

# ADVANCES IN COMPUTATIONAL FLUID DYNAMICS: TURBULENT SEPARATED FLOWS AND TRANSONIC POTENTIAL FLOWS

By  
Reece E. Neel

A DISSERTATION SUBMITTED TO THE FACULTY OF  
VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY  
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY  
IN  
AEROSPACE ENGINEERING

---

Robert W. Walters, Chairman

---

Roger L. Simpson, Co-Chairman

---

Bernard Grossman, Dept. Head

---

William H. Mason

---

Joseph A. Schetz

August 1997  
Blacksburg, Virginia

Advances in Computational Fluid Dynamics:  
Turbulent Separated Flows and  
Transonic Potential Flows

by

Reece Neel

Committee Chairman: Robert W. Walters

Committee Co-Chairman: Roger L. Simpson

Aerospace Engineering

(ABSTRACT)

Computational solutions are presented for flows ranging from incompressible viscous flows to inviscid transonic flows. The viscous flow problems are solved using the incompressible Navier-Stokes equations while the inviscid solutions are attained using the full potential equation. Results for the viscous flow problems focus on turbulence modeling when separation is present. The main focus for the inviscid results is the development of an unstructured solution algorithm.

The subject dealing with turbulence modeling for separated flows is discussed first. Two different test cases are presented. The first flow is a low-speed converging-diverging duct with a rapid expansion, creating a large separated flow region. The second case is the flow around a stationary hydrofoil subject to small, oscillating hydrofoils. Both cases are computed first in a steady state environment, and then with unsteady flow conditions imposed. A special characteristic of the two problems being studied is the presence of strong adverse pressure gradients leading to flow detachment and separation.

For the flows with separation, numerical solutions are obtained by solving the incompressible Navier-Stokes equations. These equations are solved in a time accurate manner using the method of artificial compressibility. The algorithm used is a finite volume, upwind differencing scheme based on flux-difference splitting of the convective terms. The Johnson and King turbulence model is employed for modeling the

turbulent flow. Modifications to the Johnson and King turbulence model are also suggested. These changes to the model focus mainly on the normal stress production of energy and the strong adverse pressure gradient associated with separating flows. The performance of the Johnson and King model and its modifications, along with the Baldwin-Lomax model, are presented in the results. The modifications had an impact on moving the flow detachment location further downstream, and increased the sensitivity of the boundary layer profile to unsteady flow conditions.

Following this discussion is the numerical solution of the full potential equation. The full potential equation assumes inviscid, irrotational flow and can be applied to problems where viscous effects are small compared to the inviscid flow field and weak normal shocks. The development of a code is presented which solves the full potential equation in a finite volume, cell centered formulation. The unique feature about this code is that solutions are attained on unstructured grids. Solutions are computed in either two or three dimensions. The grid has the flexibility of being made up of tetrahedra, hexahedra, or prisms. The flow regime spans from low subsonic speeds up to transonic flows. For transonic problems, the density is upwinded using a density biasing technique. If lift is being produced, the Kutta-Joukowski condition is enforced for circulation. An implicit algorithm is employed based upon the Generalized Minimum Residual method. To accelerate convergence, the Generalized Minimum Residual method is preconditioned. These and other problems associated with solving the full potential equation on an unstructured mesh are discussed. Results are presented for subsonic and transonic flows over bumps, airfoils, and wings to demonstrate the unstructured algorithm presented here.

# Acknowledgements

I would first like to thank the members of my committee. They have all assisted me in one way or another in completing this work. I would like to personally thank Dr. Walters for taking on the role as my advisor. Thanks to his instruction and insights into CFD, I have not only learned a great deal about the subject, but have come to enjoy this field of research. He has also provided me with encouragement that has allowed me to have confidence in doing the research at hand. I would also like to personally thank Dr. Simpson for taking on the role of co-advisor and for the time that he has invested in me as well. His instruction and insights into turbulent separated flows has expanded my understanding of fluid dynamics greatly. I would like to thank him for the hours of discussion on turbulence modeling and the physics of separated flows, some of which still remains a mystery to me. I would finally like to thank Dr. Mason for the sharing of his knowledge about potential flow.

Next I would like to thank the employees at AeroSoft. They have made my last few years of being a graduate student very memorable. I have come to feel very much a part of the team there. I would especially like to thank Mike Applebaum and Bill McGrory for their time and patience. They have greatly assisted me in my work on unstructured grid technology. I have learned a great deal from these two.

Finally I would like to say thanks to my family. First to my Mom and Dad for all their love and support for me in acquiring this degree. They have remained true to their roles as parents, and for this I will always be indebted to them. Next I would like to thank my fiancée Audrey for her love and encouragement. I look forward to beginning my career with her by my side. And finally, as well as most importantly, I thank God. Any change in me for the better has been a result of Him working in my

life. As a result of my relationship with him, I have come to know true peace and joy that cannot be found elsewhere. Thank you.

# Nomenclature

|               |                                                                          |
|---------------|--------------------------------------------------------------------------|
| $A$           | = cell area                                                              |
| $A^+$         | = van Driest damping constant                                            |
| $D$           | = near wall damping term                                                 |
| $\mathcal{D}$ | = turbulent diffusion rate                                               |
| $F$           | = ratio of total turbulence energy production to shear stress production |
| $K$           | = Clauser modeling constant; set to 0.0168                               |
| $L$           | = dissipation length scale                                               |
| $M$           | = Mach number                                                            |
| $P$           | = precondition matrix                                                    |
| $R$           | = specific gas constant; residual                                        |
| $T$           | = temperature                                                            |
| $V$           | = velocity                                                               |
| $\mathcal{V}$ | = cell volume                                                            |
| $a$           | = speed of sound                                                         |
| $h$           | = enthalpy                                                               |
| $\hat{n}$     | = unit normal vector                                                     |
| $p$           | = pressure                                                               |
| $q$           | = magnitude of the velocity                                              |
| $q^2$         | = turbulent flow property; $u'^2 + v'^2 + w'^2$                          |
| $r$           | = weight value                                                           |
| $s$           | = entropy                                                                |
| $u, v$        | = cartesian velocities in the $x$ and $y$ directions                     |
| $u^*$         | = wall shear velocity                                                    |

|               |                                                                  |
|---------------|------------------------------------------------------------------|
| $u_s$         | = turbulence model velocity scale parameter                      |
| $x, y, z$     | = cartesian coordinates                                          |
| $y^+$         | = $yu^*/\nu$                                                     |
| $\Gamma$      | = circulation                                                    |
| $\beta$       | = artificial compressibility parameter                           |
| $\gamma$      | = Klebanoff's intermittency function; ratio of specific heats    |
| $\delta$      | = boundary layer thickness                                       |
| $\delta^*$    | = boundary layer displacement thickness                          |
| $\theta$      | = contravariant velocity                                         |
| $\kappa$      | = von Karman's constant                                          |
| $\mu$         | = molecular viscosity; artifical dissipation parameter           |
| $\mu_t$       | = turbulent eddy viscosity                                       |
| $\nu$         | = dynamic viscosity; $\mu/\rho$                                  |
| $\rho$        | = density                                                        |
| $\sigma$      | = ratio of actual to equilibrium $\tau_m$                        |
| $\tau$        | = Reynolds shear stress divided by density; $(-\overline{u'v'})$ |
| $\tau_w$      | = wall shear stress                                              |
| $\phi$        | = potential function                                             |
| $\psi$        | = phase angle                                                    |
| $\Delta t$    | = physical time step                                             |
| $\Delta \tau$ | = pseudo-time step                                               |

Subscripts:

|      |                                                |
|------|------------------------------------------------|
| $e$  | = boundary layer edge conditions               |
| $eq$ | = equilibrium conditions                       |
| $f$  | = value at cell face                           |
| $i$  | = inner part of boundary layer                 |
| $m$  | = values of quantity where $\tau$ is a maximum |
| $o$  | = outer part of boundary layer                 |
| $s$  | = stagnation value                             |
| $x$  | = derivative with respect to $x$               |
| $y$  | = derivative with respect to $y$               |

$z$  = derivative with respect to  $z$

$t$  = derivative with respect to  $t$

$\infty$  = freestream value

Superscripts:

$k$  = polynomial degree

$m$  = pseudo-time level

$n$  = physical time level

$()'$  = fluctuating quantity

$(\bar{\phantom{x}})$  = cell average; cell center value

# Contents

|                                                    |           |
|----------------------------------------------------|-----------|
| <b>Abstract</b>                                    | <b>ii</b> |
| <b>Acknowledgements</b>                            | <b>iv</b> |
| <b>Nomenclature</b>                                | <b>vi</b> |
| <b>1 Introduction</b>                              | <b>1</b>  |
| <b>2 Introduction to Turbulent Separated Flows</b> | <b>6</b>  |
| 2.1 Overview . . . . .                             | 6         |
| 2.2 Turbulence Models . . . . .                    | 8         |
| 2.3 Johnson and King Turbulence Model . . . . .    | 11        |
| 2.4 Physics of Separated Flows . . . . .           | 13        |
| <b>3 Incompressible Flow Solver</b>                | <b>17</b> |
| 3.1 Introduction . . . . .                         | 17        |
| 3.2 Governing Equations . . . . .                  | 19        |
| 3.3 Non-dimensionalization . . . . .               | 22        |
| 3.4 Artificial Compressibility Method . . . . .    | 23        |
| 3.5 Flux Scheme . . . . .                          | 23        |
| 3.6 Time Integration . . . . .                     | 26        |
| 3.7 Linear Solvers . . . . .                       | 30        |
| 3.8 Boundary Conditions . . . . .                  | 31        |
| 3.9 Test Cases . . . . .                           | 34        |

|          |                                              |           |
|----------|----------------------------------------------|-----------|
| 3.9.1    | Circular Arc Bump . . . . .                  | 34        |
| 3.9.2    | NACA 0012 Airfoil . . . . .                  | 34        |
| 3.9.3    | One Dimensional Channel . . . . .            | 39        |
| 3.9.4    | Laminar Flat Plate . . . . .                 | 41        |
| <b>4</b> | <b>Turbulence Models</b>                     | <b>43</b> |
| 4.1      | Introduction . . . . .                       | 43        |
| 4.2      | Baldwin-Lomax Model . . . . .                | 43        |
| 4.3      | Johnson and King Model . . . . .             | 45        |
| 4.3.1    | Formulation . . . . .                        | 46        |
| 4.3.2    | Solution to ODE . . . . .                    | 48        |
| 4.3.3    | Implementation of Model . . . . .            | 49        |
| 4.4      | Modifications to the JKM . . . . .           | 50        |
| 4.4.1    | Outer Region . . . . .                       | 51        |
| 4.4.2    | Momentum Equation . . . . .                  | 52        |
| 4.4.3    | Inner Region . . . . .                       | 53        |
| 4.5      | Verification: Turbulent Flat Plate . . . . . | 56        |
| <b>5</b> | <b>Low-Speed Diffuser Results</b>            | <b>59</b> |
| 5.1      | Description of Flow . . . . .                | 59        |
| 5.2      | Steady Flow Case . . . . .                   | 62        |
| 5.3      | Unsteady Flow Case . . . . .                 | 66        |
| 5.4      | Conclusions . . . . .                        | 68        |
| <b>6</b> | <b>Flapping Foil Results</b>                 | <b>87</b> |
| 6.1      | Description of Flow . . . . .                | 87        |
| 6.2      | Computational Aspects . . . . .              | 89        |
| 6.3      | Steady Flow Case . . . . .                   | 93        |
| 6.4      | Unsteady Flow Case . . . . .                 | 95        |
| 6.5      | Conclusions . . . . .                        | 99        |

|          |                                                      |            |
|----------|------------------------------------------------------|------------|
| <b>7</b> | <b>Introduction to Potential Flow</b>                | <b>121</b> |
| 7.1      | Background of Potential Flow in CFD . . . . .        | 121        |
| 7.2      | Numerics of Potential Flow in CFD . . . . .          | 126        |
| 7.3      | Present Formulation and Objectives . . . . .         | 129        |
| <b>8</b> | <b>Potential Flow Solver</b>                         | <b>132</b> |
| 8.1      | Introduction . . . . .                               | 132        |
| 8.2      | Governing Equations . . . . .                        | 133        |
| 8.3      | Finite Volume Formulation . . . . .                  | 138        |
| 8.4      | Flux Calculation . . . . .                           | 139        |
| 8.4.1    | Method 1: Gauss Divergence . . . . .                 | 139        |
| 8.4.2    | Method 2: Modified Gauss Divergence . . . . .        | 140        |
| 8.4.3    | Method 3: K-exact . . . . .                          | 141        |
| 8.4.4    | Method 4: K-exact Least Squares . . . . .            | 142        |
| 8.5      | Artificial Viscosity . . . . .                       | 147        |
| 8.6      | Explicit Algorithms . . . . .                        | 149        |
| 8.7      | Implicit Algorithms . . . . .                        | 150        |
| 8.7.1    | Point Jacobi . . . . .                               | 151        |
| 8.7.2    | GMRES . . . . .                                      | 151        |
| 8.8      | Formulation of the Jacobian Matrix . . . . .         | 155        |
| 8.9      | Precondition Matrix . . . . .                        | 156        |
| 8.10     | Kutta Condition and Circulation . . . . .            | 161        |
| 8.11     | Boundary Conditions . . . . .                        | 163        |
| <b>9</b> | <b>Potential Flow Results</b>                        | <b>167</b> |
| 9.1      | Introduction . . . . .                               | 167        |
| 9.2      | Comparison of Gradient Schemes . . . . .             | 168        |
| 9.3      | Effect of Weights Used in Least Squares . . . . .    | 170        |
| 9.4      | Effect of Mach Number on Convergence . . . . .       | 173        |
| 9.5      | Optimal GMRES Parameters . . . . .                   | 176        |
| 9.6      | Full Potential and Euler Comparison . . . . .        | 182        |
| 9.7      | NACA 0012, NACA 4412, and LNV109A Airfoils . . . . . | 186        |

|                                                               |            |
|---------------------------------------------------------------|------------|
| 9.8 ONERA M6 Wing . . . . .                                   | 208        |
| <b>10 Conclusions</b>                                         | <b>220</b> |
| <b>A Details of the JKM Differential Equation</b>             | <b>225</b> |
| <b>B Hytopoulos Turbulence Model</b>                          | <b>229</b> |
| <b>C Minimization of <math>J(y)</math> in GMRES Algorithm</b> | <b>232</b> |
| <b>D Ghost Cell Location</b>                                  | <b>235</b> |

# List of Figures

|     |                                                                                                                                                                                     |    |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.1 | Interpolation of right and left states at a cell face . . . . .                                                                                                                     | 25 |
| 3.2 | Illustration of line Gauss-Seidel sweep algorithm. . . . .                                                                                                                          | 32 |
| 3.3 | Pressure contours for incompressible flow over a circular arc bump of $t/c = 0.042$ . . . . .                                                                                       | 35 |
| 3.4 | $C_p$ comparison between incompressible Euler and Full Potential solutions for circular arc bump of $t/c = 0.042$ . . . . .                                                         | 36 |
| 3.5 | $C_p$ comparison between incompressible Euler and experiment at $0^\circ$ and $4^\circ$ angle of attack for NACA 0012 airfoil. Experimental data taken from Reference [1] . . . . . | 37 |
| 3.6 | Pressure contours for NACA 0012 airfoil at $4^\circ$ angle of attack for the incompressible Euler equations. . . . .                                                                | 38 |
| 3.7 | Velocity values in a 1-D channel with oscillating back pressure for the incompressible Euler equations using second order time accuracy. . .                                        | 40 |
| 3.8 | Comparison of incompressible Navier-Stokes with Blasius solution for laminar flow over a flat plate. . . . .                                                                        | 42 |
| 4.1 | Wall law comparison of turbulence models with the analytical solution for a turbulent flat plate at $Re_\theta = 9,650$ . . . . .                                                   | 58 |
| 5.1 | Sketch of diffuser computational domain. . . . .                                                                                                                                    | 62 |
| 5.2 | Boundary layer thickness for steady diffuser case. . . . .                                                                                                                          | 69 |
| 5.3 | Boundary layer edge velocity for steady diffuser case. . . . .                                                                                                                      | 70 |
| 5.4 | Boundary layer displacement thickness for steady diffuser case. . . . .                                                                                                             | 71 |
| 5.5 | Friction coefficient for steady diffuser case. . . . .                                                                                                                              | 72 |

|      |                                                                                                                                                                         |    |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 5.6  | Friction coefficient for steady diffuser case. Comparison with other models. . . . .                                                                                    | 73 |
| 5.7  | $F$ factor, maximum Reynolds shear stress, and wall shear velocity along diffuser for the steady case. . . . .                                                          | 74 |
| 5.8  | Comparison of the steady diffuser boundary layer profile at 3.01 meters with several other models. . . . .                                                              | 75 |
| 5.9  | Comparison of the steady diffuser boundary layer profile at 3.01 meters.                                                                                                | 76 |
| 5.10 | Semi-log scale comparison of the steady diffuser boundary layer profile at 3.01 meters. . . . .                                                                         | 76 |
| 5.11 | Comparisons of the steady diffuser boundary layer profile at 3.42 meters.                                                                                               | 77 |
| 5.12 | Semi-log scale comparison of the steady diffuser boundary layer profile at 3.42 meters . . . . .                                                                        | 77 |
| 5.13 | Comparisons of the steady diffuser boundary layer profile at 3.68 meters.                                                                                               | 78 |
| 5.14 | Semi-log scale comparison of the steady diffuser boundary layer profile at 3.68 meters. . . . .                                                                         | 78 |
| 5.15 | Comparisons of the steady diffuser boundary layer profile at 3.97 meters.                                                                                               | 79 |
| 5.16 | Semi-log scale comparison of the steady diffuser boundary layer profile at 3.97 meters. . . . .                                                                         | 79 |
| 5.17 | Values of $\sigma(x)$ for the steady diffuser flow used in scaling the outer eddy viscosity. . . . .                                                                    | 80 |
| 5.18 | Comparison of experimental mass flux at inlet and exit planes for the unsteady diffuser case. Flux is nondimensionalized by inlet area and mean inlet velocity. . . . . | 80 |
| 5.19 | Boundary layer edge velocity for one complete cycle at 3.46 meters. .                                                                                                   | 81 |
| 5.20 | Unsteady velocity profiles at 3.05 meters for six phases in the cycle. Note the displaced ordinates. Legend is same as Figure 5.19 . . . . .                            | 82 |
| 5.21 | Unsteady velocity profiles at 3.55 meters. Note the displaced ordinates. Legend is same as Figure 5.19 . . . . .                                                        | 83 |
| 5.22 | Unsteady velocity profiles at 3.96 meters. Note the displaced ordinates. Legend is same as Figure 5.19 . . . . .                                                        | 84 |

|      |                                                                                                                                                  |     |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 5.23 | Shape factor showing hysteresis effects at 2.85 meter. Legend is same as Figure 5.19 . . . . .                                                   | 85  |
| 5.24 | Shape factor showing hysteresis effects at 3.06 meters. . . . .                                                                                  | 85  |
| 5.25 | Shape factor showing hysteresis effects at 3.51 meters. . . . .                                                                                  | 86  |
| 5.26 | Shape factor showing hysteresis effects at 3.97 meters. . . . .                                                                                  | 86  |
| 6.1  | Diagram of the flapping foil experiment. . . . .                                                                                                 | 88  |
| 6.2  | Values of $U_e$ along the hydrofoil using the baseline Johnson and King turbulence model for the grid convergence study on the steady flow case. | 101 |
| 6.3  | Upper surface boundary layer displacement thickness along hydrofoil for the grid convergence study. . . . .                                      | 102 |
| 6.4  | Comparison of boundary layer profile for different grids at $x/c = 0.972$ using the baseline Johnson and King turbulence model. . . . .          | 103 |
| 6.5  | O-Grid used in computing the flapping foil experiment for both the steady and unsteady cases ( $121 \times 101$ ). . . . .                       | 104 |
| 6.6  | Comparison of $C_p$ for the steady flow flapping foil test case. . . . .                                                                         | 104 |
| 6.7  | Values of $C_f$ on suction side of hydrofoil (steady case). . . . .                                                                              | 105 |
| 6.8  | Values of $\sigma(x)$ on the suction side of the hydrofoil used in scaling the outer eddy viscosity for the two Johnson and King models. . . . . | 106 |
| 6.9  | Steady case boundary layer profile at $x/c = 0.388$ for the flapping foil calculations. . . . .                                                  | 107 |
| 6.10 | Steady case boundary layer profile at $x/c = 0.612$ . . . . .                                                                                    | 107 |
| 6.11 | Steady case boundary layer profile at $x/c = 0.700$ . . . . .                                                                                    | 108 |
| 6.12 | Steady case boundary layer profile at $x/c = 0.900$ . . . . .                                                                                    | 108 |
| 6.13 | Steady case boundary layer profile at $x/c = 0.950$ . . . . .                                                                                    | 109 |
| 6.14 | Steady case boundary layer profile at $x/c = 0.972$ . . . . .                                                                                    | 109 |
| 6.15 | Steady case boundary layer profile at $x/c = 0.990$ . . . . .                                                                                    | 110 |
| 6.16 | Steady case boundary layer profile at $x/c = 1.000$ . . . . .                                                                                    | 110 |
| 6.17 | Semi-log scale boundary layer profile for the steady case at $x/c = 1.000$ .                                                                     | 111 |
| 6.18 | Mean $u$ velocity data on top bounding box ( $y/c = 0.2187$ ) for the unsteady flapping foil case. . . . .                                       | 112 |
| 6.19 | Unsteady harmonic data for $u$ velocity data on top bounding box. . .                                                                            | 112 |

|      |                                                                                                             |     |
|------|-------------------------------------------------------------------------------------------------------------|-----|
| 6.20 | Mean $v$ velocity data on top bounding box ( $y/c = 0.2187$ ) for the unsteady flapping foil case. . . . .  | 113 |
| 6.21 | Unsteady harmonic data for $v$ velocity data on top bounding box. . . . .                                   | 113 |
| 6.22 | Mean $C_p$ distribution along hydrofoil for the unsteady flapping foil case. . . . .                        | 114 |
| 6.23 | Unsteady harmonic data for $C_p$ on suction side of hydrofoil. . . . .                                      | 114 |
| 6.24 | Unsteady case mean boundary layer profiles at $x/c = 0.388$ . . . . .                                       | 115 |
| 6.25 | Unsteady case mean boundary layer profiles at $x/c = 0.612$ . . . . .                                       | 115 |
| 6.26 | Unsteady case mean boundary layer profiles at $x/c = 0.900$ . . . . .                                       | 116 |
| 6.27 | Unsteady case mean boundary layer profiles at $x/c = 0.972$ . . . . .                                       | 116 |
| 6.28 | Unsteady case mean boundary layer profiles at $x/c = 0.990$ . . . . .                                       | 117 |
| 6.29 | Unsteady case mean boundary layer profiles at $x/c = 1.000$ . . . . .                                       | 117 |
| 6.30 | Unsteady harmonic data from Fourier analysis of boundary layer profiles at $x/c = 0.388$ . . . . .          | 118 |
| 6.31 | Unsteady harmonic data from Fourier analysis of boundary layer profiles at $x/c = 0.612$ . . . . .          | 118 |
| 6.32 | Unsteady harmonic data from Fourier analysis of boundary layer profiles at $x/c = 0.900$ . . . . .          | 119 |
| 6.33 | Unsteady harmonic data from Fourier analysis of boundary layer profiles at $x/c = 0.972$ . . . . .          | 119 |
| 6.34 | Unsteady harmonic data from Fourier analysis of boundary layer profiles at $x/c = 0.990$ . . . . .          | 120 |
| 6.35 | Unsteady harmonic data from Fourier analysis of boundary layer profiles at $x/c = 1.000$ . . . . .          | 120 |
| 7.1  | Percent error of stagnation pressure loss as a function of upstream Mach number for a normal shock. . . . . | 129 |
| 8.1  | The different types of faces and cells in the code. . . . .                                                 | 133 |
| 8.2  | Illustration of vectors $\vec{s}$ and $\hat{n}_s$ . . . . .                                                 | 141 |
| 8.3  | Stencil selection for computing gradients at a cell face. . . . .                                           | 144 |
| 8.4  | Upwind cell and density gradients used in density biasing. . . . .                                          | 149 |
| 8.5  | An unstructured triangular grid consisting of 108 cells. . . . .                                            | 158 |

|      |                                                                                                                                                                                                                          |     |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 8.6  | Ordering of the Jacobian matrix based upon the grid generator (general sparse matrix). . . . .                                                                                                                           | 158 |
| 8.7  | Order of the Jacobian matrix based upon the Cuthill-McKee method (minimizes bandwidth). . . . .                                                                                                                          | 159 |
| 8.8  | Order of the Jacobian matrix in order to cluster terms near the main diagonal. . . . .                                                                                                                                   | 160 |
| 8.9  | Illustration of how the Kutta condition is applied. . . . .                                                                                                                                                              | 163 |
| 8.10 | Coordinate transformation to determine the Kutta plane. . . . .                                                                                                                                                          | 164 |
| 8.11 | Boundary ghost cell and stencil for flux calculation. . . . .                                                                                                                                                            | 165 |
| 9.1  | Quadrilateral grid for subsonic bump case with dimensions $81 \times 33$ . .                                                                                                                                             | 169 |
| 9.2  | Triangular grid for subsonic bump case consisting of 2244 cells. . . .                                                                                                                                                   | 170 |
| 9.3  | Comparison of different gradient schemes for subsonic bump. (quadrilateral grid) . . . . .                                                                                                                               | 171 |
| 9.4  | Comparison of different gradient schemes for subsonic bump. (triangular grid) . . . . .                                                                                                                                  | 172 |
| 9.5  | $C_p$ comparison of NACA 4412 airfoil for different weights on gradient calculation. Flow condition: $M = 0.50$ at $0^\circ$ . Grid consists of 3,364 triangular cells with 121 cells on the airfoil surface. . . . .    | 174 |
| 9.6  | Residual comparison of NACA 4412 airfoil for different weights on gradient calculation. Flow condition: $M = 0.50$ at $0^\circ$ . Grid consists of 3,364 triangular cells with 121 cells on the airfoil surface. . . . . | 175 |
| 9.7  | Circular arc bump mesh consisting of 1,861 cells with 40 cells on the arc bump. Arc bump has a radius of 3 units. . . . .                                                                                                | 177 |
| 9.8  | Residual comparison of circular arc bump for different Mach numbers. Grid consists of 1,861 triangular cells. . . . .                                                                                                    | 178 |
| 9.9  | $C_p$ results for circular arc bump at different Mach numbers. Grid consists of 1,861 triangular cells. . . . .                                                                                                          | 179 |
| 9.10 | Comparison of UPC and <i>GUST</i> solutions for circular arc bump at $M = 0.85$ . Grid consists of 1,861 triangular cells, with 40 cells on the bump. . . . .                                                            | 180 |

|      |                                                                                                                                                 |     |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 9.11 | Convergence of NACA 0012 airfoil ( $0^\circ$ ; $M=0.70$ ) for various bandwidth sizes of the precondition matrix. . . . .                       | 181 |
| 9.12 | Comparison of INS and UPC at three grid levels for NACA 0012 airfoil. Flow conditions: $0^\circ$ and $M = 0.10$ . . . . .                       | 187 |
| 9.13 | Comparison of INS and UPC at three grid levels for NACA 0012 airfoil. Flow conditions: $1.5^\circ$ and $M = 0.10$ . . . . .                     | 188 |
| 9.14 | Normalized residual versus iteration number for NACA 0012 airfoil case at $M = 0.10$ and $1.5^\circ$ . Plot corresponds to Figure 9.13. . . . . | 189 |
| 9.15 | Normalized residual versus CPU time for NACA 0012 airfoil case at $M = 0.10$ and $1.5^\circ$ . Plot corresponds to Figure 9.13. . . . .         | 190 |
| 9.16 | Comparison of <i>GUST</i> and UPC at three grid levels for NACA 0012. Flow conditions: $0^\circ$ and $M = 0.50$ . . . . .                       | 191 |
| 9.17 | Comparison of <i>GUST</i> and UPC at three grid levels for NACA 0012. Flow conditions: $1.5^\circ$ and $M = 0.50$ . . . . .                     | 192 |
| 9.18 | Normalized residual versus iteration number for NACA 0012 airfoil case at $M = 0.50$ and $0^\circ$ . Plot corresponds to Figure 9.16. . . . .   | 193 |
| 9.19 | Normalized residual versus CPU time for NACA 0012 airfoil case at $M = 0.50$ and $0^\circ$ . Plot corresponds to Figure 9.16. . . . .           | 194 |
| 9.20 | Normalized residual versus iteration number for NACA 0012 airfoil case at $M = 0.50$ and $1.5^\circ$ . Plot corresponds to Figure 9.17. . . . . | 195 |
| 9.21 | Normalized residual versus CPU time for NACA 0012 airfoil case at $M = 0.50$ and $1.5^\circ$ . Plot corresponds to Figure 9.17. . . . .         | 196 |
| 9.22 | Comparison of <i>GUST</i> and UPC at three grid levels for NACA 0012. Flow conditions: $0^\circ$ and $M = 0.80$ . . . . .                       | 197 |
| 9.23 | Comparison of <i>GUST</i> and UPC at three grid levels for NACA 0012. Flow conditions: $1.5^\circ$ and $M = 0.75$ . . . . .                     | 198 |
| 9.24 | Normalized residual versus iteration number for NACA 0012 airfoil case at $M = 0.80$ and $0^\circ$ . Plot corresponds to Figure 9.22. . . . .   | 199 |
| 9.25 | Normalized residual versus CPU time for NACA 0012 airfoil case at $M = 0.80$ and $0^\circ$ . Plot corresponds to Figure 9.22. . . . .           | 200 |

|      |                                                                                                                                                                                 |     |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 9.26 | Normalized residual versus iteration number for NACA 0012 airfoil case at $M = 0.75$ and $1.5^\circ$ . Plot corresponds to Figure 9.23. . . . .                                 | 201 |
| 9.27 | Normalized residual versus CPU time for NACA 0012 airfoil case at $M = 0.75$ and $1.5^\circ$ . Plot corresponds to Figure 9.23. . . . .                                         | 202 |
| 9.28 | Grid convergence for the potential flow solver on grids with triangles for a NACA 0012 airfoil at $1.5^\circ$ and $M = 0.75$ . . . . .                                          | 204 |
| 9.29 | $C_p$ comparison of Euler and potential flow solvers for a NACA 0012 airfoil at $1.5^\circ$ and $M = 0.75$ . . . . .                                                            | 205 |
| 9.30 | Unstructured grid used for solutions to the NACA 0012 airfoil. Grid contains 3,752 cells with 140 cells on airfoil surface. . . . .                                             | 206 |
| 9.31 | $C_p$ results for a NACA 0012 airfoil at $4^\circ$ and $M = 0.50$ . UPC and <i>GUST</i> use a grid consisting of 3,752 cells. . . . .                                           | 207 |
| 9.32 | Unstructured grid used for solutions to the NACA 4412 airfoil. Grid contains 3,678 cells with 140 cells on airfoil surface. . . . .                                             | 209 |
| 9.33 | $C_p$ results for NACA 4412 airfoil at $3^\circ$ and $M = 0.50$ . Grid contains 3,678 triangular cells with 140 cells on surface for the UPC and <i>GUST</i> solutions. . . . . | 210 |
| 9.34 | Unstructured grid used for solutions to the LNV109A airfoil. Grid contains 3,955 cells. . . . .                                                                                 | 211 |
| 9.35 | $C_p$ results for LNV109A airfoil at $2^\circ$ and $M = 0.50$ . . . . .                                                                                                         | 212 |
| 9.36 | $C_p$ results for LNV109A airfoil at $0^\circ$ and $M = 0.60$ . . . . .                                                                                                         | 213 |
| 9.37 | Results for ONERA M6 wing at 20% semi-span. Flow conditions: $M = 0.84$ at $3.06^\circ$ angle of attack. . . . .                                                                | 214 |
| 9.38 | Results for ONERA M6 wing at 44% semi-span. Flow conditions: $M = 0.84$ at $3.06^\circ$ angle of attack. . . . .                                                                | 215 |
| 9.39 | Results for ONERA M6 wing at 65% semi-span. Flow conditions: $M = 0.84$ at $3.06^\circ$ angle of attack. . . . .                                                                | 216 |
| 9.40 | Results for ONERA M6 wing at 80% semi-span. Flow conditions: $M = 0.84$ at $3.06^\circ$ angle of attack. . . . .                                                                | 217 |
| 9.41 | Results for ONERA M6 wing at 90% semi-span. Flow conditions: $M = 0.84$ at $3.06^\circ$ angle of attack. . . . .                                                                | 218 |

|      |                                                                                                                     |     |
|------|---------------------------------------------------------------------------------------------------------------------|-----|
| 9.42 | Results for ONERA M6 wing at 95% semi-span. Flow conditions:<br>$M = 0.84$ at $3.06^\circ$ angle of attack. . . . . | 219 |
| D.1  | Terminology for finding ghost cell center location. . . . .                                                         | 236 |

# Chapter 1

## Introduction

The computer has become a very valuable resource, and the continual increase in its performance gives great promises for the future. One of the areas that the computer has had a large impact in is fluid dynamics. Great progress has been made over the past several decades to incorporate computers into the prediction of fluid flow. The flow field around a complete aircraft configuration can now be calculated with the help of computers. The acronym CFD (computational fluid dynamics) represents numerical solutions to fluid problems by solving some form of the governing equations of fluid motion. Most complex engineering problems involve analysis from both CFD and experimental testing. The use of both of these approaches allows a more complete and detailed view of the flow field, enhancing the final solution. At the present growth rate of technology, research in both of these areas will continue to remain important and necessary in both the design and analysis of engineering problems.

The complete Navier-Stokes (N-S) equations are considered to be the correct mathematical description of the governing equations of fluid motion. The most accurate numerical computations in fluid dynamics come from solving the Navier-Stokes equations. The equations represent the conservation of mass, momentum, and energy. In three dimensions, the conservation of momentum is written in each of the coordinate directions, giving a total of five equations to be solved. These equations are highly coupled and non-linear. Approximations to the Navier-Stokes equations are made whenever possible. For certain types of flows, these approximations can

be made without compromising the physical modeling of the problem. Some of the various types of approximations will now be discussed.

For laminar flows, solutions to the Navier-Stokes equations are considered to be as accurate as numerical computations can be. But unfortunately, most problems where viscous effects are important are classified as turbulent flows. The unsteady Navier-Stokes equations have the ability to resolve all the small scale structure of turbulent flow. The problem is encountered with the enormous number of grid points required to capture all the physics of the flow. Therefore, it is more common to solve the Reynolds-Averaged Navier-Stokes (RANS) equations. These equations are the highest level of approximation. The equations solve for the mean flow field, which in turn requires a turbulence model for closure. In the world of CFD, there is no one turbulence model that is general enough for all flow conditions. Instead, each problem will need to be studied and have an appropriate model attached to it. In addition to a turbulence model, a relationship is required for the thermodynamic quantities. If a perfect gas is assumed, then the equation of state provides the necessary connection between thermodynamic variables such as density, pressure, and temperature. Some examples of where the RNS equations are applied are flow around a three dimensional obstacle [2], flow in an internal combustion engine [3], and unsteady oscillatory flow in an inlet [4].

The next level of approximation is to assume that the flow has a small amount of separation or backflow. The flow is predominantly in one direction with a high Reynolds number. When flows fit this description, the viscous and turbulent diffusion terms in the streamwise flow direction can be neglected. This will reduce the number of shear stress terms to be calculated, helping reduce CPU time. This approximation is known as thin shear layer (TSL). A common example of where the TSL approximation is applied is viscous flow along an airfoil [5].

For flows that are supersonic throughout the flow field (except for in the boundary layer), an approximation based upon the parabolic nature of the flow can be made. All the information needed to advance the solution at a point is assumed to be along the upstream characteristic lines. The pressure field keeps the equations from being fully parabolic, so the pressure gradient must be suppressed to maintain the parabolic

influence. Again the flow must be predominantly in one direction and supersonic. The equations for this level of approximation are called Parabolized Navier-Stokes (PNS) and are discretized such that the solution is marched downstream [6]. PNS is not valid for separated flows in the streamwise direction, but can predict crossflow separation. Care must also be taken when shocks interact with the boundary layer since the interaction may separate the flow in the viscous shear layer. In general, the accuracy of PNS will be reduced as the Reynolds number is lowered. An example of a PNS application would be supersonic flow about a cone or wedge [7].

The next level of approximation depends on how the density varies throughout the flowfield. If the density is compared to other flow parameters, such as pressure and velocity, then for most flows that occur naturally it could be stated that the variance in density is negligible compared to the other flow parameters. For example, the density of water is very close to being a constant over a large range of flow conditions. For air, the density can also be assumed constant without any significant loss of engineering accuracy for Mach numbers below 0.3. Because a large number of flows fall into this category, there exists an incompressible form of the Navier Stokes equations [8]. When density is assumed to be constant, the energy equation decouples from the mass and momentum equations, leaving one less equation to solve. For three dimensional problems, the base number of equations to solve becomes four instead of five. Not only is it more efficient to solve the incompressible Navier Stokes equations at low speeds, but the compressible form of the equations become numerically stiff at low Mach numbers, which slow convergence considerably. When the Mach number drops below 0.1, the condition number of the RANS equations become very large due to the decoupling of the energy equation from the momentum and continuity equations. Under this condition, the solution quality will degrade as well.

Getting away from the Navier-Stokes equations, the next level of approximation comes when the viscous layers are thin, allowing the separation of the viscous and inviscid parts of the flow. The viscous and turbulent shear stresses must be confined to small regions close to the wall, making the pressure field predominately influenced by the inviscid nature of the flow. When the viscous terms in the Navier-Stokes equations are dropped, the result is the Euler equations. The Euler equations still require the

solution of five equations for 3-D compressible flows, but the viscous terms no longer need to be computed. This results in a large increase in computational efficiency since the viscous terms no longer need to be computed and the computational grid does not need to be clustered near a wall to resolve the boundary layer. The Euler equations can be used to solve the inviscid flow field around a complete aircraft configuration.

The next approximation under inviscid flows is the potential equation [9]. The potential equation is limited to irrotational flows. With the inviscid irrotational flow assumption, entropy will be constant over the whole flow field if the initial conditions reflect a state of uniform entropy. For a constant entropy flow, known as isentropic flow, a set of isentropic relations exist which become the basis for simplifying the Euler equations. This level of approximation may not seem very valid for many flows, but the fact is that the potential flow model is equivalent to the Euler equations for continuous, irrotational flows. For subsonic flows around airfoils or wings, the solutions between the two should be almost identical. Only when shock waves become present are differences in the pressure and velocity field seen. As long as the normal Mach number to a shock is less than 1.3, solutions are still comparable to the Euler equations. The trade-off for the loss of accuracy associated with irrotational flow is that there is only one equation to solve with the potential equation. The single potential equation is an enormous reduction from the five equations associated with the Euler equations. Even though the potential equation is one of the lowest forms of approximation in fluid mechanics, it is still needed in engineering design and analysis. A large field of problems exist for which potential flow solutions give valid results for a reduction in computational cost over other approximation levels.

If a problem involves a slender or thin shape immersed in a flow that approaches the sonic condition, the transonic small-disturbance (TSD) equation can be applied. This equation is derived from the full potential equation, which does not assume any dominate flow direction. The TSD equation makes the assumption that the flow is predominately in one direction, with the flow in the normal direction being a small perturbation of the freestream flow. More information on TSD can be found in reference [10]. Information on solving the other forms of the governing equations can be found in references [11], [12], and [13].

The work presented here focuses on two of the approximation levels mentioned above. The first is the incompressible Navier-Stokes equations, and the second is the full potential equation. For the incompressible Navier-Stokes equations, the focus will be on improving the ability to accurately model turbulent flow separation. Turbulence modeling is one of the last remaining obstacles that hinders numerical accuracy for this level of approximation. An introduction to this topic is given in Chapter 2, followed by the work that was accomplished along with the results. For the full potential equation, the focus will be on developing an efficient inviscid flow solver for transonic flows. The time to attain a solution to the full potential equation will be faster than solving the Euler equations. This makes potential flow solvers very valuable in the design process since most of the physics will still be reflected in the solution, while the time to get the results is relatively fast. A more complete introduction to potential flows is given in Chapter 7. The remaining chapters cover the work related to potential flow.

# Chapter 2

## Introduction to Turbulent Separated Flows

### 2.1 Overview

Regions of separated flow are common in many engineering applications. Separation is the entire process in which the boundary layer flow breaks down and departs from the wall surface. This physical phenomena can have a large impact on the performance of any design. It is often possible to avoid separation by placing limitations on the operating conditions, but there are times when separated flow cannot be avoided and must therefore be dealt with. In these situations it is important to know and understand the effects of separation on a particular design.

Computational fluid dynamics has become a popular tool in performance analysis and design, but one of the major challenges of CFD is turbulent flow. There are many turbulence models available, but very few work well in regions of strong adverse pressure gradients or separation. The Johnson and King model (JKM) is one particular formulation that is specifically designed for flows with strong adverse pressure gradients. The model has also been shown to work well inside regions of separated flow [14] because it takes into account the nonequilibrium flow development that occurs when a boundary layer changes rapidly. A separated flow region

has turbulent diffusion and dissipation as the main terms in the backflow energy balance. This gives rise to nonequilibrium flow conditions. It is, therefore, important to model the nonequilibrium effects that occur in a separated flow region. The Johnson and King model is able to do this by solving a differential equation for the maximum Reynolds shear stress. This differential equation is derived from the turbulent kinetic energy equation. The solution of the equation allows the eddy viscosity to be scaled according to the nonequilibrium flow condition.

For the work involving solutions to the incompressible Navier-Stokes equations, focus is given to the Johnson and King Turbulence model. The work is aimed at performing two dimensional numerical solutions using the turbulence model so that comparisons with experiment can be made. In addition, changes to the model are made in order to improve the correlation between numerical and experimental data. The changes made to the model are based upon previous observations from the study of separated flows, as well as observations due to comparisons with experiment. Several of the changes to be presented here have also been used in a previous turbulence model for adverse pressure gradient flows. This particular turbulence model was developed by Hytopoulos [15].

Two low-speed turbulent flows will be computed using the Johnson and King turbulence model. The two test cases performed have not been done using this numerical scheme before, and will therefore provide additional cases in the research of turbulence modeling. Both test cases have regions of separated flow, and will be performed under both steady and unsteady flow conditions.

The first calculated test case to be discussed is a low-speed diffuser flow studied by Simpson *et al.* [16] and was used as a test case for the 1980-81 AFSOR-HTTM-Stanford Conference on Complex Turbulent Flows [17]. The diffuser decelerates the flow rapidly producing a large adverse pressure gradient. This causes the boundary layer to undergo massive separation. All computations are done in regions where the flow remains dominantly two-dimensional. As the flow travels further downstream, it becomes more three-dimensional and cannot be properly modeled in two-dimensions.

This same test case was used by Johnson and King [18] in developing and validating their turbulence model. Only the steady case was computed and the results

were found to agree well with experimental data. In their computations, Johnson and King used an inverse boundary-layer method to solve the separated flow region. The present work performs this same calculation except with the use of the full incompressible Navier-Stokes equations. In the present case, all the boundary layer properties are calculated, whereas in the inverse method some of the parameters are prescribed as part of the boundary layer edge conditions.

The second test case comes from an experiment performed at the Massachusetts Institute of Technology (MIT) and is known as the flapping foil experiment (FFX). FFX was performed to simulate blade loading under oscillatory conditions. The unsteady flow field acting upon the blade is modeled in the experiment as a vertical or transverse sinusoidal gust. This unsteady loading on a propeller blade is experienced on ships as the blade rotates through the wake created by the hull or appendage. This case has been performed by a number of researchers by solving the incompressible Navier-Stokes equations. Most of the results have used either the Baldwin Lomax turbulence model or the  $k - \epsilon$  model. A few other models have also been used, but the solution presented here will be the only known results using the Johnson and King model.

## 2.2 Turbulence Models

In the past century turbulence models have been invented and researched in great detail. The need for turbulence models came out of solving some form of the Navier-Stokes equations. The physical phenomena behind turbulent flow is described completely by the Navier-Stokes equations, but the problem is not in the equations themselves, but in solving them. Only a few exact solutions exist for the N-S equations, and these consist of relatively simple flows with limited applications. In order to solve the complete N-S in a more general sense, numerical methods have to be employed. Solving the full N-S equations or some reduced form of it by numerical means is the nature of CFD. As computer processing speeds become faster and memory capabilities grow, analysis and design using CFD continues to become more important. But there still remains a large gap between the resolution needed to resolve all the small

scale turbulent motion and the available computer resources to accomplish this. A few solutions have been done that have resolved all the scales of motion by solving the full N-S equations. This method of solving the Navier-Stokes equations without making any approximations is called the Direct Navier-Stokes (DNS) method. The few calculations that have been done using DNS have been valuable to the research community, providing insight into the physics of turbulence and turbulence modeling. The ability to use DNS for general problems of interest is not seen for the near future. An example of the required resources needed for a DNS calculation for a simple channel flow can be found in [19]. Because of this and the growing importance of CFD, turbulence modeling needs to be seriously studied to further improve numerical calculations.

The use of turbulence models became important after the assumption was first made that the N-S equations could be solved for the mean flow parameters. Turbulent motion is time dependent, so the mean flow represents the flow field after the turbulent fluctuations are time-averaged. This approach has become widely accepted and is now the most common approach to solving the equations. The concept was first thought up by Reynolds in the late nineteenth century, and is named after him (Reynolds averaged Navier-Stokes equations). When the N-S equations are time-averaged, the unknown variables become time-averaged quantities. This eliminates the need to resolve the time dependence of turbulent fluctuations, but also introduces new unknowns into the governing equations. The new unknowns are referred to as the turbulent Reynolds stresses. This trade-off allows solutions to turbulent flows to be attained using current computer resources. The unknowns create a closure problem, which is taken care of by using a turbulence model. There are three basic categories of turbulence modeling [12], which will be discussed next. The discussion that follows is only an overview of turbulence models. A more complete discussion on this issue has been done by Wilcox [20].

The first category for turbulence models comes from the Boussinesq assumption. Boussinesq suggested in 1877 that the apparent turbulent shearing stresses are related to the rate of mean strain through an apparent scalar turbulent viscosity. This apparent scalar viscosity is known as the eddy viscosity, and forms the basis for the

majority of turbulence models today. Turbulence models that fall into this category can be further classified according to the number of supplementary differential equations that must be solved. These supplementary equations are solved for modeling parameters, the number of which varies according to the model. Due to these extra equations, the different turbulent models are described as  $n$ -equation models where  $n$  refers to the number of differential equations that must be solved for a given model. For the case of  $n = 0$ , the model is referred to as a zero-equation or algebraic model. Prandtl suggested and developed this first type of turbulence model. Prandtl related the eddy viscosity to the density, a mixing length, and a characteristic velocity. This breakdown of the eddy viscosity still remains the basic components of algebraic models today. The appearance of one-equation models came next as a result of trying to develop a more realistic description of the flow. A model for the eddy viscosity was postulated which depended upon the kinetic energy of the turbulence. An equation for the kinetic energy was developed, which when solved, allowed the eddy viscosity to be more sensitive to the surrounding flow by taking into account the flow history. Before this, the more simple algebraic models were influenced only by the mean flow properties at a given location. By adding additional equations, for example an equation for the turbulent mixing length scale, two-equation models came into existence. Along with the added number of differential equations to be solved came increased numerical complexity. One and two-equation models are not as simple to program as algebraic models and require more computational time to solve.

The second category of turbulence modeling is referred to as Reynolds stress or stress-equation models. These models start with transport equations derived for the Reynolds stresses. The work for this class of turbulence modeling was first done by Rotta [21]. As a result of the new transport equations, more unknowns were introduced which must then be modeled based upon experimental observations. With the additional unknowns to predict, these types of turbulence models became complex, and have not demonstrated great advantages over the eddy viscosity models. A review of this class of turbulence modeling is given by Launder [22].

The last category is known as large eddy simulation (LES). This technique takes the approach that large scale turbulent motion can be solved for directly by refining

the grid. The small scale or subgrid scale turbulence is taken care of by employing a turbulence model. The theory behind this approach takes the viewpoint that the large turbulent motion is uniquely characterized by the boundary conditions, and should not be modeled, but rather computed as part of the flow solution. The smaller, subgrid scales of motions on the other hand are more isotropic and universal, which can therefore be resolved using a general model. This approach to turbulent flows is not as costly as DNS, but is still an expensive modeling technique to be used as a general engineering method. One of the earliest examples of LES was done by Deardorff [23].

The problem with turbulence models may be understood better by realizing that no one turbulence model has been found to be superior. Depending upon the problem at hand, different models may be more suitable to use than others. The increased amount of computational time and memory for the more complex models do not always produce better accuracy over the simpler models. Until more is learned about the process of turbulent flow aiding in the development of a universal turbulence model, scientists and engineers will continue to pursue the different avenues of turbulence modeling described above.

## 2.3 Johnson and King Turbulence Model

In the following chapters, the focus of the numerical solutions and discussions on turbulent flows will be directed toward the Johnson and King turbulence model. This model falls into the first category where an eddy viscosity is predicted and used in the solution of the Reynolds averaged Navier-Stokes equations. A unique feature of this model is that it lies between the simple algebraic models and the one-equation models, resulting in what is called a half-equation model. Half-equation models have the property that one of the modeling parameters are permitted to vary with the primary flow direction. The JKM has an auxiliary equation which is classified as a partial differential equation for time dependent flows or an ordinary differential equation for steady state flows. The differential equation is solved along the streamwise flow direction, making the solution process sensitive to the flow direction and grid

Table 1: Some half-equation turbulence models (taken from Reference [12]).

| Parameter determined by equation | Basis for Transport equation used | References                   |
|----------------------------------|-----------------------------------|------------------------------|
| $l_{in}$                         | Turbulence kinetic energy         | McDonald and Kreskovsky [24] |
| $l_{in}$                         | Turbulence kinetic energy         | Chan [25]                    |
| $l_{in}$                         | Turbulence kinetic energy         | Adams and Hodge [26]         |
| $l_{in}$                         | Empirical ODE for $l_{in}$        | Malik and Pletcher [27]      |
| $\mu_{to}$                       | Empirical ODE for $\mu_{to}$      | Reyhner [28]                 |
| $\mu_{to}$                       | Empirical ODE for $\mu_{to}$      | Shang and Hankey [29]        |
| $\tau_{max}$                     | Turbulence kinetic energy         | Hytopoulos [15]              |

orientation. The solution of the JKM auxiliary equation is used to scale the eddy viscosity in the outer turbulence layer. The solution process of the JKM is usually more simplified than the algorithm used in solving one-equation turbulence models. This allows the JKM to have the coding simplicity of algebraic models while taking into effect the upstream history of the flow which is found in one and two-equation models.

There have been a handful of half-equation models develop over the years. A brief listing of some of these models is given in Table 1. In this table,  $l_{in}$  stands for the inner region mixing length,  $\mu_{to}$  represents the outer layer eddy viscosity, and  $\tau_{max}$  is the maximum Reynolds shear stress. The Johnson and King model is unlike any of these models except for the last one where the differential equation is used to find the maximum Reynolds shear stress. The transport equation used in the JKM is based upon the turbulence kinetic energy equation.

In the remainder of this section, the history of the Johnson and King turbulence model will be discussed. The model was first developed by Johnson and King [30] [18] in 1985. In their original papers, Johnson and King ran several two-dimensional problems using the inverse boundary layer method to test the performance of the new model. The test cases ranged from Simpson's *et al.* [16] low-speed diffuser problem to

transonic flow over a supercritical airfoil. In a follow-up paper by Johnson [31], computations were made using JKM based upon the Reynolds averaged Navier-Stokes equations. A single case was computed which consisted of a transonic flow over an axisymmetric circular-arc bump. Johnson concluded that the JKM provided accurate predictions for the inviscid-viscous interactions without compromising numerical efficiency. The model was soon extended to 3-D flows by Abid [32]. Abid performed calculations on a low-speed infinite swept wing which included the wing-body junction. A transonic wing soon followed where the ONERA M6 wing case was used [33]. In 1990, improvements to the inner region model were made by Johnson and Coakley [34]. The wall shear velocity, which was originally used in the model, was replaced by a blending of the maximum Reynolds shear stress and the wall shear velocity. This change made significant improvements in the near wall modeling. A comparison of several eddy viscosity models with the Johnson and King model was done by Wu *et al.* [35] and Menter [14]. The models that were tested against the JKM were the Baldwin-Lomax, Baldwin-Barth, Wilcox  $k - \omega$ , and the  $k - \epsilon$  models. In both references, the JKM performed equally as well or better than the other turbulence models. Wu *et al.* [35] and Dindar and Kaynak [36] are some examples of unsteady numerical computations using JKM. In the paper by Dindar and Kaynak, a time dependent term was added to the Johnson and King differential equation, which would not normally appear for steady flows. Additional examples of the JKM can be found in references [37] and [38].

## 2.4 Physics of Separated Flows

The numerical computations involving turbulence modeling focus on flows with adverse pressure gradients and boundary layer separation. It may therefore be insightful to discuss some of the physical phenomena that are associated with separated flows. Simpson has periodically reviewed the topic of separated flow over the years [39], [40], [41], and [42] providing a history of the understanding of turbulent flow separation.

Separation is the term used to describe the entire process in which a flow detaches

from a solid surface, causing a breakdown of the boundary layer. When this process occurs, the boundary layer undergoes a sudden thickening, causing an increased interaction between the viscous-inviscid layers. When separation occurs on smooth surfaces such as airfoils and flat plates, it is usually the result of an adverse pressure gradient. As the pressure increases along the primary flow direction, the flow velocity responds by decelerating. If this pressure differential continues, the flow velocity will eventually come to zero and a reversal of the flow will occur.

Whether the flow field is considered steady or unsteady, the location at which flow reversal occurs will vary along the wall. In fact, at a given location in the vicinity of separation, flow reversal may occur only a fraction of the total time. The flow at these locations is said to be intermittent, since the flow is not fully separated. A set of quantitative definitions have been proposed by Simpson [41] to aid in the description of the flow near separation. The definitions are based on the fraction of time that the flow moves downstream. The flow locations have the following labels: incipient detachment (ID), intermittent transitory detachment (ITD), transitory detachment (TD), and detachment (D). Incipient detachment signifies the location where flow reversal occurs 1% of the time. Following this description, intermittent transitory detachment and transitory detachment occur for 20% and 50% flow reversal respectively. The detachment location is defined as the location where the averaged wall shearing stress is zero, which in most cases corresponds to transitory detachment [41]. The location and distance between each of these positions will be dependent on the flow parameters and geometry.

Most all flow fields with separation present will have a region of flow where the “law-of-the-wall” holds. Simpson [43] has observed that the mean flow upstream of ID obeys the law-of-the-wall and law-of-the-wake if the following condition is met: the maximum shearing stress is less than 1.5 times the wall shear stress. When this condition is not met, other mean-velocity correlations hold for the region upstream of ITD (ie. Perry and Schofield [44]). A major difference in the turbulence structure between the two correlations is that the maximum fluctuations are more toward the middle of the boundary layer rather than being more close to the wall as in a zero pressure gradient case. Therefore, a flow that starts off with a favorable pressure

gradient will most always obey the law of the wall. At some point the pressure gradient will become positive, creating an adverse gradient causing a deceleration of the flow. If this condition continues, the appearance of flow reversal will occur, signifying ID. Moving downstream a little more, ITD can be observed indicating more flow reversal. The mean flow at this state still follows certain “laws” which have been observed to hold under known conditions. Another observation by Simpson *et al.* [45] is that the average spanwise spacing of eddies near the wall is almost constant upstream of ITD. After detachment, the spacing increases by about an order of magnitude.

The backflow that is first observed at ID is a result of turbulent elements moving upstream for some distance and then reversing and being convected downstream. The rate at which this occurs will increase as the detachment location is approached. These elements that make their way upstream and then downstream in the region near the wall have their origin from larger eddies in the outer turbulent region. These larger eddies, which originate in the outer boundary layer, bring momentum toward the wall causing smaller eddies to form and travel in all directions once the wall is encountered. After detachment (D), the mean flow is represented by flow traveling upstream, known as backflow, which will eventually encounter the downstream flow causing the backflow to turn in the direction away from the wall and rejoining the flow traveling downstream. What is physically happening in this situation is that coherent structures in the outer boundary layer are projected into the backflow region, causing intermittent flow along the wall. In a similar manner, flow inside the backflow region transverses out into the downstream flow to conserve mass in the detached flow region. As a result of this process, the velocity fluctuations in the backflow region can be just as large or greater than the mean backflow velocity [43].

The flowfield in the backflow region can be divided up into three layers. The first layer is near the wall and has small effects from the Reynolds shearing stress, but is instead dominated by the turbulent flow unsteadiness. An outer backflow region exists that behaves as part of the large scale outer region flow, forming the second layer. Between these two layers an overlap region exists with a semilogarithmic mean velocity profile. For a large portion of the backflow profile, Simpson [46] made a correlation for the backflow velocity. The velocity is scaled on the maximum negative

backflow velocity and the distance from the wall to the location where maximum backflow occurs. Simpson's equation does not hold for every backflow condition, but has been shown to hold for a number of data sets [47].

The turbulent energy of any shear flow consists of production, convection, diffusion, and dissipation. For a zero pressure gradient flow, dissipation and production are the main contributors to the turbulent energy balance except near the outer edge. In the case of backflow, negligible turbulent energy production occurs. Instead, diffusion and dissipation are the main terms in the energy balance [16]. Near the wall, regions of low kinetic energy will exist, while away from the wall shearing stresses are large. Normal stress contributions, which are negligible in zero pressure gradient flows, become significant in the momentum and turbulence energy equations. This production of both the normal and shear stresses supply turbulence energy in the outer region through turbulent diffusion.

# Chapter 3

## Incompressible Flow Solver

### 3.1 Introduction

In this chapter, the algorithm for solving the incompressible Navier-Stokes equations will be presented. The equations are non-linear and are solved by numerical techniques applicable to general engineering applications. The algorithm used here to solve the non-linear equations is the artificial compressibility method. This method will be presented for both steady and time accurate problems. In the sections that follow, the equations governing incompressible flow will be presented. The equations will then be non-dimensionalized and have the artificial compressibility method applied. Issues concerning the discretization and boundary conditions will then be covered. The chapter concludes with a few examples of the flow solver applied to inviscid and viscous fluid flow problems. The discussion on turbulence modeling will be presented in the following chapter.

It should be mentioned here that there are two main approaches to solving the incompressible Navier Stokes equations. A solution procedure will take either the vorticity-stream function approach or the primitive variable method approach. A brief description of these two approaches will be given below.

The vorticity-stream function formulation is based on the definition of vorticity and stream function, which depends only on the components of velocity. When the definition of these two parameters are substituted into the governing equations, a

transport equation for vorticity appears in place of the momentum equations, and the continuity equation becomes only a function of the stream function. The pressure is removed from the unknowns, and the dependent variables become the vorticity and the stream function. Once the stream function is solved for, the velocity field can easily be found and the pressure can then be computed. This approach reduces the overall number of equations to solve, which in turn reduces the computational time to perform the calculation. The method can be applied to both steady and unsteady flows, but becomes more complicated for three dimensional flows. For the case of 3-D flows, the stream function is no longer available and one must use the potential formulation, adding more complexity to the system. A more complete description of the method can be found in [48].

Algorithms that fall under the primitive variable category solve directly for the pressure and velocity components, known as the primitive variables. Two main classes of algorithms have been developed under this category. The first approach, which has been the most widely used in the past, decouples the set of non-linear equations. These are sometimes known as pressure based schemes in which a pressure poisson equation is formed from the momentum and continuity equations. The velocity field is solved for using the momentum equation, which is then followed by solving for the pressure from the poisson equation. PISO [49] and SIMPLE [50] are some examples of algorithms that employ this approach. The second approach under this category solves for the unknowns as a coupled system. The main problem with this concept is coupling the pressure to the continuity equation. For incompressible flows, the continuity equation no longer has a time derivative, creating problems for time integration procedures. To overcome this obstacle, a pseudo-time derivative for the pressure is added to the continuity equation, allowing the velocity and pressure to be directly coupled. With the addition of the pseudo-time derivative, the continuity equation changes from elliptic to hyperbolic type in the space-time domain. Algorithms that have been developed for solving the compressible flow equations can then be directly applied to the new set of equations, taking advantage of all the technology in compressible flow research. This approach is given the name artificial compressibility and

was introduced by Chorin [51]. It was originally developed for steady flow computations, but can be used for unsteady flows as well by using a dual-time stepping procedure. A few examples of its time-accurate capabilities are given in [52] and [53]. More details on this method will now be given in the following sections.

## 3.2 Governing Equations

The governing equations used in computing the viscous flow calculations are the incompressible Navier-Stokes equations. These equations are based on the conservation of mass and momentum. The derivation of the governing equations can be found in most textbooks on fluid dynamics, and so the equations will only be presented here. The equations, in integral form, are given as,

$$\frac{\partial}{\partial t} \iiint_{\mathcal{V}} q \, d\mathcal{V} + \iint_S \vec{F} \cdot \hat{n} \, dS = 0 \quad (3.1)$$

where  $q$  represents the conserved variables and  $F$  is the vector containing both the inviscid and viscous fluxes.

Since turbulent flow is naturally three dimensional and time dependent, only the 3-D form of the incompressible Navier-Stokes equations are valid for turbulent separated flows. In order to solve the governing equations for a two dimensional flow, the Reynolds Averaged Navier-Stokes equations must be solved. This form of the equations solve for the mean flow parameters by time averaging the turbulent fluctuations. Therefore, if the mean flow that is being calculated is two dimensional, the 2-D form of the equations can be used.

Reynolds averaging is applied to the primitive variables as the first step in solving for the time averaged variables. The primitive variables are replaced by their time averages and the fluctuation about the time average. For the two-dimensional incompressible flow equations, the flow variables can be written as,

$$p = \bar{p} + p' \quad u = \bar{u} + u' \quad v = \bar{v} + v' \quad (3.2)$$

The time average values are relative to the characteristic fluctuation period of the turbulence, and not to any unsteadiness that may be present in the mean flowfield.

Consequently, Reynolds averaging of the governing equations will remain valid for unsteady turbulent flows.

The Reynolds Averaged Incompressible Navier-Stokes equations for a two dimensional flow will be presented next, but before that is done a few assumptions must be made about the flow. The system of equations are solved with the assumption that the fluid is Newtonian. Three assumptions are associated with a Newtonian fluid, which are given below:

1. The stress tensor is linearly proportional to the rate of strain.
2. The fluid is isotropic (properties are independent of coordinate direction).
3. When the strain rates are zero (the fluid is at rest), the stresses correspond to their hydrostatic conditions.

These assumptions allow the viscous stresses to be written in the following form for two dimensional, incompressible fluid,

$$\tau_{i,j} = \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (3.3)$$

where the values of  $i, j$  correspond to the  $x, y$  coordinate directions respectively.

The Reynolds averaged equations for mass and momentum are,

$$\frac{\partial \bar{u}}{\partial x} + \frac{\partial \bar{v}}{\partial y} = 0 \quad (3.4)$$

$$\frac{\partial \bar{u}}{\partial t} + \frac{\partial \bar{u}^2}{\partial x} + \frac{\partial \bar{u}\bar{v}}{\partial y} = \frac{-1}{\rho} \frac{\partial \bar{p}}{\partial x} + \frac{\mu}{\rho} \frac{\partial^2 \bar{u}}{\partial x^2} + \frac{\mu}{\rho} \frac{\partial^2 \bar{u}}{\partial y^2} - \frac{\partial \overline{u'u'}}{\partial x} - \frac{\partial \overline{u'v'}}{\partial y} \quad (3.5)$$

$$\frac{\partial \bar{v}}{\partial t} + \frac{\partial \bar{u}\bar{v}}{\partial x} + \frac{\partial \bar{v}^2}{\partial y} = \frac{-1}{\rho} \frac{\partial \bar{p}}{\partial y} + \frac{\mu}{\rho} \frac{\partial^2 \bar{v}}{\partial x^2} + \frac{\mu}{\rho} \frac{\partial^2 \bar{v}}{\partial y^2} - \frac{\partial \overline{u'v'}}{\partial x} - \frac{\partial \overline{v'v'}}{\partial y} \quad (3.6)$$

The last two terms of both the  $x$  and  $y$  momentum equations can be thought of as apparent stress gradients due to the transport of momentum by turbulent fluctuations. These apparent stresses are known as Reynolds stresses and take on the following form,  $-\rho \overline{u'_i u'_j}$ . Both the density and the molecular viscosity are constant for incompressible flows and are therefore taken out of the derivative.

The Boussinesq assumption is now used to introduce the eddy viscosity given as  $\mu_t$ . Recall that the Boussinesq assumption states that the apparent turbulent shearing

stresses are related to the rate of mean strain through an apparent scalar turbulent viscosity. The turbulent viscous stresses  $(\tau_{i,j})_t$  are therefore similar to the viscous stresses given in (3.3). The expression for  $(\tau_{i,j})_t$  is simply,

$$(\tau_{i,j})_t = \mu_t \left( \frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \quad (3.7)$$

The only difference between this relation and the one given for the viscous stresses is that  $\mu$  has been replaced by the eddy viscosity,  $\mu_t$ . The Reynolds stresses introduced in equations (3.5) and (3.6) are related to the turbulent viscous stresses by

$$(\tau_{i,j})_t = -\overline{\rho u'_i u'_j}$$

An example of how  $\mu_t$  is linked to the Reynolds stress  $(\overline{u'v'})$  is given below.

$$(\tau_{x,y})_t = -\overline{\rho u'v'} = \mu_t \left( \frac{\partial \bar{u}}{\partial y} + \frac{\partial \bar{v}}{\partial x} \right) \quad (3.8)$$

The remaining Reynolds stresses are expressed in a similar manner and substituted into the Reynolds averaged momentum equations. Just as the viscous stresses given by (3.3) were used to give the viscous terms in equations (3.5) and (3.6), similar terms result from the Reynolds stresses giving one new unknown, the turbulent eddy viscosity. The terms that result from the Reynolds stresses are combined with the viscous terms since the only difference is the parameter  $\mu_t$ .

The final form of the governing equations are once again given. The bars over the time averaged variables have been dropped and the turbulent eddy viscosity is introduced.

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (3.9)$$

$$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} = \frac{-1}{\rho} \frac{\partial p}{\partial x} + \frac{1}{\rho} (\mu + \mu_t) \frac{\partial^2 u}{\partial x^2} + \frac{1}{\rho} (\mu + \mu_t) \frac{\partial^2 u}{\partial y^2} \quad (3.10)$$

$$\frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} = \frac{-1}{\rho} \frac{\partial p}{\partial y} + \frac{1}{\rho} (\mu + \mu_t) \frac{\partial^2 v}{\partial x^2} + \frac{1}{\rho} (\mu + \mu_t) \frac{\partial^2 v}{\partial y^2} \quad (3.11)$$

The new unknown,  $\mu_t$ , is solved for using turbulence models such as the Johnson and King model.

### 3.3 Non-dimensionalization

Before proceeding to introduce the artificial compressibility method, the governing equations that were presented in the last section will be non-dimensionalized. This is commonly done to help scale all the flow parameters in order to reduce round-off errors during the computations. Reference conditions are specified and used to non-dimensionalize the equations. The nondimensional parameters will be denoted by the superscript  $*$  and will be used to replace the actual variables in the governing equations. Definitions for the nondimensional parameters are given below:

$$u^* = \frac{u}{V_{ref}} \quad v^* = \frac{v}{V_{ref}} \quad \rho^* = \frac{\rho}{\rho_{ref}} \quad p^* = \frac{p}{\rho_{ref} V_{ref}}$$

$$x^* = \frac{x}{L} \quad y^* = \frac{y}{L} \quad t^* = \frac{t}{L/V_{ref}} \quad \mu^* = \frac{\mu}{\mu_{ref}}$$

The above definitions are substituted into (3.9), (3.10), and (3.11). The reference variables can be grouped together in the momentum equations, while in the continuity equation they are identically zero. The governing equations are rewritten as:

$$\frac{\partial u^*}{\partial x^*} + \frac{\partial v^*}{\partial y^*} = 0 \quad (3.12)$$

$$\frac{\partial u^*}{\partial t^*} + \frac{\partial u^{*2}}{\partial x^*} + \frac{\partial u^* v^*}{\partial y^*} = \frac{-1}{\rho^*} \frac{\partial p^*}{\partial x^*} + \frac{1}{Re} \left[ (\mu^* + \mu_t^*) \left( \frac{\partial^2 u^*}{\partial x^{*2}} + \frac{\partial^2 u^*}{\partial y^{*2}} \right) \right] \frac{1}{\rho^*} \quad (3.13)$$

$$\frac{\partial v^*}{\partial t^*} + \frac{\partial u^* v^*}{\partial x^*} + \frac{\partial v^{*2}}{\partial y^*} = \frac{-1}{\rho^*} \frac{\partial p^*}{\partial y^*} + \frac{1}{Re} \left[ (\mu^* + \mu_t^*) \left( \frac{\partial^2 v^*}{\partial x^{*2}} + \frac{\partial^2 v^*}{\partial y^{*2}} \right) \right] \frac{1}{\rho^*} \quad (3.14)$$

where the Reynolds number is defined from reference conditions

$$Re = \frac{\rho_{ref} V_{ref} L}{\mu_{ref}}$$

The equations appear almost identical to the dimensional equations except for the introduction of the Reynolds number. Since both the density and the molecular viscosity are constants,  $\rho_{ref}$  and  $\mu_{ref}$  are set equal to their freestream values. This allows the density term to be taken out of the governing equations since it takes on the value of unity. The only reference values needed to be specified are  $V_{ref}$  and  $L$ . For any further reference to the governing equations, the superscript  $*$  will be dropped for clarity.

### 3.4 Artificial Compressibility Method

The solution algorithm used to solve the governing equations is based on the scheme by Rogers and Kwak [54, 55]. The algorithm employs the method of artificial compressibility in which an artificial compressibility parameter  $\beta$  is introduced into the continuity equation along with a time derivative term for the pressure.

$$\frac{1}{\beta} \frac{\partial p}{\partial \tau} + \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (3.15)$$

The addition of the pseudo-time derivative term directly couples the pressure and velocity. The set of governing equations, (3.5), (3.6), and (3.15), become hyperbolic in space-time, which is the same form as the compressible equations. This similarity allows the methods used in solving the compressible equations to be directly applied to the incompressible equations presented here. The equations will be solved by marching the solution in physical time. For steady state solutions, the pseudo-time derivative term will vanish as the solution converges, satisfying the conservation of mass. For time dependent flows, sub-iterations are performed which will satisfy continuity for each physical step in time. The time integration scheme will be discussed in more detail in a later section.

### 3.5 Flux Scheme

To solve the governing equations in a finite volume formulation, the integral form of the equations are applied to individual control volumes or cells. A structured mesh will be used here where the control volumes take on the form of quadrilaterals. Equation (3.1) is applied to each cell, which is written as,

$$\mathcal{V} \frac{\partial q}{\partial t} + \sum_{j=1}^{\#faces} (\vec{F} \cdot \hat{n}_j) \Delta A_j = 0 \quad (3.16)$$

where the number of faces will be equal to four. Since the formulation is applied to 2-D meshes, the cell volume  $\mathcal{V}$  can be thought of as the area of the quadrilateral with a unit depth in the third dimension. In a similar way, the face area  $A$  is the length of the quadrilateral side with unit depth in the third dimension. The unit normal

vector appearing in the equation is defined as  $\hat{n} = n_x \hat{i} + n_y \hat{j}$  and is defined as positive when pointing outward from the control volume surface. Also in this equation,  $q$  is the array of dependent state variables, which are solved for as cell averaged values, and  $\vec{F}$  contains the inviscid and viscous fluxes. The variables  $q$  and  $\vec{F}$  are defined as,

$$q = [p, u, v]^T \quad (3.17)$$

$$\vec{F} = (F - F_v)\hat{i} + (G - G_v)\hat{j} \quad (3.18)$$

where  $F$ ,  $G$  and  $F_v$ ,  $G_v$  are inviscid and viscous fluxes respectively. From the notation above, the fluxes labeled  $F$  are made up of those terms that have contributions to the flux only in the  $x$  direction, while  $G$  represents those terms for the  $y$  direction. Since the flux for a given face will usually have contributions from both  $F$  and  $G$ , it is more convenient to define the flux vector for an arbitrary face. This is accomplished by defining a new vector,  $\hat{F}$ , which will be a combination of the fluxes  $F$  and  $G$  based upon the unit normal vector. The normal vector at each face will determine the proper flux contribution from the  $x$  and  $y$  directions. For an arbitrary face, the term  $(\vec{F} \cdot \hat{n})$  in equation (3.16) can be expressed as  $(\hat{F} - \hat{F}_v)$  where,

$$\hat{F} = F n_x + G n_y = \begin{bmatrix} \beta \theta \\ u \theta + p n_x \\ v \theta + p n_y \end{bmatrix} \quad (3.19)$$

$$\hat{F}_v = F_v n_x + G_v n_y = (\mu + \mu_t) \begin{bmatrix} 0 \\ u_x n_x + u_y n_y \\ v_x n_x + v_y n_y \end{bmatrix} \quad (3.20)$$

The subscripts on  $u$  and  $v$  represent derivatives.  $\theta$  is the contravariant velocity defined as  $\theta = u n_x + v n_y$ . In relation to a cell face,  $\theta$  is the normal component of velocity on that face. In a similar fashion the parameter  $\phi$  can be defined as  $\phi = n_x v - n_y u$ , which is the tangential component of velocity at a cell face.

