 17
  irmax    = 25
  /
&BOUNDS
  XBH = -0.65d0, -0.045d0, 0.17d0, 0.8d0
  RBH = 0.65d0, 0.65d0, 0.65d0, 0.65d0
  XBT = -0.65d0, -0.045d0, 0.17d0, 0.8d0
  RBT = 0.84d0, 0.84d0, 0.84d0, 0.84d0
  /
&CTRLS
  CONFIG = 2,
  BLADEFIELD = "input/gold.dat",
  FLOWPATH  = "input/gold.path",
  UGRID     = "vol.ugrid",
  TECFILE   = "vol.plt",
  XTRANS    = 0.0d0
  ROTATING  = F,
  VANE      = T,
  DEBUG     = F,
  DBG_PLT(1) = F,
  DBG_PLT(2) = F,
  DBG_PLT(3) = F,
  DBG_PLT(4) = F,
  DBG_PLT(5) = F,
  DBG_PLT(6) = F,
  DBG_PLT(8) = F,
  /
SINGMAG
  2.000000      1.000000      1.000000      5.000000
  2.000000      1.000000      1.000000      5.000000
SINGFADE
  0.300000      0.300000      0.300000      0.300000
  0.300000      0.300000      0.300000      0.300000

```

E.1.b. Medium Grid

```

&GRID_SIZE
  IMAX1   = 221,
  JMAX1   = 41,
  KMAX    = 100,
  KBLH    = 2,

```

```

KBLT   = 15,
KBLB   = 15,
IMAX2  = 20,
JMAX2  = 35,
IMAX3  = 25,
JMAX3  = 15,
JMAX4  = 45,
JOFREEZE = 25,
I2D    = 15,
K2D    = 5,
/
&GRID_DIMS
S1MAX   = 0.01d0
DSMIN   = 1.2d-5
DSMAX   = 4.0d-3
USE_DSMAX = F,
DSRA    = 0.452d0
DSLE    = 1.0d-3
DSTE    = 4.0d-5
DSHUB   = 1.8d-5
DSTIP   = 1.8d-5
DELTAH  = 1.15d-3
DELTAT  = 6.0d-4
DELTAB  = 6.0d-4
/
&GRID_CTRL
KCLUSS  = 1,
B1_EPS1 = -3.0d0,
B1_EPS2 = 0.0d0,
B1_ALPHA = 1.0d0,
OMEGA   = 0.5d0,
ITER    = 500
PMAG    = 0.7d0,
QMAG    = 0.4d0,
SINGDIST = 1.5d-3, 1.2d-3
IROT    = 215, 215
/
&CAP_GRID
HUBCAP   = F,
TIPCAP   = F,
dscap    = 3.28084d-3,
dscap    = 2.5d-3,
dsclus   = 4.0625d-6
jc1max   = 11,
jc2max   = 35,

```

```

kcmx      = 70,
iadj_le   = 25,
iadj_te   = 25,
citer     = 1000,
citerl    = 1,
avg_corner = 0.5d0,
omegax    = 0.3d0,
rext      = 5.0d0,
sext      = 8.0d0,
/
&PAD_GRID
inletH    = T
outletH   = T
ilmax     = 35
irmax     = 50
/
&BOUNDS
XBH = -0.65d0, -0.045d0, 0.17d0, 0.8d0
RBH = 0.65d0, 0.65d0, 0.65d0, 0.65d0
XBT = -0.65d0, -0.045d0, 0.17d0, 0.8d0
RBT = 0.84d0, 0.84d0, 0.84d0, 0.84d0
/
&CTRLS
CONFIG = 2,
BLADEFILE = "input/gold.dat",
FLOWPATH  = "input/gold.path",
UGRID     = "vol.ugrid",
TECFILE   = "vol.plt",
XTRANS    = 0.0d0
ROTATING  = F,
VANE      = T,
DEBUG     = F,
DBG_PLT(1) = F,
DBG_PLT(2) = F,
DBG_PLT(3) = F,
DBG_PLT(4) = F,
DBG_PLT(5) = F,
DBG_PLT(6) = F,
DBG_PLT(8) = F,
/
SINGMAG
      2.000000      1.000000      1.000000      2.000000
      2.000000      1.000000      1.000000      3.000000
SINGFADE
      0.300000      0.300000      0.300000      0.300000

```

0.3000000 0.300000 0.300000 0.300000

E.1.c. Fine Grid

&GRID_SIZE

IMAX1 = 443,
JMAX1 = 81,
KMAX = 200,
KBLH = 2,
KBLT = 15,
KBLB = 15,
IMAX2 = 40,
JMAX2 = 70,
IMAX3 = 50,
JMAX3 = 30,
JMAX4 = 91,
JOFREEZE = 50,
I2D = 15,
K2D = 5,

/

&GRID_DIMS

S1MAX = 0.01d0
DSMIN = 6.0e-6
DSMAX = 4.0d-3
USE_DSMAX = F,
DSRA = 0.452d0
DSLE = 5.0d-4
DSTE = 2.0d-5
DSHUB = 9.0d-6
DSTIP = 9.0d-6
DELTAH = 1.15d-3
DELTAT = 6.0d-4
DELTAB = 6.0d-4

/

&GRID_CTRL

KCLUSS = 1,
B1_EPS1 = -15.0d0,
B1_EPS2 = 0.0d0,
B1_ALPHA = 1.0d0,
OMEGA = 0.5d0,
ITER = 1000
PMAG = 0.5d0,
QMAG = 0.4d0,
SINGDIST = 3.0d-4, 3.0d-4

```

IROT      = 430, 430
/
&CAP_GRID
HUBCAP    = F,
TIPCAP    = F,
dscap     = 3.28084d-3,
dscap     = 2.5d-3,
dsclus    = 4.0625d-6
jc1max    = 11,
jc2max    = 35,
kcmax     = 70,
iadj_le   = 25,
iadj_te   = 25,
citer     = 1000,
citerl    = 1,
avg_corner = 0.5d0,
omegax    = 0.3d0,
rext      = 5.0d0,
sext      = 8.0d0,
/
&PAD_GRID
inletH    = T
outletH   = T
ilmax     = 70
irmax     = 100
/
&BOUNDS
XBH = -0.65d0, -0.045d0, 0.17d0, 0.8d0
RBH = 0.65d0, 0.65d0, 0.65d0, 0.65d0
XBT = -0.65d0, -0.045d0, 0.17d0, 0.8d0
RBT = 0.84d0, 0.84d0, 0.84d0, 0.84d0
/
&CTRLS
CONFIG = 2,
BLADEFIELD = "input/gold.dat",
FLOWPATH  = "input/gold.path",
UGRID     = "vol.ugrid",
TECFIELD  = "vol.plt",
XTRANS    = 0.0d0
ROTATING  = F,
VANE      = T,
DEBUG     = F,
DBG_PLT(1) = F,
DBG_PLT(2) = F,
DBG_PLT(3) = F,

```

```

DBG_PLT(4) = F,
DBG_PLT(5) = F,
DBG_PLT(6) = F,
DBG_PLT(8) = F,
/
SINGMAG
  2.000000      1.000000      1.000000      2.000000
  3.000000      1.000000      1.000000      4.000000
SINGFADE
  0.200000      0.300000      0.300000      0.200000
  0.200000      0.200000      0.200000      0.200000

```

E.2. Rotor 67

This section contains the TGG input files for all of the NASA Rotor 67 cases presented in Chapters VI and VII. Refer to Appendix F for the description of the variables given. Any variable not given is assumed to have the code assigned default value. The geometry and the grid spacings given in each input file are in **feet**.

E.2.a. Inviscid Rotor 67

```

&GRID_SIZE
  IMAX1      = 131,
  JMAX1      = 17,
  KMAX       = 36,
  IMAX2      = 17,
  JMAX2      = 21,
  IMAX3      = 19,
  JMAX3      = 25,
  JMAX4      = 25,
  JOFFREEZE  = 3,
  I2D        = 15,
  K2D        = 15,
/
&GRID_DIMS
  S1MAX      = 0.015d0
  DSMIN      = 5.0d-4
  DSMIN      = 1.0d-3
  DSMAX      = 4.0d-3
  USE_DSMAX  = F,
  DSRA       = -0.47d0

```

```

DSLE      = 3.75e-4
DSTE      = 3.0d-4
DSHUB     = 2.8e-3
DSTIP     = 2.8e-3

/
&GRID_CTRL
B1_EPS1   = -3.0d0,
B1_EPS2   = 0.0d0,
B1_ALPHA  = 1.0d0,
OMEGA     = 0.5d0,
OMEGA     = 0.4d0,
ITER      = 2500,
PMAG      = 0.3d0,
QMAG      = 1.0d0,
SINGDIST  = 3.0d-3, 3.0d-3
IROT      = 4, 4
/
&CAP_GRID
HUBCAP    = F,
TIPCAP    = F,
dscap     = 3.28084d-3,
dsclus    = 4.0d-5,
jc1max    = 9,
jc2max    = 3,
kcmax     = 20,
iadj_le   = 20,
iadj_te   = 20,
citer     = 600,
citerl    = 1,
avg_corner = 0.7d0,
omegax    = 0.01d0,
rext      = 10.0d0,
sext      = 0.5d0,
/
&PAD_GRID
inletH    = T
outletH   = T
ilmax     = 30
irmax     = 45
/
&BOUNDS
XBH = -1.5d0, -0.05d0, 0.356d0, 1.775d0
RBH = 0.20d0, 0.3d0, 0.38d0, 0.38d0
XBT = -0.2d0, -0.01d0, 0.31d0, 0.5d0

```

