Full Potential Analysis and Design of Transonic Propellers

by

Philip B Poll

B.S. Texas A & M University (1988)

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science
in
Aeronautics and Astronautics
at the
Massachusetts Institute of Technology

February 1991

Signature of Author ______________________________
Department of Aeronautics and Astronautics
January 16,1991

Certified by ______________________________
Associate Professor Mark Drela
Thesis Supervisor, Department of Aeronautics and Astronautics

Accepted by ______________________________
Professor Harold Y. Wachman
Chairman, Department Graduate Committee

Full Potential Analysis and Design of Transonic Propellers

by

Philip B Poll

Submitted to the Department of Aeronautics and Astronautics
on January 14, 1991
in partial fulfillment of the requirements for the degree of
Master of Science in Aeronautics and Astronautics

An analysis and design method is developed for subsonic and transonic propellers. Finite element techniques are used to discretize the full potential equation on a three-dimensional domain and the Newton-Raphson iteration method is used to solve the nonlinear equations. Results are presented that demonstrate the analysis capability of the present scheme including results for a marine propeller and for the SR-3 Prop-Fan. The design method iteratively modifies an initial geometry and analysis solution to obtain the best fit to a specified target pressure distribution. The goal in developing the present method was to provide a fast way to make small changes to a propeller geometry designed using two-dimensional methods to correct for three-dimensional effects. Modifications to the design surface are restricted to combinations of a fixed set of geometric modes and the design surface displacement is modeled using wall transpiration. The number of global unknowns introduced for the design method is a small set of coefficients that multiply the geometric modes. Using the Newton method, mode coefficient sensitivities can be computed in about the same time required for a full direct calculation. Then, a linearly perturbed solution for a target pressure distribution can be obtained in an order of magnitude less time than a single direct calculation.

Thesis Supervisor: Mark Drela,
Associate Professor of Aeronautics and Astronautics

Acknowledgments

To my wife, Suzanne. Forgive me, but the magnitude of my appreciation is simply not expressible so I must leave these words by themselves.

A thesis is hardly an individual effort. From borrowed pencils to grueling power sessions there are many ways in which others contribute. I thank you all. Of course, there are always those who deserve special mention. So, special thanks to:

Dave Darmofal. For being so darned inspirational. My greatest regret in deciding not to begin a PhD program is in giving up the opportunity to work alongside someone so capable of achievement. I will miss our varied discussions.

Mark Drela, my advisor. With his unwavering technical competence he tutored me in the ways of Newton and Raphson and bashing and clamping and choking and all those wonderful things that really make CFD so colorful. I would especially like to thank him for his patience with my tendency to produce somewhat persistant program bugs.

Mr. Harold H. Youngren. For making me excited about being 38.

Bob Haimes. For providing some occasional computational advice and the very frequent beatings on the squash court. I regret not being able to win just one of our many matches.

My parents. For extending their motherly concern and fatherly advice some 2000 miles across the country to help Suzanne and I through this rather significant deattachment from our roots ("Suzy, I don't think we're in Texas anymore.")

This work was supported by the National Science Foundation under the NSF PYI program.

Contents

List of Figures

List of Tables

Nomenclature

A	geometric mode coefficient
B	number of blades
C_p	pressure coefficient
C_P	power coefficient
D	propeller diameter
f	geometric mode
h	local enthalpy
I	rothalpy
i, j, k	grid indexing coordinates (streamwise, blade-to-blade, radial)
$i_{max}, j_{max}, k_{max}$	grid size
k_b	number of radial stations on the blade
J	advance ratio ($= V_\infty/nD$, where n is rotations/sec)
M	Mach number
M_{crit}	critical local Mach number for artificial dissipation
N	finite element shape function
Q	Kutta condition residual
R	tip radius OR residual at a node. Usage is clear from context.
r_h	hub radius
V_w	transpiration velocity along design surface
V_∞	translational speed of propeller (along x-axis)
$\vec{V}_{adv}$	advance velocity
W	finite element weight function
X, Y, Z	absolute coordinate frame
x, y, z	coordinate frame attached to rotating propeller
$\vec{W}$	relative velocity (in (x, y, z) frame)

Nomenclature (continued)

β	blade pitch
γ	ratio of specific heats
Γ	circulation
$\tilde{\Delta}$	normal displacement of design surface
μ	artificial dissipation coefficient
μ_c	constant in expression for μ
λ	advance ratio $(= V_\infty/\lvert\vec{\Omega}\rvert R)$
λ_w	wake advance ratio
ω	underrelaxation factor for slowing Newton changes
$\vec{\Omega}$	angular velocity of propeller
ϕ	perturbation potential
ρ	density
$\bar{\rho}$	density normalized by freestream value
$\tilde{\rho}$	upwinded density
ξ, η, ζ	natural coordinates of undistorted elements

Chapter 1
Introduction

This thesis presents a three-dimensional analysis and design method based on the full-potential equation. The goal in developing the present method was to reduce the typically long computation times required for a three-dimensional design calculation in which an initial geometry is modified to satisfy a specified target pressure distribution. An excellent review of aerodynamic design methods is given in a paper by R.C. Lock [10] in which two and three-dimensional direct and inverse methods are discussed for transonic flow. The reader is referred to this paper for more information on previous work in transonic design.

For any design method, considerable effort is required each design cycle to 1) specify the new geometry for the flow solver (i.e. re-gridding) and 2) compute a direct solution for the new shape. The first of these tasks can be avoided by modeling geometry changes with a distribution of a wall blowing velocity along the original shape. This technique is often used in viscous-inviscid interaction methods to model the displacement effect of a boundary layer and has been used by other design methods to model geometry perturbations. A grid is generated for the new shape only at the end of the design procedure to verify the wall blowing approximation.

The second task, computing a new direct solution each time the geometry is modified, is by far the most expensive step in a design method. In the present method, the solution is updated without computing a full direct solution. Instead, updating the solution requires inverting a small system of algebraic equations (~ 20) and computing a series of algebraic operations for each grid point. The geometry is restricted to move according to a fixed set of distortion modes so that the resulting shape is completely specified by a fixed number of mode coefficients. The mode coefficients are introduced as global unknowns into a Newton-Raphson system for the full analysis solution. The Newton-Raphson method efficiently provides sensitivities to global unknowns so that the change in the analysis solution (to first order) is simply an algebraic function of the changes to the mode coefficients. The details of this process are described in Chapter 6.

The purpose of the geometric modes is to permit rapid design changes to be made. After an initial non-linear solution is obtained, a modified geometry and an approximate, linearly perturbed solution can be computed without regridding the domain or recomputing the full non-linear analysis solution. The linearized perturbations can be made exact by reconverging the non-linear solution when determined necessary. The non-linear convergence step is quick, since the linearized perturbations provide a very good initial guess for the non-linear algorithm.

The specific application investigated by the present research is the analysis of transonic propellers. One example is the Prop-Fan (a relatively highly loaded, multi-bladed advanced turboprop) that has been proposed as an alternative to turbofans for modern transport aircraft. Interest in the design of Prop-Fans increased drastically in the mid to late 1970's due to increasing costs of aviation fuel. Its projected fuel savings on an aircraft cruising at Mach 0.8 is 15 to 30 percent over an equivalent technology turbofan [12]. The operating conditions of the Prop-Fan guarantee that the flow around the outer portions of the propeller blades will be transonic. The compressibility effects near the tip must be considered by any analysis/design tool aimed at this application in order for the method to be effective.

A very different example of propeller driven aircraft designed for transonic flow is a series of high altitude research aircraft being developed to conduct experiments on the upper atmosphere [11, 2]. Theseus is one such aircraft and is being designed for an operating altitude of 100,000 feet. The density of air at these extreme altitudes is so low that these planes will fly at transonic speeds simply to produce enough lift to sustain level flight. Fuel economy is a primary concern since flight times as high as 44 hours will be required to conduct the atmospheric research. Designing efficient propellers for these aircraft will require paying attention to minimizing losses from transonic effects near the tip.

This thesis develops an analysis/design method aimed at the design of transonic propellers like the examples given above. The analysis algorithm is independent of the design procedure, the latter being only a modification to the boundary conditions. The description of the overall method therefore begins with the analysis method alone. Chapter 2 discusses the full potential equation and its application to transonic propeller flow. The full potential equation is discretized on a computational grid using finite element techniques. The discretization is described in Chapter 3 including special treatment for nodes on the boundaries. Chapter 3 also discusses the solution of the non-linear system of discrete equations using the Newton-Raphson iteration method. Chapter 4 presents some results that demonstrate the analysis capability of the presented scheme. A brief introduction to the design philosophy and the intended use of the present method is given in Chapter 5 before the details of the design procedure are described in Chapter 6. Finally, the design method is demonstrated in Chapter 7 and concluding remarks are given in Chapter 8.

Chapter 2

Physics/Governing Equations

Consider a propeller rotating with angular velocity $\vec{\Omega}$ and translating in a direction parallel to its axis of rotation with speed V_∞. Since the axis of rotation is aligned to the oncoming freestream, the resulting flow through the propeller is periodic in rotation with period equal to the blade spacing angle. Two coordinate systems are useful and these are illustrated in Figure 2.1.

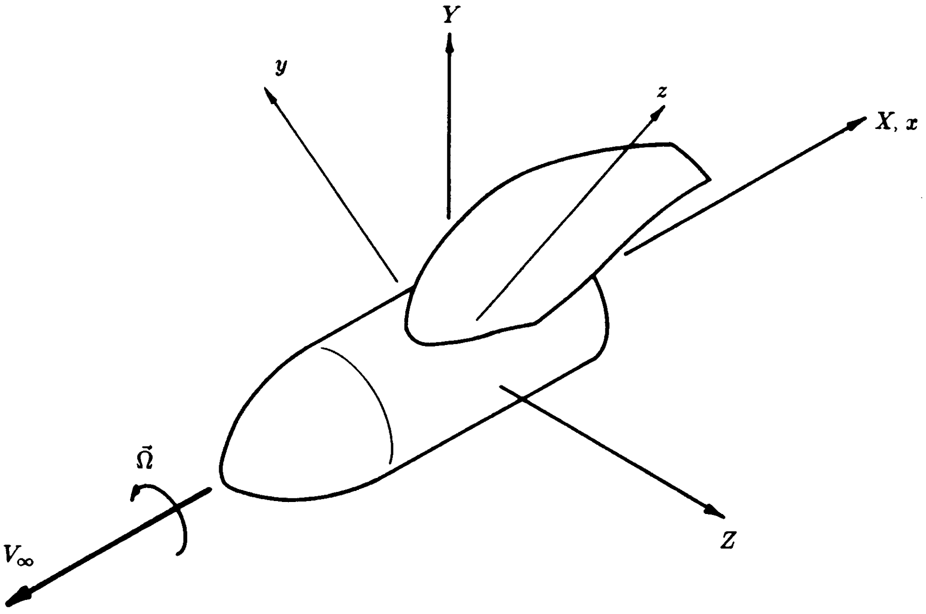


Figure 2.1: Coordinate systems for propeller problem.

In the absolute coordinate system (X, Y, Z) the propeller advances through the air with velocity $\vec{V}_{adv} = V_\infty \, \hat{\imath} + \vec{\Omega} \times \vec{r}$. The second coordinate system (x, y, z) is attached to the rotating propeller. In this coordinate frame, the propeller 'sees' an oncoming velocity field given by $-\vec{V}_{adv}$. The governing equations for the propeller flowfield are presented in the frame of reference attached to the propeller.

The total relative velocity at any point in the flowfield $\vec{W}$ is the sum of the oncoming stream and the perturbations caused by the propeller. The perturbation velocity is assumed to be governed by a potential so that $\vec{W}$ can be expressed as

$$\vec{W} = \nabla\phi - \vec{V}_{adv}$$

The assumption of potential flow places limitations on the type of flow for which the present model is applicable. In the strictest sense, potential flow is inviscid and isentropic and has constant total enthalpy (or constant rothalpy for the rotating flowfield through a propeller). In general, entropy increases across a shock wave so the potential approximation is not strictly valid for shocked transonic flows. However, for transonic flows where the supersonic region is terminated by weak shock waves, the flow can be reasonably well approximated by neglecting the increase in entropy. In practice, the full potential equation is used routinely to get satisfactory results for many transonic flow problems. For design purposes, since strong shock waves are not a desirable feature for good propeller performance, the isentropic assumption should be entirely adequate. Large viscous effects are confined to a thin boundary layer and a thin wake trailing behind the propeller blade, provided no separation occurs on the blade. Although the outer flowfield can be closely approximated by inviscid flow, the effect of the boundary layer can be significant. Propeller sections tend to be very thin and cambered near the blade tip. The boundary layer tends to uncamber the trailing edges of these sections and thus affect the radial load distribution and the overall propeller performance. The viscous effects are not considered by the current method, so results must be interpreted carefully.

For a rotating flowfield, the rothalpy (I) is defined as

$$I = h + \frac{|\vec{W}|^2}{2} - \frac{|\vec{\Omega}|^2 r^2}{2}$$

In the above definition, h is the local enthalpy of the flow. The assumption of constant rothalpy does not introduce errors into the approximation but limits the applicability to flows with uniform freestream rothalpy.

When the flow is assumed to be inviscid and isentropic and to have constant rothalpy, the momentum equation reduces to a statement that the perturbation velocity is irrotational and is therefore governed by a potential. Conservation of mass can be written as

$$\nabla \cdot [\bar{\rho}(\nabla \phi - \vec{V}_{adv})] = 0 \qquad (2.1)$$

Equation 2.1 is the familiar full potential equation. An expression for the density ($\bar{\rho}$) can be derived using the definition of rothalpy and the assumption of isentropic flow.

$$\bar{\rho} = \frac{\rho}{\rho_\infty} = \left[1 + \frac{\gamma - 1}{2} M_\infty^2 \left(1 - \frac{|\vec{W}|^2}{V_\infty^2} + \frac{|\vec{\Omega}|^2}{V_\infty^2}\right)\right]^{\frac{1}{\gamma - 1}}$$

The full potential equation is nonlinear due to the dependence of $\bar{\rho}$ on ϕ.

Chapter 3

Discretization and Solution of Equations

This chapter discusses the procedure used to solve the full potential equation on a discrete domain for the propeller problem. The analysis problem is discussed to present the basic application of the finite element method to discretize the governing equations. The discussion includes the treatment of the boundary conditions, implementation of the Kutta constraints, and the solution of the resulting nonlinear discrete equations with a Newton-Raphson iteration procedure.

3.1 Grid System

The propeller domain is represented by a discrete number of grid nodes and the continuous full potential equation (Equation 2.1) is discretized over these nodes. For all flows considered in this thesis, the flowfield is periodic in the blade-to-blade direction with period equal to $2\pi/B$ where B is the number of propeller blades. All grids used in this thesis are structured grids. The finite element method can be applied very naturally to an unstructured grid system, however the generation of unstructured grids for complex geometries is itself the subject of much current research in computational fluid dynamics and is not within the scope of this thesis. The grid indexing coordinates are such that i is in the streamwise direction, j is in the blade-to-blade direction, and k extends radially outward from the hub. The grid coordinates are referenced to the coordinate system attached to the propeller (x, y, z). The grid is generated as follows.

Two dimensional (constant radius, constant k), H-type grids are generated for k_b radial stations along the blade. Each two-dimensional grid describes the local cascade domain at radius r_k. The blade upper surface (suction side) is the $j = j_{max}$ boundary and the lower surface is the $j = 1$ boundary. The wake advance ratio (or, equivalently, the pitch of the trailing helical wake) is prescribed by the user and is usually obtained from a lifting line model for a propeller at a given loading. All streamwise grid lines ($j =$ constant) are parallel to the wake at the outflow boundary. A sample two-dimensional grid is illustrated in Figure 3.1.

The last two-dimensional grid on the propeller defines the blade tip. The blade section at the tip is collapsed to zero thickness so that the entire $j = 1$ and $j = j_{max}$ grid lines are periodic. The camber of the tip section is retained. The two-dimensional, constant radius grids are simply stacked radially to form the three-dimensional computational grid. The two-dimensional grid at the propeller tip is copied (after scaling for larger radius) to radial stations off the tip of the propeller to some prescribed tip farfield radius. The three-dimensional grid is mapped into (x, y, z) space to complete the grid generation. A sample grid is shown in Figure 3.2.