The summation term in (3.16) requires the evaluation of the fluxes on each face of the cell. For the inviscid fluxes, each cell face uses a flux-difference-splitting scheme. The flux on an arbitrary face is defined as [56],

$$\hat{F}_{j+1/2} = \frac{1}{2} \left[ \hat{F}(q^R) + \hat{F}(q^L) - |\tilde{A}|_{j+1/2} (q^R - q^L) \right] \quad (3.21)$$

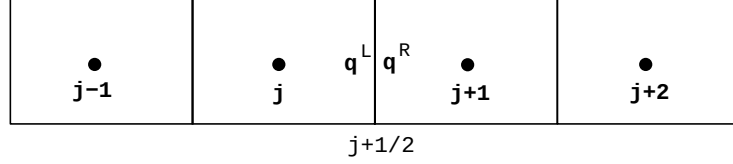


Figure 3.1: Interpolation of right and left states at a cell face

The term  $\tilde{A}$  is the Roe averaged Jacobian matrix defined by  $\tilde{A} = \tilde{T}|\tilde{\Lambda}|\tilde{T}^{-1}$ . The tilde indicates that the values of  $q$  are evaluated as averages. The superscripts  $R$  and  $L$  correspond to the right and left states to indicate which side the interpolation of  $q$  is performed for the cell face. For example, using Figure 3.1, a first order interpolation to  $q_{j+1/2}$  would have  $q^R = q_{j+1}$  and  $q^L = q_j$ . If the left and right states are set equal, the full flux is recovered. For higher order flux differencing, the MUSCL (Monotone Upstream-centered Schemes for Conservation Laws) approach [57] is used for the evaluation of  $q$ . Shown below are formulas for the left and right states of  $q$  based upon MUSCL.

$$\begin{aligned} q_{j+\frac{1}{2}}^L &= q_j + \frac{\varphi}{4}[(1 - \kappa) \nabla_j + (1 + \kappa) \Delta_j]q \\ q_{j+\frac{1}{2}}^R &= q_{j+1} - \frac{\varphi}{4}[(1 - \kappa) \Delta_{j+1} + (1 + \kappa) \nabla_{j+1}]q \end{aligned} \quad (3.22)$$

The operators  $\Delta$  and  $\nabla$  represent forward and backward difference operators respectively. The value of  $\kappa$  controls the order of the interpolation and can be shown to yield:

$$\kappa = -1 \quad \text{2nd order; one-sided} \quad (3.23)$$

$$\kappa = 1 \quad \text{2nd order; central scheme} \quad (3.24)$$

$$\kappa = 1/3 \quad \text{3rd order; upwind} \quad (3.25)$$

First order interpolation is easily recovered by setting  $\varphi$  to zero, otherwise it is set to unity for the higher order schemes.

The Roe averaged Jacobian matrix, as defined previously, is derived by first forming the Jacobian matrix for the flux vector, and then diagonalizing that matrix. The results presented here follow the work of Anderson *et al.* [58]. The Jacobian for  $\hat{F}$

becomes,

$$[\hat{A}] = \frac{\partial \hat{F}}{\partial q} = \begin{bmatrix} 0 & \beta n_x & \beta n_y \\ n_x & \theta + n_x u & n_y u \\ n_y & n_x v & \theta + n_y v \end{bmatrix} \quad (3.26)$$

The eigenvalues of  $\hat{A}$  are,

$$\lambda_1 = \theta \quad \lambda_2 = \theta + c \quad \lambda_3 = \theta - c \quad (3.27)$$

where  $c$  is defined as  $[\theta^2 + \beta]^{\frac{1}{2}}$ . Once the eigenvalues are known, the matrix  $\Lambda$  is the diagonal matrix containing the eigenvalues.

$$[\Lambda] = \begin{bmatrix} |\theta| & 0 & 0 \\ 0 & |\theta + c| & 0 \\ 0 & 0 & |\theta - c| \end{bmatrix} \quad (3.28)$$

The eigenvectors correspond to the columns of  $T$ , which are given below along with the inverse matrix of  $T$ .

$$[T] = \begin{bmatrix} 0 & -c(\theta - c) & c(\theta + c) \\ -n_y & n_x c - n_y \phi & -(n_x c + n_y \phi) \\ n_x & n_y c + n_x \phi & -(n_y c - n_x \phi) \end{bmatrix} \quad (3.29)$$

$$[T]^{-1} = \begin{bmatrix} -\frac{\phi}{c^2} & -\frac{\phi \theta n_x + n_y c^2}{c^2} & -\frac{\phi \theta n_y - n_x c^2}{c^2} \\ \frac{1}{2c^2} & \frac{(\theta + c)n_x}{2c^2} & \frac{(\theta + c)n_y}{2c^2} \\ \frac{1}{2c^2} & \frac{(\theta - c)n_x}{2c^2} & \frac{(\theta - c)n_y}{2c^2} \end{bmatrix} \quad (3.30)$$

## 3.6 Time Integration

The system of equations given by (3.16) can be written in the following form

$$V \frac{\partial q}{\partial t} + R = 0 \quad (3.31)$$

where  $R$  is the residual. The method of computing the residual was discussed in the previous section, and so now attention must be given to the time derivative. In applying the time integration scheme, one must decide on the order of accuracy of the

time derivative, and on whether the system will be explicit or implicit. The approach taken here will be to apply a general time integration formula which will be a function of several parameters that control the accuracy and type of system to solve, either explicit or implicit. The general time integration formula with its truncation error is [59],

$$\left[\frac{\partial q}{\partial t}\right]^n = \frac{1}{\Delta t} \left[ \frac{(1 + \zeta)\Delta^n - \zeta \nabla^n}{1 + \vartheta \Delta^n} \right] q^n + O[(\vartheta - \zeta - \frac{1}{2})\Delta t] + O(\Delta^2 t) \quad (3.32)$$

The superscript  $n$  denotes the physical time level corresponding to  $\Delta t$ . If  $\vartheta$  is set to zero, then the time integration will be explicit while  $\vartheta$  not equal to zero gives an implicit integration scheme. There will be two time integration schemes used for all the computational results presented. For steady state flows, Euler implicit will be used which requires that  $\vartheta = 1$  and  $\zeta = 0$ . The time accuracy is first order accurate which gives better convergence rates over the higher order schemes. Recall that time accuracy is not needed for steady state flows since the time derivative term vanishes as the solution is approached. For unsteady flows, a three-point backward time integration is used which is indicated by  $\vartheta = 1$  and  $\zeta = \frac{1}{2}$ . This yields a second order time accurate solution.

Before (3.32) is applied to the time derivative of equation (3.31), the artificial pressure term in the continuity equation must be dealt with. Whether a computation is done for a steady or unsteady flow, the pseudo-time derivative must always be zero to satisfy mass conservation. For steady state flows the time derivative will vanish as the solution is converged, satisfying continuity. For an unsteady flow, the pseudo-pressure time derivative must go to zero for each physical step in time,  $\Delta t$ . A pseudo-time level is therefore introduced and is indicated by the superscript  $m$ . The corresponding pseudo-time level step is represented by  $\Delta \tau$ . The time integration formula is now applied along with the pseudo-time level concept. The following system of equations is a result of this process.

$$\begin{aligned} \left[ V I_{tr} + \vartheta \left( \frac{\partial R}{\partial q} \right)^{n+1,m} \right] (q^{n+1,m+1} - q^{n+1,m}) = \\ -R_{jk}^{n+1,m} - \frac{V I_m}{\Delta t} \left[ (1 + \zeta)\Delta^n - \zeta \nabla \right] q \end{aligned} \quad (3.33)$$

where

$$I_{tr} = \text{diag}\left[\frac{1}{\Delta\tau}, \frac{(1+\zeta)}{\Delta t}, \frac{(1+\zeta)}{\Delta t}\right]$$

and

$$I_m = \text{diag}[0, 1, 1]$$

To perform time accurate calculations,  $\Delta\tau$  is set to a high value (i.e.  $\Delta\tau = 10^{12}$ ) and iterated upon until the divergence of the velocity reaches some convergence criterion, indicating that continuity has been satisfied for that time step.

The above linearization requires the derivation of the flux Jacobians, given by  $\partial R/\partial q$ . If an exact linearization is performed for the given flux scheme, the solution to the linear system will yield quadratic convergence. This would be the ideal case, but is not always easy to attain. For a first order inviscid flux, the left hand side of (3.33) will consist of the diagonal plus four off-diagonal terms. In this case, the system is banded and methods are available that can solve systems of this form, giving quadratic convergence for exact flux Jacobians. If a third order flux calculation is performed, then the number of off-diagonals will increase, requiring more computational work to solve the system. Due to the increased difficulties in forming the Jacobians with higher order calculations, it is common to use linearizations that are approximate. The linear system used here after forming the approximate Jacobians will be banded, consisting of two diagonals surrounding the main diagonal and two other diagonals making up the outer bands. Two approaches to computing the inviscid flux linearizations will be presented next, followed by the linearization of the viscous terms.

The first linearization scheme starts with the flux formula given by equation (3.21). With the flux evaluated using a first order approximation to  $q$  at the face, the linearization is performed. To illustrate how the Jacobian is formed for a given face, a first order flux is given here for the  $(j + 1/2)$  cell face.

$$\hat{F}_{j+1/2} = \frac{1}{2} \left[ \hat{F}(q_{j+1}) + \hat{F}(q_j) - |\tilde{A}|_{j+1/2} (q_{j+1} - q_j) \right] \quad (3.34)$$

Linearization of the flux at this face will give contributions to  $\Delta q_j$  and  $\Delta q_{j+1}$  in equation (3.33). The flux Jacobians are,

$$\frac{\partial \hat{F}}{\partial q_j} = \frac{1}{2} \left[ \hat{A}_j + |\hat{A}|_{j+1/2} + \frac{\partial |\hat{A}|_{j+1/2}}{\partial q_j} (q_{j+1} - q_j) \right]$$

$$\frac{\partial \hat{F}}{\partial q_{j+1}} = \frac{1}{2} \left[ \hat{A}_{j+1} - |\hat{A}|_{j+1/2} - \frac{\partial |\hat{A}|_{j+1/2}}{\partial q_{j+1}} (q_{j+1} - q_j) \right]$$

This gives the exact linearization for a first order flux. The last term in each Jacobian above is small compared to the first two terms and is neglected. This gives the approximate flux Jacobian for a cell face.

The second Jacobian scheme begins with the full flux at a face given,  $\hat{F}$ . For an upwind scheme, like the one used here, the flux can be represented as,

$$\hat{F} = \hat{F}^+ + \hat{F}^- \quad (3.35)$$

In this definition,  $\hat{F}^+$  represents the flux contribution from information traveling on characteristics running in the positive direction, while  $\hat{F}^-$  is for characteristics running in the negative direction. The approximate flux Jacobian for a flux at the  $(j + 1/2)$  face will be,

$$\frac{\partial \hat{F}}{\partial q_j} = (\hat{A}^-)_{j+1/2} \quad \frac{\partial \hat{F}}{\partial q_{j+1}} = (\hat{A}^+)_{j+1/2}$$

The advantage of this formulation is that the Jacobians can be evaluated with higher order approximations of  $q$ . In the above Jacobians,  $(\hat{A}^-)_{j+1/2}$  will be evaluated with  $q^L$  and  $(\hat{A}^+)_{j+1/2}$  with  $q^R$  for that face.

Both inviscid flux Jacobian schemes were implemented and compared. It was found that both provided good linearizations for the inviscid fluxes (convergence was nearly identical). The second flux Jacobian scheme was favored over the first due to its simplicity in coding and its ability to incorporate the higher order reconstruction for the  $q$  vector.

The viscous Jacobians are also approximated due to the complexity of the viscous flux calculation. The non-orthogonal terms are neglected, giving only the contributions to the two adjacent cells surrounding a cell face. This is a valid approximation for the viscous terms since the mesh is nearly orthogonal in the near wall region due to the clustering of cells. The flux Jacobian contribution for each cell is given by,

$$\frac{\partial \hat{F}_v}{\partial q} = \hat{A}_v = (\mu + \mu_t) \begin{bmatrix} 0 & 0 & 0 \\ 0 & \Delta s & 0 \\ 0 & 0 & \Delta s \end{bmatrix} \frac{1}{\mathcal{V}}$$

where  $\Delta s$  is the face area and  $\bar{\mathcal{V}}$  is the cell volume averaged to the cell face. The Jacobian above gives a positive contribution to the main diagonal term and a negative contribution to the off-diagonal terms. Hence the Jacobian adds stability to the linear system allowing larger time steps to be taken to enhance the convergence rate.

### 3.7 Linear Solvers

The linear system represented by (3.33), assuming an implicit scheme is used, can be solved by a number of techniques. This system could be solved directly, requiring the decomposition of the left-hand-side matrix. For most problems the memory requirements for a direct solution of the linearized system would be very large. In most cases, the exact solution is not necessary since the solution to the linearized system is an approximation to the overall problem, which is non-linear. Alternate techniques are therefore used instead of direct methods. Two algorithms were incorporated into the code for solving the linear system: the approximate factorization scheme and the line Gauss-Seidel method. Several other schemes have been implemented by Rogers and are discussed in [60].

The approximate factorization scheme is a popular linear solver used successfully in both compressible and incompressible flows. Details of the algorithm will not be presented here, but can be found in [61]. The method factors equation (3.33) into two sets of linear equations which are then solved sequentially. A factorization error results, but will have no effect on steady state calculations once the solution is converged. For time accurate calculations, the error term cannot be ignored. In general, the approximate factorization scheme is second order accurate in time. A problem with the pseudo compressibility formulation arises with  $\beta$ . This term appears in the factorization term as  $\beta^2$ . For large values of  $\beta$ , which are desirable for the optimum rate of convergence, the error term can reduce the time accuracy of the problem. This problem is not present in the Gauss-Seidel method which will therefore be used for all time dependent problems.

The line Gauss-Seidel method is simple to implement and has been known to work well for a wide class of problems. The scheme is based on sweep directions, which will

be in either the  $\xi$  or  $\eta$  coordinate directions. To illustrate the procedure, assume the sweeps are in the  $\xi$  ( $j$  index) direction. This is graphically shown in Figure 3.2, which shows the current status of the cells during a sweep. New values for the cell centers are at time level  $n + 1$  while cells that have not yet been updated are at time level  $n$ . Lines are formed which contain cells which vary in the  $\eta$  ( $k$  index) direction only (constant  $j$  index cells). The group of cells that form the current “line” of unknowns to solve for at the current sweep location are indicated by solid circles in the figure. The cells that have been previously updated are indicated by solid squares. For each line, a system is formed which solves for the values along that line. The matrix will be tri-diagonal, which can be efficiently solved by direct methods. For the system at line  $j$ , values at  $j - 1$  and  $j + 1$  are used (this results from the first order linearizations). The current values are always used, which is the basis of the Gauss-Seidel algorithm. Once a sweep is made, all the cells will be updated to level  $n + 1$ . The procedure is iterative such that multiple sweeps are made to attain a good estimate for the unknowns. To maintain stability for larger time steps, each coordinate sweep is done in the forward and then backward direction. This can be repeated for a set number of sweeps. The technique was illustrated for sweeps in the  $\xi$  direction, but can just as easily be done in the  $\eta$  direction.

### 3.8 Boundary Conditions

For the artificial compressibility algorithm, the equations are hyperbolic. Characteristic waves are therefore present with finite speeds and direction. At a grid boundary, characteristic waves will be either entering or leaving the domain. The waves can be thought of as carrying information about the flow field [62]. At a boundary, the information carried by the characteristics will reflect either the freestream conditions, imposed conditions, or conditions inside the computational domain. The boundary conditions need to be applied so that the unknown variables at a boundary reflect the correct information. For an incompressible flow, the most common variables that are specified at a boundary are the  $u$  and  $v$  velocity components, static pressure, and total pressure. Since only the 2-D incompressible fluid flow equations are being solved

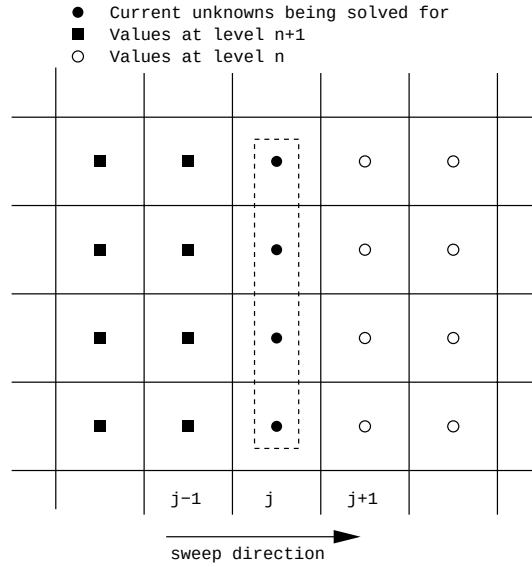


Figure 3.2: Illustration of line Gauss-Seidel sweep algorithm.

here, three independent variables or flow conditions will need to be specified at each boundary.

The first boundary condition to be discussed is flow entering into the domain. This is commonly referred to as the inflow boundary condition. For the hyperbolic equations (3.5), (3.6), and (3.15), there will be a total of three characteristic waves, one associated with each equation. Along any line in the flowfield, two of the characteristic waves will travel in the direction of the flow along that line, while the third wave will travel in the opposite direction. At an inflow boundary, this will mean that two waves will be entering into the domain, while one wave will be leaving. Two pieces of information will therefore come from outside the grid domain. The information will come from either the freestream conditions or some other imposed value. The information that is taken from inside the domain is done by extrapolation to the boundary face. Two different sets of inflow conditions are available for the present code. The first specifies the value of  $u$  and  $v$  and extrapolates  $p$  from the interior. The velocity can be specified as either the freestream condition or a given velocity profile. The second inflow condition sets total pressure to a constant, and the normal component of velocity to zero. The remaining velocity component is extrapolated

from the interior cells. With the velocity known at the boundary after extrapolation, the pressure can be found using the value for total pressure. This boundary condition is useful when the inflow velocity profile is unknown.

Boundaries that have flow leaving the grid domain are known as outflow boundaries. One characteristic wave will be traveling into the flow domain from the outside while two are traveling out of the domain. The outflow boundary condition will have only one variable specified, the pressure. The  $u$  and  $v$  velocity components will be extrapolated from the interior cells. This is the most commonly used condition for flow leaving a boundary.

The third type of boundary is a wall or solid surface. If the flow is inviscid, then a condition is imposed that no flow passes through the surface. This is known as the tangency or non-penetration condition. The normal component of velocity to the boundary face is set to zero. The tangential component of the flow is simply the magnitude of the flow vector at the wall. One way to attain this value is by extrapolating the magnitude of the cell centered velocity vectors to the wall boundary. The remaining condition is that the pressure gradient normal to the wall be zero. For viscous flows, the boundary conditions become much more simple. The  $u$  and  $v$  velocities are set to zero, and the same condition for pressure is imposed as for the inviscid wall condition. This last boundary condition is called the no-slip condition.

The extrapolation of information to the boundary face can be done with first or second order accuracy. For first order spatial accuracy, the value at the boundary face is simply the cell centered value of the adjacent cell to the boundary. For higher order accuracy, additional cells are used.

For explicit schemes, the boundary values are updated explicitly after the cell center values are updated. For implicit schemes, it is important to linearize the boundary conditions and include them into the linear system of equations represented by equation (3.33). This will increase the stability of the system and allow for larger time steps to be taken to speed up convergence. The boundary linearizations are required to be first order, following the linearizations of the flux Jacobians. Second order boundary conditions can still be attained by using higher order extrapolation formulas when information is taken from the interior of the domain. The boundary

values are updated using the new values at the cell centers.

## 3.9 Test Cases

Several test cases are now discussed to show the performance of the artificial compressibility method presented in this chapter. The test problems were chosen to verify the steady and unsteady capabilities of the code. The test cases include flow over a circular arc bump, a NACA 0012 airfoil, a 1-D oscillating channel, and laminar flow over a flat plate. All computational data is nominally third order accurate in space and second order accurate in time for the unsteady case.

### 3.9.1 Circular Arc Bump

The first test case is flow over a circular arc bump. The flow is assumed to be inviscid and steady. The grid consisted of 80 cells in the streamwise direction and 32 cells in the direction normal to the bump. A total of 31 points were used to describe the bump section. The inflow boundary is located two units upstream of the bump and the exit plane is two unit downstream where one unit is the length of the bump section. The boundary opposite to the arc bump is two units away. A value of  $\beta = 1000$  was used. A plot of the pressure contours over the bump is given in Figure 3.3 and the pressure coefficient is given in Figure 3.4. Values of  $C_p$  are compared to a solution of the full potential equation on the same mesh. The potential flow solver used here is explained in more detail in Chapter 8. The potential equation is second order accurate while the Euler solution is spatially third order accurate. This difference in accuracy is observed in the peak  $C_p$  value.

### 3.9.2 NACA 0012 Airfoil

The next test case is a NACA 0012 airfoil. Solutions to the airfoil were attained on a C-type grid which had 32 cells in the normal direction and 144 cells in the wrap-around direction. The farfield boundary was five chord lengths away from the airfoil, and used the inflow and outflow boundary conditions. On the surface of the

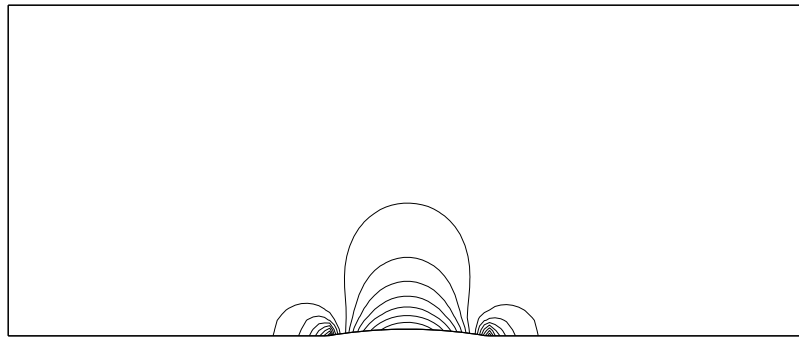


Figure 3.3: Pressure contours for incompressible flow over a circular arc bump of  $t/c = 0.042$ .

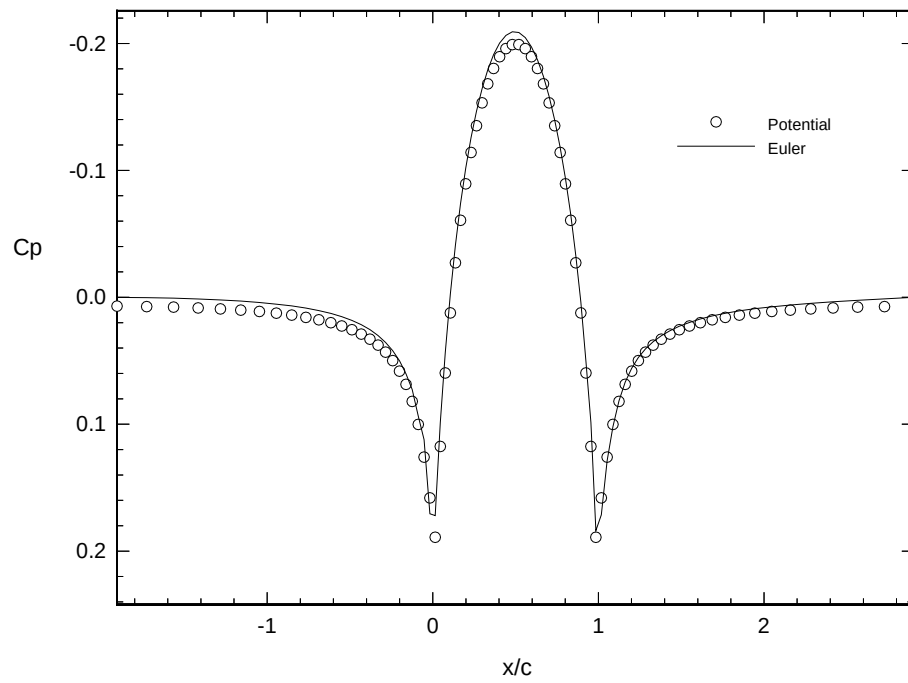


Figure 3.4:  $C_p$  comparison between incompressible Euler and Full Potential solutions for circular arc bump of  $t/c = 0.042$ .

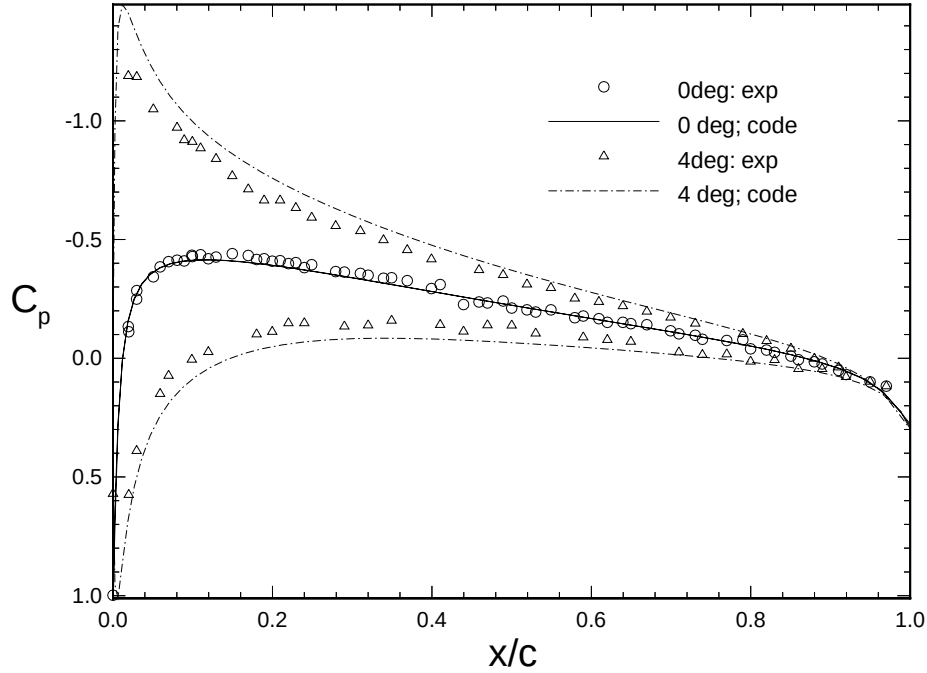


Figure 3.5:  $C_p$  comparison between incompressible Euler and experiment at  $0^\circ$  and  $4^\circ$  angle of attack for NACA 0012 airfoil. Experimental data taken from Reference [1]

airfoil the tangency condition was enforced. An inviscid solution was attained from the code and compared to experimental data at two angles of attack. Figure 3.5 shows the  $C_p$  distribution for the airfoil at  $M = 0.3$ , and at angles of attack of zero and four degrees. The experimental data was taken from reference [1]. The artificial compressibility code was ran with  $\beta = 1000$  and a CFL number of 200.

Good agreement is seen for the zero angle of attack case. For the 4 degree case, differences occur between the two solutions. This is due to the viscous effect which becomes more apparent as angle of attack increases. Comparison with the data does reflect the correct  $C_p$  profile. The pressure contours at  $4^\circ$  is shown in Figure 3.6.

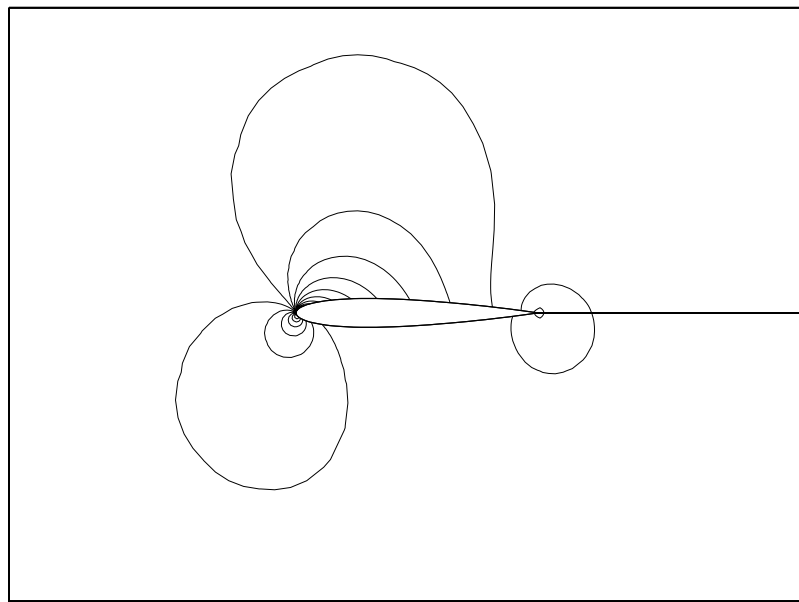


Figure 3.6: Pressure contours for NACA 0012 airfoil at  $4^\circ$  angle of attack for the incompressible Euler equations.

### 3.9.3 One Dimensional Channel

To verify the unsteady capability of the code, flow through an inviscid one-dimensional channel was calculated. This test problem was used by Merkle and Athavale [63] and Rogers and Kwak [54] to verify their methods. An exact analytical solution exists for the flow when the magnitude of the back-pressure oscillation remains small compared to the mean back pressure. The exact solution is given by

$$u(t) = 1 - \frac{p_e}{1 + \omega^2} [\sin(\omega t) - \omega \cos(\omega t) - \omega e^{-t}] \quad (3.36)$$

$$p(x, t) = p_o + p_e \sin(\omega t) + (x - 1) \frac{p_e \omega}{1 + \omega^2} [\cos(\omega t) + \omega \sin(\omega t) + e^{-t}] \quad (3.37)$$

The mean back pressure is  $p_o$  and the instantaneous back pressure is  $p_e$ . The test conditions for the present calculation had a  $p_e$  to  $p_o$  ratio of 0.1. The mean velocity was unity and the channel had a length of unity as well.

The initial conditions for pressure and velocity were found using the exact solution given in (3.36) and (3.37) for  $t = 0$ . For the inflow boundary condition, constant total pressure was held. The back pressure was set using (3.37) with  $x = 1$ . The grid consisted of 20 cells in the flow direction, and 2 cells in the normal direction. The case was run as a two-dimensional problem even though the solution is purely 1-D.

The value for  $\omega$  was 10. With this frequency, the physical time step was taken so that one complete cycle would be completed in 30 steps. This follows the calculation by Rogers and Kwak. The time step is therefore given by

$$\Delta t = \frac{\pi}{15\omega}$$

The pseudo-time step was set to  $10^{12}$  and 10 sub-iterations were done for each physical time step. This ensured that the RMS norm of the velocity divergence was less than  $10^{-6}$ .

Figure 3.7 shows the channel velocity as a function of time for a number of cycles. Excellent agreement is seen between the computed solution and the exact solution. The value of  $\beta$  for the given solution is 10.  $\beta$  was allowed to vary as high as 1000 with little loss in accuracy of the solution.

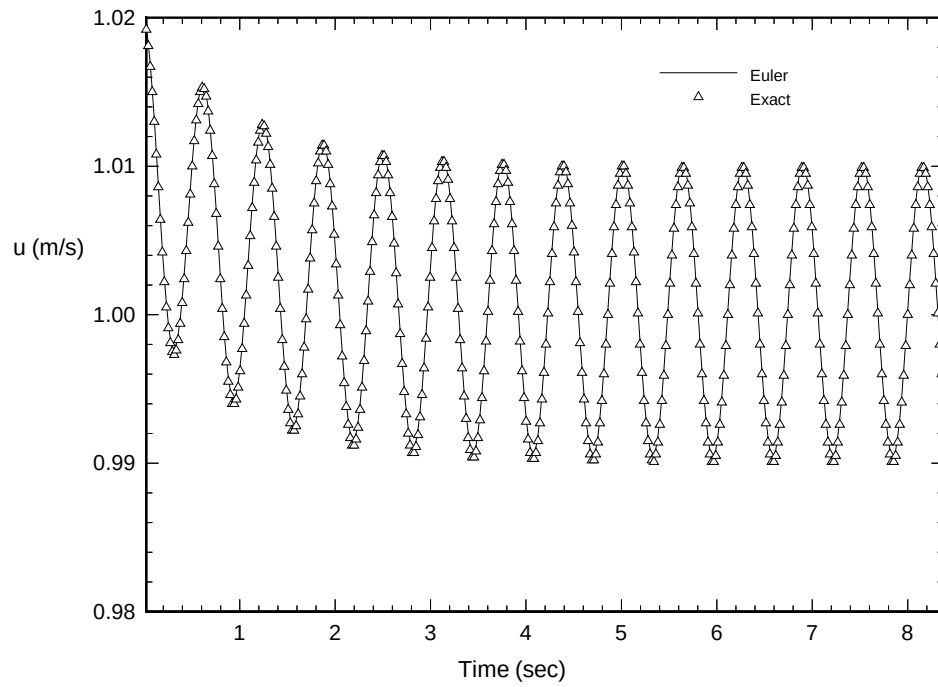


Figure 3.7: Velocity values in a 1-D channel with oscillating back pressure for the incompressible Euler equations using second order time accuracy.

### 3.9.4 Laminar Flat Plate

One of the most basic test cases for laminar flow is the steady, constant property flow over a flat plate. For this flow, no pressure gradient exists and so a constant boundary layer edge velocity will occur. This flow also gives rise to similar solutions. With the proper non-dimensionalization, all the profiles along the plate can be represented by a single curve. The exact solution for this flow is known and a more complete discussion of it can be found in reference [64]. The solution to this constant property, flat plate flow is known as the Blasius solution, after H. Blasius who discovered it.

The numerical results for this test case are compared to the Blasius solution in Figure 3.8. The profile is nondimensionalized by the following parameter in the normal direction.

$$\eta = y \sqrt{\frac{U_e}{2\nu x}}$$

The velocity parallel to the plate is nondimensionalized by the edge velocity,  $U_e$ , which is also the freestream velocity for this particular case. Excellent agreement exists between the code and the exact solution.

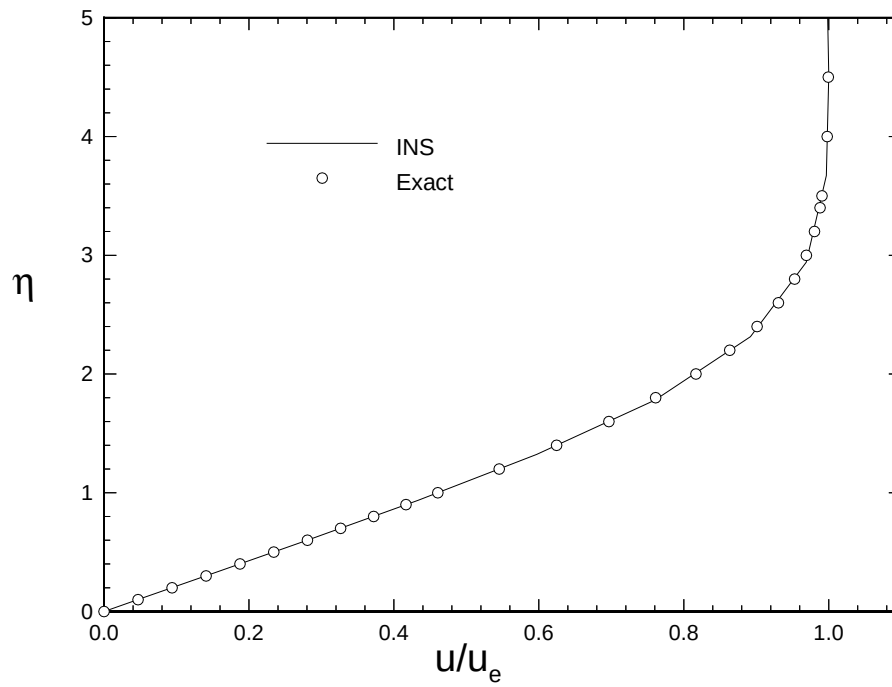


Figure 3.8: Comparison of incompressible Navier-Stokes with Blasius solution for laminar flow over a flat plate.

# Chapter 4

## Turbulence Models

### 4.1 Introduction

The turbulence models used in the flow solver will be described in this chapter. Two models will be discussed in detail, the Baldwin-Lomax model (BLM) and the Johnson and King model (JKM). Special emphasis is given to the JKM model since modifications to the model will also be presented in this chapter. The chapter will conclude with a demonstration of the turbulence models by computing the turbulent flow over a flat plate.

### 4.2 Baldwin-Lomax Model

The Baldwin and Lomax turbulence model [65] is an algebraic eddy viscosity, zero equation model. The model is based on the mixing length hypothesis, and is very similar to the Cebeci-Smith model [66]. The model has been widely used and tested, and remains a very popular turbulence model. Several advantages of the model are its simplicity in coding and the fact that it does not require the boundary layer edge thickness or edge velocity as an input parameter. The calculation of  $\delta$  is not always straightforward and can lead to errors in the turbulence model if not computed correctly. The Baldwin and Lomax model works best in wall bounded flows with favorable pressure gradients. As the flow physics become more complicated, as well

as the geometry of the model being tested, the performance of this turbulence model greatly decreases.

The Baldwin-Lomax model contains an inner and outer model for the turbulent boundary layer region. These are given as,

$$\mu_{to} = \rho K C_{cp} F_{wake} F_{kleb}(y) \quad (4.1)$$

$$\mu_{ti} = \rho l^2 |\omega| \quad (4.2)$$

The switch from the inner viscosity to the outer viscosity is made when  $\mu_{ti}$  becomes larger than  $\mu_{to}$ .

For the inner region, the following terms are defined.

- $l = \kappa y \left[ 1 - \exp\left(\frac{-y^+}{A^+}\right) \right]$
- $|\omega| = \left| \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right|$
- $y^+ = \frac{\rho u^* y}{\mu}$

And for the outer region, the following terms are defined.

- $F_{wake} = \min \left[ y_{max} F_{max}, C_{wk} y_{max} \left( \frac{U_{diff}^2}{F_{max}} \right) \right]$
- $F_{max} = \max F(y)$
- $F(y) = y |\omega| \left[ 1 - \exp\left(\frac{-y^+}{A^+}\right) \right]$
- $y_{max} = \text{value of } y \text{ at } F_{max}$
- $F_{kleb}(y) = \left[ 1 + 5.5 \left( \frac{y C_{kleb}}{y_{max}} \right)^6 \right]^{-1}$
- $U_{diff} = \left[ (u^2 + v^2)^{\frac{1}{2}} \right]_{max}$  (for solid wall calculations)

As seen above, the outer-layer length scale is no longer directly computed from the boundary layer thickness, but is based on the vorticity in the layer.

The following list contain the modeling constants for the Baldwin-Lomax turbulence model.

$$A^+ = 26$$

$$C_{cp} = 1.6$$

$$C_{kleb} = 0.3$$

$$C_{wk} = 0.25$$

$$\kappa = 0.4$$

$$K = 0.0168$$

### 4.3 Johnson and King Model

The Johnson and King turbulence model [18] is known as a half-equation model. It is composed of an algebraic eddy viscosity equation and a differential equation. The differential equation has a time dependent term which allows unsteady effects to be taken into account within the model. For steady flows, the differential equation reduces to an ordinary differential equation (ODE). The differential equation describes the streamwise development of the maximum Reynolds shear stress and is used to scale the eddy viscosity in the outer portion of the boundary layer. The scaling takes into account the nonequilibrium effects when the flow changes rapidly, as in the case of a strong adverse pressure gradient. Nonequilibrium conditions occur when the turbulence production no longer equals the turbulent dissipation energy. Under these conditions the traditional eddy viscosity models fail to correctly predict the actual flow physics. The Johnson and King model is specifically intended to model flows with strong adverse pressure gradients or flows where separation is present. It is in these flow regimes that nonequilibrium effects must be accounted for and properly modeled.

The Johnson and King turbulence model will be described next. The formulation that is discussed will be referred to as the baseline JKM (Base JKM). The baseline JKM is the original model introduced by Johnson and King [18] along with the correction made by Johnson and Coakley [34]. Any changes that are made to the baseline Johnson and King model will be referred to as the modified JKM (Mod JKM).

### 4.3.1 Formulation

For the Johnson and King model, the equation for the eddy viscosity is:

$$\nu_t = \frac{\mu_t}{\rho} = \nu_{to} \left[ 1 - \exp\left(\frac{-\nu_{ti}}{\nu_{to}}\right) \right] \quad (4.3)$$

where  $\nu_{to}$  and  $\nu_{ti}$  are the outer and inner viscosities respectively and are defined by,

$$\nu_{to} = \sigma(x) K U_e \delta^* \gamma \quad (4.4)$$

$$\nu_{ti} = D^2 \kappa y u_s \quad (4.5)$$

This provides for a smooth blending of the inner and outer viscosities. A different form for the outer eddy viscosity was formulated by Yu *et al.* [67], but was found to be less robust and stable than the original formulation by Johnson and King. In the above equations,  $\gamma$  is Klebanoff's intermittency function,  $D$  is the near wall damping term defined as

$$D = 1 - \exp\left[\frac{-y u_D}{\nu A^+}\right]$$

and  $u_s$  is the velocity scale parameter. The velocity is scaled on both the Reynolds shear stress and the wall shear velocity [34]. A blending between the two velocity scales is used for a smooth transition. The equation for  $u_s$  is

$$u_s = u^* (1 - \gamma_2) + (-\overline{u'v'}_m)^{\frac{1}{2}} \gamma_2 \quad (4.6)$$

where the term  $\gamma_2$  is given as  $\gamma_2 = 1 - \exp(\frac{-y}{L_c})$  and  $u^*$  is the wall shear velocity. The term  $L_c$  is given by

$$L_c = \frac{u^* y_m}{u^* + (-\overline{u'v'}_m)^{\frac{1}{2}}} \quad (4.7)$$

and finally  $u_D$  is

$$u_D = \max[u^*, (-\overline{u'v'}_m)^{\frac{1}{2}}] \quad (4.8)$$

The subscript  $m$  indicates that the variable is evaluated at the  $y$  location where the Reynolds shear stress is a maximum. The maximum shear stress location will vary in the streamwise direction. The value of  $A^+$  is 15,  $K$  is 0.0168, and that of von Karman's constant,  $\kappa$ , is 0.40.

The Johnson and King model requires the solution of the following differential equation,

$$\frac{L_m}{\tau_m a_1} \left( \frac{\partial \tau_m}{\partial t} + \bar{u}_m \frac{\partial \tau_m}{\partial x} \right) = (\tau_{m,eq})^{\frac{1}{2}} - (\tau_m)^{\frac{1}{2}} - \frac{L_m D_m}{\tau_m} \quad (4.9)$$

where  $\tau_m = (-\overline{u'v'})_m$  and the turbulent diffusion term,  $D_m$ , is given by:

$$D_m = \frac{C_{dif}(\tau_m)^{\frac{3}{2}}}{a_1 \delta [0.7 - (\frac{y}{\delta})_m]} |1 - \sigma(x)^{\frac{1}{2}}| \quad (4.10)$$

This partial differential equation will reduce to an ordinary differential equation if the time dependent term is dropped, which is normally the case. The time derivative term is retained here since time accurate solutions are desired. A simple transformation can be performed to aid in the solution of the above equation. The following change of variables are made:

$$g = (\tau_m)^{-\frac{1}{2}} \quad \text{and} \quad g_{eq} = (\tau_{m,eq})^{-\frac{1}{2}}$$

With this transformation, the differential equation becomes,

$$\frac{\partial g}{\partial t} + \bar{u}_m \frac{\partial g}{\partial x} + \frac{a_1}{2L_m} \left[ \left( \frac{g}{g_{eq}} - 1 \right) - \frac{C_{dif} L_m |1 - \sigma(x)^{\frac{1}{2}}|}{a_1 \delta [0.7 - (\frac{y}{\delta})_m]} \right] = 0 \quad (4.11)$$

In the above equations,  $C_{dif}$  is a modeling constant taken as 0.5 and  $L_m$  is a dissipation length scale. This length scale is based on the maximum Reynolds shear stress height for the inner region and the boundary layer thickness for the outer region. The values for  $L_m$  are,

$$L_m = 0.4y_m \quad \frac{y_m}{\delta} \leq 0.225 \quad (4.12)$$

$$L_m = 0.09\delta \quad \frac{y_m}{\delta} > 0.225 \quad (4.13)$$

The differential equation is used to control the value of  $\sigma$  at each streamwise location in the flow. Therefore, only  $\nu_{to}$  is affected by the solution of the differential equation. This makes the outer eddy viscosity strongly dependent on the development of the Reynolds shear stress instead of just the mean velocity profile. The eddy viscosity model is used to determine the shear stress and the ODE is then used to control the level of the shear stress through the  $\sigma$  parameter. At each  $x$  location

a value of  $\sigma$  is used to scale equation (4.4) such that the following relationship is satisfied,

$$\nu_{t,m} = \frac{(-\overline{u'v'})_m}{(\partial u / \partial y)_m} \quad (4.14)$$

There are several ways to calculate the value of  $\sigma$ . Two of these ways are presented below:

(I)

$$\sigma(x)^{n+1} = \sigma(x)^n \frac{\tau_{m,ODE}}{\tau_{m,actual}} \quad (4.15)$$

(II)

$$\sigma(x)^{n+1} = \sigma(x)^n \frac{\nu_{to,m}^{k+1}}{\nu_{to,m}^k|_{k=1}} \quad (4.16)$$

where the following terms are defined in the second method,

$$\nu_{to,m}^{k+1} = \nu_{to,m}^k \frac{\tilde{\nu}_{t,m}}{\nu_{t,m}^k} \quad (4.17)$$

$$\nu_{t,m}^{k+1} = \nu_{to,m}^{k+1} [1 - \exp(\frac{-\nu_{ti,m}}{\nu_{to,m}^{k+1}})] \quad (4.18)$$

In the second method, only several iterations for  $k$  are needed to update  $\sigma(x)$ . The value of  $\tilde{\nu}_{t,m}$  is obtained using the maximum strain rate and  $\tau_m$  is found from the solution of the differential equation. Both methods were found to be equally adequate for computing  $\sigma(x)$ . For the results using the JKM, method I will be used due to its simplicity over method II.

### 4.3.2 Solution to ODE

There are various options to solving the ODE for the Johnson and King model. One method that is simple to program and has been found to work well is the Euler implicit algorithm. This method will be presented here. Starting with equation (4.11), define the following terms:

$$C_1 = \frac{-a_1}{2L_m \bar{u}_m g_{eq}} \quad (4.19)$$

$$C_2 = \frac{a_1}{2L_m \bar{u}_m} + \frac{C_{dif} |1 - \sigma(x)^{\frac{1}{2}}|}{2\bar{u}_m \delta [0.7 - (\frac{y}{\delta})_m]} \quad (4.20)$$

The ODE, with the above terms substituted in, becomes the following,

$$\frac{1}{\bar{u}_m} \frac{\partial g}{\partial t} + \frac{\partial g}{\partial x} = C_1 g + C_2 \quad (4.21)$$

For Euler Implicit, the unknowns are written at time level  $n+1$  where time level  $n$  are the known values of  $g$  before solving the ODE. The time derivative and convective term is discretized using a first order one-sided difference. The equation after the discretization becomes:

$$\frac{1}{\bar{u}_m} \frac{g_j^{n+1} - g_j^n}{\Delta t} + \frac{g_j^{n+1} - g_{j-1}^{n+1}}{\Delta x} = C_1 g_j^{n+1} + C_2 \quad (4.22)$$

The index  $j$  represents the current cell and  $j-1$  represents the upwind direction. The above equation is solve for  $g_j^{n+1}$  at each cell location.

### 4.3.3 Implementation of Model

The algorithm is now presented for which  $\nu_t$  is calculated. Note that values of  $\sigma(x)$ ,  $g$ ,  $g_{eq}$ , and  $\nu_t$  are stored from the previous iteration.

1. Strain rate is calculated:  $\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}$
2. Boundary layer edge properties are found:  $\delta, U_e, \delta^*$
3. Reynolds shear stress is calculated using the strain rate and the previous value of  $\nu_t$ :  $(-\overline{u'v'}) = \nu_t (\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x})$   
 $(-\overline{u'v'})_m$  can then be found at each streamwise location.
4. Compute:  $u_{max}, y_{max}$ , and  $\Delta x_{max}$  at each streamwise location. Values correspond to the  $y$  location where  $(-\overline{u'v'})$  is maximum. Also calculate  $L_m$ .
5. Calculate equilibrium Reynolds shear stress  $(-\overline{u'v'})_{eq}$ :

$$(-\overline{u'v'})_{eq} = \nu_{t,eq} \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)$$

$$\nu_{t,eq} = \nu_{to,eq} \left[ 1 - \exp\left(\frac{-\nu_{ti,eq}}{\nu_{to}}\right) \right]$$

$$\nu_{to,eq} = KU_e \delta^* \gamma$$

$$\nu_{ti,eq} = D^2 \kappa y u_s$$

The variable  $(-\overline{u'v'_m})^{\frac{1}{2}}$  in  $u_s$  is from  $(-\overline{u'v'_m})_{eq}$  in the previous iteration. Values for  $g_{eq}$  are stored at this point.

6. Solve the differential equation for  $g$ . Store values of  $g$  at this point.
7. Compute a new  $\sigma(x)$  and store values.
8. Calculate new eddy viscosity,  $\nu_t$ . Use the value for  $(-\overline{u'v'_m})$  found in (3) for the  $u_s$  term.

An initial boundary layer profile must be given to start the calculations. The method which was implemented here found a boundary layer solution for the given problem using the Baldwin-Lomax model. Calculations were then restarted using the JKM. Initial values for  $g$  and  $g_{eq}$  were found from the Baldwin-Lomax solution.  $\sigma(x)$  was set to unity for the first iteration.

## 4.4 Modifications to the JKM

Modifications made to the baseline Johnson and King turbulence model are now discussed. These modifications take on the form of changes made to the existing turbulence model, or additional terms added to the JKM. A change is also proposed to the modeling of the Reynolds normal stress in the momentum equations. These modifications fall into three categories (discussed below) which help clarify the presentation of each change or addition made.

The purpose of the JKM is to produce an eddy viscosity which is then added to the laminar viscosity in the momentum equation. The eddy viscosity is composed of two parts, referred to as the inner and outer eddy viscosities. Both the inner and outer regions are composed of a length scale and a velocity scale. The outer model has the addition of a scaling parameter,  $\sigma$ , which is a multiplier of the outer eddy viscosity. Changes will first be presented that deal with the outer region eddy viscosity model.

Production of energy from the Reynolds normal stress is included into the outer eddy viscosity. These changes will focus on the differential equation used to compute the scaling parameter  $\sigma$ . Next, an addition to the turbulence modeling is suggested that is independent of the JKM. This new term is a model for the normal stress that appears in the  $x$ -momentum equation after Reynolds averaging. Finally, changes to the inner region will be discussed. A new velocity scale will be proposed, along with a term that is added to the inner model. The added term will focus on effects from the strong pressure gradient.

#### 4.4.1 Outer Region

It has been observed from experiment [16] that the normal stress production term becomes important in regions of strong adverse pressure gradient and separation. In these regions, the normal stresses can contribute as much as 30% or more of the total turbulent energy production. The ratio of the total turbulence energy production to shear stress production is defined as

$$F = 1 - \frac{(\overline{u'^2} - \overline{v'^2})\partial U/\partial x}{-\overline{u'v'}\partial U/\partial y} \quad (4.23)$$

Shiloh *et al.* [68] uses the  $F$  factor at  $-\overline{u'v'}_m$  to make the following correlation between the Reynolds shear stress and the TKE,

$$\frac{-\overline{u'v'}}{\overline{q^2}} F^\alpha = A_2 = 0.15 \quad (4.24)$$

where  $\alpha$  is best approximated by 1.25 from experimental observations [68]. An expression is also given for the normal stresses relating  $F$  and  $\overline{q^2}$  as,

$$\overline{u'^2} - \overline{v'^2} = \frac{C_2 \overline{q^2}}{F^{0.25}} \quad (4.25)$$

Using equations (4.23)–(4.25),  $F$  can be re-written as the following,

$$F = \left[ 1 + \frac{C_2}{A_2} \frac{\partial U/\partial x}{\partial U/\partial y} \right]^{-1} \quad (4.26)$$

where the ratio  $C_2/A_2$  is closely approximated by a value of 2.79 according to experiment.

The equation for the maximum Reynolds shear stress as given in equations (4.9) and (4.10) does not take into account production from the normal stresses, but only the shear stresses. To include the normal stress terms, several modifications are done to equation (4.9). Johnson and King used the assumption that  $\tau_m/k_m = 0.25$ , which leaves out the  $F$  factor. In this work, the following substitution is made instead,

$$\tau_m/k_m = 2A_2/F^\alpha \quad (4.27)$$

where again  $A_2$  has the value of 0.15. The second modification is in the shear production term given by  $\tau_m \partial U / \partial y$ . This term can be multiplied by the  $F$  factor in order to take into account the normal stress contributions to the production, giving

$$F\tau_m \partial U / \partial y \quad (4.28)$$

These corrections are made to equation (4.9) which can now be written as,

$$\frac{L_m \bar{u}_m F^\alpha}{2\tau_m A_2} \frac{\partial \tau_m}{\partial x} = F(\tau_{m,eq})^{\frac{1}{2}} - (\tau_m)^{\frac{1}{2}} - \frac{L_m \mathcal{D}_m}{\tau_m} \quad (4.29)$$

where the time dependent term has been dropped for simplicity. More details on this derivation can be found in Appendix A. It was observed that this modification, when used throughout the flow, caused separation to occur earlier than the baseline Johnson and King model. If equation (4.29) was used only when backflow was present, then the model had less impact on the detachment location ( $C_f = 0$ ) and more emphasis on the backflow region. Therefore, the modified ODE was used in the JKM only in the separated regions where backflow was present.

#### 4.4.2 Momentum Equation

Another location where the normal stresses can be considered is in the Reynolds averaged Navier Stokes equations. In the x-momentum equation, two terms appear after performing Reynolds averaging. These two terms appear on the left hand side of the equation as

$$\frac{\partial(-\overline{u'v'})}{\partial y} + \frac{\partial(-\overline{u'^2})}{\partial x} \quad (4.30)$$

and can be considered as the apparent shearing and normal stresses, respectively. It was shown in equation (3.8) how the shearing term was modeled using an eddy viscosity type formulation. The normal stress is modeled in a similar manner giving,

$$(\overline{-u'^2}) = \frac{\mu_t}{\rho} \frac{\partial u}{\partial x} \quad (4.31)$$

It is suggested that this does not properly model the normal shearing stress in regions around separation. A new approximation is therefore made in order to better model the normal stress effects. From equations (4.24) and (4.25), an expression can be formed for the normal stress term. For strong adverse pressure gradients and separated flows, the following equation is given,

$$\overline{u'^2} - \overline{v'^2} = \frac{C_2}{A_2} F(\overline{-u'v'}) \quad (4.32)$$

Observing the experimental data for the present flow being studied, the normal stresses from  $u$  are much larger than those stresses from  $v$  [68]. It is also observed that the normal stresses only become large in the outer portion of the boundary layer [69]. Using these two observations, the following model was made for the normal stress term:

$$\frac{\partial(\overline{-u'^2})}{\partial x} = -\frac{C_2}{A_2} \frac{\partial}{\partial x} [F(\overline{-u'v'})] \quad (4.33)$$

The normal stress contributions from this model are added to all locations above the maximum Reynolds shear stress location,  $y_m$ , and set to zero for all locations below it. This is consistent with experimental observations.

### 4.4.3 Inner Region

The changes made so far have been focused on the outer eddy viscosity, since the ODE is used to compute  $\sigma(x)$  which is multiplied by  $\mu_{to}$ . The inclusion of equation (4.32) will also have a greater impact on the outer region. There still seems to be a need to improve the inner model to help better capture the near wall effects. The value of  $\mu_{ti}$  seems to be smaller than what is required to capture the near wall profiles in the region leading up to separation, but slightly larger in the separated region. The inner region velocity scale parameter,  $u_s$ , was originally set to  $(-\overline{u'v'_m})^{\frac{1}{2}}$  by Johnson and

King [18], but was later modified by Johnson and Coakley [34] in order to improve the near wall profiles. This modification to the turbulence model, which is given by equation (4.6), was used in the present computations as the baseline JKM. Johnson and Coakley improved the term  $u_s$  by making it a function of the parameters  $u^*$  and  $(-\overline{u'v'_m})^{\frac{1}{2}}$ . The reason for the improvement was due to the growth of  $(-\overline{u'v'_m})^{\frac{1}{2}}$  as separation was reached, giving a higher value for  $\mu_{ti}$ . The wall shear velocity will not grow in magnitude due to separation, making a combination of the two parameters a better choice for the velocity scale. It was found that changing the value of  $u_s$  can have a large impact on the profiles and the detachment location. It is not clear what the best parameter is for the inner region velocity scale that will capture the physics of the flow, but it was observed that the local value of the Reynolds shear stress could be used instead of a constant maximum value [70].

$$u_s = (-\overline{u'v'})^{\frac{1}{2}} \quad (4.34)$$

When the above velocity scale term was used, the model behaved well throughout the flowfield domain.

The second modification to the inner eddy viscosity is based on the pressure gradient. To develop a relation that involves the pressure, it is helpful to start with the mean flow momentum equation for steady, 2-D flow which is written as,

$$U \frac{\partial U}{\partial x} + V \frac{\partial U}{\partial y} = -\frac{1}{\rho} \frac{\partial P}{\partial x} + \nu \frac{\partial^2 U}{\partial y^2} + \frac{1}{\rho} \frac{\partial(-\rho \overline{u'v'})}{\partial y} \quad (4.35)$$

Further details of this equation can be found in [71]. Coles integrated this equation with respect to  $y^+$  to give the following,

$$\frac{\tau_l + \tau_t}{\tau_w} = 1 + \frac{y}{\rho u^{*2}} \frac{\partial P}{\partial x} + \frac{\nu}{u^{*2}} \frac{du^*}{dx} \int_0^{y^+} h^2(y^+) dy^+ \quad (4.36)$$

where  $h(y^+) = (u/u^*)$ . Solving for the turbulent shear stress and keeping all the terms yields the following equation,

$$\tau_t = \tau_w \left[ 1 + \frac{y}{\rho u^{*2}} \frac{\partial P}{\partial x} + \frac{\nu}{u^{*2}} \frac{du^*}{dx} \int_0^{y^+} h^2(y^+) dy^+ \right] - \tau_l \quad (4.37)$$

For most flows, it is common to neglect the pressure gradient and the integral term. Also note that for the inner logarithmic region,  $\tau_l$  is much smaller than the turbulent

viscosity,  $\tau_t$ . If these three terms are neglected, then what remains is simply  $\tau_t = \tau_w$ . This approximation forms the basis for most algebraic turbulence models such as the Baldwin-Lomax model, the Cebeci-Smith model, and even the Johnson and King model. Instead of neglecting the pressure gradient term in (4.37), it will be kept intact in order to see the impact on the inner region model.

The eddy viscosity formulation states that  $\tau_t = \mu_t \partial U / \partial y$ . The velocity gradient in the log region can be replaced with  $(u^* / \kappa y)$ , which comes from the law-of-the-wall. With these substitutions and neglecting the integral and laminar viscosity terms, (4.37) becomes,

$$\mu_t = \kappa \rho u^* y + \kappa \rho u^* y \left[ \frac{y}{\rho u^{*2}} \frac{\partial P}{\partial x} \right] \quad (4.38)$$

If the first term on the right hand side is multiplied by the near wall damping term,  $D$ , and the velocity scale parameter is substituted in for  $u^*$ , then this term becomes the inner eddy viscosity for the JKM. The second term is multiplied by the wall damping term as well to give the following modified equation for the inner eddy viscosity,

$$\nu_{ti} = \nu_{ti(JKM)} + \kappa u^* y D \left[ \frac{y}{\rho u^{*2}} \frac{\partial P}{\partial x} \right] \quad (4.39)$$

Note that the equation was divided through by the density. It is important to also note that the additional term in (4.39) was made based on law-of-the-wall assumptions. For this reason, the pressure term is set to zero inside the backflow region, where the law-of-the-wall is not valid. It was also found that when using the additional pressure gradient term, the wall shear velocity was the best choice for  $u_s$ .

The effects of the integral term in (4.37) were also studied. From a computational perspective, the integral term did not “behave” well and caused problems with convergence. The main problem arose from the wall shear velocity gradient. The gradient was near zero in the separated region and would switch signs. If the magnitude of the integral term ever caused  $\nu_{ti}$  to become negative anywhere in the boundary layer, numerical problems resulted. In a similar manner, the pressure gradient term could cause problems with negative values of  $\nu_{ti}$  as well. Since the pressure gradient term is of interest in adverse pressure gradient conditions, hence a positive pressure gradient, the additional pressure gradient term was set to zero whenever a favorable or negative pressure gradient occurred. The pressure gradient term was numerically observed to

behave well during computations. No other numerical problems were found using the modifications mentioned above.

It is worth noting here that the idea of incorporating both the normal stress production term and the pressure gradient was previously used by Hytopoulos [15]. Hytopoulos developed a new turbulence model based upon Prandtl's mixing length. In a similar fashion to the Johnson and King model, an auxiliary equation was solved for to scale the Reynolds stress in the outer portion of the viscous shear layer. The auxiliary equation came from an integral form of the TKE equation, and solved for the maximum Reynolds shear stress. Once the equation was solved, the mixing length was scaled such that the turbulence model predicted a maximum Reynolds shear stress equal to the one predicted by the auxiliary equation. The scaling allowed the non-equilibrium effects to be taken into account. The same  $F$  factor presented in a previous section was also used in Hytopoulos's model to take into account the production of normal stresses in the turbulent kinetic energy equation. This approach is similar to the modified Johnson and King model, except that the JKM solves for an eddy viscosity and scales the value of it based on  $\sigma(x)$ , while Hytopoulos's model solves for a mixing length, which is scaled by a mixing length parameter. The mixing length for the inner region was derived from equation (4.37), where terms in addition to the pressure gradient parameter were retained. Because of the assumptions used in developing the model, the expression for the inner region was not applicable to separated flows. The focus of the model was on flows with strong adverse pressure gradients and not separation because the model would break down shortly after flow detachment occurred. An overview of the model is presented in Appendix B.

## 4.5 Verification: Turbulent Flat Plate

To verify the Baldwin-Lomax and Johnson and King turbulence models, the subsonic flow over a flat plate was computed. The boundary layer profile at a momentum Reynolds number of  $Re_\theta = 9,650$  was chosen for comparison. A plot of the law-of-the-wall is given in Figure 4.1. The plot reveals the inner region of the boundary layer which consists of a laminar sublayer near the wall and the overlap region connecting

the inner and outer region. The analytical formula for the laminar sublayer is

$$u^+ = y^+ \quad (4.40)$$

and the expression for the overlap region is obtained using the coefficients developed by Clauser [71]. The overlap region spans roughly  $30 \approx y^+ \approx 300$  and has the following equation,

$$u^+ = 5.6 \log(y^+) + 4.9 \quad (4.41)$$

It is important in the computation of turbulent flows that there is sufficient grid resolution near the wall. This is not as important for laminar flows, but for turbulent flows the closest cell center to the wall should be around  $y^+ \approx 1$ . This is necessary to resolve the flow in the laminar sublayer, which extends up to only  $y^+ \approx 5 - 7$ . Since it is impossible to compute  $y^+$  before generating the grid, the spacing near the wall must be estimated. This is done with the following expression for the minimum distance between the wall and first grid point [72].

$$\frac{\Delta y_{min}}{L} = \frac{0.7}{\sqrt{R_{eL}}} \quad (4.42)$$

The total length of the wall surface is taken to be  $L$  and  $R_{eL}$  is the Reynolds number based on  $L$ .

From the figure, both models are seen to capture the physics associated with turbulent flows. The laminar sublayer and overlap region both agree well with the computational data.

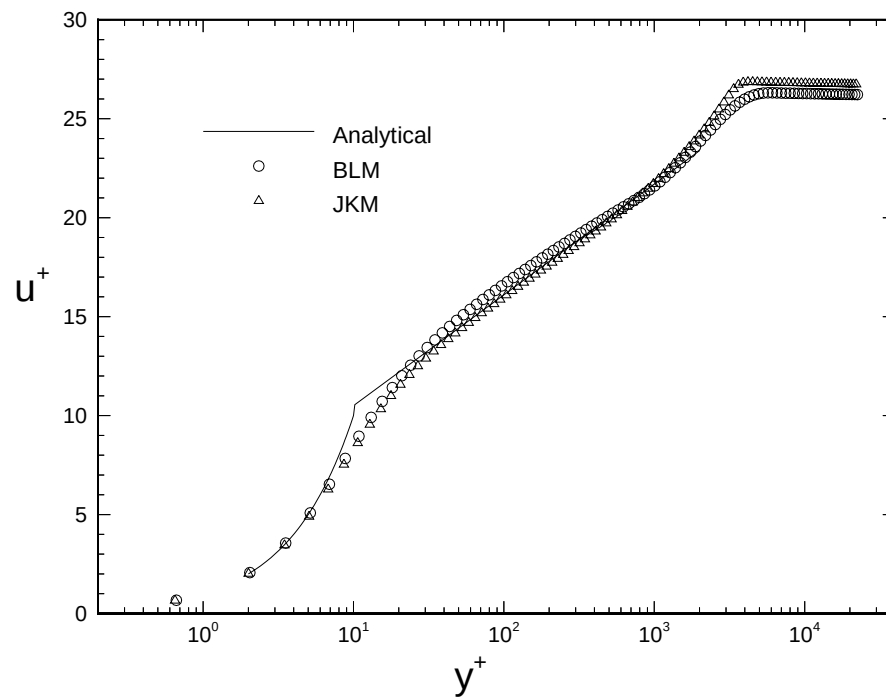


Figure 4.1: Wall law comparison of turbulence models with the analytical solution for a turbulent flat plate at  $R_{e\theta} = 9,650$ .

# Chapter 5

## Low-Speed Diffuser Results

### 5.1 Description of Flow

A low-speed diffuser flow is discussed in which numerical computations were done and compared with experimental data. The numerical computations are based on an experiment which studied steady and unsteady turbulent boundary layer flow. The steady flow was studied by Simpson, Chew, Shivaprasad, and Shiloh in [16], [69], and [68]. The unsteady case is presented in references [73] and [74]. Results from the steady flow case appeared at the 1980-81 AFSOR-HTTM-Stanford Conference on Complex Turbulent Flows. The test case was labeled as Case 0431. The experiment is a diffuser flow that has a large separated region which forms in the diverging part of the duct (see Figure 5.1). Experimental data taken for this test case allows for comparisons to be made between numerical calculations and the experiment.

The diffuser flow has the property of a large adverse pressure gradient. This is due to the rapidly expanding nozzle flow which causes the boundary layer to undergo massive separation. The flow is dominantly two-dimensional throughout the first portion of the separated region. As the flow travels further downstream, it becomes more three-dimensional and cannot be properly modeled as a two-dimensional flow. The first part of the separated region exhibits properties of a 2-D flow which makes comparisons to 2-D numerical computations very insightful.

The experimental data was taken by Simpson, Chew, and Shivaprasad at the

Table 2: Position of upper wall for low-speed diffuser.

| x, meters | y, meters | x, meters | y, meters |
|-----------|-----------|-----------|-----------|
| 0.289     | 0.2541    | 2.855     | 0.2535    |
| 0.553     | 0.2314    | 3.011     | 0.2731    |
| 0.794     | 0.2118    | 3.115     | 0.2894    |
| 1.118     | 0.1954    | 3.230     | 0.3099    |
| 1.334     | 0.1879    | 3.417     | 0.3473    |
| 1.632     | 0.1881    | 3.681     | 0.3935    |
| 1.886     | 0.1933    | 3.973     | 0.4565    |
| 2.223     | 0.2042    | 4.341     | 0.5149    |
| 2.701     | 0.2355    |           |           |

Southern Methodist University. Upstream of separation, single and cross-wire hot-wire anemometer measurements were recorded. In the separated zone, a directionally sensitive laser-anemometer system was used. Single wire data was also taken in the separated region in the outer portion of the boundary layer.

Certain features of the experiment influenced how the numerical computations were set up. Experimental data was mainly recorded on the lower wall surface, where the separation region existed. Velocity profiles near the entrance to the diffuser, as well as downstream of the detachment location determined the starting and ending boundaries for the grid. In the experiment, a streamline for the upper wall surface was determined from the boundary layer displacement thickness. This streamline was used to determine the shape of the upper wall. Table 2 gives the position of the inviscid flow streamline for the upper wall. The lower wall is a flat plate. Figure 5.1 shows the resulting computational domain.

The length of the experimental test section was 7.62 meters. The flow was tripped at the test section entrance to force a turbulent boundary layer. Experimental data for the  $u$  and  $v$  velocity profiles are given at a streamwise location of 0.289 meters. The computational region therefore begins at this location for both the steady and unsteady flow cases. Figure 5.1 shows a diagram of the computational region. As the flow passes through the throat of the diffuser, a strong adverse pressure gradient

forms due to the large expansion that the flow undergoes. From experiment, the fully developed separation location occurs at 3.55 meters for the steady flow case. The separated region continues to grow and remains strongly two-dimensional up to 4.34 meters. At this streamwise location, the flow is still separated and the boundary layer occupies over 60% of the test section height. Because of the 3-D nature of the flow beyond this location, the computational domain ends at this point. For the steady flow case, the Reynolds number based on the entrance conditions is 9.6 million per meter.

The computational grid consists of 80 cells in the streamwise direction and 100 cells in the direction normal to the flow. A grid convergence study was done and revealed that this grid provided the necessary resolution for the computational results. The upper wall boundary is treated as a streamline, so clustering of cells is done only for the lower wall boundary layer. The tangency boundary condition is used on the upper wall. The lower wall is treated with the no-slip condition. The inflow boundary uses the  $u$  and  $v$  experimental data to set the velocity profile. Pressure at the inflow is extracted from the interior cells. The outflow boundary for the diffuser is slightly more complex since it occurs inside the separated region. Flow is both leaving and entering at the downstream boundary. Therefore a combination of inflow/outflow conditions are used at the exit boundary. Where the flow is leaving the computational domain, a back pressure is specified and the  $u$  and  $v$  velocity is extracted from the interior cells. For the region near the lower wall where separation forces the flow to enter back into the boundary from outside, the  $u$  and  $v$  velocities are set using experimental data. The pressure is extracted from the interior cells. All boundary conditions are spatially second order accurate. Computational results are calculated using third order flux differencing for the interior cell faces.

Results from the Baldwin-Lomax turbulence model, as presented in section 4.2, will be presented along with the Johnson and King model and its modifications for comparison. The Baldwin-Lomax model reveals the typical behavior of an equilibrium eddy viscosity formulation. The model was formulated for near zero pressure gradient flows, and is not a good model for strong adverse pressure gradient flows as will be seen from the results. Some of the results from the Hytopoulos turbulence model [15]

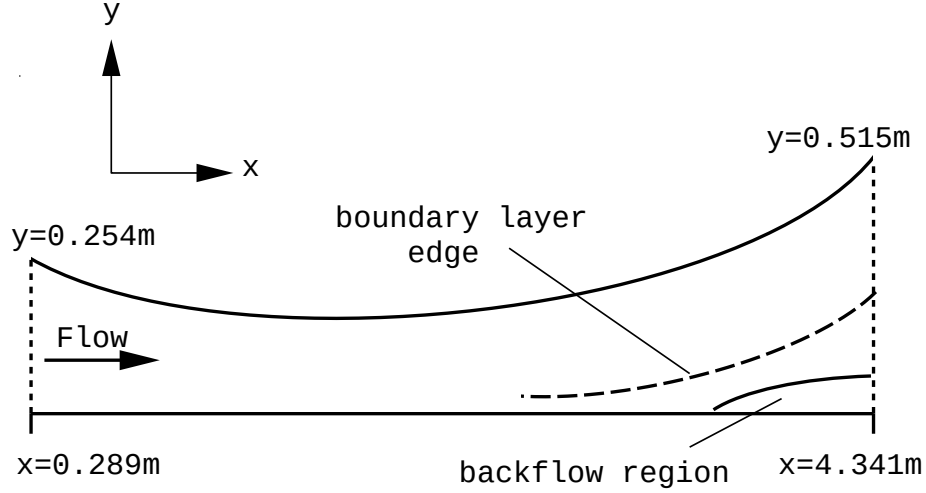


Figure 5.1: Sketch of diffuser computational domain.

will also be presented here. The Hytopoulos turbulence model, discussed in section 4.4.3 and given in Appendix B, also used Simpson's low-speed diffuser as a test case. In his work, Hytopoulos produced results for both the steady and unsteady cases, but did so using a boundary layer code. The code did not allow for calculations into the backflow region, so all results were limited to the attached flow regions. All data representing the Hytopoulos model was taken from Reference [15].

## 5.2 Steady Flow Case

For the steady diffuser case, a CFL of  $10^{12}$  was used. This value could only be implemented after the boundary layer and backflow region had been properly developed. This was due to the use of experimental data as boundary conditions. A lower time step was necessary to allow the boundary layer to adjust to the applied boundary conditions. The value of  $\beta$  was 1000 and the Reynolds number was set to match the experiment. For the steady flow, no inner iterations were performed and the pseudo

time step was set equal to the physical time step. Vertical line Gauss-Seidel was employed for this case with one sweep in each direction.