```

XBT = -1.5d0, -0.01d0, 0.31d0, 1.775d0
RBT = 0.842d0, 0.842d0, 0.842d0, 0.81d0
/
&CTRLS
CONFIG = 2,
BLADEFILE = "input/geom.dat",
FLOWPATH = "input/r67_ext.path",
UGRID = "vol.ugrid",
TECFILE = "vol.plt",
XTRANS = 0.0d0
FSURF = -4,
ROTATING = T,
VANE = F,
DEBUG = F,
DBG_PLT(1) = F,
DBG_PLT(2) = F,
DBG_PLT(3) = F,
DBG_PLT(4) = F,
DBG_PLT(5) = F,
DBG_PLT(6) = F,
DBG_PLT(8) = F,
/
SINGMAG
      3.000000      3.000000      3.000000      3.000000
      3.000000      3.000000      3.000000      3.000000
SINGFADE
      0.7500000    0.7500000    0.7500000    0.7500000
      0.200000    0.3000000    0.2000000    0.2000000

```

E.2.b. Turbulent Rotor 67: Rubbing

```

&GRID_SIZE
IMAX1 = 155,
IMLE = 25,
IMTE = 15,
JMAX1 = 75,
KMAX = 135,
IMAX2 = 11,
JMAX2 = 27,
IMAX3 = 11,
JMAX3 = 27,
JMAX4 = 25,
JOFREEZE = 30,
I2D = 15,

```



```

K2D      = 15,
ICLUSB   = T,
IMCLUSB  = 15,
IC_LEFT  = 33,
IC_RIGHT = 41,
/
&GRID_DIMS
S1MAX    = 2.0d-2
DSMIN    = 5.15625d-5
DSMAX    = 1.5d-4
USE_DSMAX = F,
DSRA     = -0.47d0
DSLE     = 2.5d-5
DSTE     = 1.0d-4
DSHUB    = 6.0d-6
DSTIP    = 3.0d-6
DELTAH   = 1.8d-3
DELTAT   = 1.8d-3
DELTAB   = 6.0d-4
VARY_Y1  = T,
YPLUS    = 2.0d0
EVEN_LETE = T,
LETE_SPAN = 0.00, 0.5d0, 1.0d0
DSLE_AF  = 6.0d-3, 4.75d-3, 3.125d-3
DSTE_AF  = 5.5d-3, 5.0d-3, 3.1d-3
/
&GRID_CTRL
KCLUSS   = 1,
B1_EPS1  = -15.0d0,
B1_EPS2  = 0.0d0,
B1_ALPHA = 1.0d0,
OMEGA    = 0.6d0,
ITER     = 5000
PMAG     = 0.2d0,
QMAG     = 0.55d0 !0.55d0,
SINGDIST = 0.0015d0, 0.0035d0
IROT     = 9, 9
INEX_EQUALDT = T
/
&CAP_GRID
HUBCAP   = F,
TIPCAP   = F,
dscap    = 2.5d-3,
dsclus   = 4.0625d-6
jc1max   = 5,

```

```

jc2max      = 17,
kcmx       = 70,
iadj_le    = 20,
iadj_te    = 20,
citer      = 100,
citerl     = 1,
avg_corner = 0.5d0,
omegax     = 0.3d0,
rext       = 1.0d0,
sext       = 10.0d0,
/
&PAD_GRID
inletH     = T
outletH    = T
ilmax      = 35
irmax      = 35
/
&BOUNDS
XBH = -1.5d0, -0.05d0, 0.356d0, 1.775d0
RBH = 0.20d0, 0.3d0, 0.38d0, 0.38d0
XBT = -1.5d0, -0.01d0, 0.31d0, 1.775d0
RBT = 0.842d0, 0.842d0, 0.842d0, 0.81d0
/
&CTRLS
CONFIG = 2,
BLADEFIL = "input/geom.dat",
FLOWPATH = "input/r67_ext.path",
UPSTRMFILE = "input/velin_100pwhspd.dat",
UGRID     = "vol.ugrid",
TECFIL   = "vol.plt",
XTRANS   = 0.0d0
FSURF    = -4 ! set fixed surfaces to rotating
ROTATING = T,
VANE     = F,
DEBUG    = F,
DBG_PLT(1) = F,
DBG_PLT(2) = F,
DBG_PLT(3) = F,
DBG_PLT(4) = F,
DBG_PLT(5) = F,
DBG_PLT(6) = F,
DBG_PLT(8) = F,
/
SINGMAG
2.00 1.00 1.00 1.00

```

```

    3.00  3.00  3.00  3.00
SINGFADE
    0.70  0.70  0.70  0.70
    0.70  0.50  0.50  0.70

```

E.2.c. Turbulent Rotor 67: Clearance

```

&GRID_SIZE
  IMAX1   = 155,
  IMLE    = 25,
  IMTE    = 15,
  JMAX1   = 75,
  KMAX    = 175,
  IMAX2   = 11,
  JMAX2   = 27,
  IMAX3   = 11,
  JMAX3   = 27,
  JMAX4   = 25,
  JOFFREEZE = 30,
  I2D     = 15,
  K2D     = 15,
  ICLUSB  = T,
  IMCLUSB = 15,
  IC_LEFT = 33,
  IC_RIGHT = 41,
  /
&GRID_DIMS
  S1MAX   = 2.0d-2
  DSMIN   = 5.15625d-5
  DSMAX   = 1.5d-4
  USE_DSMAX = F,
  DSRA    = -0.47d0
  DSLE    = 2.5d-5
  DSTE    = 1.0d-4
  DSHUB   = 6.0d-6
  DSTIP   = 3.0d-6
  DELTAH  = 1.8d-3
  DELTAT  = 1.8d-3
  DELTAB  = 6.0d-4
  VARY_Y1 = T,
  YPLUS   = 2.0d0
  EVEN_LETE = T,
  LETE_SPAN = 0.00, 0.5d0, 1.0d0
  DSLE_AF  = 6.0d-3, 4.75d-3, 3.125d-3

```

```

DSTE_AF = 5.5d-3, 5.0d-3, 3.1d-3
/
&GRID_CTRL
KCLUSS = 1,
B1_EPS1 = -15.0d0,
B1_EPS2 = 0.0d0,
B1_ALPHA = 1.0d0,
OMEGA = 0.6d0,
ITER = 5000
PMAG = 0.2d0,
QMAG = 0.55d0 !0.55d0,
SINGDIST = 0.0015d0, 0.0035d0
IROT = 9, 9
INEX_EQUALDT = F
/
&CAP_GRID
HUBCAP = F,
TIPCAP = T,
dscap = 3.3d-3,
dsclus = 4.0d-6
jc1max = 9,
jc2max = 25,
kcmx = 40,
iadj_le = 20,
iadj_te = 20,
citer = 500,
citerl = 1,
avg_corner = 0.5d0,
omegax = 0.3d0,
rext = 1.0d0,
sext = 8.0d0,
/
&PAD_GRID
inletH = T
outletH = T
ilmax = 35
irmax = 35
/
&BOUNDS
XBH = -1.5d0, -0.05d0, 0.356d0, 1.775d0
RBH = 0.20d0, 0.3d0, 0.38d0, 0.38d0
XBT = -1.5d0, -0.01d0, 0.31d0, 1.775d0
RBT = 0.842d0, 0.842d0, 0.842d0, 0.81d0
/
&CTRLS

```