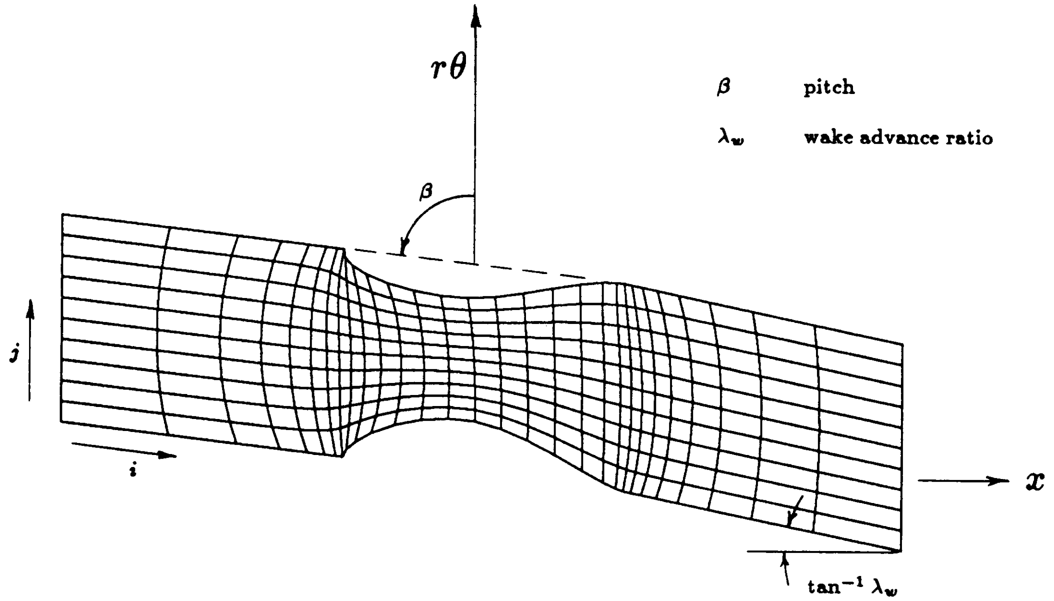


Figure 3.1: Constant radius cascade grid.

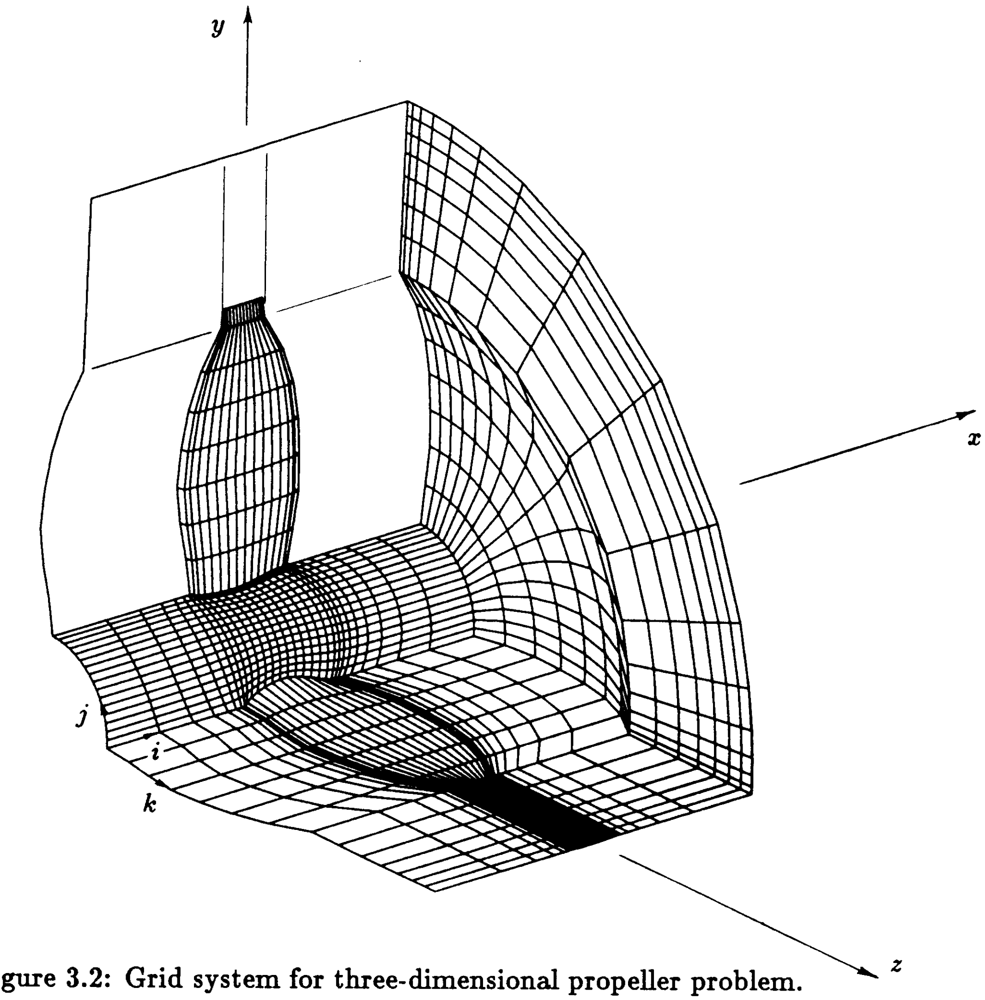


Figure 3.2: Grid system for three-dimensional propeller problem.

3.2 Application of Finite Element Method

The grid divides the domain into hexahedral elements with 8 nodes each. The perturbation potential (ϕ) is represented functionally within an element by

$$\phi^e = \sum_{i=1}^{8} N_i \phi_i^e$$

$$N_i = \frac{1}{8}(1 + \xi\xi_i)(1 + \eta\eta_i)(1 + \zeta\zeta_i)$$

where the superscript e indicates a given element. The trilinear shape functions N_i are defined in the natural coordinate system (ξ, η, ζ) of an undistorted element as illustrated in Figure 3.3. The shape function for a given node is 1 at that node and varies linearly to 0 at each neighboring node. At each point in the element, the sum of the 8 nodal shape functions is 1 so the approximation is consistent. The variation of ϕ within an element, then, is a weighted average of the values of ϕ at the nodes of the element.

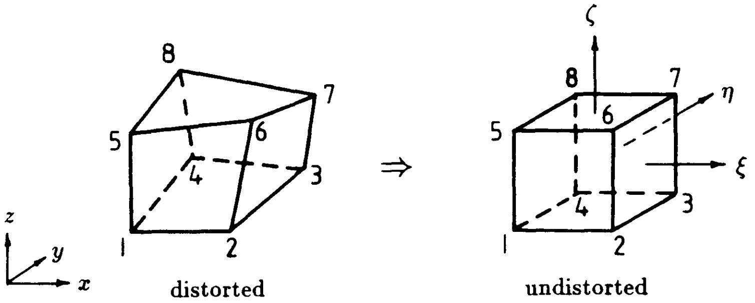


Figure 3.3: Natural coordinate system of undistorted element.

The concept of isoparametric elements is adopted so that the physical coordinates (x, y, z) are represented functionally in the same manner as ϕ.

$$x^e = \sum_{i=1}^{8} N_i x_i^e \qquad y^e = \sum_{i=1}^{8} N_i y_i^e \qquad z^e = \sum_{i=1}^{8} N_i z_i^e$$

Application of the Galerkin weighted residual method to the full potential equation (2.1) consists of volume integrals of the form

$$\int_D W \left\{ \nabla \cdot [\bar{\rho}(\nabla\phi - \vec{V}_{adv})] \right\} dV = 0. \tag{3.1}$$

The discrete equation for each node in the domain is obtained by choosing a separate weight function W for each node. The weight function is chosen to be the same trilinear shape function described above to represent ϕ. For a given node ℓ, $W_\ell = 1$ at ℓ and varies linearly to zero for all immediately neighboring nodes. $W = 0$ for all other nodes in the domain. With this choice of weight function, Equation 3.1 for node ℓ can be expressed as

$$\sum_{e=1}^{8} \int_{V_e} N_\ell^e \left\{ \nabla \cdot [\bar{\rho}(\nabla\phi - \vec{V}_{adv})] \right\} dV = 0 \tag{3.2}$$

where the sum is taken over the 8 elements that share node ℓ. N_ℓ^e is the shape function for the node of element e that corresponds to node ℓ in the global node numbering system.

The next step is to substitute the functional form of ϕ into Equation 3.2. The equation, however, has second derivatives that cannot be represented with linear shape functions. This problem is avoided by integrating the integrals of Equation 3.2 by parts, giving

$$\sum_{e=1}^{8}\left\{\int_{V_e} \nabla N_\ell^e \cdot [\bar{\rho}(\nabla\phi - \vec{V}_{adv})]\; dV - \oint_{S_e} N_\ell^e \bar{\rho}(\nabla\phi - \vec{V}_{adv}) \cdot \hat{n}\; dS\right\} = 0 \qquad (3.3)$$

where $\hat{n}$ is the outward pointing normal for the element. At this point, the density is assumed to be constant over a given element so that Equation 3.3 becomes

$$\sum_{e=1}^{8}\left\{\bar{\rho}^e \int_{V_e} \nabla N_\ell^e \cdot (\nabla\phi - \vec{V}_{adv})\; dV - \bar{\rho}^e \oint_{S_e} N_\ell^e\; \vec{W} \cdot \hat{n}\; dS\right\} = 0 \qquad (3.4)$$

where $\vec{W}$ $(= \nabla\phi - \vec{V}_{adv})$ is the relative velocity as before.

Substituting the functional form of ϕ into Equation 3.4 gives

$$\sum_{e=1}^{8}\left\{\bar{\rho}^e \int_{V_e} \nabla N_\ell^e \cdot \left[\left(\sum_{j=1}^{8} \nabla N_j^e \phi_j^e\right) - \vec{V}_{adv}\right]\; dV - \bar{\rho}^e \oint_{S_e} N_\ell^e\; \vec{W} \cdot \hat{n}\; dS\right\} = 0$$

which can be written as

$$\sum_{e=1}^{8}\left\{\bar{\rho}^e \left(\sum_{j=1}^{8} \phi_j^e k_{\ell j}^e\right) - \bar{\rho}^e f_{v_\ell}^e - \bar{\rho}^e \oint_{S_e} N_\ell^e\; \vec{W} \cdot \hat{n}\; dS\right\} = 0 \qquad (3.5)$$

where

$$\begin{aligned} k_{ij}^e &= \int_{V_e} \nabla N_i \cdot \nabla N_j\; dV \\ f_{v_\ell}^e &= \int_{V_e} \nabla N_\ell^e \cdot \vec{V}_{adv}\; dV \end{aligned}$$

In practice, the 8 × 8 matrix of integrals k_{ij} and the 8 integrals f_{v_ℓ} are precalculated and stored for each element in the domain. The integration is performed over the natural cell using 2 × 2 × 2 point numerical Gaussian integration. Note that k_{ij} is symmetric.

The sum of the surface integrals in Equation 3.5 is zero when ℓ is an interior node because $N_\ell = 0$ on the surface of the supercell for node ℓ. The supercell is the volume formed by the 8 cells that share node ℓ. Contributions for surfaces where N_ℓ is nonzero sum to zero due to internal surface cancellations. In general, the surface integral in Equation 3.5 is nonzero only when node ℓ lies on a boundary of the domain. Therefore, Equation 3.5 for a general interior node reduces to

$$\sum_{e=1}^{8} \left\{ \bar{\rho}^e \left(\sum_{j=1}^{8} \phi_j^e k_{\ell j}^e \right) - \bar{\rho}^e f_{v_\ell}^e \right\} = 0 \tag{3.6}$$

Equation 3.6 is the discretized form of Equation 3.1. It is a nonlinear equation in ϕ due to the dependence of density on ϕ. Equation 3.6 is expressed from the viewpoint of visiting a node and summing over the 8 elements surrounding the node. In practice, each element of the domain is visited once and computed quantities are distributed to the nodes of that element.

3.3 Boundary nodes

Figure 3.4 shows the computational domain for a generic propeller. The different types of domain boundaries are indicated in the figure. Table 3.1 presents a summary of the discrete equations used for nodes on the various domain boundaries. The different types of boundaries are discussed following the table.

The 'natural', or mass flux, boundary condition for a node on a domain boundary is applied through the surface integral term of Equation 3.5 $(\ldots\bar{\rho}^e \oint_{S_e} N_\ell^e\ \vec{W}\cdot\hat{n}\ dS)$. Recall that for all interior nodes this term is identically zero because N_ℓ is zero on the supercell surface. However, for a node on a domain boundary, N_ℓ is non-zero across the domain surface. If there is mass flux across this surface, then the surface integral will contribute to the residual for the surface node. The natural boundary condition is illustrated for a two-dimensional domain in Figure 3.5.

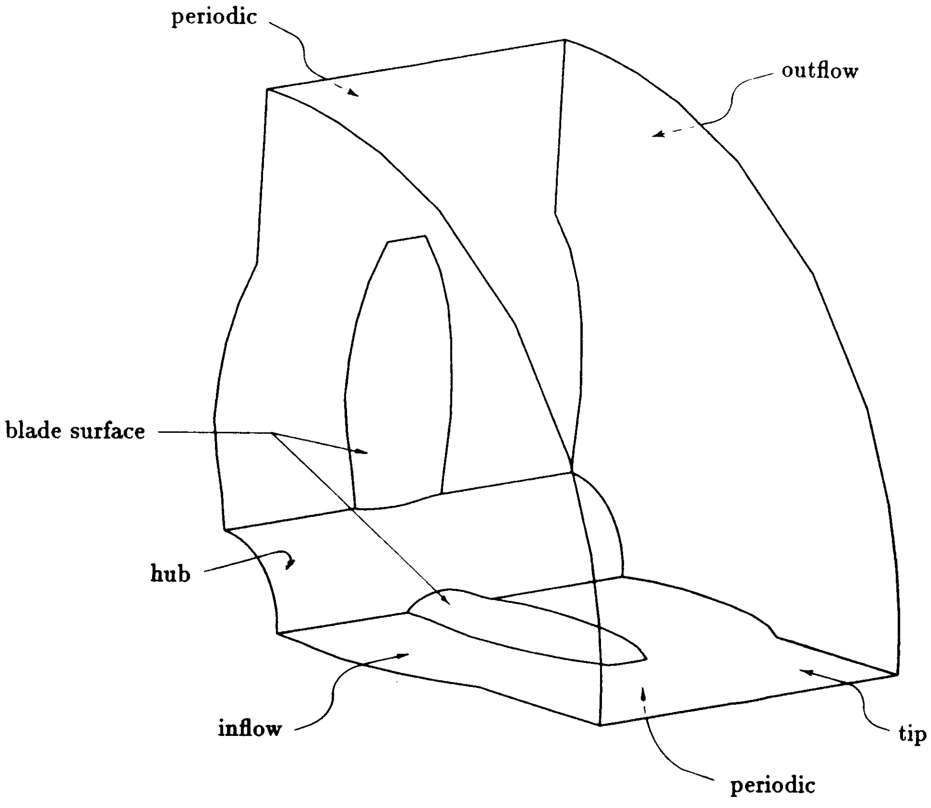


Figure 3.4: Computational domain and boundaries.

Table 3.1: Summary of Discrete Equations for Boundary Nodes.

Boundary Type	Discrete Equation
solid surface	$\sum_{e=1}^{\leq 4} \left\{ \rho^e \left(\sum_{j=1}^{8} \phi_j^e k_{\ell j}^e \right) - \rho^e f_{v_\ell}^e \right\} = 0$
periodic, $j = 1$	$\sum_{e=1}^{\leq 8} \left\{ \rho^e \left(\sum_{j=1}^{8} \phi_j^e k_{\ell j}^e \right) - \rho^e f_{v_\ell}^e \right\} = 0$
periodic, $j = j_{max}$	$\phi_{j_{max}} - \phi_1 = 0$ (general) $\phi_{j_{max}} - \phi_1 - \Gamma_k = 0$ (on wake)
inflow (farfield)	$\phi - \phi_{ff} = 0$
tip (farfield)	$\sum_{e=1}^{\leq 4} \left\{ \rho^e \left(\sum_{j=1}^{8} \phi_j^e k_{\ell j}^e \right) - \rho^e f_{v_\ell}^e - \rho^e \oint_{S_e} N_\ell^e \ \vec{W}_{ff} \cdot \hat{n} \, dS \right\} = 0$
outflow	$\phi_{i_{max}} - \phi_{i_{max}-1} = 0$

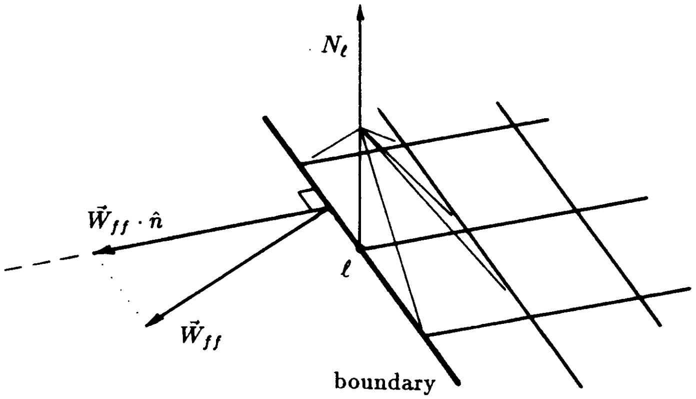


Figure 3.5: The natural boundary condition for a two-dimensional domain.