Results for the steady flow case are presented in Figures 5.2 through 5.17. In all the steady case results presented, the experimental data are presented with the baseline JKM and Baldwin-Lomax turbulence model. Along with these two models, the modifications presented in the previous chapter are also shown for the Johnson and King model. The modified JKM consists of the new ODE, equation (4.29), applied in the detached flow zone, the changes to the velocity scale parameter, and the inner eddy viscosity term given in equation (4.39). The normal stress model, (4.33), for the  $x$ -momentum equation was not included with the other modifications. It was found from implementing the model with the baseline JKM that no noticeable improvements occurred. The eddy viscosity formulation for the Reynolds shear stress dominated the turbulence modeling throughout the diffuser and the effects from the normal stress model were not seen.

Figure 5.2 displays the boundary layer thickness  $\delta$  along the diffuser. The baseline and modified JKM compare very well with the experimental thickness. The Baldwin-Lomax model overpredicts  $\delta$  in the separated region. Due to the thicker boundary layer prediction, the boundary layer edge velocity is lower than experiment as shown in Figure 5.3. The baseline and modified JKM had much better agreement with the experiment for  $U_e$ . The boundary layer displacement thickness is given in Figure 5.4, and includes results from the Hytopoulos model. Again, the agreement with experiment is good for both the JKM models. Computational data from the Hytopoulos model ends right near flow detachment, but tends to behave similar to the Baldwin-Lomax model.

The skin friction coefficient is shown in Figures 5.5 and 5.6. The first of the skin friction plots shows only the JKM models, the Baldwin-Lomax model, and experiment. The second plot shows additional data from not only Hytopoulos, but other computations presented at the 1980-81 AFSOR-HTTM-Stanford Conference on Complex Turbulent Flows. Looking at Figure 5.5, the Baldwin-Lomax model overpredicts the skin friction in the region leading up to separation and then quickly drops to zero, resulting in a very early prediction of flow detachment. The baseline Johnson

and King model predicted  $C_f$  well for most of the flow leading up to the strong adverse pressure gradient. Near detachment, the baseline model tended to underpredict  $C_f$  resulting in an early detachment location as well. The JKM modifications had a large impact on  $C_f$ . As the adverse pressure gradient increased, the skin friction values did not decrease as fast as the experiment showed. This effect on  $C_f$  came mainly from the inner region pressure gradient term and the use of  $u^*$  as the velocity scale. The modified JKM overpredicted the experimental skin friction for most of the region leading up to separation. The separation location occurred at 3.45 meters for the modified JKM and 3.30 meters for the baseline model. The addition of the pressure term in (4.39) had the largest impact on moving the detachment location downstream, making the modified JKM have the closest agreement with experiment for the separation location of 3.55 meters.

In Figure 5.6, additional skin friction data is given for the region leading up to flow detachment. The results from Pletcher *et al.* and Mellor *et al.* are taken from Reference [17]. The turbulence model used by Pletcher *et al.* was based upon a one-half equation model, where the velocity and length scales for the outer region are obtained from the TKE equation and an additional auxiliary equation. The turbulence model implemented by Mellor *et al.* used a five equation Reynolds stress model. The results show that flow detachment is not easily predicted, and only two of the models are close to predicting the separation location correctly, the modified JKM and the Hytopoulos model. Unlike the modified JKM, the model by Hytopoulos follows the experimental  $C_f$  prediction more closely, resulting in a good correlation with experiment.

A few of the numerical parameters used in modeling the turbulence are shown in Figure 5.7. The F factor is shown to be near unity until the adverse pressure gradient starts to become strong. During the strong adverse pressure gradient region, the ratio of total turbulence energy production to shear stress production begins to grow until it reaches a peak of 1.5. The computation reflects the fact that the shearing stress is not the only contributor to the turbulence energy production. Also shown in the plot are the variables used in modeling the velocity scale parameter: wall shear velocity and maximum Reynolds shear stress. The two variables are in close

agreement throughout the first part of the diffuser. As separation occurs,  $(-\overline{u'v'})$  becomes larger while  $u^*$  decreases toward zero.

The mean velocity profiles at four streamwise locations are presented in Figures 5.8-5.16. At each location, the boundary layer profiles are shown in both normal scale and semi-log scale. The profiles in Figures 5.8 to 5.10 are in the region leading up to detachment where the pressure gradients have become strong. In the experimental calculations, the normal stress terms at this location make up over 40% of the total production of energy. Figures 5.11 and 5.12 shows the mean velocity profile just before the detachment point while Figures 5.13 through 5.16 show velocity profiles inside the backflow region.

In all the boundary layer velocity profiles the Baldwin-Lomax model had the least favorable agreement with experiment. The velocities near the wall were always larger than experiment and a thicker boundary layer was predicted in the separation region. Figure 5.8 shows the performance of the Hytopoulos model in the region near flow detachment. The results are similar to the Baldwin-Lomax model in that it has a fuller profile. The Hytopoulos model is responding quicker to the adverse pressure gradient than the BLM model, but not as well as the Johnson and King models.

The inner region modification to the JKM improved the velocity profile in the region leading up to flow detachment. It was in this region of the flow that the best improvement over the baseline Johnson and King model was found. The semi-log plot of the velocity at  $x = 3.01$  meters in Figure 5.15 shows how strong the effect of the pressure gradient had on the inner region profile. The modified JKM follows the experimental data much more closely near the wall due to the added viscosity from equation (4.39). In the region after the detachment location, the baseline and modified models gave very similar profiles, both of which agreed well with the experimental data.

The impact of adding the normal stress contribution to the ODE can be seen in Figure 5.17. In this plot, the ratio of actual to equilibrium maximum Reynolds shear stress is given along the diffuser.

### 5.3 Unsteady Flow Case

The unsteady flow results are presented in Figures 5.18 through 5.26. The entrance flow is periodic and has a cycle time of 1.676 sec. The experimental data was divided into 96 ensembles or phases for each period. In the numerical computations a time step was taken equivalent to each phase, allowing the experimental data to be used directly for the inflow and outflow boundary conditions. In performing the time-accurate calculations, 70 sub-iterations were performed for each time step to bring the velocity divergence RMS norm down 6 orders of magnitude. This was necessary to enforce the velocity divergence condition at each step in physical time. The value of  $\Delta\tau$  was kept constant at  $10^{12}$ . The computational grid was the same for both the steady and unsteady cases.

To avoid flow separation on the upper wall during the actual unsteady experiment, air was blown through a jet positioned along the top wall. This was not modeled in the computations due to numerical difficulties and unknown conditions, and caused differences between the experimental and computational results. By studying the experimental data, it was learned that the additional flow from the jet caused changes in the mass flow rate. Therefore, at a given phase in the flow, the experimental flow rate would vary through the diffuser, while the computational mass flow rate would be constant. Figure 5.18 shows the mass flux at the inlet and exit planes, along with the computational flux for a single station inside the diffuser. The inlet and exit flow rates are different, revealing that the mass flow rate changes throughout the diffuser as a function of the phase angle. The computational mass flux at any location (only one arbitrary location is given in the figure) is the same as that for the inlet. Another factor that may have caused differences in the comparison is the upper wall geometry. The streamline on the upper wall surface may have changed throughout the cycle due to the jet, changing the geometry of the diffuser.

The boundary layer edge velocity for one complete period or cycle is presented in Figure 5.19 at an  $x$  location of 3.46 meters. The computations agree best during the acceleration phase of the cycle, and tends to under-predict the value of  $U_e$  at all other phases of the flow. This poor agreement with  $U_e$  during the deceleration phase of the

cycle is thought to be a result of the added flow from the jet. As the flow begins to decrease, the higher velocity from the jet requires a higher mass flow rate causing  $U_e$  to be larger.

Figures 5.20, 5.21, and 5.22 compare boundary layer velocity profiles at three different locations along the duct. At each location, six phases are shown to represent one complete periodic cycle. The computations seem to match the data best when the pressure gradient is changing the least, giving rise to slower changes in the profiles. During the acceleration phases, the modified model tends to improve the near wall profiles. This can be seen most clearly in Figure 5.20. The modified JKM responds to the pressure gradient by increasing the viscosity in the near wall region, moving the velocity profile closer to the experimental data. As the flow slows back down and the pressure gradient becomes adverse, the model does not respond as quickly, creating a lag in that portion of the cycle.

Both JKM models had similar profiles in the outer region of the boundary layer. The modifications to the outer region tended to have less impact on the unsteady flow case than the inner region changes. Of all the modifications made, the inclusion of the pressure gradient term in the inner eddy viscosity model had the largest impact on the unsteady flow calculation.

It was found that by removing the time dependency term in equation (4.11), no noticeable changes to the velocity profiles occurred. Therefore, for this case the differential equation could have been reduced to an ODE without any loss of accuracy for the time dependent problem.

The final four plots, Figures 5.23, 5.24, 5.25, and 5.26, show the hysteresis effect for two stations before separation and two stations inside the separated region. The plots show the shape factor  $H$  versus the edge velocity normalized by the phase averaged edge velocity. In these plots, the phase angle  $\omega t$  is indicated by solid squares and circles for  $180^\circ$  and  $360^\circ$  respectively. In Figure 5.23, the modified JKM has slightly more hysteresis than the baseline, but both models do not predict as strong a hysteresis as the experiment. The JKM models are also seen to have a slight phase lag compared to the experimental data. The phase lag becomes even more apparent at locations further downstream as seen in the remaining two figures. The best

agreement between experiment and computation is found in Figure 5.26. The correct magnitude of the hysteresis is seen in the computations, but the velocity profiles at this location still reflect the phase lag.

The only other known computations of the unsteady diffuser test case is by Hytopoulos. Hytopoulos performed calculations up to an  $x$  location of 2.75 meters, which avoids any flow separation. The Hytopoulos model predicted mean skin friction and velocity profiles that correlated very well with experiment. Results from the Hytopoulos model can be found in Reference [15].

## 5.4 Conclusions

The Johnson and King turbulence model was used to compare computational results with experimental data for a low-speed diffuser flow. In the steady flow case, improvements in the velocity profile were observed with the modified JKM in the strong adverse pressure gradient region. The improvements were due mainly to the inner region modifications. The separation location was better predicted from the modified JKM, but both the baseline and modified models still predicted an early detachment point. In the separation region, the baseline and modified JKM performed about the same, giving good velocity profile agreement with experiment. The model for the  $x$ -momentum normal stress term had no noticeable effect on the present diffuser test case.

In the unsteady diffuser case, the computational results did not agree as well with experimental data. The experiment had a blowing jet on the upper wall to keep the flow attached on that surface. This was not modeled in the computations which explained some of the differences in the data. The velocity profiles had the best agreement with experiment during the accelerating phases of the flow. The modifications to the JKM were seen to have the largest impact in the inner region. The sensitivity of the flow velocity near the wall increased as a result of the modifications made to the model.

The JKM had a time dependent term in the differential equation which was also studied for this flow. The time dependency was removed, reducing the differential

equation down to an ODE. When this was performed, no changes in the mean velocity profiles were observed, indicating that the time dependency in the model was not very strong.

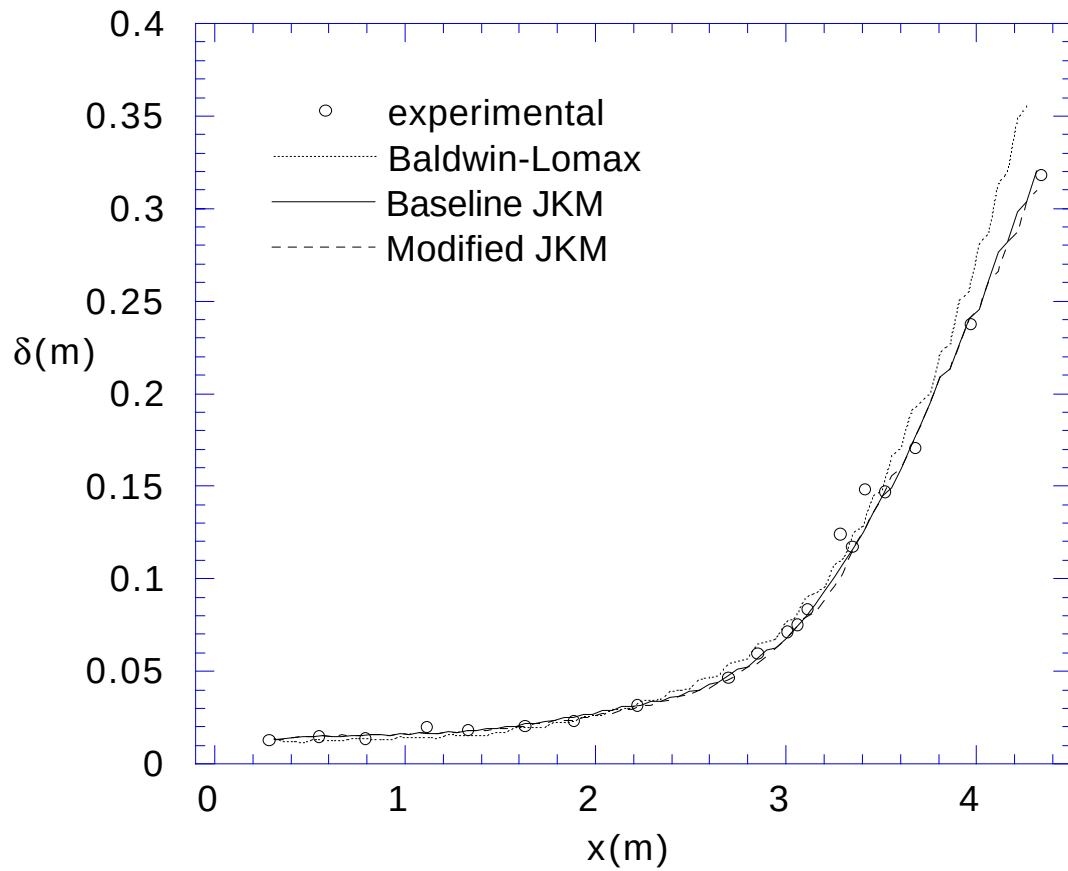


Figure 5.2: Boundary layer thickness for steady diffuser case.

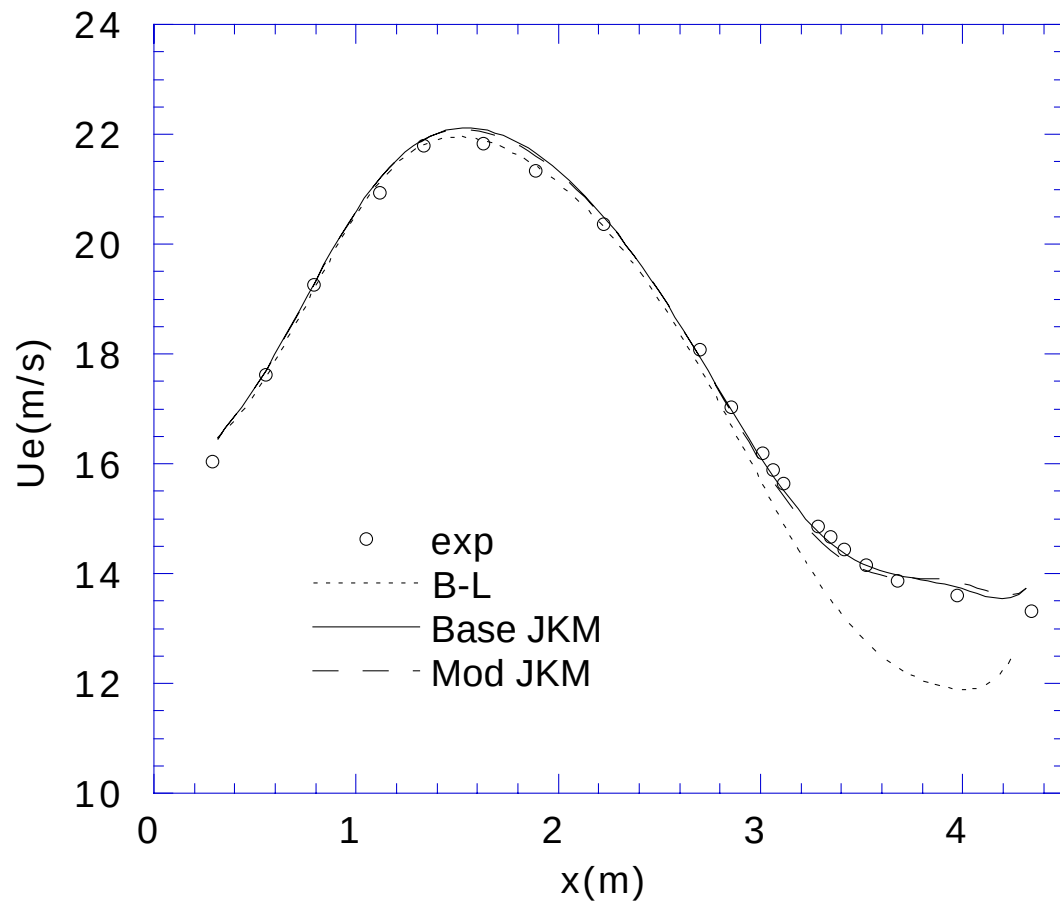


Figure 5.3: Boundary layer edge velocity for steady diffuser case.

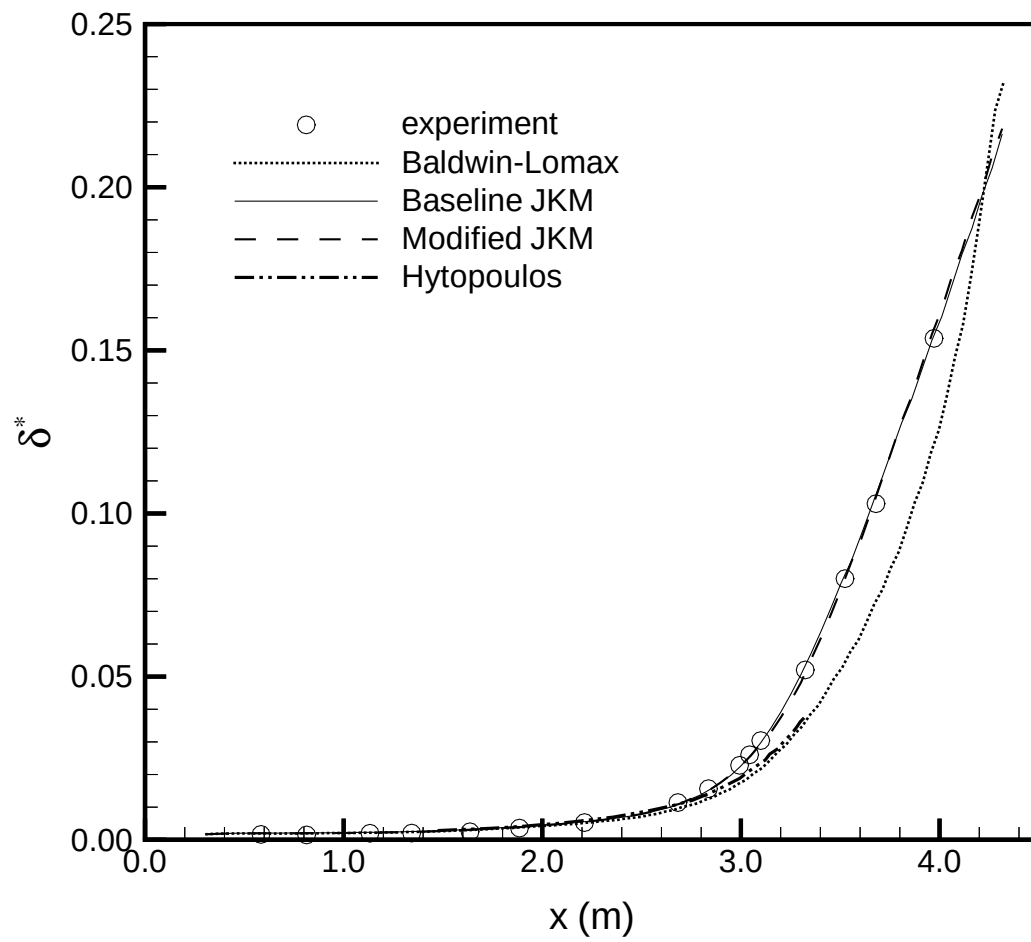


Figure 5.4: Boundary layer displacement thickness for steady diffuser case.

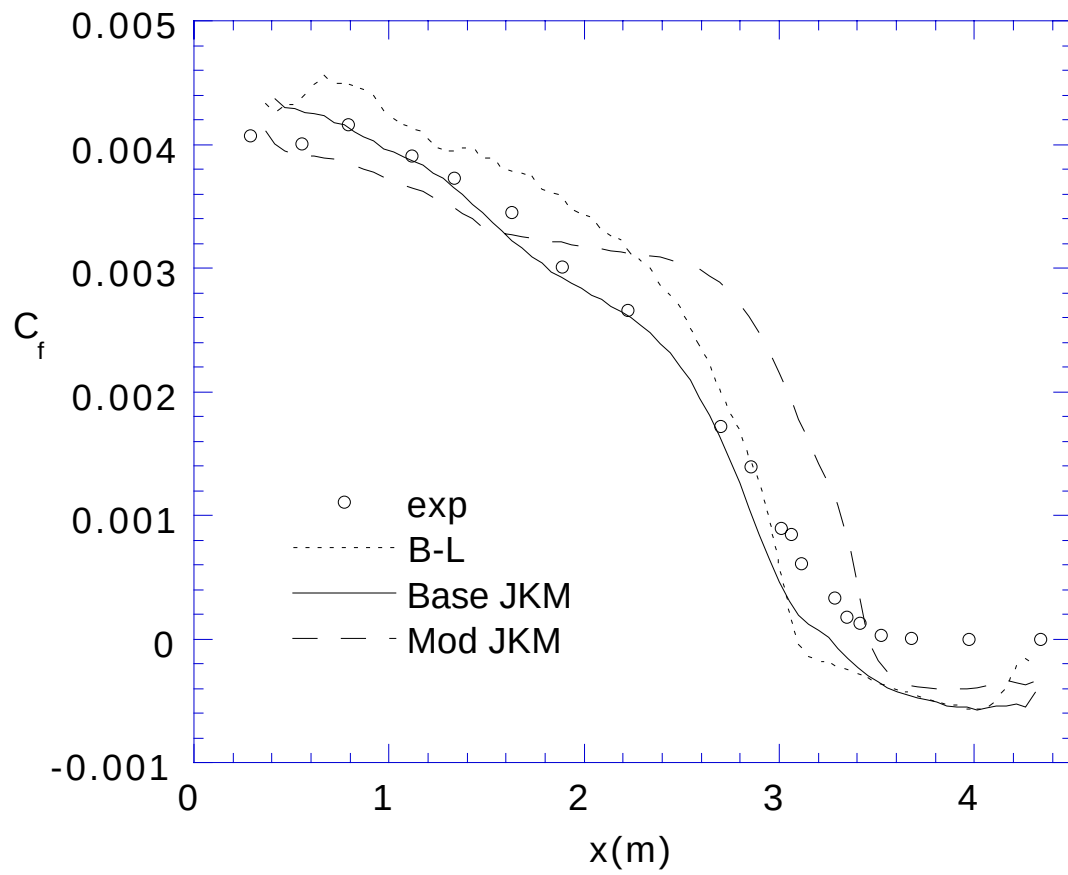


Figure 5.5: Friction coefficient for steady diffuser case.

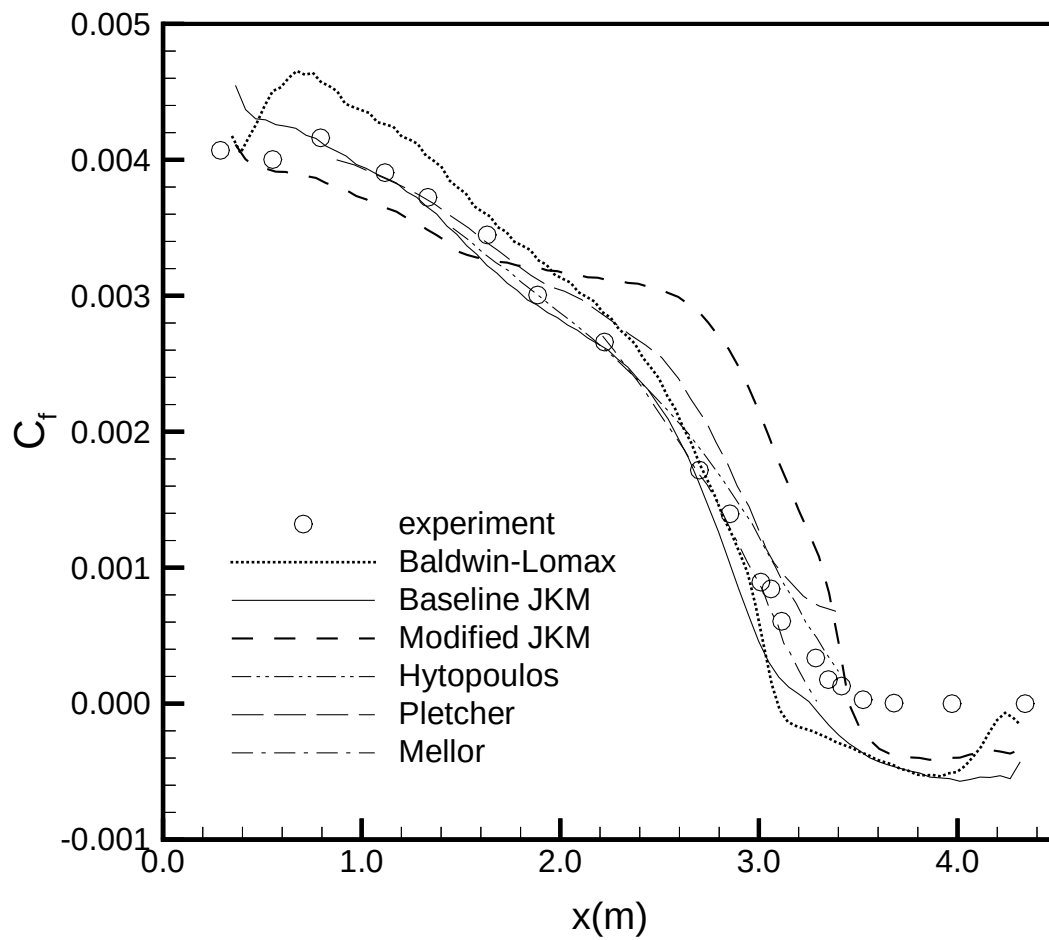


Figure 5.6: Friction coefficient for steady diffuser case. Comparison with other models.

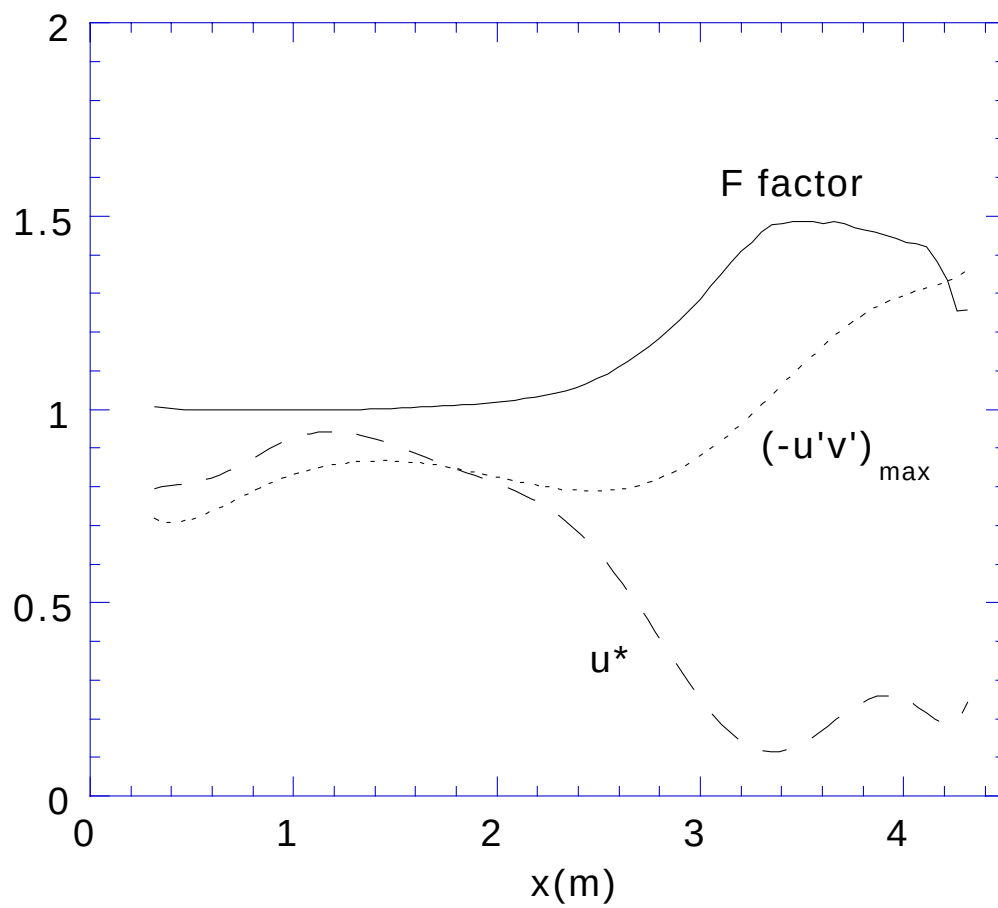


Figure 5.7:  $F$  factor, maximum Reynolds shear stress, and wall shear velocity along diffuser for the steady case.

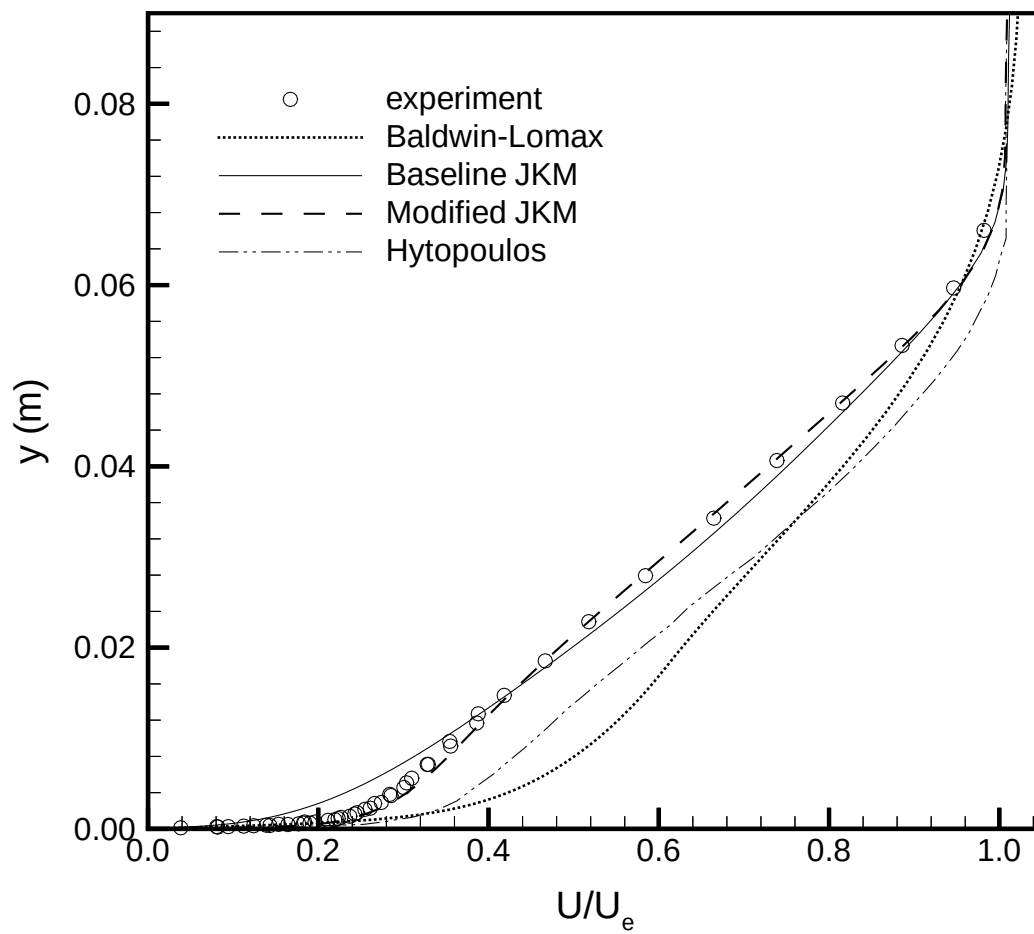


Figure 5.8: Comparison of the steady diffuser boundary layer profile at 3.01 meters with several other models.

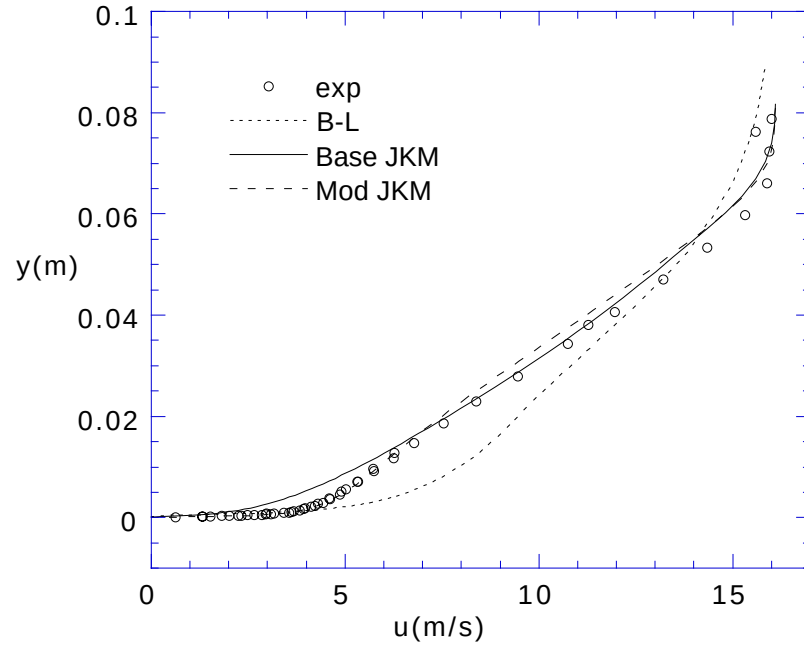


Figure 5.9: Comparison of the steady diffuser boundary layer profile at 3.01 meters.

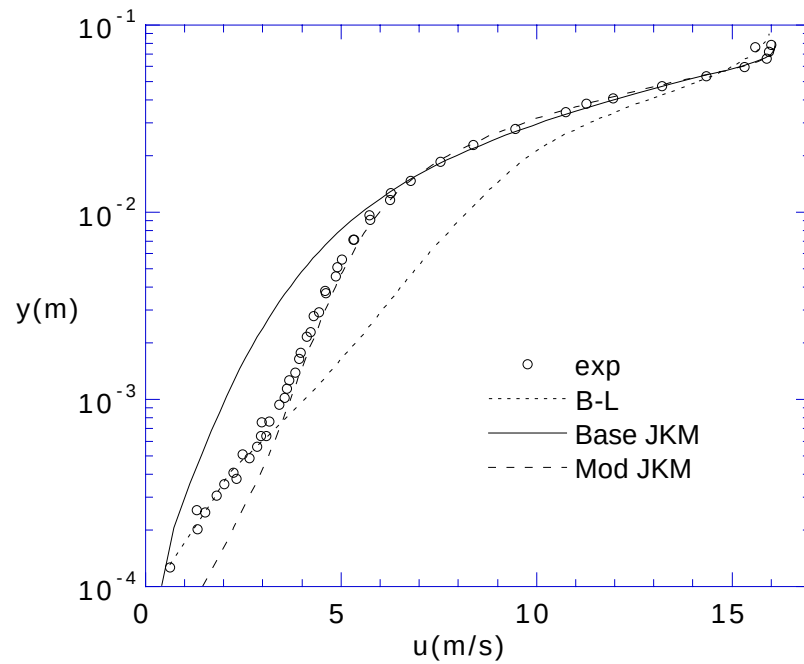


Figure 5.10: Semi-log scale comparison of the steady diffuser boundary layer profile at 3.01 meters.

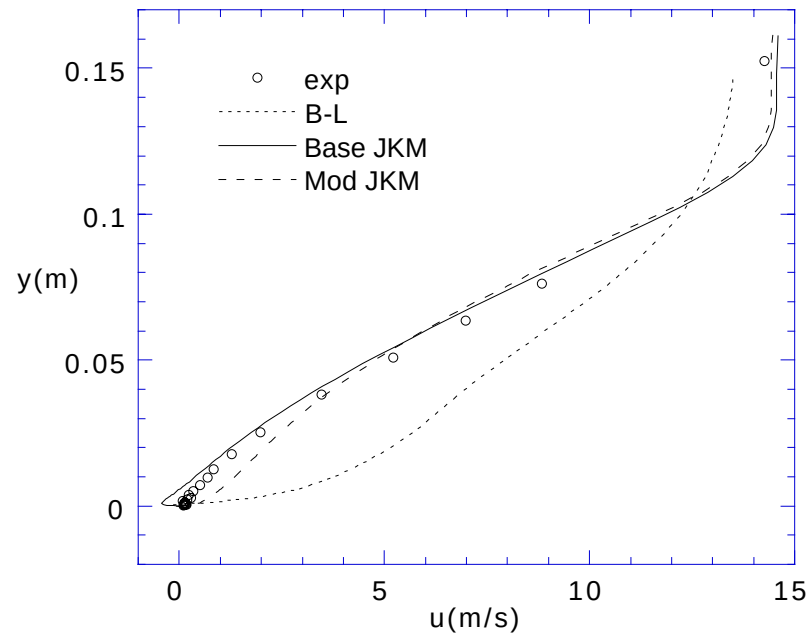


Figure 5.11: Comparisons of the steady diffuser boundary layer profile at 3.42 meters.

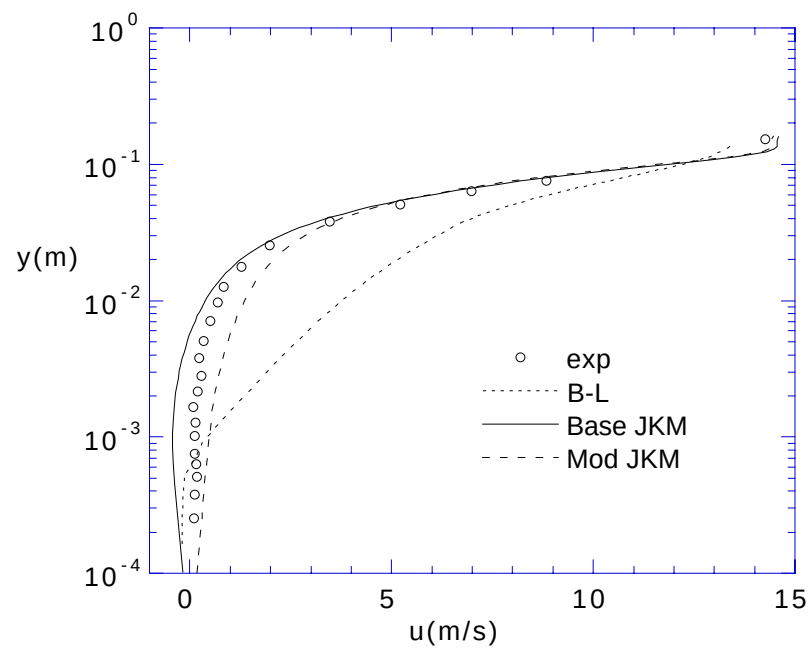


Figure 5.12: Semi-log scale comparison of the steady diffuser boundary layer profile at 3.42 meters

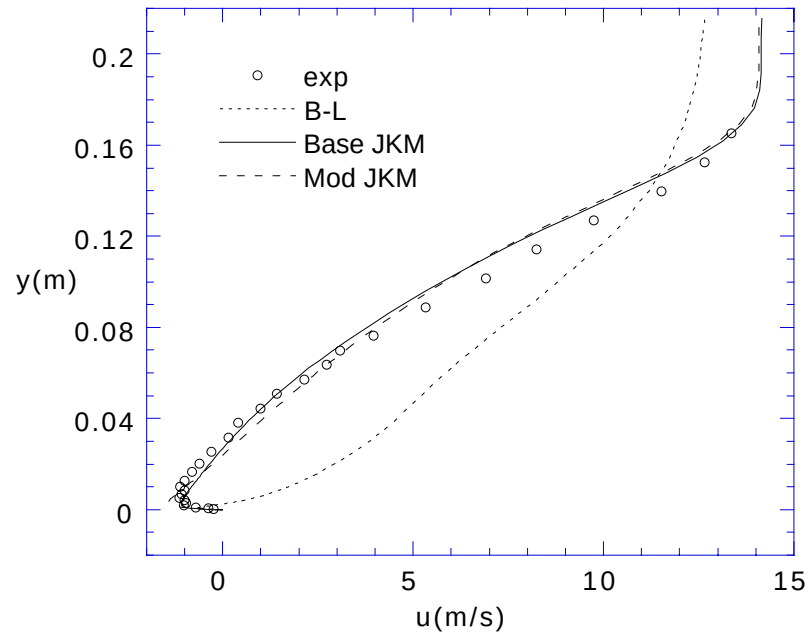


Figure 5.13: Comparisons of the steady diffuser boundary layer profile at 3.68 meters.

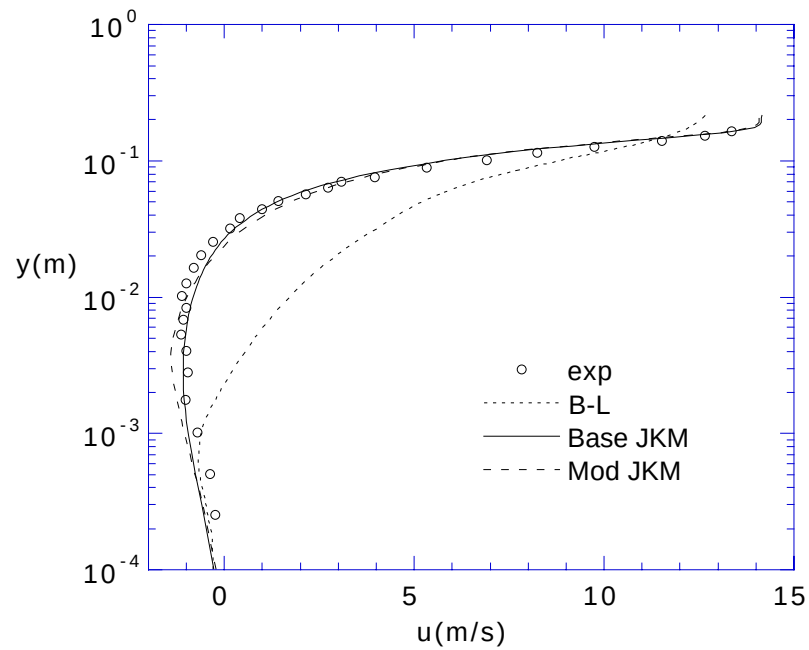


Figure 5.14: Semi-log scale comparison of the steady diffuser boundary layer profile at 3.68 meters.

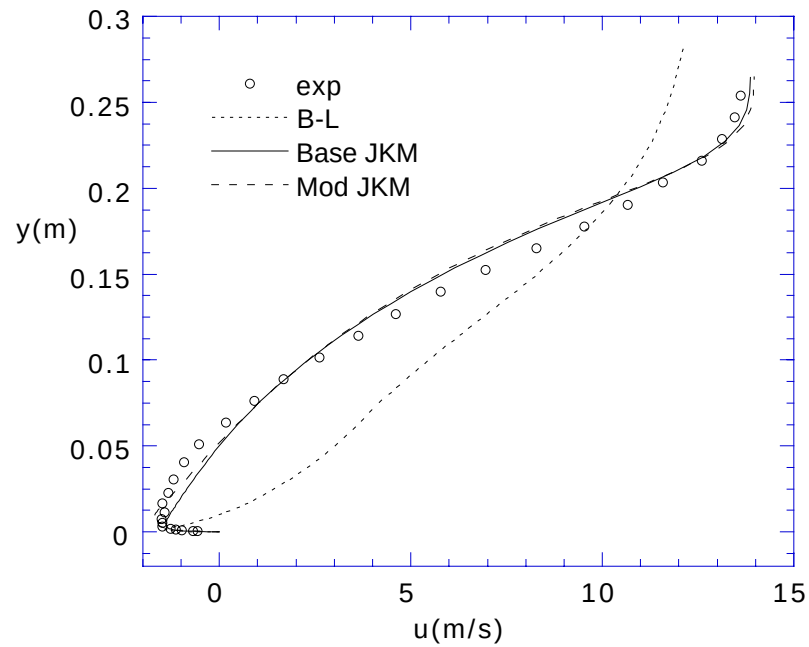


Figure 5.15: Comparisons of the steady diffuser boundary layer profile at 3.97 meters.

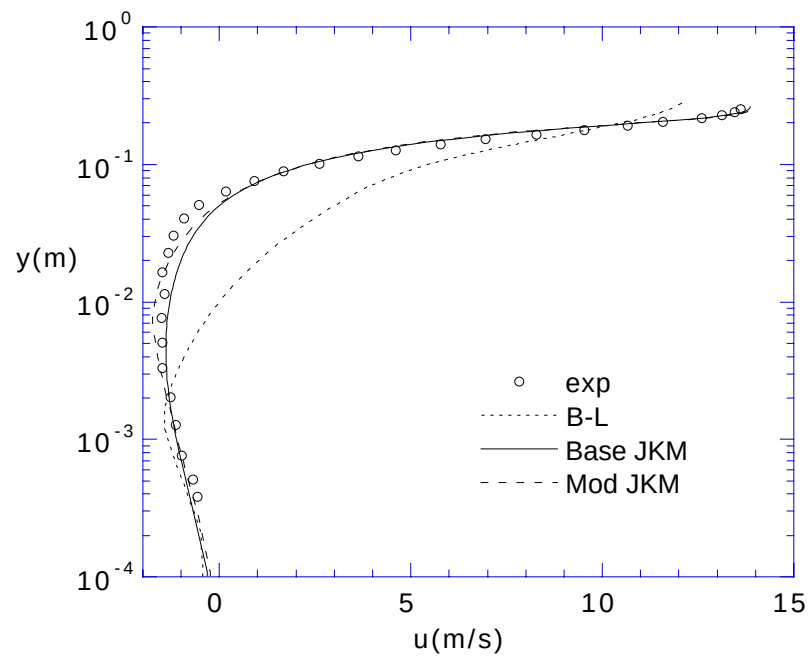


Figure 5.16: Semi-log scale comparison of the steady diffuser boundary layer profile at 3.97 meters.

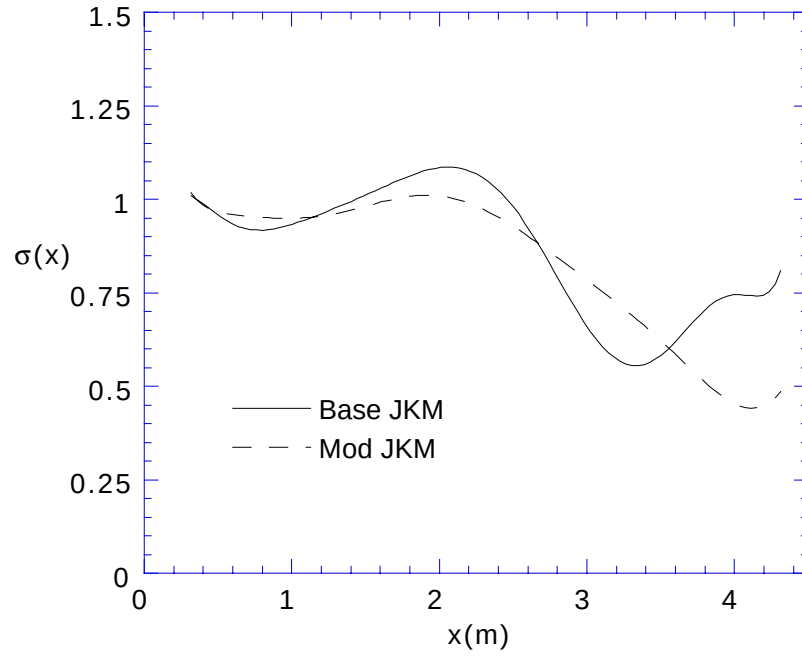


Figure 5.17: Values of  $\sigma(x)$  for the steady diffuser flow used in scaling the outer eddy viscosity.

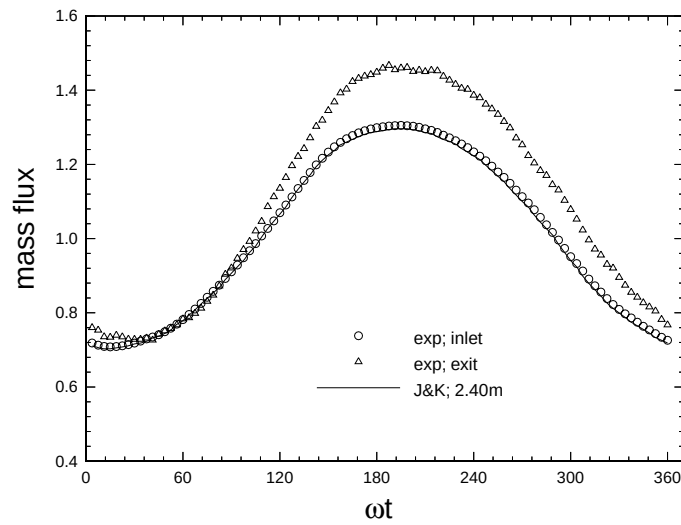


Figure 5.18: Comparison of experimental mass flux at inlet and exit planes for the unsteady diffuser case. Flux is nondimensionalized by inlet area and mean inlet velocity.

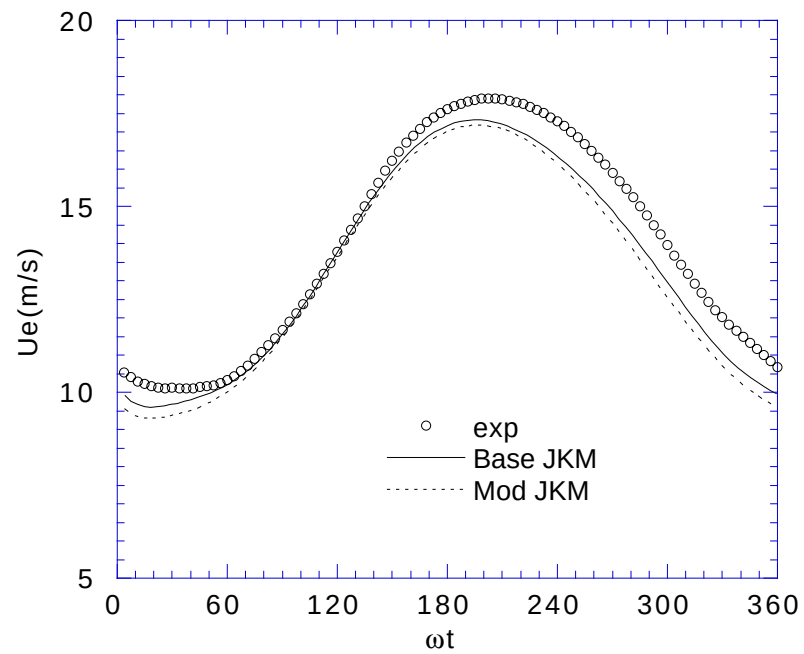


Figure 5.19: Boundary layer edge velocity for one complete cycle at 3.46 meters.

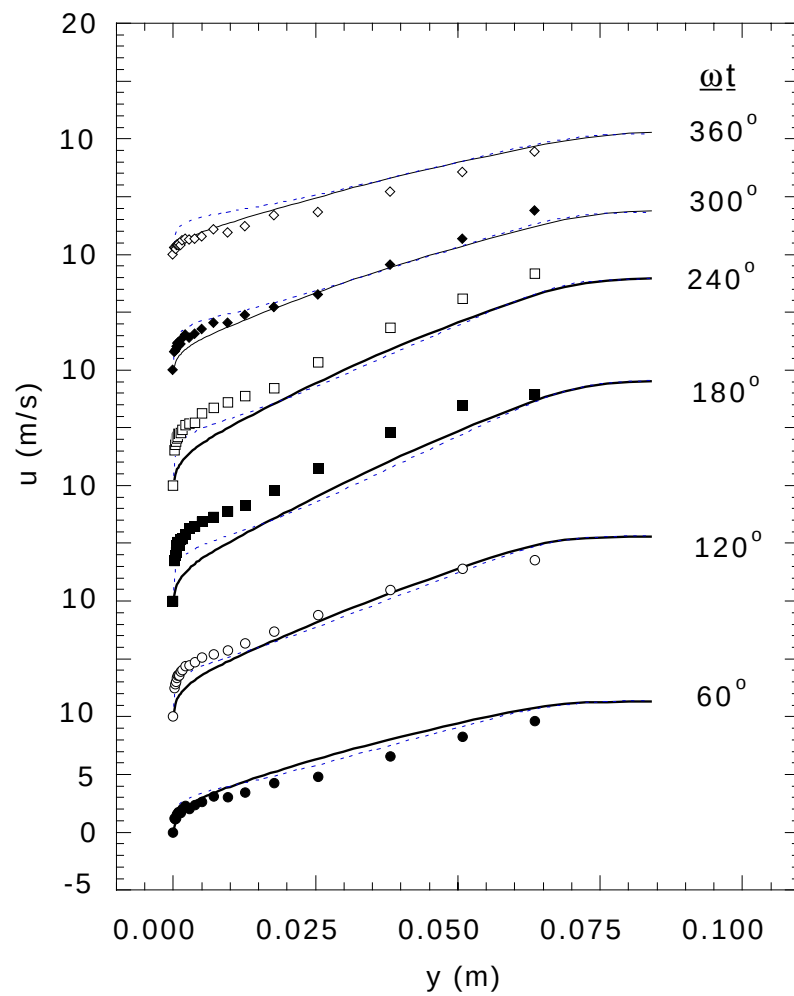


Figure 5.20: Unsteady velocity profiles at 3.05 meters for six phases in the cycle. Note the displaced ordinates. Legend is same as Figure 5.19

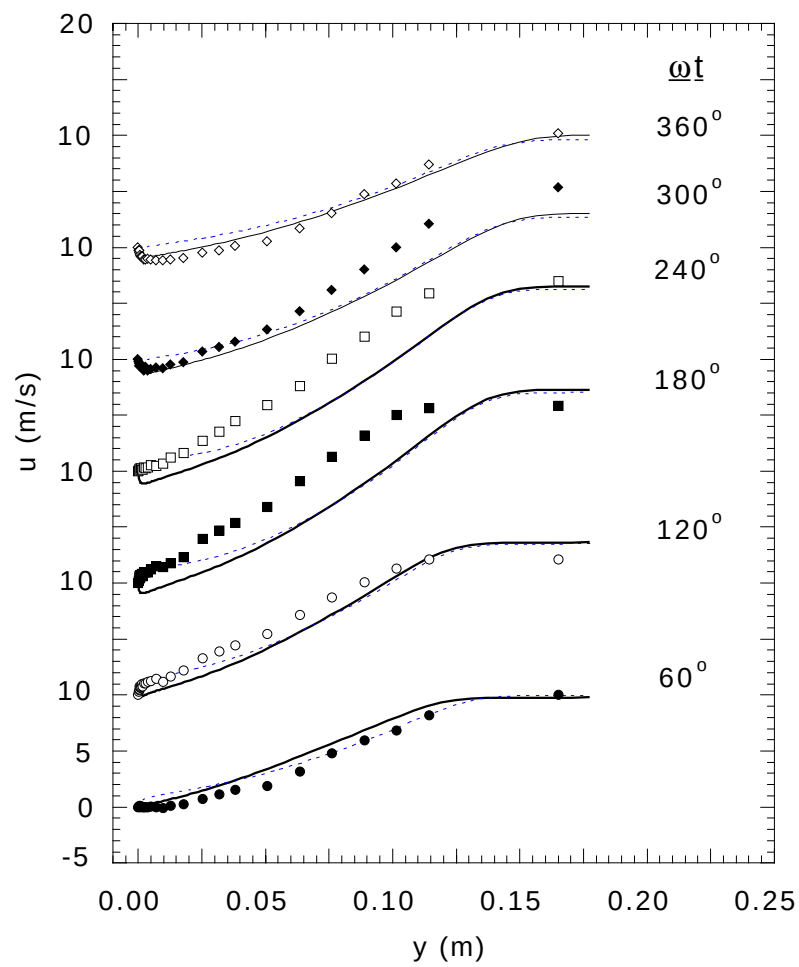


Figure 5.21: Unsteady velocity profiles at 3.55 meters. Note the displaced ordinates. Legend is same as Figure 5.19

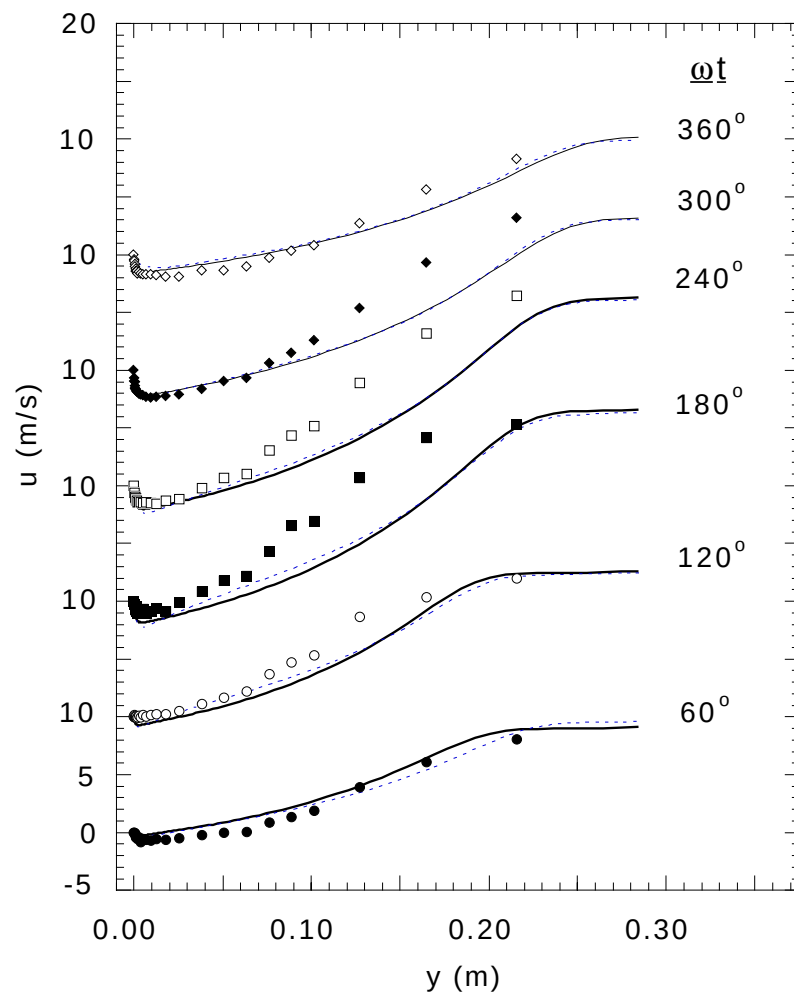


Figure 5.22: Unsteady velocity profiles at 3.96 meters. Note the displaced ordinates. Legend is same as Figure 5.19

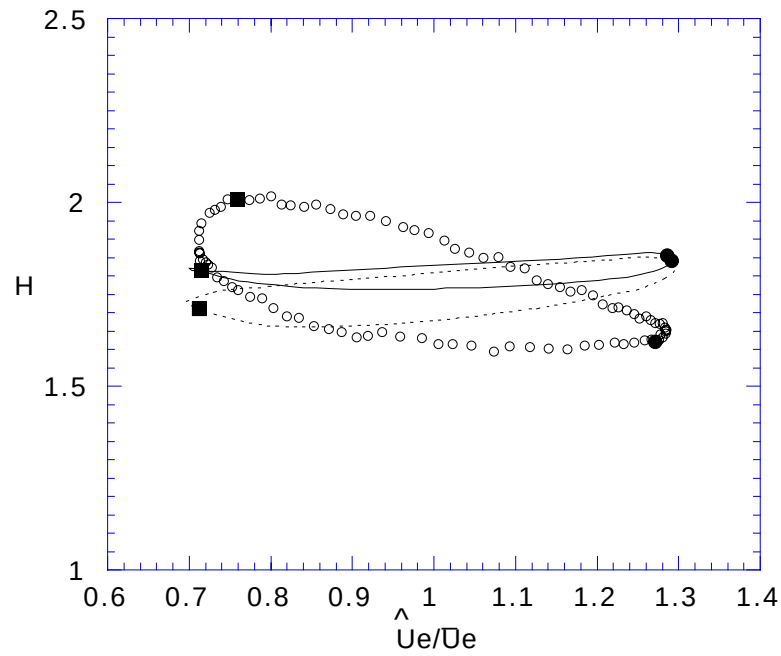


Figure 5.23: Shape factor showing hysteresis effects at 2.85 meter. Legend is same as Figure 5.19

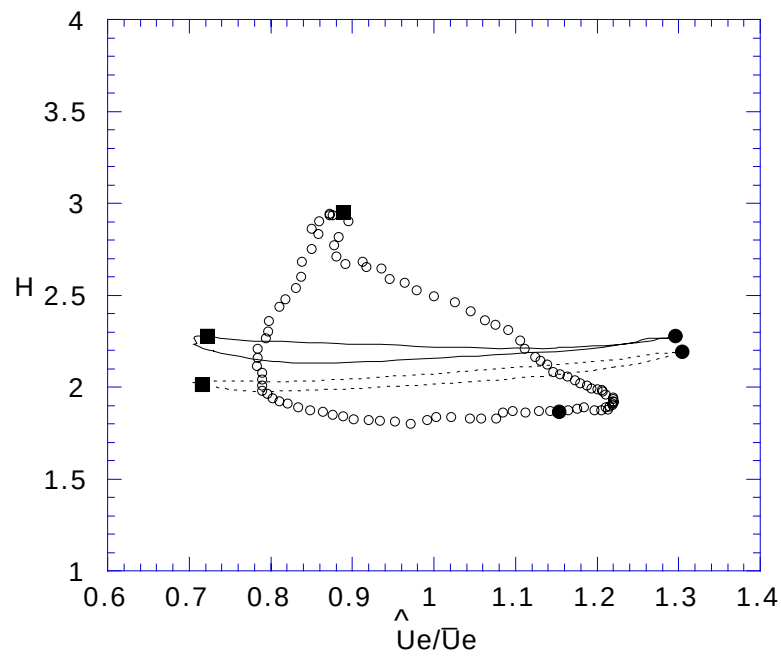


Figure 5.24: Shape factor showing hysteresis effects at 3.06 meters.

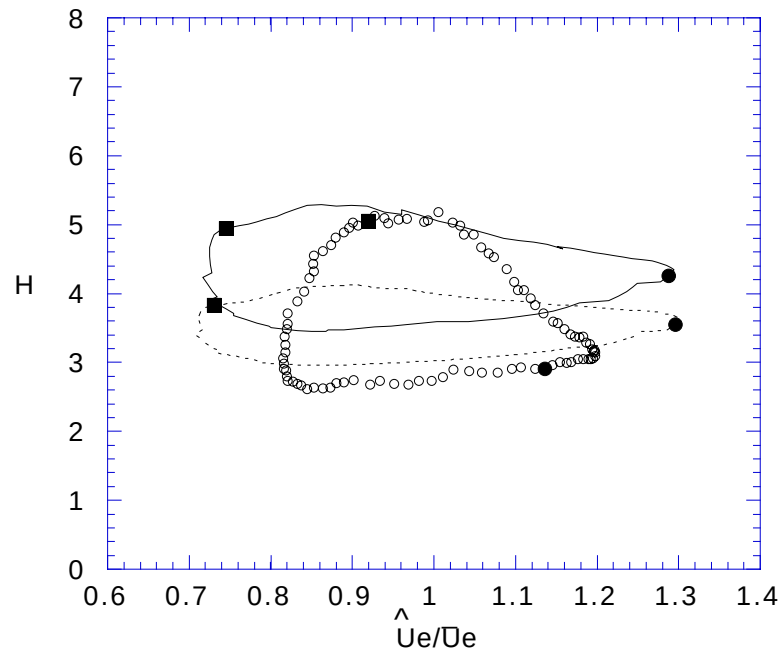


Figure 5.25: Shape factor showing hysteresis effects at 3.51 meters.

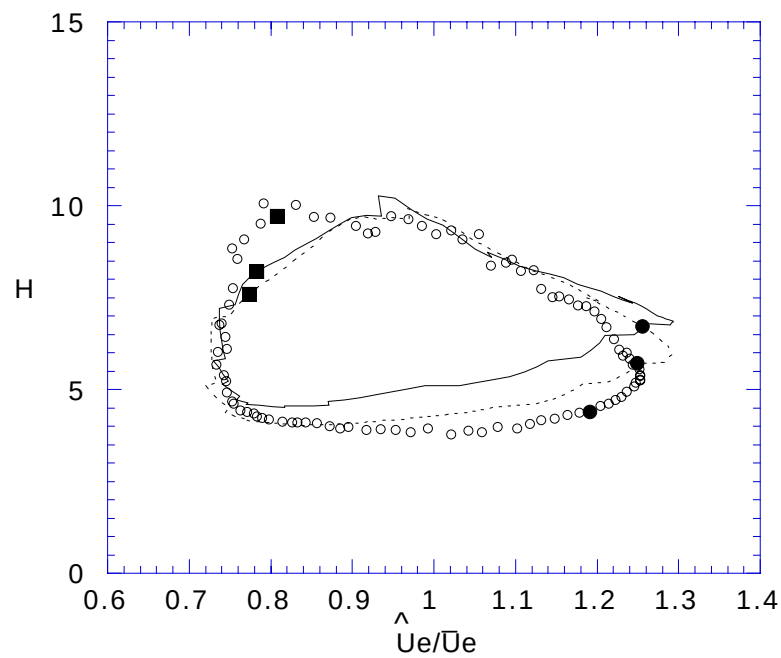


Figure 5.26: Shape factor showing hysteresis effects at 3.97 meters.

# Chapter 6

## Flapping Foil Results

### 6.1 Description of Flow

This chapter discusses computational results for an experiment performed at the Massachusetts Institute of Technology (MIT). The MIT experiment is known as the flapping foil experiment (FFX). The first computational results of this experiment were presented at the March 1993 Office of Naval Research (ONR)/MIT Unsteady-Flow Workshop. At this meeting, computational results were submitted with only the experimental conditions and boundary data known beforehand. Since then, the experimental data has been released and additional computations have been performed (see references [75], [76], [77], [78], and [79]).

This section gives a brief description of the experiment. The following section will explain how the computational flow domain was set up along with the grids that were used for the computation. In the remaining sections, numerical and experimental data will be compared for cases of steady and unsteady flow. For more details on the experiment, see References [80],[81], and [82].

FFX was performed to simulate blade loading under oscillatory conditions. The unsteady flow field acting upon the blade is modeled in the experiment as a vertical or transverse sinusoidal gust. This unsteady loading on a propeller blade is experienced on ships as the blade rotates through the wake created by the hull or appendage.

The experiment took place at the MIT Variable Pressure Water Tunnel. FFX

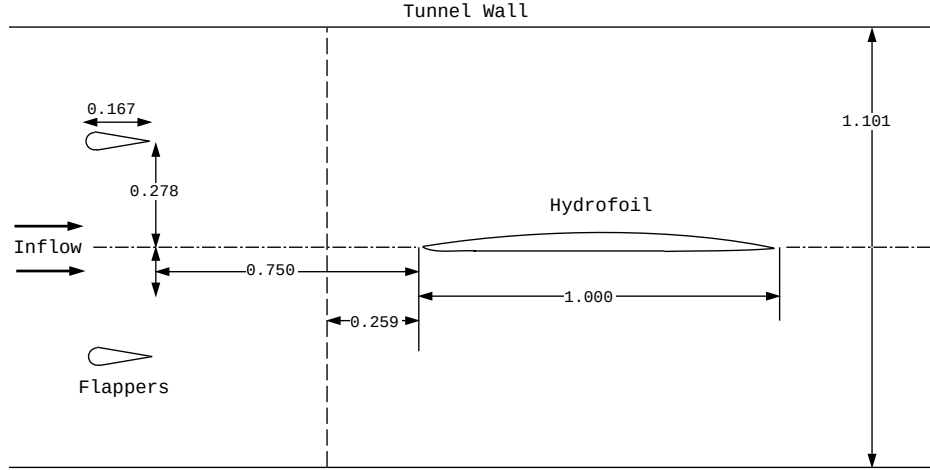


Figure 6.1: Diagram of the flapping foil experiment.

was designed to produce a two-dimensional flow around a stationary hydrofoil. The hydrofoil was mounted in the test section along the centerline. A diagram of the experimental setup is shown in Figure 6.1, where all measurements have been nondimensionalized by the chord length. The vertical gusts imposed on the stationary hydrofoil were generated by two smaller hydrofoils known as the “flappers”. These were located upstream of the stationary foil, symmetrically offset from the centerline. The flappers would undergo an oscillatory motion in which both flappers would rotate in phase. The sinusoidal gust loading imposed on the stationary foil was created by the vortex sheets being shed from the flappers.

The stationary foil had a NACA 16 thickness form with maximum thickness of 8.84%, a NACA  $a=0.8$  mean line with maximum camber of 2.576%, and a beveled anti-singing trailing edge modification. The coordinates of the hydrofoil as measured from the experiment are given in Table 3. The trailing edge of the test hydrofoil was offset below the tunnel centerline. The angle of attack was measurable to within 0.1 degrees and was approximated to be between 1.15 and 1.25 degrees. Boundary layer trips were located at  $x/c = 0.105$  on both the upper and lower surfaces. The transition location was forced using 0.050 inch diameter epoxy disks 0.008 inches high, and 0.050 inches in separation.

The flappers were NACA 0025 hydrofoils. The chord length of the flappers were

1/6 the chord of the stationary hydrofoil. One cycle or period of the gust consisted of the flappers moving from  $0^\circ$ , to  $-6^\circ$ , to  $0^\circ$ , to  $+6^\circ$ , and back to  $0^\circ$  angle of attack. The flappers rotated about their midchord. The oscillations were done at a reduced frequency of 3.6. In physical time, this translates into 0.0625 seconds per cycle.

All velocity profiles measured in the FFX were done using a laser Doppler velocimeter (LDV). Boundary layer profiles were taken normal to the hydrofoil surface where the LDV was used to measure the tangential velocity component. All data was normalized with respect to the chord length of the stationary foil and to the nominal freestream velocity. Static and total pressures were taken upstream and downstream of the hydrofoil using a Pitot-static probe.

The experiment was performed at steady and unsteady flow conditions. The flow parameters are given in Table 4. In the steady case, the flappers were locked into position at zero angle of attack. Boundary layer measurements were taken at eight locations on the suction side of the hydrofoil and at seven locations on the pressure side. For the unsteady case, velocity data was obtained by phase-averaging the bursts of data from the LDV. One harmonic cycle or period was divided into 180 parts or phases. A Fourier analysis was done on the unsteady data to give amplitude and phase information for the time domain. Profiles at six locations were recorded on the suction side and two on the pressure side in the unsteady case.