```
CONFIG = 2,  
BLADEFILE = "input/geom.dat",  
FLOWPATH = "input/r67_ext.path",  
UPSTRMFILE = "input/velin_100pwhspd.dat",  
UGRID = "vol.ugrid",  
TECFILE = "vol.plt",  
XTRANS = 0.0d0  
FSURF = -3  
ROTATING = T,  
VANE = F,  
DEBUG = F,  
DBG_PLT(1) = F,  
DBG_PLT(2) = F,  
DBG_PLT(3) = F,  
DBG_PLT(4) = F,  
DBG_PLT(5) = F,  
DBG_PLT(6) = T,  
DBG_PLT(8) = F,  
/  
SINGMAG  
    2.00  1.00  1.00  1.00  
    3.00  3.00  3.00  3.00  
SINGFADE  
    0.70  0.70  0.70  0.70  
    0.70  0.50  0.50  0.70
```

APPENDIX F

TURBOMACHINERY GRID GENERATOR MANUAL

– Version 300 –

F.1. Introduction

Turbomachinery Grid Generator (TGG) is a three-dimensional grid generation code for turbomachinery blades. TGG can generate grids for both linear and annular cascades for use with the flow solver known as UNS3D using an unstructured UGRID format grid file. The code uses a structured multi-block grid system to discretize a single blade passage. The blade passage is discretized using an O-4H multi-block configuration. The structured grid blocks are converted to an unstructured format in the final step.

Additional grid blocks can be added to enhance the grid or model different aspects of the given turbomachinery flow. H-grids can be add upstream and downstream of the main O-4H block to extend the distance from the boundaries to the blade. TGG also offers the ability mesh the region between the tip of a rotor or stator and the adjacent endwall. For this task, an O-H grid block set is used to discretize the space “inside” a blade for a given clearance distance.

This manual describes how to both compile and use this code. The compilation and general execution of TGG is discussed first. Secondly, the input files needed by TGG are described. Lastly, the output from TGG is described.

F.2. Compiling and Running TGG

TGG is written completely in FORTRAN, and can be compiled using the makefile included with the source code. TGG does require the Lapack and Blas libraries. These libraries should be compiled using the user’s compiler of choice prior to the

compilation of TGG. The LIBS entry of the makefile must be updated to reflect the location of the Lapack and Blas libraries. To compile TGG, change to the top source directory and execute the following command: `make -f makefile`. The resultant executable will be called `tgg`.

TGG is executed via command line statements. The executable should be copied or symbolically linked (recommended) to the directory in which the code input files are located. The command line statement to execute TGG is: `./tgg <input file>`. The name of desired input is the only argument, `<input file>`, to be passed to TGG. The main input file will be presented in the following chapter. Of note, the default geometry type for TGG is for annular cascades. Linear cascades can be created using `IGEOM=0`, as will be shown in Section F.3.a.vii.

F.3. Input Files

This section describes the input files required by TGG to construct the desired turbomachinery grid. The files described within this section are the main input file, the flowpath description file, the blade geometry description file, and an optional freestream description file.

F.3.a. Main Input File

The main input file is a namelist format file. The name of the main input file is given to TGG by way of a code argument. For example: `./tgg input.nml`, where `input.nml` is the name of the file. Additional information is also contained after the last namelist card at the end of the file. Each namelist as well as the after-namelist variables are defined below.

F.3.a.i. `&grid_size`

Defines the number of grid points in each block. Refer to Fig. F.1 for grid block locations and (i, j) origins.

KMAX	Total number of grid layers from hub to tip in the complete grid. <i>Integer</i> . Default = 25
IMAX1	Total number of i grid points in Block 1. Must be odd. <i>Integer</i> .
JMAX1	Total number of j grid points in Block 1. <i>Integer</i> .
IMAX2	Total number of i grid points in Block 2. <i>Integer</i> .
JMAX2	Total number of j grid points in Block 2. Must be odd. <i>Integer</i> .
IMAX3	Total number of i grid points in Block 3. <i>Integer</i> .
JMAX3	Total number of j grid points in Block 3. Must be odd. <i>Integer</i> .
JMAX4	Total number of j grid points in Block 4. Must be odd. <i>Integer</i>
JOFREEZE	Number of j layers adjacent to the blade surface in the O-grid (Block 1) that are not smoothed by the elliptical solver. Use variable to adequately capture the boundary layer as the orthogonality of the un-smoothed hyperbolic O-grid is maintained. <i>Integer</i> . Default = 15
I2D	Total number of i grid points used to define the meridional grid. See Fig. F.2a. <i>Integer</i> .
K2D	Total number of meridional grid layers (or k -layers). See Fig. F.2a. $K2D \geq 4$ for final production run, but can be set to 2 or 3 if <code>DBG_PLT(6) = T</code> . <i>Integer</i> .
KBLH	Number of spanwise layers clustered near the hub endwall. Used if <code>KLUSS = 2</code> . See Fig. F.4. <i>Integer</i> .
KBLT	Number of spanwise layers clustered near the tip endwall. Used if <code>KLUSS = 2</code> . See Fig. F.4. <i>Integer</i> .
KBLB	Number of spanwise layers clustered around the tip of the blade for clearance flow cases (<code>HUBCAP</code> or <code>TIPCAP = T</code>). Used only if <code>KLUSS = 2</code> . See Fig. F.4b-c. <i>Integer</i> .
IMLE	Number of evenly spaced points placed around the leading edge if <code>EVEN_LETE = T</code> . Must be an odd. <i>Integer</i>

IMTE Number of evenly spaced points placed around the trailing edge if `EVEN_LETE = T`. Must be odd. *Integer*

F.3.a.ii. `&grid_dims`

Defines the physical dimensions of the grid using user specified units (m, mm, ft, etc.). Additional surface clustering is also defined by this card.

S1MAX	Height of the un-smoothed O-grid around the airfoil section. See Fig. F.3. Height will change when smoothed. <i>Real</i>
DSMIN	Uniform first cell height adjacent to the blade surface. See Fig. F.3. <i>Real</i>
DSMAX	Initial grid spacing adjacent to O-grid outer boundary. See Fig. F.3. Spacing will not be maintained during grid smoothing procedure. Only used if <code>USE_DSMAX = T</code> . <i>Real</i>
USE_DSMAX	If T the O-grid spacing along constant i lines are set using a double sided spacing with <code>DSMIN</code> next to the airfoil and <code>DSMAX</code> at the O-grid boundary. <i>Logical</i> . Default = F
DSRA	Ratio of the pressure side surface arc-length (TE→LE) to the entire surface arc-length (TE→TE). Used to set the location of the leading edge. <i>Real</i> . Default = 0.5
DSLE	Surface spacing applied on either side of the leading edge point (defined using <code>DSRA</code>). See Fig. F.3. Used if <code>EVEN_LETE = F</code> . <i>Real</i>
DSTE	Surface spacing applied on either side of the trailing edge point ($i = 1$). See Fig. F.3. Used if <code>EVEN_LETE = F</code> . <i>Real</i>
DSHUB	Grid spacing used to set a uniform first grid height off of the hub endwall. See Fig. F.4. <i>Real</i>
DSTIP	Grid spacing used to set a uniform first grid height off of the tip endwall. See Fig. F.4. <i>Real</i>
DELTAH	Height of clustered grid layer region near the hub endwall (single sided clustering using <code>DSHUB</code> and <code>KBLH</code>). See Fig. F.4. Only used if <code>KCLUSS = 2</code> . <i>Real</i>
DELTAT	Height of clustered grid layer region near the tip endwall (single sided clustering using <code>DSTIP</code> and <code>KBLT</code>). See Fig. F.4. Only used if <code>KCLUSS = 2</code> . <i>Real</i>

DELTAB	Height on either side of the blade tip in which grid layers are clustered using a single sided clustering (DSCLUS and KBLB). See Fig. F.4b–c. Only used if HUBCAP or TIPCAP = T and KCLUSS = 2. <i>Real</i>
ICLUSB	If true, the original grid spacing around the airfoil is modified by adding points between two specified i -index points (IC_LEFT and IC_RIGHT). The points between the selected nodes are removed and replaced using a user specified number of points (IMCLUSB). A double sided grid spacing algorithm is used to smoothly add the new points with the initial spacings being proportional to the neighboring grid spacing. The corresponding points on the opposite surface are also modified in the same manner to maintain periodicity. See Fig. F.3. <i>Logical</i> . Default = F
IMCLUSB	Number of points added to each side of the airfoil if ICLUS = T. $IMAX1 = IMAX1 + 2(ICLUSB - IC_RIGHT + IC_LEFT - 1)$. See Fig. F.3. <i>Integer</i>
IC_LEFT	Left endpoint of the airfoil surface clustering region. Can be chosen to be on either the pressure side or the suction side so long as IC_LEFT < IC_RIGHT. See Fig. F.3. <i>Integer</i>
IC_RIGHT	Right endpoint of the airfoil surface clustering region. Must be greater than IC_LEFT. See Fig. F.3. <i>Integer</i>
VARY_Y1	Determines the method for which the viscous spacing is applied to the airfoil surface. <i>Logical</i> F Constant spacing using DSMIN (default) T Blasius boundary layer solution used to find streamwise varying y_1 for a specified, constant y^+ . Requires a user specified freestream velocity profile (UPSTRMFILE).
YPLUS	Desired blade surface y^+ number used to find y_1 . Used if VARY_Y1 = T. <i>Real</i> . Default = 1.0
EVEN_LETE	Sets the grid point spacing around the leading and trailing edges of each airfoil section. <i>Logical</i> F Spacing at leading and trailing edges set using DSLE and DSTE, respectively. (default) T Grid points are evenly spaced around the leading and trailing edge points.

DSLE_AF	Percentage of the total surface arc-length that defines the equally spaced leading edge region. Vector of length 3, where each component corresponds to a given spanwise location (LETE_SPAN). <i>Real</i>