Not all domain boundaries are correctly treated by applying the natural boundary condition. The nodes on these boundaries require special treatment to enforce some other physical condition besides a specified mass flux.

3.3.1 Solid Surfaces

Nodes that lie on the propeller hub and blade are solid surface nodes. The physical boundary condition for these nodes is that the relative velocity must be tangent to the surface, which implies that $\vec{W} \cdot \hat{n} = 0$. The natural boundary condition correctly treats nodes on the solid surfaces since the condition of no mass flux across the solid boundary is specified when the surface integral in Equation 3.5 is identically zero. Note that the supercell for a boundary node includes at most 4 cells and as few as 1 (e.g. in domain corners). The discrete equation for solid surface nodes, then, is the same as for an interior node (3.6) but with the sum taken over at most 4 elements.

3.3.2 Periodic Boundaries

The flow through the propeller is periodic in the blade-to-blade direction. Only one period of the domain is represented by the grid and periodic boundary conditions are applied at the $j = 1$ and $j = j_{max}$ boundaries illustrated in Figure 3.6. The periodic boundaries are not physical boundaries of the flow so the surface integral $\bar{\rho}^e \oint_{S_e} N_\ell^e \, \vec{W} \cdot \hat{n} \, dS$

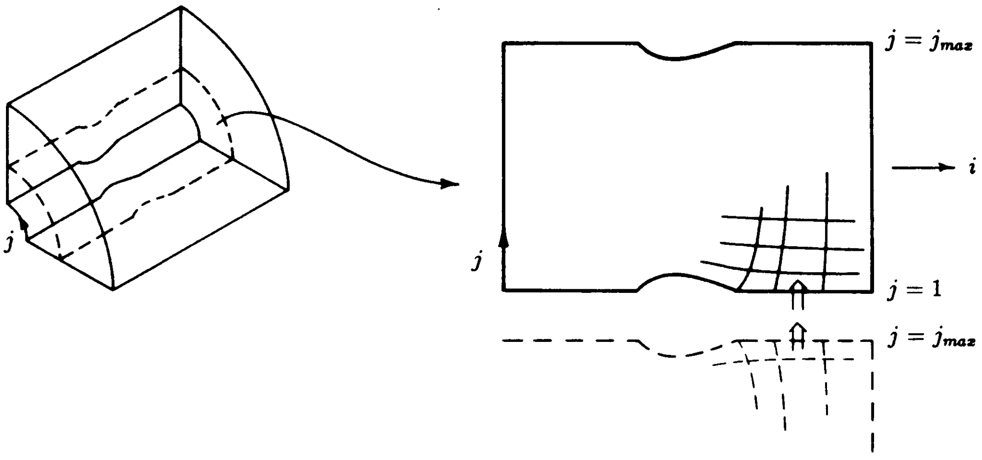


Figure 3.6: Periodic domain boundaries.

is not generally zero. However, the value of the integral for a face on the j=1 boundary is equal and opposite to the value for the corresponding face on the $j = j_{max}$ boundary since $\left(\vec{W}\cdot\hat{n}\right)_{j=1} = -\left(\vec{W}\cdot\hat{n}\right)_{j=j_{max}}$. Physically, what flows out of the $j = 1$ face must flow into the $j = j_{max}$ face if the boundaries are periodic.

When the discrete equation for a node at $j = j_{max}$ is added to the equation for the corresponding node at $j = 1$, the surface integrals for faces on the periodic boundaries exactly cancel each other. The new discrete equation at $j = 1$ is the same as Equation 3.6 for an interior point. A statement of periodicity ($\phi_{j_{max}} = \phi_1$) replaces the equation at $j = j_{max}$. The treatment of periodic boundaries is summarized below.

before applying periodic condition:

$$j = 1 \qquad \sum_{e=1}^{\leq 4}\left\{\bar{\rho}^e\left(\sum_{jj=1}^{8}\phi^e_{jj}k^e_{\ell jj}\right) - \bar{\rho}^e f^e_{v_\ell} - \bar{\rho}^e\oint_{S_e} N^e_\ell\,\left(\vec{W}\cdot\hat{n}\right)_{j=1}\,dS\right\} = 0$$

$$j = j_{max} \qquad \sum_{e=1}^{\leq 4}\left\{\bar{\rho}^e\left(\sum_{jj=1}^{8}\phi^e_{jj}k^e_{\ell jj}\right) - \bar{\rho}^e f^e_{v_\ell} - \bar{\rho}^e\oint_{S_e} N^e_\ell\,\left(\vec{W}\cdot\hat{n}\right)_{j=j_{max}}\,dS\right\} = 0$$

after applying periodic condition:

$$j = 1 \qquad \sum_{e=1}^{\leq 8} \left\{ \bar{\rho}^e \left(\sum_{\ddot{j}=1}^{8} \phi^e_{\ddot{j}} k^e_{\ell \ddot{j}} \right) - \bar{\rho}^e f^e_{v_\ell} \right\} = 0$$

$$j = j_{max} \qquad \phi_{j_{max}} - \phi_1 = 0$$

$$\phi_{j_{max}} - \phi_1 - \Gamma_k = 0 \qquad \text{(on propeller wake)}$$

Note that nodes which lie on the wake sheet are periodic with a jump in ϕ equal to the circulation on the propeller blade. The circulation at each radial station on the propeller blade is a global unknown Γ_k, $k = 1, \ldots, k_b$ where k_b is the number of radial stations on the blade. The details of finding Γ_k are discussed in Section 3.5.

3.3.3 Inflow and Tip

The inflow and tip computational boundaries are farfield boundaries. Physically, all perturbations are zero far upstream and far off the tip of the propeller. Because the computational domain is finite, the physical condition of zero perturbation is not an accurate boundary condition for the farfield boundaries. Instead, an analytic model for the leading effect of the propeller is used to calculate perturbations at the farfield boundaries.

A loaded propeller has some distribution of circulation along the propeller blade. As the circulation changes along the blade, vortex filaments are formed that trail behind the propeller in helical paths. The system of helical vortex filaments forms the wake of the propeller. In the present method, the circulation Γ is defined at a discrete number of stations along the propeller blade. The wake is modeled as a discrete number of vortex filaments corresponding to the discrete changes in circulation along the blade as illustrated in Figure 3.7.

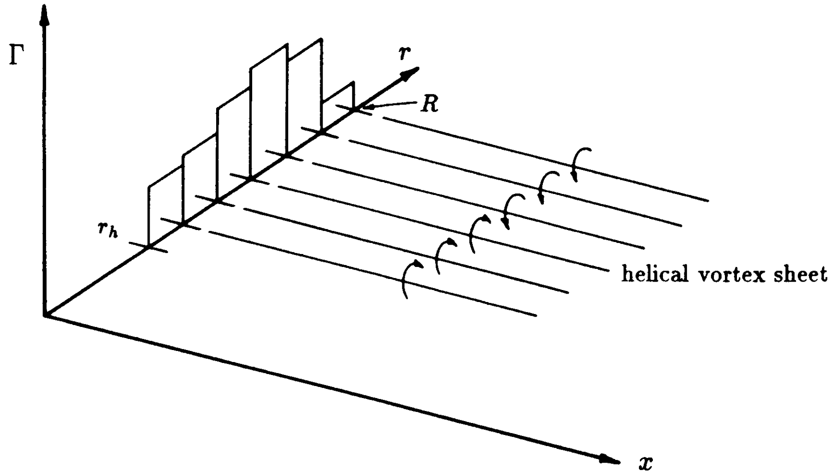


Figure 3.7: Discrete model of trailing vortices.

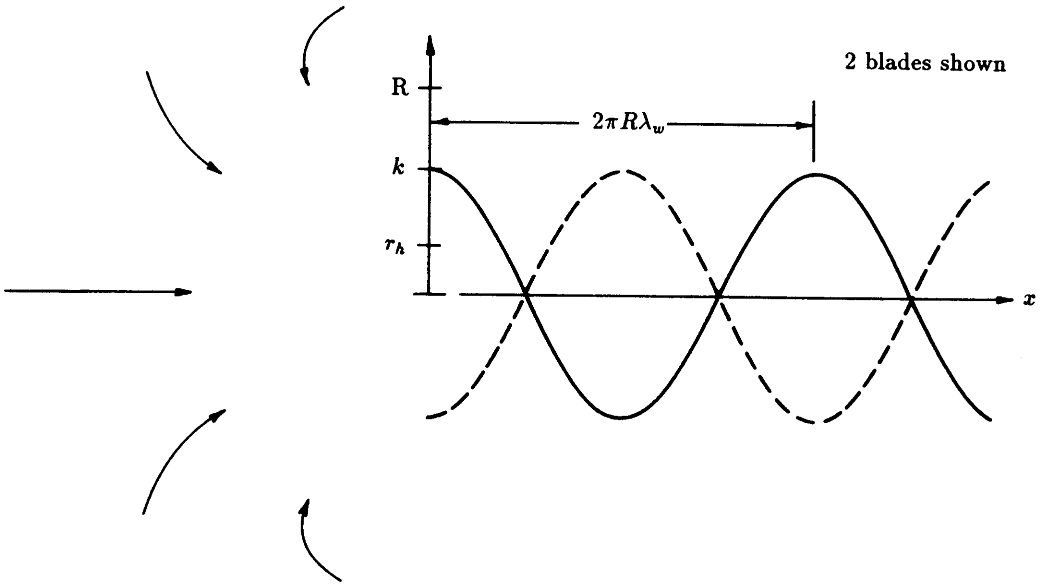
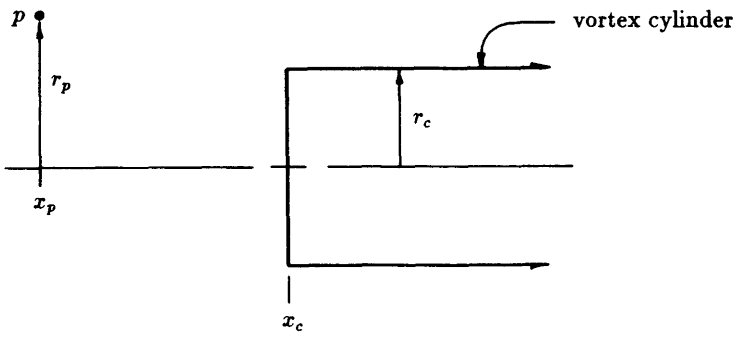


Figure 3.8: Approximate wake model.

Consider the trailing vortex filament at station k. Neglecting the axial component of vorticity, the helical filaments can be approximated as a system of vortex rings extending streamwise to $x = +\infty$. Figure 3.8 illustrates that in $2\pi\lambda_w R$ of wake, there are B vortex rings of strength $\Gamma_k - \Gamma_{k-1}$ as well as a corresponding system of image vortex rings at $r = \frac{r_h^2}{r_k}$ to account for the effect of the solid hub of radius r_h. The farfield effect of this system of vortex rings is the same as for a semi-infinite cylinder with circulation density $\frac{\partial\Gamma}{\partial x} = \frac{B(\Gamma_k - \Gamma_{k-1})}{2\pi\lambda_w R}$ at $r = r_k$ and a conjugate semi-infinite vortex cylinder with the same circulation density (opposite sign) at $r = \frac{r_h^2}{r_k}$. In his Ph.D. thesis, Scully [13, pages 149-151] developed the following approximation for the potential induced by a semi-infinite vortex cylinder:

$$\phi_p \cong \frac{1}{4}\frac{\partial\Gamma}{\partial x}\left(\frac{r_c}{\sqrt{r_p^2 + (x_c - x_p)^2}}\right) \tag{3.7}$$



Equation 3.7 can be used directly to specify a Dirichlet boundary condition at the farfield (i.e. $\phi_{ff} = \phi_p$) or the equation can be differentiated to obtain an expression for the axial and radial components of the velocity induced by the vortex cylinder. The induced velocity at the farfield can be used to apply a natural boundary condition for nodes on farfield boundaries. The actual conditions at the farfield boundary are a simple superposition of the effects from the trailing vortex cylinder at each discrete radial station.

It should be noted that the distribution of circulation along the propeller blade changes as the solution develops. The farfield boundary conditions are updated at the end of each Newton iteration cycle (see Section 3.5). The dependence of the residual equation for farfield boundary nodes on the circulation distribution is known, so solution convergence is not slowed by updating the farfield boundary conditions.

3.3.4 Outflow

The physical boundary condition in the wake downstream of a propeller is that all perturbations along helicoidal lines vanish. The grid system used in this research has i-lines which coincide with helicoidal lines near the outflow boundary. Thus, the condition of helical symmetry is obtained numerically by simply setting $\phi_{i_{max}-1} = \phi_{i_{max}}$.

3.4 Artificial Compressiblity

The discretization discussed thus far works only in subcritical flows. When the local Mach number exceeds unity, some sort of artificial dissipation is necessary to guarantee a unique solution. Since potential flow is fully isentropic, non-physical expansion shocks are valid solutions to the equations along with the usual compression shocks. Physically, the requirement that entropy must always increase eliminates the possibility of expansion shocks in real flows. This physical condition is not satisfied automatically by the full potential equation. The consideration of artificial dissipation for finite element methods in transonic flow is well presented in a paper by Deconinck and Hirsch [3].

Several methods for introducing the necessary dissipation into discrete equations have been successfully adopted. For finite element methods, a modified form of the density can be introduced that allows the standard Galerkin treatment to be applied in both subsonic and transonic flow regions [3]. In [6], Habashi and Hafez discuss several techniques for introducing viscosity into a finite element discretization of the full potential equation in two dimensions. These authors suggested modifying the full potential equation as follows:

$$\nabla \cdot \left[\tilde{\rho}(\nabla\phi - \vec{V}_{adv})\right]$$

$$\tilde{\rho} = \bar{\rho} - \mu \left(\frac{\partial \bar{\rho}}{\partial s}\right) \Delta s$$

After numerical experimentation with the operator μ, Habashi and Hafez concluded that the following form produced the best results for the two-dimensional full potential flow

$$\mu = \max(0, \mu_i, \mu_{i-1})$$

$$\mu_i = 1 - \frac{1}{M_i^2}$$

where i is an index in the stream direction.

A slightly modified form of the above scheme is used in this thesis. First, the grid coordinates in the stream direction (i) are assumed to be an adequate approximation to the actual streamlines of the flow in order to simplify the expression for $\tilde{\rho}$.

$$\tilde{\rho} = \bar{\rho}_i - \mu(\bar{\rho}_i - \bar{\rho}_{i-1})$$

Next, the operator μ_i is defined as

$$\mu_i = \mu_c(1.0 - \frac{M_{crit}^2}{M_i^2})$$

The factors μ_c and M_{crit} adjust the amount of artificial dissipation being introduced into the solution as shown in Figure 3.9. In [5], Giles presents an analytic argument for choosing the proper value for μ_c so that numerical stability is obtained without excessive total pressure errors for Euler solutions in smooth supersonic regions. Giles demonstrated analytically and confirmed numerically that for the Euler equations, μ_c should be about $\frac{1.0}{\gamma+1}$. In this thesis, adequate dissipation is achieved with $\mu_c = 1.0$ and $M_{crit} = 0.95$. For analysis solutions, the dissipation requirements can be much lower depending on the severity of the discontinuities in the flow.

Section 3.5 discusses the solution of the discrete equations using the Newton-Raphson method. In Section 3.6, solution strategies for shocked flow are discussed that include using the parameters μ_c and M_{crit} to not only provide numerical stability but to also aid in the convergence rate of the Newton iteration method.

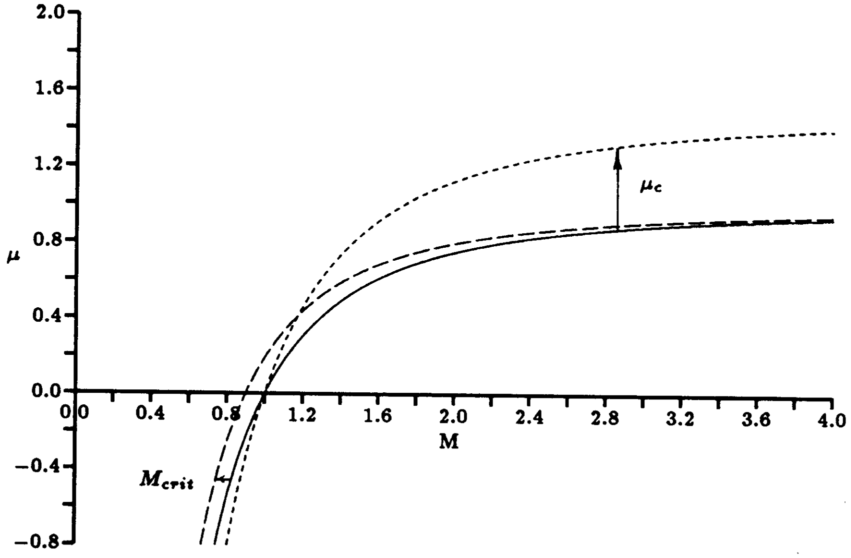


Figure 3.9: Dependence of μ on μ_c and M_{crit}.