## 6.2 Computational Aspects

The experimental data and set-up allow for three different computational approaches when performing FFX calculations. The first approach is to include the complete domain in the computation. This involves grids for both the hydrofoil and the flappers. The advantage of this approach is that very little experimental data is required in performing the calculation. On the other hand, the complexity of the problem is increased since the oscillatory motion of the flappers must be simulated. The next approach is to eliminate the flappers and use the experimental data as the inflow condition for the hydrofoil. This results in a much simpler computational domain, but relies more heavily on the experimental data for the inflow conditions. The last

Table 3: Stationary Hydrofoil Surface Coordinates.

| Upper Surface |        | Lower Surface |         |
|---------------|--------|---------------|---------|
| x/c           | z/c    | x/c           | z/c     |
| 0.0000        | 0.0000 | 0.0000        | 0.0000  |
| 0.0006        | 0.0028 | 0.0013        | -0.0026 |
| 0.0033        | 0.0061 | 0.0049        | -0.0050 |
| 0.0085        | 0.0097 | 0.0109        | -0.0071 |
| 0.0165        | 0.0138 | 0.0196        | -0.0087 |
| 0.0278        | 0.0183 | 0.0312        | -0.0101 |
| 0.0427        | 0.0231 | 0.0463        | -0.0115 |
| 0.0618        | 0.0281 | 0.0656        | -0.0128 |
| 0.0854        | 0.0334 | 0.0893        | -0.0139 |
| 0.1139        | 0.0388 | 0.1178        | -0.0149 |
| 0.1473        | 0.0443 | 0.1511        | -0.0158 |
| 0.1849        | 0.0496 | 0.1886        | -0.0166 |
| 0.2258        | 0.0546 | 0.2292        | -0.0173 |
| 0.2691        | 0.0591 | 0.2722        | -0.0180 |
| 0.3141        | 0.0628 | 0.3168        | -0.0184 |
| 0.3601        | 0.0658 | 0.3622        | -0.0188 |
| 0.4063        | 0.0679 | 0.4079        | -0.0189 |
| 0.4524        | 0.0693 | 0.4535        | -0.0188 |
| 0.4977        | 0.0699 | 0.4983        | -0.0186 |
| 0.5418        | 0.0697 | 0.5418        | -0.0182 |
| 0.5842        | 0.0689 | 0.5837        | -0.0177 |
| 0.6248        | 0.0674 | 0.6236        | -0.0170 |
| 0.6634        | 0.0653 | 0.6617        | -0.0162 |
| 0.7001        | 0.0626 | 0.6978        | -0.0153 |
| 0.7348        | 0.0592 | 0.7319        | -0.0144 |
| 0.7674        | 0.0553 | 0.7640        | -0.0134 |
| 0.7979        | 0.0510 | 0.7941        | -0.0124 |
| 0.8258        | 0.0464 | 0.8219        | -0.0114 |
| 0.8509        | 0.0419 | 0.8471        | -0.0105 |
| 0.8730        | 0.0376 | 0.8694        | -0.0096 |
| 0.8918        | 0.0337 | 0.8884        | -0.0087 |
| 0.9073        | 0.0303 | 0.9042        | -0.0079 |
| 0.9204        | 0.0273 | 0.9174        | -0.0071 |
| 0.9317        | 0.0245 | 0.9287        | -0.0064 |
| 0.9420        | 0.0216 | 0.9390        | -0.0058 |
| 0.9618        | 0.0152 | 0.9594        | -0.0042 |
| 0.9718        | 0.0117 | 0.9698        | -0.0033 |
| 0.9816        | 0.0079 | 0.9802        | -0.0023 |
| 0.9911        | 0.0040 | 0.9903        | -0.0012 |
| 1.0000        | 0.0000 | 1.0000        | 0.0000  |

Table 4: Experimental flow conditions.

|                                        |                    |
|----------------------------------------|--------------------|
| nominal freestream velocity, $U_{inf}$ | 20.62 <i>fps</i>   |
| chord length, $c$                      | 1.5 <i>ft</i>      |
| water temperature                      | 25.6°C             |
| Reynolds number, $R_e$                 | $3.8 \times 10^6$  |
| reduced frequency, $k$                 | 3.6                |
| angle of attack                        | $\sim 1.2$ degrees |

approach shrinks the computational domain even further by eliminating the tunnel walls. For this case, the domain is simply a box that incloses the hydrofoil. Experimental data was recorded on such a “bounding box” which provided data for all the outer computational boundaries. In all three approaches, the no-slip condition is used for the surface of the hydrofoil.

The first computational results presented at the MIT/ONR FFX workshop included calculations from all three approaches. The most common approach performed was to use the bounding box data, which excluded the tunnel walls and flappers from the computational domain. No conclusions were made from the workshop about which approach was best. Since then, a number of other researchers have performed the numerical calculation. Paterson and Stern [75] performed the FFX computation using all three approaches. When including the flappers, a chimera overlaid-grid scheme was used with a pressure based Navier-Stokes flow solver. To model the turbulence, the Baldwin-Lomax model was employed. For both the steady and unsteady cases they found similar agreement between the three approaches. The boundary conditions, CPU time, and storage requirements were found to be the major differences. Their results did not conclude that using the complete domain was the best approach. The work by Ho and Lakshminarayana [79] did not include the upstream flappers, but used the experimental data for the inflow conditions. The turbulence model used was the  $k - \epsilon$  two-equation model by Chien [83].

Several other computations were done that used only the complete FFX configuration as the computational domain. Taylor *et al.* [76] modeled the unsteady flappers

using a multi-block method. The algorithm consisted of the artificial compressibility formulation and Baldwin-Lomax turbulence model. Both the chimera overlapped and multi-block patched grid schemes were done by Kiris *et al.* [77]. Again the artificial compressibility formulation was used, but with the Baldwin-Barth one equation model. And finally, Yang *et al.* [78] used a multi-domain, pressure based N-S flow solver with the  $k - \epsilon$  turbulence model.

The approach taken here will be to use the experimental data for the inflow boundary conditions. This means that the hydrofoil and tunnel walls will be included in the computational domain, but the flappers will not. The upstream location of the experimental data in both the steady and unsteady computations is indicated by a dashed line in Figure 6.1. This line of data is 0.259 chord lengths in front of the foil leading edge. This fixes the inflow boundary location for the numerical simulation. The computational domain for the steady case could be extended further upstream, putting more distance between the inflow boundary and the leading edge of the foil. This was in fact tested. The inflow boundary was made one chord length away from the foil and a constant inflow velocity, the nominal freestream velocity, was used instead of the experimental data. To verify this modeling approach, the inflow experimental data was compared to the computations. The agreement between the data was good. Since this approach required a different grid than the unsteady case, this approach was not taken. Instead, the grids for both the steady and unsteady cases were made identical. The exit boundary location for both cases is located one chord length downstream from the foil trailing edge.

A grid convergence study was done in order to determine the appropriate grid dimensions. The steady flow case using the baseline Johnson and King turbulence model was used to compare computational data on different grids. A total of four different O-grids were tested. The grids had dimensions  $(101 \times 81)$ ,  $(121 \times 101)$ ,  $(121 \times 121)$ , and  $(141 \times 101)$ . The first grid dimension describes the number of points that wrap around the hydrofoil surface, while the second dimension is the number of grid points normal to the foil surface. Plots of the boundary layer edge velocity and displacement thickness is given in Figures 6.2 and 6.3. Both  $U_e$  and  $\delta^*$  are boundary layer parameters used in the turbulence models. Because of this, it is important

to check that these parameters are calculated correctly. All four grids give almost identical results for the two parameters. From these two plots, it appears that 101 grid points would provide an adequate number of cells to describe the hydrofoil surface. Next the boundary layer profile at  $x/c = 0.972$  was compared. This is shown in Figure 6.4. Again all of the grids compare well against one another. The grid with 81 points normal to the surface had the least number of cells describing the boundary layer. As a result, a slight decrease in the profile resolution is seen in the outer portion of the boundary layer. In each of the grids, the spacing next to the solid surface was chosen such that on average  $y^+ \approx 1$ .

For the computational results that follow, the grid chosen for both the steady and unsteady cases has dimensions  $(121 \times 101)$ . The grid is shown in Figure 6.5. In order to provide more cells at the inflow boundary for interpolating the experimental inflow data, extra grid points were chosen to describe the hydrofoil surface, giving 121 points on the hydrofoil. The experimental data for the inflow boundary is given at fifteen locations, six behind each of the flappers and 3 directly in front of the hydrofoil. For the grid chosen, there are anywhere from 45 to 65 grid points inside the boundary layer depending on the location around the hydrofoil.

### 6.3 Steady Flow Case

The steady flow case was computed using the Baldwin-Lomax model, the baseline Johnson and King model, and the modified Johnson and King model. All cases were computed using  $\beta = 1000$  and a CFL of 30. For each iteration, two sweeps were done around the airfoil, one in the clockwise direction followed by a sweep in the opposite direction. The compressibility parameter  $\beta$  was varied from 500 to 5000 to see if any changes in the solution occurred, but none were observed. For the Johnson and King models,  $\sigma$  was allowed to start varying after the transition location. The lower (or pressure) surface was computed using equilibrium conditions.

For the inflow boundary condition, the  $u$  and  $v$  velocities were set using the experimental data and the pressure was extracted from the interior cells of the computational domain. The tunnel walls were treated as an inviscid streamline and on the

hydrofoil surface, the no-slip condition was imposed. At the downstream location, the outflow boundary condition was imposed where  $u$  and  $v$  was computed using the interior values and pressure was specified.

The first data comparison for the steady case is  $C_p$ , shown in Figure 6.6. Both the modified and baseline JKM had the best agreement on the suction and pressure sides of the hydrofoil. The difference between the two models for  $C_p$  is very small. The Baldwin-Lomax model predicted a smaller  $C_p$  along the upper foil surface, resulting in a larger boundary layer edge velocity than the other models and experiment. There are only four experimental data points on each surface, making it difficult to judge the accuracy of the computational results, especially near the trailing edge where the flow undergoes slight separation.

The coefficient of friction,  $C_f$ , is plotted in Figure 6.7 for the upper surface. The Johnson and King models have noticeable disagreement only near the trailing edge. The baseline model begins to have lower  $C_f$  values around  $x/c = 0.7$  and predicts a flow detachment at  $x/c = 0.975$ . The modified JKM never predicts separation, while the B-L model also predicts detachment, but at  $x/c = 0.967$ . The experiment shows that some backflow starts to occur between  $x/c = 0.972$  and  $0.990$ . Even though the B-L model has a good prediction of the separation location, it seems to overpredict the value of  $C_f$  along most of the upper surface. The inability of the modified model to predict separation is due to the pressure gradient modification. This will become more clear by looking at the velocity profiles near the trailing edge.

A plot of  $\sigma$  for the upper surface is given in Figure 6.8. The modified model does not predict as strong of a nonequilibrium region near the trailing edge as the baseline model. The other difference between the two models is observed near the mid-chord where the the modified model predicts values of  $\sigma$  slightly less than unity while the baseline model remains at or above one until much later.

The eight boundary layer profiles for the suction side are presented in Figures 6.9 to 6.16. In all the profiles, the Baldwin-Lomax model has the least favorable agreement with experiment. The BLM overpredicts the eddy viscosity, which results in a thicker boundary layer for most all the profile stations. The increased viscosity in the inner region also contributes to the higher skin friction that was mentioned above.

Both the Johnson and King models do an excellent job predicting the boundary layer profiles. For profiles up to  $x/c = 0.900$ , the results are almost identical. At stations  $x/c = 0.388$  and  $0.612$ , the experimental data give smaller values of velocity for the points near the wall. It is not clear why there exists such a disagreement. One possibility for the discrepancy may be inaccuracies in the experimental measurement locations near the wall.

For the region near the trailing edge, the baseline model does a better job overall of predicting the velocity. Near the wall, the modified JKM has an increased prediction of eddy viscosity. This is mainly a result of the pressure term added to the inner JKM model. As observed from the diffuser case, the pressure term tends to delay the detachment point. This was an improvement for the diffuser case, but is not as necessary for the flapping foil problem discussed here. The other difference in the two JKM models is near the boundary layer edge. The modified JKM predicts a slightly smaller boundary layer thickness. This difference is small compared to the difference between the modified and baseline models in the inner region. The very last profile at  $x/c = 1.00$  shows the impact of the adverse pressure gradient that exists on the suction side. The flow has reattached at this point, but the velocity profile still reveals some backflow velocity. This type of flow presents a challenge to any turbulence model, yet the baseline JKM does an excellent job of matching the experimental data. To show the near wall region even better, a semi-log plot of the profile is given in Figure 6.17.

## 6.4 Unsteady Flow Case

In order to make comparisons with the experimental data for the unsteady case, a Fourier analysis was done. This allows the amplitude and phases of the harmonics to be calculated, along with their mean value. The computational data was post-processed using the following Fourier formula.

$$U(x, y, t) = U_o(x, y) + \sum_{n=1}^k U_n(x, y) \sin[n\omega t + \psi_n(x, y)] \quad (6.1)$$

The amplitude,  $U_n$ , and phase angle,  $\psi$ , for each harmonic is defined by

$$U_n(x, y) = \sqrt{a_n^2 + b_n^2} \quad (6.2)$$

$$\psi_n(x, y) = \tan^{-1}(a_n/b_n) \quad (6.3)$$

The Fourier coefficients appearing in the above equations are expressed as

$$a_n = \frac{2}{T} \int_{t=0}^T U(x, y, t) \cos(n\omega t) dt$$

$$b_n = \frac{2}{T} \int_{t=0}^T U(x, y, t) \sin(n\omega t) dt$$

The phase angle is given with respect to the flapper angle. The zeroth harmonic is denoted by  $U_o$ , and is obtained by taking the average value of the data over the period.

$$U_o = \frac{1}{N} \sum_{j=1}^N U(t_j)$$

where  $N$  is the number of time steps per period.

For the time-accurate calculations, a time step was taken equivalent to one phase (180 phases per cycle). Several runs were made with the baseline model in which the number of subiterations was set to 50, 70, 100, and 140. This was done to test the number of inner iterations needed to be performed in order to properly satisfy conservation of mass. The solutions showed that 100 iterations per physical time step was more than enough to properly satisfy continuity. Therefore, all the unsteady computations presented below were performed using 100 inner iterations. The norm of the continuity equation was reduced down to  $10^{-4}$  or less when using 100 subiterations. The parameter  $\beta$  was again held at 1000 as in the steady case and the pseudo-time step ( $\Delta\tau$ ) was kept constant at  $10^{12}$  for time accuracy.

Results for the bounding box data are presented first. Figures 6.18 and 6.20 plot the average  $u$  and  $v$  velocities along the top surface of the bounding box. The data is located at  $y/c = 0.2187$ , which is outside the boundary layer, but still influenced by the flow around the stationary foil. A slight difference is observed between the baseline JKM and the other two models. Figures 6.19 and 6.21 plot the amplitude and phase data from the Fourier analysis. The phase data from each turbulence model is

in excellent agreement for both the  $u$  and  $v$  velocities. The phase angle is decreasing in the downstream direction, confirming the traveling sinusoidal wave being shed from the flappers. The numerical velocity amplitudes near the leading edge of the hydrofoil are higher for the  $u$  velocity, but lower for the  $v$  velocity. This same trend in the  $u$  velocity data was also seen in the results of Taylor *et al.* [76]. The differences may be largely due to the limited experimental data used for the inflow boundary condition.

The mean  $C_p$  results are shown in Figure 6.22. The modified JKM seems to correlate better with the Baldwin-Lomax model over the baseline JKM. There is still not sufficient data for the pressure to determine which model gives a more accurate prediction of  $C_p$ . The amplitude and phase data from the harmonic analysis is shown in Figure 6.23 for the suction side only. The modified JKM has higher amplitudes along the foil surface over the other two models and the experiment. For the phase data, the numerical pressure prediction shows to be lagging the experimental  $C_p$  by a small amount. Near the trailing edge, there appears to be a large change in the phase for both the computations and experiment. Experimental data and BLM show a phase lag while both JKM results indicate the tendency for the pressure to have a phase lead.

Boundary layer profiles were compared at six stations along the upper foil surface. The mean tangential velocity profiles are given in Figures 6.24 to 6.29. The corresponding amplitude and phase information is plotted in Figures 6.30 to 6.35.

The baseline Johnson and King model had the best overall agreement with experiment for the mean velocity profiles. There are not as many experimental data points near the wall, but the baseline JKM seems to predict the correct amount of mean backflow velocity at  $x/c$  of 0.990 and 1.000. In the steady case, the modified JKM had similar performance to the baseline model for most of the profiles, but this was not the trend in the unsteady case. The modified model predicted higher velocities for most every location, especially near the trailing edge. The displacement thickness for the modified model tended to be closer to the BLM prediction, which agrees with the close  $C_p$  results between the two models. Neither the modified JKM or BLM captured any of the backflow velocity that was occurring at the trailing edge.

For FFX, the velocity profiles do not undergo a very large change in their shape due

to the unsteady nature of the flow. This was not the case for the diffuser flow, which produced large changes in the profile shape as reflected through the shape parameter  $H$ . Because of this, the amplitude and phase data from the Fourier analysis provides a good way to compare the data. For the profiles near the trailing edge, a distinct phase shift occurs near the edge of the boundary layer. This shift in the phase is also seen in the computational results. The modified JKM appears to have better agreement with the experiment than the baseline model. At the edge of the boundary layer, the baseline JKM is leading all the other models and experimental data. Near the wall, both models tend to have a phase lead over the experimental data, but again the baseline model has the largest phase lead. The overall trend seen in the experiment for the amplitude data is reflected in the computations as well. The modified JKM had the largest peaks in the amplitude. This increased sensitivity to the unsteady flow may be a large result of the inner eddy viscosity model being a function of the pressure gradient.

The time dependent term in the Johnson and King differential equation was observed to have no effect on the velocity profiles in FFX, just as it was observed for the diffuser flow. But for the FFX, a difference did appear near the trailing edge in the amplitude and phase plots. In these figures, the baseline JKM without the time dependent term given in equation (4.9) is plotted along with the other models. A larger phase difference near the wall appears to occur when the time dependent Reynolds shear stress is not included in the differential equation. This was the only effect observed by removing the time dependency in the Johnson and King model.

Overall, the best results are from the baseline Johnson and King model. The JKM has excellent agreement with the experimental data for the velocity profiles and does an adequate job predicting the phase angle. The results presented here for the Baldwin-Lomax model are comparable to the results by Taylor *et al.* [76] and Paterson and Stern [75], who also used the Baldwin-Lomax turbulence model.

## 6.5 Conclusions

In this chapter numerical results were presented for the MIT flapping foil experiment. For the steady FFX case, both the modified and baseline JKM resulted in better predictions of  $C_p$  over the Baldwin-Lomax model. The flow detachment location was best predicted by the baseline Johnson and King model. The modified JKM was not able to predict flow separation due to the additional pressure gradient term added to the inner region JKM. This increased the eddy viscosity, preventing any flow separation. From the tangential velocity profiles, the Baldwin-Lomax model showed poor comparison with experimental data. The eddy viscosity was overpredicted, resulting in a thicker boundary layer and higher skin friction values. For the JKM models, excellent agreement with experiment was observed. The modified model had the most trouble near the trailing edge where the near wall results did not correlate well with experiment. As for the baseline JKM, it matched experimental data well, even at the trailing edge where some backflow velocity was observed.

In the FFX unsteady flow case, the best agreement with the mean velocity profiles was found with the baseline JKM. It again matched experimental data near the trailing edge, predicting separation and then reattachment. Amplitude data showed the modified JKM to be more sensitive to the unsteady flow conditions over the baseline model. Estimates of the phase angle were slightly better for the modified model compared to the baseline model.

The model for the momentum equation normal stress term was tested for the FFX case. No noticeable change was observed in the results from adding the new normal stress model.

As in the low-speed diffuser case, the time dependent term in the JKM differential equation was studied. When the time dependency was removed from the model, the impact on the boundary layer profiles was negligible. The mean flow velocities showed no changes. Recall that this was observed in the unsteady test case for the low-speed diffuser flow. For the case presented here, the time dependency did have an effect on the phase angle and amplitude for the boundary layer profiles, but only in the vicinity of separation. So some effects were seen from including the time dependent

term into the JKM.

Overall, the baseline Johnson and King turbulence model performed well, giving good comparisons with experimental data. For the majority of the computed flow, the model appeared to capture the physics associated with strong adverse pressure gradients and separation. The modified JKM had difficulties predicting flow separation, and matching the mean velocity profiles for the unsteady case. The amplitude and phase angle results for the modified model were slightly better than the baseline model.

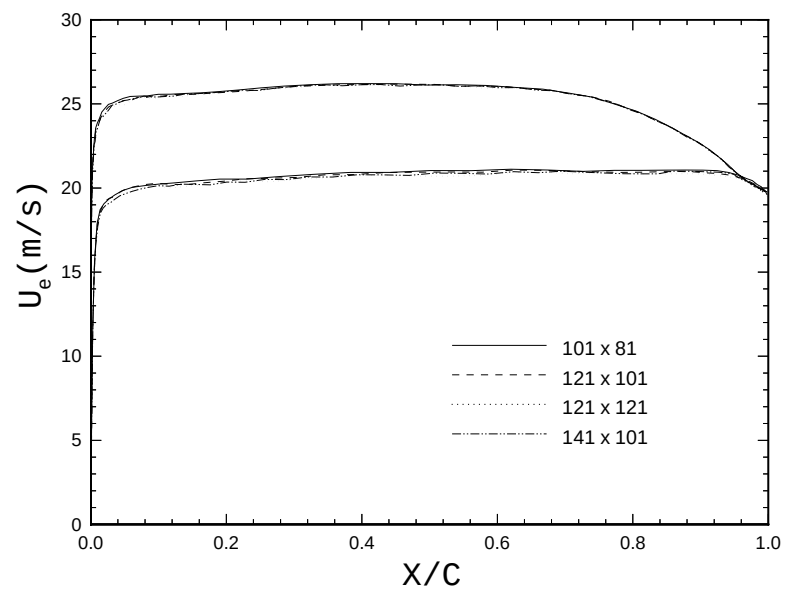


Figure 6.2: Values of  $U_e$  along the hydrofoil using the baseline Johnson and King turbulence model for the grid convergence study on the steady flow case.

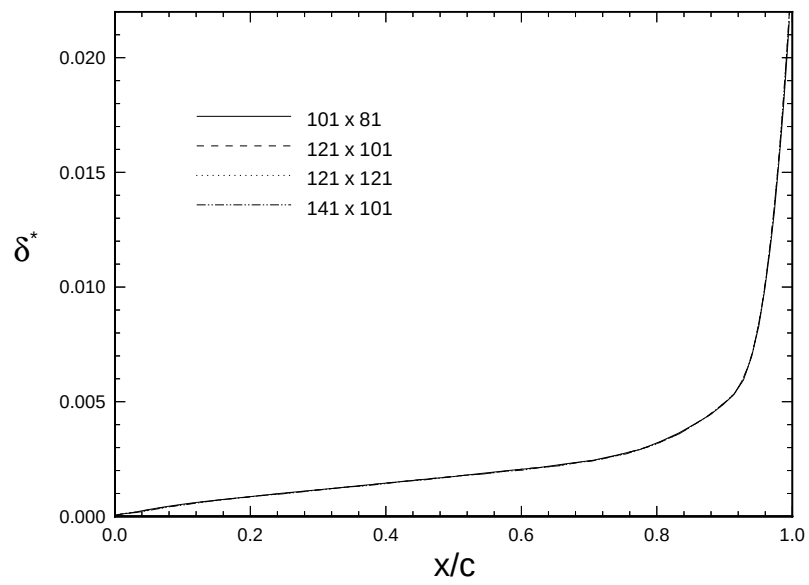


Figure 6.3: Upper surface boundary layer displacement thickness along hydrofoil for the grid convergence study.

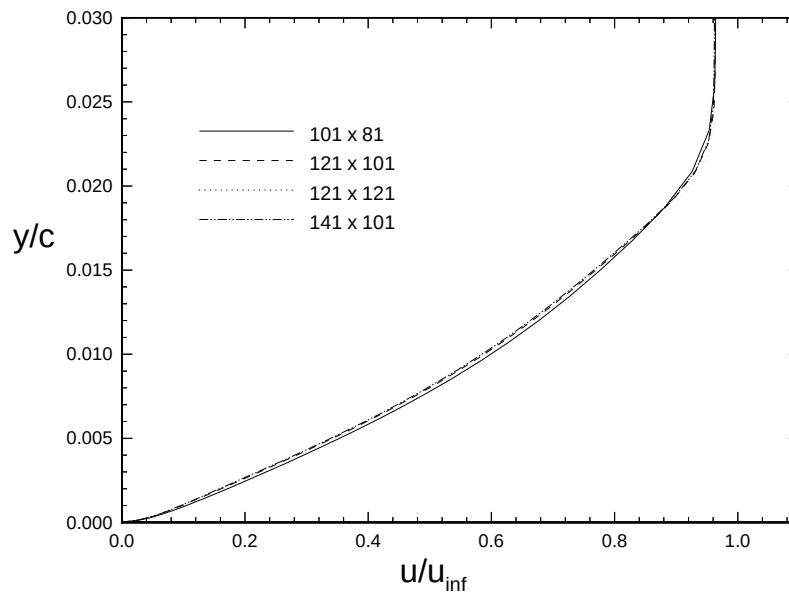


Figure 6.4: Comparison of boundary layer profile for different grids at  $x/c = 0.972$  using the baseline Johnson and King turbulence model.

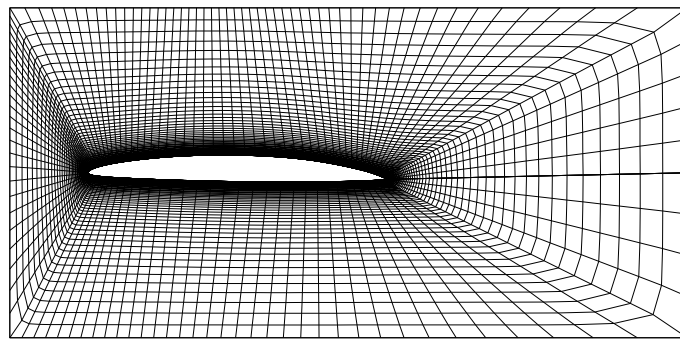


Figure 6.5: O-Grid used in computing the flapping foil experiment for both the steady and unsteady cases ( $121 \times 101$ ).

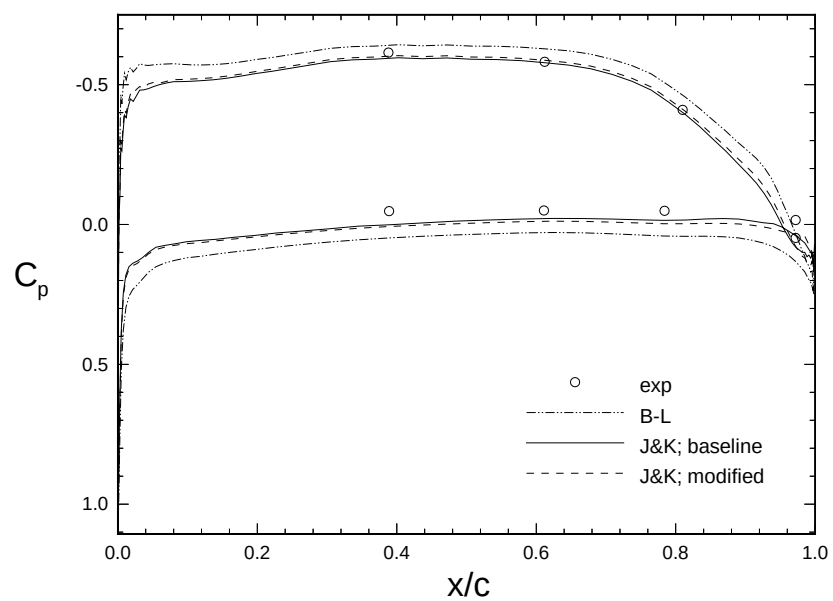


Figure 6.6: Comparison of  $C_p$  for the steady flow flapping foil test case.

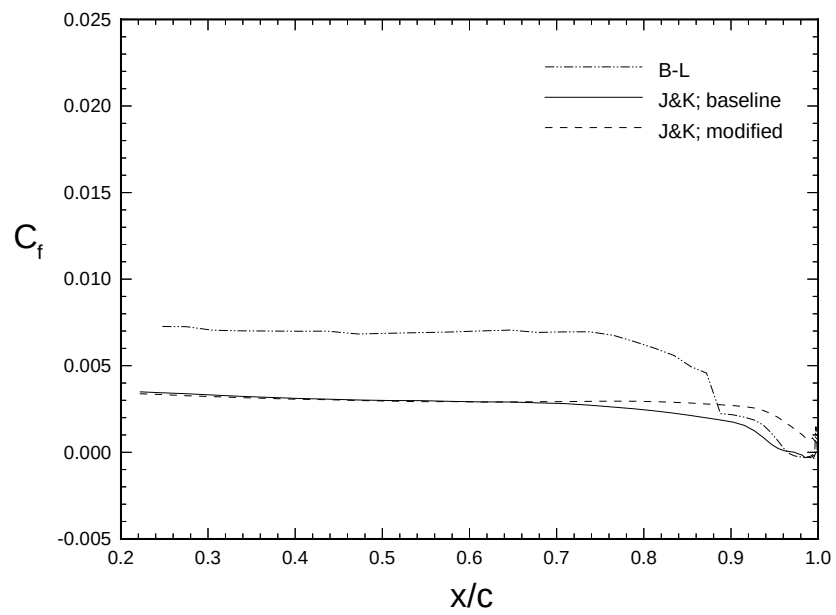


Figure 6.7: Values of  $C_f$  on suction side of hydrofoil (steady case).

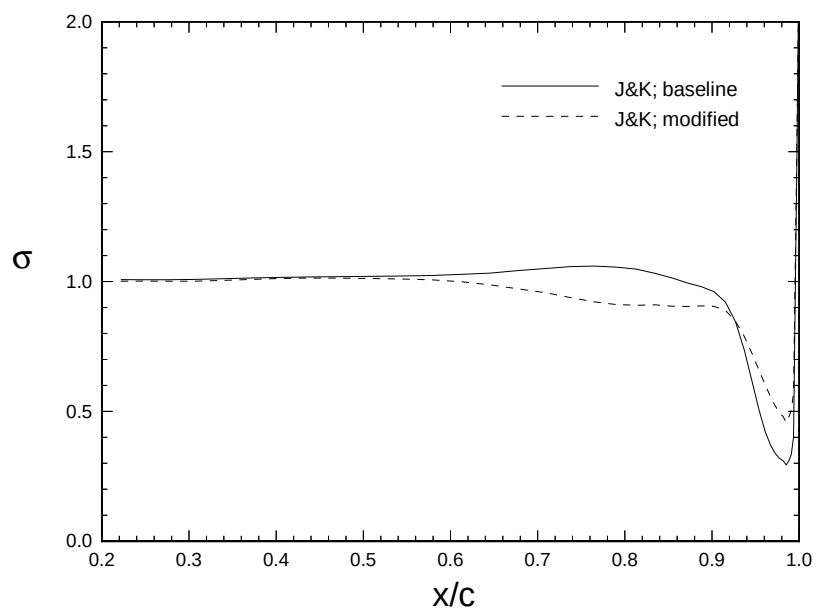


Figure 6.8: Values of  $\sigma(x)$  on the suction side of the hydrofoil used in scaling the outer eddy viscosity for the two Johnson and King models.

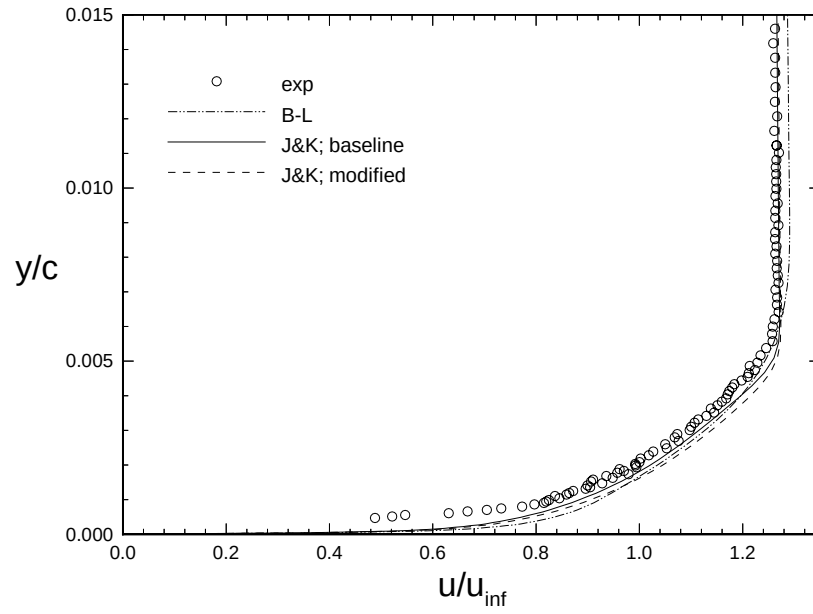


Figure 6.9: Steady case boundary layer profile at  $x/c = 0.388$  for the flapping foil calculations.

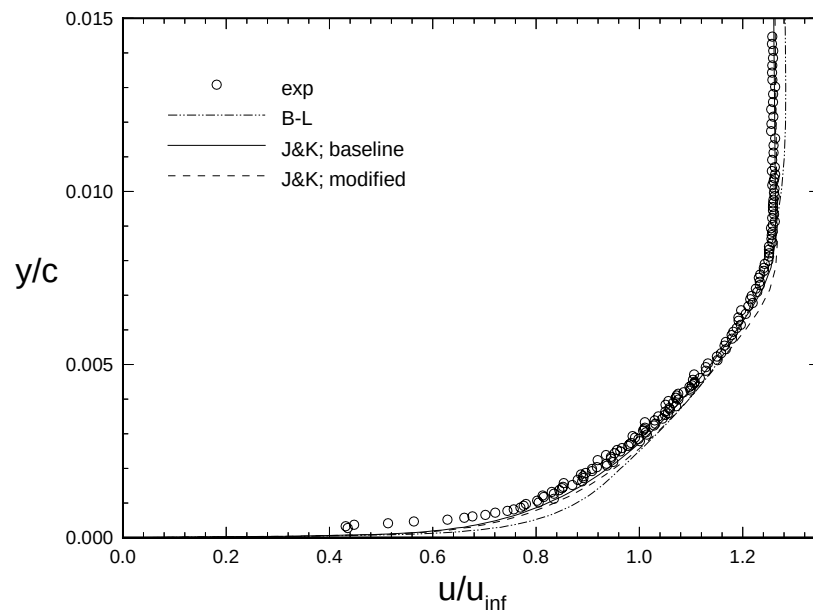
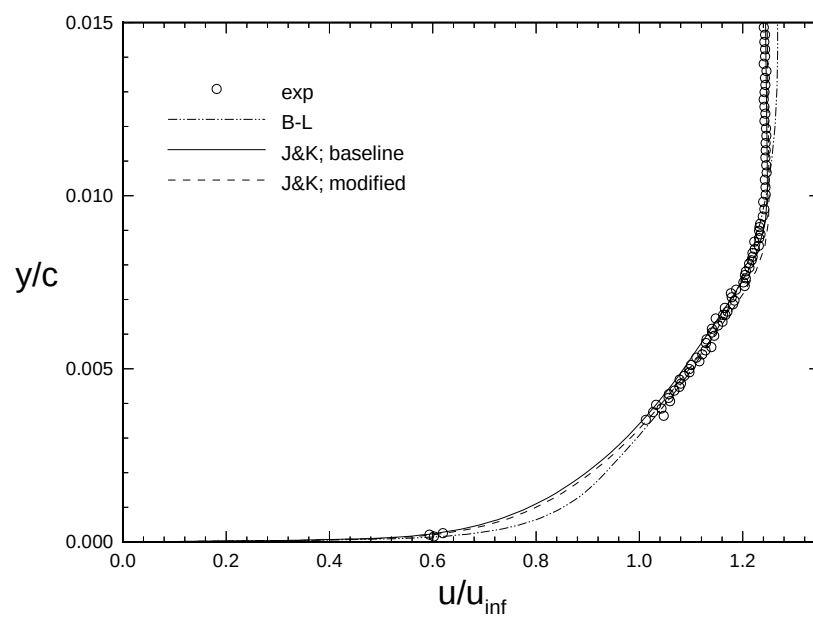
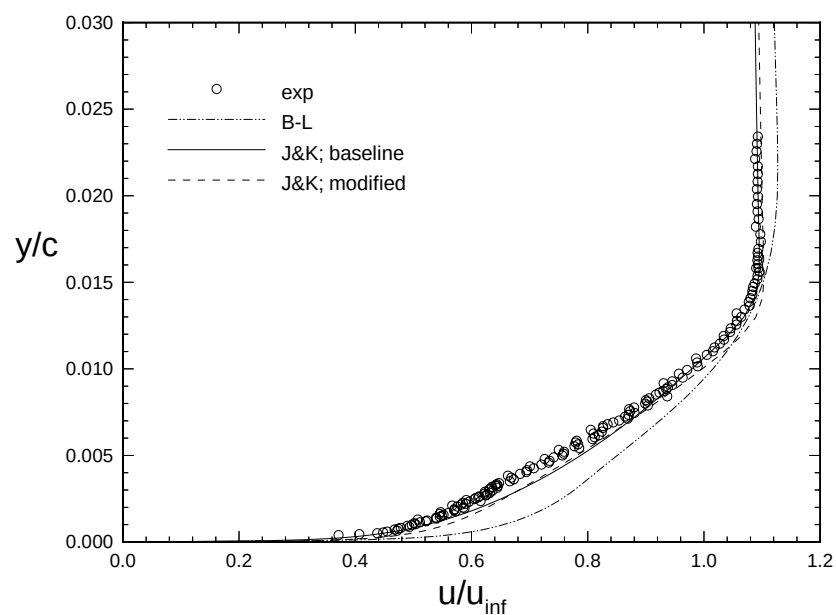
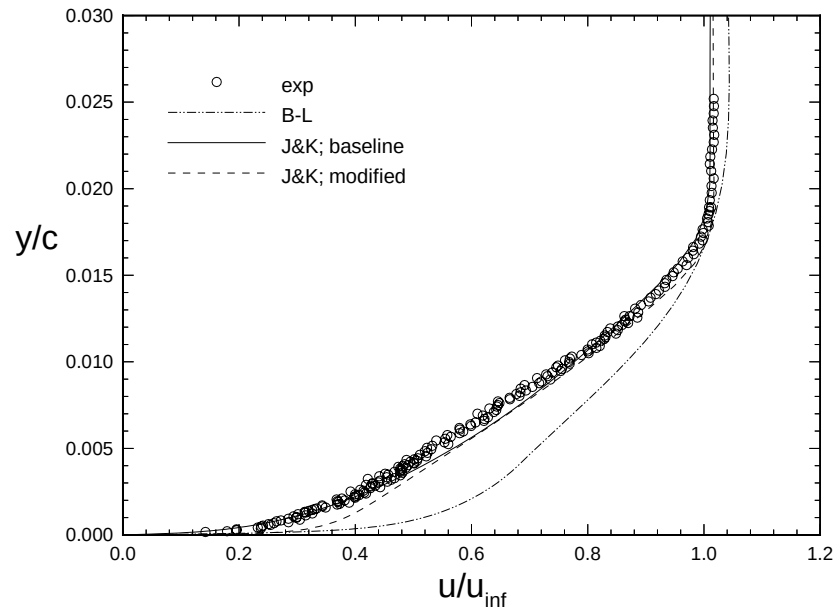
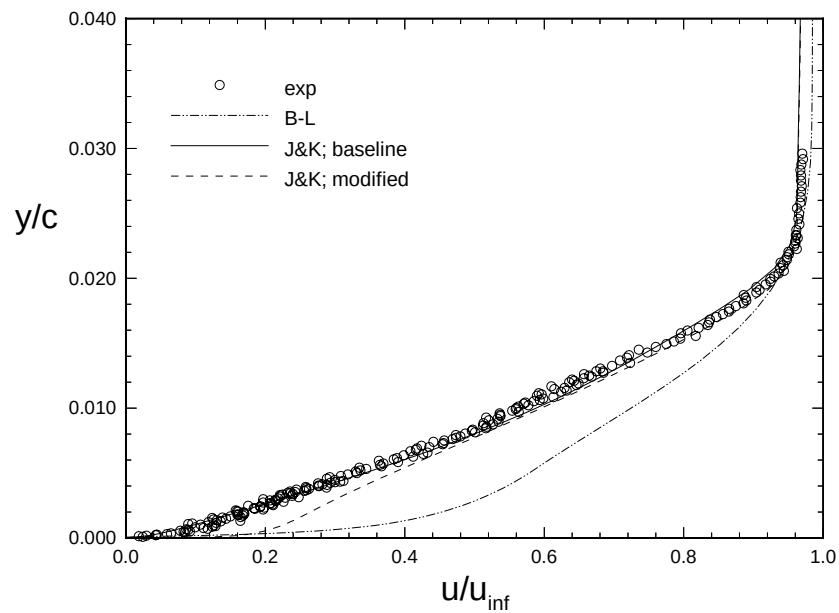
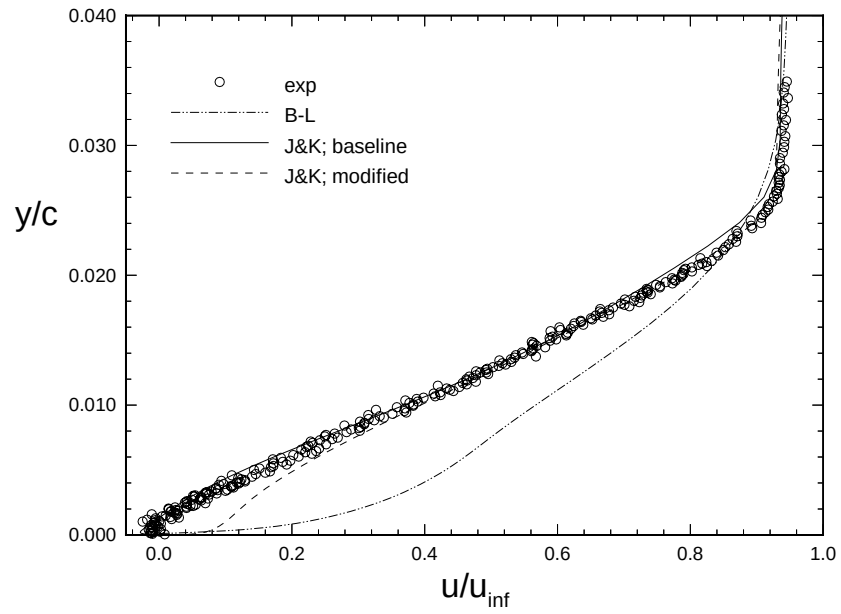
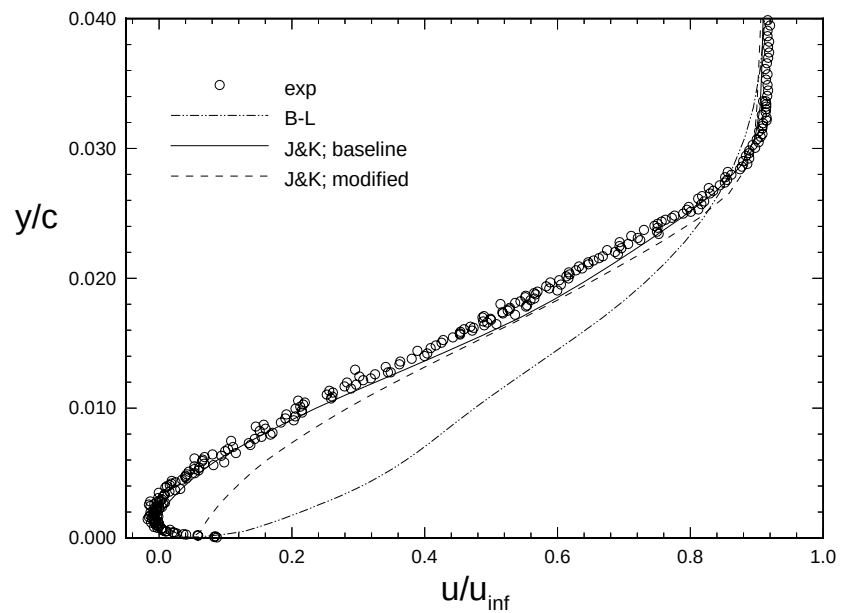


Figure 6.10: Steady case boundary layer profile at  $x/c = 0.612$ .

Figure 6.11: Steady case boundary layer profile at  $x/c = 0.700$ .Figure 6.12: Steady case boundary layer profile at  $x/c = 0.900$ .

Figure 6.13: Steady case boundary layer profile at  $x/c = 0.950$ .Figure 6.14: Steady case boundary layer profile at  $x/c = 0.972$ .

Figure 6.15: Steady case boundary layer profile at  $x/c = 0.990$ .Figure 6.16: Steady case boundary layer profile at  $x/c = 1.000$ .

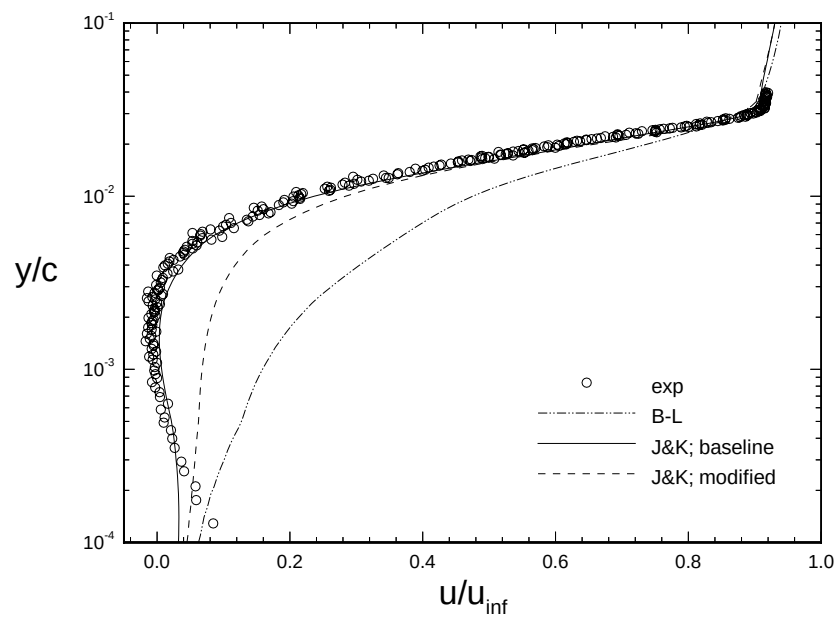


Figure 6.17: Semi-log scale boundary layer profile for the steady case at  $x/c = 1.000$ .

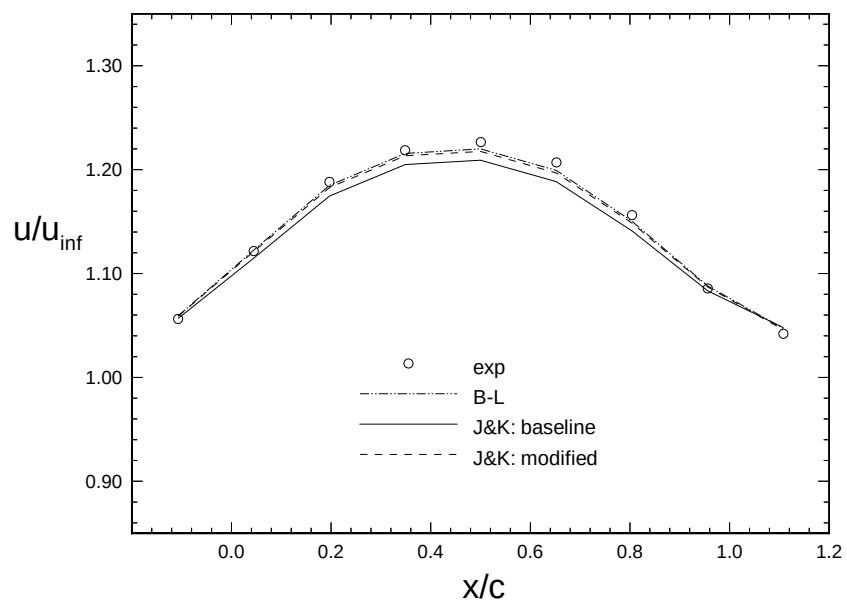


Figure 6.18: Mean  $u$  velocity data on top bounding box ( $y/c = 0.2187$ ) for the unsteady flapping foil case.

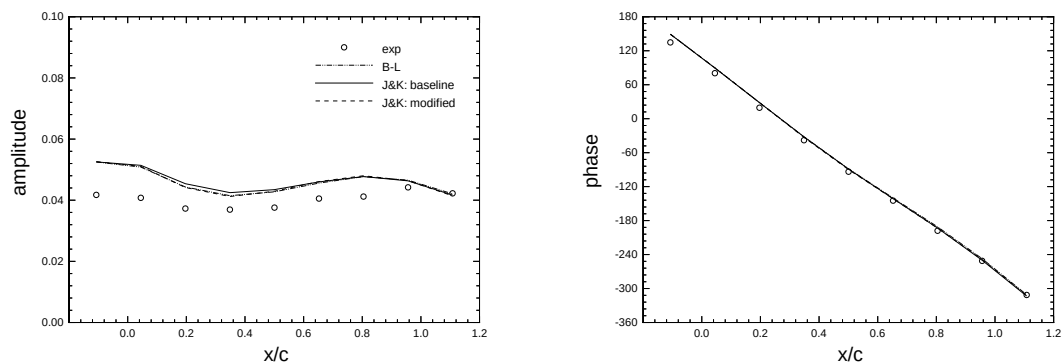


Figure 6.19: Unsteady harmonic data for  $u$  velocity data on top bounding box.

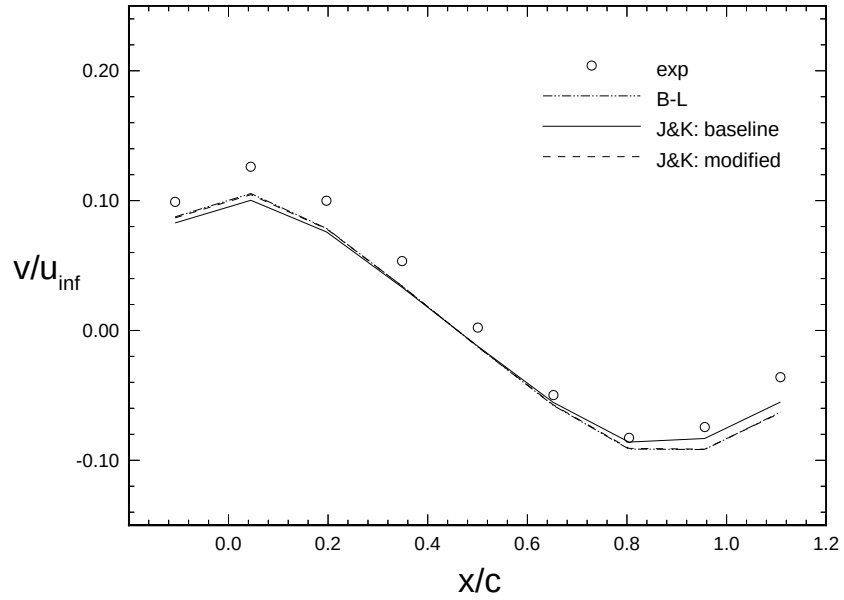


Figure 6.20: Mean  $v$  velocity data on top bounding box ( $y/c = 0.2187$ ) for the unsteady flapping foil case.

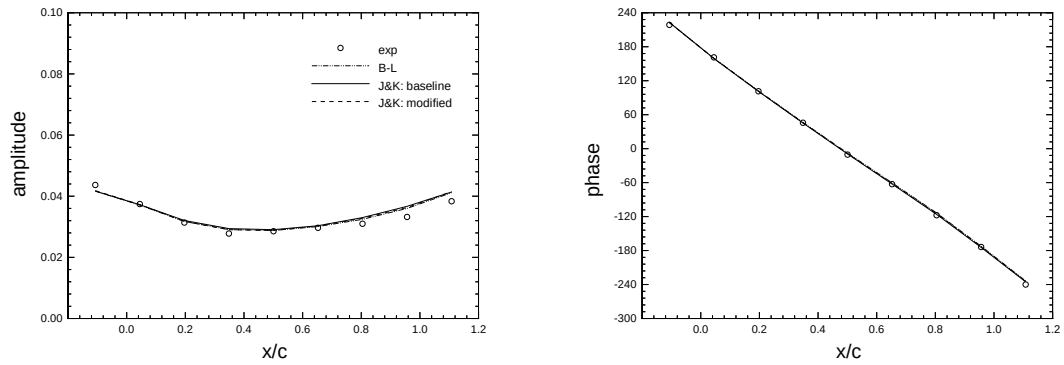


Figure 6.21: Unsteady harmonic data for  $v$  velocity data on top bounding box.

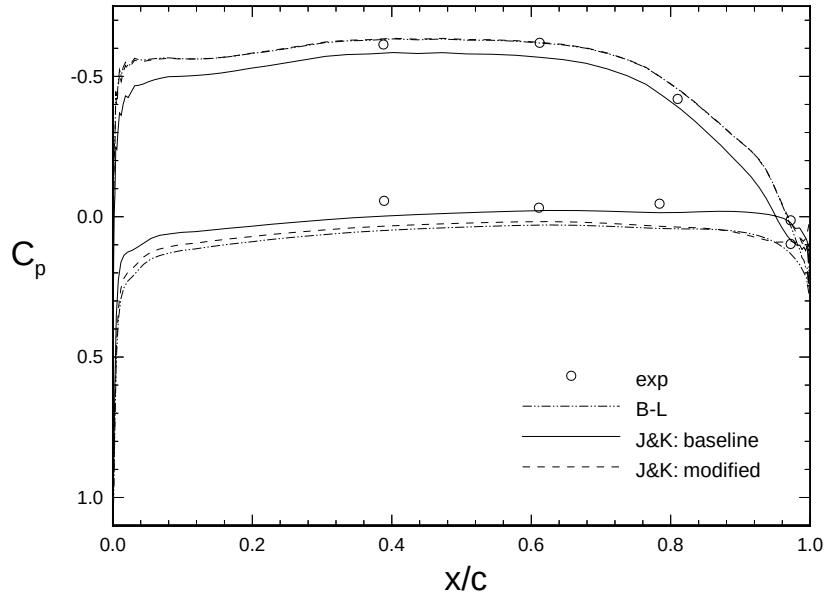


Figure 6.22: Mean  $C_p$  distribution along hydrofoil for the unsteady flapping foil case.

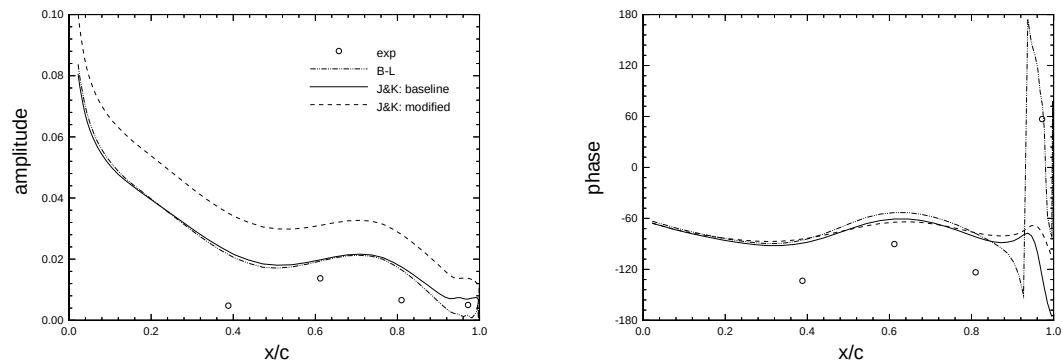
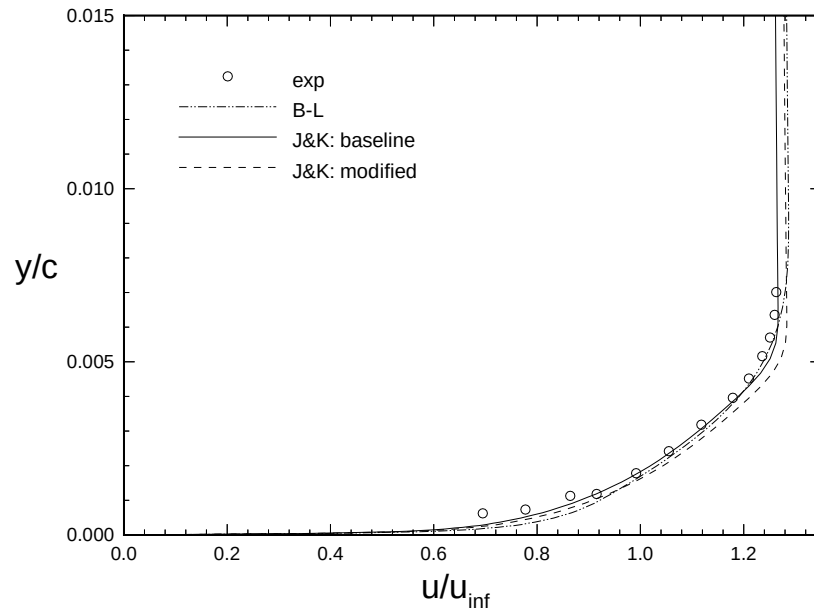
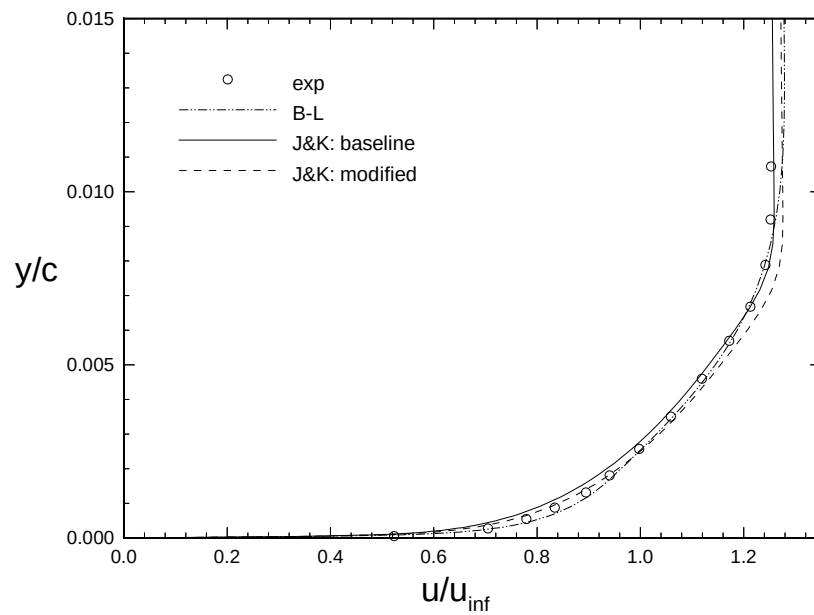
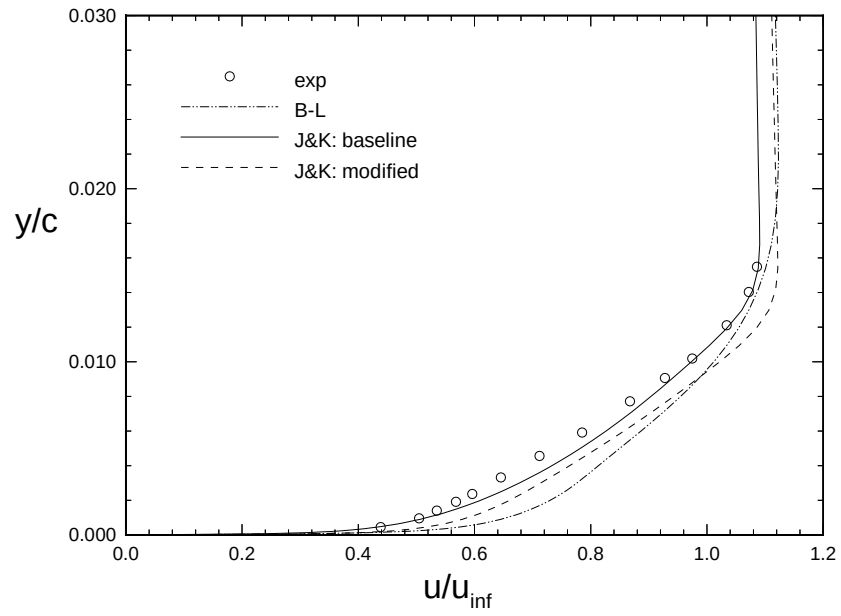
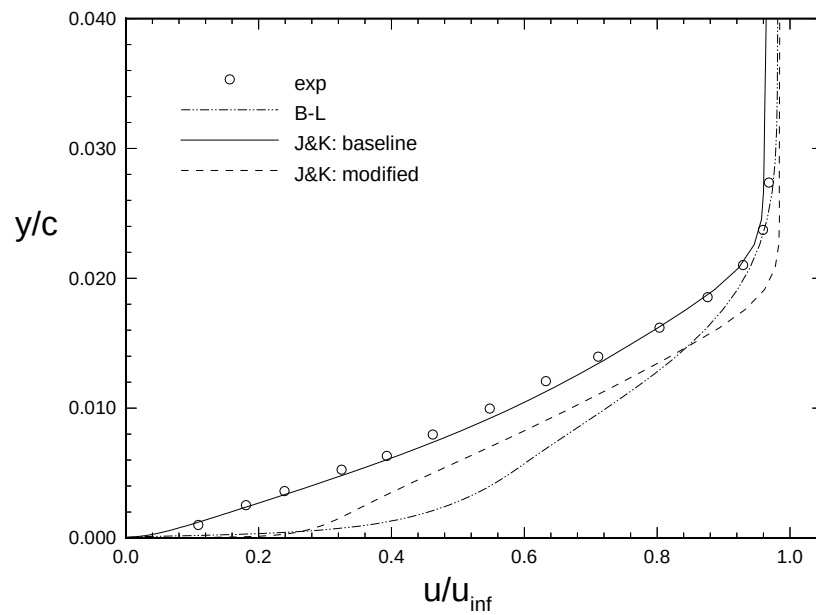
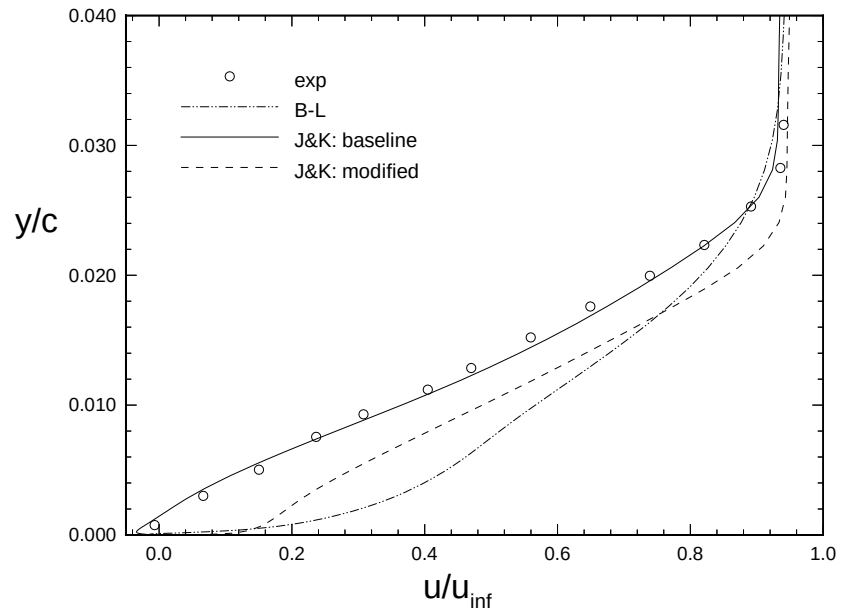
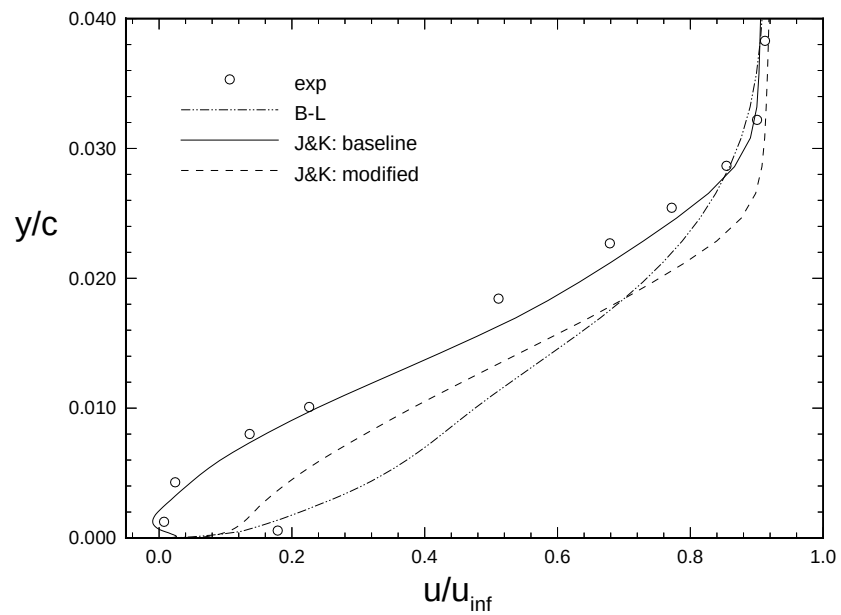


Figure 6.23: Unsteady harmonic data for  $C_p$  on suction side of hydrofoil.

Figure 6.24: Unsteady case mean boundary layer profiles at  $x/c = 0.388$ .Figure 6.25: Unsteady case mean boundary layer profiles at  $x/c = 0.612$ .

Figure 6.26: Unsteady case mean boundary layer profiles at  $x/c = 0.900$ .Figure 6.27: Unsteady case mean boundary layer profiles at  $x/c = 0.972$ .

Figure 6.28: Unsteady case mean boundary layer profiles at  $x/c = 0.990$ .Figure 6.29: Unsteady case mean boundary layer profiles at  $x/c = 1.000$ .

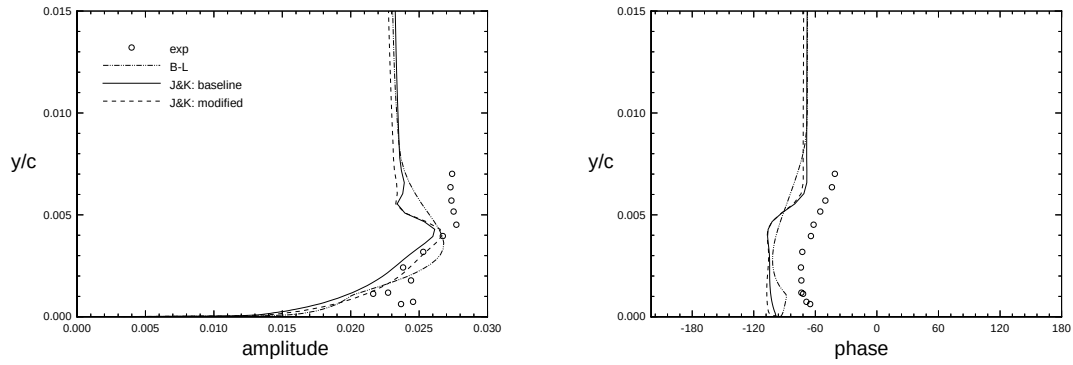


Figure 6.30: Unsteady harmonic data from Fourier analysis of boundary layer profiles at  $x/c = 0.388$ .

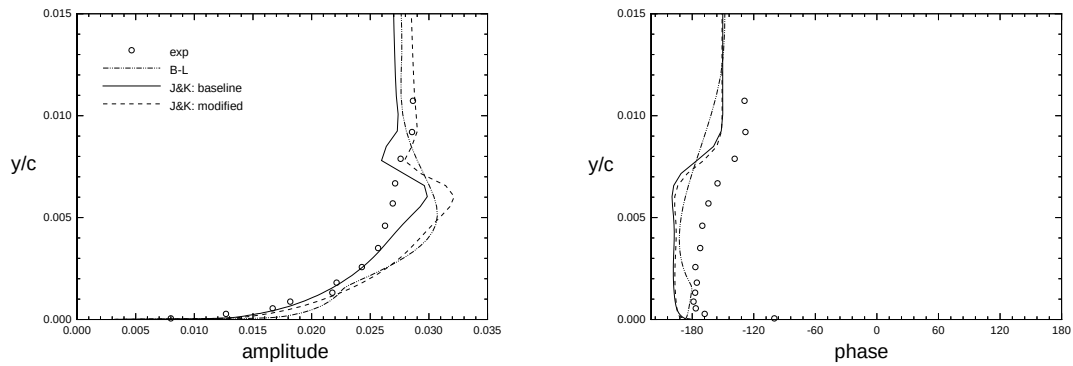


Figure 6.31: Unsteady harmonic data from Fourier analysis of boundary layer profiles at  $x/c = 0.612$ .

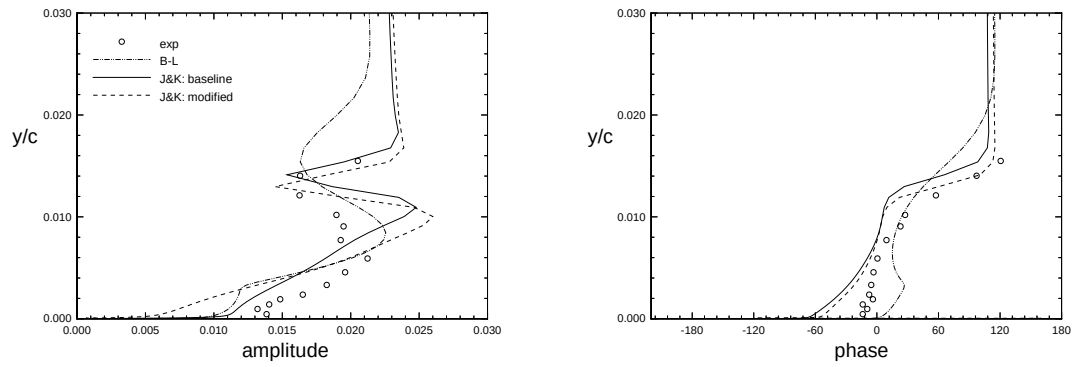


Figure 6.32: Unsteady harmonic data from Fourier analysis of boundary layer profiles at  $x/c = 0.900$ .

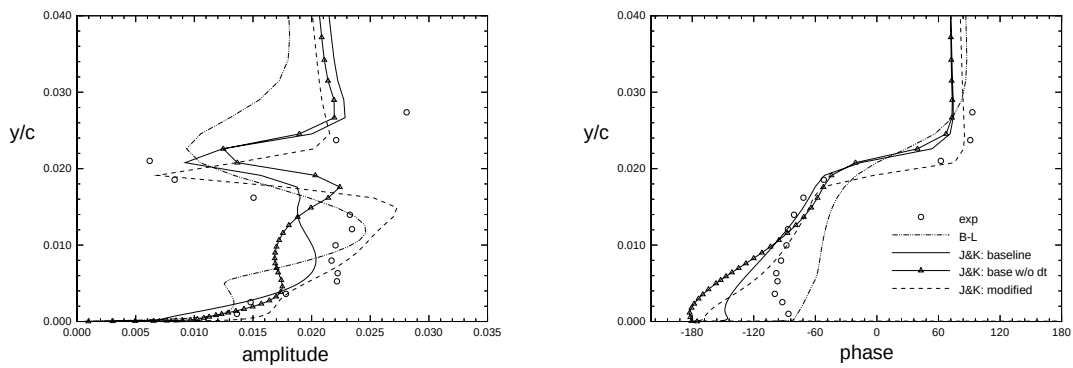


Figure 6.33: Unsteady harmonic data from Fourier analysis of boundary layer profiles at  $x/c = 0.972$ .

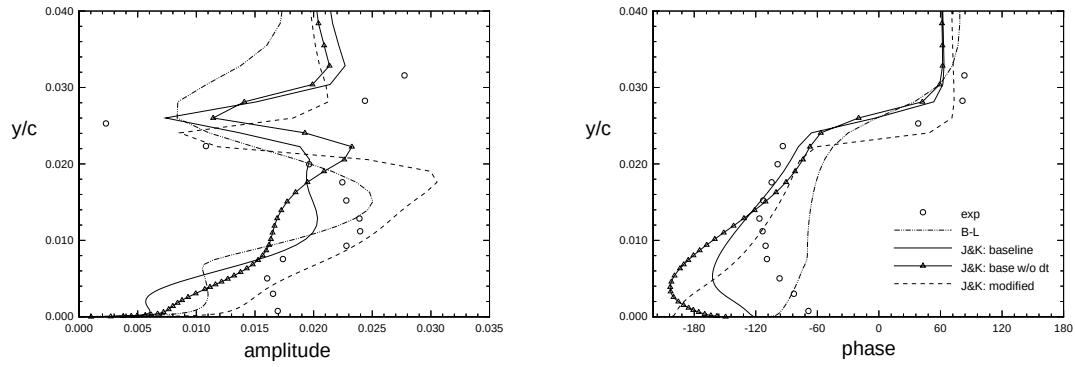


Figure 6.34: Unsteady harmonic data from Fourier analysis of boundary layer profiles at  $x/c = 0.990$ .

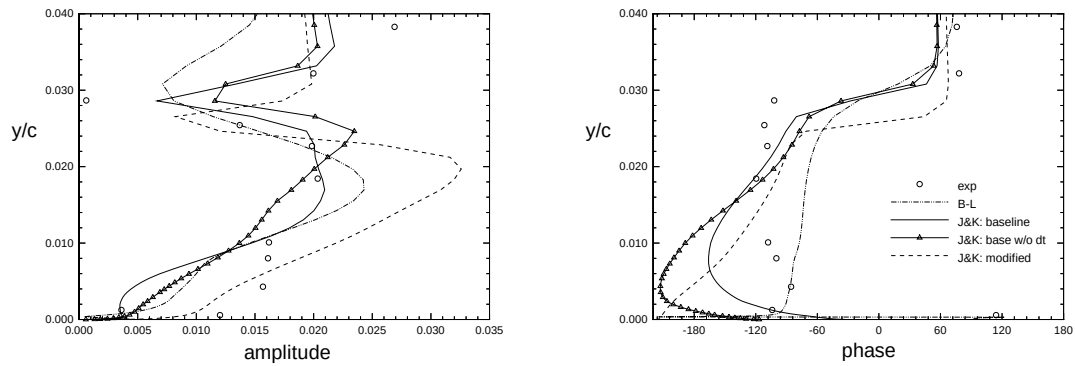


Figure 6.35: Unsteady harmonic data from Fourier analysis of boundary layer profiles at  $x/c = 1.000$ .