DSTE_AF	Percentage of the total surface arc-length that defines the equally spaced trailing edge region. Vector of length 3, where each component corresponds to a given spanwise location (LETE_SPAN). <i>Real</i>
LETE_SPAN	Vector of length 3 that defines the normalized spanwise locations of DSLE_AF and DSTE_AF. <i>Real</i> . Default = (/0.0d0, 0.5d0, 1.0d0/)

F.3.a.iii. &grid_ctrl

B1_EPS1	Implicit artificial dissipation for the hyperbolic O-grid algorithm. Use value less than 0. <i>Real</i> . Default = -3.0
B1_EPS2	Explicit artificial dissipation for the hyperbolic O-grid algorithm. Use value less than or equal to 0; however, start with 0. <i>Real</i> . Default = 0.0
B1_ALPHA	Volume relaxation factor for the hyperbolic O-Grid algorithm. Values less than 1 result in diminishing volumes. Recommend using default value. <i>Real</i> . Default=1.0
OMEGA	Smoothing relaxation factor. <i>Real</i> . Default=0.1
ITER	Total number of elliptic smoothing iterations. <i>Integer</i> . Default=3000
PMAG	O-grid orthogonality control function parameter. The orthogonality of the grid maintained further away from the airfoil surface for lower values of PMAG. <i>Real</i> . Default=0.3
QMAG	O-grid spacing control function parameter. The original grid spacing is maintained further away from the airfoil surface for lower values of QMAG. <i>Real</i> . Default=0.5
SINGDIST	Ideal length for the edges connected to the “singularity” points, open circles in Fig. F.1. Varied linearly from hub to tip. SINGDIST is array of length 2. <i>Real</i>

	<code>singdist(1)</code>	Defines variable on hub grid layer; constant at all four points
	<code>singdist(2)</code>	Defines variable on tip grid layer; constant at all four points
IRROT		Moves the original $i = 1$ point to $i = \text{IRROT}$. See Fig. F.3. Can be useful for geometries with large stagger angles. However, only a single value can be applied to the entire blade; therefore, the default value is recommended for highly twisted blades. <i>Integer</i> (vector of length 2). Default=1

F.3.a.iv. `&cap_grid`

Specifies the addition of the clearance flow grids adjacent to either the hub or tip endwalls. Only one set of grids can be constructed, i.e. HUBCAP and TIPCAP can not both be T.

HUBCAP	Flag for generating clearance flow grids adjacent to the hub endwall. <i>Logical</i> F No grids constructed. (default) T O-H grids constructed
TIPCAP	Flag for generating clearance flow grids adjacent to the tip endwall. <i>Logical</i> F No grids constructed. (default) T O-H grids constructed
DSCAP	Distance from adjacent endwall to the end of the blade. See Fig. F.4bc. <i>Real</i>
DSCLUS	Distance from blade tip layer to each adjacent grid layer. Use to set the viscous spacing on the blade tip surface. See Fig. F.4bc. <i>Real</i>
KCMAX	Number of grid layers in the clearance region (includes endwall and blade tip layers). See Fig. F.4bc. <i>Integer</i>
JC1MAX	Number of j grid layers in the cap H-grid. Must be odd. See Fig. F.5a. <i>Integer</i>
JC2MAX	Number of j grid layers in the cap O-grid. See Fig. F.5a. <i>Integer</i>

CITER	Number of elliptic smoothing loops to run. Grid boundaries updated at beginning of loop. <i>Integer</i> . Default=600
CITERL	Number of iterations per smoothing loop. One is usually sufficient. <i>Integer</i> . Default=1
AVG_CORNER	Weighting factor used to smoothly set H-grid corner points. See Fig. F.5b. <i>Real</i> . Default=0.7
OMEGAX	Smoothing relaxation factor. <i>Real</i> . Default=0.5
REXT	O-grid orthogonality control function parameter. The orthogonality of the mesh is maintained further towards the center of the cap grid for lower values of REXT. <i>Real</i> . Default=20.0
SEXT	O-grid spacing control function parameter. The original O-grid spacing along constant <i>i</i> grid lines is maintained further into the domain for lower values. <i>Real</i> . Default=20.
IADJ_LE	Variable no longer used. Here for backwards compatibility. <i>Integer</i>
IADJ_TE	Variable no longer used. Here for backwards compatibility. <i>Integer</i>

F.3.a.v. &pad_grid

Defines the additional upstream and downstream H-grids used to pad the blade-to-blade O-4H grid set.

INLETH	Specifies creation of an upstream H-grid. <i>Logical</i> F No H-grid created. (default) T Generate H-grid
OUTLETH	Specifies creation of a downstream H-grid. <i>Logical</i> F No H-grid created. (default) T Generate H-grid
ILMAX	Total number of grid points in the streamwise direction (<i>i</i> -dir.) that define the upstream H-grid. <i>Integer</i>
IRMAX	Total number of grid points in the streamwise direction (<i>i</i> -dir.) that define the downstream H-grid. <i>Integer</i>

F.3.a.vi. &bounds

Defines the meridional bounds of the blade-to-blade grid blocks, the optional upstream H-grid, and the optional downstream H-grid.

XBH	Axial locations of the grid block boundaries along the hub surface. Vector length of 4. See Fig. F.2a. <i>Real</i>
RBH	Radial locations of the grid block boundaries along the hub surface. Vector length of 4. See Fig. F.2a. <i>Real</i>
XBT	Axial locations of the grid block boundaries along the tip surface. Vector length of 4. See Fig. F.2a. <i>Real</i>
RBT	Radial locations of the grid block boundaries along the tip surface. Vector length of 4. See Fig. F.2a. <i>Real</i>

F.3.a.vii. &ctrls

Defines additional code control, filenames, and boundary condition flags.

CONFIG	Defines the blade-to-blade grid topology. <i>Integer</i> 1 O-H grid set + triangles (defunct) 2 O-4H grid set (default)
IGEOM	Defines the geometry type. <i>Integer</i> 0 Linear cascade geometry 1 Annular cascade geometry (default)
BLADEFIELD	Defines the filename for the given blade geometry file. See Section F.3.c for file format. <i>Character</i> . Default = "rotor.def"
FLOWPATH	Defines the filename for the given flowpath geometry file. See Section F.3.b for file format. <i>Character</i> . Default = "flowpath.def"
UGRID	Defines the filename for the output .ugrid format grid file. See Section F.4.a for file format. <i>Character</i> . Default = "vol.ugrid"
TECFIELD	Defines the filename for the output unstructured Tecplot data file. See Section F.4.c for file format. <i>Character</i> . Default = "rotor.plt"

UPSTRMFILE	Defines the filename for the input freestream velocity distribution definitions file. See Section F.3.d for file format. <i>Character</i> . Default: “upstrmV.dat”
IBPA	Inter-blade phase angle in degrees. Used to define the number of blade passages to be constructed: $n = 360/\text{IBPA}$. A value of 0 results in one blade row. <i>Integer</i> . Default = 0
PITCH	Linear distance between blades in a linear cascade. Used to set the periodic boundaries only for <code>igeom = 0</code> . <i>Real</i> . Default = 1.0
XTRANS	Axial translation factor for the input blade geometry. <i>Real</i> . Default = 0.0
ROTATING	Sets the system rotation parameter and determines the hub boundary condition. <i>Logical</i> F Assumes $\omega_x = 0$; hub boundary condition set using FSURF (default) T Assumes $\omega_x \neq 0$; hub boundary condition set using RSURF
VANE	Set the boundary conditions on the blade and hub surfaces. <i>Logical</i> F Blade surface set using FSURF; hub surface set by ROTATING (default) T Blade surface set using RSURF; hub surface set by RSURF
DEBUG	The entry into each subroutine and other information is printed to screen when activated. <i>Logical</i> . Default = F
DBG_PLT	Logical vector which generates intermediate plot files at key grid generation locations. See Table F.1 for output locations. <i>Logical</i> . Default = F
DBG_BREAK	Causes TGG to exit early after the blade-to-blade grids have been constructed on the meridional grid layers (if <code>dbg_plt(6) = T</code>). Use when initially constructing the grid with <code>K2D = 3</code> or 4 for quick turnaround times. <i>Logical</i> . Default = F
INLET	UNS3D consistent inlet boundary condition flag. <i>Integer</i> . Default = -1
OUTLET	UNS3D consistent outlet boundary condition flag. <i>Integer</i> . Default = -2
FSURF	UNS3D consistent fixed wall boundary condition flag. <i>Integer</i> . Default = -3

RSURF	UNS3D consistent rotating wall boundary condition flag. <i>Integer</i> . Default = -4
SYMMETRY	UNS3D consistent symmetry boundary condition flag. Only used for linear cascade geometries. <i>Integer</i> . Default = -7
MASTER	UNS3D consistent master periodic boundary condition flag. <i>Integer</i> . Default = -100
SLAVE	UNS3D consistent slave periodic boundary condition flag. <i>Integer</i> . Default = 0

F.3.a.viii. Post Namelist Input

Two variables used to control the grid around the “singularity” points, open circles in Fig. F.1, are defined after the last namelist card. The two variables defined are **SINGMAG** and **SINGFADE**. Both variables are used to control how rigidly **SINGDIST** is maintained at the “singularity” points. The two variables are used to specify an exponential decay in the computation space about the singularity point:

$$f_s(\ell) = \text{SINGMAG} \cdot \exp(-\text{SINGFADE } \ell), \quad \ell = 1, 2, \dots, n_{ce}$$

where ℓ is a logical distance and n_{ce} is the number of edge segments connected to the “singularity” point along a continuous edge. The recommended starting value for **SINGMAG** is 0 for all four singularity points. The value should be increased if the surrounding grid looks poor. Values for **SINGFADE** are typically between 0 and 1.

The variables are defined in the input file such that the variable name is given first followed by two lines of four values. **SINGMAG** must come first and the names of the variables must be given otherwise an error will occur. An example of how the two variables must appear in the input file is given below.