3.5 Solving the Discrete Equations

Equation 3.6 and the equations in Table 3.1 form a set of non-linear discrete equations for the unknown nodal values of ϕ. These equations are solved iteratively using the Newton-Raphson method (this method will be referred to as Newton's method). At any given iteration, the current values of ϕ and Γ do not satisfy the discrete equations, so for each node there is some residual. In general, the residual at node ℓ is a function of ϕ and Γ everywhere in the domain.

$$R_\ell = R_\ell(\phi, \Gamma) \tag{3.8}$$

(Note that only residuals for nodes on farfield boundaries or on the wake at $j = j_{max}$ depend on Γ.) The system of equations expressed by Equation 3.8 is linearized around the current values of ϕ and Γ so that the residuals at the next iteration are approximated by

$$R_\ell^{n+1} = R_\ell^n + \left.\frac{\partial R_\ell}{\partial \phi}\right|^n \Delta\phi + \left.\frac{\partial R_\ell}{\partial \Gamma}\right|^n \Delta\Gamma \tag{3.9}$$

where n is an iteration counter.

Since the goal is to drive R_ℓ to 0, setting $R_\ell^{n+1} = 0$ in Equation 3.9 gives a linear system of equations for $\Delta\phi$ and $\Delta\Gamma$:

$$\left.\frac{\partial R_\ell}{\partial \phi}\right|^n \Delta\phi + \left.\frac{\partial R_\ell}{\partial \Gamma}\right|^n \Delta\Gamma = -R_\ell^n \tag{3.10}$$

Equation 3.10 is deceptively simple. The term $\frac{\partial R_\ell}{\partial \phi}$ is an $N \times N$ matrix (called the Jacobian matrix) where N is the number of nodes in the domain. For a three-dimensional problem, N can easily become very large ($\sim 10^5$). Also, the convergence rate of the Newton iteration scheme depends heavily on how accurately the Jacobian matrix is calculated. If the system of equations is exactly linearized, Newton's method exhibits quadratic convergence when ϕ and Γ are sufficiently close to the correct values. Small errors in the linearization of the equations can slow convergence considerably. The details of the linearization are placed in Appendix A. In general, each nodal residual (R_ℓ) depends on 36 nodal values of ϕ (contributions from 12 cells) with the extra 9 nodes (4 cells) due to the upwinding of the density in supersonic regions.

Newton's method can be summarized as follows:

1. Initialize ϕ, Γ ($= 0$ is adequate)
2. Calculate $-R_\ell^n$ based on current ϕ and Γ
3. Calculate derivates $\frac{\partial R}{\partial \phi}, \frac{\partial R}{\partial \Gamma}$
4. Invert the resulting linear system to obtain $\Delta\phi, \Delta\Gamma$
5. Update ϕ, Γ (i.e. $\phi^{n+1} = \phi^n + \Delta\phi$)
6. If $\Delta\phi, \Delta\Gamma$ are less than a prescribed tolerance, then system is converged; else, return to step 2 and continue iteration

Steps 2-6 are referred to as a Newton cycle or step.

3.5.1 Treatment of Global Unknowns, Γ

One detail that remains to be discussed is the determination of the circulation along the blade (Γ). The circulation distribution introduces $k_b - 1$ global unknowns into the Newton system where k_b is the number of radial grid stations on the blade (the circulation at the tip is zero). The equations necessary to close the system are provided by expressing the Kutta condition at $k_b - 1$ stations along the blade. The Kutta-Joukowsky condition implies that the pressure loading at the trailing edge of the blade must be zero. For the full potential formulation in this thesis, this condition is satisfied by enforcing that the magnitude of the velocity at the trailing edge is the same on the upper and lower surfaces. Figure 3.10 illustrates how the velocity is extrapolated near the trailing edge to get an approximation for the trailing edge value. Note that the velocity is not interpolated across the trailing edge stagnation point.

The Kutta residual (Q_k) is expressed as

$$Q_k = \left|\vec{W}_{te_{j=1}}\right|^2 - \left|\vec{W}_{te_{j=j_{max}}}\right|^2 \tag{3.11}$$

When discretized, Equation 3.11 is also a non-linear equation in ϕ since $\vec{W} = \nabla\phi - \vec{V}_{adv}$. Each Newton cycle, Q_k is linearized (Equation 3.12) to close the algebraic system of equations.

$$\left.\frac{\partial Q_k}{\partial \phi}\right|^n \Delta\phi = -Q_k^n \tag{3.12}$$

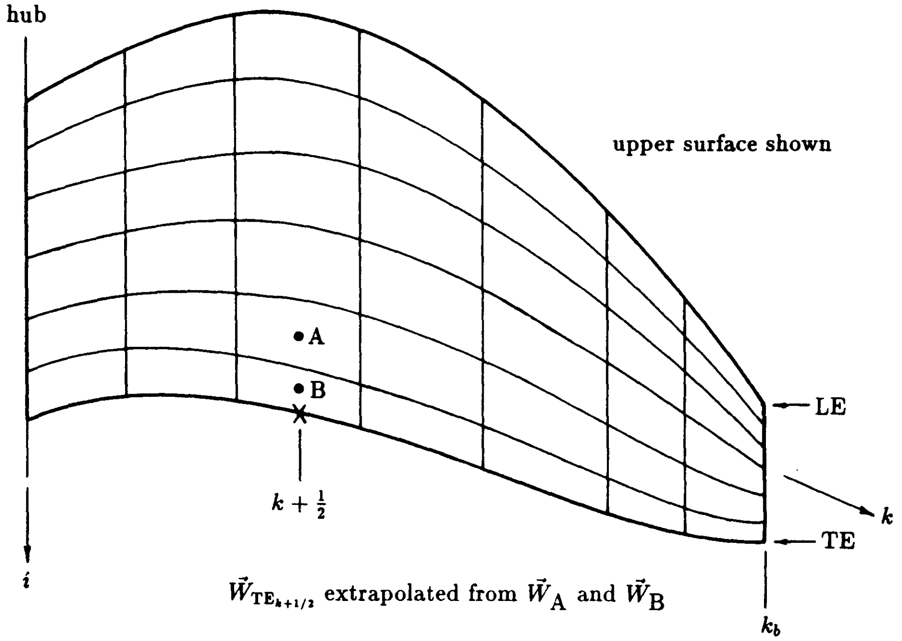

hub
upper surface shown
•A
•B
LE
$k + \frac{1}{2}$
k
TE
i
$\vec{W}_{\mathrm{TE}_{k+1/2}}$ extrapolated from $\vec{W}_{\mathrm{A}}$ and $\vec{W}_{\mathrm{B}}$
k_b

Figure 3.10: Calculation of velocity at trailing edge.

3.5.2 Solution Details

Equations 3.10 and 3.12 form a complete, linear system of equations that determine the nodal unknowns $\Delta\phi$ and the global unknowns $\Delta\Gamma_k$ for one Newton step. In matrix form, the system is written

$$\left[\begin{array}{c|c} \frac{\partial R}{\partial \phi}\Big|^n & \frac{\partial R}{\partial \Gamma}\Big|^n \\ \hline \frac{\partial Q}{\partial \phi}\Big|^n & 0 \end{array}\right] \left\{\begin{array}{c} \Delta\phi \\ \hline \Delta\Gamma \end{array}\right\} = \left\{\begin{array}{c} -R^n \\ \hline -Q^n \end{array}\right\} \tag{3.13}$$

Although Equation 3.13 represents a large system of equations, the Jacobian matrix $\left[\frac{\partial R}{\partial \phi}\right]$ is very sparse. The following procedure is used to compute $\Delta\phi$ and $\Delta\Gamma$ at each Newton cycle.

Equation 3.13 is written as

$$\left[\quad \frac{\partial R}{\partial \phi} \quad\right] \left\{\Delta\phi\right\} = \left\{-R\right\} + \sum_{k=1}^{k_b-1} \Delta\Gamma_k \left\{-\frac{\partial R}{\partial \Gamma_k}\right\} \tag{3.14}$$

The Jacobian matrix is effectively inverted and multiplied through Equation 3.14 to give

$$\left\{\Delta\phi\right\} = \left\{r\right\} + \sum_{k=1}^{k_b-1} \Delta\Gamma_k \left\{\frac{\partial \phi}{\partial \Gamma_k}\right\} \tag{3.15}$$

where

$$\left\{r\right\} = \left[\quad \frac{\partial R}{\partial \phi} \quad\right]^{-1} \left\{-R\right\}$$

$$\left\{ \frac{\partial \phi}{\partial \Gamma_k} \right\} = \left[\quad \frac{\partial R}{\partial \phi} \quad \right]^{-1} \left\{ -\frac{\partial R}{\partial \Gamma_k} \right\}$$

Since $\left[\frac{\partial R}{\partial \phi} \right]$ is large and sparse, its inverse is not computed directly. Instead, an iterative method (succesive line over relaxation or SLOR) is used to find $\left\{ r \right\}$ and $\left\{ \frac{\partial \phi}{\partial \Gamma_k} \right\}$. Implicit i relaxation sweeps are alternated with implict j-sweeps as shown in Figure 3.11 to help speed convergence of the matrix inversion.

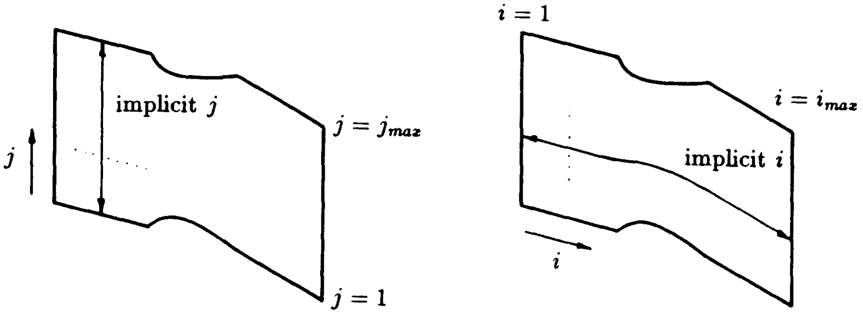


Figure 3.11: SLOR sweep directions for 1 iteration.

The j-sweeps quickly communicate the solid wall boundary conditions at $j = 1$ and $j = j_{max}$ into the domain. Also, since the periodic condition for nodes at $j = j_{max}$ $(\phi_{j_{max}} - \phi_1 = -R_{j_{max}})$ upsets the diagonal dominance of the overall system, the implicitness in the j-direction is necessary for the system to converge.

In general, a propeller grid will have a large number of points in the streamwise direction (i) so that shock waves are sharply defined. The number of points in the blade-to-blade direction (j) is less important and spacing in this direction can be quite large. Near the tip, these different spacing requirements can result in elements with an extremely high aspect ratio. The implicit-i sweeps help to overcome possible stiffness problems that result from the extreme aspect ratio of the distorted elements.

Equation 3.15 expresses each $\Delta\phi$ as a function of $\Delta\Gamma$ so there are $k_b - 1$ unknowns remaining. However, there are $k_b - 1$ remaining equations of the form

$$\left[\quad \frac{\partial Q}{\partial \phi} \quad\right]\left\{\Delta\phi\right\} = \left\{-Q\right\} \tag{3.16}$$

Substituting Equation 3.15 into Equation 3.16 gives

$$\left[\quad \frac{\partial Q}{\partial \phi} \quad\right]\left[\left\{r\right\} + \sum_{k=1}^{k_b-1} \Delta\Gamma_k \left\{\frac{\partial \phi}{\partial \Gamma_k}\right\}\right] = \left\{-Q\right\}$$

which can be written as

$$\left[\quad S_{ij} \quad\right]\left\{\Delta\Gamma\right\} = \left\{-Q\right\} - \left[\quad \frac{\partial Q}{\partial \phi} \quad\right]\left\{r\right\} \tag{3.17}$$

$$S_{ij} = \lfloor \quad \partial Q_j/\partial \phi \quad \rfloor \left\{\frac{\partial \phi}{\partial \Gamma_i}\right\}$$

Equation 3.17 is a simple linear system for the unknowns $\Delta\Gamma_k$. The matrix $[\ S\ ]$ is, in general, a full matrix. The system, however, is small since in practice $k_b \sim 10$. The system for $\Delta\Gamma$ is solved directly using Gaussian elimination.

A more detailed summary of the steps in one Newton cycle is outlined below.

1. Calculate $-R_\ell^n$ for current iteration based on current ϕ, Γ
2. Calculate derivatives $\frac{\partial R}{\partial \phi}, \frac{\partial R}{\partial \Gamma_k}$
3. Calculate $\left\{r\right\} = \left[\ \frac{\partial R}{\partial \phi}\ \right]^{-1}\left\{-R\right\}$ using SLOR
4. Calculate $\left\{\frac{\partial \phi}{\partial \Gamma_k}\right\} = \left[\ \frac{\partial R}{\partial \phi}\ \right]^{-1}\left\{\frac{\partial R}{\partial \Gamma_k}\right\}$ for $k = 1, \ldots, k_b - 1$ using SLOR
5. Compose $[\ S\ ]$ and right hand side of Equation 3.17

6. Solve for $\Delta\Gamma$ using Gaussian elimination

7. Use $\Delta\Gamma$ to calculate $\Delta\phi$ (Equation 3.15)

8. Update ϕ, Γ (e.g. $\phi^{n+1} = \phi^n + \Delta\phi$)

9. Check for $\Delta\phi, \Delta\Gamma <$ tolerance

3.6 Discussion of Newton's Method

It was mentioned earlier that Newton's method for solving non-linear differential equations converges quadratically when sufficiently near the correct solution. This feature of Newton's method is especially useful in a design environment where small changes are made to an initial geometry and a new solution is computed to investigate the effects of the design change. The converged solution for the initial geometry is a very good guess for the new solution.

In practice, obtaining a solution for a highly non-linear case (strong or rapidly moving shock waves) requires some extra care during the solution process. Each Newton step represents a linear perturbation from the current solution. When the actual changes are highly nonlinear, the linear approximation tends to overshoot the solution and give a new distribution of ϕ that is non-physical (i.e. negative densities). This problem is very evident when the difference between an initial solution and the converged solution is a large change in shock position. For example, if a particular geometry is being analyzed for a series of Mach numbers, each converged solution at a given Mach number represents a good initial guess for the solution to the next Mach number in the series (assuming reasonable changes like $M = .4,\ .45,\ .5,\ \ldots$). Once a shock wave develops on the blade, however, any further increase in Mach number typically moves the shock downstream on the blade. Since each Newton step is a linear perturbation to the previous iteration, the shock can only be moved reliably about one shock width per cycle. Figure 3.12 presents an example of the problems associated with large shock movement.

For the case presented in the figure, a shock wave formed near the leading edge of

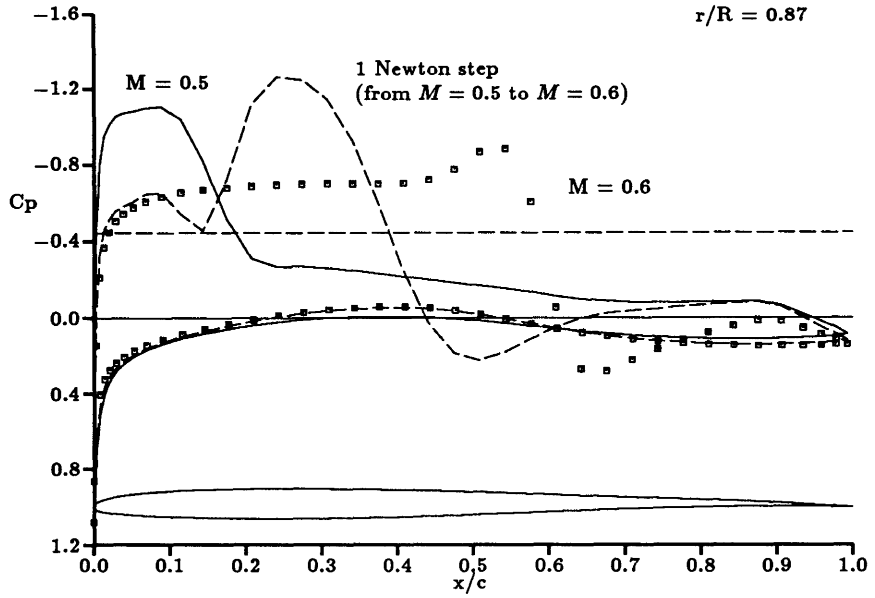


Figure 3.12: The problem with shock wave movement.

the blade at $M = 0.5$. The next desired analysis point was at $M = 0.6$, and the symbols show the actual converged solution at $M = 0.6$. The dotted line shows the solution after one Newton cycle. Convergence from this first step was impossible. A better strategy for the above example is to first smear the $M = 0.5$ solution by increasing the amount of artificial dissipation. With the shock wave effectively smoothed, moving the shock does not create the severe problems shown in Figure 3.12. Once the new shock position is established, the dissipation can be reduced to the usual level and the analysis solution for $M = 0.6$ can be 'cleaned up' to the solution shown by the symbols in the figure.