# Chapter 7

## Introduction to Potential Flow

### 7.1 Background of Potential Flow in CFD

In the 1990's, research and development of potential flow solvers has taken a back seat to Euler and Navier-Stokes solvers. The speed and memory capabilities of computers has made solutions to the Navier-Stokes equations more common. This has been made possible due to the technology developed during the past decade. But researchers were not always concerned with solving the Navier-Stokes equations. Interest in the potential equation began with transonic flow. Solutions involving transonic flow were desired in the 1960's, and were sought out by solving the Euler equations. A transonic solution using the potential equation in the early 1970's began the push to solve the various forms of the potential equation. Research on potential flow thrived throughout the 1970's and into the early 1980's. In fact, so much work was done on potential flow during that time frame that there has been very little "new" discoveries in potential flow since then. Many of the codes developed during that time are still in use today. Potential flow analysis still plays an important role in aircraft design and optimization. Because of the speed at which potential solutions can be attained, potential solvers fit well into the design process. Even though the accuracy of potential solutions will be less than that of Euler and Navier-Stokes solutions, the basic physics of an inviscid flow field are still captured. The remainder of this section will cover some of the discoveries and accomplishments in potential flow research during the

1970's and 1980's.

By the beginning of the 1970's, finite difference schemes were already in existence for solving Laplace's equation. There were even some solutions for transonic flows, but no real breakthrough had occurred yet. The earliest research into transonic potential flows started with the Transonic Small Disturbance Equation (TSDE). Solving this equation provided a stepping stone for solving the full potential equation. The first suitable scheme for solving the TSDE equation for supersonic flows was done by Murman and Cole [84] in 1971. They made an important observation concerning transonic flows by acknowledging a domain of interest when differencing the derivatives. Each mesh point was correctly treated using a type-dependent difference based upon the domain of dependence. In their paper, a two-dimensional non-lifting airfoil was solved as an example. This paper laid the ground work for the years that followed. A few years later 1973, Murman [85] presented a formulation which efficiently solved the TSDE by using both central and upwind differencing. Central differencing was used for subsonic points, upwind differencing for supersonic points, and special parabolic and shock point operators for points entering and leaving the supersonic region.

In 1972, after Murman and Cole's original paper, a number of other researchers presented solutions to both the TSDE and full potential equation. Steger and Lomax [86] and Garabedian and Korn [87] presented transonic airfoil solutions. Steger and Lomax presented the concept of relaxation with the successive overrelaxation (SOR) scheme to solve the potential equation. They presented solutions to airfoils with both camber and angle of attack. The thin airfoil theory was no longer applied, which was the case in Murman and Cole's original paper. Garabedian and Korn developed a second order accurate version of Murman and Cole's scheme. They also solved the full potential equation in non-conservative form. The conformal mapping scheme of Sells [88] was used to transform the coordinate system. In this mapping scheme, the exterior profile of the airfoil was mapped onto the interior of a circle. This grid transformation method became widely used and accepted. Also in that same year, Ballhaus and Bailey [89] and Bailey and Steger [90] both performed solutions for transonic flow about wings using the TSDE.

In 1974, Jameson [91] came up with a more flexible scheme for the potential equation. A problem existed when the flow was supersonic but not aligned with the principle flow direction. The proper domain of dependence changes with the flow direction, which Jameson took into account in the discretization. Thus came into existence Jameson's well known rotated difference scheme. Jameson used conformal mapping to solve both two and three dimensional problems. Also in 1974, Murman [92] showed that the potential equations needed to be solved in conservation form in order to have the correct jump conditions. For weak shocks the non-conservative form performed well, but for stronger shocks the proper jump conditions became an issue. Murman solved the TSDE using the non-conservative form of the equation, but introduced a shock-point difference operator. When this special shock-point operator was used, the solution became equivalent to solving the conservative form. In the following year, Jameson [93] took one step further and solved the full potential equation in conservation form. This gave an improved representation of shock waves in comparison with the non-conservative schemes. To handle the supersonic regions of the flow, Jameson applied the concept of artificial viscosity to the difference scheme.

Work using the finite element method was also taking place during the 1970's, but at a slightly slower pace. The first solutions of incompressible potential flow using the finite element method came from Argyris *et al.* [94] (1969) and De Vries and Norrie [95] (1971). Moving from Laplace's equation to the nonlinear compressible equation, Thompson [96] (1974), Periaux [97] (1975), and Shen and Habashi [98] (1976) performed a number of solutions using quadrilateral elements where the discretization came from either the potential function or the stream function. The first transonic flow computations for finite elements came in 1976 by Glowinsky *et al.* [99] and Ecer and Akay [100].

The first finite volume calculations were done by Jameson and Caughey [101] in 1977. They developed a finite volume approach where the fluxes at a cell face were obtained using averages from the corner points. A staggered box scheme was used such that velocity and density were calculated in the primary cells and a flux balance was then performed in the secondary cells. Another important result of their paper was that non-orthogonal curvilinear grids were used successfully for aligning the grid

with complex boundaries. Results were given for a swept wing and a wing-cylinder combination.

A new trend in solving the full potential equation came about in 1978 by Hafez *et al.* [102]. The conservative form of the full potential equation was solved by employing artificial compressibility. The concept of artificial compressibility was first introduced by Harten [103] for a single conservation law described by a hyperbolic equation. When applied to the potential equation, the needed viscosity in the supersonic region was introduced through the density. The density would become modified in regions of supersonic flow such that the needed upwinding would be provided. The scheme has been given the name density biasing. The new concept applied central differencing everywhere, even in the supersonic regions. Examples of flow around a cylinder and a NACA 0012 were given. The different iterative procedures of that time were also discussed in this paper. The most common methods for solving either the TSDE or the full potential equation was the implicit SOR and alternating direction implicit (ADI) methods. These methods were not new, but had been around since the 1950's. The first version of the SOR scheme was developed by Frankel [104] (1950) and was used to solve Laplace's equation. ADI schemes were first seen in 1955 by Peaceman and Rachford [105]. Explicit and direct methods were also discussed in the paper by Hafez *et al.*, but were shown to be much slower than SOR and ADI.

More efficient schemes for solving both the TSDE and the full potential equation soon appeared. The various schemes were based upon the approximate factorization (AF) method, which was the basis for the ADI scheme currently being used. A number of new AF schemes were developed by Ballhaus and Steger [106], Ballhaus *et al.* [107], and Baker [108]. These new schemes all proved to be faster than the existing ADI and SOR methods. Baker noted that the speed up of the AF3 scheme was a factor of five to ten times faster than SLOR. A more complete description of each AF scheme is given by Holst [109] and Holst and Ballhaus [110].

Along with the new AF methods, multi-grid methods were also being researched and applied to potential flows. The first implementation of the multi-grid method to transonic flow was by South and Brandt [111] (1976). They solved the TSDE

equation using multiple grids for a nonlifting airfoil problem. A few other multigrid applications from that time were documented by Fuchs [112], Arlinger [113], and Holst [114]. In 1979 Jameson [115] was the first to apply multigrid to the conservative full potential equation. Jameson used the alternating direction iteration scheme on each mesh before moving to the next level. The results for two dimensional airfoils using conformal mapping were impressive. Results could be attained within several multigrid cycles. This increase in efficiency was also confirmed by Boerstool [116], who also noted that the shock position was not efficiently updated by the multigrid method. Different multigrid strategies and smoothing operators for solution of the potential equation are discussed more by Van der Wees *et al.* [117].

The technique of artificial compressibility was being widely used by the early 1980's. One more concept in the area of upwinding was to be introduced. This new concept was given the name flux biasing or flux upwinding. The idea came from the developments of monotone schemes for the Euler equations. Flux biasing helped capture the shock location better and eliminated most of the oscillations that occurred around the discontinuity. The idea was initially introduced by Engquist and Osher [118] in 1980. Some of the first applications to the full potential equation were made by Boerstool [116] (1982) and Osher [119] (1982). A comparison between the flux biasing and density biasing schemes was done by Volpe and Jameson [120]. In their paper they showed that for weak shocks the flux biasing scheme gave sharper resolution of the shock. As the shock strength increased, the two schemes performed equally well in solution quality, speed, and robustness.

The most recent contributions to research in the potential flow area has been made by Holst [121]. Holst developed a full potential flow solver using the Chimera grid approach. The algorithm is applicable to three dimensional flows over wings or wing/body geometries. An inner grid is used to describe the wing surface, while an outer grid describes the farfield region. The grids overlap, taking advantage of the Chimera grid approach.

## 7.2 Numerics of Potential Flow in CFD

The potential equation has been widely used for transonic flow fields. A typical transonic flow field will consist mostly of subsonic flow with embedded regions of supersonic flow. Airfoils and wings will sometimes fit into this category when the outer freestream flow is subsonic while the flow over the upper surface contains a region of supersonic flow. The outer subsonic region will be elliptic in nature, giving rise to signals propagating in all directions. In elliptic solutions, every point in the domain will be influenced by every other point. As the flow velocity nears the sonic condition, these signals will have a stronger propagation strength in the flow direction. In other words, a point in the flow will be more affected by what is happening upstream of that point. As the flow becomes supersonic, the nature of the flow changes to hyperbolic, and the information signals will travel only in the flow direction. A domain of dependence is formed where the flow properties at a single point are only affected by what is happening in this upstream domain. In a similar way a domain of influence exists where a point in the flow can only impact other points that lie in a domain downstream of the point. When the domain of dependence and influence exist, information in the subsonic field traveling upstream must move around the supersonic region or zone. Sonic lines and shock waves will be present in these situations and serve as boundaries between the elliptic and hyperbolic regions. It is easy to see that the flowfield becomes more complex with transonic flows, making numerical solutions more of a challenge as well. The time it takes to attain a transonic solutions is typically slow compared to solutions where the flow is entirely subsonic or supersonic.

Since the potential assumption assumes irrotational flow, the local vorticity production will be zero. But for an airfoil that produces lift, there is a finite circulation value (non-zero vorticity) around the airfoil. The circulation is given the symbol  $\Gamma$ . In order to have a lifting airfoil in a potential flow field, a circulation must be imposed. If this were not done, the airfoil or wing would produce zero lift at any angle of attack. The value of  $\Gamma$  will be unique for each problem, and cannot be determined beforehand. For lifting airfoils, the circulation value is obtained by the Kutta-Joukowski condition. This condition states that the best approximation to the circulation value for

modeling viscous unseparated flows will be obtained if a stagnation point is located at the trailing edge. For a sharp trailing edge, the stagnation condition is satisfied if either the pressure or velocity on the top and bottom surfaces of the trailing edge are equal. To apply the circulation value to an airfoil, an artificial boundary or cut must be made from the airfoil surface to the farfield boundary. Along this cut, a jump in the potential function is enforced. The potential jump is constant at each point along the cut and must correspond to the value of the circulation. To satisfy mass conservation over the cut, all flow variables must be continuous. This will mean that the derivatives of the potential across the cut must not be discontinuous, even though the potential function itself will be.

For steady state problems, a solution of the full potential equation will be irrotational and have the property of constant entropy and total enthalpy. Physical problems that reflect these properties will imply that the full potential equation gives a solution equivalent to the Euler equations. When conservation of mass is satisfied, the conservation of momentum and energy will also be met. So a potential solution for a subsonic airfoil should be identical to an Euler solution. The only difference will occur in how the equations are numerically solved. The potential model begins to break down in the presence of discontinuities such as shock waves due to the change in entropy that physically occurs in nature. The Rankine-Hugoniot relations show that a rise in entropy occurs across a shock. For a uniform shock intensity, the entropy downstream of the shock will be constant, but at a different value than the entropy upstream of the shock wave. This constant entropy state implies that the flow downstream will also be irrotational. For most problems, the shock wave will be curved and therefore have a varying entropy after the shock as well. The flow will then become irrotational after the shock, indicating that the full potential model will not have consistent solutions with the Euler equations. The potential flow equations will allow a shock discontinuity to occur, but the jump relations across the shock will remain isentropic. Therefore, the Rankine-Hugoniot relations cannot all be satisfied by the full potential equation. In fact, only the conservation of momentum across a discontinuity will not be satisfied, while the relations for mass and energy will still hold true [9].

So the question of how accurate the potential flow equation is in the presence of shocks is now asked. In addition to this, a second question may be at what Mach number does a potential solution become unreliable? These questions may be addressed by observing the properties across a normal shock assuming both isentropic and non-isentropic flow. For an isentropic jump, the total pressure will remain constant across the shock, while for a non-isentropic jump the ratio of total pressures across the normal shock is given as [122]:

$$\frac{p_{o2}}{p_{o1}} = \exp\left(-\frac{\Delta s}{R}\right) = \left[\frac{(\gamma+1)M_1^2}{(\gamma-1)M_1^2+2}\right]^{\frac{\gamma}{\gamma-1}} \left[\frac{\gamma+1}{2\gamma M_1^2-(\gamma-1)}\right]^{\frac{1}{\gamma-1}}$$

In this equation, the subscript 1 indicates the flow condition upstream of the shock and 2 indicates conditions downstream. A plot is given in Figure 7.1 revealing the percent error for the isentropic jump relation with increasing Mach number. For  $M_1 \leq 1.30$ , the total pressure loss is lower than 2%. For most engineering applications, this is an acceptable level. The Mach number downstream of the shock can also be observed for both cases. For  $M_1 = 1.25$ , the percent error in the isentropic case is roughly 3%. So the assumption of isentropic flow across a shock is not a bad one if the Mach number remains close to unity. Therefore, errors associated with the potential assumption will remain low if the normal component of the Mach number stays below  $M \approx 1.25 - 1.30$ .

If the shock location and strength from potential solutions were compared to solutions from the Euler equations, a general trend would be observed for transonic flows. Due to the isentropic assumption, the potential shock strength will always be stronger than a shock from an Euler solution. This will in turn cause the location of the shock to be further downstream on an airfoil or wing. The higher the Mach number, the stronger these effects will be on the solution.

For transonic flow cases, it has been observed that solutions to the potential flow equation are not always unique. This was first observed by Steinhoff and Jameson [123]. They performed calculations on a symmetric airfoil at various angles of attack and Mach number. They discovered that three solutions were possible for the NACA 0012 airfoil at zero angle of attack. The lift would be either positive or negative, corresponding to a non-physical solution, or zero corresponding to the physical solution.

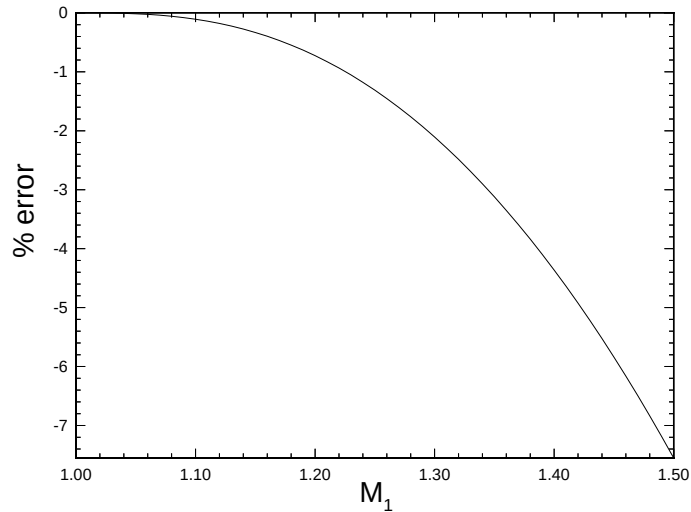


Figure 7.1: Percent error of stagnation pressure loss as a function of upstream Mach number for a normal shock.

The results were found to be independent of the grid type and size. The nonunique solutions have no physical significance. These results were later verified by Salas *et al.* [124] [125]. The problem with nonunique solutions has been observed in internal flows as well. For example, an infinite number of shock positions exist for a given back pressure or exit Mach number. This is due to the fact that the flow is isentropic, in which case nothing connects the shock position with the outlet conditions. The shock does not feel the back pressure, which is necessary to give a unique shock location. A more detailed analysis of the internal nozzle problem is given by Mason [126]. The nonuniqueness problem has only been found to occur for conservative formulations. The non-conservative formulation has not been found to have this problem, but this formulation does not conserve mass across a shock.

### 7.3 Present Formulation and Objectives

In the present formulation of the full potential flow solver, a cell centered finite volume method is used. Other alternatives for discretization include the finite difference and

finite element methods. With the finite volume method, the integral form of the governing equations are applied to a control volume, also referred to as cells. In the past it has been common for these control volumes to take on the shape of hexahedrals (3-D) or quadrilaterals (2-D). With these shapes, the cells can be fitted together and numbered such that given the index of a cell, the neighboring cells that surround it are automatically known. When a code takes advantage of the directional ordering of cells within a grid, it is referred to as a structured flow solver. The finite difference method bases its formulation on this ordering of grid points in order to discretize the differential form of the governing equations. If tetrahedras or prisms are used as control volumes instead of hexahedras, the ability to logically order the cells in the grid is lost. In this case a book keeping system is required in order to keep track of neighboring cells. A flow solver which uses this type of methodology is referred to as unstructured. Both the finite volume and finite element methods are capable of either grid structure, while the finite difference method has a dependence on the grid ordering. Similar to the finite volume method, the finite element method also begins with an integral form of the governing equations, usually known as the weak formulation. Both finite element and finite volume methods have had great success in solving the full potential equation on structured grids. On unstructured grids, most applications involve solving the Euler or Navier-Stokes equations. A few methods have been developed for solving the potential equation using the finite element method on unstructured meshes. The first reported case was by Vigneron *et al.* [127], which was later followed by Kinney *et al.* [128] [129]. Kinney solved the conservative form of the full potential equation on tetrahedral grids using a preconditioned conjugate gradient squared algorithm. Using a different approach based upon a Cartesian mesh, Madson *et al.* [130] developed an unstructured algorithm which embeds a geometry in an initially uniform Cartesian grid. The grid is then cut away to provide the surface definition of the geometry. The finite element method will not be discussed here in any more detail, but instead focus will be given to the finite volume method for solutions to the full potential equation on unstructured grids. Currently there are no known finite volume based full potential flow solvers for unstructured meshes.

The production of unstructured codes in the CFD community has been steadily

growing. A large advantage to unstructured codes is the use of tetrahedral cells in the grid generation process. There is much more freedom in the size and number of cells that can be placed around geometries to model the flowfield. A structured grid is usually composed of H, C, or O-type grid structures. These grid configurations may work well for simple geometries, but for complex bodies the number of cells used to capture the necessary physics may not be efficient. There may be clustering of cells away from high gradient regions, resulting in unnecessary cells used to describe the flow. The greater flexibility with unstructured grids allow fewer cells to be needed to attain the same level of accuracy on structured grids. Clustering of cells is more efficient, allowing the physics to be captured better.

The objective here is to develop an unstructured full potential flow solver. Solutions are to be performed for either two or three dimensional flows. The flow solver is to be capable of flows ranging from low subsonic to transonic conditions. To take advantage of unstructured grid technology, the flow solver will assume control volumes of varying shapes and sizes. This will allow basic cell shapes like tetrahedrals, prisms, and hexahedrals to be used without any unnecessary changes made to the solver. With these features, the present flow solver will be a valuable tool for engineers. Solutions from the full potential equation are useful for design analysis or for initialization of Euler flow solvers. Solving the full potential equation should be more efficient than solving the Euler equations, giving faster run times. This can be very useful during the design process where multiple computations need to be performed as small changes to the geometry are made.

# Chapter 8

## Potential Flow Solver

### 8.1 Introduction

In this chapter, the methodology developed to solve the full potential equation on unstructured grids will be covered. The finite volume approach is applied where the unknowns are located at the cell centers. The potential equation is written in integral form which is then applied to each cell, also referred to as a control volume. The types of cells that the potential equation will be applied to are hexahedrals, tetrahedrals, and prisms (see Figure 8.1). The sides or faces of these cells are either triangular or quadrilateral. The formulation to be presented is very general, such that any of the cell types mentioned above can be used without any special modifications to the algorithm. The algorithm is also applicable to either two or three dimensional flow problems.

In the following sections, the full potential equation will be derived along with other auxiliary equations needed in the calculations. The governing equation will then be discretized based upon the finite volume technique. The formulation of the fluxes will be covered along with how to treat the hyperbolic nature for supersonic flows. The artificial viscosity term for upwinding is done through the density. Various iterative schemes will then be discussed. The iterative schemes fall into two categories, explicit and implicit. Most of the attention will be given to the implicit solvers. Finally, the chapter will end with a discussion on the boundary conditions and the

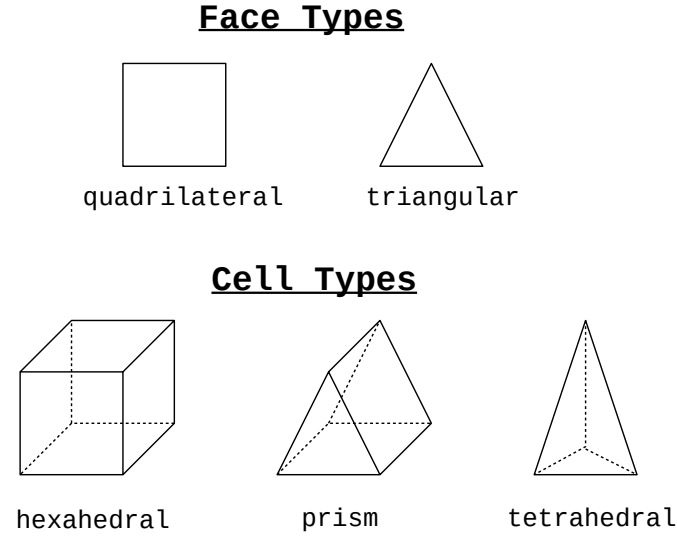


Figure 8.1: The different types of faces and cells in the code.

Kutta condition.

## 8.2 Governing Equations

The full potential equation with the necessary equations for closure will be presented in this section. All derivations will be done in two dimensions for clarity. Extension to three dimensions is straight forward. The approach taken here to derive the full potential equation is just one of several different ways. More information on the derivation can be found in references [131] and [132].

An equation that will be of use in deriving some of the relations in the potential formulation is the Gibbs relation,

$$Tds = dh - \frac{1}{\rho}dp \quad (8.1)$$

In this equation,  $s$  is the specific entropy and  $h$  is enthalpy. The total enthalpy is defined as,

$$h_o = h + \frac{u^2 + v^2}{2}$$

Next, the  $x$  and  $y$  momentum equations for an inviscid fluid flow are written in

conservative, differential form.

$$\frac{\partial \rho u}{\partial t} + \frac{\partial \rho u^2}{\partial x} + \frac{\partial \rho uv}{\partial y} = -\frac{\partial p}{\partial x} \quad (8.2)$$

$$\frac{\partial \rho v}{\partial t} + \frac{\partial \rho uv}{\partial x} + \frac{\partial \rho v^2}{\partial y} = -\frac{\partial p}{\partial y} \quad (8.3)$$

The only assumption made to the momentum equations is that of inviscid flow. These equations are still valid for unsteady, compressible flows. The continuity equation, without any assumptions made, is given as,

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} = 0 \quad (8.4)$$

Some manipulation will now be done using the above equations. The goal is to eliminate pressure and density in favor of entropy, enthalpy, and temperature. The momentum equations are expanded as follows using the chain rule,

$$\rho \frac{\partial u}{\partial t} + u \frac{\partial \rho}{\partial t} + u \frac{\partial \rho u}{\partial x} + \rho u \frac{\partial u}{\partial x} + u \frac{\partial \rho v}{\partial y} + \rho v \frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} \quad (8.5)$$

$$\rho \frac{\partial v}{\partial t} + v \frac{\partial \rho}{\partial t} + v \frac{\partial \rho u}{\partial x} + \rho u \frac{\partial v}{\partial x} + v \frac{\partial \rho v}{\partial y} + \rho v \frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} \quad (8.6)$$

The continuity equation can be used to cancel out some of the terms to reduce the set of equations to the following form,

$$\rho \left[ \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right] = -\frac{\partial p}{\partial x} \quad (8.7)$$

$$\rho \left[ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right] = -\frac{\partial p}{\partial y} \quad (8.8)$$

Pressure and density can now be eliminated with Gibbs equation.

$$\left[ \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right] = T \frac{\partial s}{\partial x} - \frac{\partial h}{\partial x} \quad (8.9)$$

$$\left[ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right] = T \frac{\partial s}{\partial y} - \frac{\partial h}{\partial y} \quad (8.10)$$

One further substitution is made using total enthalpy. This will yield the final form of the momentum equations.

$$\frac{\partial u}{\partial t} + v \left[ \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} \right] = T \frac{\partial s}{\partial x} - \frac{\partial h}{\partial x} \quad (8.11)$$

$$\frac{\partial v}{\partial t} + u \left[ \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right] = T \frac{\partial s}{\partial y} - \frac{\partial h}{\partial y} \quad (8.12)$$

The vorticity can now be introduced from equations (8.11) and (8.12). The vorticity,  $\Omega$ , appears in the brackets. At this stage, the equations can be written in vector form using the fact that vorticity appears explicitly in the equations.

$$T \nabla s + \vec{u} \times \vec{\Omega} = \nabla h_o + \frac{\partial \vec{u}}{\partial t} \quad (8.13)$$

Equation (8.13) is known as the unsteady Crocco's equation. If the isentropic assumption is now made ( $s = \text{constant}$ ), a potential function  $\phi$  can be defined. For completeness, the potential function will be defined for three dimensional flows as follows,

$$\frac{\partial \phi}{\partial x} = u \quad \frac{\partial \phi}{\partial y} = v \quad \frac{\partial \phi}{\partial z} = w \quad (8.14)$$

and

$$\frac{\partial \phi}{\partial t} = -h_o = - \left[ h + \frac{u^2 + v^2 + w^2}{2} \right] \quad (8.15)$$

Due to the isentropic assumption, (8.14) and (8.15) are only valid for irrotational flows. The potential function describes the velocity flow field. The velocity vector in three dimensions can be written in terms of the potential as,

$$\vec{V} = \vec{\nabla} \phi = \phi_x \hat{i} + \phi_y \hat{j} + \phi_z \hat{k} \quad (8.16)$$

where the subscripts on  $\phi$  represent partial derivatives with respect to  $x$ ,  $y$ , and  $z$ . The unsteady full potential equation comes directly from substituting equation (8.16) into the continuity equation. The governing equation for potential flow is therefore,

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x} \left( \rho \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left( \rho \frac{\partial \phi}{\partial y} \right) + \frac{\partial}{\partial z} \left( \rho \frac{\partial \phi}{\partial z} \right) = \rho_t + \vec{\nabla} \cdot (\rho \vec{\nabla} \phi) = 0 \quad (8.17)$$

For an incompressible steady flow, density is constant and drops out of the equation. When this happens, the above potential equation is referred to as Laplace's equation ( $\phi_{xx} + \phi_{yy} + \phi_{zz} = 0$ ). Laplace's equation is a second order linear equation in  $\phi$  and applies only to incompressible flowfields. For the compressible form, it appears that equation (8.17) has two unknowns,  $\rho$  and  $\phi$ . It can be shown that density is a function of  $\phi$  only, giving only one equation with one unknown.

To write density as a function of  $\phi$  only, additional relationships are needed. The first is the isentropic relation for pressure and density,

$$\frac{p}{p_\infty} = \left( \frac{\rho}{\rho_\infty} \right)^\gamma \quad (8.18)$$

where  $\gamma$  is the ratio of specific heats and the subscript  $(\infty)$  denotes some reference state, taken here as the freestream condition. The second relation is the perfect gas equation of state which assumes a thermally perfect gas,

$$p = \rho RT \quad (8.19)$$

A few other relations to be mentioned are for the stagnation, critical, and local speed of sound denoted by  $a_o$ ,  $a_*$ , and  $a$  respectively. More details on these equations can be found in reference [133].

$$a_o^2 = (\gamma - 1) \left[ \frac{a^2}{\gamma - 1} + \frac{u^2 + v^2 + w^2}{2} \right] \quad (8.20)$$

$$a_*^2 = \frac{2(\gamma - 1)}{\gamma + 1} \left[ \frac{a^2}{\gamma - 1} + \frac{u^2 + v^2 + w^2}{2} \right] \quad (8.21)$$

$$a^2 = a_o^2 - \frac{\gamma - 1}{2} (u^2 + v^2 + w^2) \quad (8.22)$$

Combining equations (8.14), (8.15), (8.18), and (8.19) gives the desired relationship between  $\rho$  and  $\phi$ . One final step to be done before giving the density relationship is to choose a nondimensionalization scheme. This is necessary in order to have some reference state for the isentropic relations. Two reference conditions are commonly used in the literature, both of which will be covered here for completeness. In the equations that follow, the superscript  $(\prime)$  indicates the nondimensionalized values.

1. Nondimensionalization done by freestream density and velocity:

$$\rho' = \frac{\rho}{\rho_\infty} = \left[ 1 + \frac{\gamma - 1}{2} M_\infty^2 (1 - \phi'_t - q'^2) \right]^{\frac{1}{\gamma - 1}} \quad (8.23)$$

In this equation,  $q$  is the magnitude of the velocity normalized by  $||\vec{V}_\infty|| = q_\infty$  and  $\phi_t$  is normalized by  $q_\infty^2$ . Equations for the speed of sound and pressure are as follows.

$$a'^2 = \frac{a^2}{q_\infty^2} = \frac{\rho'^{(\gamma - 1)}}{M_\infty^2} \quad (8.24)$$

$$p' = \frac{\rho'^\gamma}{\gamma M_\infty^2} \quad (8.25)$$

This nondimensional scheme will be used to compute the density in the present flow solver. Some references where this normalization was done are [134], [135], [101], and [136].

2. Nondimensionalization done by stagnation density and critical speed of sound:

$$\rho' = \frac{\rho}{\rho_s} = \left[1 - \frac{\gamma - 1}{\gamma + 1} q'^2\right]^{\frac{1}{\gamma-1}} = \left[\frac{2a'^2}{\gamma + 1}\right]^{\frac{1}{\gamma-1}} \quad (8.26)$$

The normalization of  $q$  is done using  $a_s$ . As before, equations for the speed of sound and pressure are given below. Several different forms for  $a'$  exist, all of which are equivalent.

$$\begin{aligned} a'^2 = \frac{a^2}{a_s^2} &= (\gamma - 1) \left[ \frac{\gamma + 1}{2(\gamma - 1)} - \frac{q'^2}{2} \right] \\ &= \frac{\gamma + 1}{2} \left[ 1 - \frac{\gamma - 1}{\gamma + 1} q'^2 \right] \\ &= \frac{\gamma p'}{\rho'} \end{aligned} \quad (8.27)$$

And finally, the equation for pressure.

$$p' = \frac{(\gamma + 1)\rho'^\gamma}{2\gamma} \quad (8.28)$$

Some references where this normalization can be found are [109], [137], [110], and [138].

The mathematical classification of the steady, two dimensional full potential equation is elliptic for subsonic flows, parabolic for sonic flow, and hyperbolic for supersonic flows. This can be most easily seen from the nonconservative form of the equation. In two dimensions, the equation for steady flow is given by

$$(a^2 - \phi_x^2)\phi_{xx} - 2\phi_x\phi_y\phi_{xy} + (a^2 - \phi_y^2)\phi_{yy} = 0 \quad (8.29)$$

where  $a$  is the local speed of sound. This is a quasi-linear, second order partial differential equation of the form

$$A\phi_{xx} + B\phi_{xy} + C\phi_{yy} = D$$

where the two characteristic directions are given by

$$\left(\frac{dy}{dx}\right)_{1,2} = \frac{B \pm \sqrt{B^2 - 4AC}}{2A}$$

Following this formula, the characteristic directions for equation (8.29) are

$$\left(\frac{dy}{dx}\right)_{1,2} = \frac{-\phi_x \phi_y \pm \sqrt{a^2(q^2 - a^2)}}{a^2 - \phi_x^2} \quad (8.30)$$

The term under the square root (the discriminant) will determine the nature of the equation. As can be seen, when the local velocity is subsonic ( $q$  less than  $a$ ), the characteristics are unique and real, giving rise to the elliptic nature of the flow. As the velocity becomes greater than the sonic condition, the sign of the discriminant changes to negative, resulting in imaginary values which indicate a hyperbolic flow. And if the velocity is at the sonic state, the flow is parabolic.

### 8.3 Finite Volume Formulation

To solve the full potential equation using the finite volume formulation, it is convenient to start with the integral form of the continuity equation. This is written below for steady flows.

$$\iint_S \rho \vec{V} \cdot \hat{n} dS = 0 \quad (8.31)$$

The unstructured mesh will consist of individual control volumes or cells for which the above integral equation will be applied. A cell centered approach is taken such that the unknowns are solved for at the cell centers. The alternate approach would be a node based approach where the unknowns are located at the nodal points. The integral equation can be recast for each control volume as,

$$\sum_{j=1}^{\#faces} (\rho \vec{V} \cdot \hat{n}_j) \Delta A_j = R_{cell} = 0 \quad (8.32)$$

where the surface integral over each face is replaced by a pointwise evaluation of the mass flux, and  $R_{cell}$  represents the residual for the cell.

## 8.4 Flux Calculation

To form the residual defined by (8.32), the flux on each cell face must be computed. The flux at an arbitrary face consists of the velocity vector and the density. Both the velocity and the density are a function of  $\phi$  only. From equation (8.23), the density on a face can be calculated directly once the velocity vector is known at the face. The main obstacle in forming the flux is therefore calculating the velocity vector at each cell face.

From equation (8.16), the velocity field can be found by calculating the gradients of  $\phi$  in the  $x$ ,  $y$ , and  $z$  directions. For a structured grid, this calculation is straight forward due to the logical ordering of the cells. For an unstructured grid, it is less clear how to formulate and compute the gradients of  $\phi$ . Several algorithms were tested out until a satisfactory method of computing the gradients was found. Several of these methods and their problems will be discussed first. Following this discussion will be the chosen method, explained in more detail.

A quick note should be made about selecting a gradient method. The criteria for selecting the best gradient method consists mainly of how accurate the method is in computing the gradients and how generally applicable the algorithm can be made. An initial test for judging how accurate the gradients are is to see if a linear distribution of  $\phi$  can be reproduced as a constant value for the gradient. In other words, can the method reproduce a freestream velocity applied to an arbitrary mesh. If not, then the accuracy of the method will not be acceptable. The second criteria deals with how general the method is. It is desirable to be able to attain solutions on grids consisting of either hexahedrals, prisms, tetrahedrals, or a mixture of the different cell types. The gradient method should therefore be easily adaptable to cells with an arbitrary number of faces.

### 8.4.1 Method 1: Gauss Divergence

The first method attempted for calculating the gradient of the potential function is based directly on the Gauss Divergence theorem. This theorem can be expressed as

[139],

$$\iint_{\mathcal{V}} \nabla \phi d\mathcal{V} = \oint_A \phi \hat{n} dA \quad (8.33)$$

When this equation is applied to each control cell, a gradient approximation is given at the cell center. An example of the cell centered value of  $\phi_x$  for an arbitrary cell is given below,

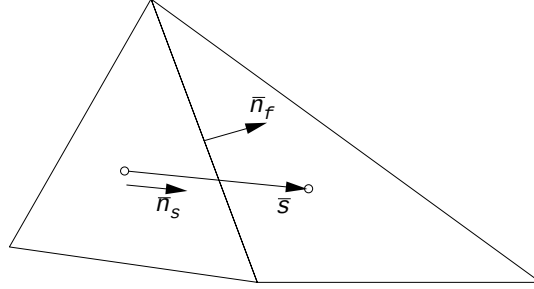
$$\phi_x = \frac{1}{\mathcal{V}} \sum_{j=1}^{\#faces} (\phi_f n_x A_f)_j \quad (8.34)$$

In this calculation,  $\phi_f$  is the value of  $\phi$  at each face and is found by averaging the two surrounding adjacent cell center values to the face based upon the distances from each cell center to the face. Once the gradients are found for each cell center, averaging is done to calculate the face gradients based upon the same method that was used to find  $\phi_f$ . It should be noted that this method will only return the exact gradient solution to a linear distribution when the segment connecting the cell centroids bisects the cell face. For both quadrilateral and triangular faces, this method will generally not reproduce a linear distribution. The solutions from this method were found to be very spurious instead of smooth. It was also observed that this method gave better results for quadrilaterals faces than for triangles.

### 8.4.2 Method 2: Modified Gauss Divergence

Method 1 above was modified to give the second method for computing the gradients [140]. The Gauss Divergence theorem was again used to find the gradients at the cell face, but the values were then modified to improve the accuracy. First the gradients at the cell face were found using the first method, which will be denoted by  $\vec{V}_I = \phi_x \hat{i} + \phi_y \hat{j} + \phi_z \hat{k}$ . A vector,  $\vec{s}$ , was then formed that connected the center of the two cells adjacent to the cell face (see figure 8.2). The gradient in the direction of the vector was found by differencing the value of  $\phi$  at the cell centers. This new gradient is referred to as  $\phi_s$ . If  $\hat{n}_s$  is the unit vector in the direction of  $\vec{s}$ , then the gradient in that direction can be subtracted out of  $\vec{V}_I$  by the following formula,

$$\vec{V}_{II} = \vec{V}_I - (\vec{V}_I \cdot \hat{n}_s) \hat{n}_s \quad (8.35)$$

Figure 8.2: Illustration of vectors  $\vec{s}$  and  $\hat{n}_s$ .

The new gradient,  $\phi_s$ , can then be added to  $\vec{V}_{II}$  in a similar manner.

$$\vec{V}_f = \vec{V}_{II} + (\phi_s)\hat{n}_s \quad (8.36)$$

The new velocity vector,  $\vec{V}_f$ , represents an improved approximation to the gradient at the cell face. This second method improves the accuracy over the first method, but a linear distribution is still not attained for an arbitrary mesh. The solution with this method behaved in a similar manner to the first method, and was not found to be satisfactory.

### 8.4.3 Method 3: K-exact

The next method to be discussed comes from K-exact reconstruction developed by Barth [141]. The K-exact method is used in both structured and unstructured flow solvers to attain higher order polynomial curve fits through cell centers. From the polynomial curve fit, variables can be reconstructed to cell faces. Since a polynomial curve is formed, it is quite easy to compute the gradient at a cell face as well. The first type of K-exact reconstruction discussed here requires a set number of cells to be selected for a given gradient calculation. This set of cells forms what is called a stencil for the reconstruction at a cell face. For a first order polynomial in three dimensions, the stencil will need to contain four cells. An issue with the K-exact method is the selection of the cells which form the stencil. The method was implemented for a first order polynomial reconstruction, which results in second order gradient approximations. The cells in the stencil were chosen to minimize the distance between the cell

face center and the centroid of the combined cells in the stencil. The first two cells in the stencil were always the two cells adjacent to the cell face. The remaining cell or cells in the stencil were then selected based upon the above criteria. Care must be taken to ensure that the cell centers in the stencil are not co-linear. Boundary cells were allowed to be part of the stencil as well. The gradients calculated using the K-exact method could reproduce a linear distribution but the method was still not satisfactory. The requirement that a fixed number of cells be selected takes away from the desired generality of the scheme. The number of cells in the stencil varies depending on whether the grid is 2-D or 3-D. The advantage of K-exact reconstruction is its ability to provide the desired accuracy. Because of this, changes were made to the formulation to produce the desired gradient calculation method, which is discussed next.

#### 8.4.4 Method 4: K-exact Least Squares

The method that was found to give the best gradient approximation will now be discussed. This method has the ability to reconstruct a linear distribution and works equally well for triangles as for quadrilaterals. This scheme is based on the K-exact reconstruction method mentioned above. A more detailed look into the above K-exact method is now given.

In the K-exact method, a  $k$  degree polynomial is fitted through a selected number of cells. The information needed to construct the polynomial comes from the cell averaged values located at the geometric center of the control volumes. From this polynomial, values at any location can be reconstructed along with its gradients. A polynomial  $P$  of degree  $k$  can be written as

$$P^k(x, y, z) = \sum_{i=0}^k \sum_{j=0}^{k-i} \sum_{l=0}^{k-i-j} C_{i,j,l} x^i y^j z^l \quad (8.37)$$

where  $C_{i,j,l}$  are the reconstruction coefficients. The number of unknown coefficients depend upon the degree  $k$  of the polynomial. In two dimensions, the number of unknowns will be  $(k+1)(k+2)/2$  and in three dimensions the number of unknowns will be  $(k+1)(k+2)(k+3)/6$ . This number corresponds to the number of cells that

will be needed to do the reconstruction. Therefore, for a fixed degree polynomial with unique coefficients, there will have to be a fixed number of cells for the reconstruction stencil. The selection of these cells may not be an easy task. This restriction is removed to allow for more than the minimum number of cells to be included in the stencil, making selection of the stencil more simplified. But with an increase in the number of cells in the stencil, there will no longer be a unique polynomial that will fit through all the cell center values. Attention is now turned to increasing the number of cells in the stencil.

When doing K-exact reconstruction, it is necessary that the reconstruction polynomial, when integrated over a cell in the stencil, recovers the cell average defined by

$$\bar{\phi} = \frac{1}{\mathcal{V}} \iiint_{\mathcal{V}} \phi(x, y, z) d\mathcal{V} \quad (8.38)$$

Using this equation and equation (8.37) above, a constraint equation can be formed. If this constraint equation is satisfied, the reconstruction polynomial will recover the cell average value when integrated over the cell. The constraint equation is given by,

$$\bar{\phi} = \sum_{i=0}^k \sum_{j=0}^{k-i} \sum_{l=0}^{k-(i+j)} \frac{1}{\mathcal{V}} C_{i,j,l} \iiint_{\mathcal{V}} x^i y^j z^l d\mathcal{V} \quad (8.39)$$

For the method where the stencil contains a set number of cells, an equation is formulated for each cell in the stencil by applying the Mean Value Theorem to the constraint equation. When this is done, the number of equations will equal the number of unknowns. This will not be the case when the number of cells are allowed to increase, but the number of unknowns stay the same.

To explain how the new K-exact method works, an example is given below. Consider a  $k = 1$  degree polynomial in two dimensions. There will be three unknown coefficients to solve for. Figure 8.3 shows a portion of an unstructured 2-D mesh made up of triangles. The cells in this figure make up the stencil used to compute the polynomial associated with the face common to cells 1 and 2. From this figure, the system will consist of six equations, but with only three unknowns to solve for. A weighted least squares polynomial fit is therefore done to determine the polynomial coefficients. Weights are placed on cells 1 and 2 such that the constraint equation

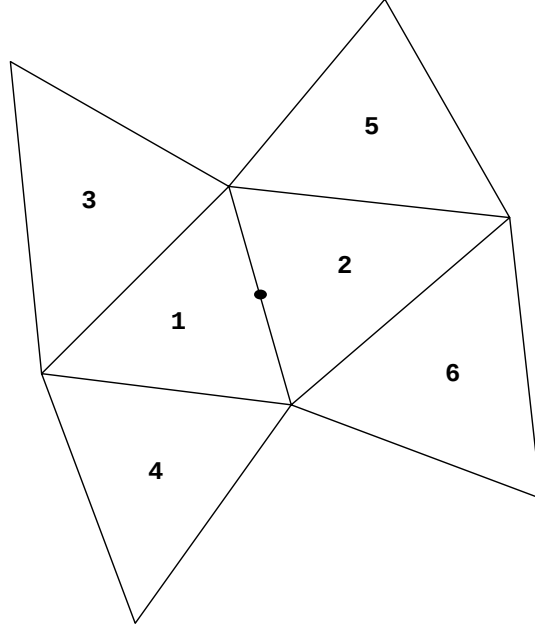


Figure 8.3: Stencil selection for computing gradients at a cell face.

is closely satisfied for the two cells surrounding the face. The polynomial in this example can be written as,

$$P^1(x, y) = C_{0,0} + C_{1,0}x + C_{0,1}y \quad (8.40)$$

where the gradients at the cell face are actually the coefficients  $C_{1,0}$  and  $C_{0,1}$  in the  $x$  and  $y$  directions respectively. The constraint equation is closely satisfied by only two of the cells in the stencil, while the remaining cells minimize the error associated with not satisfying it. In the code, a  $k = 1$  degree polynomial is used which gives second order accurate approximations to the gradients.

Continuing on with the example, the constraint equation is applied to each of the six cells in the stencil. With the use of the Mean Value Theorem, the six constraint equations in this problem are,

$$\begin{aligned} \bar{\phi}_1 &= C_{0,0} + C_{1,0}\bar{x}_1 + C_{0,1}\bar{y}_1 \\ \bar{\phi}_2 &= C_{0,0} + C_{1,0}\bar{x}_2 + C_{0,1}\bar{y}_2 \\ \bar{\phi}_3 &= C_{0,0} + C_{1,0}\bar{x}_3 + C_{0,1}\bar{y}_3 \end{aligned}$$

$$\begin{aligned}
\bar{\phi}_4 &= C_{0,0} + C_{1,0}\bar{x}_4 + C_{0,1}\bar{y}_4 \\
\bar{\phi}_5 &= C_{0,0} + C_{1,0}\bar{x}_5 + C_{0,1}\bar{y}_5 \\
\bar{\phi}_6 &= C_{0,0} + C_{1,0}\bar{x}_6 + C_{0,1}\bar{y}_6
\end{aligned}$$

If only three constraint equations existed, the coefficients in the reconstruction polynomial could be found directly by solving the equations for the unknowns. Instead, there are more equations than unknowns which indicates that a unique solution will not occur. Since cells 1 and 2 in Figure 8.3 are closest to the cell face in question, it is desirable that the constraint equations that correspond to these two cells be satisfied. This will give the necessary condition to give a unique solution, one which minimizes the error associated with satisfying the constraint equation. A weighted least squares fit is therefore done to solve for the three unknown coefficients. To begin the least squares process, the deviation or error associated with each cell based upon the constraint equation is defined as,

$$e_i = C_{0,0} + C_{1,0}\bar{x}_i + C_{0,1}\bar{y}_i - \bar{\phi}_i \quad (8.41)$$

where  $i$  indicates a specific cell in the stencil. A function is formed that represents the sum of the squares of the deviations. Weights are placed on each cell such that the sum of squares function takes the following form,

$$\begin{aligned}
F(C_{0,0}, C_{1,0}, C_{0,1}) &= \sum_{i=1}^m r_i (e_i)^2 \\
&= \sum_{i=1}^m r_i (C_{0,0} + C_{1,0}\bar{x}_i + C_{0,1}\bar{y}_i - \bar{\phi}_i)^2
\end{aligned} \quad (8.42)$$

The index  $m$  indicates the number of cells used in the reconstruction stencil and  $r_i$  is the weight factor corresponding to cell  $i$ . With given weights on each cell, a good approximation to the coefficients will occur when the function  $F$  is minimized. This is the basis behind the least squares method. The minimum can be attained by taking partial derivatives of  $F$  with respect to the unknowns. This is given below.

$$\begin{aligned}
\frac{\partial F}{\partial C_{0,0}} &= 2 \sum_{i=1}^m r_i (C_{0,0} + C_{1,0}\bar{x}_i + C_{0,1}\bar{y}_i - \bar{\phi}_i) = 0 \\
\frac{\partial F}{\partial C_{1,0}} &= 2 \sum_{i=1}^m r_i (C_{1,0} + C_{1,0}\bar{x}_i + C_{0,1}\bar{y}_i - \bar{\phi}_i) \bar{x}_i = 0
\end{aligned}$$

$$\frac{\partial F}{\partial C_{0,1}} = 2 \sum_{i=1}^m r_i (C_{0,1} + C_{1,0} \bar{x}_i + C_{0,1} \bar{y}_i - \bar{\phi}_i) \bar{y}_i = 0 \quad (8.43)$$

The problem has now been reduced down to three equations with three unknowns. The system can be represented in matrix form as,

$$\begin{bmatrix} \sum r_i & \sum r_i \bar{x}_i & \sum r_i \bar{y}_i \\ \sum r_i \bar{x}_i & \sum r_i \bar{x}_i^2 & \sum r_i \bar{x}_i \bar{y}_i \\ \sum r_i \bar{y}_i & \sum r_i \bar{x}_i \bar{y}_i & \sum r_i \bar{y}_i^2 \end{bmatrix} \begin{pmatrix} C_{0,0} \\ C_{1,0} \\ C_{0,1} \end{pmatrix} = \begin{bmatrix} \sum r_i \bar{\phi}_i \\ \sum r_i \bar{x}_i \bar{\phi}_i \\ \sum r_i \bar{y}_i \bar{\phi}_i \end{bmatrix} \quad (8.44)$$

where each summation is from 1 to  $m$ , the total number of cells in the stencil. The components of the matrix are made up only of cell center geometric values and weights. This means that the matrix can be computed once and then stored. Instead of storing the matrix itself, it is more efficient to store its inverse. If the matrix is denoted by  $A$ , then its inverse can be expressed as,

$$[A]^{-1} = \begin{bmatrix} a'_{11} & a'_{12} & a'_{13} \\ a'_{21} & a'_{22} & a'_{23} \\ a'_{31} & a'_{32} & a'_{33} \end{bmatrix} \quad (8.45)$$

With the matrix  $[A]^{-1}$  stored, the coefficients of the polynomial can be computed as,

$$C_{0,0} = a'_{11} \sum_{i=1}^m r_i \bar{\phi}_i + a'_{12} \sum_{i=1}^m r_i \bar{x}_i \bar{\phi}_i + a'_{13} \sum_{i=1}^m r_i \bar{y}_i \bar{\phi}_i \quad (8.46)$$

$$C_{1,0} = a'_{21} \sum_{i=1}^m r_i \bar{\phi}_i + a'_{22} \sum_{i=1}^m r_i \bar{x}_i \bar{\phi}_i + a'_{23} \sum_{i=1}^m r_i \bar{y}_i \bar{\phi}_i \quad (8.47)$$

$$C_{0,1} = a'_{31} \sum_{i=1}^m r_i \bar{\phi}_i + a'_{32} \sum_{i=1}^m r_i \bar{x}_i \bar{\phi}_i + a'_{33} \sum_{i=1}^m r_i \bar{y}_i \bar{\phi}_i \quad (8.48)$$

For this 2-D example, only  $C_{1,0}$  and  $C_{0,1}$  would be needed since they correspond to velocities  $u$  and  $v$  respectively.

The extension into three dimensions is straight forward. The matrix system represented by (8.44) would have four unknowns instead of three. The polynomial for a second order reconstruction appears as,

$$P^1(x, y) = C_{0,0,0} + C_{1,0,0}x + C_{0,1,0}y + C_{0,0,1}z \quad (8.49)$$

The terms in the matrix can be derived in a similar manner following the 2-D case and is presented below.

$$\begin{bmatrix} \sum r_i & \sum r_i \bar{x}_i & \sum r_i \bar{y}_i & \sum r_i \bar{z}_i \\ \sum r_i \bar{x}_i & \sum r_i \bar{x}_i^2 & \sum r_i \bar{x}_i \bar{y}_i & \sum r_i \bar{x}_i \bar{z}_i \\ \sum r_i \bar{y}_i & \sum r_i \bar{x}_i \bar{y}_i & \sum r_i \bar{y}_i^2 & \sum r_i \bar{y}_i \bar{z}_i \\ \sum r_i \bar{z}_i & \sum r_i \bar{x}_i \bar{z}_i & \sum r_i \bar{y}_i \bar{z}_i & \sum r_i \bar{z}_i^2 \end{bmatrix} \begin{pmatrix} C_{0,0,0} \\ C_{1,0,0} \\ C_{0,1,0} \\ C_{0,0,1} \end{pmatrix} = \begin{bmatrix} \sum r_i \bar{\phi}_i \\ \sum r_i \bar{x}_i \bar{\phi}_i \\ \sum r_i \bar{y}_i \bar{\phi}_i \\ \sum r_i \bar{z}_i \bar{\phi}_i \end{bmatrix} \quad (8.50)$$

The K-exact method can also be used to compute the gradients at the cell center. This is done by choosing all the neighboring cells, as well as the center cell as the stencil. No weights are necessary in this case. Once the stencil is chosen, the procedure to compute the gradients is the same as that for the cell face gradients presented above.

## 8.5 Artificial Viscosity

The characteristic type of equation (8.17) changes when the flow becomes supersonic. The equation is elliptic in subsonic flows and hyperbolic for supersonic flows. This switching in equation type changes the diffusive character of the elliptic flow field to the propagation-dominated behavior associated with the hyperbolic equation. Because of this change, methods used exclusively for subsonic flows will not work for transonic or supersonic flow regions. The discretization in the supersonic regions must take into account the existence of characteristics in the flow field. The most common modification to take into account this change in the equation type is through the addition of an artificial viscosity term appearing in the density. The two most popular methods in which this can be done is density biasing and flux biasing. The first method, density biasing, has been implemented into the code and will be described here.

The potential equation takes on the same form as before, except the density is now replaced with  $\tilde{\rho}$ , the artificial density. The governing equation for steady flows is rewritten as,

$$\frac{\partial}{\partial x}(\tilde{\rho}\phi_x) + \frac{\partial}{\partial y}(\tilde{\rho}\phi_y) + \frac{\partial}{\partial z}(\tilde{\rho}\phi_z) = 0 \quad (8.51)$$

where for the density biasing scheme,  $\tilde{\rho}$  is expressed as

$$\tilde{\rho} = \rho - \mu \left[ \frac{u}{q} \rho_x \Delta x + \frac{v}{q} \rho_y \Delta y + \frac{w}{q} \rho_z \Delta z \right] \quad (8.52)$$

The parameter  $\mu$  is a switching function. If  $\mu$  is zero, the equation reduces to the subsonic potential formulation and if  $\mu$  takes on some finite value, the density starts taking into account the upwind flow conditions. The parameter  $\mu$  is computed by the following formula,

$$\mu = \max \left[ 0, \left( 1 - \frac{M_c^2}{M^2} \right) \right] C M^2 \quad (8.53)$$

where  $M$  is the local Mach number,  $M_c$  is a cut-off Mach number, and  $C$  is a constant that aids in the stability of the code by increasing the amount of artificial viscosity. The value of  $C$  will normally be between 1 and 2 and that of  $M_c$  around 0.98. Depending on how complicated the flow is, values of  $C$  and  $M_c$  may need to be adjusted to maintain stability. When the switching function is non-zero, the local solution is no longer second order accurate, but approaches first order accuracy.

Two different methods have been implemented to compute the density gradients required in (8.52). Both of these methods are based on the K-exact method used to compute the potential gradients. The first method computes the density gradient at the cell center using the density from that cell center and the surrounding cell centers. This requires the velocity to be computed at the cell centers as well, adding extra computational work. The second method also computes the gradient at the cell center, but uses the density from the surrounding cell faces. This second method is favored over the first since the density is already computed at the cell faces for the flux.

The steps taken to add the artificial density term are straight forward. For a given face, the velocity normal to the face is computed, indicating an upwind and downwind direction for that face. Figure 8.4 may help to visualize the situation. The cell center gradient that lies in the upwind direction is used in (8.52). The Mach number is evaluated at the center of the upwind cell. This helps prevent spikes or overshoots at a shock. The values of  $\Delta x$ ,  $\Delta y$ , and  $\Delta z$  are twice the distance from the cell face to the upwind cell center. This provides a general calculation procedure for these parameters regardless of face or cell type.

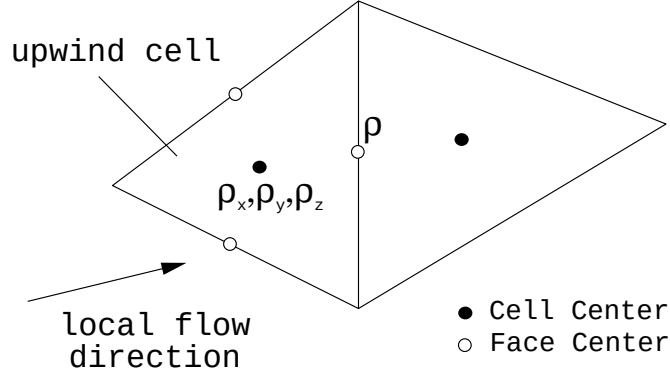


Figure 8.4: Upwind cell and density gradients used in density biasing.

## 8.6 Explicit Algorithms

An iteration scheme must now be employed to drive the residual, defined in equation (8.32), to zero. An explicit algorithm is easy to implement, and will be discussed before moving on to implicit methods. The major drawback with explicit algorithms is their convergence rate. Explicit methods can perform an iteration on the solution much faster than implicit methods, but require more iterations to converge. In practice, most all full potential codes are converged using implicit algorithms. Several explicit methods were implemented into the flow solver. These methods will be discussed below for completeness.

The first explicit method implemented into the unstructured code was a 2 level explicit algorithm. If  $\beta$  represents a convergence parameter, then the algorithm is expressed as,

$$\Delta^n \phi_{cell} = \frac{1}{\beta} R_{cell}^n \quad (8.54)$$

In this and every scheme that follows,  $n$  is a parameter indicating the iteration level where  $\Delta^n \phi = \phi^{n+1} - \phi^n$  and  $\phi^{n+1}$  will always represent the most recent approximation to the solution. The new value for the potential is updated by  $\phi^{n+1} = \phi^n + \Delta^n \phi$ .

The other explicit method that was verified is only slightly more complex and is called a 3 level explicit scheme. The general form of this scheme is,

$$\alpha \tilde{\rho} (\Delta^n \phi - \Delta^{n-1} \phi) + \beta \tilde{\rho} \Delta^n \phi = R^n \quad (8.55)$$

where now two convergence parameters exist,  $\alpha$  and  $\beta$ . The 3 level scheme uses information from the previous iteration and therefore provides more information to update the solution. The 3 level scheme will normally converge faster than the 2 level scheme.

## 8.7 Implicit Algorithms

The implicit schemes presented here begin with the following residual expression.

$$R^{n+1} = 0 \quad (8.56)$$

For most all potential flow cases the residual equation is non-linear in  $\phi$ . Only if density is constant will the potential equation reduce down to Laplaces equation, which is linear. Due to the non-linearity of (8.56), a Newton linearization is performed. The result is a linear system,

$$\frac{\partial R}{\partial \phi} \Delta^n \phi = -R^n \quad (8.57)$$

The first term in equation (8.57) is referred to as the Jacobian. The Jacobian matrix is only a function of the grid geometry and the potential function at level  $n$ . For discussion, the linear system will be written as  $Ax = b$ , where  $A$  is the Jacobian matrix,  $x$  the unknown vector, and  $b$  the right hand side vector. The matrix  $A$  is sparse due to the lack of logical ordering of the cells in the mesh. This is in contrast to a structured code, where the matrix  $A$  is usually penta-diagonal. Iterative schemes have been devised to solve (8.57) which take advantage of the structure of  $A$ . Due to the sparse matrix resulting from the unstructured formulation, different alternatives in solving the linear system must be considered. These methods will be discussed below. Note that the solution to (8.57) does not provide the final solution to the problem, but only serves as an approximation to be used in the next step of the iteration process. With each solution to the linear system, a new residual and Jacobian is formed until the nonlinear problem is converged.

Direct solvers are not considered in solving (8.57) for two reasons. For most problems, the linear system is rather large resulting in enormous memory requirements for the direct solvers. Secondly, since the linear system is only an approximation to

the overall nonlinear problem, an approximate solution will suffice. The better the solution is to (8.57), the faster the rate of convergence will be. So the choice of which linear solver to implement will impact the overall number of iterations needed for convergence. Three iterative schemes for solving linear systems will now be discussed.

### 8.7.1 Point Jacobi

The first linear solver is the point Jacobi method. For illustration purposes, it is convenient to decompose  $A$  according to

$$A = L + D + U$$

where  $L$  consists of all the elements below the diagonal,  $D$  is a diagonal matrix, and  $U$  consists of all the elements above the diagonal. The point Jacobi algorithm can then be given as

$$\Delta^n \phi^{(k)} = D^{-1}[-R^n - L\Delta \phi^{(k-1)} - U\Delta \phi^{(k-1)}] \quad (8.58)$$

This algorithm is an iterative scheme itself. Iterations are represented and done in  $k$  while the iteration level  $n$  is held constant. Iterations in  $k$  are usually performed until the solution is considered to be a good approximation to the exact solution. Only the diagonal matrix is inverted for this method, while the lower and upper matrices are lagged in  $k$  to avoid any further matrix inversions. Due to this, the point Jacobi scheme is simple to implement and requires no additional memory requirements. The major drawback of this scheme is that it requires many iterations in  $k$  to produce a really good approximate solution to the linear system.

### 8.7.2 GMRES

The Generalized Minimum Residual (GMRES) algorithm was developed by Saad and Schultz [142] and is a conjugate gradient method for non-symmetric matrices. Other examples of conjugate gradient methods that have gained popularity are GCR [143] and ORTHODIR [144]. GMRES is theoretically equivalent to these other methods, but is less costly both in terms of storage and arithmetic [145]. Another advantage of

GMRES is that the solution process will not break down unless the exact solution is found. Some examples of GMRES in CFD can be found in Wigton [146] for structured grids, Barth [147] for unstructured meshes, and Hixon [148] for unsteady flows.

The idea behind GMRES is that it constructs  $k$  orthogonal and orthonormal search directions. The system of vectors that are formed are part of the Krylov subspace,  $K_m$ . Using the Krylov subspace, a least squares problem is then solved to minimize the norm of the residual vector. In the present case, the residual vector is defined by

$$r_k = R^n + A\Delta^n\phi$$

The number of search directions will correspond to how well the residual vector is minimized. If the number of search directions is equal to the number of unknowns in the system, the exact solution will be computed. The value of  $k$  will need to be chosen for each problem. The main factor in determining  $k$  is storage requirements. For a system with  $N$  unknowns, the needed storage for the GMRES solver is roughly  $(k+4)N$  [146]. Since there will be limitations on  $k$ , which in turn places a limitation on the solution, inner iterations using the GMRES algorithm are normally done. This type of algorithm is referred to as GMRES( $m$ ) where  $m$  is the number of inner iterations to perform. A new solution is formed for each GMRES application. The new solution is then used as a starting point for the next iteration until  $m$  iterations are performed. The value of  $m$  can be set ahead of time or determined by monitoring the residual of the linear system.

The GMRES(m) procedure is now given below which solves the linear system given by  $Ax = b$ .

1. Choose vector  $x_o$  as an initial guess to the solution. Possible choices for  $x_o$  are  $x_o = \vec{0}$  or  $x_o = \Delta^n\phi$ , the solution from the previous iteration.
2. Compute the residual vector;

$$r_o = b - Ax_o$$

3. Compute the  $l_2$  norm and construct the first Krylov subspace vector;

$$\beta = \|r_o\|_2 \qquad v_1 = \frac{r_o}{\beta}$$

4. Iterate to compute the remaining subspace vectors. This is done by forming  $Av_j$  and orthogonalizing it against the previous subspace vectors.

For  $j = 1, 2, \dots, k$

$$\begin{aligned}
 w &= Av_j \\
 h_{i,j} &= \langle w, v_i \rangle \quad \text{for } i = 1, 2, \dots, j \\
 v_{j+1} &= w - \sum_{i=1}^j h_{i,j} v_i \\
 h_{j+1,j} &= \|v_{j+1}\|_2 \\
 v_{j+1} &= \frac{v_{j+1}}{h_{j+1,j}}
 \end{aligned}$$

5. An upper Hessenberg matrix,  $H_k$ , is defined such that the nonzero entries are the coefficients  $h_{i,j}$  where  $(1 \leq i \leq j+1)$ ,  $(1 \leq j \leq k)$ . The matrix  $H_k$  will therefore have the dimension  $(k+1) \times k$ . The additional row will have only one nonzero element in the  $(k+1, k)$  position. The set of subspace vectors computed above are represented by  $V_k = [v_1, v_2, \dots, v_k]$ .
6. A vector  $y$  is found that minimizes  $J(y) = \|\beta e_1 - H_k y\|$  where  $e_1$  is the first column of the identity matrix, given by  $e_1 = [1, 0, \dots, 0]^T$ . The determination of  $y$  is a least squares problem. More details are given in Appendix C.
7. Next the approximate solution is formed.

$$x_1 = x_o + V_k y$$

8. A stopping criteria can now be tested. One parameter which can be used to monitor convergence is  $\|\beta e_1 - H_k y\|$ . When this value reaches a certain tolerance level set by the user, the GMRES(m) algorithm is stopped and the solution to the linear system is given by  $x_1$ . If more iterations are to be performed,  $x_1$  becomes the initial guess in (1) and the process is repeated.

This algorithm performs well for sparse matrix systems since the inversion of  $A$  is never required, but only the product of  $A$  with vectors  $v_j$  and  $x_o$ .

The speed at which GMRES is able to attain a good approximation to the solution will depend heavily on the eigenvalues of the system. The more clustered together the eigenvalues are, the faster GMRES will converge. A preconditioner is therefore used to give a better clustering of the eigenvalues in the linear system. The idea behind preconditioning is to replace a given problem with another equivalent problem which has the same solution. The equivalent problem will have a more favorable distribution of eigenvalues which will in turn improve convergence.

There are a variety of ways to invoke a preconditioner into the solution process. A discussion on GMRES acceleration using preconditioning for general CFD codes is given in [146]. The approach taken here will be to cluster the eigenvalues around unity. This can be accomplished by using the inverse of the linear operator. For example, if a linear problem is given by

$$Ax - b = 0 \quad (8.59)$$

then the matrix  $A$  is considered the linear operator of the system. An equivalent problem to the one above is

$$A^{-1}(Ax - b) = 0 \quad (8.60)$$

These two systems will have the same solution, but the eigenvalues of the latter will allow GMRES to converge faster. But note, the inverse of  $A$  cannot be found very easily. If so, then there would be no need for an iterative solver like GMRES or point Jacobi. But if  $P$  is an approximation to  $A^{-1}$ , then  $P$  will serve as a preconditioner to the system. The optimal choice for a precondition matrix will always be  $A^{-1}$ , but any matrix which clusters the eigenvalues more closely will aid in the convergence process. From here on, the precondition matrix will be denoted by  $P$ , and equation (8.60) will become

$$P(Ax - b) = 0 \quad (8.61)$$

If a precondition matrix  $P$  is found, equation (8.61) can easily be solved instead of (8.59) with only a few modifications to the GMRES algorithm. To use the above GMRES algorithm with preconditioning, steps (2) and (4) must be modified. Step (2) becomes

$$r_o = P(b - Ax_o)$$

and in step (4), the value of  $w$  is changed to

$$w = P(Av_j)$$

The GMRES method requires more storage and longer execution times than the point Jacobi method, but the improved convergence rate makes up for these differences.

## 8.8 Formulation of the Jacobian Matrix

For Laplaces equation, the Jacobian matrix is a function of the grid metrics and can be computed once and stored. When the density is not constant, the Jacobian matrix becomes a function of  $\phi$  in addition to the grid metrics. For subsonic flows, the density has only a small contribution to the Jacobian matrix and will allow the Jacobians to be stored and then updated periodically. When the flow is transonic, the Jacobians may need to be computed more frequently, especially during the shock formation and positioning, in order to take into account the compressibility effects.

To compute the Jacobians analytically, it is necessary to start with the definition of the residual. Recall from equation (8.32) that the residual is simply the summation of the fluxes over each face of the cell. This equation is rewritten below in expanded form,

$$R_{cell} = \sum_{j=1}^{\#faces} \rho(un_x + vn_y + wn_z)\Delta A_j \quad (8.62)$$

If the density is assumed constant during the differentiation, the Jacobian can be expressed as,

$$\frac{\partial R}{\partial \phi} = \sum_{j=1}^{\#faces} \rho \left( \frac{\partial u}{\partial \phi} n_x + \frac{\partial v}{\partial \phi} n_y + \frac{\partial w}{\partial \phi} n_z \right) \Delta A_j \quad (8.63)$$

The Jacobian will have contributions from every cell that is used in computing the fluxes on each face of that cell. For three dimensional problems, this can involve as many as 25 cells for hexahedrals and 17 cells for tetrahedrals. Storage and book keeping of this magnitude is not very practical and efficient. A second approximation is therefore made that only the neighboring cells will have contributions to the Jacobian

matrix for any given cell. This reduces the number of cells involved in a calculation to 7 for hexahedrals and 5 for tetrahedrals. This approximation is justified by recalling that weights are placed on the two adjacent cells surrounding a cell face. These weighted cells will have the dominant contribution to the Jacobians when compared to the remaining cells in the stencil.

Returning now to the analytical Jacobian calculation, the following terms can be expressed using equations (8.47) and (8.48) for a two dimensional case.

$$\frac{\partial u}{\partial \phi_c} = a'_{21}r_c + a'_{22}r_c\bar{x}_c + a'_{23}r_c\bar{y}_c \quad (8.64)$$

The subscript  $c$  indicates the cell for which the Jacobian is being computed for.

For flows with supersonic regions, the Jacobian matrix must have additional terms added to it for stability. When the flow reaches the cut-off Mach number (usually  $M_c = 0.98$ ; see section 8.5), the density is upwinded to take into account the hyperbolic nature of the flow. A similar change is also done to the Jacobian when the flow reaches  $M_c$ . The modification to the Jacobian terms are explained next.

After the basic Jacobian is computed for elliptic flows, each face is looked at by having the local Mach number computed. If the local Mach number is greater than the cut-off Mach number, the upwind and downwind cells are determined. This procedure is very similar to the one described in section 8.5. The absolute value of the main diagonal term in the Jacobian matrix for the downwind cell is increased by

$$\beta \frac{V_{local}}{V_\infty \Delta s} \quad (8.65)$$

while the same term is subtracted from the off-diagonal upwind Jacobian term. In the above expression,  $V_{local}$  and  $V_\infty$  are the local and freestream velocities respectively. The parameter  $\beta$  controls the level of upwinding. The value of  $\Delta s$  is taken to be the distance between cell centers.

## 8.9 Precondition Matrix

The reason for preconditioning was explained in section 8.7.2. It was stated that a linear system given by  $Ax - b = 0$  could be replaced with a preconditioned system,

$P(Ax - b) = 0$ . This new system will have better convergence properties due to the improved clustering of the eigenvalues. Attention will now be turned to computing a precondition matrix,  $P$ .

The matrix  $P$  needs to approximate the inverse of the Jacobian matrix. The difficulty with this is that for unstructured grids, the Jacobian matrix is sparse, making it difficult to get a good approximation. If the grid were structured, the Jacobian matrix would then be banded, having a set number of known diagonals in the matrix. A good approximation to the Jacobian would then be to use the main diagonal and the two off-diagonals closest to it. This results in a tridiagonal matrix which requires only a small fraction of memory to store when compared to the complete matrix. This same concept can be applied to unstructured grids if a reordering of the cells is done such that the cells are clustered around the main diagonal. Methods for reordering the cells will be discussed next. The first method, Cuthill-McKee, orders the cells to give the minimum bandwidth. The second method to be discussed clusters as many cells next to the main diagonal as possible, which will be the preferred method in this case.

The concept and theory behind the Cuthill-McKee method is given in [149] and an example of its application is given in [150]. As stated above, the goal of the method is to order the cells for the smallest bandwidth. The method makes use of level sets,  $S_i$ , which are made up of the cells. A starting cell is selected and becomes the first set,  $S_1$ . This cell is usually the first cell given by the grid generator. All cells that are neighbors to the starting cell make up set  $S_2$ . Set  $S_3$  consists of all the neighboring cells of  $S_2$  that are not in  $S_2$  or  $S_1$ . The process is repeated until all the cells are placed in a set. Each cell will then belong to only one set. Ordering of the cells is done by starting with  $S_1$ , followed by sets  $S_2, S_3, \dots, S_n$ , where  $n$  is the total number of sets. Cells within a set  $S_i$  are ordered by taking those cells that are neighbors of the first cell in  $S_{i-1}$ , followed by the neighbors of the second cell in  $S_{i-1}$  and so on. An example of the Cuthill-McKee ordering is now given. A triangular mesh is shown in Figure 8.5 consisting of 108 cells. The location of the Jacobian terms for the ordering that results from the grid generator is given in Figure 8.6. With the Cuthill-McKee ordering, the matrix has a distinct bandwidth which is shown in Figure 8.7.

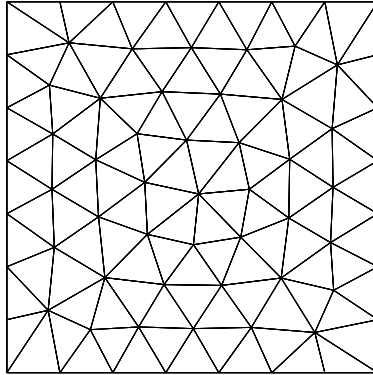


Figure 8.5: An unstructured triangular grid consisting of 108 cells.

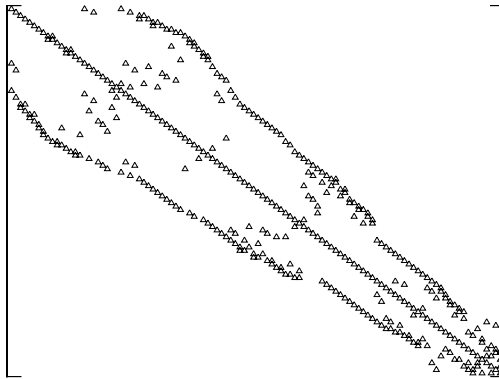


Figure 8.6: Ordering of the Jacobian matrix based upon the grid generator (general sparse matrix).