```

SINGMAG
  2.00  1.00  1.00  1.00
  3.00  3.00  3.00  3.00
SINGFADE
  0.70  0.70  0.70  0.70

```


0.70 0.50 0.50 0.70

Each line for a given variable corresponds to values for the hub and tip grid layers, with hub values on the first line. The four values for each grid layer correspond to individual “singularity” points. The order corresponds to the Roman numerals in Fig. F.1.

F.3.b. Flow Path Definition

The flow path is defined by the shape of the hub and tip endwalls. For annular geometries, the hub and tip endwall profiles are defined using the meridional coordinates (x, r) , as shown by the open circles in Fig. F.2a. The meridional coordinates of the endwalls are read from an external file, whose name was defined in the main input file by the variable `FLOWPATH` in `&ctrls`. The snippet of FORTRAN code used to read the file is:

```
read(io,*) nph, npt           ! # points hub; # points tip
read(io,*) ( x0h(n), n = 1,nph ) ! hub endwall x-coordinates
read(io,*) ( r0h(n), n = 1,nph ) ! hub endwall r-coordinates
read(io,*) ( x0t(n), n = 1,npt ) ! tip endwall x-coordinates
read(io,*) ( r0t(n), n = 1,npt ) ! tip endwall r-coordinates
```

No external file is required to define the flow path of a linear geometry. The arrays defined in `&bounds` are used to set the flow path for linear geometries. The z coordinate of the hub endwall is hard set to 0. The distance between the hub and tip surfaces are given by: `rbt(2) - rbh(2)`.

F.3.c. Blade Surface Definition

The blade geometry is defined by an external ASCII file, whose name was defined in the main input file by the variable `BLADEFIELD` in `&ctrls`. Cylindrical coordinates are used to define the blade geometry for annular geometries, and Cartesian coordinates for linear geometries.

The file header defines the number of points to be read, the number of blades around the annulus, and the format of the given geometry description. The header of the input file is read using the FORTRAN snippet:

```
read(io,*) nbs, npb, nblade, merid
```

where

<code>nbs</code>	Number of spanwise blade sections used to define the geometry of the blade.
<code>npb</code>	Number of points defining each blade section. Must be a constant number for each section.
<code>nblade</code>	Number of blades around the annulus. Used to define the pitch for annular geometries.
<code>merid</code>	Defines the format of the given geometry description. <ul style="list-style-type: none"> 0 Complete definition of blade sections (including leading and trailing edges). Blade sections are stacked from hub to tip. Grid points are given starting at the trailing edge and then wrapping clockwise around the blade section back to and repeating the trailing edge. This is the only usable option as of V.300. 2 Each blade section is defined by separate upper and lower surface arrays with no leading and trailing edge definitions. Leading and trailing edge circles are added automatically by TGG. Blade sections are stacked from hub to tip, and points go from the leading edge to trailing edge. This option is not fully implemented as of V.300.

After the header, the geometry description is read based on the value of `merid`. For linear geometries y replaces θ and z replaces r . For `merid=0` the blade geometry is read by:

```
do k = 1,nbs
  read(io,*) ( xx(i, k), i = 1,npb) ! x
  read(io,*) ( tt(i, k), i = 1,npb) ! theta (y)
  read(io,*) ( rr(i, k), i = 1,npb) ! radius (z)
```

```
end do
```

For `merid=2` the blade geometry is read by:

```
do k = 1,nbs
  read(io,*) (xbl(i,k), i = 1,npb) ! x (both surfaces)
end do
do k = 1,nbs
  read(io,*) (rbl(i,k), i = 1,npb) ! r (both surfaces)
end do
do k = 1,nbs
  read(io,*) (t1bl(i,k), i = 1,npb) ! theta (upper surface)
end do
do k = 1,nbs
  read(io,*) (t2bl(i,k), i = 1,npb) ! theta (lower surface)
end do
```

F.3.d. Freestream Profile Definition

The radially varying freestream conditions needed to specify a constant y^+ on the blade surface are read from an external ASCII file. The name of the file is given in `&ctrls` by the variable `UPSTRMFILE`. The units of the variables found in the freestream definitions file must match the grid units of the mesh, e.g. m/s with m.

The freestream definition file is made up a namelist card at the top of the file followed by the velocity profile. The namelist card is called `&freestream`, and contains three variables:

<code>nvmax</code>	Number of spanwise velocity values to read. <i>Integer</i> . Default = 2
<code>rho</code>	Freestream density, in grid consistent units. <i>Real</i> . Default = 1.225 kg/m ³
<code>mu</code>	Freestream dynamic viscosity, in grid consistent units. <i>Real</i> . Default = 1.716×10^{-5} kg/(m s)

The file can be read using the following FORTRAN snippet:

```

read(1, freestream)
do n = 1, nvmax
  read(1, *) pspin(n), velin(n)
end do

```

`pspin` is the percent span in percent ($0 \leq \text{pspin} \leq 100$) of the given profile. `velin` is the velocity profile in grid consistent units (m/s, cm/s, ft/s, etc.).

F.4. Output Files

This section describes the relevant output files that are generated by TGG. These files include the resultant grid file, the additional files used by the flow solver (UNS3D), and the different Tecplot data files to visualize the constructed grid.

F.4.a. Unstructured Grid File

The primary output file is the unstructured grid file which completely describes the constructed mesh. The grid file utilizes the UGRID format. The name of the file was given by `UGRID` in the input file. The FORTRAN code used to write the grid file is given below. The FORTRAN code provided does not include the cell-to-node connectivity for tetrahedral and pyramidal cell types as `nctets=0` and `ncpent5=0` is always true.

```

write(io,*) nnode, nbtrias, nbquads, nctets, ncpent5, ncpent6, nchexs
do i = 1, nnode ! cartesian grid coordinates
  write(io,*) x(i), y(i), z(i)
end do
do i = 1, nbtrias ! triangular bndry face connectivity
  write(io,*) (bf2n3(i,j), j = 1,3)
end do
do i = 1, nbquads ! quadrilateral bndry face connectivity
  write(io,*) (bf2n4(i,j), j = 1,4)
end do

```

```

do i = 1,nbtrias ! triangular bndry face types
  write(io,*) bcid3(i)
end do
do i = 1,nbquads ! quadrilateral bndry face types
  write(io,*) bcid4(i)
end do
do i = 1, ncpent6 ! prismatic cell-to-node connectivity
  write(io,*) (c2n6(i,j), j = 1,6)
end do
do i = 1,nchexs ! hexahedral cell-to-node connectivity
  write(io,*) (c2n8(i,j), j = 1,8)
end do

```

F.4.b. Ancillary Flow Solver Files

The flow solver must be provided additional information for annular geometry cases for flow initialization and to simplify boundary condition calculations. Information for flow initialization is contained with the file `XleXteRte`. Either `ijk.def` or `outlet_ij.def` must be present when using the radially varying back pressure option for the outlet boundary condition. The description of the three files now follows.

F.4.b.i. Annular Flow Initialization: XleXteRte

The flow initialization file is an un-formatted file which defines a unit relative velocity vector at each grid point. The direction of the relative velocity vector at a node is set using slope of the mean camber line at the given axial location. The format of the file is given by the following FORTRAN snippet:

```

write(iox) nnode ! number of nodes
do i = 1,nnode
  write(iox) uu(i), vv(i), ww(i) ! unit rel. velocity components
end do

```

F.4.b.ii. Complete Structured-to-Unstructured Pointer: `ijk.def`

The structured nature of the grid is retained by the pointer variable `ijk` within TGG and by the external file `tgg.def`. The purpose of `tgg.def` is to provide the flow solver with the structured grid connectivity so that boundary faces may be quickly identified without excessive logic, computation, or parallel communication. The file can also be used to generate block structured Tecplot data files for solution post-processing.

The format of `ijk.def` is similar to that of a PLOT3D format file. Grid block sizes are given first, followed by arrays of the unstructured node number for a given (i, j, k) . A list of each grid block with additional information, such as type, is given at the very end of the file. The FORTRAN code used to write `ijk.def` is given below.

```
write(io,*) ng, maxval(iimax), maxval(jjmax), maxval(kkmax)
do n = 1,ng
  write(io,*) iimax(n), jjmax(n), kkmax(n)
end do
do n = 1,ng
  write(io,*) &
    (((ijk(i,j,k,n), i=1,iimax(n)), j=1,jjmax(n)), k=1,kkmax(n))
end do
do n = 1,ng
  write(io,*) gtype(n), iimax(n), jjmax(n), kkmax(n), ilu(n), &
    ilt(n), nhub(n), ntip(n), nin(n), nex(n), 1
end do
```

The following is a description of the variables found within `ijk.def`:

<code>ng</code>	Number of grid blocks
<code>iimax</code>	Maximum i grid dimensions for each block
<code>jjmax</code>	Maximum j grid dimensions for each block

<code>kkmax</code>	Maximum k grid dimensions for each block
<code>ijk</code>	Structured-to-unstructured pointer. Gives unstructured node number for a given (i, j, k)
<code>gtype</code>	Specifies the grid block type 1 Inlet or outlet H-grid 10 O-grid in the O-4H grid block set 11 Any H-Grid in the O-4H grid block set 12 Clearance grid H-grid 13 Clearance grid O-Grid
<code>ilu</code>	Not currently used, set to zero
<code>ilt</code>	Not currently used, set to zero
<code>nhub</code>	Links the clearance grids, located adjacent to the hub endwall, with the O-4H O-grid > 0 O-grid block number connected to clearance grids -1 Identifies the clearance H-grid < -1 For clearance O-grids. Gives block number of the connected clearance H-grid. Use abs. value
<code>ntip</code>	Links the clearance grids, located adjacent to the tip endwall, with the O-4H O-grid. The same number system as <code>nhub</code> is used for <code>ntip</code>
<code>nin</code>	Identifies which block(s) the domain inlet is located within 0 No inlet boundary present 333 Inlet boundary is split across three H-grids of the O-4H blade-to-blade block set 999 Inlet boundary is located in the upstream “inlet” H-grid
<code>nex</code>	Identifies which block(s) the domain outlet is located within 0 No outlet boundary present 333 Outlet boundary is split across three H-grids of the O-4H blade-to-blade block set 999 Outlet boundary is located in the downstream “outlet” H-grid

F.4.b.iii. Outlet Structured-to-Unstructured Pointer: `outlet_ij.def`

In some instances the grid may not be fully structured, such as when `config=1` is used. However, the domain outlet is still “structured” as it was built using grid

layers of increasing height and made up of quadrilaterals. Therefore a structured-to-unstructured pointer file for the outlet boundary mesh can be constructed.

The format of the outlet pointer file is identical to that of `ijk.def`, with only one grid “block” being written. The FORTRAN code used to write `outlet_ij.def` is given below followed by a description of the variables found within the file. All hard-coded values should not be modified, else errors will occur within the flow solver.