One other technique is useful for keeping especially large changes under control. At the end of each Newton cycle, the predicted solution is checked for non-physical features. An under-relaxation factor is computed based on the most non-physical update in the solution. This under-relaxation factor is then applied to all the Newton changes.

$$\begin{aligned}\phi^{n+1} &= \phi^n + \omega\Delta\phi \\ \Gamma^{n+1} &= \Gamma^n + \omega\Delta\Gamma \\ \omega &< 1\end{aligned}$$

The above technique is extremely useful for cases difficult to converge. The amount of under-relaxation can be monitored as an indicator of large changes taking place in the field. A very small ω indicates that artificial dissipation should be increased until the solution starts to settle down. Table 3.2 summarizes the Newton convergence history for the case shown in Figure 3.12. The values in the table are representative values for the convergence parameters used in the present method.

One final remark. When large changes are taking place in the flow, the matrix inversion by SLOR to find the Newton changes does not have to be completely converged. Typically 10 to 20 SLOR sweeps are applied to the inversion each Newton cycle until the solution begins to settle. Several hundred SLOR sweeps may be necessary to converge the matrix inversion completely.

Table 3.2: Summary of Newton Convergence for $M = 0.5$ to $M = 0.6$.

	ω	$\Delta\phi$	M_{crit}	μ_c	SLOR sweeps	comments
1	0.997	0.75×10^{-2}	0.90	1.75	10	shock moving
2	1.000	0.49×10^{-2}	0.90	1.75	10	.
3	1.000	0.51×10^{-2}	0.90	1.75	50	.
4	1.000	0.19×10^{-2}	0.90	1.75	50	shock stable
5	1.000	0.44×10^{-3}	0.90	1.75	150	.
6	1.000	0.42×10^{-4}	0.90	1.75	200	.
7	1.000	0.73×10^{-6}	0.90	1.75	250	.
8	1.000	0.66×10^{-2}	0.95	1.00	10	sharpening shock
9	1.000	0.16×10^{-2}	0.95	1.00	10	
10	1.000	0.47×10^{-3}	0.95	1.00	10	
11	1.000	0.58×10^{-3}	0.95	1.00	10	
12	1.000	0.93×10^{-4}	0.95	1.00	50	
13	1.000	0.93×10^{-6}	0.95	1.00	300	

3.7 Final Comments

The solution procedure requires the calculation of $\left[\frac{\partial R}{\partial \phi}\right]^{-1}\left\{\frac{\partial R}{\partial \Gamma_k}\right\}$ for $k_b - 1$ stations on the blade for each Newton cycle. At first glance these large matrix inversions may seem to be prohibitively expensive for a practical solution method. However, several features significantly reduce the time required to converge a solution. First, the Γ sensitivities ($\frac{\partial \phi}{\partial \Gamma}$) do not change much from cycle to cycle (even though the circulation on the blade may change a large amount). In practice relatively few SLOR sweeps are required to converge each Γ sensitivity for all but the first Newton step. Second, the SLOR operations to find the Γ sensitivities for all the radial stations along the blade are completely independent of each other and can therefore be computed in parallel. This research was conducted on a Stellar GS2000 computer with 4 vector processers running in parallel. Hence, 4 Γ sensitivities were computed simultaneously on this computer resulting in great time savings. Finally, it may be possible to initialize the Γ sensitivites using an analytic model. Each Γ sensitivity represents the perturbation of ϕ throughout the domain for a trailing helical vortex doublet filament of unit strength at each radial station. An analytic solution based on this model could be used to initialize $\frac{\partial \phi}{\partial \Gamma}$ and thereby reduce the significant computer time required to invert the matrix for each radial station for the first Newton cycle.

Chapter 4
Analysis Results

This chapter presents some results that demonstrate the analysis capability of the presented scheme. Three test cases are presented: 1) the transonic flow through a cascade, 2) a marine propeller in incompressible flow, and 3) the SR-3 Prop-Fan operating in transonic flow. In all cases, a converged solution is one in which the Newton updates $(\Delta\phi, \Delta\Gamma)$ are less than 10^{-6}.

4.1 Two-Dimensional Cascade Flow

The purpose of this first test case is to verify that the basic analysis algorithm of the present method is correct. The geometry is an unloaded cascade composed of NACA 0024 sections spaced 1.5 chords apart. The computational domain is sketched with the type of boundaries indicated in Figure 4.1. The domain is three-dimensional but the flow through the cascade is two-dimensional because the geometry is constant in the z-direction.

Solutions from the present method are compared to results from a two-dimensional Euler solver [4, 5] in Figures 4.2-4.4 for subcritical and supercritical flows. The agreement is very good for the subcritical case as expected since the flow is fully isentropic. For the supercritical case, Figure 4.3 shows the expected trend for a full-potential solution. A shock has formed on the cascade at about 45 percent chord. The full potential solution agrees very well with the Euler solution except in the prediction of the shock location. At the shock, errors are introduced into the momentum equation because the flow is assumed to be isentropic. The shock wave acts like a distribution of momentum sources so that no rotational flow exists downstream of the shock. In practice, the full potential approximation is acceptable for shocked flows where the normal Mach number upstream of the shock is less than 1.3. For the supercritical flow presented in Figure 4.3, the Mach number is 1.31 upstream of the shock. Mach contours (with levels from 0 to 1.0) are shown in Figure 4.4.

Converged solutions for the cascade flow required 7 to 8 Newton cycles. Each cycle was lengthy because of the fine grid ($120 \times 15 \times 3$) used to obtain solutions. The SLOR method was very slow at eliminating large wavelength errors on the fine grid so that 300 to 400 iterations were required for each order of magnitude reduction in residual error during the Jacobian matrix inversion. An acceleration scheme such as multigrid would greatly reduce the number of relaxation iterations required each Newton cycle, especially on very fine grids.

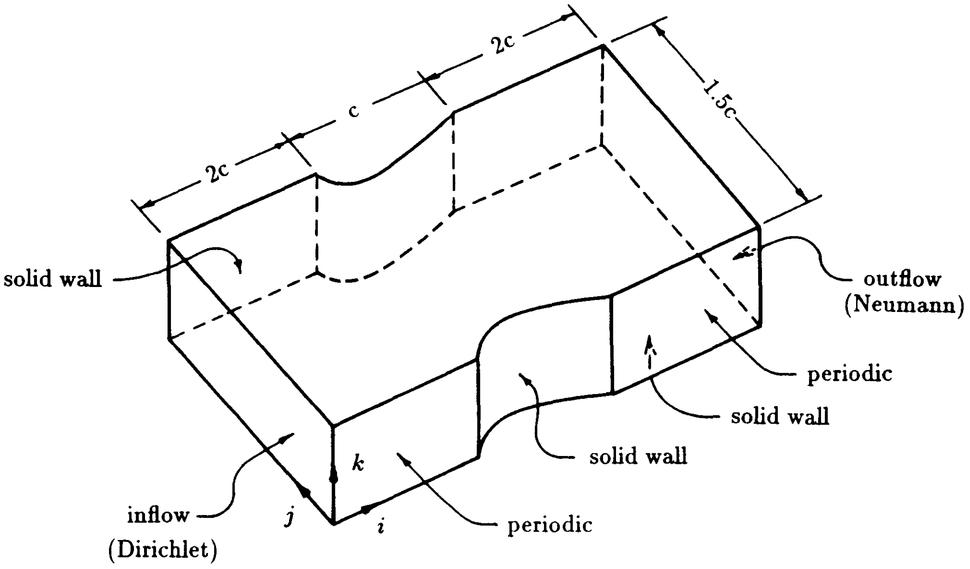


Figure 4.1: Computational domain for cascade flow.

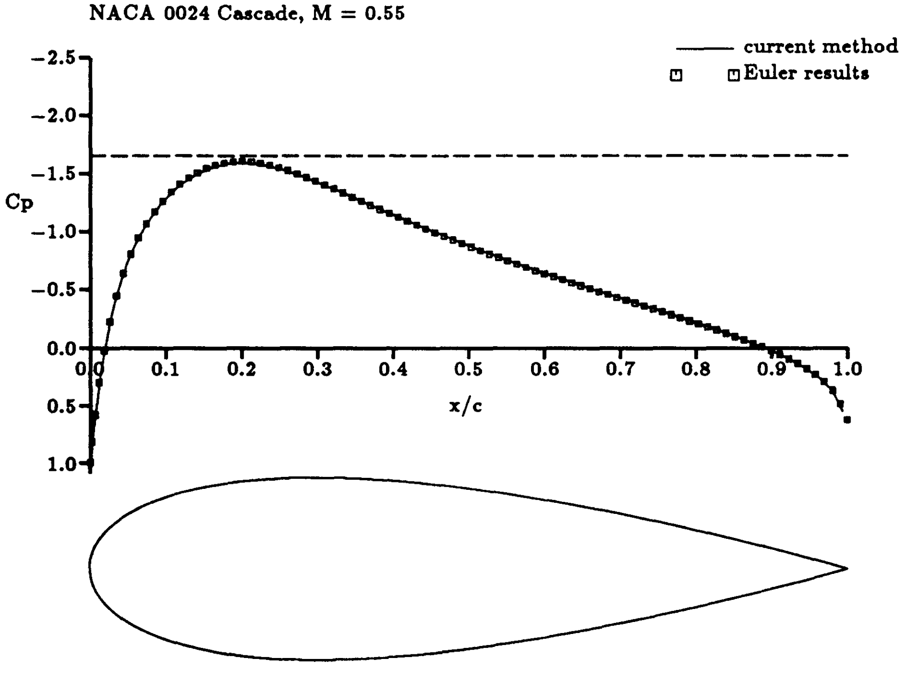


Figure 4.2: Pressure distribution for subcritical cascade.

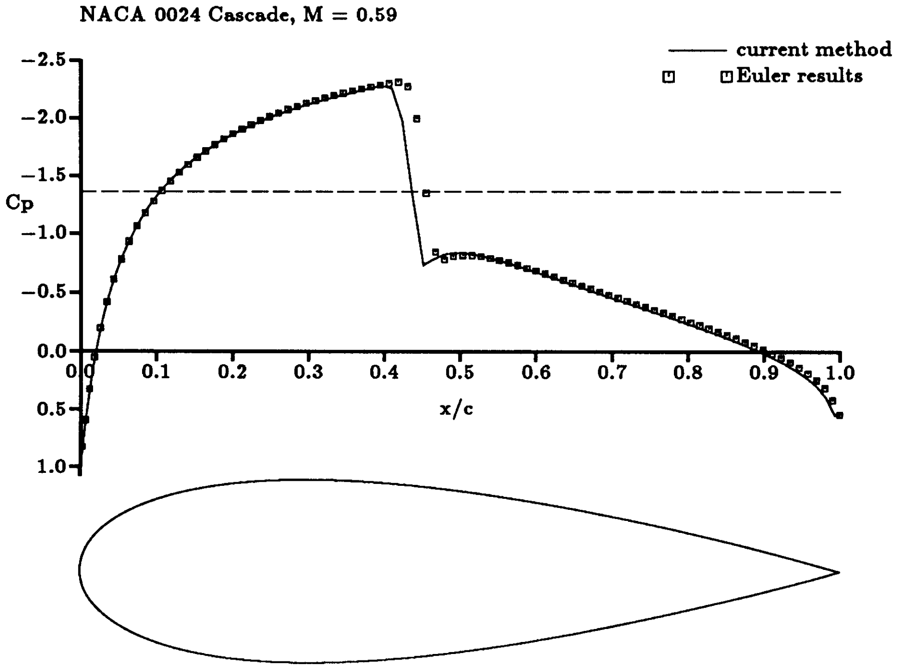


Figure 4.3: Pressure distribution for supercritical cascade.

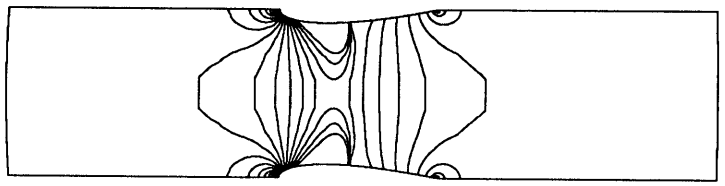


Figure 4.4: Mach contours for supercritical cascade (increments from $M = 0$ to $M = 1$).

4.2 Marine Propeller in Incompressible Flow

This second test case was chosen because of the availability of high quality numerical data for the selected propeller [8, 7]. The propeller geometry [1] is plotted in Figure 4.5. The numerical results were obtained using panel methods which, for incompressible flow, should be equivalent to the full potential analysis of the present method.

Obtaining a solution for this case presented some difficulties because of the tip geometry of the marine propeller. The propeller has a rounded tip that is pitched only 18 degrees. The computational grid contained highly skewed, high aspect ratio cells in the tip region that were unavoidable with the structured grids used in this thesis. The rapid decrease in chord seemed to present the most difficulty, so the tip was squared off to obtain the results presented in this section. The difference between the actual propeller and the computational model is plotted in Figure 4.5. The results presented here were obtained using a $60 \times 25 \times 20$ grid with 40 chordwise points and 15 radial stations on the blade. For incompressible flow, the only nonlinearity in the equations is the Kutta constraints on the trailing-edge velocities. A converged solution was obtained in 3 Newton cycles.

Pressure distributions at several radial stations are plotted in Figure 4.6. The agreement is quite good with the largest differences occuring at the leading edge of the propeller near the outer portion of the blade. There are some differences in the numerical models for the two methods that should be mentioned. First, the present method uses a constant radius hub extending to infinity upstream and downstream of the propeller. The panel method results are for a propeller with a nacelle that extends to infinity upstream but extends only 0.65 radii downstream of the propeller. Second, the panel method was developed for marine propellers and could more accurately represent the rounded tip geometry. Finally, the pressure distribution near the leading edge could be better resolved with more grid points resulting in better agreement especially for the lower surface. Overall, the agreement is entirely satisfactory.

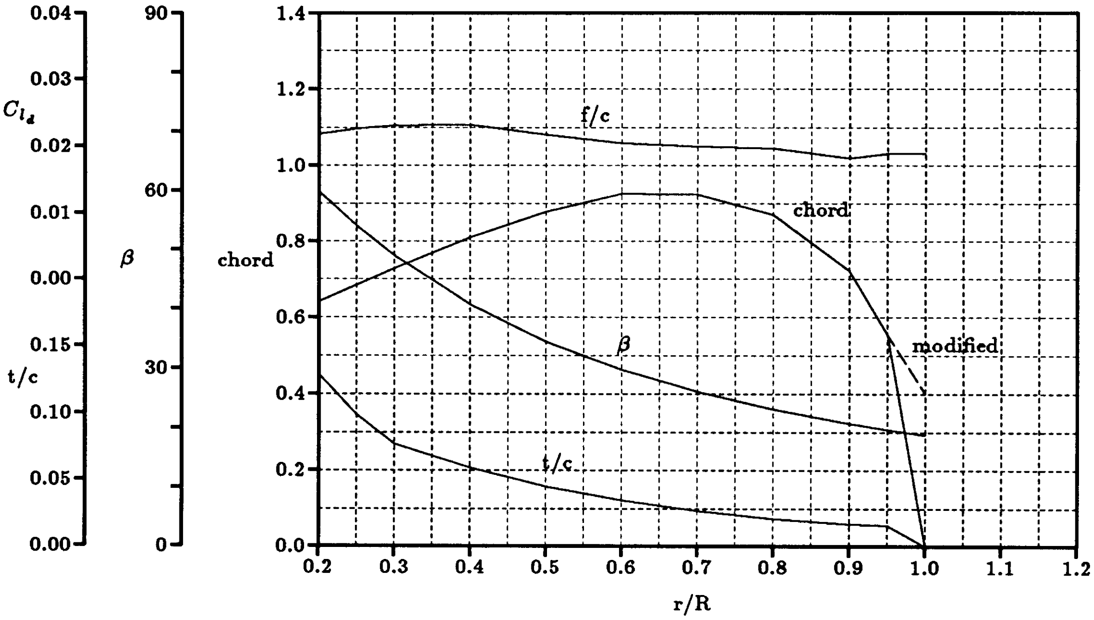


Figure 4.5: Geometry of marine propeller.