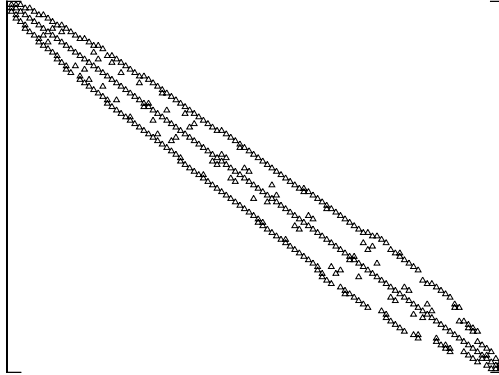


Figure 8.7: Order of the Jacobian matrix based upon the Cuthill-McKee method (minimizes bandwidth).

Even though the bandwidth is much smaller than the original matrix, the Cuthill-McKee ordering fails to cluster many of the cells next to the main diagonal. Instead of using level sets, a reordering method can be done that steps from one cell to the next. This method is performed as follows. A starting cell is chosen as before. The neighbors of this cell are then chosen next in the ordering. Out of those cells, the one with the most neighboring cells that has not yet been ordered becomes the next starting cell. The process continues until all the neighboring cells have already been reordered. When this occurs, a new starting cell is randomly chosen from the remaining cells to be ordered, and the process begins again. The matrix will remain sparse, but most every row will have cells close to the main diagonal. Using the same grid as before, the matrix using this re-ordering scheme is shown in Figure 8.8. Unless otherwise stated, the precondition matrix will use this second ordering method to cluster the cells next to the main diagonal.

The steps taken to perform preconditioning will now be covered. The first step is to perform the reordering procedure described above. Instead of actually reordering the cells in the code, a mapping from the original order to the new ordering will be done. The mapping is stored in a single array with length equal to the total number of cells. The approximate inverse matrix  $P^{-1}$  will be filled based upon the

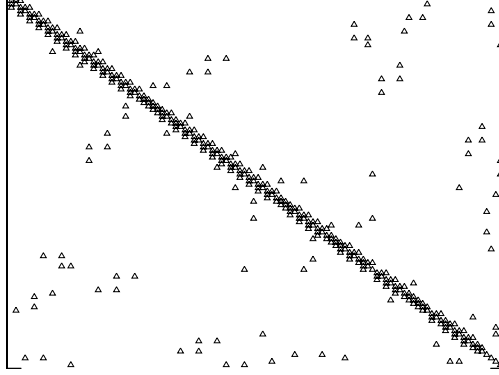


Figure 8.8: Order of the Jacobian matrix in order to cluster terms near the main diagonal.

new ordering. All operations performed on  $P^{-1}$  are done with the new ordering by mapping the original order to the new one. The matrix  $P^{-1}$  is filled and stored in compact matrix form based upon a chosen bandwidth. Since the only operation performed on the precondition matrix  $P$  will be multiplication with a vector, the LU decomposition procedure can be used. The steps taken to perform the multiplication between  $P$  and a vector  $w$  is given below where  $x$  is the solution vector.

- The problem is stated as:  $x = Pw$
- It is rewritten as a linear system such that,  $P^{-1}x = w$
- LU decomposition is performed on  $P^{-1}$  to give  $[L][U]x = w$
- Perform forward substitution:  $[U]x = [L]^{-1}w$
- Perform backward substitution to solve for the unknown:  $x = [U]^{-1}([L]^{-1}w)$

Once  $P^{-1}$  is computed it can be factored into  $[L][U]$  and stored. This will allow the number of arithmetic operations in the precondition process to be reduced since only forward-backward substitution will need to be done. LU decomposition replaces the original vector  $P^{-1}$  with the lower and upper banded matrices so that no extra storage is needed for  $[L][U]$ .

One final issue remains, the number of diagonals to include when approximating the Jacobian matrix  $P^{-1}$ . In other words, a bandwidth must be chosen. A tradeoff is made between storage and convergence. The larger the bandwidth, the better the preconditioner will be, but the storage requirements will increase. The best choice for the bandwidth may therefore be problem dependent. For most problems, a bandwidth between 1 and 5 will be sufficient. In order to handle a variable bandwidth matrix, a banded LU solver was used which stored the matrix in compact form. Only the necessary diagonals are stored, making the algorithm memory efficient.

## 8.10 Kutta Condition and Circulation

With an unstructured grid, implementation of the Kutta condition is not as straight forward as with structured grids. The first step in applying the Kutta condition is determining the circulation around the airfoil or wing. Next a line or plane must be chosen in which to enforce a jump in the potential across. The potential will be discontinuous across this surface, but all other properties, like velocity, will remain continuous.

Figure 8.9 gives an illustration for a 2-D airfoil problem. The circulation value is found from the following:

$$\Gamma_c = \phi_{top,te} - \phi_{bottom,te} \quad (8.66)$$

where  $\phi_{top,te}$  is the potential value at the trailing edge for the top surface and  $\phi_{bottom,te}$  is the potential for the bottom trailing edge point. Since the two values of  $\phi$  are not known at the trailing edge, an extrapolation is done. There are two ways in which this could be accomplished. The first uses the Taylor series expansion. The value of  $\phi$  at the trailing edge is computed from,

$$\phi_{te} = \phi_{face} + \Delta s \frac{\partial \phi_{face}}{\partial s} \quad (8.67)$$

The length  $s$  is the distance from the boundary face to the trailing edge. The partial derivative is simply the velocity tangent to the boundary face. The potential at the trailing edge is therefore estimated once the velocity is calculated on the boundary. The second method uses the K-exact least squares reconstruction, as explained in

section 8.4.4. With this method, the velocity does not have to be found, reducing some of the computational work. Both methods were tested and gave essentially identical results. The second method was chosen to be implemented for all future calculations. It proved to be more efficient to code and was simple in its formulation.

For problems where lift is generated, convergence is slowed due to the implementation of the Kutta condition. The value of  $\Gamma$  is updated each iteration, and has a large impact on the iteration scheme. To help speed up convergence, the value of  $\Gamma$  is over-relaxed by the following formula,

$$\Gamma^{new} = \theta\Gamma^{n+1} + (1 - \theta)\Gamma^n \quad (8.68)$$

The relaxation parameter  $\theta$  is chosen between 1 and 2. Another technique that helps convergence is to freeze the value of  $\Gamma$  once it ceases to change by a certain amount. This is done by monitoring each new update of  $\Gamma$  and calculating the percent change of the new value from the previous value.

$$\%change = \frac{\Gamma^{n+1} - \Gamma^n}{\Gamma^{n+1}} \times 100$$

When the % change reaches 0.02%, the value of  $\Gamma$  is frozen for the remainder of the convergence.

Once the circulation value is calculated, the next step is to find which cells to apply the potential jump to. In two-dimensions, a reference line is first drawn from the trailing edge to the farfield boundary (see Figure 8.9). Given the trailing edge point and the reference line, a new coordinate system is defined where the origin is located at the trailing edge. See Figure 8.10 for location of the new coordinate system. The cells are then transformed into this new coordinate system and a search algorithm is implemented in order to find all the cells that intercept the reference line. Not every cell that intersects the reference line will be needed in the Kutta condition. Only the cells that have a neighboring cell on the opposite side of the reference line will have the Kutta condition applied to it. This is determined by checking the cell centers. For a cell to have the Kutta condition enforced on it, the cell center must lie on one side of the reference line while at least one of its neighboring cells has its cell center on the opposite side of the reference line. All the cells that satisfy this

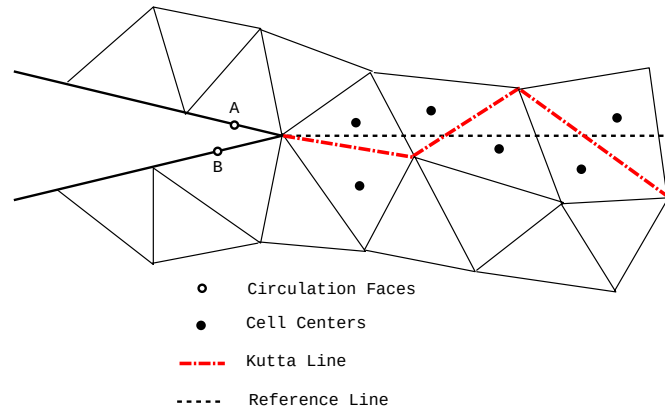


Figure 8.9: Illustration of how the Kutta condition is applied.

condition are called Kutta cells and will have the value of their potential modified to satisfy the proper jump condition. The cell faces that lie between the cells that are on opposite sides of the reference line form the Kutta line. The Kutta line does not have to be straight, but simply has to connect the trailing edge of the airfoil to the farfield boundary.

For three dimensions, the reference line becomes a reference plane and the above procedure is carried out in a similar manner. The circulation value will not be constant over the span of the wing. In practice, the circulation is near a maximum at the center of the wing and goes to zero at the wing tip. Therefore a smooth transition occurs for cells that border the reference plane on the outer wing tip boundary.

## 8.11 Boundary Conditions

In general, boundary conditions can be applied to either boundary faces or exterior boundary cells. Exterior cells lie outside the boundary and are sometimes referred to as ghost cells. In the present potential code, exterior cells are used to enforce boundary conditions. The boundary values of  $\phi$  are taken to be at the center of the ghost cells. When computing the interior gradients at faces, the ghost cells are used in the calculations whenever applicable. For the gradients on the boundary faces, a

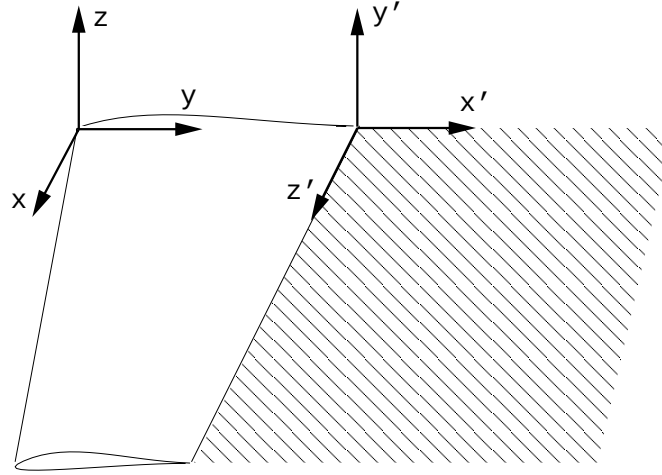


Figure 8.10: Coordinate transformation to determine the Kutta plane.

similar method is used that resembles what is done for the interior faces. The K-exact least squares method is applied on each boundary face to attain the velocity and density that is required for computing the residual. When the method is applied to a boundary face, the following cells are used to form the stencil: the ghost cell, the adjoining interior cell, and the cells surrounding the interior cell. An example of a stencil used to compute the gradient on a boundary face is shown in Figure 8.11. The ghost cell is formed such that it is a mirror reflection of the adjacent interior cell. This ensures that the vector connecting the center of the ghost cell with the adjacent interior cell center lies in the direction of the boundary normal vector. This condition on the ghost cell is important in satisfying the tangency boundary condition, which states that the velocity normal to the boundary must be zero. The tangency condition is discussed in more detail below. The details on forming the ghost cell is given in Appendix D.

Attention is now turned to the boundary conditions. Potential flows are widely used for airfoils and other flows where a farfield boundary exists. At these outer boundaries, the flow field is assumed to be known and is commonly taken as the freestream condition. The potential at a farfield boundary, assuming freestream conditions, can be expressed as,

$$\phi = V_{\infty} \cdot \vec{x} \quad (8.69)$$

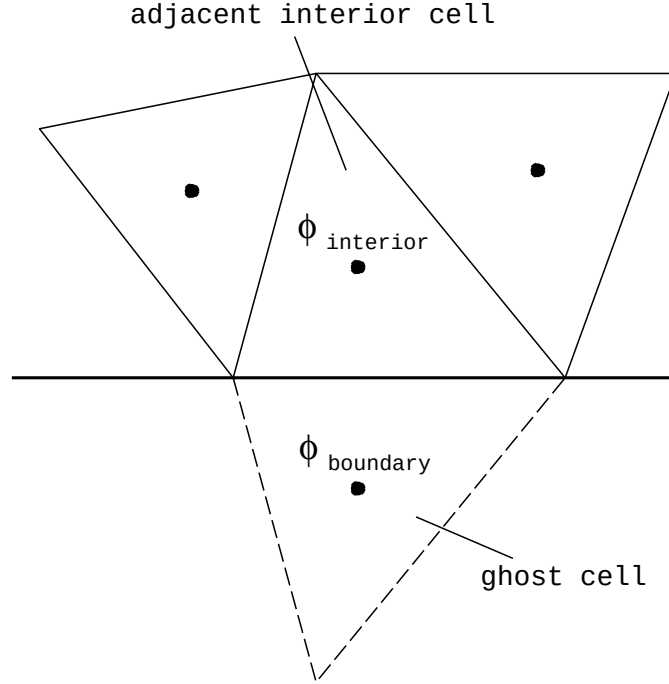


Figure 8.11: Boundary ghost cell and stencil for flux calculation.

where  $\vec{x}$  is a vector originating from a reference point to the point on the boundary. The reference point for the present code is taken as the origin of the grid coordinate system. This farfield condition assumes a non-lifting body. When lift is being produced, the Kutta condition is enforced giving rise to a jump in the potential across the reference plane. Because of this, the above boundary condition is modified to take into account the circulation and the jump in  $\phi$  across the wake. The vortex of strength  $\Gamma$  is added to the boundary condition. The farfield boundary condition for a lifting airfoil becomes,

$$\phi = V_{\infty} \cdot \vec{x} + \frac{\Gamma \theta}{2\pi} \quad (8.70)$$

In the above equation,  $\theta$  is the angle measured from the reference line. This condition allows the jump in  $\phi$  due to the Kutta condition be properly taken care of at the boundary where the reference line intersects it. For wings or other three dimensional problems, adding a vortex to the farfield boundary is not straight forward. Therefore a different boundary condition is used which follows that of Holst [151]. The boundaries downstream of the wing are initially set to the freestream condition using equation

(8.69). As the adjacent interior cell to the boundary changes, the corresponding boundary cell changes by the same amount. This can be stated as follows,

$$\Delta^n \phi_{interior} = \Delta^n \phi_{boundary}$$

This condition allows the jump in the potential function to be properly handled at the boundary where the reference plane intersects it.

For a solid wall at rest, the boundary condition is the no penetration (or tangency) condition given as,

$$\rho \vec{V} \cdot \hat{n} = \rho \frac{\partial \phi}{\partial n} = 0 \quad (8.71)$$

This condition states that the gradient of  $\phi$  normal to the wall must be zero. If the ghost cell is a mirror reflection of the interior cell, then the following equation will hold true,

$$\vec{V} \cdot \hat{n} = \frac{\partial \phi}{\partial n} = \frac{\phi_{interior} - \phi_{boundary}}{\Delta n}$$

where  $\Delta n$  is the distance between the ghost cell center and the adjacent interior cell center. The tangency condition will then be satisfied by setting  $\phi_{boundary} = \phi_{interior}$ . This allows for a very straight forward way of updating the boundary value.

# Chapter 9

## Potential Flow Results

### 9.1 Introduction

In this chapter, results will be presented for the unstructured potential flow solver. Two dimensional solutions will be given for a circular arc bump and several airfoils. The airfoils include: NACA 0012, NACA 4412, and LNV109A. A more indepth discussion of NACA airfoils can be found in Abbott and Von Doenhoff [152]. The LNV109A airfoil was tested experimentally by Liebeck and Camacho [153]. The ONERA M6 [1] wing will be used for three dimensional results.

The chapter will begin with a discussion on several aspects of the flow solver. This will include a comparison of three of the gradient schemes described in Chapter 8. The effect of Mach number and the weight parameter used in the K-exact least squares method is studied to determine the effect on solution quality and convergence. The GMRES and Point Jacobi schemes are also discussed, along with GMRES preconditioning.

In the remaining portion of the chapter, solutions of various flow cases will be presented. In addition to the unstructured potential flow solver, solutions will be presented for an unstructured compressible Euler solver (*GUST*), an incompressible Euler solver (INS), a two dimensional structured potential code (FLO36), and a three dimensional structured potential code (TOPS). The unstructured Euler flow solver *GUST* [154] uses a finite volume, cell centered formulation similar to the potential flow

solver discussed in the previous chapter. All solutions using *GUST* are performed with the Point Jacobi iteration scheme, and are spatially second order accurate. The incompressible Euler solver is the same flow solver that was discussed in Chapter 3. Solutions to this code are third order accurate and use the line Gauss-Seidel iteration scheme. The two dimensional potential code, FLO36, was written by Jameson [115] and uses the ADI implicit iteration algorithm. The code also takes advantage of multi-grid convergence acceleration. The conformal mapping scheme of Sells [88] is applied to the mesh, which is internally generated within the code. The three dimensional potential code, TOPS, was written by Holst [151] and [121]. This code uses the approximate factorization algorithm, along with a chimera grid scheme.

In the discussion that follows, the acronyms *GUST*, INS, FLO36, and TOPS will be used to represent their respective codes mentioned above. For the unstructured potential code, the acronym UPC (unstructured potential code) will be used. All the unstructured grids that are used in this chapter were made using a grid generator that accompanies the flow solver /GUSTnos [154]. For the structured grids, the Gridgen [155] grid generator package was used.

## 9.2 Comparison of Gradient Schemes

The first case to be discussed is the subsonic flow over a bump. The problem consists of a uniform flow being perturbed by a small disturbance, in this case the bump.

Computational results for the different gradient schemes discussed in section 8.4 will be given for the subsonic bump case. The circular arc bump has a thickness to chord ratio of 0.042.  $C_p$  results will be given for both a quadrilateral and a triangular 2-D mesh. The quadrilateral grid has dimensions  $81 \times 33$  which gives a total of 2,640 cells. The bump surface is described by 30 cells. The triangular grid has a total of 2,244 cells where 29 of these cells are located on the bump surface. Plots of the two grids are given in Figures 9.1 and 9.2. Four gradient schemes were introduced back in 8.4. These were the Gauss Divergence, modified Gauss Divergence, K-exact and K-exact least squares methods. The K-exact method will not be discussed here since the least squares version was favored. Any further reference to the K-exact method

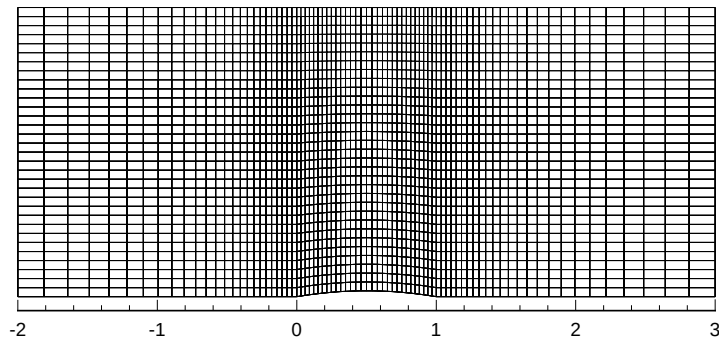


Figure 9.1: Quadrilateral grid for subsonic bump case with dimensions  $81 \times 33$ .

will imply the least squares version. The pressure coefficient for the remaining three schemes is presented in Figures 9.3 and 9.4. The freestream velocity for this case was  $50m/s$ . Even though the speed of this flow would be considered incompressible, density was allowed to vary. The weights used in the K-exact least squares method were set at 1,000.

The solution for the quadrilateral mesh given in Figure 9.3 appears to be smooth for all three schemes. The K-exact method reproduces the freestream conditions best at the outer boundaries. This is observed by seeing how close  $C_p$  goes to zero away from the bump. A slight indentation is observed for the modified Gauss Divergence scheme at the peak velocity point. This seems to be due to the grid since the indentation did not show up on the triangular grid. For the triangular grid, Figure 9.4, problems are immediately seen for the two Gauss Divergence methods. The  $C_p$  for these two schemes are very spurious and are not very accurate compared to the K-exact method. From these figures, it can be observed that methods using the Gauss Divergence formulation are not suitable for grids using triangle cells.

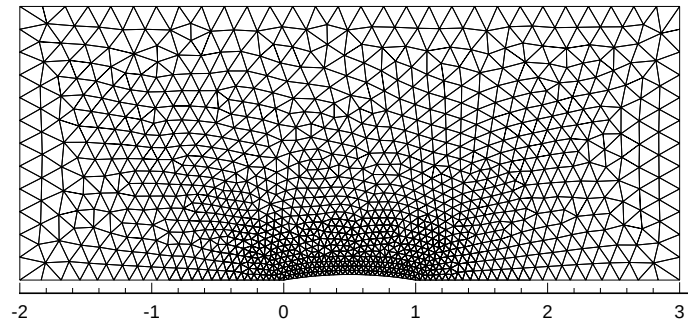


Figure 9.2: Triangular grid for subsonic bump case consisting of 2244 cells.

### 9.3 Effect of Weights Used in Least Squares

In section 8.4.4, a K-exact method was described that used a weighted least squares concept to compute gradients. In forming the gradient of the potential function, a constant weight value is chosen. The effect that different values of the weight have on both convergence and  $C_p$  will now be studied.

The NACA 4412 airfoil is used in the calculations. The airfoil is computed at zero angle of attack, at a Mach number of 0.50. The grid consists of 3,364 triangles where 121 of the cells are used to describe the airfoil surface. Five different values of the weight, denoted by  $r_w$ , are used in the computations. The values are: 100, 500, 1000, 5000, and 20000. The implicit GMRES method was used to converge the solution. A weight value of 100 causes instabilities in convergence, and a solution for this weight is not attained. A minimum value of  $r_w$  therefore exists for each problem. If a value of  $r_w$  is below this minimum, instabilities will be introduced into the implicit scheme. For most systems of equations, after the linearization is performed a term is added to the main diagonal to make the scheme stable. For the present system the

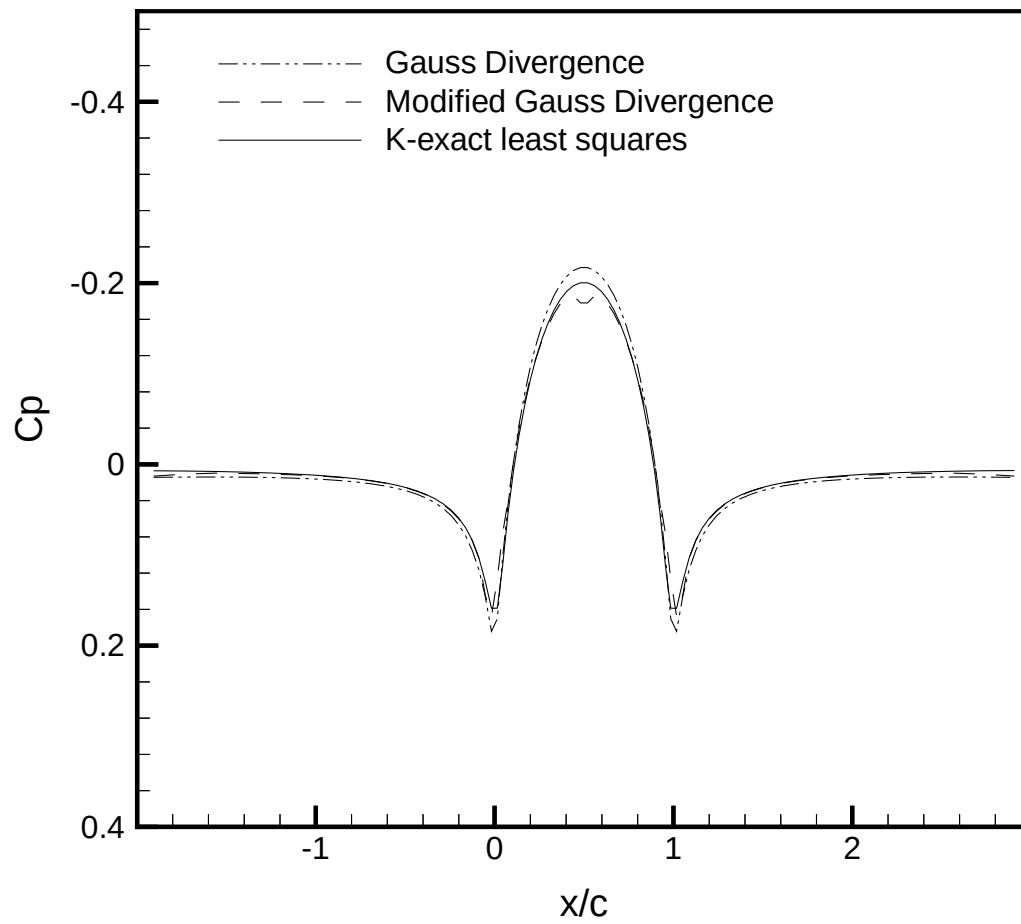


Figure 9.3: Comparison of different gradient schemes for subsonic bump. (quadrilateral grid)

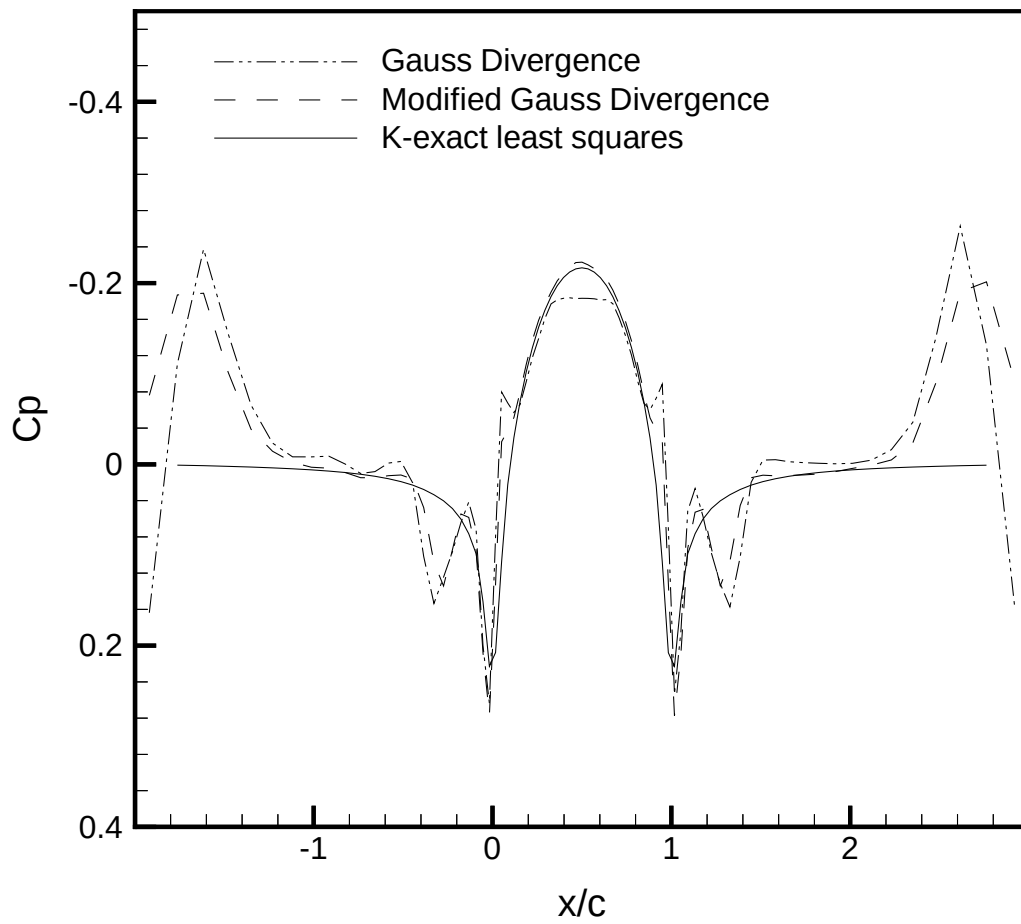


Figure 9.4: Comparison of different gradient schemes for subsonic bump. (triangular grid)

stability from using higher values of  $r_w$  comes from not using all the cells that have contributions to the Jacobian matrix for a given cell. Recall that only the adjacent cells are used to form the off-diagonal terms in the Jacobian matrix, which are the only cells that are weighted.

The convergence plot for the NACA 4412 airfoil is given in Figure 9.6 for the values of  $r_w$ . The lower values of  $r_w$  tend to converge better at first, but get “hung” after a certain level of convergence. For values of 5,000 and 20,000, the convergence was not hindered. The sudden drop in convergence around 180 iterations is due to the freezing of  $\Gamma$ . The  $C_p$  distribution is given in Figure 9.5 for  $r_w$  values of 500 and 20,000. Very little change in the  $C_p$  plot is observed by changing the value of the weight. Therefore a larger value of  $r_w$  can be used to ensure a stable convergence without any loss of accuracy in the solution.

## 9.4 Effect of Mach Number on Convergence

Due to the mathematical nature of the potential equation, the Mach number has a large impact on convergence. This is mainly due to the equation changing from elliptic to hyperbolic as the Mach number goes supersonic. When the Mach number increases, the upwind dependence will increase as well. As the nature of the flow approaches hyperbolic, a decrease in the convergence rate will occur due to the upwind influence on the solution. When the flow field consists of both elliptic and hyperbolic regions, the convergence rate will be the slowest.

Flow over a circular arc bump is computed for various Mach numbers to study the convergence problems associated with increasing Mach number. The mesh consists of 1,861 triangle cells, with 40 cells on the bump. The grid is shown in Figure 9.7. The GMRES algorithm is used with preconditioning along with three inner iterations. The Mach numbers tested are: 0.10, 0.25, 0.50, 0.75, and 0.85. Only the Mach 0.85 flow is supersonic. For this case, the values of  $C$  and  $M_c$  used in equation (8.53) are 2.5 and 0.95 respectively. If the value of  $C$  is much larger, instability in the density upwinding occurs. Also for this case, the value of  $\beta$  in equation (8.65) is set to zero.

A plot of the convergence is given in Figure 9.8 for each Mach number. The

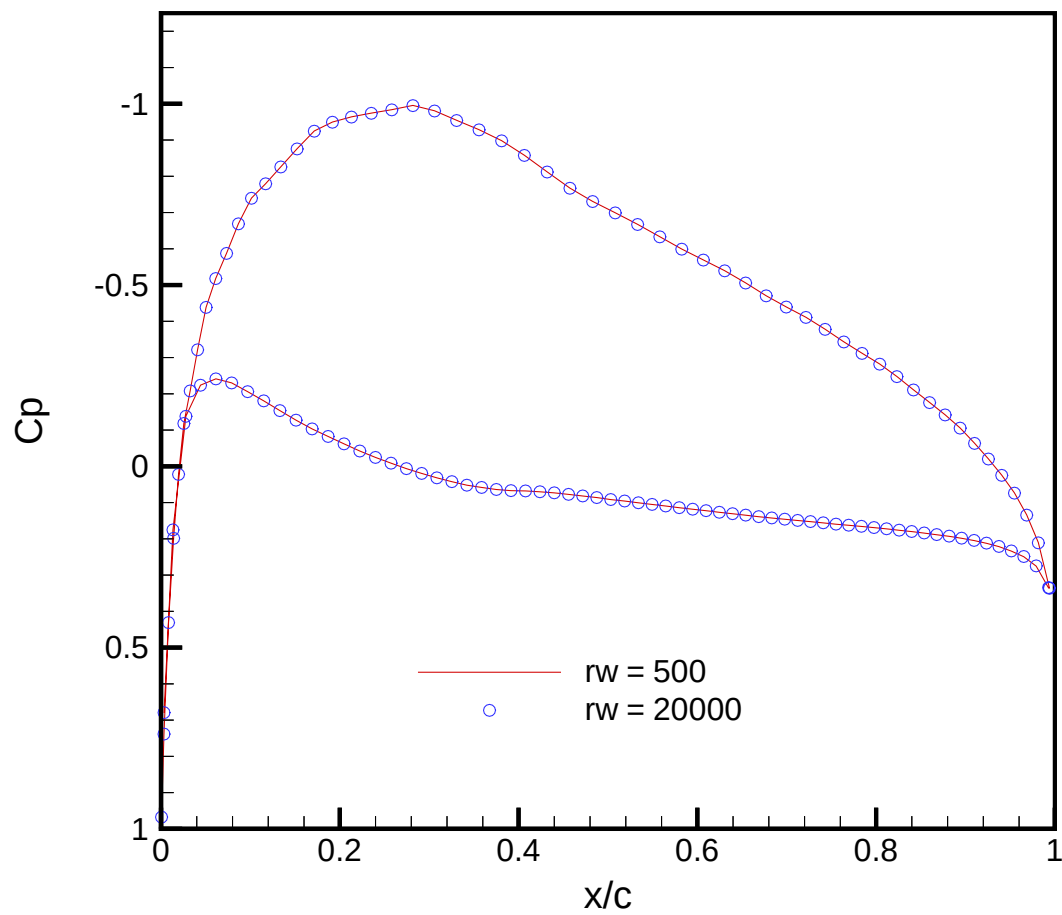


Figure 9.5:  $C_p$  comparison of NACA 4412 airfoil for different weights on gradient calculation. Flow condition:  $M = 0.50$  at  $0^\circ$ . Grid consists of 3,364 triangular cells with 121 cells on the airfoil surface.

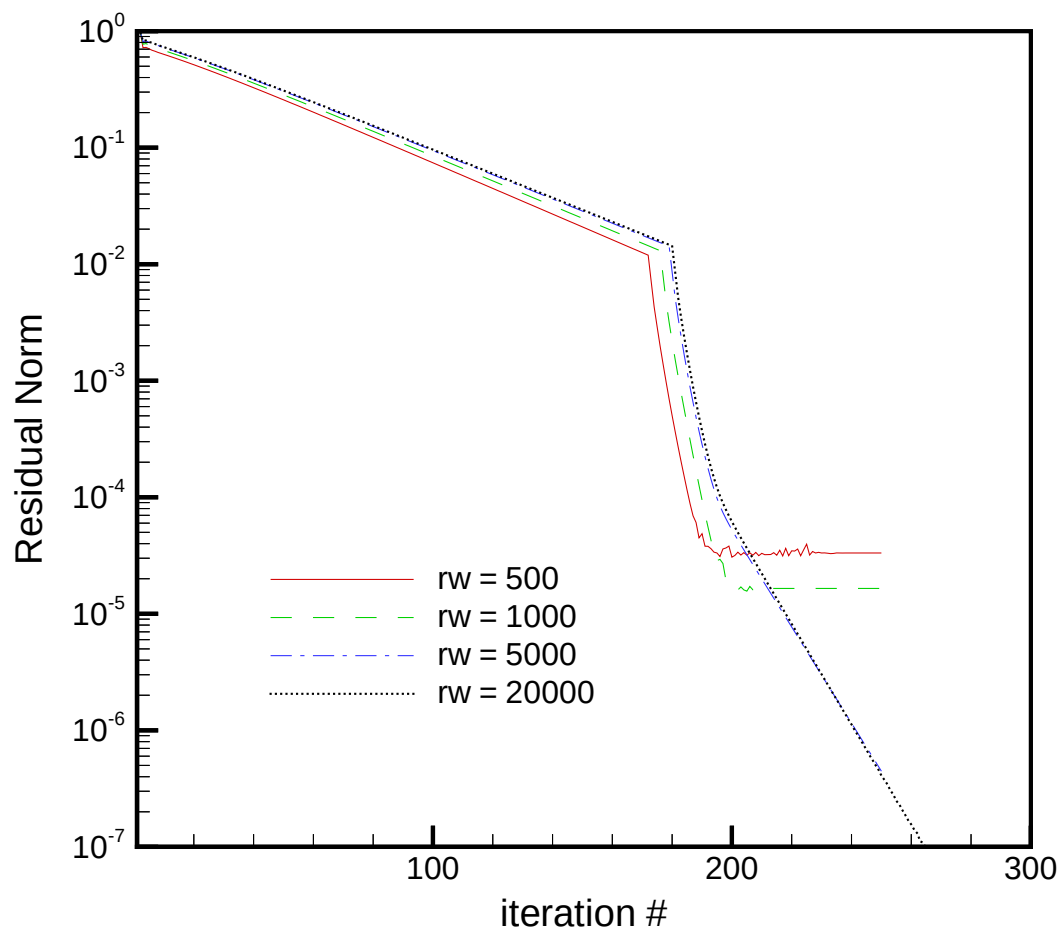


Figure 9.6: Residual comparison of NACA 4412 airfoil for different weights on gradient calculation. Flow condition:  $M = 0.50$  at  $0^\circ$ . rid consists of 3,364 triangular cells with 121 cells on the airfoil surface.

pressure coefficient is given in Figure 9.9 for the subsonic flows and in Figure 9.10 for the  $M = 0.85$  case. For the shock case, the solution from *GUST* using the same mesh is plotted for comparison. The Mach 0.10 and 0.25 cases have only a slight difference in both the convergence and the  $C_p$  solution. The flow is close to being incompressible at these speeds, giving very little difference in the results. At  $M = 0.50$  the flow is becoming more compressible and at  $M = 0.75$  the compressibility effects are very evident. For the subsonic flows, the increase in Mach number increases the number of iterations to converge, but not by a large amount (less than 35% from  $M = 0.1$  to 0.75). For the  $M = 0.85$  case, convergence is increased from 52 iterations at  $M = 0.75$  to 175 iterations. This extra increase in iterations is mainly due to the shock getting positioned. The potential solution compares well to the Euler solution, shown in Figure 9.10. The potential solution does not have as sharp of a shock as the Euler solution, but both shock locations are at nearly the same position. The potential code predicts a maximum Mach number of 1.17, which falls within the potential flow range.

## 9.5 Optimal GMRES Parameters

In this section, several aspects of the GMRES algorithm will be studied in order to improve convergence. The first will be the effect of preconditioning on the GMRES algorithm. For this the convergence will be looked at for the NACA 0012 airfoil. The second area that impacts the convergence is the number of Krylov subspace vectors and the number of inner iterations performed on the algorithm. To study these two parameters, the iteration and CPU times will be observed for flow over a circular arc bump and a NACA 4412 airfoil.

The main reason to use a precondition matrix is to improve the convergence rate by clustering the eigenvalues closer together. The bandwidth of the precondition matrix, as described in section 8.9, will be varied to observed the impact on convergence. The larger the bandwidth, the better the clustering of the eigenvalues should be. The bandwidth, denoted by  $bw$ , will be varied from 0 to 4. A  $bw$  of 0 corresponds to a precondition matrix consisting of the main diagonal only, while a  $bw$  of 4 consists of

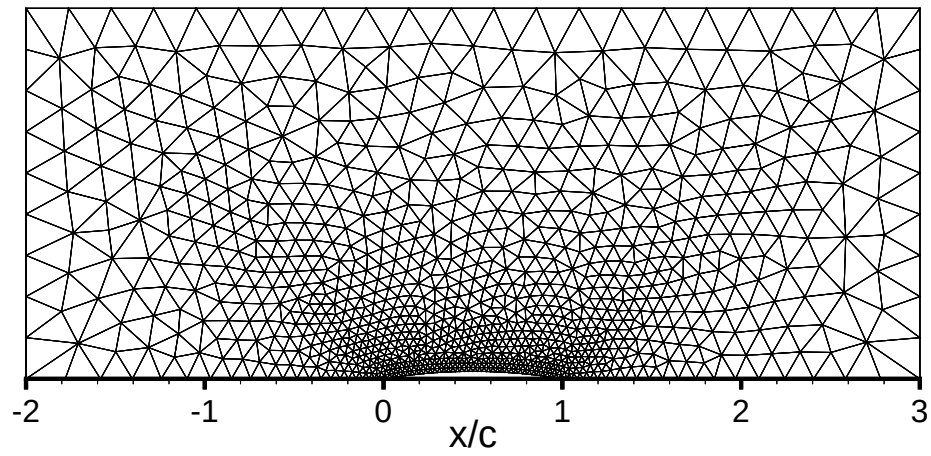


Figure 9.7: Circular arc bump mesh consisting of 1,861 cells with 40 cells on the arc bump. Arc bump has a radius of 3 units.

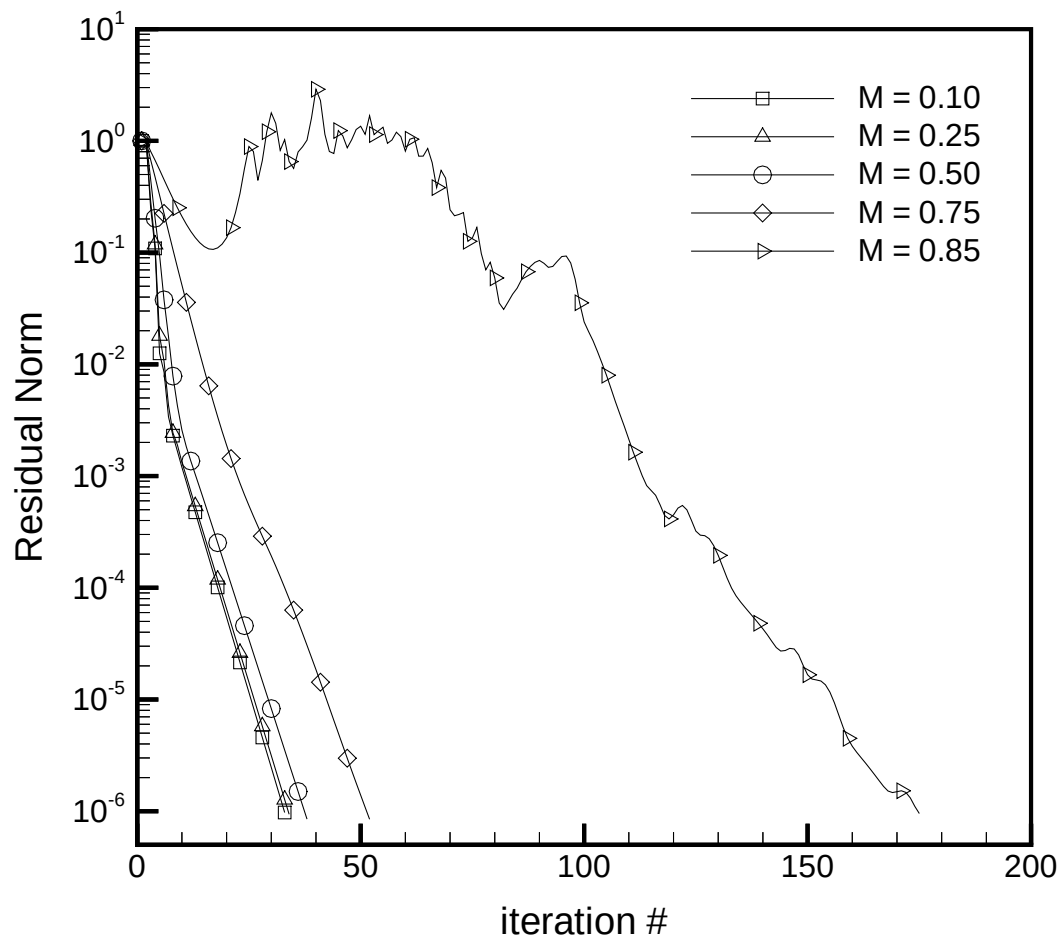


Figure 9.8: Residual comparison of circular arc bump for different Mach numbers. Grid consists of 1,861 triangular cells.

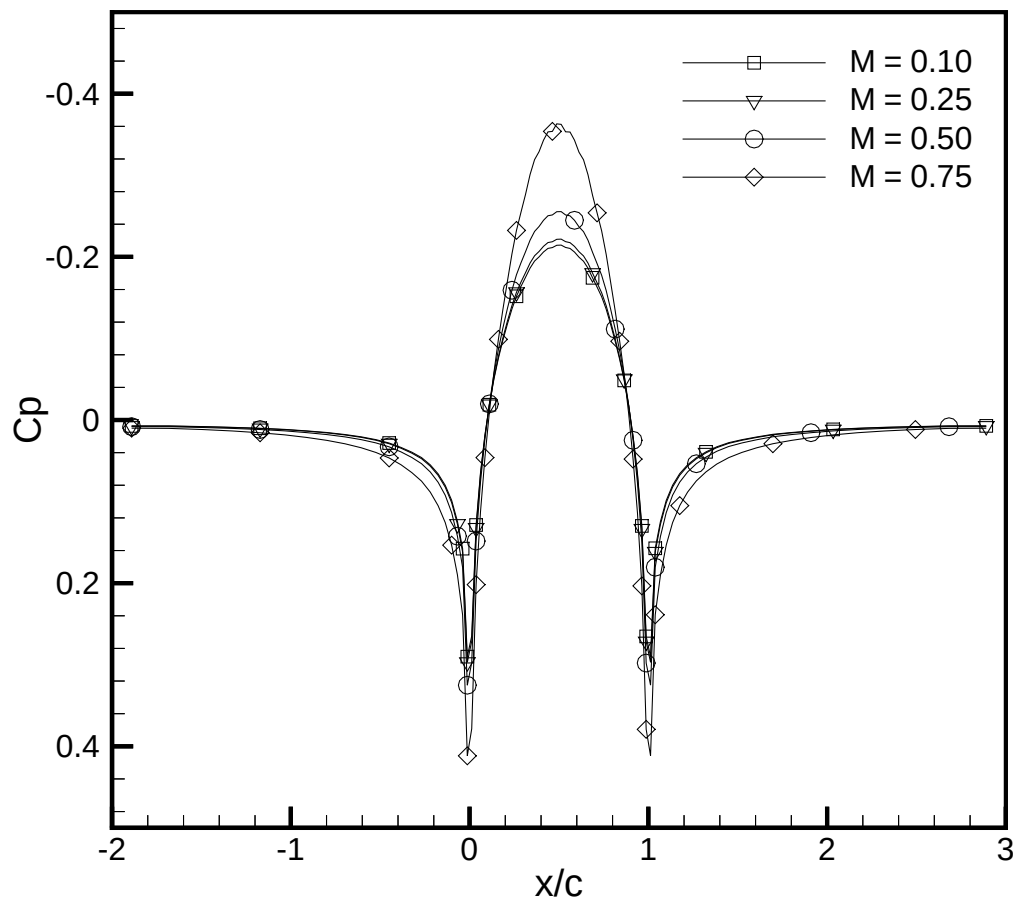


Figure 9.9:  $C_p$  results for circular arc bump at different Mach numbers. Grid consists of 1,861 triangular cells.

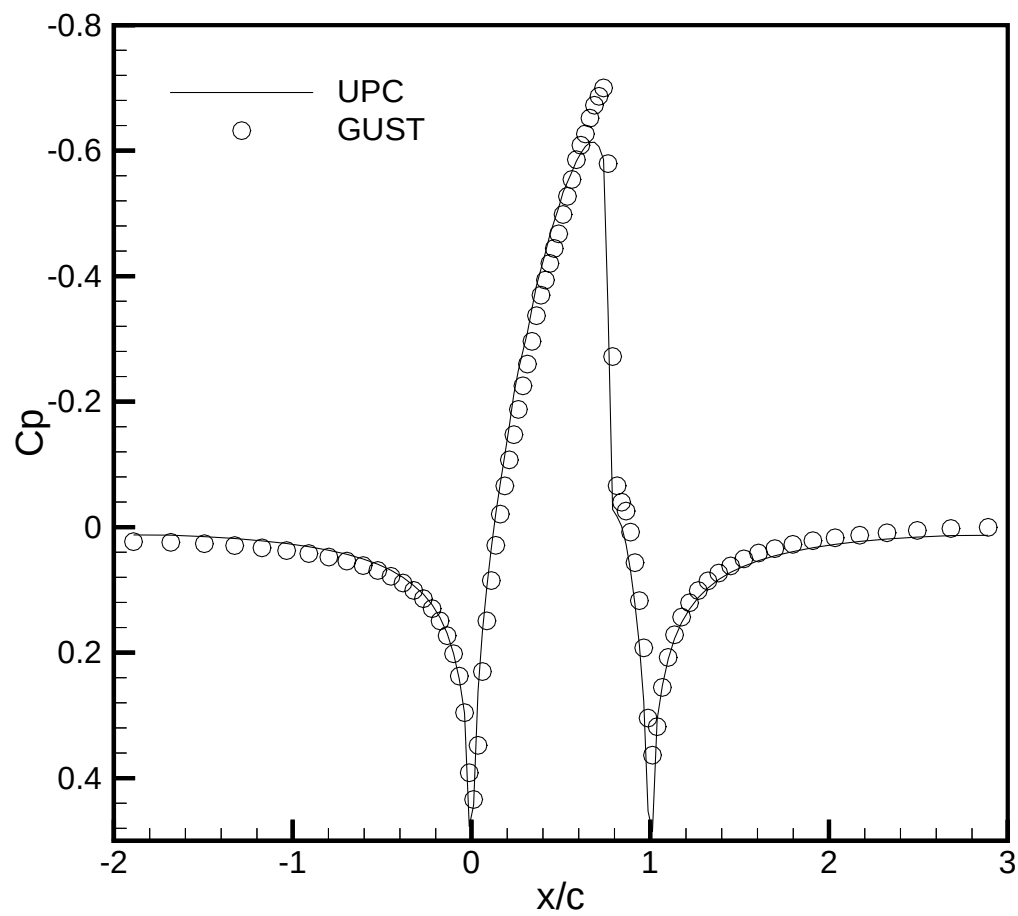


Figure 9.10: Comparison of UPC and *GUST* solutions for circular arc bump at  $M = 0.85$ . Grid consists of 1,861 triangular cells, with 40 cells on the bump.

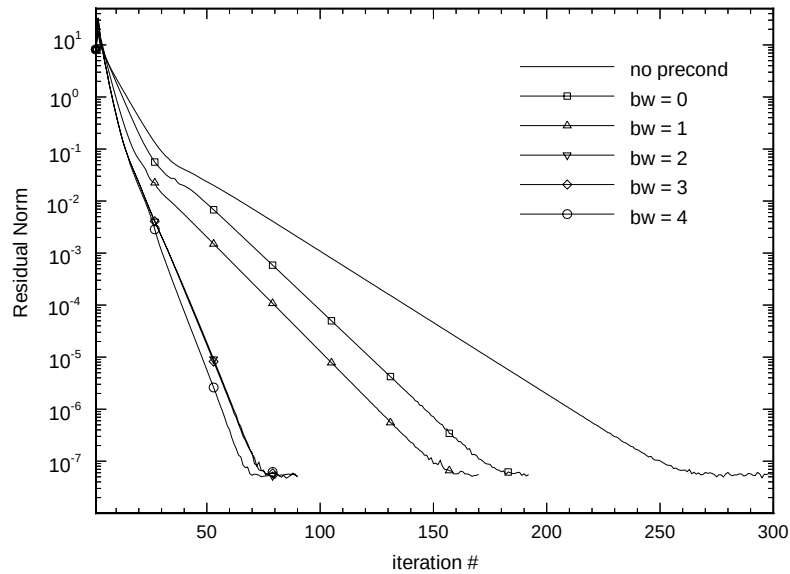


Figure 9.11: Convergence of NACA 0012 airfoil ( $0^\circ$ ;  $M=0.70$ ) for various bandwidth sizes of the precondition matrix.

eight off diagonals plus the main diagonal. Convergence is given for the NACA 0012 airfoil at zero degrees angle of attack and Mach 0.70. The value of  $r_w$  is 1000 and the grid is composed of triangles.

The convergence plot is shown in Figure 9.11. It is immediately clear that the larger values of  $bw$  give the best convergence rate. For this case,  $bw$  values of 2, 3, and 4 are approaching the optimal convergence rate. Even for  $bw = 0$  a considerable improvement is observed in the convergence. The best trade-off between memory and convergence for this case will be around  $bw = 2$ .

Details of the GMRES algorithm were discussed in section 8.7.2. The number of Krylov subspace vectors is denoted by  $k$  and the number of inner iterations is given by  $m$ . Increasing  $k$  will require more storage since each subspace vector must be stored for that iteration, as well as increase CPU time. Increasing  $m$  will only increase CPU time per iteration. The optimal values of  $k$  and  $m$  are therefore desired.

Tables 5 and 6 give both the CPU time and number of iterations to converge for different values of  $k$  and  $m$ . Table 5 corresponds to the circular arc bump case

at  $M = 0.50$ . The grid was identical to the one used in section 9.4. Table 6 is for the NACA 4412 airfoil at zero angle of attack. The grid for this case is the same as the one used in section 9.3. In both cases the bandwidth for the preconditioner was  $bw = 2$ , and for the weight  $r_w = 5000$ . Note that the NACA 4412 airfoil will have lift, and therefore have the Kutta condition applied. The CPU times correspond to a Silicon Graphics R8000 machine.

The best CPU times for each value of  $k$  is marked by an asterisk. For both cases, as  $k$  increases the best CPU time is attained by using fewer inner iterations. For the circular arc bump case, the minimum number of iterations for the solution to converge is attained as  $m$  increases. For most problems, the value of  $m$  that is needed to attain the fewest number of iterations will be very large, usually resulting in high CPU times. For most problems, a value of  $k = 10$  and  $m = 2$  to 5 is a good balance between storage requirements and CPU time.

For the circular arc bump case, the Point Jacobi scheme was also run to observe its performance against the GMRES algorithm. The CPU and iteration count for various inner iterations is listed in Table 7. The Point Jacobi scheme requires a much greater number of inner iterations to reach the minimum number of iterations to converge the case. The lowest CPU time for Point Jacobi is 9 times larger than the lowest time for GMRES. Because of this, GMRES is the linear solver of choice.

## 9.6 Full Potential and Euler Comparison

This section discusses a comparison between the codes UPC, *GUST*, and INS. The comparison consists of computing six cases, two of which are incompressible and the remaining four compressible. Only UPC and INS are used to run the incompressible cases, and only UPC and *GUST* are used for the compressible cases. The same grid is used for each flow solver. Comparisons will be made between the pressure coefficient, the number of iterations for convergence, and the CPU time to converge. In addition to this, a total of three grids are used to study how each flow solver performs on different grid levels. The grids are made up of quadrilateral cells and have the following dimensions:  $51 \times 31$ ,  $91 \times 41$ , and  $151 \times 51$ .

Table 5: Iteration and CPU times for  $M = 0.5$  circular arc bump using GMRES. Iterations based upon a  $10^{-6}$  residual.

|     |              | m=1 | 3   | 5  | 10  | 20  | 40 |
|-----|--------------|-----|-----|----|-----|-----|----|
| k=5 | # iterations | 262 | 130 | 51 | 29  | 15  | 13 |
|     | seconds      | 83  | 50  | 26 | 19  | 16* | 22 |
| 10  | # iterations | 107 | 36  | 26 | 15  | 13  | 13 |
|     | seconds      | 38  | 20  | 18 | 15* | 22  | 39 |
| 15  | # iterations | 50  | 20  | 16 | 13  | 13  | 13 |
|     | seconds      | 22  | 13* | 14 | 20  | 33  | 62 |
| 20  | # iterations | 30  | 16  | 13 | 13  | 13  | 13 |
|     | seconds      | 14* | 14* | 16 | 26  | 43  | 81 |

\* minimum CPU time for  $k$

Table 6: Iteration and CPU times for NACA 4412 airfoil using GMRES. Iterations based upon a  $10^{-5}$  residual. Flow conditions:  $M = 0.50$  at  $0^\circ$  degrees.

|     |              | m=1 | 2    | 3   | 5   | 10   |
|-----|--------------|-----|------|-----|-----|------|
| k=5 | # iterations | 957 | 569  | 409 | 255 | 164  |
|     | seconds      | 414 | 279  | 239 | 184 | 163* |
| 10  | # iterations | 396 | 193  | 185 | 155 | 143  |
|     | seconds      | 192 | 126* | 145 | 162 | 226  |
| 15  | # iterations | 262 | 159  | 154 | 144 | 140  |
|     | seconds      | 153 | 122* | 155 | 190 | 326  |

\* minimum CPU time for  $k$

Table 7: Iteration and CPU times for  $M = 0.5$  circular arc bump using Point Jacobi. Iterations based upon a  $10^{-6}$  residual.

| inner iterations | 10   | 100 | 500  | 1000 | 2500 |
|------------------|------|-----|------|------|------|
| # iterations     | 1951 | 221 | 51   | 28   | 15   |
| seconds          | 610  | 157 | 123* | 128  | 165  |

\* minimum CPU time

The NACA 0012 airfoil is used in each of the six cases. For the incompressible cases, a Mach number of 0.10 is used at  $0^\circ$  and  $1.5^\circ$  angle of attack. The  $C_p$  plots for these two cases are shown in Figures 9.12 and 9.13. The INS code is third order accurate while UPC is second order. This difference is seen in the high gradient regions near the leading edge. The INS code gives a higher peak in velocity for the medium and fine grids. In both cases, the UPC code gives nearly the same  $C_p$  profile for each grid, while the INS solution is more sensitive to the grid refinement. Therefore, a more coarse grid could be used with the potential flow solver to attain a grid converged solution, while the Euler solver would require more grid points. This can only be said for incompressible flow conditions. For the lifting case, the convergence versus both iterations and CPU time is given in Figures 9.14 and 9.15 respectively. For each of the three grids, the potential code reached the convergence criteria in fewer iterations in comparison to the Euler flow solver. This indicates that the potential equation will converge faster than the incompressible Euler equations. Even though the potential code converges faster, Figure 9.15 reveals that the Euler code takes less CPU time to reach the final solution. The reason for this result is in the structured versus unstructured algorithms. The Euler code has the advantage of using the cell indexing, which helps reduce the amount of CPU time to perform a single iteration on the solution. This one advantage gives the Euler flow faster run times for each of the three grid levels.

The four compressible flow cases are now discussed. The first two were performed for Mach 0.50 at  $0^\circ$  and  $1.5^\circ$  angle of attack. The  $C_p$  solution for these two cases are given in Figures 9.16 and 9.17. For both cases, the Euler solver (*GUST*) gives better resolution of the high gradient regions near the leading edge. Even though both codes are spatially second order accurate, the potential code is based on central differencing while the Euler flow solver is based on upwinding. This difference in the discretization has its biggest impact in regions where the gradient is large. The sensitivity to the grid refinement is about equal between the two flow solvers. There is a considerable amount of change in the  $C_p$  between the coarse and medium grids for both codes. Between the medium and fine grids, the solution change is not as large for either flow solver. Therefore, the sensitivity to the grid levels appears to be about

the same for both codes in regions of high subsonic, compressible flow.

The convergence plots for both iteration number and CPU time is given in Figures 9.18 to 9.21 for each of the two cases. Figures 9.18 and 9.19 correspond to the  $0^\circ$  case, and Figures 9.20 and 9.21 to the  $1.5^\circ$  case. For both cases, the potential flow solver requires fewer iterations to converge when compared to each grid solved using the Euler code. In fact, the number of iterations to converge the fine grid for UPC almost equals the number of iterations to solve the coarse grid case using *GUST*. Unlike the incompressible case, the potential code requires less CPU time for each case over the Euler flow solver.

The remaining two cases are transonic problems. The first has a freestream Mach number of 0.80 at  $0^\circ$  angle of attack. The pressure coefficient for this flow is shown in Figure 9.22. For the coarse grid, the potential code predicts no shock at all while the Euler code has a highly smeared shock. Shock resolution increases as the grid is refined. The shock location for both codes is at nearly the same position on the fine grids.  $C_p$  solutions for both UPC and *GUST* change dramatically from the coarse to medium grids. But from the medium to fine grid, the Euler solution shows to be more grid converged over the potential solution. This same trend is also observed for the final case which has a freestream Mach number of 0.75 at  $1.5^\circ$  angle of attack. The  $C_p$  solution for the final case is shown in Figure 9.23. Therefore, it appears from these two cases that the potential solver is more sensitive to grid refinement for transonic flows, and requires a fine grid to place the shock. For the last case, the shock is positioned further downstream compared to *GUST*. This is commonly seen for potential flow solvers when compared to Euler solutions.

The convergence plots for both cases are given in Figures 9.24 to 9.27. The same trend is seen here for the transonic cases as for the compressible subsonic cases. The potential code converges in fewer iterations than the Euler code, and does so in less CPU time. The savings in CPU time are necessary to justify using a potential flow solver over and Euler solver.

A summary of the lifting airfoil cases is given in Table 8. The coefficient of lift is given for each of the cases where lift was non-zero. The potential code gives higher values of  $C_L$  for each flow case and grid type when compared to the Euler solutions.

Table 8: Lift results for UPC, INS, and *GUST* codes for NACA 0012 airfoil. (CPU ratio is Euler CPU time divided by Potential CPU time)

| Case                   | Grid            | $C_L$  |             |        | CPU ratio |
|------------------------|-----------------|--------|-------------|--------|-----------|
|                        |                 | INS    | <i>GUST</i> | UPC    |           |
| $M = 0.10$<br>1.5° aoa | $51 \times 31$  | 0.1328 | —           | 0.1681 | 0.4       |
|                        | $91 \times 41$  | 0.1499 | —           | 0.1797 | 0.5       |
|                        | $151 \times 51$ | 0.1475 | —           | 0.1795 | 0.3       |
| $M = 0.50$<br>1.5° aoa | $51 \times 31$  | —      | 0.1858      | 0.1971 | 5.4       |
|                        | $91 \times 41$  | —      | 0.1983      | 0.2121 | 5.6       |
|                        | $151 \times 51$ | —      | 0.1882      | 0.2117 | 4.1       |
| $M = 0.75$<br>1.5° aoa | $51 \times 31$  | —      | 0.2621      | 0.2996 | 3.0       |
|                        | $91 \times 41$  | —      | 0.2957      | 0.3452 | 4.9       |
|                        | $151 \times 51$ | —      | 0.2863      | 0.3593 | 4.6       |

For the transonic case, the difference in lift is larger than the subsonic cases. This is mainly due to the isentropic condition on the potential equation which is most prevalent when shock exist in the flow. The ratio of CPU times is also given. For the incompressible case, the potential code is actually slower than the Euler code as discussed above. For the two compressible cases, the potential code ranges from 3 to 6 times faster than Euler. For the cases where lift is zero, the potential code would have an even higher ratio since the Kutta condition does not have to be implemented. Flores *et al.* [156] did a similar study on comparing the full potential and Euler equations. In that study, several different structured codes for both potential and Euler were compared. The potential codes were found to be a factor of 4 to 10 times faster than Euler codes.

## 9.7 NACA 0012, NACA 4412, and LNV109A Airfoils

In this section, results are presented for the following airfoils: NACA 0012, NACA 4412, and LNV109A. A grid convergence study will be presented first for the NACA

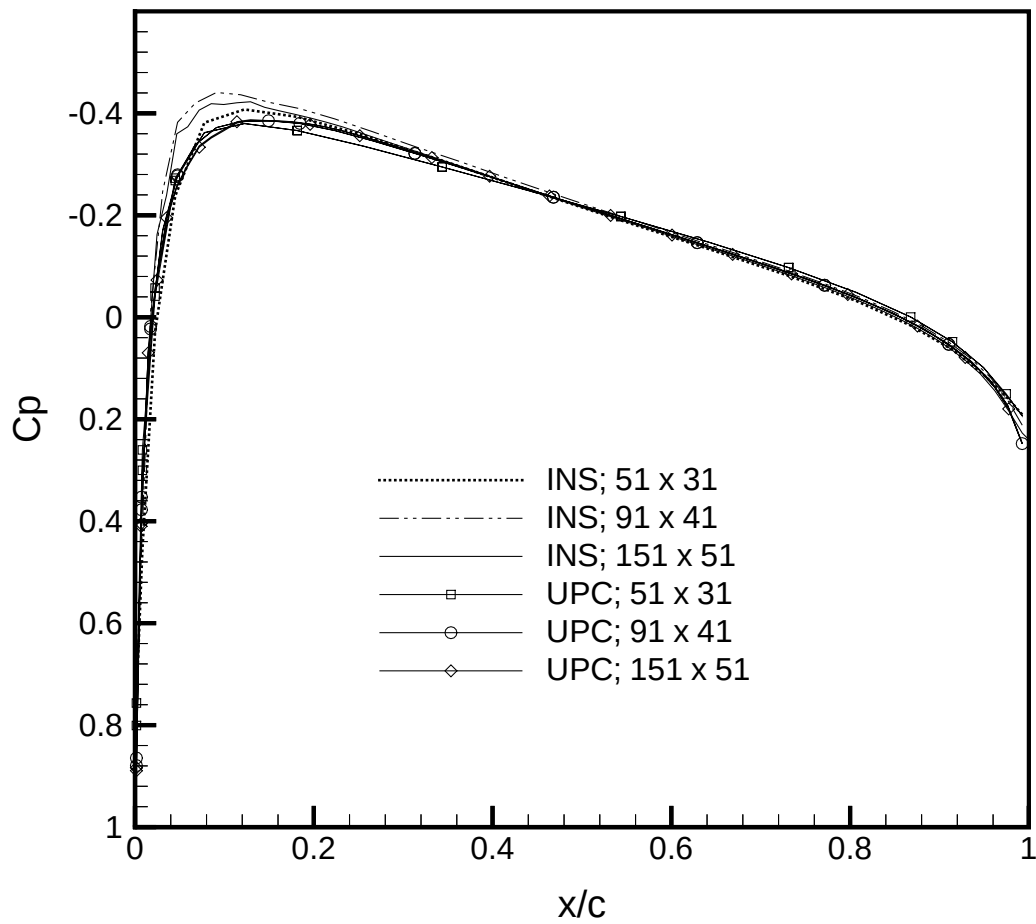


Figure 9.12: Comparison of INS and UPC at three grid levels for NACA 0012 airfoil. Flow conditions:  $0^\circ$  and  $M = 0.10$ .

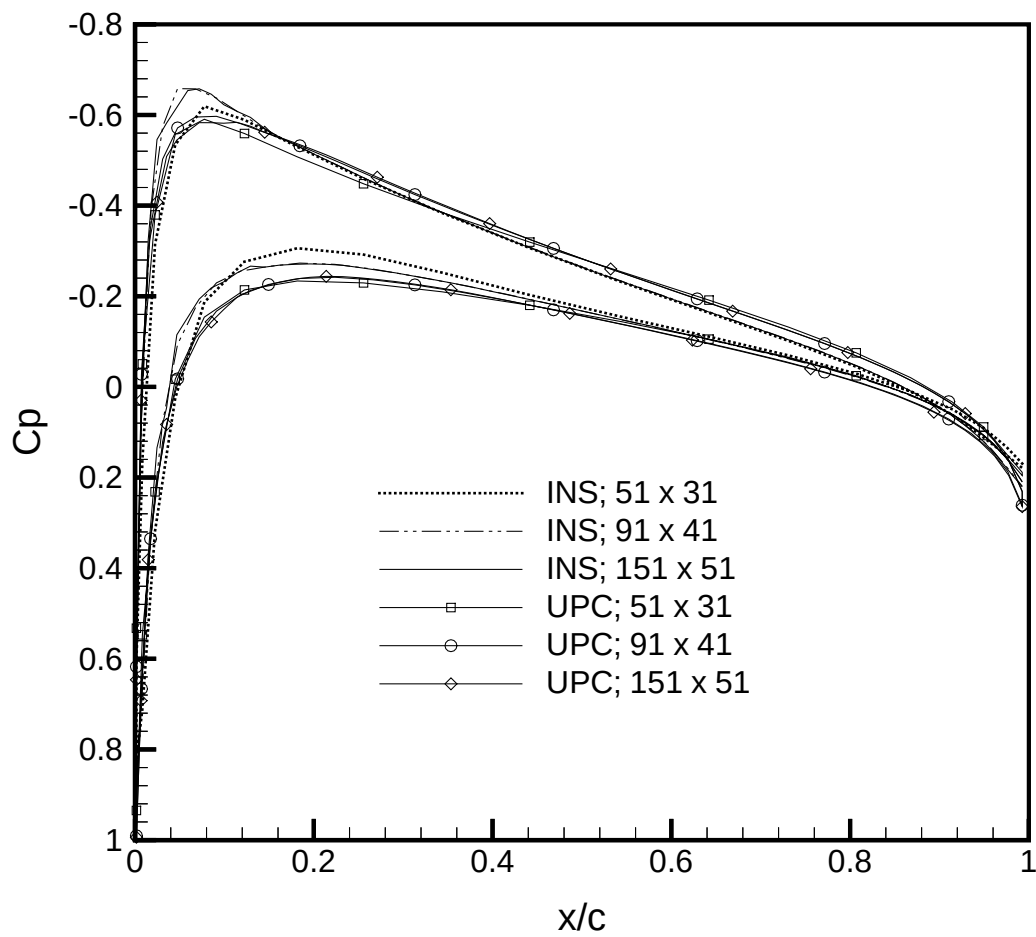


Figure 9.13: Comparison of INS and UPC at three grid levels for NACA 0012 airfoil. Flow conditions:  $1.5^\circ$  and  $M = 0.10$ .

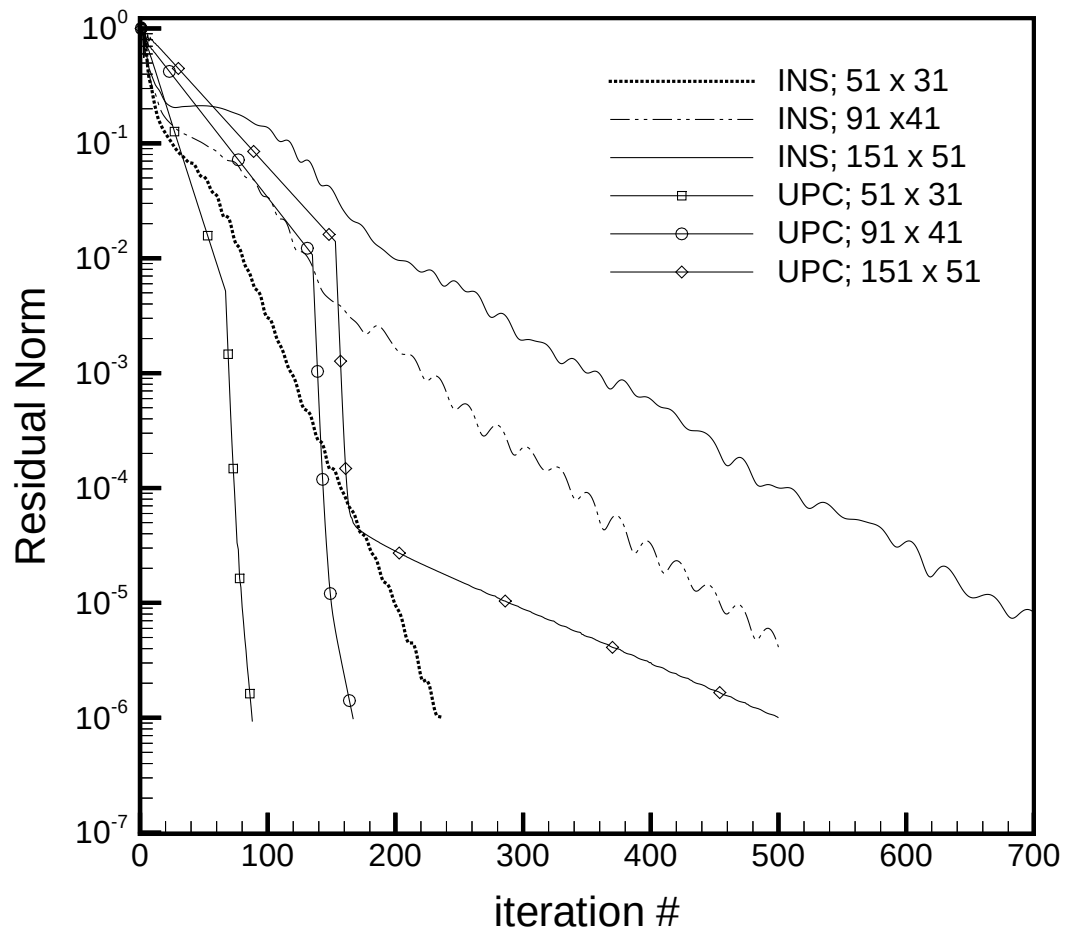


Figure 9.14: Normalized residual versus iteration number for NACA 0012 airfoil case at  $M = 0.10$  and  $1.5^\circ$ . Plot corresponds to Figure 9.13.

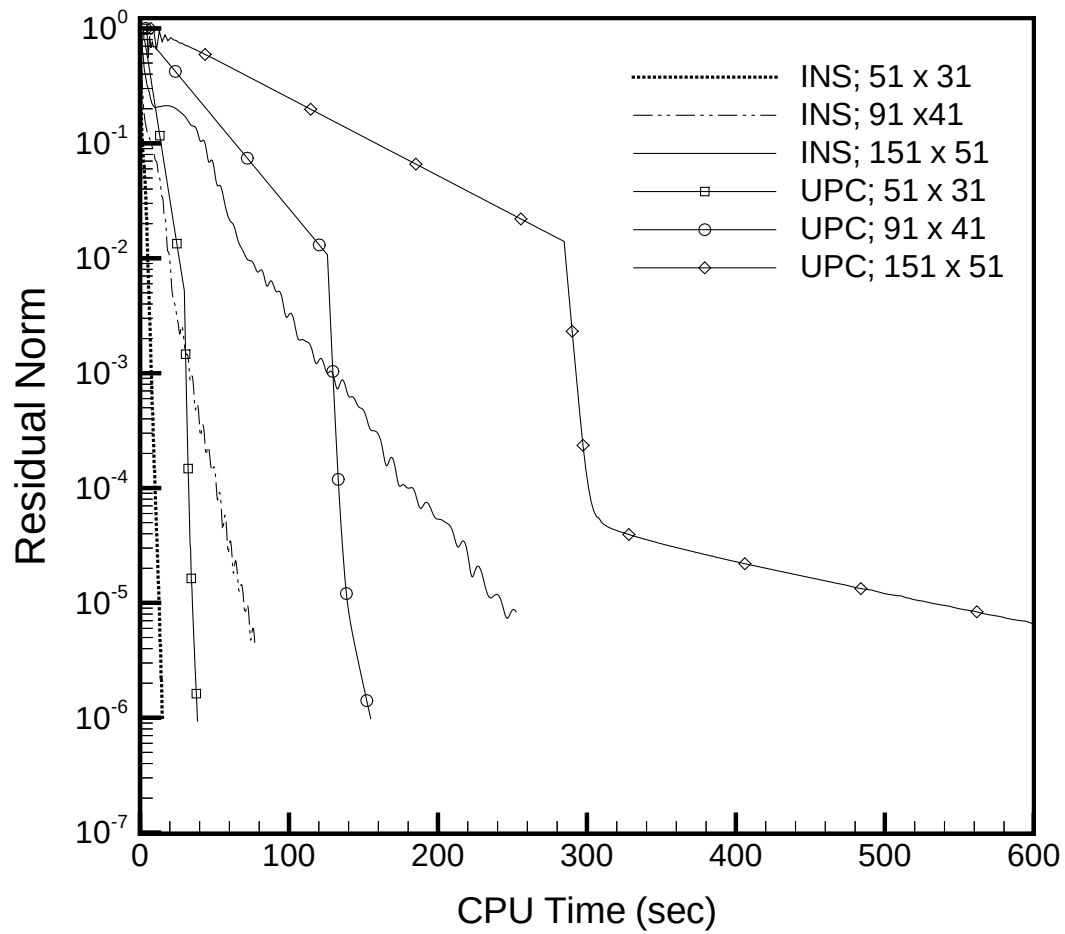


Figure 9.15: Normalized residual versus CPU time for NACA 0012 airfoil case at  $M = 0.10$  and  $1.5^\circ$ . Plot corresponds to Figure 9.13.