```
write(io,*) 1, 1, jrmax, kmax
write(io,*) 1, jrmax, kmax
write(io,*) ((( ojk(i,j,k), i=1,1), j=1,jrmax), k=1,kmax)
write(io,100) -2, 1, jrmax, kmax, 0, 0, 0, 0, 0, 0, omx
write(io,110) "gtype", "imax", "jmax", "kmax", "i1", "i2", "nhub", &
             "ntip", "nin", "nex", "om-x"
```

<code>jrmax</code>	Composite maximum j grid dimension
<code>kmax</code>	Maximum k grid dimension
<code>ojk</code>	Gives unstructured node number for given (j, k) on the outlet boundary
<code>gtype</code>	The value “-2” indicates outlet boundary, and must not be changed
<code>omx</code>	Indicates rotating flow – not currently used by flow solver 0 No rotation 1 Rotation present (ROTATING=T or VANE=F)

F.4.c. Unstructured Tecplot Data File

An unstructured Tecplot data file is generated for annular geometries only. The name of the file is defined by the variable `TECFILE` in the `&ctrls` card of the input file. The Tecplot data file contains initial velocity field (what is written to `XleXteRte`). The FORTRAN code used to write the data file is given below.

```
write(iot, *) 'variables= "x", "y", "z", "u", "v", "w", "|V|", "r"'
write(iot,*) 'zone T="hexahedrals", datapacking=point, nodes=', &
```



```

        nnode, ', elements=', nchexs+ncpent6,', zonetype=febrick'

do i = 1,nnode
    write(iot,*) x(i), y(i), z(i), uu(i), vv(i), ww(i), &
        sqrt(uu(i)**2 + vv(i)**2 + ww(i)**2), &
        sqrt(y(i)*y(i) + z(i)*z(i)))
end do

! Write the cell-to-node connectivity for the tecplot file
do i = 1,nchexs ! hexahedral cells
    write(iot,*) (c2n8(i,j), j = 1,8)
end do
do i = 1,ncpent6 ! prismatic cells
    write(iot,*) (c2n6(i,j), j = 1,3), c2n6(i,3), &
        (c2n6(i,j), j = 4,6), c2n6(i,6)
end do

```

F.4.d. Structured Tecplot Data File

A structured block format Tecplot data file is generated for `config=2` grids (both annular and linear). The purpose of the file is to allow the user to examine the each grid block individually. The name of the data file is hard-coded as `04H.plt`. A snippet of FORTRAN code is provided below illustrating how the file is written for a single grid block. Each “zone” segment would be repeated for every grid block in the mesh.

```

open(io, File="04H.plt", status="replace", action="write")
write(io,*) "variables = x, y, z, percent_span"