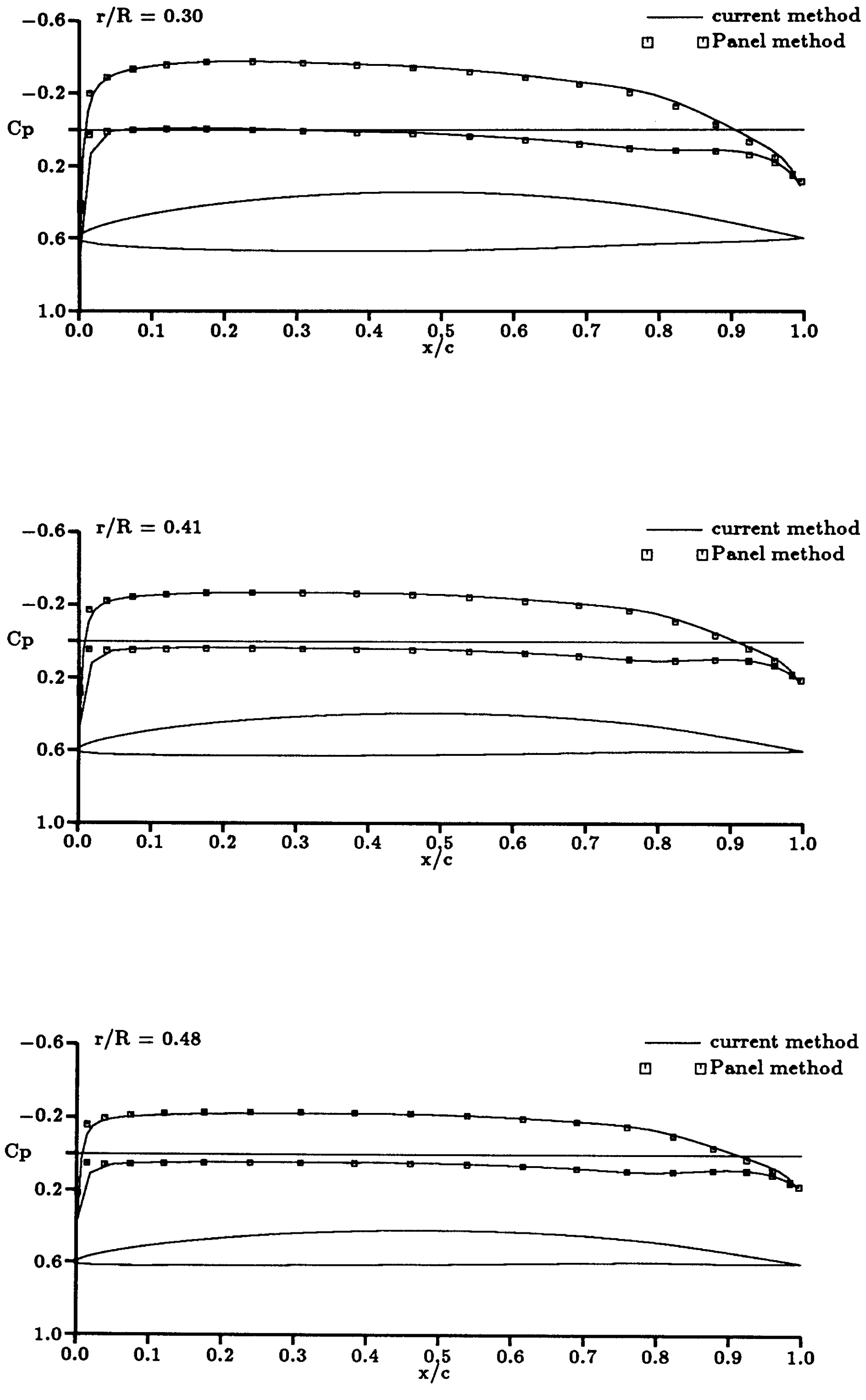


Figure 4.6: Pressure distribution for marine propeller; J=0.833.

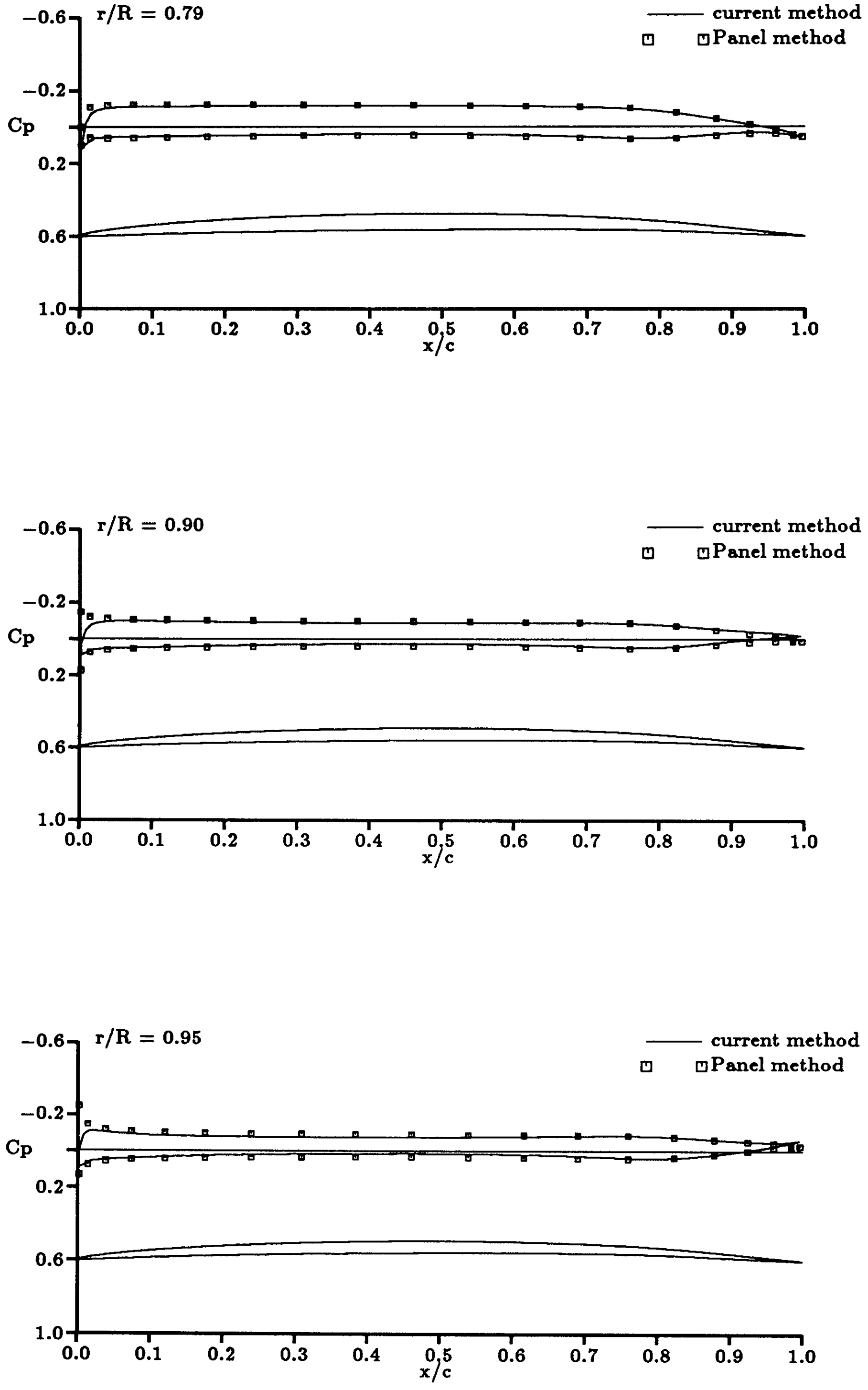


Figure 4.6: (continued).

4.3 Prop-Fan in Transonic Flow

This testcase is representative of the type of propellers for which the current method was developed. The propeller is an 8-bladed Prop-Fan designed for an operating Mach number of 0.8 at an altitude of 35,000 ft. Reference [12] contains a detailed description of the propeller geometry as well as a summary of wind tunnel data obtained during testing in 1982. The wind tunnel data does not include detailed measurements of the blade pressure distributions but does contain complete power coefficient and efficiency data for a range of Mach numbers and advance ratios.

Several data points were obtained using the current method at Mach numbers 0.45, 0.6, and 0.8. All results were obtained using a $60 \times 21 \times 20$ grid with 40 chordwise and 15 spanwise stations on the blade. The data points are plotted with the wind tunnel data in Figures 4.8, 4.9, and 4.10. The plots show power coefficient[1] ($C_P = \frac{P}{n^3 D^5}$) versus advance ratio ($J = \frac{V}{nD}$) for each of the 3 Mach numbers analyzed. Comparisons between the numerical results and the wind tunnel data must be made carefully. For $M = 0.6$ and J between 3.7 and 4.1, the numerical results show the same trend as the experimental data but are shifted as if a constant pitch increment of $\sim$ 2.5 degrees were added to each section of the blade. For advance ratios below 3.7, the trend indicated by the numerical results is quite different from the experimental data. As the advance ratio decreases below 3.7, the character of the flow around the Prop-Fan prevents the full potential equation from being a good approximation to the real flow.

The problem is illustrated in Figure 4.11. The figure compares the pressure distributions at $r/R = 0.94$ for $J = 4.0$ (symbols) and $J = 3.4$ (solid line). At the higher advance ratio, the local section is close to its design lift condition. For the lower advance ratio, a severe expansion occurs around the small leading edge radius and is terminated by a strong shock wave. The peak Mach number in this expansion is $\sim$ 1.8. The assumption of isentropic, inviscid flow results in a very poor approximation to the real flow near the leading edge. The comparison in Figure 4.11 is typical for sections in the

[1] All presented data are nondimensionalized by the reference diameter of the SR-3 to be consistent with published data. See [12, page 9] for more information.

outer 30% of the Prop-Fan blade for $M = 0.6$, $J = 3.4$.

The power coefficient is underpredicted when the leading edge flow becomes severe because the present scheme is inviscid. The real flow most likely forms a separation bubble at the leading edge and may include shock induced separation from the strong shock wave. The resulting increase in profile drag would increase the power coefficient substantially above that predicted by any inviscid method. This argument also applies to the other two cases presented.

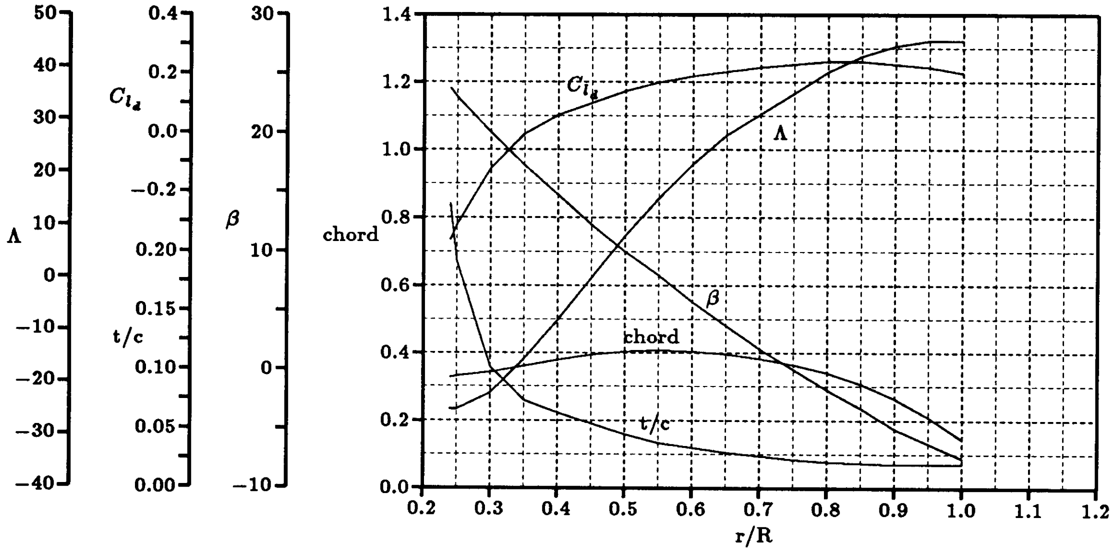


Figure 4.7: SR-3 Prop-Fan geometry.

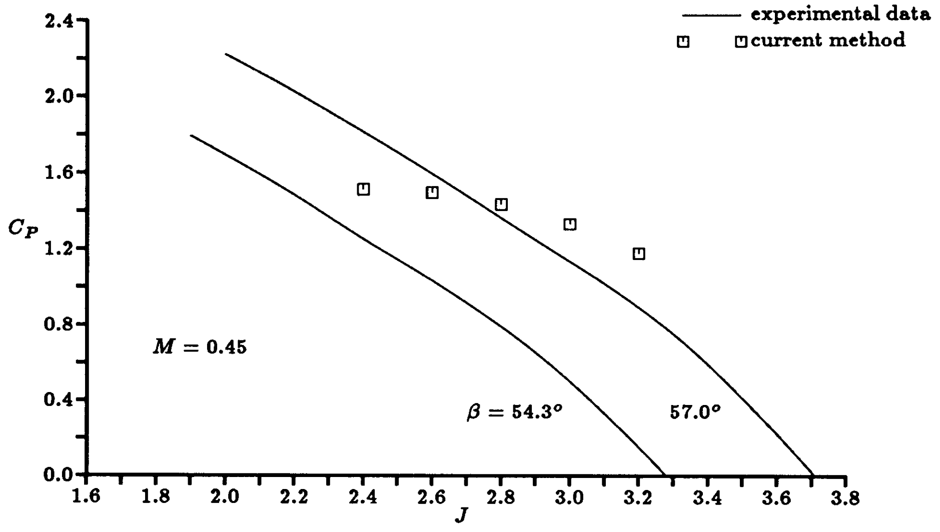


Figure 4.8: SR-3 analysis results; $M = 0.45$, $\beta_{0.75R} = 57.0°$.

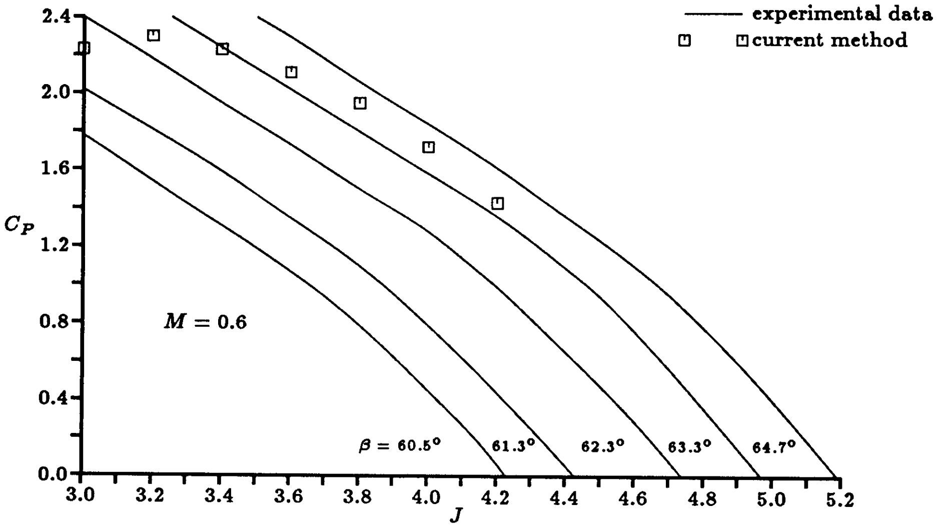


Figure 4.9: SR-3 analysis results; $M = 0.60$, $\beta_{0.75R} = 61.3°$.

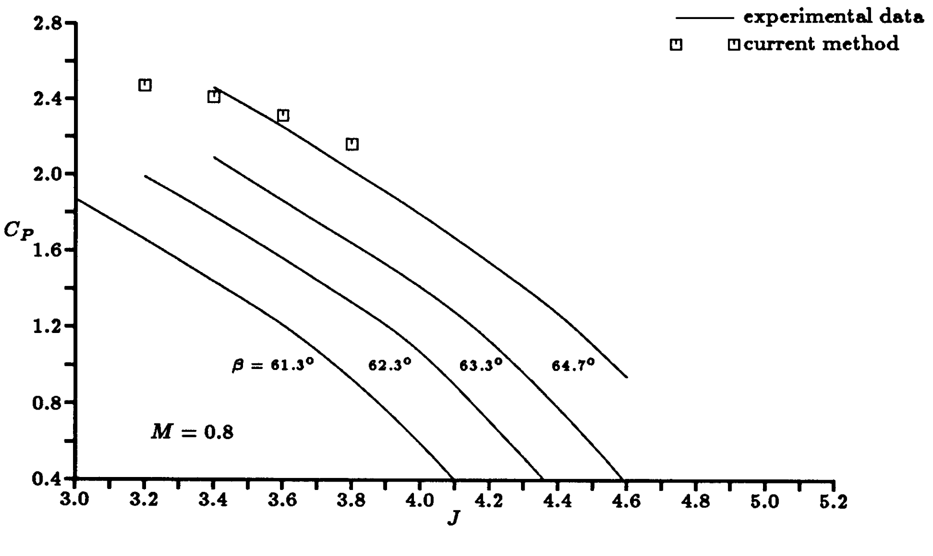


Figure 4.10: SR-3 analysis results; $M = 0.80$, $\beta_{0.75R} = 61.3^o$.