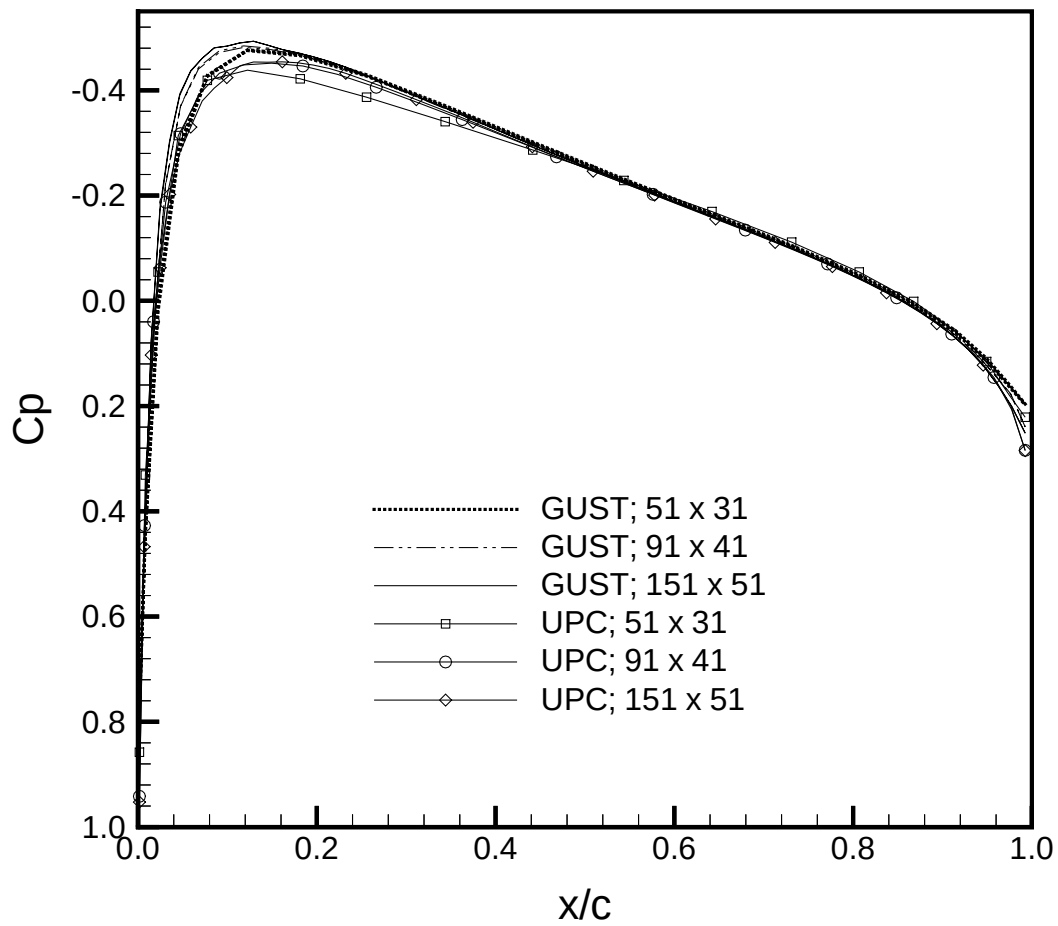


Figure 9.16: Comparison of *GUST* and UPC at three grid levels for NACA 0012. Flow conditions:  $0^\circ$  and  $M = 0.50$ .

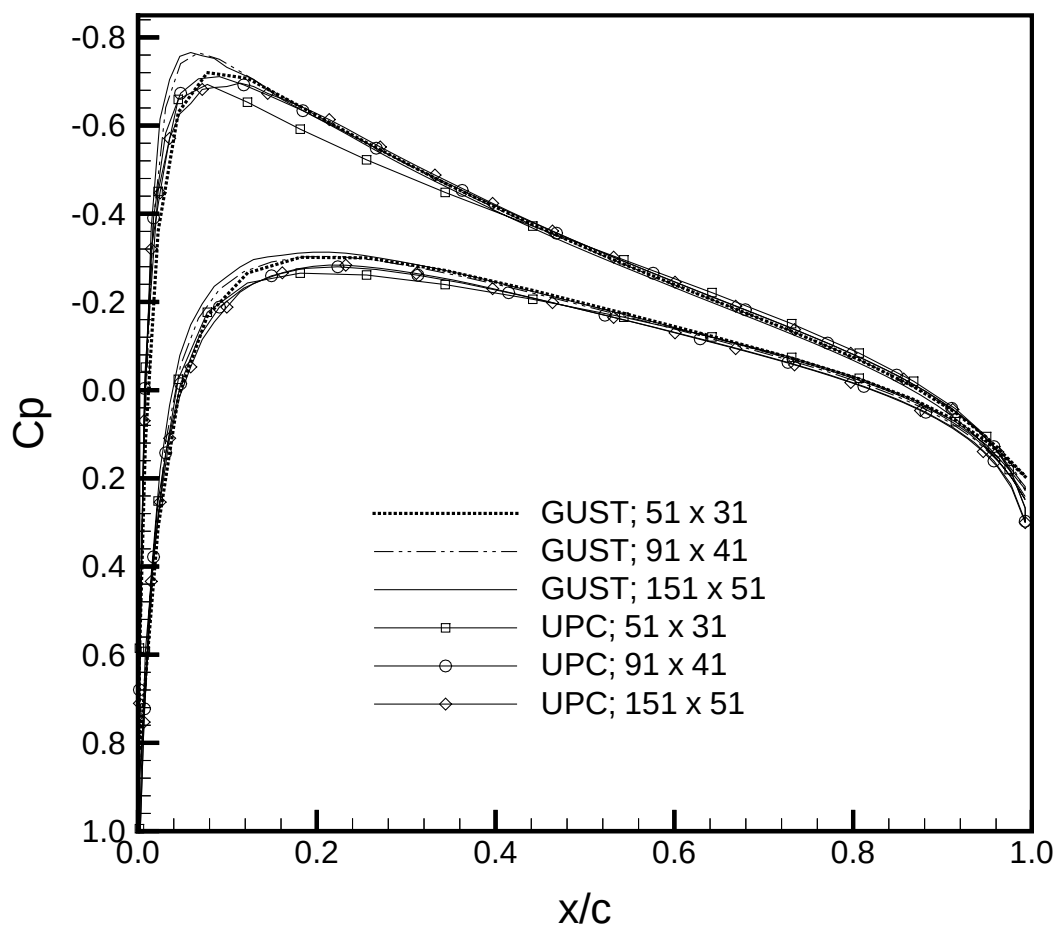


Figure 9.17: Comparison of *GUST* and UPC at three grid levels for NACA 0012. Flow conditions:  $1.5^\circ$  and  $M = 0.50$ .

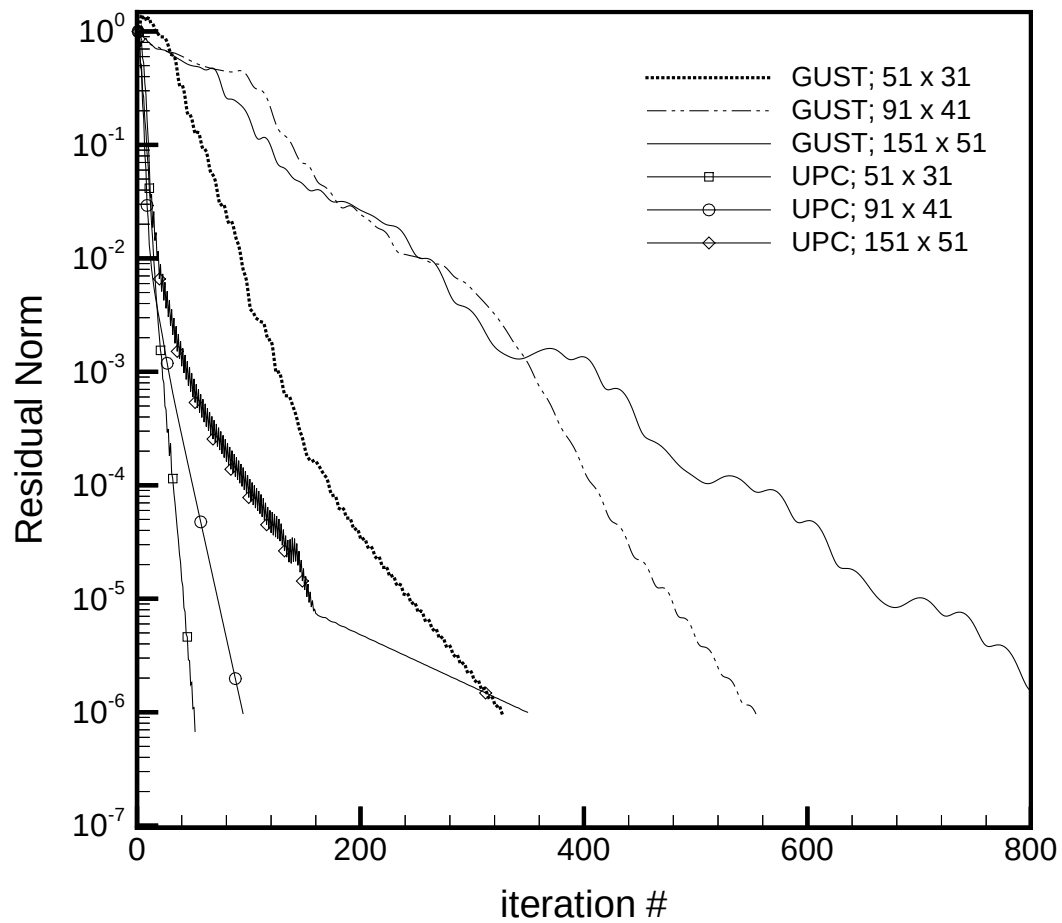


Figure 9.18: Normalized residual versus iteration number for NACA 0012 airfoil case at  $M = 0.50$  and  $0^\circ$ . Plot corresponds to Figure 9.16.

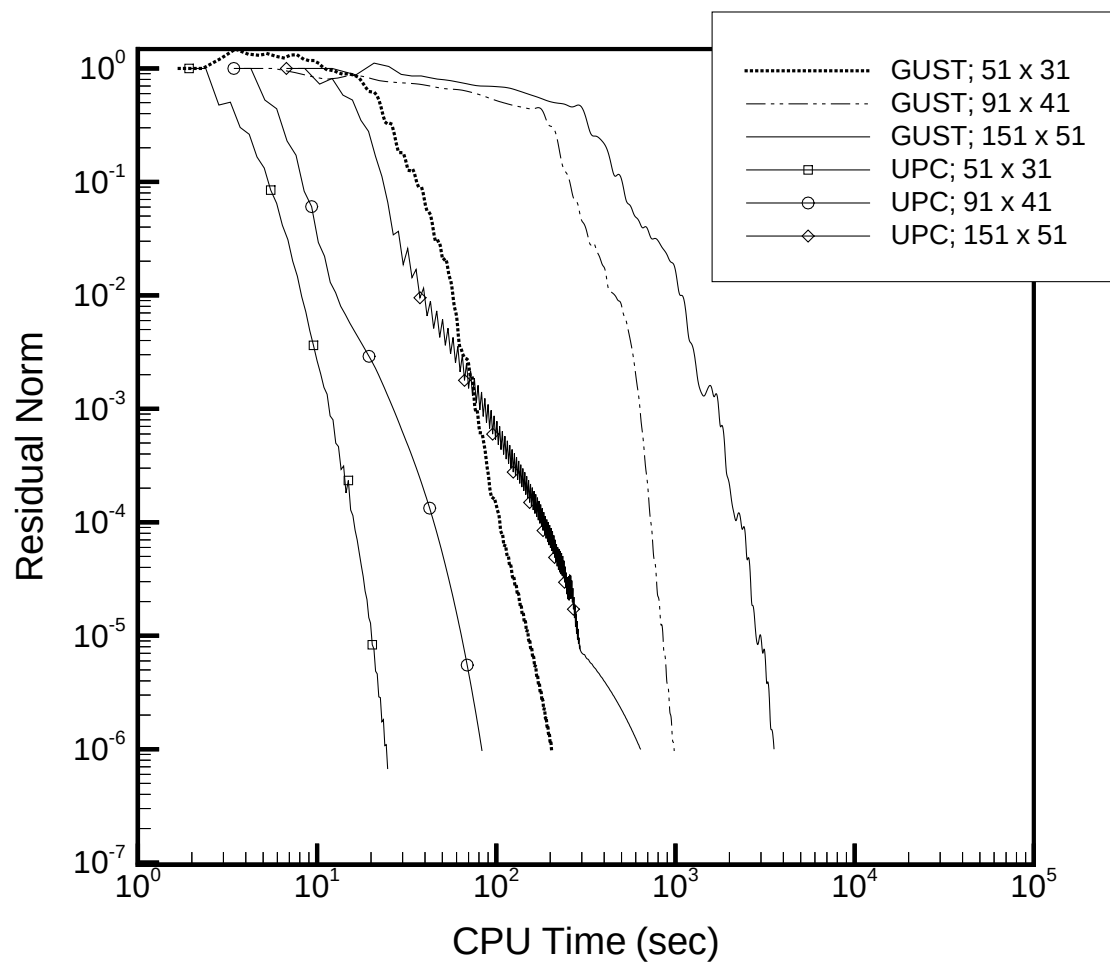


Figure 9.19: Normalized residual versus CPU time for NACA 0012 airfoil case at  $M = 0.50$  and  $0^\circ$ . Plot corresponds to Figure 9.16.

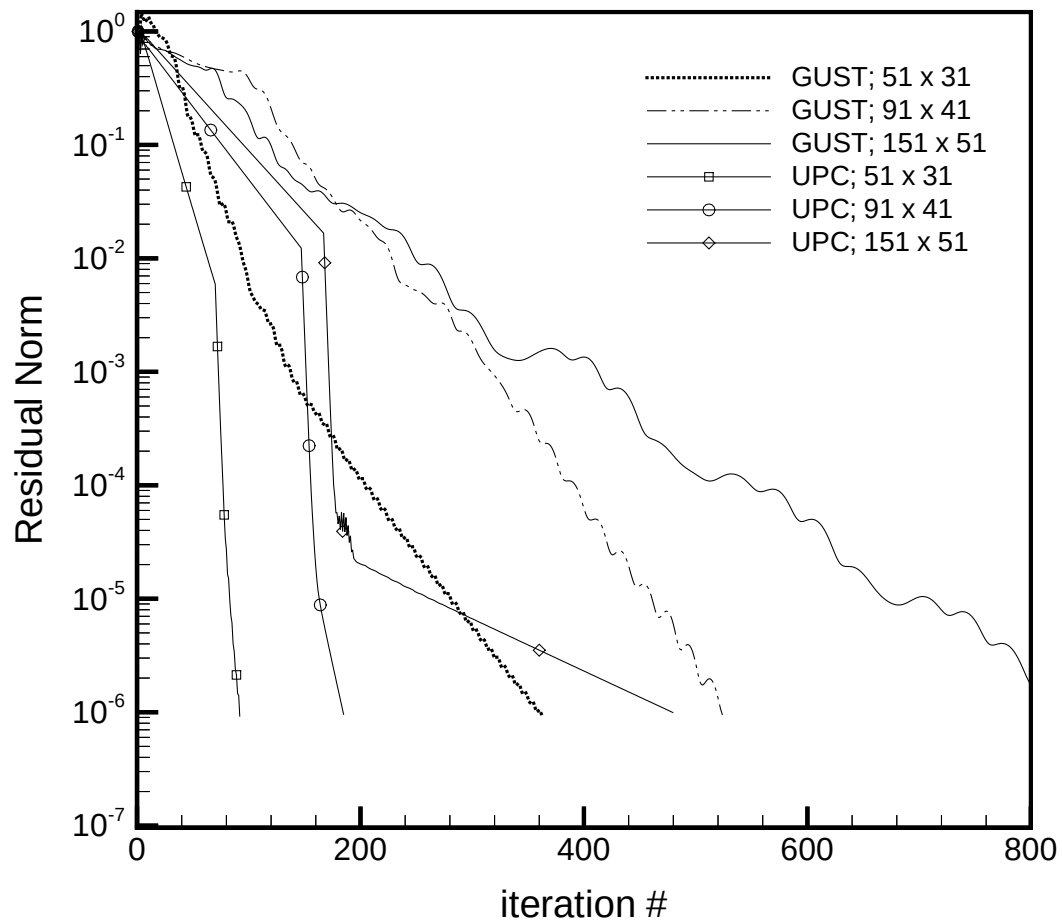


Figure 9.20: Normalized residual versus iteration number for NACA 0012 airfoil case at  $M = 0.50$  and  $1.5^\circ$ . Plot corresponds to Figure 9.17.

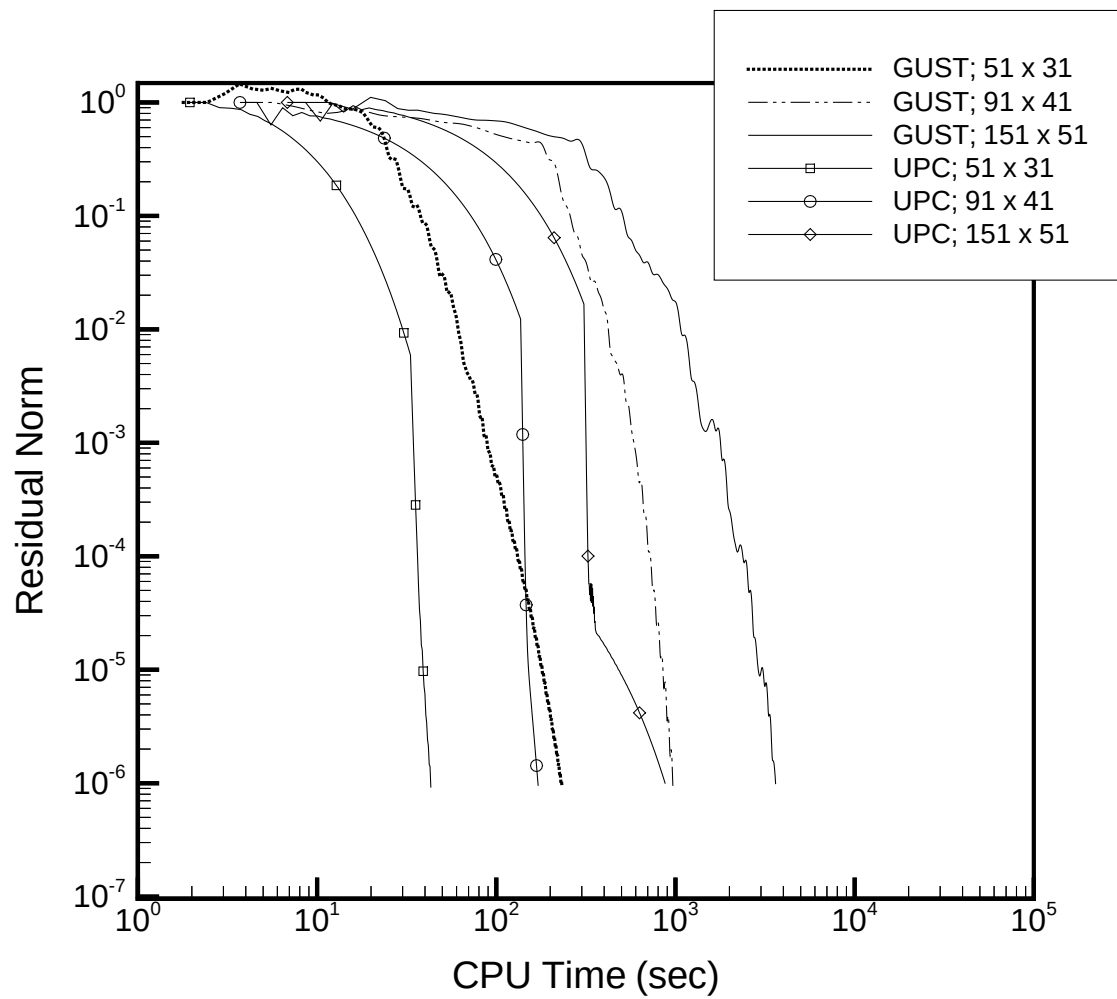


Figure 9.21: Normalized residual versus CPU time for NACA 0012 airfoil case at  $M = 0.50$  and  $1.5^\circ$ . Plot corresponds to Figure 9.17.

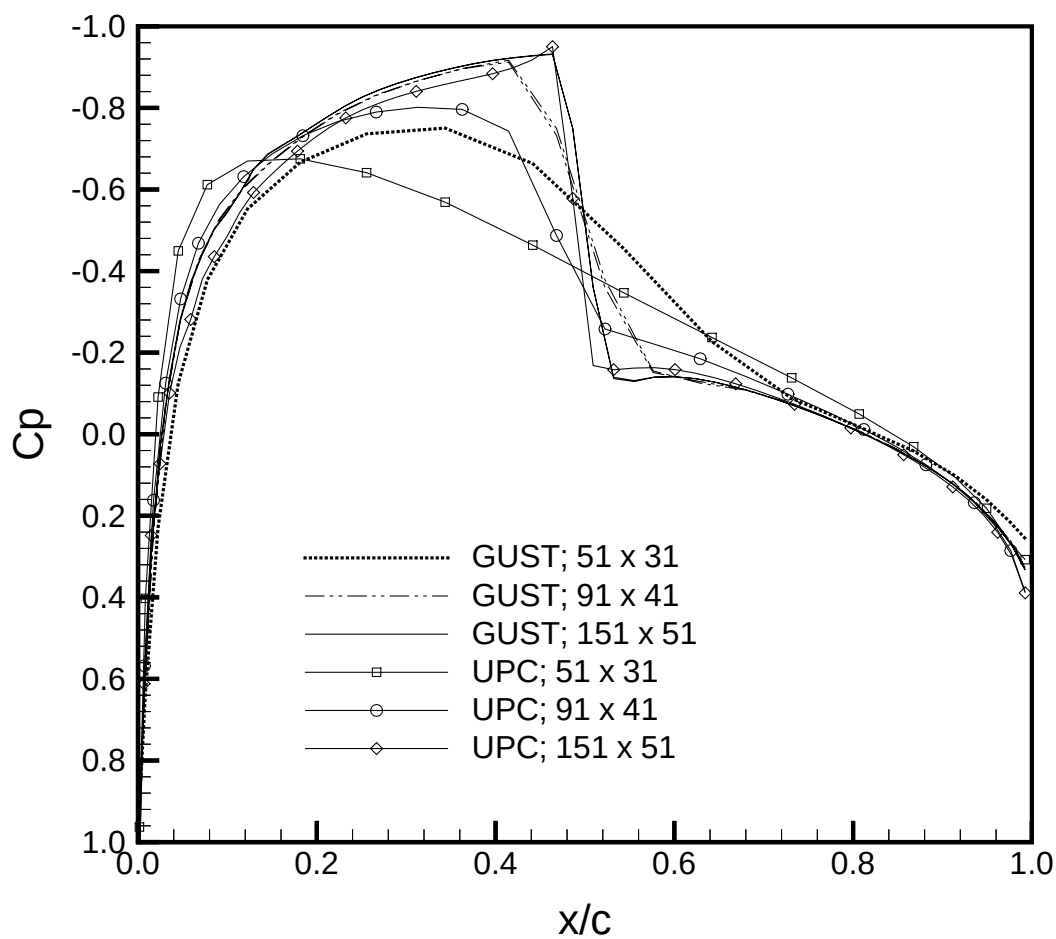


Figure 9.22: Comparison of *GUST* and UPC at three grid levels for NACA 0012. Flow conditions:  $0^\circ$  and  $M = 0.80$ .

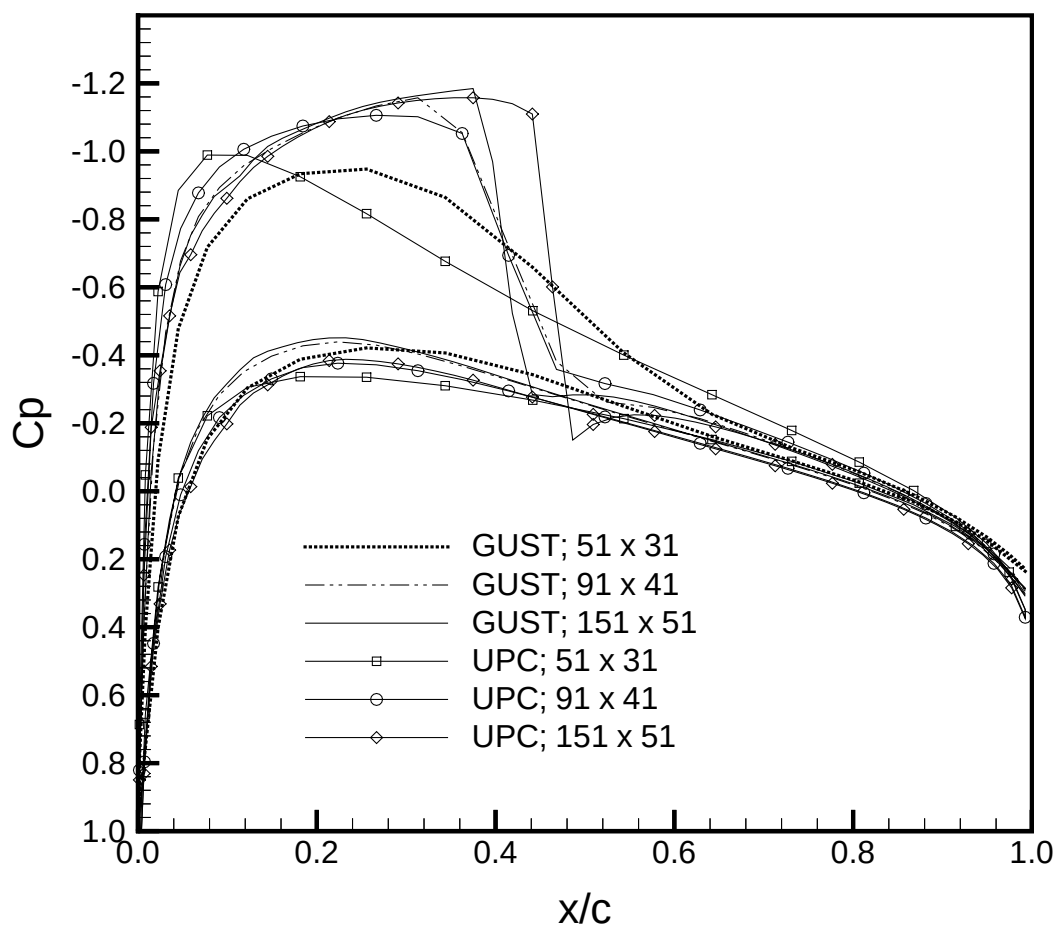


Figure 9.23: Comparison of *GUST* and UPC at three grid levels for NACA 0012. Flow conditions:  $1.5^\circ$  and  $M = 0.75$ .

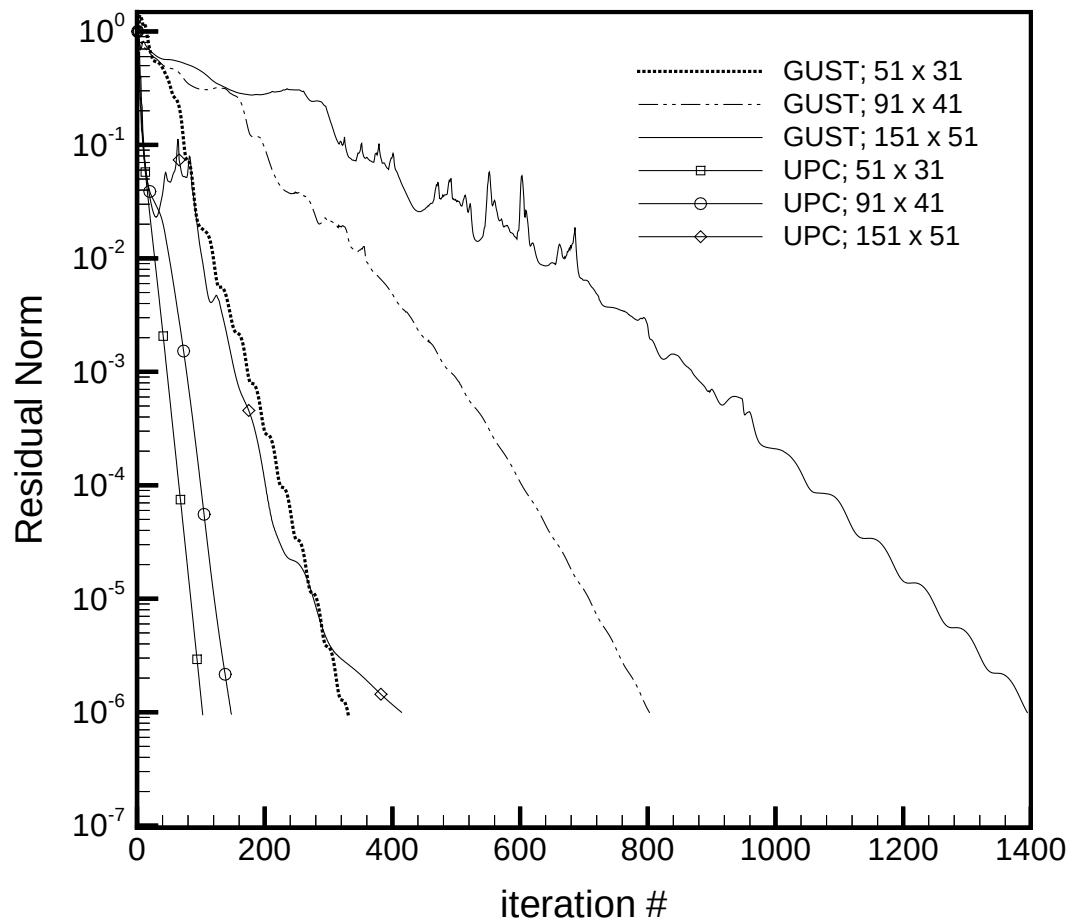


Figure 9.24: Normalized residual versus iteration number for NACA 0012 airfoil case at  $M = 0.80$  and  $0^\circ$ . Plot corresponds to Figure 9.22.

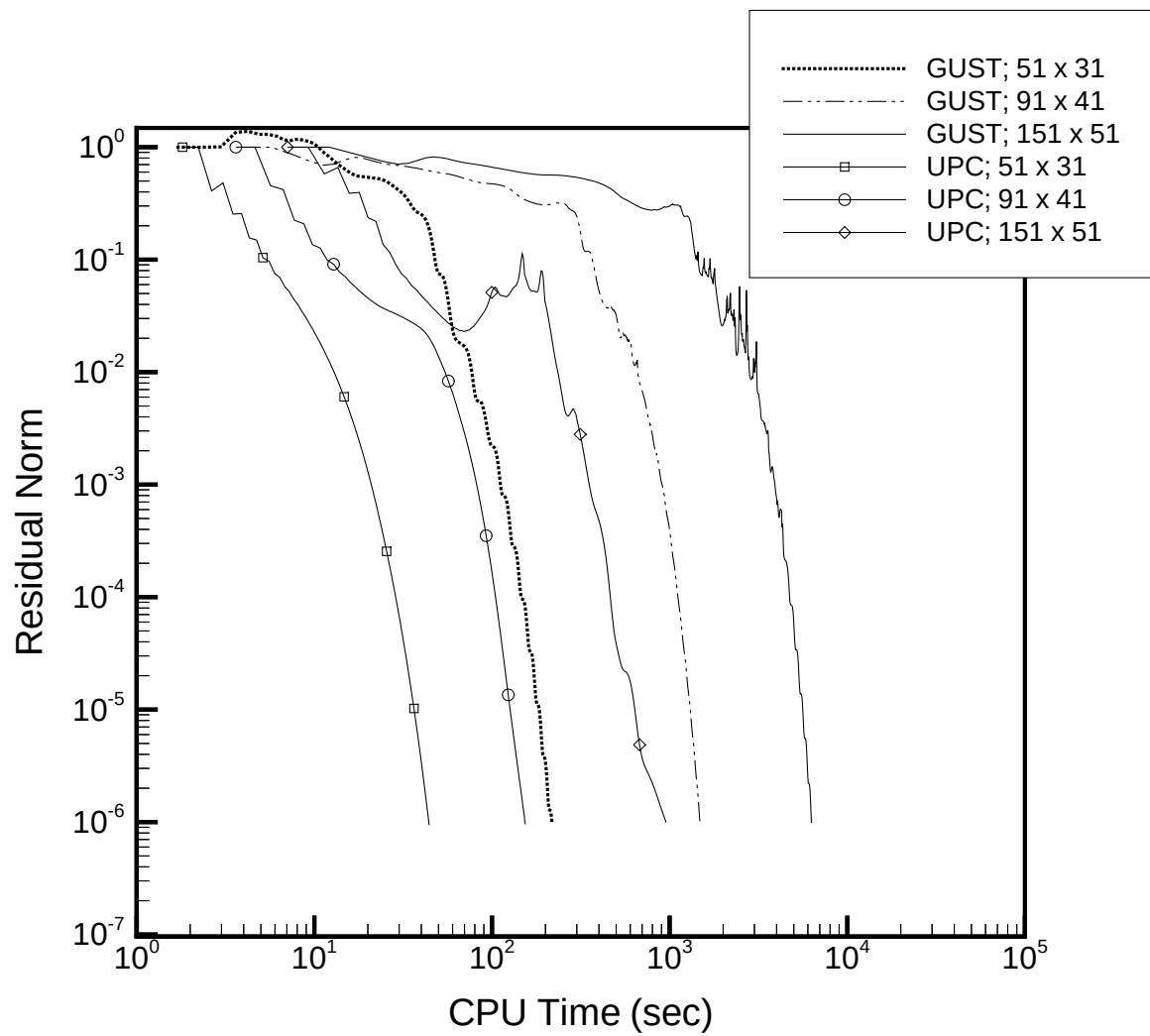


Figure 9.25: Normalized residual versus CPU time for NACA 0012 airfoil case at  $M = 0.80$  and  $0^\circ$ . Plot corresponds to Figure 9.22.

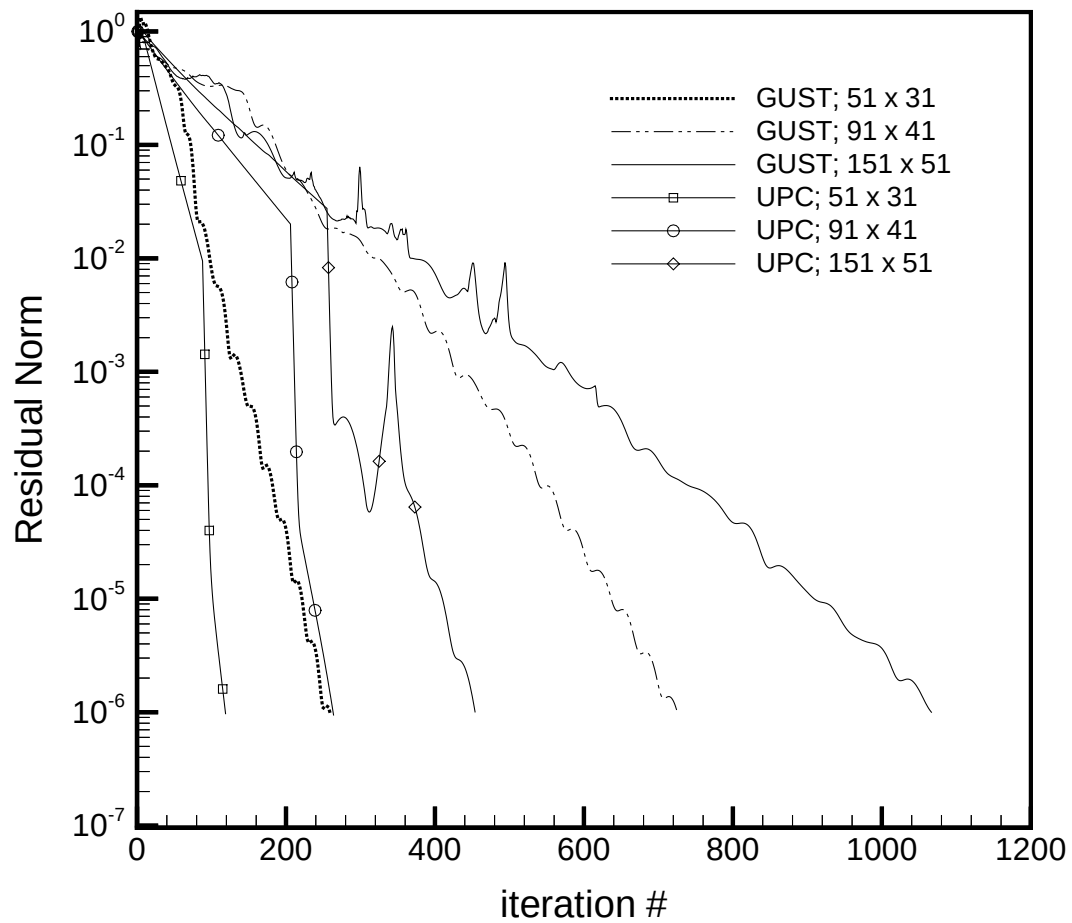


Figure 9.26: Normalized residual versus iteration number for NACA 0012 airfoil case at  $M = 0.75$  and  $1.5^\circ$ . Plot corresponds to Figure 9.23.

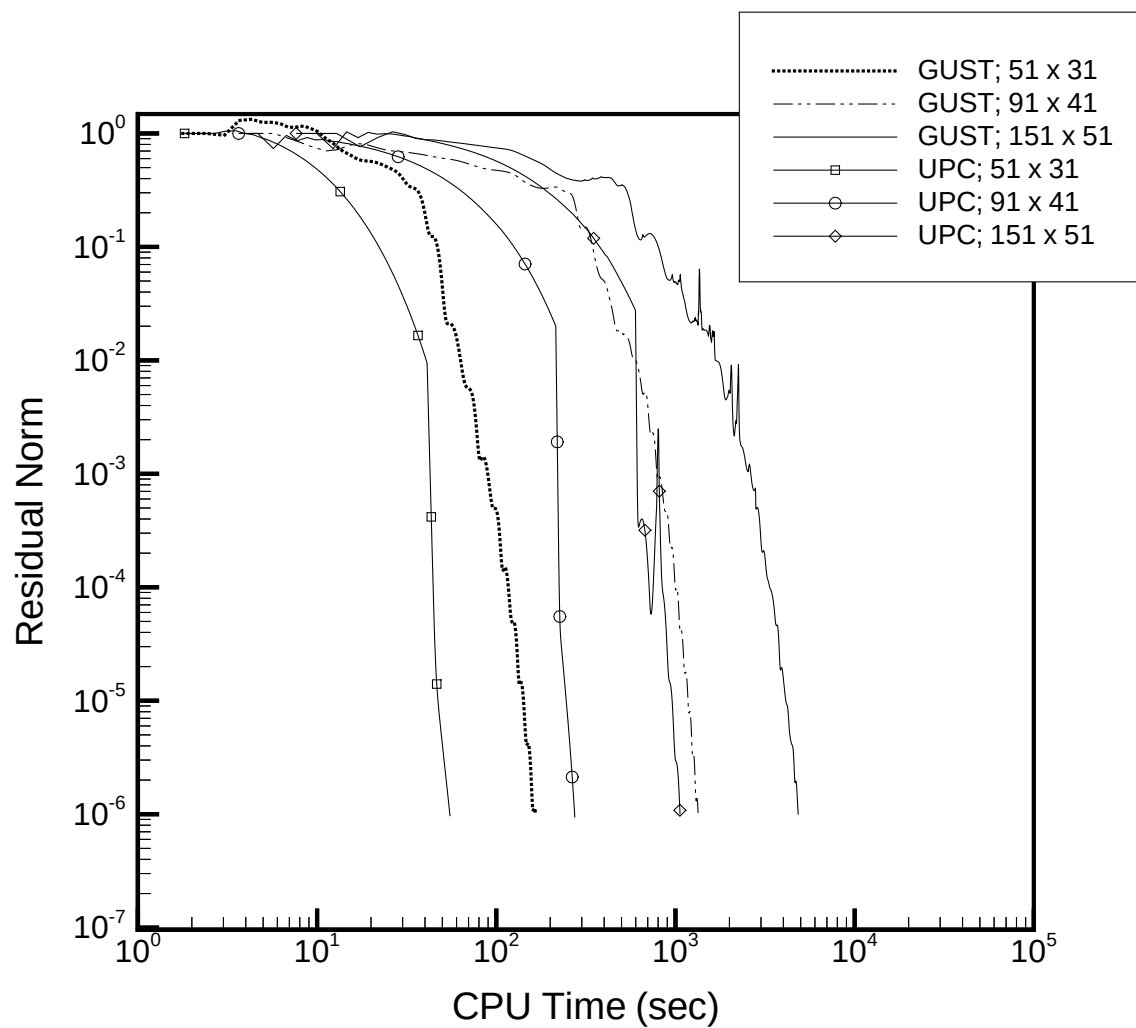


Figure 9.27: Normalized residual versus CPU time for NACA 0012 airfoil case at  $M = 0.75$  and  $1.5^\circ$ . Plot corresponds to Figure 9.23.

0012 airfoil at  $M = 0.75$  and  $1.5^\circ$  angle of attack. This flow condition was computed in the previous section, except here the grid will consist of triangles. Following this,  $C_p$  solutions will be given for the NACA 0012 airfoil at  $M = 0.50$  and  $4^\circ$ , the NACA 4412 airfoil at  $M = 0.50$  and  $3^\circ$ , and the LNV109A airfoil at  $M = 0.60$  and  $0^\circ$  and  $M = 0.50$  and  $2^\circ$ . All the solutions in this section will be computed on meshes with triangles.

Figure 9.28 shows the result of the grid convergence study for the NACA 0012 airfoil. Three unstructured grids were used for the study. The grids had 2,422, 3,752, and 4,992 cells, where the number of cells used to describe the airfoil surface was 169, 140, and 70 respectively. The following parameters were set for the density biasing calculations:  $C = 2.0$  and  $M_c = 0.95$ . In Figure 9.28, it can be observed that the shock location does not move very much as the grid is refined. With the unstructured grids, cells are easily clustered around the airfoil. This allows better shock resolution for a given number of cells in the grid. The  $C_p$  solution using the triangle grids produced a shock that was not as “sharp” as those observed on the quadrilateral grids. The maximum Mach number at the shock was 1.28 for the fine grid.

Figure 9.29 shows the same case, but with additional information. The solution from the fine grid is plotted with solutions from FLO36 and *GUST*. The *GUST* solution uses the same grid as UPC. Recall that FLO36 uses an internally generated grid based upon conformal mapping. The shock position from UPC is farther aft of the shock position from *GUST*, but not quite as far downstream as the FLO36 shock.

The medium grid (3,752 cells) for the NACA 0012 airfoil is shown up close in Figure 9.30. This grid was also used to compute the pressure distribution at freestream conditions of  $M = 0.50$  and  $4^\circ$ . The solution is plotted in Figure 9.31 for *GUST*, UPC, and FLO36. The *GUST* solution is performed on the same grid as UPC. The solution from UPC and *GUST* are very close in agreement on the upper surface. The peak values in  $C_p$  are almost identical. On the lower surface, *GUST* and FLO36 correlate the best.

The grid used for the NACA 4412 case is given in Figure 9.32. The grid contains 3,678 cells with 140 cells on the airfoil surface. The corresponding pressure solution is

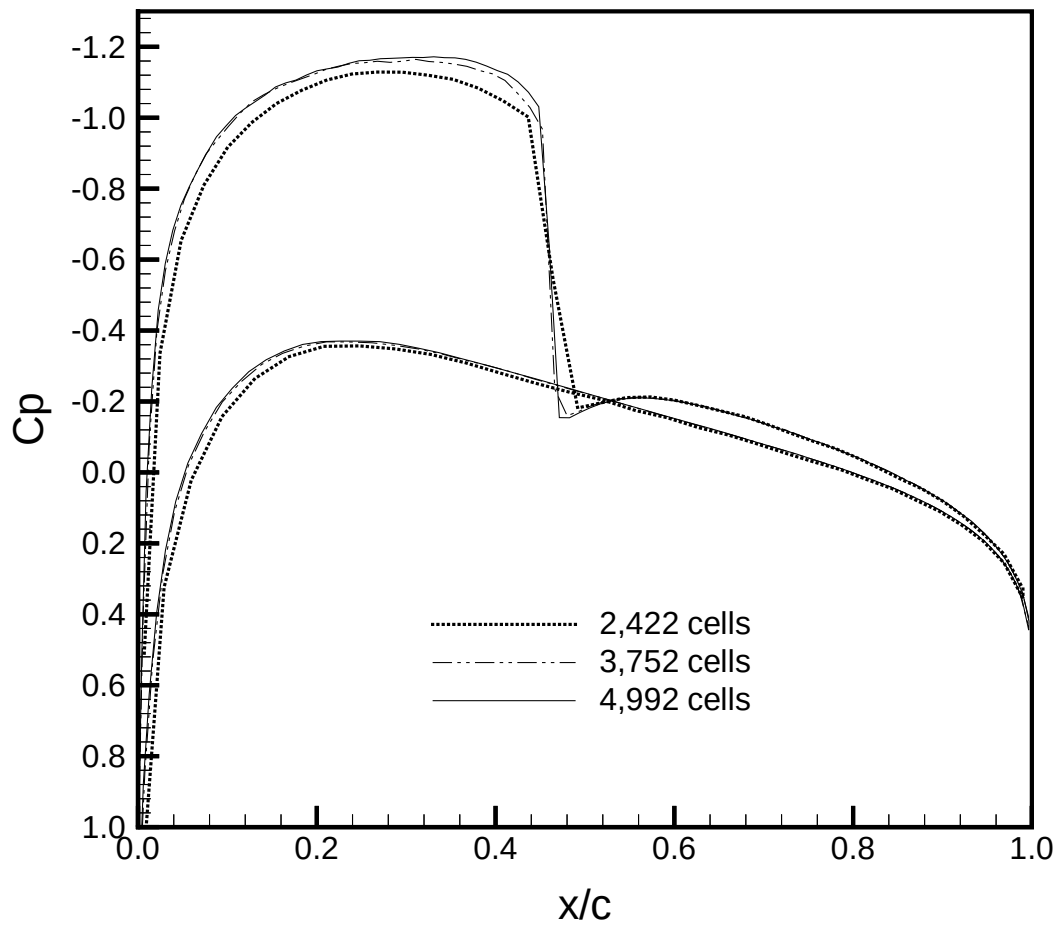


Figure 9.28: Grid convergence for the potential flow solver on grids with triangles for a NACA 0012 airfoil at  $1.5^\circ$  and  $M = 0.75$ .

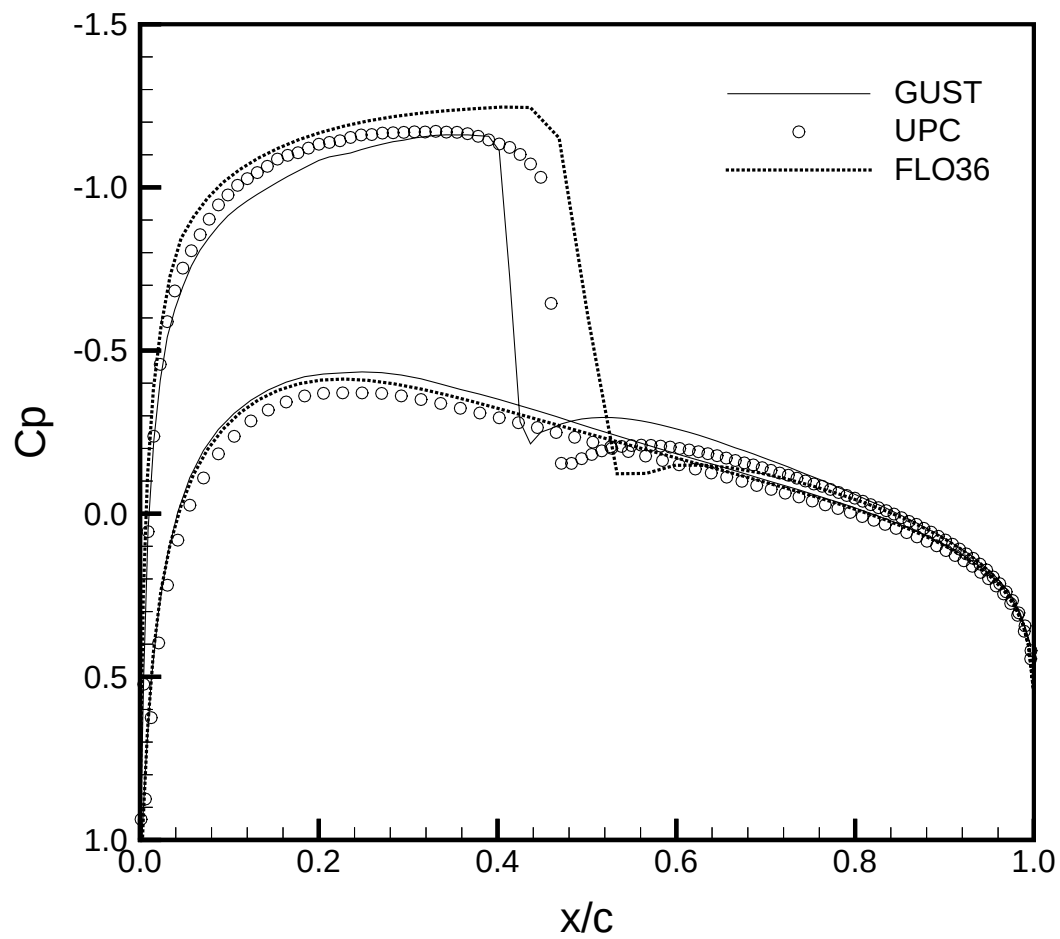


Figure 9.29:  $C_p$  comparison of Euler and potential flow solvers for a NACA 0012 airfoil at  $1.5^\circ$  and  $M = 0.75$ .

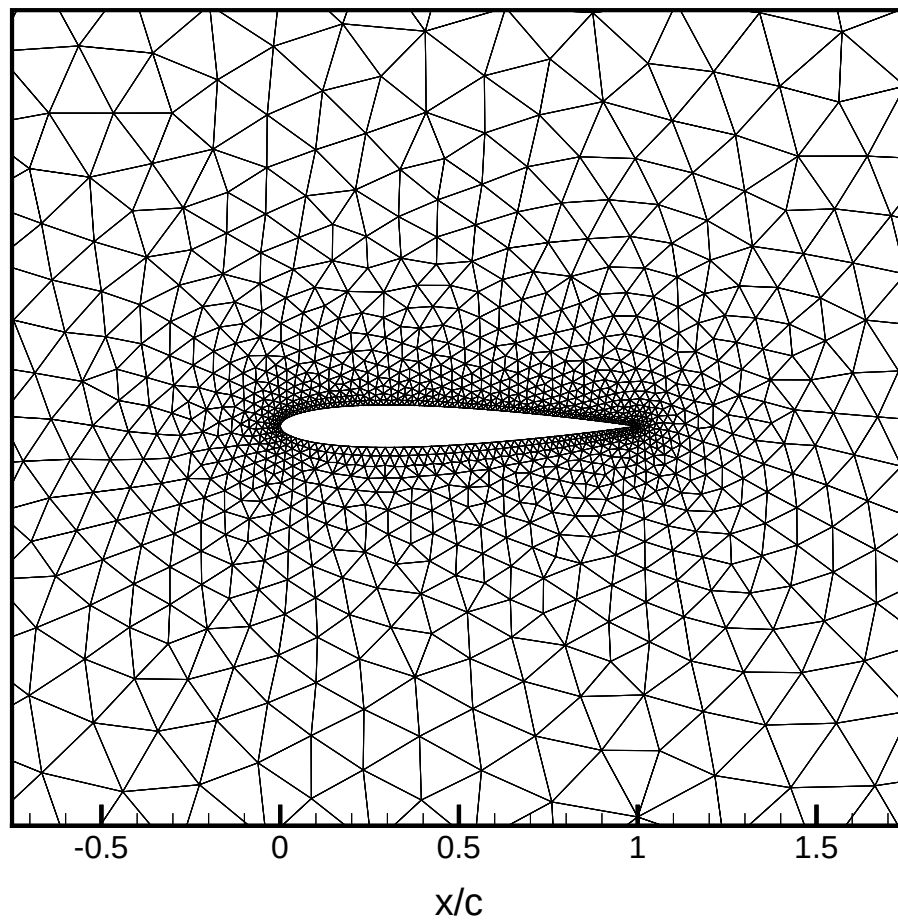


Figure 9.30: Unstructured grid used for solutions to the NACA 0012 airfoil. Grid contains 3,752 cells with 140 cells on airfoil surface.

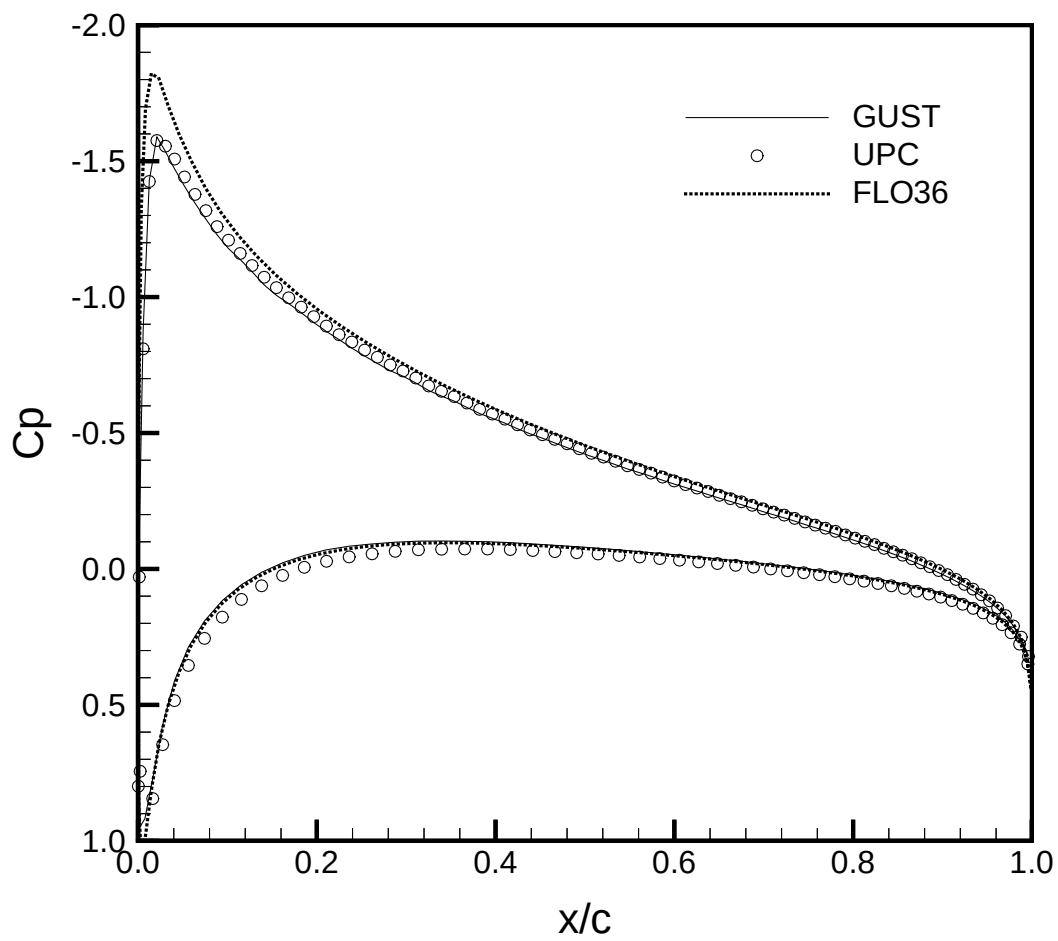


Figure 9.31:  $C_p$  results for a NACA 0012 airfoil at  $4^\circ$  and  $M = 0.50$ . UPC and *GUST* use a grid consisting of 3,752 cells.

shown in Figure 9.33 with FLO36 and *GUST* for comparison. On the upper surface, the UPC code predicts a higher velocity when compared to *GUST*, but is not quite as large as the FLO36 prediction. Since the flow is subsonic, the solutions from each flow solver should be identical. Any differences are most likely to be from the discretization approaches in each of the codes.

The last airfoil to be presented is the LNV109A. A view of the airfoil grid is given in Figure 9.34. The grid consists of 3,955 cells, where 157 cells are on the airfoil surface. The two  $C_p$  solutions for this airfoil are shown in Figures 9.35 and 9.36. The first figure is at subsonic conditions ( $M = 0.50$ ,  $2^\circ$ ), while the second is transonic ( $M = 0.60$ ,  $0^\circ$ ). Due to the geometry of the airfoil, only the *GUST* solution is presented with the unstructured potential code. For the subsonic case shown in Figure 9.35, the Euler and potential solutions are almost identical. The largest difference in the solutions is on the lower surface where a peak occurs in the velocity right after the stagnation point. The solutions usually differ the most in high gradient regions, which occurs here on the lower surface. For the transonic case, Figure 9.36, very good agreement is also observed between the two flow solvers. The shock position is nearly identical, except for the sharper shock with the Euler solver. The maximum Mach number in the potential flow solution is 1.35.

## 9.8 ONERA M6 Wing

For a three dimensional case, the ONERA M6 wing is computed. The wing is calculated at Mach 0.84 and  $3.06^\circ$  angle of attack. Results are given for the UPC and TOPS codes, as well as the experimental data [1] for this flow condition. The same grid is used for both the TOPS and UPC codes, and consists of hexahedrals. The dimension of the grid is  $113 \times 31 \times 17$ .

The pressure coefficient at six locations along the span are presented in Figures 9.37 through 9.42. These locations correspond to 20%, 44%, 65%, 80%, 90%, and 95% of the semi-wing span. Several observations can be made from the six pressure plots. The shock location is always predicted earlier for the UPC code versus the TOPS code (except for the 95% station). This trend was also observed in the airfoil cases when

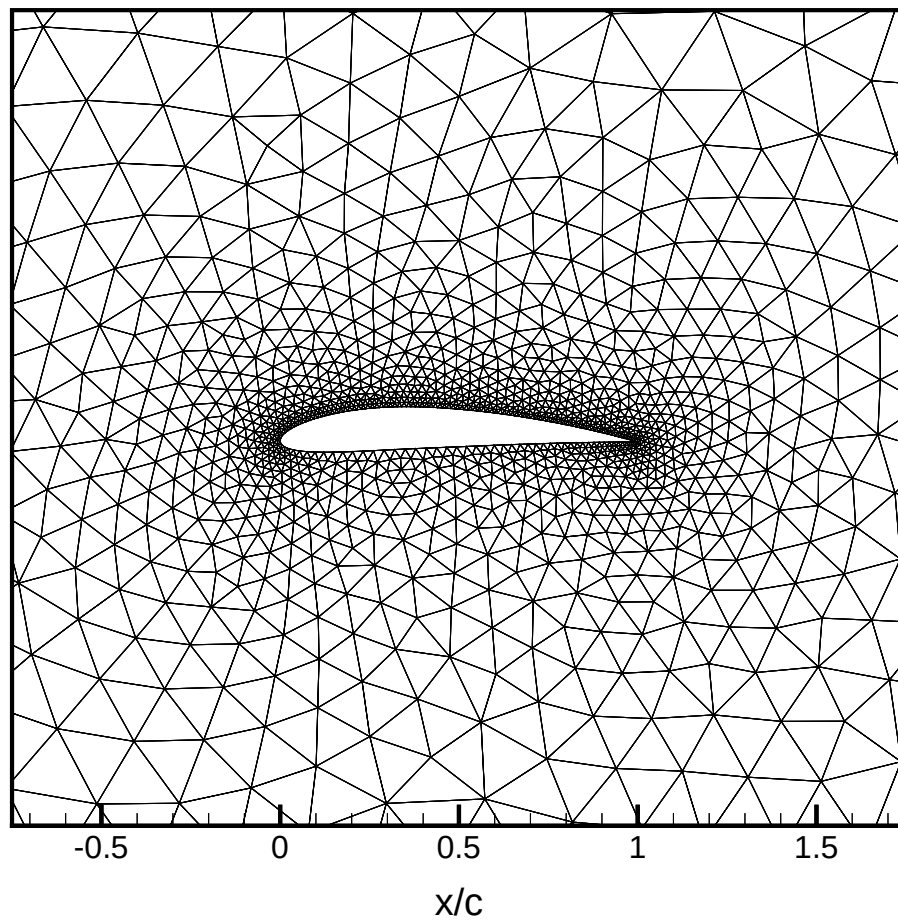


Figure 9.32: Unstructured grid used for solutions to the NACA 4412 airfoil. Grid contains 3,678 cells with 140 cells on airfoil surface.

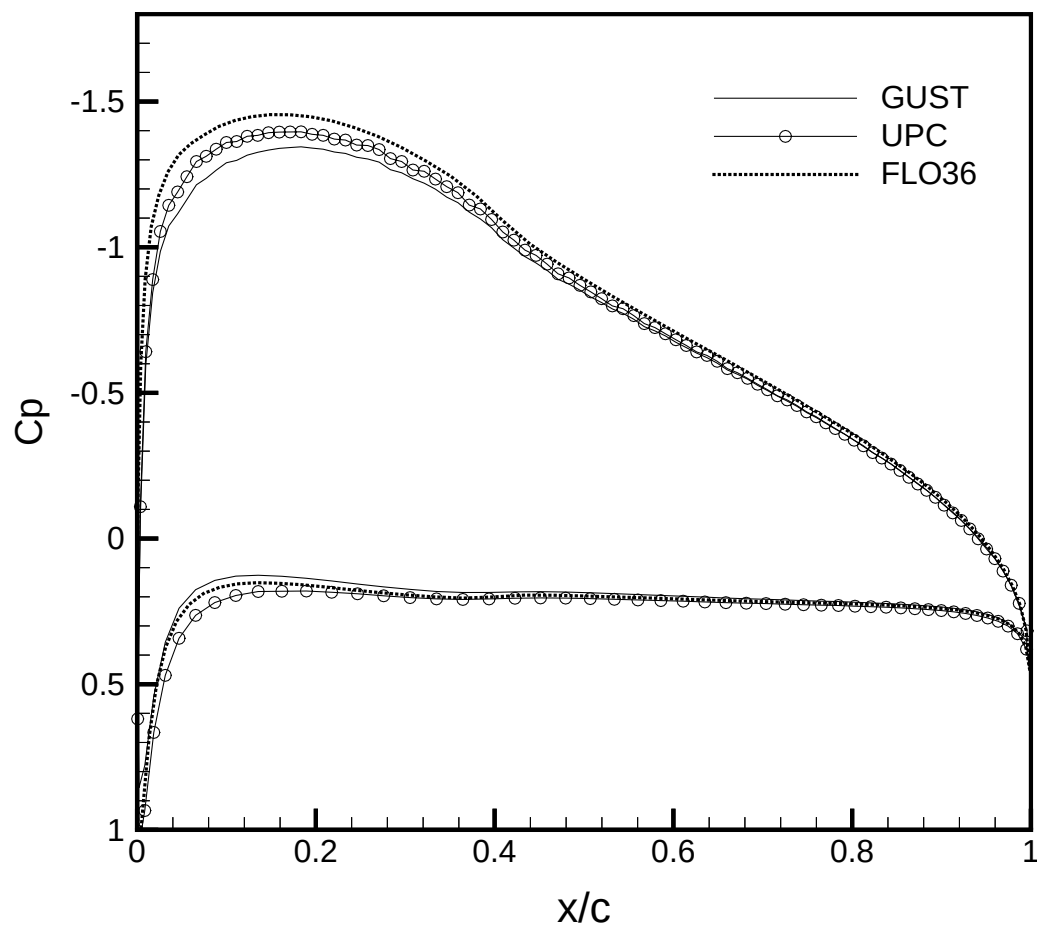


Figure 9.33:  $C_p$  results for NACA 4412 airfoil at  $3^\circ$  and  $M = 0.50$ . Grid contains 3,678 triangular cells with 140 cells on surface for the UPC and *GUST* solutions.

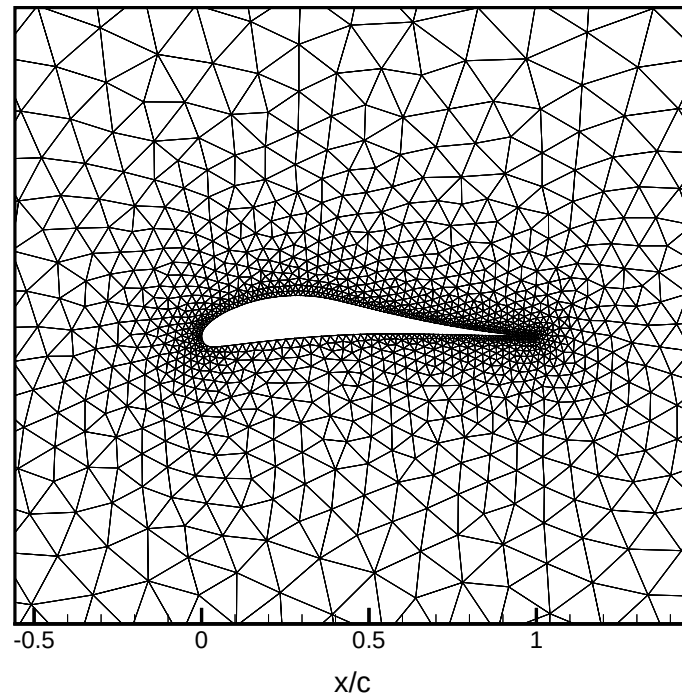


Figure 9.34: Unstructured grid used for solutions to the LNV109A airfoil. Grid contains 3,955 cells.

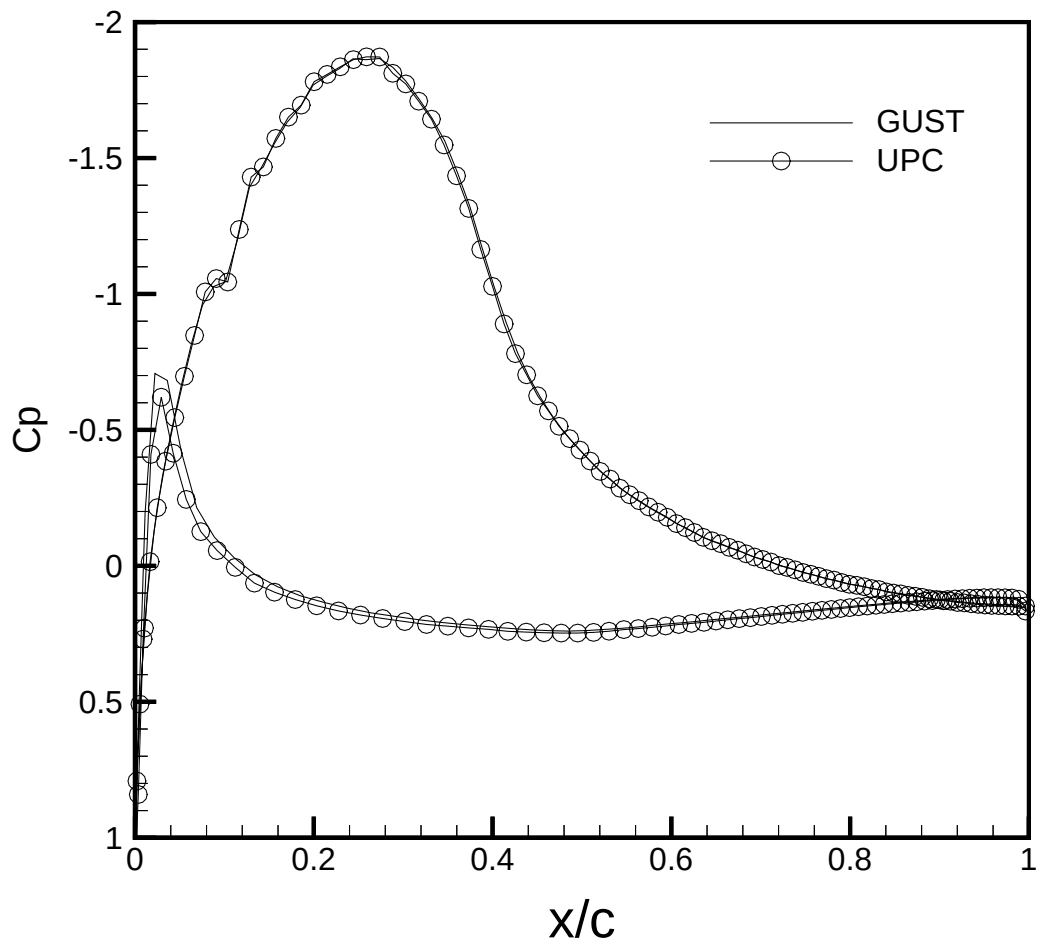


Figure 9.35:  $C_p$  results for LNV109A airfoil at  $2^\circ$  and  $M = 0.50$ .

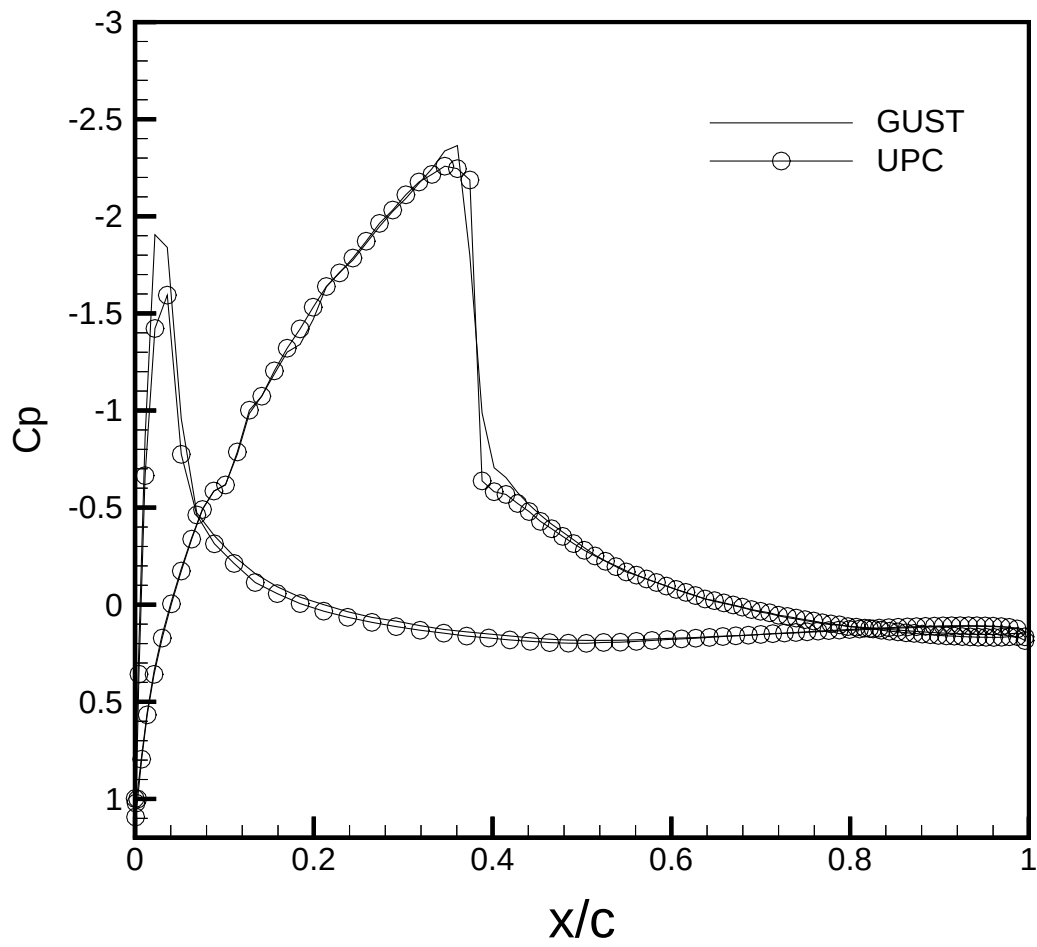


Figure 9.36:  $C_p$  results for LNV109A airfoil at  $0^\circ$  and  $M = 0.60$ .