! The following zone segment is repeated for every grid block
write(io, '(A,A,3(A,I4),A)') 'zone T="' ,trim(title),'", I=', &
    im, ', J=', jm, ', K=', km, ', datapacking=block'
write(io, '(30(G21.13,1x))') x
write(io, '(30(G21.13,1x))') y
write(io, '(30(G21.13,1x))') z
write(io, '(30(G21.13,1x))') p

```

F.4.e. Tecplot Data Files For Debugging

A number of Tecplot data files can be optionally written during the grid generation process for “debugging” purposes. The files can be written by setting the logical index value of `DBG_PLT` to `T`. The available debug Tecplot data files are given in Table F.1.

The most useful is index 6, which generates a data file for the O-4H grids that were generated on the meridional grid layers. Using this option, with `DBG_BREAK`, allows for a quick look at the grid before all of the grid layers are added and output files generated. It is highly suggested that this debug option combination, along with $2 \leq K2D \leq 4$, be used during initial grid construction until the user is satisfied with the resultant grid. Once satisfied, increase `K2D` back to a reasonable number (like 15), and turn off both debug options.

Table F.1: Turbomachinery grid generator debug Tecplot data file descriptions.

Array Index	File Name	File Description
1	<code>blade_input.plt</code>	Input blade geometry as read by TGG. Also includes any coordinate transformations (e.g. $XTRANS \neq 0$)
2	<code>blade_redist.plt</code>	Input blade geometry after the <code>IMAX1</code> points have been added and clustered around each blade section.
3	<code>meridgrid.plt</code>	Meridional grid used to define the flow path and blade geometry