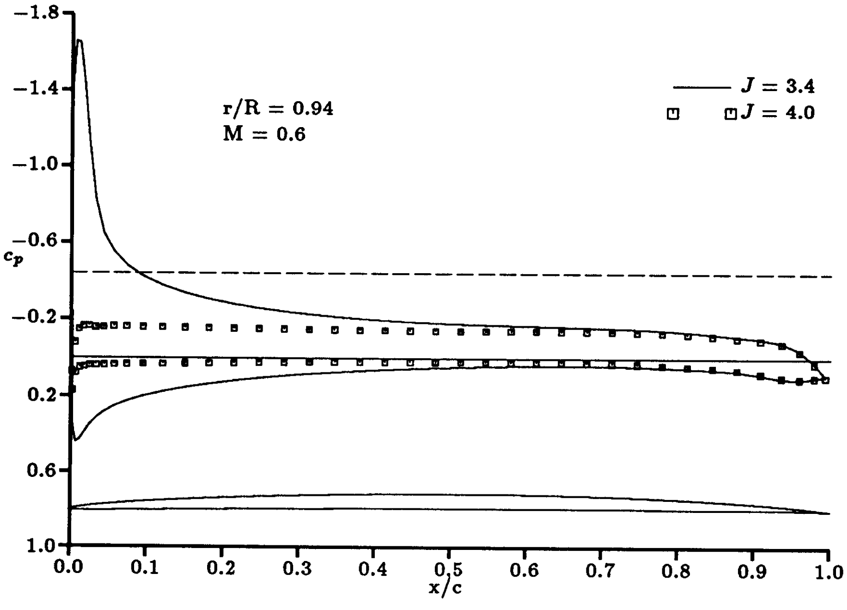


Figure 4.11: SR-3 pressure distributions for $J = 4.0$ and $J = 3.4$.

Chapter 5
Philosophy of Design Method

The goal in developing the design capability of the present method was to provide a quick way to modify a propeller blade that has been designed in a two-dimensional sense to correct for three-dimensional effects. Design methods for 2d airfoils and cascades have become very mature in recent years. In [4], Drela presents a 2d design method based on the Euler equations coupled with interactive boundary layer analysis. The method has demonstrated an impressive ability to calculate airfoil shapes from specified surface pressure distributions in viscous subsonic, transonic, or supersonic flows. An initial propeller design, then, could be composed by stacking together a set of 2d cascades that have been developed in this manner for the local flow at several radial stations along the blade. Such a design would predict the propeller loading as a similarly stacked set of 2d pressure distributions. Presumably, the designed loading would satisfy some constraints on propeller efficiency and noise characteristics.

When the design developed by 2d methods is analyzed in fully 3d flow, the actual pressure distribution on the blade will not, in general, agree with the stacked 2d solutions. The initial design, though, should be reasonably close. The scheme developed in this research handles the problem of making small adjustments to the initial design to correct for 3d effects. An effective pressure distribution obtained from the 2d methods would be specified as the target pressure distribution for the 3d design method.

The importance of the above methodology is that the 3d design capability needs to quickly make small changes to a 'near-optimum' propeller blade. Consistent with this philosophy, a quick, approximate inverse method is described in the next chapter which will drive a given seed propeller blade to satisfy a given target pressure distribution. Of course, many flows can be conceived in which a small geometry change has a dramatic effect on the global flowfield (i.e. the difference between a choked and an unchoked cascade). Strategies for these more severe cases will be considered after the design method is described.

Chapter 6

Description of Design Method

Three points are important in developing the current design method.

1. The geometry changes in a reasonably small number of smooth modes which define movements of surface points in a direction normal to the surface.
2. The normal surface movement is modeled as an effective wall transpiration.
3. The target pressure distribution is not satisfied exactly but in a 'best-fit' or least squares sense.

6.1 Perturbation in Modes

The most general geometry redesign method would allow each point on the design surface to move independently of the other points and in any of the 3 coordinate directions. This freedom would introduce 3 global unknowns $(\Delta x, \Delta y, \Delta z)$ for each point on the design surface. For a medium sized transonic flow grid containing $\sim$ 1200 surface points (80 around the blade, 15 stations radially), the number of unknowns introduced makes such generality very expensive.

When geometric design modes are used, each surface point is restricted to move normal to the surface and in set combinations with all other surface points. The concept is illustrated for a two-dimensional section in Figure 6.1. The mode shown in the figure tends to change the thickness of the section since the upper and lower surfaces move in opposite directions.

When designing with modes, the number of global unknowns introduced into the

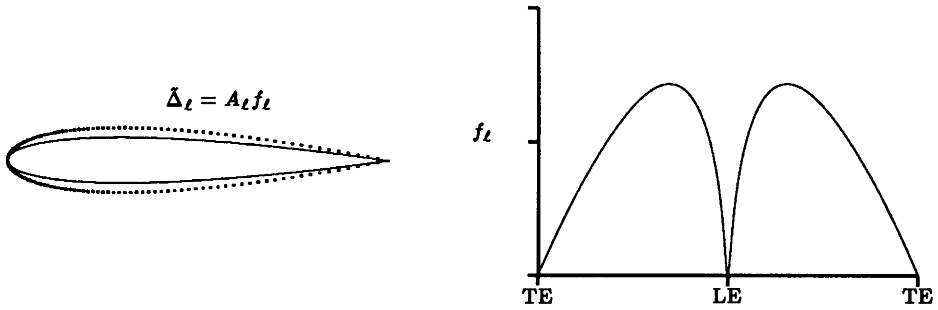


Figure 6.1: Illustration of the concept of design modes.

basic system is reduced to one coefficient (A_ℓ) that multiplies each geometric mode. For a given set of modes and coefficients, the normal displacement of a point on the design surface is simply

$$\tilde{\Delta} = \sum_{\ell=1}^{L} A_\ell f_\ell$$

The coordinates of the new position of the grid point can be expressed as

$$(x, y, z)_{\text{new}} = (x, y, z)_{\text{seed}} + \tilde{\Delta}\hat{n} \tag{6.1}$$

where $\hat{n}$ is the unit outward normal for the design surface.

The geometric mode concept has other advantages over unrestricted point movement in addition to great reduction of global unknowns. The perturbed geometry is guaranteed to remain smooth if smooth design modes are used. Also, a design region is easily specified by simply using modes that are identically zero outside the design region. Finally, design constraints can be imposed by using modes that themselves satisfy the design constraints (i.e. fixed leading-edge, fixed mean line).

The geometric mode concept may seem somewhat restrictive. However, a general two-dimensional airfoil can be satisfactorily described by ~ 15 Fourier modes except in the most extreme cases. In three dimensions the appropriate number of geometric modes is higher since radially varying modes must be included as well as the chordwise

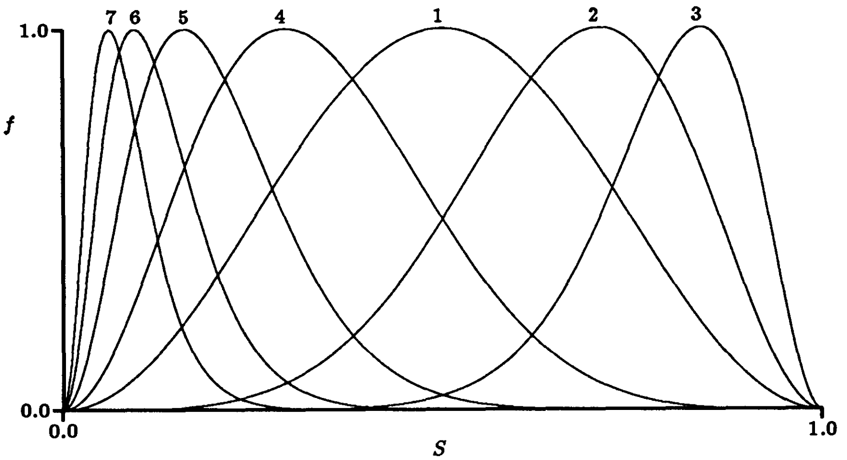


Figure 6.2: Basic set of geometric modes.

modes. However, the generality with which spanwise variations need to be included is not clear. Perhaps only 4 or 5 spanwise modes may be sufficient. It is clear, though, that the choice of geometric modes will affect how well a target pressure distribution can be satisfied.

Figure 6.2 is a plot of the 7 basic mode shapes used in the present method. Each three-dimensional geometric mode is a combination of any two of the basic shapes (1 spanwise and 1 chordwise). An additional shape that is not plotted in Figure 6.2 is a spanwise mode that is constant along the span except very near the tip where it varies rapidly to zero.

6.2 Modeling the Displacement Effect

In order for a design method to properly redesign a blade shape to satisfy a given pressure distribution, the method must have information on how the solution will change when the geometry is perturbed. For non-linear problems, these sensitivities cannot be obtained analytically. The sensitivities can be obtained by actually perturbing the geometry by a small but finite amount, and calculating new solutions for each of the perturbed shapes, although this technique is very expensive in three-dimensional flows. In the present method, calculation of the sensitivities is greatly simplified by approximating the displacement of the design surface with wall transpiration. This approach is often used in interacting boundary layer methods where the displacement effect of the boundary layer is modeled by a distribution of wall blowing. The amount of wall blowing (V_w) for a given normal displacement ($\tilde{\Delta}$) of a solid surface can be expressed as

$$\rho V_w = \nabla \cdot (\rho \vec{W} \tilde{\Delta}) \tag{6.2}$$

Equation 6.2 can be conveniently reduced as follows:

$$\begin{aligned} \rho V_w &= \left[\nabla \cdot (\rho \vec{W})\right] \tilde{\Delta} + (\rho \vec{W}) \cdot \nabla \tilde{\Delta} \\ &= \rho \, \vec{W} \cdot (\nabla \tilde{\Delta}) \end{aligned}$$

By combining the above wall transpiration with the current inviscid stream, a new flow is obtained that, to first order in $\tilde{\Delta}$, is equivalent to the flow around the displaced surface. The influence of the wall blowing on the basic system of equations is applied as a natural boundary condition via the surface integral term in Equation 3.5. Recall that the residual for a node on a domain boundary included the following surface integrals:

$$R_\ell = \sum_{e=1}^{\leq 4} \left\{ \cdots \; - \tilde{\rho} \oint_{S_e} N_\ell^e \; \vec{W} \cdot \hat{n} \; dS \right\} \tag{6.3}$$

For an analysis solution, $(\rho \vec{W}) \cdot \hat{n} = 0$ on a solid surface to enforce flow tangency. However, in the design method, $(\rho \vec{W}) \cdot \hat{n} = -\rho V_w$ whenever there is wall blowing to model a geometry perturbation (the minus sign results from $\hat{n}$ being defined outward for the element). Thus, the residual for points on design surfaces depends on the design mode coefficients through $\tilde{\Delta}$. Note that the geometry distortion modes define motion of the design surface normal to the surface, which is precisely the displacement needed to calculate the wall blowing distribution. Equation 6.3 is simplified by assuming that the wall blowing is constant over any given element face on the design surface and equal to the value at the face centroid. The surface integrals in Equation 6.3 are taken over all 6 faces that form each element. The only faces that actually contribute to R_ℓ are those faces on the design surface for which $\vec{W} \cdot \hat{n}$ is non-zero. (The contribution from the other faces of the elements that share node ℓ is zero because of internal surface cancellations or because the weight function N_ℓ is zero on supercell surfaces.) For node d on a design surface, Equation 6.3 simplifies to

$$R_d = \cdots \sum_{ef=1}^{\leq 4} (\rho V_w)^{ef} \int_{S_{ef}} N_d \ dS \tag{6.4}$$

where ef indexes over the element faces that share node d.

6.3 Finding the Best Fit Pressure Distribution

The goal of any inverse method is to find a geometry that will result in a given target pressure distribution. This problem is ill-posed, however, when the target pressure distribution is non-physical or otherwise mathematically impossible. Lighthill [9] derived three explicit integral constraints for the surface pressure distribution in two-dimensional, incompressible flow. These contraints enforce that the resulting shape is closed (no trailing edge thickness, no trailing edge streamwise mismatch), and that the surface pressures are consistent with the freestream pressure. Although similar constraints must exist for three-dimensional compressible flow, they are not likely to be expressible in closed form. An inverse method, then, must deal with the possibility that a target pressure distribution is not mathematically possible.

Since the surface movement in the present method is not completely arbitrary but is restricted to move based on a finite set of design modes, there are limits on which target loadings can be satisfied by perturbing the geometry. The target surface pressures are satisfied in a least squares sense, and since the perturbed geometry is always physical, the resulting surface pressures are physical. Specifically, the present design method perturbs the current geometry via the geometric design modes so as to minimize

$$I = \oint_{S_{blade}} \frac{1}{2}\left(C_p - C_{p_{targ}}\right)^2 dS \tag{6.5}$$

where C_p is the current pressure and $C_{p_{targ}}$ is the specified target pressure. This approach completely sidesteps the issue of Lighthill's well-posedness constraints on the target pressure distribution.

6.4 Details of Design Procedure

The design process begins with a converged solution for a given propeller blade (referred to as the seed geometry). For each geometry mode, the movement of the surface is completely specified by setting the mode coefficient A_ℓ. The first step, then, is to obtain the solution sensitivity $(\frac{\partial \phi}{\partial A_\ell})$ to movement of the design surface in each of the geometric modes. A modification of the Newton cycle described in Section 3.5.2 allows $\frac{\partial \phi}{\partial A_\ell}$ to be computed from $\frac{\partial R}{\partial A_\ell}$.

Equation 3.14 is modified to include the dependence of R on A_ℓ.

$$\left[\quad \frac{\partial R}{\partial \phi} \quad\right]\left\{\Delta\phi\right\} = \sum_{k=1}^{k_b-1} \Delta\Gamma_k \left\{-\frac{\partial R}{\partial \Gamma_k}\right\} + \sum_{\ell=1}^{L} \Delta A_\ell \left\{-\frac{\partial R}{\partial A_\ell}\right\} \tag{6.6}$$

Note that the residual $(-R)$ is zero since the analysis solution is converged. When the matrix $\left[\frac{\partial R}{\partial \phi}\right]$ is inverted and multiplied through Equation 6.6, the resulting expression contains the desired sensitivities $\frac{\partial \phi}{\partial A_\ell}$.

$$\left\{\Delta\phi\right\} = \sum_{k=1}^{k_b-1} \Delta\Gamma_k \left\{\frac{\partial \phi}{\partial \Gamma_k}\right\} + \sum_{\ell=1}^{L} \Delta A_\ell \left\{\frac{\partial \phi}{\partial A_\ell}\right\} \tag{6.7}$$

Computing $\frac{\partial R}{\partial A_\ell}$ involves a straightforward differentiation of the discrete equation residuals. The details are placed in Appendix A.

Equation 6.7 is a bit deceiving. The change in the solution ($\Delta\phi$) for a given change in the mode coefficient (ΔA_ℓ) cannot be computed with a simple application of Equation 6.7. Changes in Γ are coupled to changes in A, since the Kutta conditions (Equation 3.16) must hold after any change in A:

$$\begin{bmatrix} & \frac{\partial Q}{\partial \phi} & \end{bmatrix} \begin{Bmatrix} \\ \Delta\phi \\ \\ \end{Bmatrix} = \begin{Bmatrix} -Q \end{Bmatrix}$$

When Equation 6.7 is substituted into the above Kutta conditions, the result is a linear system of equations relating $\Delta\Gamma$ to ΔA.

$$\begin{bmatrix} & S_{ij} & \end{bmatrix} \begin{Bmatrix} \Delta\Gamma \end{Bmatrix} = -\sum_{\ell=1}^{L} \begin{bmatrix} & \frac{\partial Q}{\partial \phi} & \end{bmatrix} \begin{Bmatrix} \\ \frac{\partial \phi}{\partial A_\ell} \\ \\ \end{Bmatrix} \Delta A_\ell \tag{6.8}$$

where S_{ij} is defined as before and Q is zero because the starting point is a converged solution. For a given change in ΔA, Equation 6.8 can be solved to get $\Delta\Gamma$, and then $\Delta\Gamma$ and ΔA are substituted into Equation 6.7 to update the solution.

Once the relation between the solution and the mode coefficients is known, the next step is to formulate a set of equations to constrain the mode coefficients. Recall that the goal is to minimize the function defined as

$$I = \oint_{S_{blade}} \frac{1}{2}(C_p - C_{p_{targ}})^2 dS$$

The set of equations that constrain A_ℓ is simply

$$R_\ell = \frac{\partial I}{\partial A_\ell} = \oint_{S_{blade}} (C_p - C_{p_{targ}}) \frac{\partial C_p}{\partial A_\ell} dS = 0 \tag{6.9}$$

$\frac{\partial C_p}{\partial A_\ell}$ is easily computed from known sensitivities by applying the chain rule.

$$\frac{\partial C_p}{\partial A_\ell} = \frac{\partial C_p}{\partial \phi}\frac{\partial \phi}{\partial A_\ell}$$

An expression for $\frac{\partial C_p}{\partial \phi}$ is given in Appendix A.