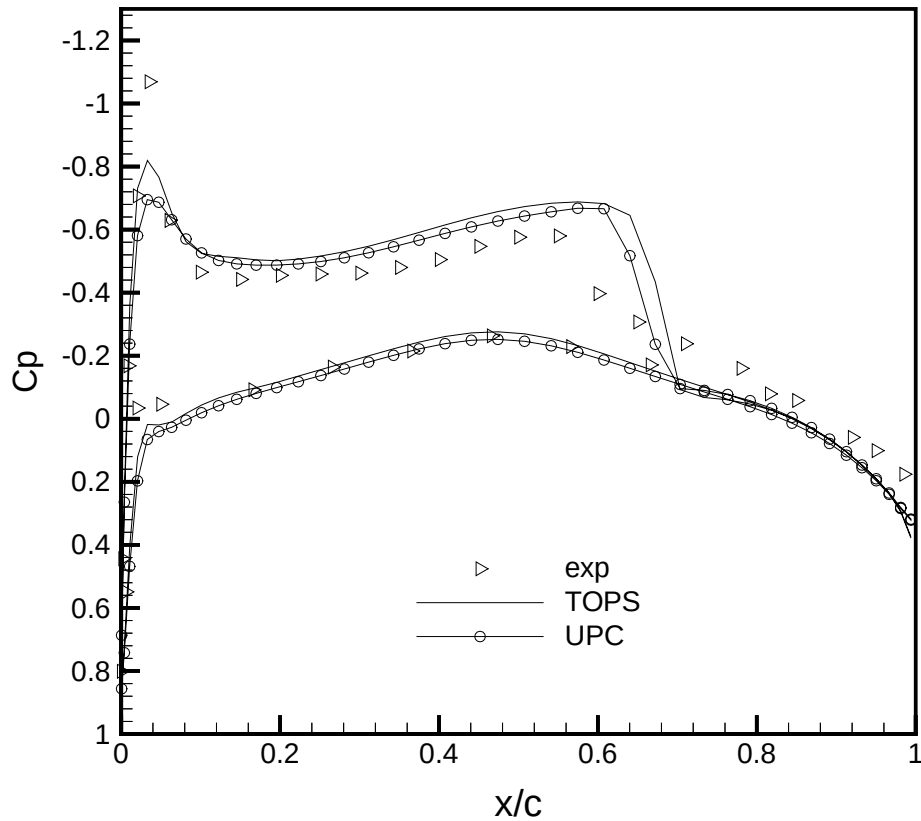


Figure 9.37: Results for ONERA M6 wing at 20% semi-span. Flow conditions:  $M = 0.84$  at  $3.06^\circ$  angle of attack.

compared to the TOPS code. The peaks in  $C_p$  are better captured with the TOPS code, and the velocity is slightly larger on the upper surface of the wing compared to UPC. Both potential codes only predict the single shock for the outboard locations, while experiment shows a double shock. This may be due to the grid, since a fine grid was not used for this test case. The goal here is to compare the present code with the TOPS code. For the lower wing surface, all three data sets correlate well. Overall, the UPC code is able to perform well against the TOPS code in comparing to the experimental data.

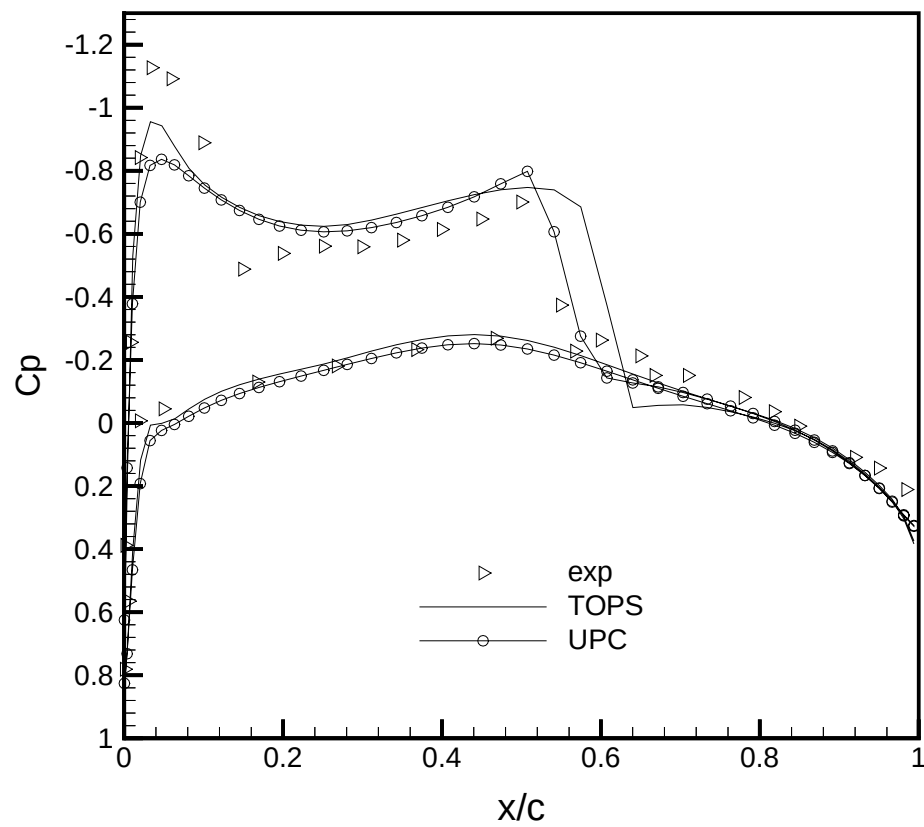


Figure 9.38: Results for ONERA M6 wing at 44% semi-span. Flow conditions:  $M = 0.84$  at  $3.06^\circ$  angle of attack.

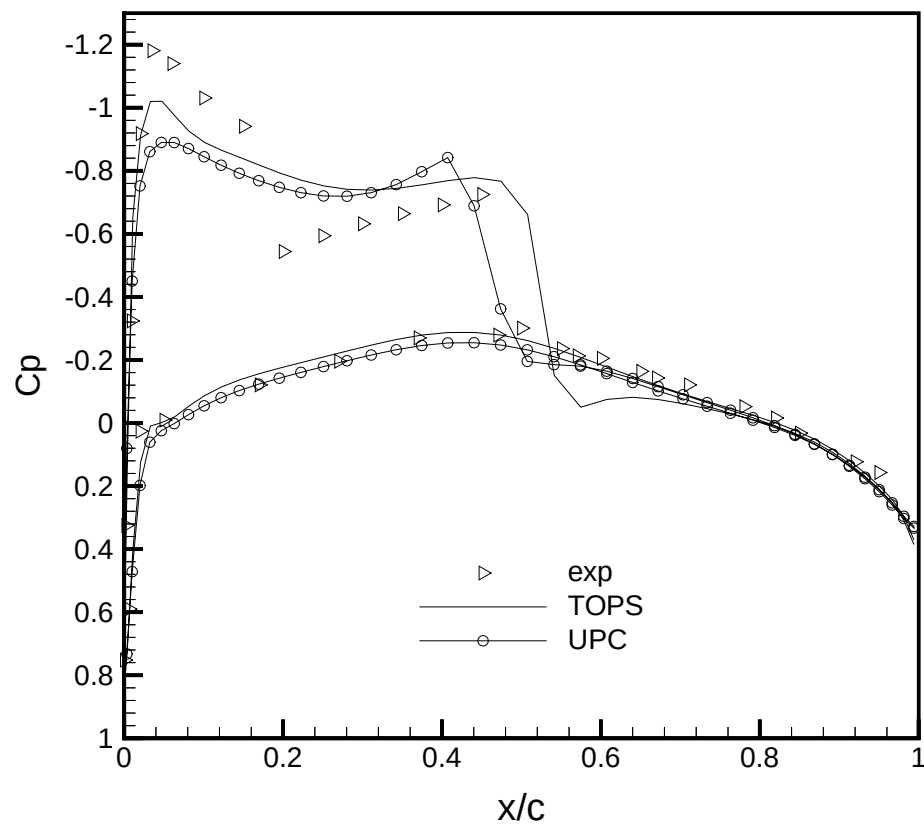


Figure 9.39: Results for ONERA M6 wing at 65% semi-span. Flow conditions:  $M = 0.84$  at  $3.06^\circ$  angle of attack.

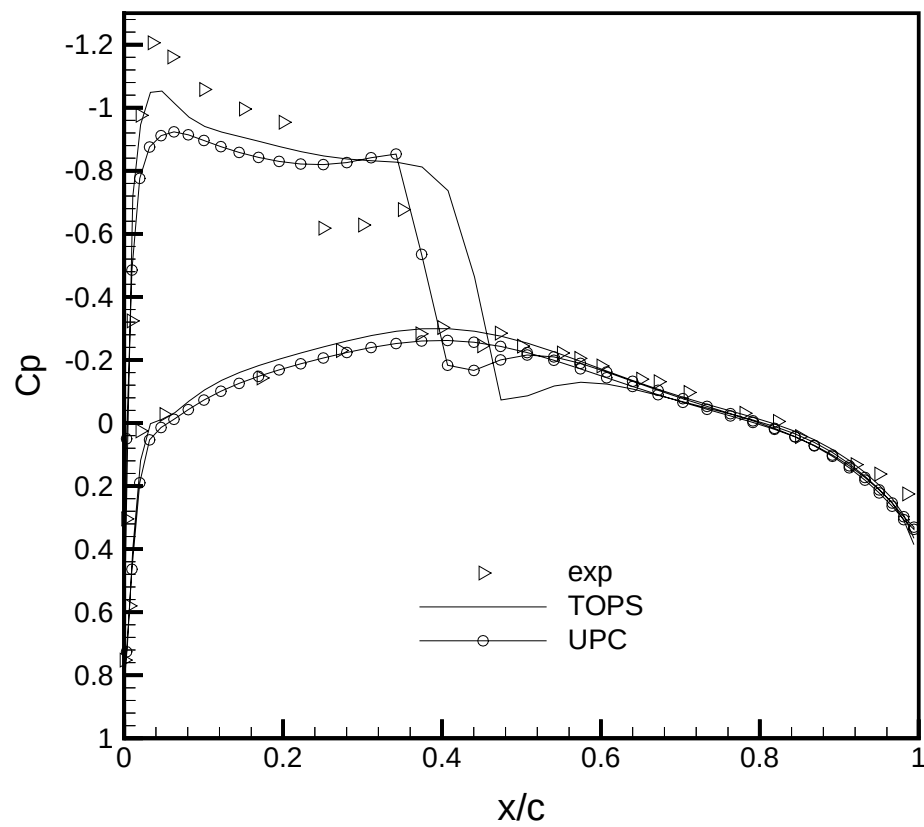


Figure 9.40: Results for ONERA M6 wing at 80% semi-span. Flow conditions:  $M = 0.84$  at  $3.06^\circ$  angle of attack.

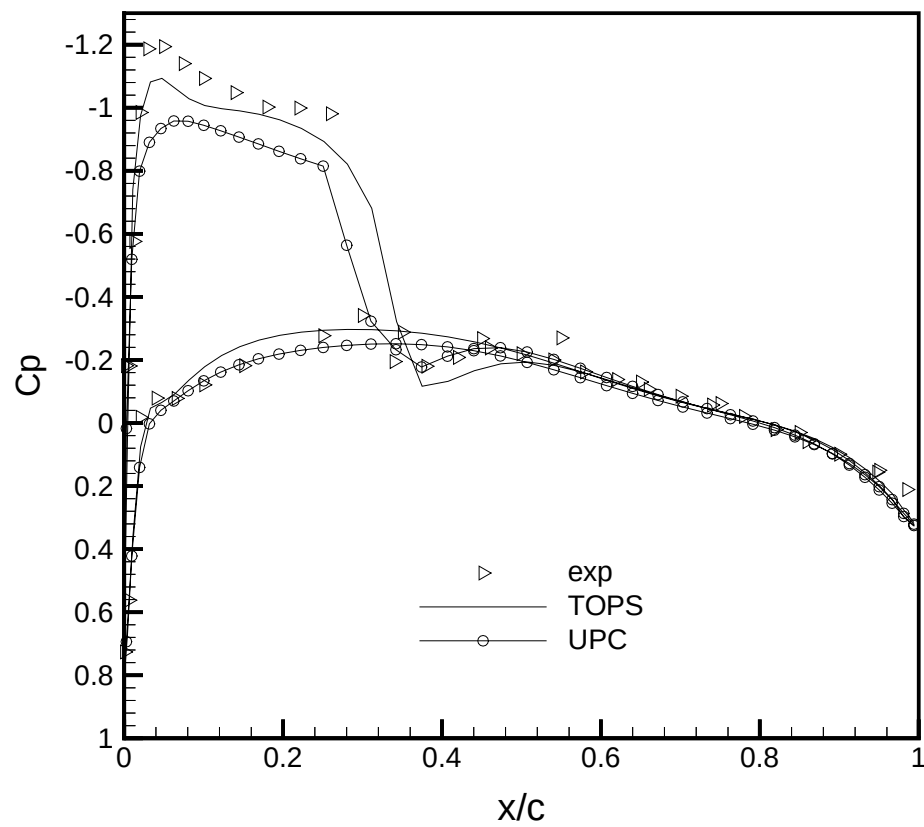


Figure 9.41: Results for ONERA M6 wing at 90% semi-span. Flow conditions:  $M = 0.84$  at  $3.06^\circ$  angle of attack.

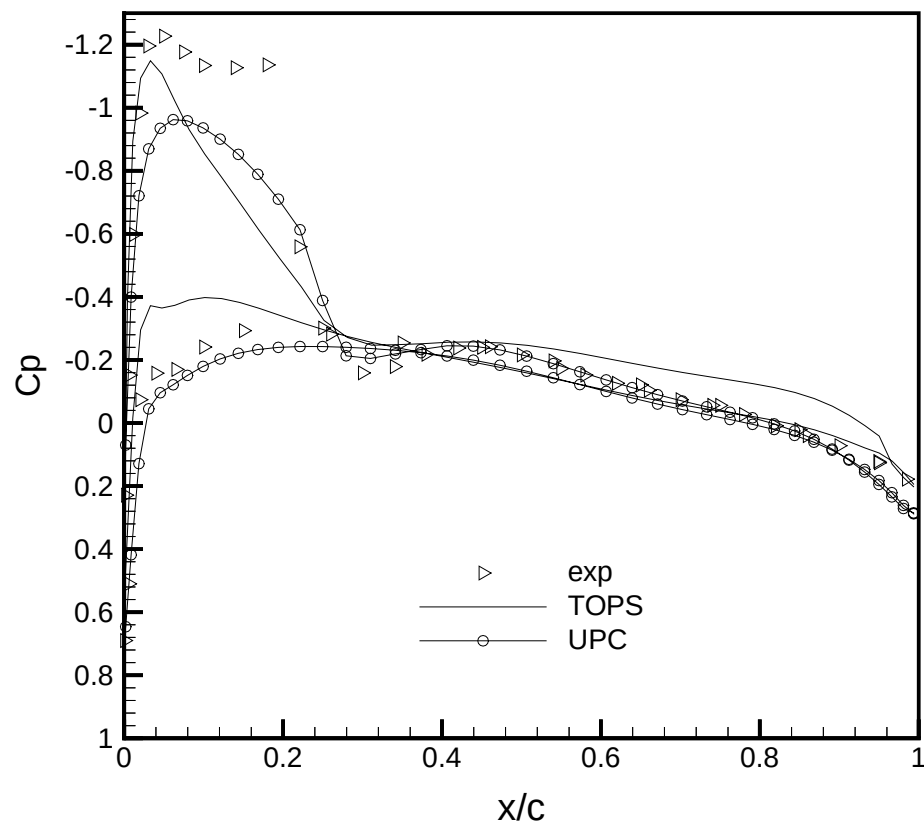


Figure 9.42: Results for ONERA M6 wing at 95% semi-span. Flow conditions:  $M = 0.84$  at  $3.06^\circ$  angle of attack.

# Chapter 10

## Conclusions

Numerical computations were presented for the incompressible, Reynolds Averaged Navier-Stokes equations and the full potential equation. Both sets of equations have their origin with the Navier-Stokes equations. The Navier-Stokes equations represent the governing equations for fluid mechanics. Due to the complexity of the complete Navier-Stokes equations, approximations are made to solve the equations. Depending upon the application, different levels of approximations are used.

The incompressible Navier-Stokes equations can be used when density is essentially constant throughout the flow field. This occurs in flows with low Mach numbers, or flows involving water as the fluid medium. For the Reynolds Averaged equations, viscous effects are captured using a turbulence model. Depending upon the turbulence model, the numerical results may vary greatly when regions of strong adverse pressure gradients or separation exist. The Johnson and King turbulence model was used because of its success in modeling turbulent separated flows. Results for the model were given for flow in a converging-diverging diffuser and the MIT flapping foil experiment. These cases were computed in both a steady and unsteady flow environment.

Some changes were also made to the turbulence model in an attempt to improve the modeling performance by taking into account some physical observations from experiment. The pressure gradient and the normal stress production of energy were incorporated into the Johnson and King model. A model was also formulated for the

apparent normal stress that results from Reynolds Averaging.

The Johnson and King model compared well with experiment for both the diffuser and flapping foil test cases. In predicting the flow detachment or separation location, the model predicted early detachment for the diffuser case, but agreed with experiment for the FFX case. For the steady flow cases, the boundary layer profiles captured the flow physics, giving a good correlation with experimental data. For the unsteady flow cases, good agreement was again observed for the FFX case, but some difficulties appeared for the diffuser flow. The computational model for the unsteady diffuser case was not able to match the experimental conditions, giving some discrepancies in the results.

These same cases were run with the modifications to the Johnson and King model. The flow detachment location was impacted by the changes made to the model. For the diffuser case, the separation location was in better agreement with experiment, while for the flapping foil case the agreement was worse. The boundary layer profiles remained very favorable for the diffuser case, matching up well with the experimental data. For the FFX case, the modifications had a greater impact on the profiles, especially near the wall. The near wall region had more influence from the pressure gradient, which increased the flow velocity near the wall. For the unsteady flow cases, the modifications made the model more sensitive to the unsteady flow field. This tended to give better near wall performance for the diffuser case for the acceleration phases, and a slight phase lag when the flow decelerated in the flow cycle. In the FFX case, the mean boundary layer profiles did not have as good of an agreement with experiment.

The time dependent term that appears in the Johnson and King model was studied for the unsteady flow cases. Computations were performed both with and without the time dependency in the model and then compared. The results showed that for the boundary layer profiles, no noticeable changes occurred for either test case. For the flapping foil case, amplitude and phase data was computed using a Fourier analysis on the boundary layer profiles. Changes were seen in this data, which showed that the time dependency in the Johnson and King model had some impact on the phase and amplitude. The data had better agreement with experiment when the time

dependency remained in the model.

For the normal stress model which appears in the Reynolds averaged  $x$ -momentum equation, a new model was formulated. Both the diffuser and FFX cases were calculated with the new model. The changes made to the turbulence modeling did not have any significant impact on the computational results for either test case. It appeared that the eddy viscosity model for the apparent Reynolds shear stress term dominated the turbulence modeling, and prevented the new normal stress model from making any significant contributions to the mean flow field.

The work presented here for turbulence modeling is intended to be another step in the process of improving turbulence models in CFD. The Johnson and King turbulence model was demonstrated for two test cases involving strong adverse pressure gradients and separation. The model was shown to perform well for these flows, giving confidence to use the model for more complicated geometries. The changes made to the model were an attempt to incorporate more of the flow physics of separated flow into the turbulence modeling. The changes did not necessarily improve the results in every situation or case. It was shown that the modifications impacted the flow detachment location, which is important for determining the overall drag calculation. The results from the modifications also indicate that the Johnson and King model still has room for improvements, and the changes that were made to the model reveal several areas that improvements could be implemented. When performing numerical computations, it is important to keep in mind that the turbulence model may or may not perform well under the given flow conditions. With this in mind, the modifications presented here may give significant changes in the results compared to the original Johnson and King model. Such changes in the results may indicate areas in which the results may need further investigation. The ideas presented here may also serve as a starting point for other researchers to build upon, since the quest for better turbulence models is still in progress.

The other level of approximation that was addressed here was the full potential equation. The Navier-Stokes equations are reduced down to a single equation, using the assumptions that the flow is inviscid and irrotational. The full potential equation is useful for design and analysis of airfoils and wings. Computations can usually be

performed much quicker than those solving the Euler or Navier-Stokes equations. The full potential equation can be used for transonic flows, where a lot of design issues are of interest.

An algorithm for solving the full potential equation on unstructured grids was developed. The algorithm solves the full potential equation for both subsonic and transonic flow conditions. The use of unstructured grids has become popular for Euler and Navier-Stokes flow solvers. Unstructured grids allow more flexibility in modeling complex geometries, as well as grid adaptation. Due to the advantages of unstructured grids, the concept was applied to the full potential equation. This is a new area for finite volume full potential solvers, and should provide a starting point for future research in this area.

Several schemes were studied for computing the potential gradient, which was necessary for calculating the velocity vector. The scheme that was found to work best used the K-exact concept. Using the K-exact approach, a second order polynomial was fitted through a given number of cells. Once the coefficients of the polynomial are computed, the gradients are known as well. To make the scheme applicable to unstructured grids, the number of cells in a stencil must be able to vary. This was accomplished by combining K-exact with the least squares method. The resulting scheme is second order accurate and applicable to any cell type.

The potential equation was solved implicitly using the GMRES algorithm. In order to help improve the convergence rate of GMRES, the linear system was preconditioned. This was accomplished by reordering the cells in the grid such that more terms in the Jacobian matrix were near the main diagonal. The preconditioning greatly improved the performance of the code. Other critical areas of the algorithm were also discussed, including the Kutta condition and density biasing. The Kutta condition is implemented to allow for lifting airfoils and wings, and the density biasing scheme is necessary for transonic flows.

Results from the full potential code were given for several airfoil cases and the ONERA M6 wing. The potential solution was compared with Euler calculations, as well as other structured full potential flow solvers. The results from the unstructured algorithm were favorable. The full potential solver was shown to be at least three

times faster than an unstructured Euler flow solver. In addition to computer time, the potential results gave good estimates of the shock location compared to Euler. The shock location was always in agreement with Euler, or slightly downstream which is justified by the isentropic condition. Compared to other potential flow solvers, high gradient regions were not captured as well with the unstructured flow solver versus the other potential solvers. For example, the flow around the leading edge of an airfoil at angle of attack produces a peak in velocity, which is always greater for the structured flow solvers. The main reason for this is how the gradients are computed. This also impacts the shock location. The unstructured solver always had the shock positioned further upstream over the other potential solutions compared to. This is not a disadvantage of the algorithm since the physical shock is almost always further upstream than what a potential code predicts.

Both the full potential equation and the incompressible Navier Stokes equations are of great interest in engineering design and analysis. The research presented here is an effort to improve numerical computations of these equations. A discussion of these equations, as well as sample applications, has been given. The results demonstrate the usefulness and capabilities of CFD in fluid mechanics.

# Appendix A

## Details of the JKM Differential Equation

The assumptions made in deriving the differential equation used in the baseline Johnson & King model will be discussed here. For comparison with the baseline model, the assumptions made to improve the model will also be summarized.

The partial differential equation, PDE, is used to attain the value for the maximum Reynolds shear stress,  $\tau_m$ , at each streamwise location in the flow field. The value of  $\tau_m$  from the PDE is considered to be the actual value of the Reynolds shear stress. The PDE is derived from the turbulence kinetic energy equation. More specifically, the equation is taken along the path of maximum kinetic energy. For an incompressible boundary layer, the T.K.E. equation for maximum kinetic energy is given as,

$$\frac{\partial k_m}{\partial t} + \bar{u}_m \frac{\partial k_m}{\partial x} = (-\overline{u'v'}) \left[ \frac{\partial u}{\partial y} \right]_m - \frac{\partial}{\partial y} \left[ \frac{\overline{p'v'}}{\rho} + \frac{1}{2} \overline{q^2 v'} \right] - \epsilon \quad (\text{A.1})$$

The first term in the above equation is the T.K.E. time derivative. If the flow were considered steady, then this term would disappear and the PDE would reduce to an ordinary differential equation. The other terms in the equation, from left to right, are convection, production, diffusion, and dissipation. Right away it is clear that every term, except for the production term, will need to be modeled using experimental observations. The production term contains  $\tau_m$ , and so there appears to be no need to model this quantity. But as shown in the following paragraphs, some modeling of

this term will have to be done as well.

The first assumption is for modeling the kinetic energy, which relates it to the maximum Reynolds shear stress. The ratio of  $\tau_m$  to  $k_m$  is considered to be a constant, as expressed below.

$$(1) \quad \frac{\tau_m}{k_m} = a1$$

This substitution is made into the time derivative and the convection part of the T.K.E. equation above. The second assumption models the dissipation term. The following substitution is directly made.

$$(2) \quad \epsilon = \frac{(\tau_m)^{\frac{3}{2}}}{L_m}$$

It is important to note that this assumption also involves a length scale which must be computed as well. The next assumption involves the gradient in the production term. For flows where production and dissipation are the main contributors to the T.K.E., the mean velocity gradient multiplied by the length scale would represent the square root of the turbulent shear stress. Since this is found in equilibrium type flows, the Reynolds stress is represented as  $\tau_{m,eq}$ . This substitution into the PDE is given as,

$$(3) \quad L_m \frac{\partial U}{\partial y} = (\tau_{m,eq})^{\frac{1}{2}}$$

The only remaining term is the diffusion. It is still unclear how to model this term, and more research needs to be done to provide an accurate model for the diffusion. The substitution that Johnson and King made is given here, and it is noted that no other improvements have been made since then.

$$(4) \quad D_m = \frac{C_{dif}(\tau_m)^{\frac{3}{2}}}{a_1 \delta [0.7 - (\frac{y}{\delta})_m]} |1 - \sigma(x)^{\frac{1}{2}}|$$

This completes all the assumptions and substitutions that were made to arrive at the baseline Johnson and King model PDE. The equation with all substitutions made takes the final form below.

$$\frac{1}{a_1} \frac{\partial \tau_m}{\partial t} + \frac{u_m}{a_1} \frac{\partial \tau_m}{\partial x} = \frac{\tau_m (\tau_{m,eq})^{\frac{1}{2}}}{L_m} - D_m - \frac{(\tau_m)^{\frac{3}{2}}}{L_m} \quad (A.2)$$

This equation is equivalent to (4.9).

In the same manner as above, the assumptions made in modifying the Johnson

& King model will be presented. These assumptions are based on experimental observations as well, with the differences being in that they apply more specifically to strong adverse pressure gradient and separated flows.

It has been observed that the normal shear stress production is no longer negligible compared to the shear stress production when the flow is near detachment. The first assumption can be expressed as the following definition,

$$(1) \quad F = \frac{\text{Total Production}}{\text{Shear Stress Production}}$$

$F$  is computed using (4.26), and will be evaluated at the maximum shear stress location when used in the PDE.

As before, the kinetic energy is again related to the maximum Reynolds shear stress. Although the ratio is again a constant, it will vary along the path of maximum Reynolds shear stress, due to  $F$ .

$$(2) \quad \frac{\tau_m}{k_m} = \frac{2A_2}{F^\alpha}$$

From experimental correlation, the value of  $\alpha$  is taken as 1.25 and that of  $A_2$  is 0.15. The next assumption is made by observing that the normal stress production does not appear in (A.1), but only the shear stress term. The effects can therefore be included by simply multiplying the production type term by  $F$ .

$$(3) \quad F\tau_m \frac{\partial U}{\partial y}$$

And as done in the original model, the gradient term can be substituted with the following,

$$(4) \quad L_m \frac{\partial U}{\partial y} = (\tau_{m,eq})^{\frac{1}{2}}$$

The final assumption returns to the diffusion term. Data from experiments still cannot predict a good modeling term for the diffusion of kinetic energy. The basic assumption that was made in the baseline model will again be used here, except that the constant  $a_1$  will be replaced with  $\frac{2A_2}{F^\alpha}$ .

$$(5) \quad D_m = \frac{C_{dif}(\tau_m)^{\frac{3}{2}} F^\alpha}{2A_2 \delta [0.7 - (\frac{y}{\delta})_m]} |1 - \sigma(x)^{\frac{1}{2}}|$$

As a final note, the modeling term for the dissipation is kept the same as before. When these assumptions and observations are substituted into (A.1), the resulting equation takes the form,

$$\frac{F^\alpha}{2A_2} \frac{\partial \tau_m}{\partial t} + \frac{u_m F^\alpha}{2A_2} \frac{\partial \tau_m}{\partial x} = \frac{F \tau_m (\tau_{m,eq})^{\frac{1}{2}}}{L_m} - D_m - \frac{(\tau_m)^{\frac{3}{2}}}{L_m} \quad (\text{A.3})$$

# Appendix B

## Hytopoulos Turbulence Model

An overview of the turbulence model developed by Hytopoulos [15] will be presented here. The turbulence model is applicable to two-dimensional, steady and unsteady boundary layers in adverse pressure gradients. The turbulence model solves for the mixing length,  $l_m$ , thus using the mixing length formulation for the Reynolds shear stress.

$$\tau_t = -\overline{\rho u'v'} = \rho l_m^2 \left| \frac{\partial U}{\partial y} \right| \frac{\partial U}{\partial y} \quad (\text{B.1})$$

A hyperbolic tangent function is used for blending the inner and outer mixing length regions. The expression for the mixing length is:

$$l_m = l_{m,o} \left[ \tanh \left( \frac{l_{m,in}}{l_{m,o}} \right) \right] \quad (\text{B.2})$$

where  $l_{m,o}$  represents the mixing length in the outer region and  $l_{m,in}$  for the inner region. For the inner region, the assumption is made that the Law-of-the-Wall remains valid in the presence of strong adverse pressure gradients. With this assumption, the expression for  $l_{m,in}$  can be derived from the momentum equation for unsteady, two-dimensional flow.

$$l_{m,in} = \kappa y D \left( 1 + \frac{y}{\rho u^{*2}} \frac{\partial P}{\partial x} + \frac{\nu}{u^{*2}} \frac{du^*}{dx} \int_0^{y^+} h^2(y^+) dy^+ + \frac{y}{u^{*2}} h^2(y^+) \frac{du^*}{dt} \right)^{\frac{1}{2}} \quad (\text{B.3})$$

In the above expression,  $D$  is the van Driest damping function,  $\kappa$  is the von Karman constant, and  $h(y^+) = (u/u^*)$ . The inner mixing length expression takes into account

the pressure gradient and both steady and unsteady convection terms. These terms become more important as an adverse pressure gradient strengthens. For the outer region, the mixing length for most algebraic models is taken to be a constant. For the model presented here, the mixing length is allowed to vary to take into account non-equilibrium effects as the flow approaches separation. The expression for the outer region mixing length is,

$$l_{m,o} = C\delta \quad (\text{B.4})$$

The value of  $C$  is determined at each boundary layer station and time step. The value of  $C$  is determined from a partial differential equation governing the maximum shear stress evolution. The differential equation is an integral form of the turbulent kinetic energy equation. The integral form approach is taken to eliminate the diffusion and laminar viscous dissipation terms. The equation is cast in terms of the maximum Reynolds shear stress,  $\tau_{max}$ , and allows the model to take into account the history effects. The differential equation is given below, along with the integral terms that help define the equation.

$$\begin{aligned} \frac{\partial}{\partial t}(\tau_{max}I_t) + \frac{\partial}{\partial x}(\tau_{max}I_c) - (\tau_{max}I_p) + (\tau_{max}^{3/2}I_d) &= 0 \quad (\text{B.5}) \\ I_t &= \int_0^\infty \frac{fF^p}{A_1} dy & I_c &= \int_0^\infty U \frac{fF^p}{A_1} dy \\ I_p &= \int_0^\infty \frac{\partial U}{\partial y} fF dy & I_d &= \int_0^\infty U \frac{f^{3/2}}{L_d} dy \end{aligned}$$

The parameter  $F$  is the ratio of the total turbulence energy production to shear stress production while  $f$  is a function which gives the non-dimensionalized stress profile for each station. The value of  $A_1$  is taken to be 0.25, and the value of  $p$  is 1.25. The dissipation length,  $L_d$ , is similar to the mixing length in the Law-of-the-Wall region. The dissipation length is computed as  $L_d = C_d\delta$  with  $C_d = 0.082$ . As  $C$  is reduced by the differential equation in regions of strong adverse pressure gradients, the value of  $C_d$  is assumed to be  $(0.082/0.09)C$ .

Finally, the value of  $C$  is found from a relationship that ties together the outer mixing length constant and the maximum shear stress.

$$l_m^2(C) \left( \frac{\partial U}{\partial y} \right)_{max} = \tau_{max} \quad (\text{B.6})$$

By satisfying this expression, the maximum Reynolds shear stress predicted by the model will equal  $\tau_{max}$  from equation (B.5).

## Appendix C

# Minimization of $J(y)$ in GMRES Algorithm

One of the main steps in the GMRES algorithm is the minimization of the function  $J(y) = \|\beta e_1 - H_k y\|$ . For  $k$  search directions, the upper Hessenberg matrix  $H_k$  has dimensions  $(k+1) \times k$ . Due to the extra row in  $H_k$ , the minimization problem cannot be solved by a direct solver, but requires the solution by the least squares method. This process will be discussed in this section, along with an example for  $k = 3$ .

A quick note is made about the minimization problem. The upper Hessenberg matrix, excluding the extra row added on to it, is defined by

$$H = V_k^T A V_k$$

To get the exact minimum of the function  $J(y)$ , then  $k$  would have to equal the number of unknowns in the system. If this were the case, the minimum vector  $y$  could be attained directly by

$$y = H^{-1} \beta e_1$$

But since the Hessenberg matrix had the additional row, a direct inversion of the matrix in order to solve the system is not possible. Instead, there is one extra equation than the number of unknowns, making the problem solvable by the method of least squares [157].

Applying the least squares concept to the minimization of  $J(y)$  results in the following linear system to solve.

$$[LS] y = r \quad (\text{C.1})$$

where the matrix  $[LS]$  is symmetric with entries for the upper half and main diagonal given by,

$$(LS)_{i,j} = \sum_{l=1}^{i+1} h_{l,j} h_{l,i} \quad (\text{C.2})$$

The right hand side vector is

$$r = \beta h_{1,i} \quad \text{for } i = 1, 2, \dots, k \quad (\text{C.3})$$

The system is usually small enough to be solved by direct methods. Two methods which are common to use in this situation are QR factorization and LU decomposition. QR factorization is especially intended for symmetric matrices and is normally used in solutions to the least squares problem, but either method will give the desired solution vector  $y$ .

An example where  $k = 3$  will now be covered to help clarify and explain the least squares method applied to  $J(y)$ . To begin, the terms in the function are written out.

$$\begin{aligned} \beta e_1 - H_k y &= \begin{bmatrix} \beta \\ 0 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} \\ h_{2,1} & h_{2,2} & h_{2,3} \\ 0 & h_{3,2} & h_{3,3} \\ 0 & 0 & h_{4,3} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \\ &= \begin{bmatrix} \beta - h_{1,1} y_1 - h_{1,2} y_2 - h_{1,3} y_3 \\ -h_{2,1} y_1 - h_{2,2} y_2 - h_{2,3} y_3 \\ -h_{3,2} y_2 - h_{3,3} y_3 \\ -h_{4,3} y_3 \end{bmatrix} \end{aligned}$$

Next the square of the  $l_2$  norm is taken

$$\begin{aligned} \|\beta e_1 - H_k y\|^2 &= [\beta - h_{1,1} y_1 - h_{1,2} y_2 - h_{1,3} y_3]^2 + \\ &\quad [-h_{2,1} y_1 - h_{2,2} y_2 - h_{2,3} y_3]^2 + \\ &\quad [-h_{3,2} y_2 - h_{3,3} y_3]^2 + [-h_{4,3} y_3]^2 \end{aligned}$$

Partial derivatives with respect to  $y_1, y_2, \text{ and } y_3$  are now computed. If these three derivatives are set to zero, three equations will result. The least squares method will therefore seek to minimize the partial derivatives. The three equations that result from taking the derivative of  $J^2(y) = \|\beta e_1 - H_k y\|^2$  are as follows,

$$\begin{aligned}
\frac{\partial[J^2(y)]}{\partial y_1} &= 0 = 2[\beta - h_{1,1} y_1 - h_{1,2} y_2 - h_{1,3} y_3](-h_{1,1}) \\
&\quad 2[-h_{2,1} y_1 - h_{2,2} y_2 - h_{2,3} y_3](-h_{2,1}) \\
\frac{\partial[J(y)^2]}{\partial y_2} &= 0 = 2[\beta - h_{1,1} y_1 - h_{1,2} y_2 - h_{1,3} y_3](-h_{1,2}) + \\
&\quad 2[-h_{2,1} y_1 - h_{2,2} y_2 - h_{2,3} y_3](-h_{2,2}) + \\
&\quad 2[-h_{3,2} y_2 - h_{3,3} y_3](-h_{3,2}) \\
\frac{\partial[J(y)^2]}{\partial y_3} &= 0 = 2[\beta - h_{1,1} y_1 - h_{1,2} y_2 - h_{1,3} y_3](-h_{1,3}) + \\
&\quad 2[-h_{2,1} y_1 - h_{2,2} y_2 - h_{2,3} y_3](-h_{2,3}) + \\
&\quad 2[-h_{3,2} y_2 - h_{3,3} y_3](-h_{3,3}) + 2[-h_{4,3} y_3](-h_{4,3})
\end{aligned}$$

Rearranging terms and placing the three equations in system form gives the final result  $[LS] y = r$ . Note that the matrix is symmetric, and agrees with equation (C.2).

$$[LS] = \begin{bmatrix} (h_{1,1}^2 + h_{2,1}^2) & (h_{1,1}h_{1,2} + h_{2,2}h_{2,1}) & (h_{1,3}h_{1,1} + h_{2,3}h_{2,1}) \\ (h_{1,1}h_{1,2} + h_{2,2}h_{2,1}) & (h_{1,2}^2 + h_{2,2}^2 + h_{3,2}^2) & (h_{1,3}h_{1,2} + h_{2,3}h_{2,2} + h_{3,3}h_{3,2}) \\ (h_{1,3}h_{1,1} + h_{2,3}h_{2,1}) & (h_{1,3}h_{1,2} + h_{2,3}h_{2,2} + h_{3,3}h_{3,2}) & (h_{1,3}^2 + h_{2,3}^2 + h_{3,3}^2 + h_{4,3}^2) \end{bmatrix}$$

$$r = \begin{bmatrix} \beta h_{1,1} \\ \beta h_{1,2} \\ \beta h_{1,3} \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

# Appendix D

## Ghost Cell Location

Proper determination of the ghost cell location is very important in the unstructured formulation. The major requirement for the ghost cell is that the vector connecting the center of the ghost cell with the adjacent interior cell center lies in the direction of the boundary face normal vector. This condition is most easily satisfied by creating a ghost cell that is a mirror reflection of the adjacent interior cell. The determination of the cell center for the ghost cell is explained below assuming a mirror reflection.

Figure D.1 shows an arbitrary boundary cell with its mirror ghost cell. All distances and points discussed below will be in reference to the figure. The necessary information that will be needed in the calculation is as follows: boundary cell center  $(x_c, y_c, z_c)$ , boundary face center  $(x_f, y_f, z_f)$ , and the boundary normal vector  $\hat{n}$ . With this information given, the following steps can be taken to locate the ghost cell center  $(x_g, y_g, z_g)$ .

1. Compute the vector  $\vec{r}$  which begins at  $(x_f, y_f, z_f)$  and ends at  $(x_c, y_c, z_c)$ .
2. Compute  $dr$  which is the magnitude of  $\vec{r}$ , defined by:

$$dr = \sqrt{(x_c - x_f)^2 + (y_c - y_f)^2 + (z_c - z_f)^2}$$

3. Perform the dot product  $(\hat{n} \cdot \vec{r})$ . This is done because of the following inequality:  
 $\vec{F} \cdot \vec{G} = \|F\| \|G\| \cos(\theta)$ . Since  $\|\hat{n}\| = 1$ , the angle  $\theta$  can be defined as,

$$\cos(\theta) = \frac{(\hat{n} \cdot \vec{r})}{dr}$$

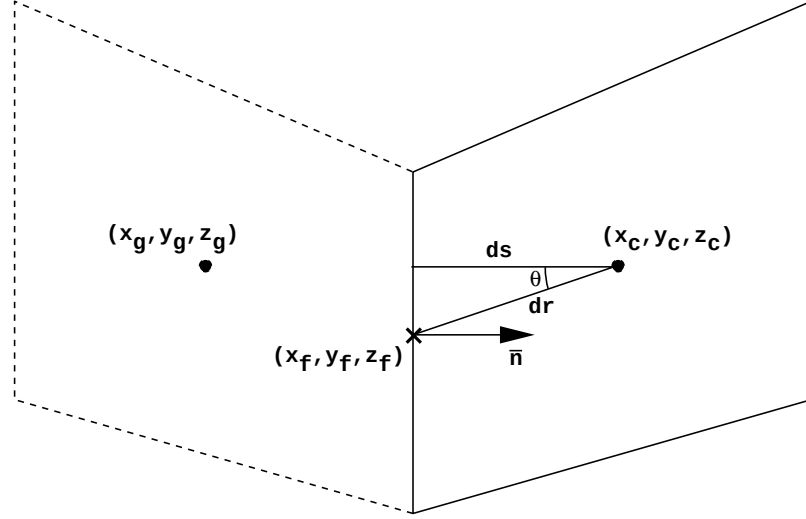


Figure D.1: Terminology for finding ghost cell center location.

4. Since the line  $ds$  intersects the boundary face at a  $90^\circ$  angle, the following trigonometric identity can be applied,  $\cos(\theta) = \frac{ds}{dr}$ . Substituting in the above expressions give the following result:

$$ds = dr \cos(\theta) = dr \frac{(\hat{n} \cdot \vec{r})}{dr} = (\hat{n} \cdot \vec{r})$$

5. With the distance  $ds$  known, the location of the ghost cell center is  $2(ds)$  distance away from  $(x_c, y_c, z_c)$  in the opposite direction of the unit normal. The result is therefore:

$$\begin{aligned} x_g &= x_c - 2(ds)n_x \\ y_g &= y_c - 2(ds)n_y \\ z_g &= z_c - 2(ds)n_z \end{aligned}$$

where  $ds$  is an absolute value and  $\hat{n}$  is assumed to be pointing inward, as shown in the figure.

# Bibliography

- [1] “Experimental Data Base for Computer Program Assessment,” tech. rep., AGARD Advisory Report No. 138.
- [2] Rieger, H. and Jameson, A., “Solution of the Three-Dimensional Compressible Euler and Navier-Stokes equations by an implicit LU Scheme,” AIAA Paper 88-0619, 1988.
- [3] Schock, H. J., Sosoka, D. J., and Ramos, J. I., “Formation and Destruction of Vortices in a Motored Four-Stroke Piston-Cylinder Configuration,” *AIAA Journal*, vol. 22, pp. 948–949, 1984.
- [4] Newsome, R. W. J., “Numerical Simulation of Near-Critical and Unsteady Subcritical Inlet Flow Fields,” AIAA Paper 83-0175, 1983.
- [5] Pulliam, T. H. and Steger, J. L., “Recent Improvements in Efficiency, Accuracy and Convergence for Implicit Approximate Factorization Algorithms,” AIAA Paper 85-0360, 1985.
- [6] Hirsch, C., *Numerical Computation of Internal and External Flows: Vol 2*. John Wiley & Sons, 1990. ISBN 0-471-92452-0, pp. 70–72.
- [7] McRae, D. S. and Hussaini, M. Y., “Numerical Simulation of Supersonic Cone Flow at High Angle of Attack,” ICASE Report No. 78-21, 1978.
- [8] Fletcher, C., *Computational Techniques for Fluid Dynamics, Volume II*. Springer-Verlag, 1991. ISBN 0-387-53601-9, pp. 26–33.

- [9] Hirsch, C., *Numerical Computation of Internal and External Flows: Vol 1*. John Wiley & Sons, 1988. ISBN 0-471-92385-0, pp. 108–127.
- [10] Holst, T., Slooff, J., Yoshihara, H., and Ballhaus, W. J., “Applied Computational Transonic Aerodynamics,” AGARDograph No. 266, 1982.
- [11] Hirsch, C., *Numerical Computation of Internal and External Flows: Vol 2*. John Wiley & Sons, 1990. ISBN 0-471-92452-0.
- [12] Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*. Hemisphere Publishing Corporation, 1984. ISBN 0-89116-471-5, pp. 221–235.
- [13] Fletcher, C., *Computational Techniques for Fluid Dynamics, Volume II*. Springer-Verlag, 1991. ISBN 0-387-53601-9, pp. 373–395.
- [14] Menter, F. R., “Performance of Popular Turbulence Models for Attached and Separated Adverse Pressure Gradient Flow,” AIAA Paper 91-1784, June 1991.
- [15] Hytopoulos, E., *A Turbulence Model for Steady and Unsteady Boundary Layers in Strong Pressure Gradients*. PhD thesis, Virginia Polytechnic Institute and State University, VPI, Virginia, 1994.
- [16] Simpson, R. L., Chew, Y.-T., and Shivaprasad, B. G., “The Structure of a Separating Turbulent Boundary Layer. Part I. Mean Flow and Reynolds Stresses,” *Journal of Fluid Mechanics*, vol. 113, pp. 23–51, 1981.
- [17] Kline, S. J., Cantwell, B. J., and Lilley, G. M., “1980-81 AFOSR-HTTM-Stanford Conference on Complex Turbulent Flows,” Stanford University, California, 1981.
- [18] Johnson, D. A. and King, L. S., “A Mathematically Simple Turbulence Closure Model for Attached and Separated Turbulent Boundary Layers,” *AIAA Journal*, vol. 28, pp. 1684–1692, Nov. 1985.

- [19] Kim, J., Moin, P., and Moser, R., "Turbulence Statistics in Fully Developed Channel Flow at Low Reynolds Number," *Journal of Fluid Mechanics*, vol. 177, pp. 133–166, 1987.
- [20] Wilcox, D. C., *Turbulence Modeling for CFD*. DCW Industries, Inc., 1993. ISBN 0-9636051-0-0.
- [21] Rotta, J., "Statistische Theorie nichthomogener Turbulenz," *Z. Phys.*, vol. 129, pp. 547–572, 1951.
- [22] Launder, B. E., "Stress-Transport Closures: Into the Third Generation," in *Proceedings First Symposium on Turbulent Shear Flows*, Springer-Verlag, New York, 1979.
- [23] Deardorff, J. W., "A Numerical Study of Three-dimensional Turbulent Channel Flow at Large Reynolds Numbers," *Journal of Fluid Mechanics*, vol. 41, pp. 453–480, 1970.
- [24] McDonald, H. and Kreskovsky, J. P., "Effect of Free Stream Turbulence on the Turbulent Boundary Layer," *International Journal of Heat Mass Transfer*, vol. 17, pp. 705–716, 1974.
- [25] Chan, Y. Y., "Compressible Turbulent Boundary Layer Computations Based on an Extended Mixing Length Approach," *Canadian Aeronautics and Space Institute Transactions*, vol. 5, pp. 21–27, 1972.
- [26] Adams Jr., J. C. and Hodge, B. K., "The Calculation of Compressible, Transitional, Turbulent, and Relaminarizational Boundary Layers Over Smooth and Rough Surfaces Using an Extended Mixing Length Hypothesis," AIAA Paper 77-682, 1977.
- [27] Malik, M. R. and Pletcher, R. H., "A Study of Some Turbulence Models for Flow and Heat Transfer in Ducts of Annular Cross-Section," *Journal of Heat Transfer*, vol. 103, pp. 146–152, 1981.

- [28] Reyhner, T. A., "Finite-Difference Solution of the Compressible Turbulent Boundary Layer Equations," in *Proceedings Computation of Turbulent Boundary Layers-1968 AFOSR-IFP-Stanford Conference, vol 1*. Stanford University, California, 1968.
- [29] Shang, J. S. and Hankey Jr., W. L., "Supersonic Turbulent Separated Flows Utilizing the Navier-Stokes Equations," *Flow Separation*, AGARD-CCP-168, 1975.
- [30] Johnson, D. A. and King, L. S., "A New Turbulence Closure Model for Boundary Layer Flows with Strong Adverse Pressure Gradients and Separation," AIAA Paper 84-0175, June 1991.
- [31] Johnson, D. A., "Transonic Separated Flow Predictions with an Eddy Viscosity/Reynolds Stress Closure Model," *AIAA Journal*, vol. 25, pp. 252–259, Feb. 1987.
- [32] Abid, R., "Extension of the Johnson-King Turbulence Model to the 3-D Flows," AIAA Paper 88-0223, Jan. 1988.
- [33] Abid, R., Vatsa, V., Johnson, D., and Wedan, B., "Prediction of Separated Transonic Wing Flows with a Non-Equilibrium Algebraic Model," AIAA Paper 89-0558, Jan. 1989.
- [34] Johnson, D. A. and Coakley, T. J., "Improvements to a Nonequilibrium Algebraic Turbulence Model," *AIAA Journal*, vol. 28, pp. 2000–2003, Nov. 1990.
- [35] Wu, J., Sankar, L., and Huff, D., "Evaluation of Three Turbulence Models for the Prediction of Steady and Unsteady Airloads," AIAA Paper 89-0609, Jan. 1989.
- [36] Dindar, M. and Kaynak, U., "Effect of Turbulence Modeling on Dynamic Stall of a NACA 0012 Airfoil," AIAA Paper 92-0027, Jan. 1992.
- [37] Johnson, D. A., "Nonequilibrium Algebraic Turbulence Modeling Considerations for Transonic Airfoils and Wings," AIAA Paper 92-0026, Jan. 1992.

- [38] Driver, D. M., "Reynolds Shear Stress Measurements in a Separated Boundary Layer Flow," AIAA Paper 91-1787, June 1991.
- [39] Simpson, R. L., "A Review of Some Phenomena in Turbulent Flow Separation," *Journal of Fluids Engineering*, vol. 102, pp. 520–533, 1981.
- [40] Simpson, R. L., "Two-Dimensional Turbulent Separated Flow," *AGARDograph 287*, vol. 1, 1985.
- [41] Simpson, R. L., "Turbulent Boundary-Layer Separation," *Annual Review of Fluid Mechanics*, vol. 21, pp. 205–234, 1989.
- [42] Simpson, R. L., "Aspects of Turbulent Boundary-Layer Separation," *Progress in Aerospace Science*, vol. 32, pp. 457–521, 1996.
- [43] Simpson, R. L., "The Structure of the Near-Wall Region of Two-Dimensional Turbulent Separated Flow," *Phil. Trans. R. Soc. Lond. A*, vol. 336, pp. 5–17, 1991.
- [44] Perry, A. E. and Schofield, W. H., "Mean Velocity and Shear Stress Distributions in Turbulent Boundary Layers," *Physics of Fluids*, vol. 16, pp. 2068–2074, 1973.
- [45] Simpson, R. L., Strickland, J. H., and Barr, P. W., "Failures of a Separating Turbulent Boundary Layer in the Vicinity of Separation," *Journal of Fluid Mechanics*, vol. 79, pp. 553–594, 1977.
- [46] Simpson, R. L., "A Model for the Back-Flow Mean Velocity Profile," *AIAA Journal*, vol. 21, pp. 142–143, 1983.
- [47] Agarwal, N. K. and Simpson, R. L., "Backflow Structure of Steady and Unsteady Separating Turbulent Boundary Layers," *AIAA Journal*, vol. 28, pp. 1764–1771, Oct. 1990.
- [48] Fletcher, C., *Computational Techniques for Fluid Dynamics, Volume II*. Springer-Verlag, 1991. ISBN 0-387-53601-9, pp. 373–395.

- [49] Issa, R. I., "Solution of the Implicitly Discretised Fluid Flow Equations by Operator-Splitting," *Journal of Computational Physics*, vol. 62, pp. 40–65, 1985.
- [50] Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*. Hemisphere, Washington D.C., 1980.
- [51] Chorin, A. J., "A Numerical Method for Solving Incompressible Viscous Flow Problems," *Journal of Computational Physics*, vol. 2, pp. 12–26, 1967.
- [52] Merkle, C. L. and Athavale, M., "Time-Accurate Unsteady Incompressible Flow Algorithms Baed on Artificial Compressibility," AIAA Paper 87-1137, 1987.
- [53] Arabshahi, A., Taylor, L., and Whitfield, D., "UNCLE: Toward a Comprehensive Time-Accurate Incompressible Navier-Stokes Flow Solver," AIAA Paper 95-0050, 1995.
- [54] Rogers, S. E. and Kwak, D., "An Upwind Differencing Scheme for the Time-Accurate Incompressible Navier-Stokes Equations," *AIAA Journal*, vol. 28, pp. 253–262, 1990.
- [55] Rogers, S. E. and Kwak, D., "Steady and Unsteady Solutions of the Incompressible Navier-Stokes Equations," *AIAA Journal*, vol. 29, pp. 603–610, Apr. 1991.
- [56] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectoroes, and Difference Schemes," *Journal of Computational Physics*, vol. 43, pp. 357–372, 1981.
- [57] Hirsch, C., *Numerical Computation of Internal and External Flows: Vol 2*. John Wiley & Sons, 1990. ISBN 0-471-92452-0, pp. 494–498.
- [58] Anderson, W. K., Rausch, R. D., and Bonhaus, D. L., "Implicit/Multigrid Algorithms for Incompressible Turbulent Flows on Unstructured Grids," *Journal of Computational Physics*, vol. 128, no. 2, pp. 391–408, Oct. 1996.

- [59] Hirsch, C., *Numerical Computation of Internal and External Flows: Vol 2*. John Wiley & Sons, 1990. ISBN 0-471-92452-0, pp. 309–310.
- [60] Rogers, S. E., “A Comparison of Implicit Schemes for the Incompressible Navier-Stokes Equations With Artificial Compressibility,” AIAA Paper 95-0567, 1995.
- [61] Hirsch, C., *Numerical Computation of Internal and External Flows: Vol 2*. John Wiley & Sons, 1990. ISBN 0-471-92452-0, pp. 88–92.
- [62] Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*. Hemisphere Publishing Corporation, 1984. ISBN 0-89116-471-5, pp. 260–272.
- [63] Merkle, C. L. and Athavale, M., “Time-Accurate Unsteady Incompressible Flow Algorithms Based on Artificial Compressibility,” AIAA Paper 87-1137, 1987.
- [64] Schetz, J. A., *Boundary Layer Analysis*. Prentice-Hall, 1993. ISBN 0-13-086885-X, pp. 81–85.
- [65] Baldwin, B. S. and Lomax, H., “Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows,” AIAA Paper 78-257, 1978.
- [66] Cebeci, T. and Smith, A. M. O., “Analysis of Turbulent Boundary Layers,” *Applied Mathematics and Mechanics*, vol. 25, 1974.
- [67] Yu, N. J., Allmaras, S. R., and Moschetti, K. G., “Navier-Stokes Calculations for Attached and Separated Flows Using Different Turbulence Models,” AIAA Paper 91-1791, 1991.
- [68] Shiloh, K., Shivaprasad, B. G., and Simpson, R. L., “The Structure of a Separating Turbulent Boundary Layer. Part III. Transverse Velocity Measurements,” *Journal of Fluid Mechanics*, vol. 113, pp. 75–90, 1981.
- [69] Simpson, R. L., Chew, Y.-T., and Shivaprasad, B. G., “The Structure of a Separating Turbulent Boundary Layer. Part II. Higher-Order Turbulence Results,” *Journal of Fluid Mechanics*, vol. 113, pp. 53–73, 1981.

- [70] Huang, P. and Bradshaw, P., “Law of the Wall for Turbulent Flows in Pressure Gradients,” *AIAA Journal*, vol. 33, pp. 624–632, Apr. 1995.
- [71] Schetz, J. A., *Boundary Layer Analysis*. Prentice-Hall, 1993. ISBN 0-13-086885-X, pp. 233–247.
- [72] Lecture Notes in CFD, Virginia Polytechnic Institute and State University, Blacksburg, VA, 1994.
- [73] Simpson, R. L., Shivaprasad, B. G., and Chew, Y.-T., “The Structure of a Separating Turbulent Boundary Layer. Part IV. Effects of Periodic Free-Stream Unsteadiness,” *Journal of Fluid Mechanics*, vol. 127, pp. 219–26, 1983.
- [74] Simpson, R. L. and Shivaprasad, B. G., “The Structure of a Separating Turbulent Boundary Layer. Part V. Frequency Effects on Periodic Unsteady Free-Stream Flows,” *Journal of Fluid Mechanics*, vol. 131, pp. 319–339, 1983.
- [75] Paterson, E. and Stern, F., “Computation of Unsteady Viscous Flow with Application to the MIT Flapping Foil Experiment,” Sixth International Conference on Numerical Ship Hydrodynamics, Iowa City, Iowa, Aug. 1993.
- [76] Taylor, L., Busby, J., Jiang, M., Arabshahi, A., Sreenivas, K., and Whitfield, D., “Time Accurate Incompressible Navier-Stokes Simulation of the Flapping Foil Experiment,” Sixth International Conference on Numerical Ship Hydrodynamics, Iowa City, Iowa, Aug. 1993.
- [77] Kiris, C., Rogers, S., Kwak, D., and Lee, Y., “Time Accurate Incompressible Navier-Stokes Computations With Overlapped Moving Grids,” *Advances in Computational Methods in Fluid Dynamics American Society of Mechanical Engineers*, 1994.
- [78] Yang, H. Q., Ho, S. Y., and Przekwas, A. J., “A Numerical Study of a Two-Dimensional Foil Subject to High Reduced Frequency Gust Loading,” *AIAA Paper 95-2266*, June 1995.

- [79] Ho, Y.-H. and Lakshminarayana, B., "Computation of Unsteady Flowfield over a Hydrofoil, Including Boundary Layer and Wakes," *AIAA Journal*, vol. 35, no. 1, pp. 40–50, Jan. 1997.
- [80] Rice, J., "Investigation of a Two-Dimensional Hydrofoil in Steady and Unsteady Flows," Master's thesis, Massachusetts Institute of Technology, MIT, Massachusetts, 1991.
- [81] Delperio, P. M., "Investigation of Flows Around a Two Dimensional Hydrofoil Subject to a High Reduced Frequency Gust Loading," Master's thesis, Massachusetts Institute of Technology, MIT, Massachusetts, Feb. 1992.
- [82] Lurie, E. A., *Investigation of High Reduced Frequency, Separated Trailing Edge Flows*. PhD thesis, Massachusetts Institute of Technology, MIT, Massachusetts, Sept. 1996.
- [83] Chien, K.-Y., "Prediction of Channel and Boundary-Layer Flows with a Low-Reynolds-Number Turbulence Model," *AIAA Journal*, vol. 20, no. 1, no. 1, pp. 33–38, 1982.
- [84] Murman, E. M. and Cole, J. D., "Calculation of Plane Steady Transonic Flows," *AIAA Journal*, vol. 9, pp. 114–121, 1971.
- [85] Murman, E. M., "Analysis of Embedded Shock Waves Calculated by Relaxation Methods," in *Proceedings First AIAA Computational Fluid Dynamics Conference*. Palm Springs, July 1973.
- [86] Steger, J. L. and Lomax, H., "Transonic Flow About Two-Dimensional Airfoils by Relaxation Procedures," *AIAA Journal*, vol. 10, pp. 49–54, 1972.
- [87] Garabedian, P. R. and Korn, D., "Analysis of Transonic Airfoils," *Communications on Pure and Applied Mathematics*, vol. 24, pp. 841–851, 1971.
- [88] Sells, C. C. L., "Plane Subcritical Flow Past a Lifting Aerofoil," *Proceedings of the Royal Society London*, vol. 308, pp. 377–401, 1968.

- [89] Ballhaus, W. F. and Bailey, F. R., "Numerical Calculation of Transonic Flow About Swept Wings," AIAA Paper 72-677, 1972.
- [90] Bailey, F. R. and Steger, J. L., "Relaxation Techniques for Three-Dimensional Transonic Flow About Wings," AIAA Paper 72-189, Jan. 1972.
- [91] Jameson, A., "Iterative Solutions of Transonic Flows Over Airfoils and Wings, Including Flows at Mach 1," *Communications on Pure and Applied Mathematics*, vol. 27, pp. 283–309, 1974.
- [92] Murman, E. M., "Analysis of Embedded Shock Waves Calculated by Relaxation Methods," *AIAA Journal*, vol. 12, no. 5, no. 5, pp. 626–635, 1974.
- [93] Jameson, A., "Transonic Potential Flow Calculations Using Conservative Form," in *Proceedings AIAA Second Computational Fluid Dynamics Conference*. Hartford, 1975.
- [94] Argyris, J. H., Mareczek, G., and Sharpf, D. W., "Two and Three Dimensional Flow Using Finite Elements," *Aeronautical Journal of the Royal Aero. Society*, vol. 73, pp. 961–964, 1969.
- [95] De Vries, G. and Norrie, D. H., "The Application of the Finite Element Technique to Potential Flow Problems," *Trans. ASME—Journal of Applied Mechanics*, pp. 798–802, 1971.
- [96] Thompson, D. S., "Flow Through a Cascade of Airfoils," In J. T. Oden (ed.), *Finite Elements in Flow Problems*, URH Press, 1974.
- [97] Periaux, J., *Resolution de quelques problemes non-lineaires en Aerodynamique par des methodes d'elements finis et de moindres carres fonctionnels*. PhD thesis, Universite Paris VI, France, 1975.
- [98] Shen, S. F. and Habashi, W., "Local Linearization of the Finite Element Method and its Applications to Compressible Flows," *Int. Journal for Numerical Methods in Engineering*, vol. 10, pp. 565–577, 1976.

- [99] Glowinsky, R., Periaux, J., and Pironneau, O., "Transonic Flow Simulation by the Finite Element Method Via Optimal Control," J. Chichester (ed.) *Finite Elements in Fluids*, Vol 3, Wiley & Sons, pp. 249–259, 1976.
- [100] Ecer, A. and Akay, H. U., "Application of Finite Element Method for the Solution of Transonic Flow," J. Chichester (ed.) *Finite Elements in Fluids*, Vol 3, Wiley & Sons, pp. 191–201, 1976.
- [101] Jameson, A. and Caughey, D. A., "A Finite Volume Method For Transonic Potential Flow Calculations," in *Proceedings AIAA Third Computational Fluid Dynamics Conference*. Albuquerque, 1977.
- [102] Hafez, M. M., South, J. C., and Murman, E. M., "Artificial Compressibility Methods for the Numerical Solution of the Full Potential Equation," *AIAA Journal*, vol. 17, pp. 838–844, 1978.
- [103] Harten, A., "The Artificial Compression Method for Computation of Shocks and Contact Discontinuities," ICASE Report No. 77-2, Feb. 1977.
- [104] Frankel, S. P., "Convergence Rates of Iterative Treatments of Partial Differential Equations," *Mathematical Tables and Other Aids to Computation*, vol. 4, pp. 65–75, 1950.
- [105] Peaceman, D. W. and Rachford, H. H., "The Numerical Solution of Parabolic and Elliptic Differential Equations," *Journal of Society Ind. Applied Mathematics*, vol. 3, pp. 28–41, 1955.
- [106] Ballhaus, W. F. and Steger, J. L., "Implicit Approximate Factorization Schemes for the Low-Frequency Transonic Equations," NASA TM X-73, 082, 1975.
- [107] Ballhaus, W. F., Jameson, A., and Albert, J., "Implicit Approximate Factorization Schemes for the Efficient Solution of Steady Transonic Flow Problems," *AIAA Journal*, vol. 16, no. 6, pp. 573–579, June 1978.
- [108] Baker, T. J., "Potential Flow Calculation by the Approximate Factorization Method," *Journal of Computational Physics*, vol. 42, pp. 1–19, 1981.

- [109] Holst, T. L., "Numerical Computation of Transonic Flow Governed by the Full Potential Equation," in *Lecture Series on Computational Fluid Dynamics*. Rhode-St-Genese, Belgium, 1983.
- [110] Holst, T. L. and Ballhaus, W. F., "Fast, Conservative Schemes for the Full Potential Equation Applied to Transonic Flows," *AIAA Journal*, vol. 17, no. 2, pp. 146–152, Feb. 1979.
- [111] South, J. J. C. and Brandt, A., "Application of a Multi-Level Grid Method to Transonic Flow Calculations," ICASE Report No. 76-8, 1976.
- [112] Fuchs, L. J., "Finite-Difference Methods for Plane Steady Inviscid Transonic Flows," Report TRITAGAD-2, Stockholm, May 1977.
- [113] Arlinger, B., "Multigrid Technique Applied to Lifting Transonic Flow Using Full Potential Flow Equation," SAAB-Scania Report L-0-1 B439, Dec. 1977.
- [114] Holst, T. L., "An implicit Algorithm for the Conservative Transonic Full Potential Equation Applied to Transonic Flows," *AIAA Journal*, vol. 17, no. 10, no. 10, pp. 1038–1045, 1979.
- [115] Jameson, A., "Acceleration of Transonic Potential Flow Calculations on Arbitrary Meshes by the Multi-Grid Method," AIAA Paper 79-1458, 1979.
- [116] Boerstoele, J. W., "A Multi-Grid Algorithm for Steady Transonic Potential Flows Around Airfoils Using Newton Iteration," *Journal Computational Physics*, vol. 48, pp. 314–343, 1982.
- [117] Van der Wees, A. J., Van der Vooren, J., and Meelker, J. H., "Robust Calculation of 3D Transonic Potential Flow Based on the Non-Linear FAS Multi-Grid Method and Incomplete LU Decomposition," AIAA Paper 83-1950, 1983.
- [118] Engquist, B. and Osher, S., "Stable and Entropy Satisfying Approximations for Transonic Flow Calculations," *Mathematics of Computation*, vol. 34, pp. 45–75, 1980.

- [119] Osher, S., “Shock Modelling in Aeronautics,” In K. W. Morton and M. J. Baines (eds.), *Numerical Methods for Fluid Dynamics*, London: Academic Press, pp. 179–218, 1982.
- [120] Volpe, G. and Jameson, A., “Transonic Potential Flow Calculations by Two Artificial Density Methods,” AIAA Paper 86-1084, 1986.
- [121] Holst, T. L., “Numerical Solution of the Full Potential Equation Using a Chimera Grid Approach,” NASA TM 110360, 1995.
- [122] “NACA 1135: Equations, Tables, and Charts for Compressible Flow,” tech. rep., National Advisory Committee for Aeronautics, NASA Ames Research Center, Moffett Field, California.
- [123] Steinhoff, J. and Jameson, A., “Multiple Solutions of the Transonic Potential Flow Equations,” *AIAA Journal*, vol. 20, pp. 1521–1525, Nov. 1982.
- [124] Salas, M., Jameson, A., and Melnik, R., “A Comparative Study of the Non-Uniqueness Problem of the Potential Equation,” AIAA Paper 83-1888, 1983.
- [125] Salas, M. and Gumber, C., “Breakdown of the Conservative Potential Equation,” AIAA Paper 85-0367, 1985.
- [126] Mason, W., “On the Use of the Potential Flow Model for Aerodynamic Design at Transonic Speeds,” AIAA Paper 95-0741, 1995.
- [127] Vigneron, Y., Lejal, T., and Collecandy, R., “Computation of Transonic Flows With Nacelle Simulation,” Societe Nationale Industrielle Aerospatiale, 1987.
- [128] Kinney, D., Hafex, M., and Gelhausen, P., “Validation of a New Unstructured Full Potential Formulation,” AIAA Paper 95-1765, 1995.
- [129] Kinney, D. J., *Finite Element Solution of the Full Potential Equation Over Aircraft Configurations Using Unstructured Tetrahedral Grids*. PhD thesis, University of California, Davis, 1994.

- [130] Madson, M., Moyer, S., and Cenke, A., “TranAir Computations of the Flow About a Generic Wing/Pylon/Finned-Store Configuration,” AIAA Paper 94-0155, Jan. 1994.
- [131] Hirsch, C., *Numerical Computation of Internal and External Flows: Vol 2*. John Wiley & Sons, 1990. ISBN 0-471-92452-0, pp. 4–6.
- [132] Nixon, D., ed., *Unsteady Transonic Aerodynamics*. American Institute of Aeronautics and Astronautics, 1989. ISBN 0-930403-52-5, pp. 61–64.
- [133] John D. Anderson, J., ed., *Modern Compressible Flow With Historical Perspective*. McGraw-Hill, 1990. ISBN 0-07-001673-9, pp. 52–60.
- [134] Hafex, M., South, J., and Murman, E., “Artificial Compressibility Methods for Numerical Solutions of Transonic Full Potential Equation,” *AIAA Journal*, vol. 17, pp. 838–844, Aug. 1979.
- [135] Shankar, V., Hiroshi, I., Gorski, J., and Osher, S., “A Fast, Time-Accurate Unsteady Full Potential Scheme,” AIAA Paper 85-1512, 1985.
- [136] Caughey, D. A. and Jameson, A., “Progress in Finite-Volume Calculations For Wing-Fuselage Combinations,” *AIAA Journal*, vol. 18, pp. 1281–1288, 1980.
- [137] Goorjian, P. M., Meagher, M. E., and Buskirk, R. V., “Monotone Switches in Implicit Algorithms for Potential Equations Applied to Transonic Flows,” *AIAA Journal*, vol. 23, no. 4, pp. 492–498, Apr. 1985.
- [138] Holst, T. L., “Fast, Conservative Algorithm for Solving the Transonic Full-Potential Equation,” *AIAA Journal*, vol. 18, no. 12, pp. 1431–1439, Dec. 1980.
- [139] Karamcheti, K., *Principles of Ideal-Fluid Aerodynamics*. Robert E. Krieger Publishing Company, 1966. ISBN 0-89874-113-0, p. 131.
- [140] McGrory, B. Private Communication, AeroSoft Inc., VA, 1996.

- [141] Barth, T. J. and Frederickson, P. O., "Higher Order Solution of the Euler Equations on Unstructured Grids Using Quadratic Reconstruction," AIAA Paper 90-0013, Jan. 1990.
- [142] Saad, Y. and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *Siam J. Sci. Stat. Comput.*, vol. 7, no. 3, pp. 856–869, July 1986.
- [143] Sonneveld, P., "CGS, a Fast Lanczos-type Solver for Nonsymmetric Linear Systems," *Siam J. Sci. Stat. Comput.*, vol. 7, pp. 856–869, 1986.
- [144] Hageman, A. L. and Young, D. M., *Applied Iterative Methods*. Academic Press, New York, 1981.
- [145] Brown, P. N. and Saad, Y., "Hybrid Krylov Methods for Nonlinear Systems of Equations," *Siam J. Sci. Stat. Comput.*, vol. 11, no. 3, pp. 450–481, May 1990.
- [146] Wigton, L., Yu, N. J., and Young, D. P., "GMRES Acceleration of Computational Fluid Dynamics Codes," AIAA Paper 85-1494, 1985.
- [147] Barth, T. J. and Linton, S. W., "An Unstructured Mesh Newton Solver for Compressible Fluid Flow and Its Parallel Implementation," AIAA Paper 95-0221, Jan. 1995.
- [148] Hixon, R. and Sankar, L., "Application of a Generalized Minimal Residual Method to 2D Unsteady Flows," AIAA Paper 92-0422, 1992.
- [149] Duff, I., Erisman, A., and Reid, J., *Direct Methods for Sparse Matrices*. Clarendon Press, Oxford, 1986, p. 154.
- [150] Slack, D. C., *The Development of Solution Algorithms for Compressible flows*. PhD thesis, Virginia Polytechnic Institute and State University, Jan. 1991.
- [151] Holst, T. L., "User's Manual for the Transonic Overset Potential Solver (TOPS) Computer Program," Advanced Computational Methods Branch, Ames Research Center, Nov. 1996.

- [152] Abbott, I. H. and Von Doenhoff, A. E., *Theory of Wing Sections*. Dover Publications, Inc., 1959. ISBN 0-486-60586-8.
- [153] Liebeck, R. H. and Camacho, P. P., “Airfoil Design at Low Reynolds Number with Constrained Pitching Moment,” in *Proceedings of the Conference on Low Reynolds Number Airfoil Aerodynamics*, pp. 27 – 35, 1985.
- [154] AeroSoft, Inc., 1872 Pratt Drive, Blacksburg, VA 24060, *GUST User’s Manual: General Unstructured Software Toolkit*.
- [155] Pointwise, Inc., Fort Worth, TX 76095, *Users’s Manual For Gridgen: Structured Grid Generation Software*.
- [156] Flores, J., Barton, J., Holst, T., and Pulliam, T., “Comparison of the Full-Potential and Euler Formulations for Computing Transonic Airfoil Flows,” NASA TM 85983, 1984.
- [157] Burden, R. L. and Faires, D. J., *Numerical Analysis*. PWS-Kent Publishing Company, 1989. ISBN 0-53491-585-X, pp. 436–444.