4	<code>blade_inrsct.plt</code>	Blade geometry after intersection with meridional grid
5		Not currently in use
6	<code>merid_blade2blade.plt</code>	Structured O-4H grid blocks generated on the meridional grid layers. A clearance O-H grid layer is generated on either the hub or tip grid layer if either <code>HUBCAP</code> or <code>TIPCAP</code> is T, respectively.
7		Not currently in use
8		Not currently in use

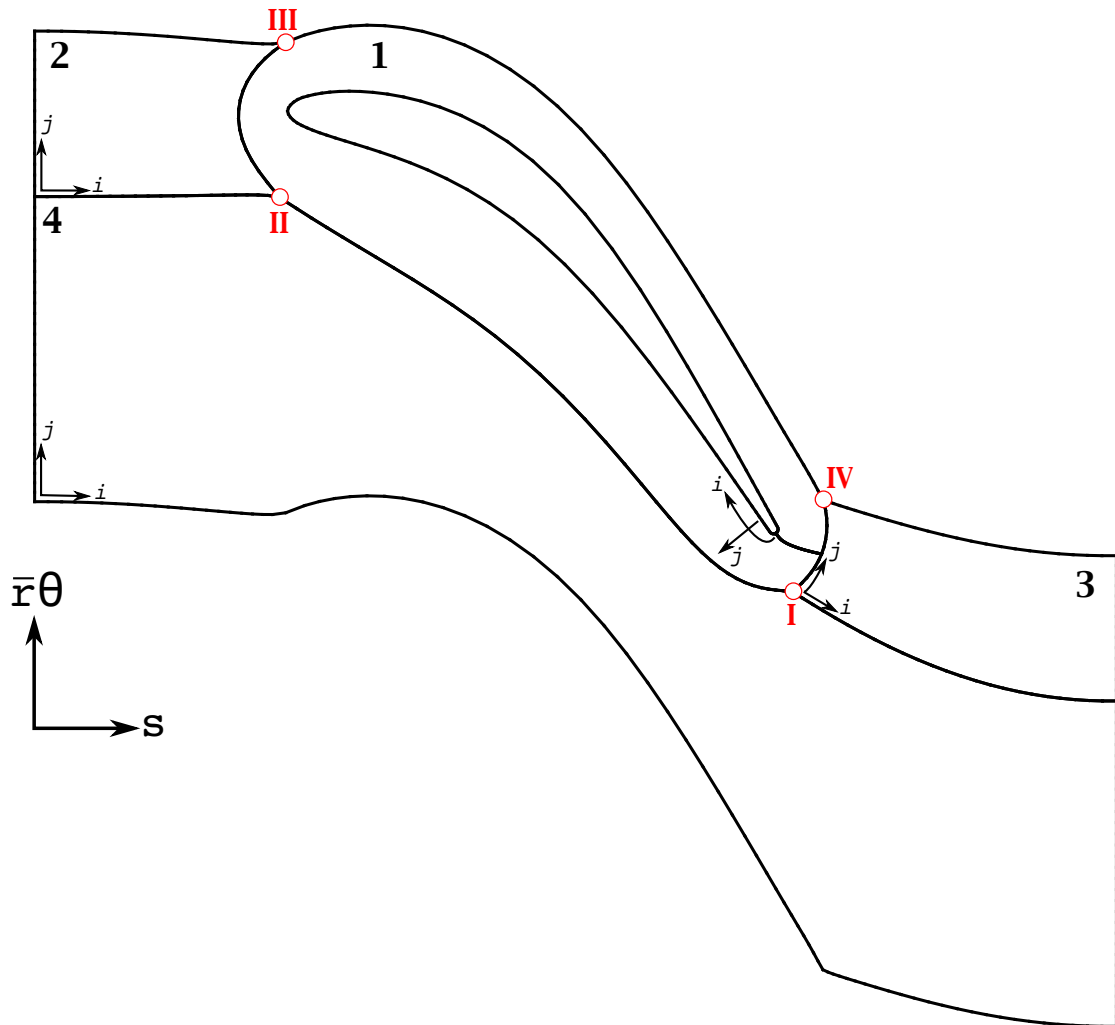
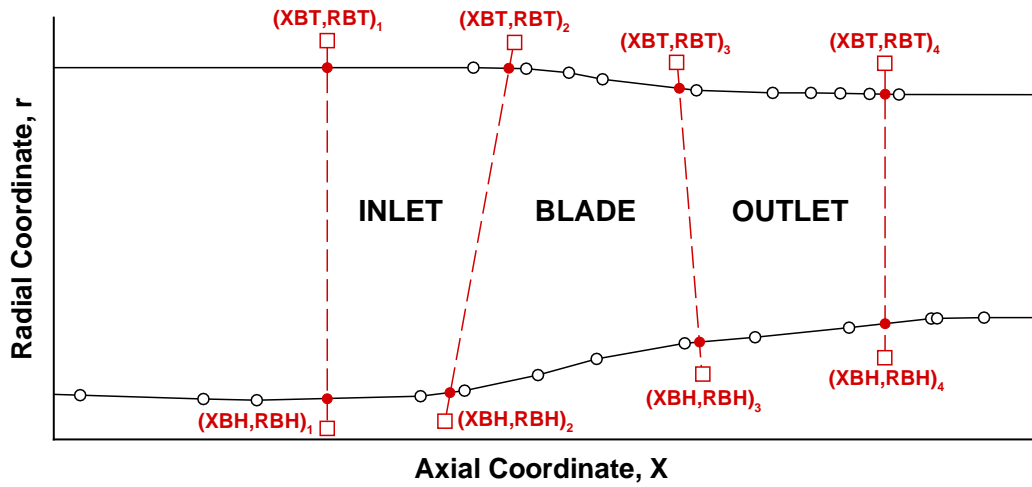
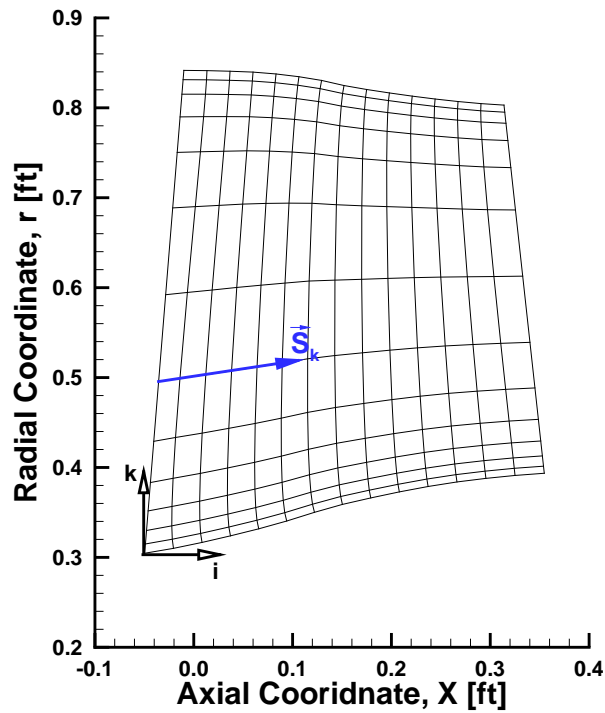


Figure F.1: Blade-to-blade O-4H grid layout. Grid block numbers given by Arabic numerals and grid “singularity” points numbered by Roman numerals. $(i, j) = (1, 1)$ at index origins.



(a) Defining the meridional grid with user prescribed boundaries. “o” define endwall profiles, “—” are user defined grid boundaries, and “•” are intersected boundary points.



(b) Typical meridional grid defining the indicial coordinate system and spanwise arc length distribution vector, \vec{S}_k .

Figure F.2: Defining the meridional grid.

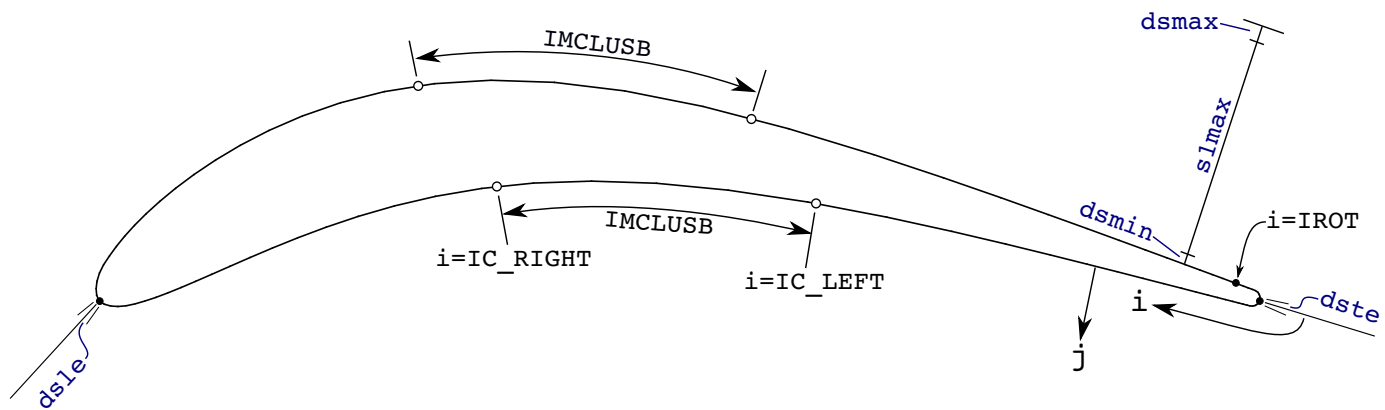
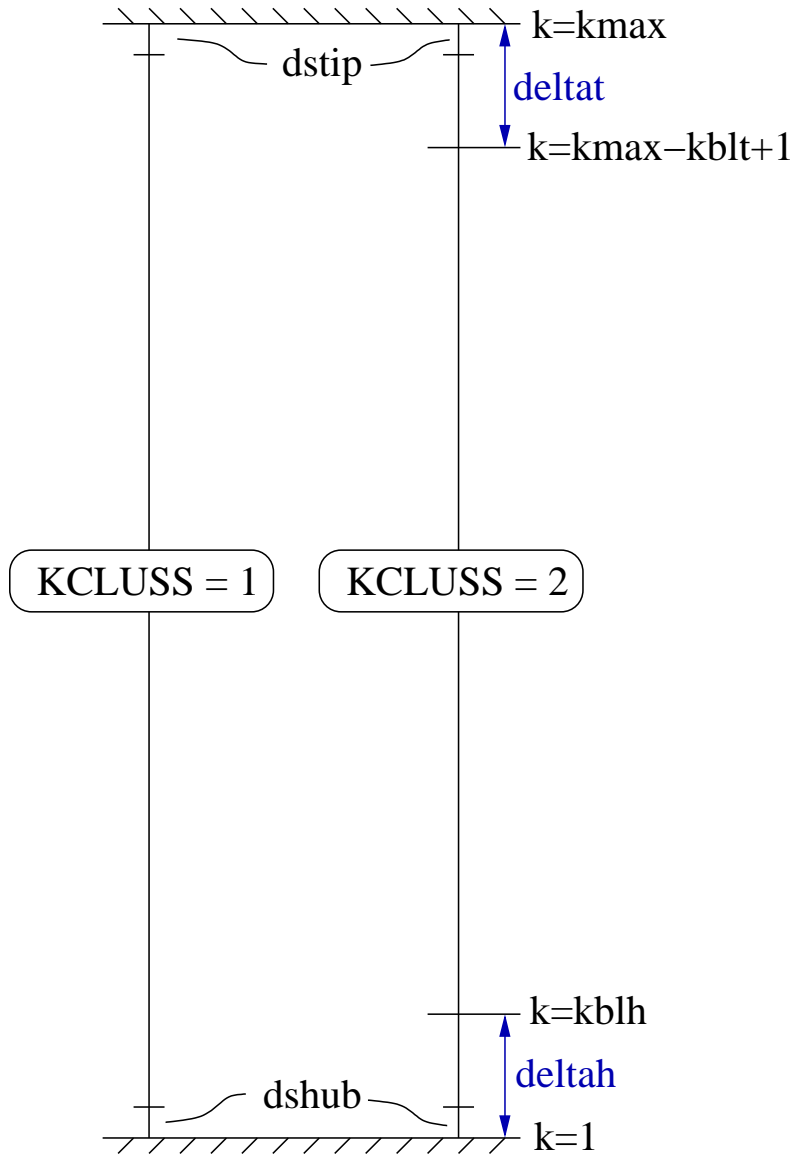
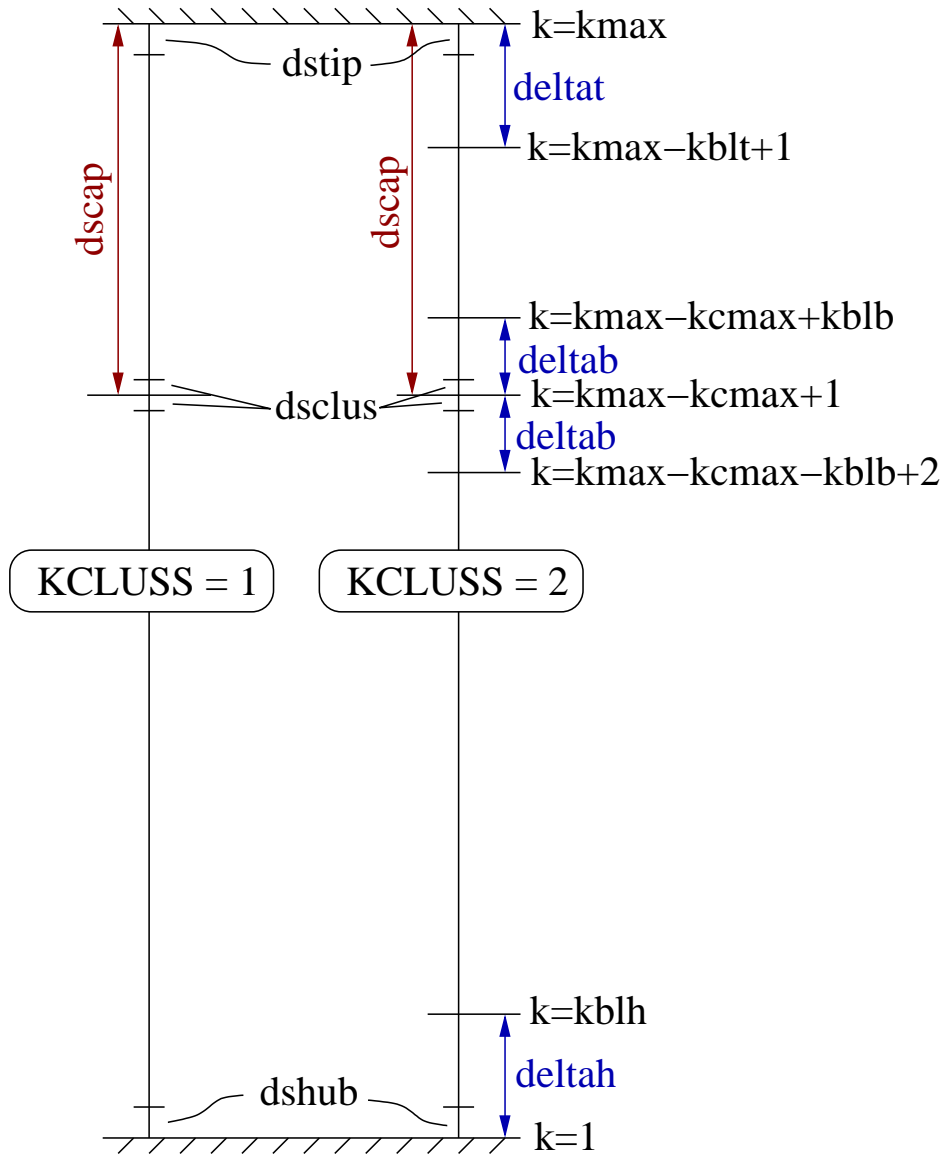


Figure F.3: Grid spacing around an airfoil section.



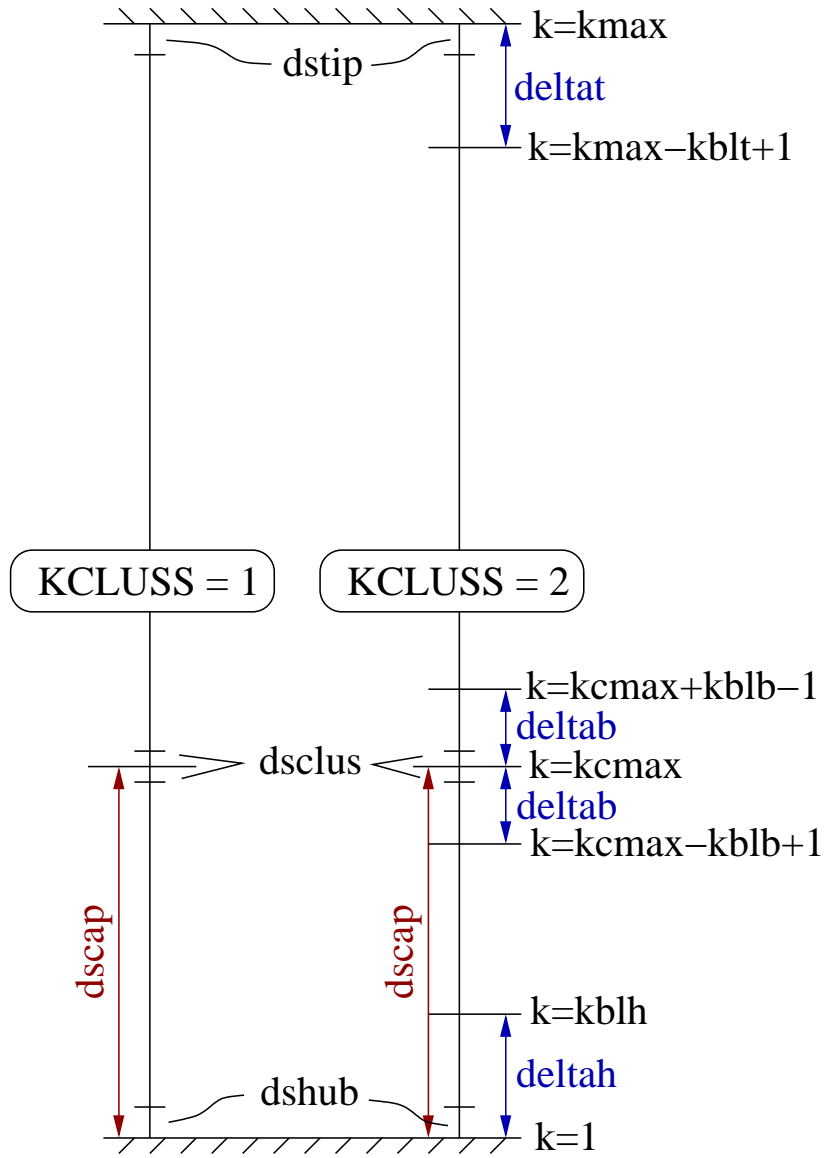
(a) No clearance grids (rubbing).

Figure F.4: Spanwise grid spacing option definitions.



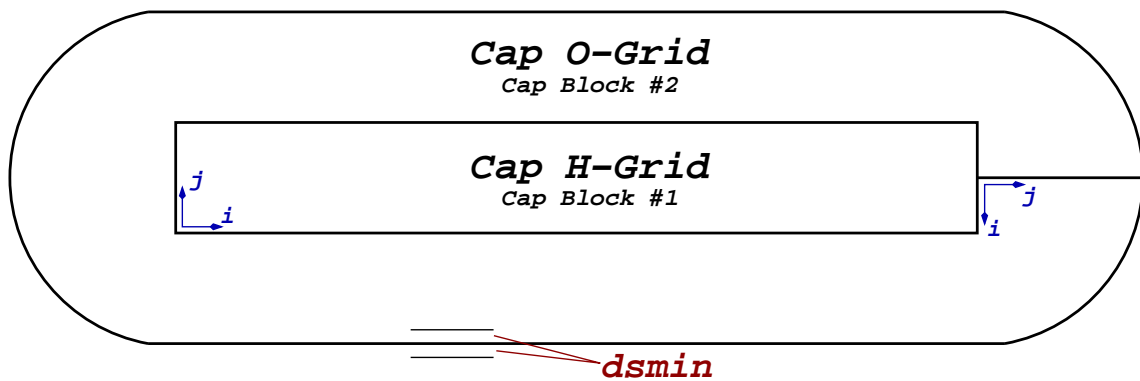
(b) Clearance grids at tip.

Figure F.4: Continued.

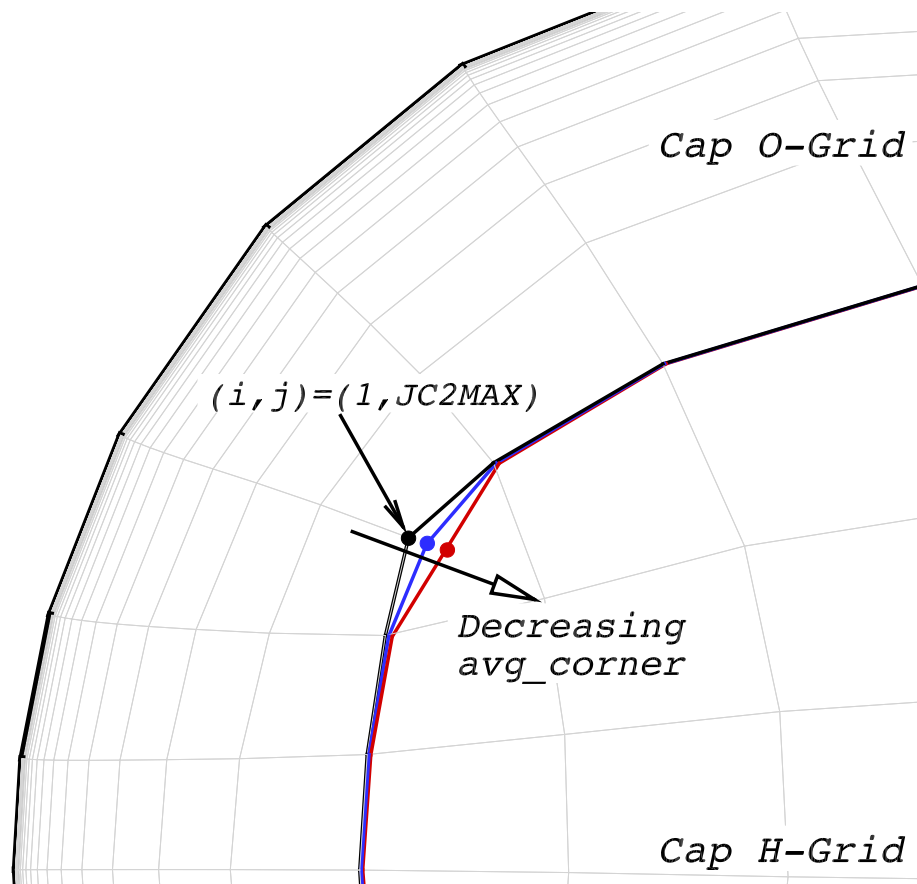


(c) Clearance grids at hub.

Figure F.4: Continued.



(a) Clearance O-H grid layout.



(b) Effect of `avg_corner`. Values of `avg_corner` shown are 0.7, 0.3, and 0.0.

Figure F.5: Defining the clearance O-H grids.