The non linear system represented by Equation 6.9 is solved iteratively using the Newton-Raphson method. The non-linear residual R_ℓ is a function of the unknown mode coefficients A_ℓ. Each design Newton cycle, R_ℓ is linearized around the current levels of A_ℓ to obtain a system of linear equations for the Newton update ΔA_ℓ. In the present method, the quantity $\frac{\partial C_p}{\partial A_\ell}$ is assumed constant when the Newton system is being updated. This approximation does not significantly slow the convergence of the iteration scheme. The resulting linear system of equations is

$$\left[\quad \frac{\partial\left(\frac{\partial I}{\partial A_\ell}\right)}{\partial A_m} \quad \right] \left\{ \Delta A_m \right\} = \left\{ - R_\ell \right\} \tag{6.10}$$

$$\frac{\partial\left(\frac{\partial I}{\partial A_\ell}\right)}{\partial A_m} = \oint_{S_{blade}} \frac{\partial C_p}{\partial A_m}\frac{\partial C_p}{\partial A_\ell} dS$$

The design process is summarized below:

1. For each design mode f_ℓ, construct the sensitivities $\frac{\partial \phi}{\partial A_\ell}$
2. Read in a target pressure distribution $C_{p_{targ}}$
3. Use Newton-Raphson method to compute updates to the geometry mode coefficients to minimize $\oint_{S_{blade}} \frac{1}{2}(C_p - C_{p_{targ}})^2 \, dS$
4. Compute $\Delta\Gamma$ from ΔA
5. Use $\Delta\Gamma$ and ΔA to update solution
6. Check for convergence ($\Delta A <$ tolerance)

6.5 Discussion

The expensive step in the design process is the calculation of $\frac{\partial \phi}{\partial A}$. This step, however, is only required once for each initial condition. Once the mode coefficient sensitivities are computed, any number of target pressure distributions can be specified with little additional cost. The iterative design cycle that drives the current solution to satisfy the target loading does not require regridding the perturbed geometry or computing a full non-linear analysis solution at each iteration. Only after the design Newton cycle is converged should a new geometry be computed and run through a complete analysis solution to verify predicted results. For small geometry perturbations (consistent with the design philosophy discussed in Chapter 5), the approximations made within the design cycle should be reasonable.

A design strategy can be developed in which 3 design levels must be satisfied. This chapter discussed the details of design level 1 (DL1) in which the solution is changed to satisfy a target blade loading without actually moving the design surface or recomputing full analysis solutions. This 'linearly perturbed' solution is one Newton step towards the converged new geometry. In DL2, the mode coefficients output from DL1 are used to compute the effective wall transpiration along the design surface and a direct solution is computed with the new boundary condition. Note that even though DL2 requires a full analysis solution, the initial guess for ϕ and Γ given by DL1 should be extremely close to the converged solution. When the analysis solution is reconverged for the wall blowing boundary condtion, DL1 is repeated to get the next Newton step towards the new geometry.

The final design level (DL3) involves actually computing a new grid for the modified blade shape. DL3 verifies the approximation made when assuming geometry perturbations can be modeled as a wall transpiration. Again, a good initial guess to the converged solution is given by the solution from the inner design levels, so convergence should be quick.

DL1 satisfies the goals set when developing the present inverse method. DL1 provides an extremely quick (after computing $\frac{\partial \phi}{\partial A}$) way to make small changes to an initial geometry to satisfy a given blade loading. The next chapter presents some examples of how the developed design method works in practice. In the examples given in this thesis, DL2 was not implemented. The mode coefficients output from DL1 were used to compute a new geometry for DL3 in which the predicted modified solution was checked with a full direct calculation for the new shape.

Chapter 7
Design Results

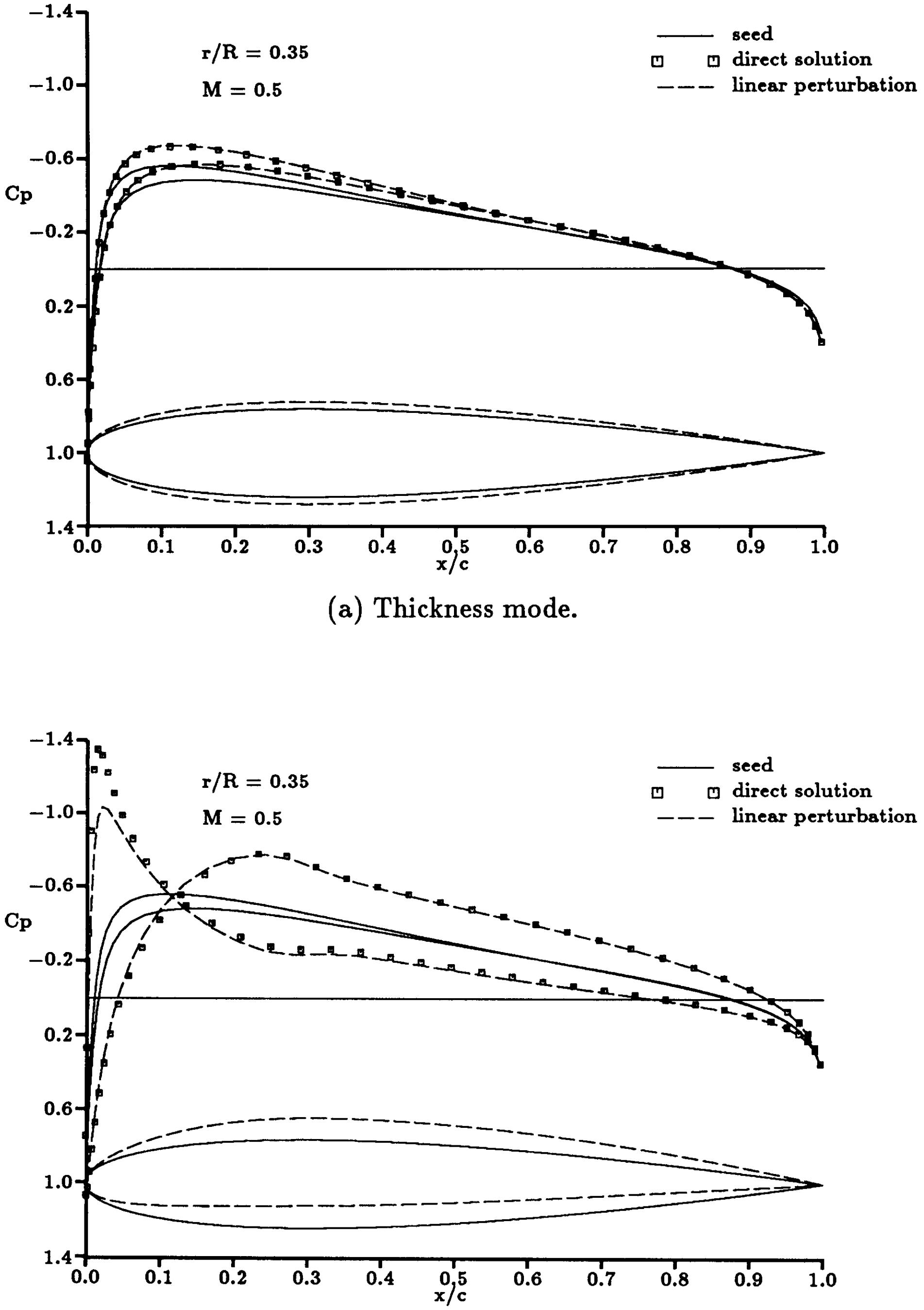
The purpose of this chapter is to present some demonstrations showing the design capability of the present scheme. When designing for transonic flow conditions, some strategy is necessary in order for the method to be useful. These issues are discussed below.

7.1 Subcritical Mode Coefficient Sensitivities

The purpose of this demonstration is to test the quality of the computed solution sensitivites ($\frac{\partial \phi}{\partial A}$) in subcritical flow. The initial geometry is an 8-bladed 'invisible' propeller that is twisted so that when the advance ratio is 1, each local section is aligned to the flow. The blades are constant chord with an aspect ratio of 10. The blade cross section is the NACA 0012 airfoil. The simplified geometry is designed to help illustrate the concepts presented in the previous chapter.

Two distortion modes are defined: 1) a thickness mode that increases the section thickness ratio by 2 percent (resulting in a NACA 0014 thickness distribution), and 2) a camber mode defined such that the distorted shape represents a NACA 3312 airfoil.

An initial solution is computed for the basic propeller at Mach 0.5 and then the design modes are input and the solution sensitivities are calculated for each mode. Figure 7.1 shows comparisons between the predicted change in the solution (based on $\frac{\partial \phi}{\partial A}$ with A = 1.0) and the actual fully computed solution for the distorted shapes. The comparison is quite good. The results from this test verify that the wall blowing approximation is a valid method for representing geometry changes. Note that the camber mode represents a very large distortion of the initial geometry and yet the effect on the solution is predicted remarkably well.



(a) Thickness mode.

(b) Camber mode.

Figure 7.1: Comparison of linearly perturbed solution and exact solution for two geometry perturbations.

7.2 The Trouble with Shock Waves

Figure 7.2 shows a potential problem when using the present design method for shocked transonic flows. The initial geometry is a non-rotating, 8-bladed propeller with constant chord and constant NACA 0012 section. The propeller blade is twisted 4 degrees so that lift develops on the blade. A converged solution is obtained for the initial blade at Mach = 0.7. The pressure distribution shows a shock wave at about 25 percent chord. The thickness mode described in the previous example is input and a solution sensitivity is computed. Again, the mode coefficient is set explicitly to 1.0 to investigate the predicted changes to the flow. Also plotted in the figure (symbols) is the converged solution for the thicker blade. Here, the predicted solution is quite poor.

The problem is that the correct effect of the geometry distortion is an aft movement of the shock wave. The issue of moving shock waves with a Newton iteration scheme was discussed in Section 3.4. The points made in that section apply in the inverse method as well. An effective design strategy is also very similar to the solution strategy for cases where highly non-linear changes are taking place in the flow. A converged solution for the initial geometry is obtained using a high level of dissipation. With the shock wave substantially smoothed, the design method works as intended. Figure 7.3 repeats the excercise of Figure 7.2 with this strategy implemented and gives much better results. The shock wave can always be sharpened when a final solution is required.

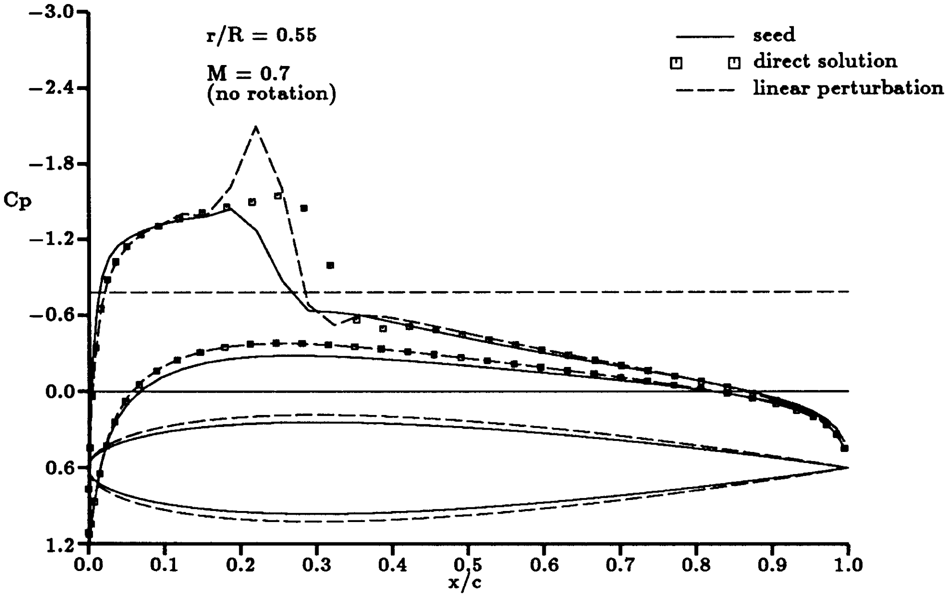


Figure 7.2: Problem with transonic redesign.

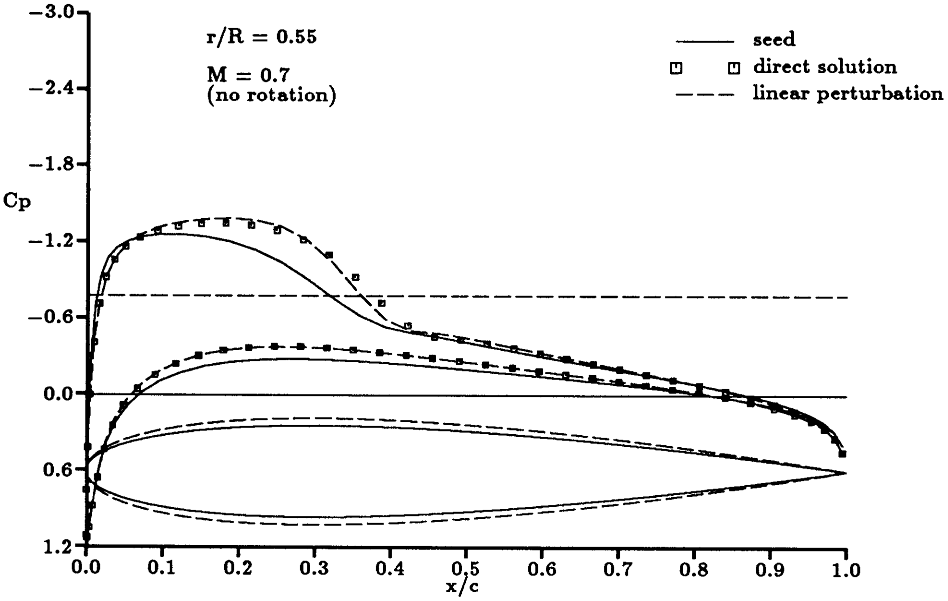


Figure 7.3: Transonic redesign with extra dissipation.

7.3 A Simple Example in Redesign

Figure 7.5 presents analysis results for a sample propeller designed for high altitude transonic operation. A lifting line approximation provided the initial analysis to choose a planform and twist distribution to give a constant spanwise loading so as to minimize induced losses. Analysis results were computed for $M = 0.6$ and $\lambda = 0.7$. Under these operating conditions, the relative oncoming Mach number for the tip of the propeller is 1.1. In Figure 7.5, the solid line represents the analysis results from the present method for the initial propeller geometry. The dashed line represents results for the same planform and twist distribution but with the blade swept to help relieve compressiblity losses. In addition, a two-dimensional Euler solution [4, 5] is plotted for $r/R = 0.77$ for comparison to the three-dimensional results. In this particular example, the effects of the three-dimensional relief are dramatic. For the two-dimensional solution, the shock on the upper surface is at the trailing edge of the section and the lower surface is supersonic from about .25c to .6c where the supersonic flow is terminated by a shock wave. In the three-dimensional solution the upper surface shock is at the midchord and the lower surface is entirely subsonic. Spanwise dilation of the streamtubes provides the three-dimenional relief while lower inboard Mach number (local relative Mach number varies with radial position) also contributes significantly to the differences in the flow. The Euler solution was obtained at the same local relative Mach number (= 0.89) and lift coefficient (= 0.49) as the three-dimensional solution.

For the inner sections of the unswept blade (solid line), the flow compresses isentropically from a mildly supersonic suction peak. At $r/R = 0.61$, the peak Mach number at the leading edge is 1.33 and a shock wave appears to be forming at about 30% chord. The flow at $r/R = 0.66$ confirms the existence of a shock wave on the upper surface of the blade. The shock wave spans the outer 45% of the propeller.

Figure 7.4 compares the planform and leading edge sweep of the unswept and swept propellers. Note that the inner sections are swept forward. For structural reasons the forward sweep is desirable so that centrifugal forces do not cause unacceptably high stresses at the propeller hub. The sweep has a significant effect on the pressure distribution that cannot be predicted with a two-dimensional method. Again referring to Figure 7.5, Near the propeller hub, the leading edge suction peak is slightly higher for the swept blade (peak Mach number = 1.24). This increase is the result of the influence of the trailing helical vortex sheet. Because the propeller sweeps forward at the root, the trailing vortex sheet and the image vortex sheet (inside the hub) interact to create an upwash at the root section. Away from the hub section, the increased sweep provides more three-dimensional relief for the local sections so that the suction peak is reduced. At $r/R = 0.61$, the swept blade is still shock free. The increased blade sweep delays the formation of the shock spanwise to about $r/R = 0.72$. For the outer sections of the blade, even though the shock is displaced forward, the local lift is not significantly reduced.

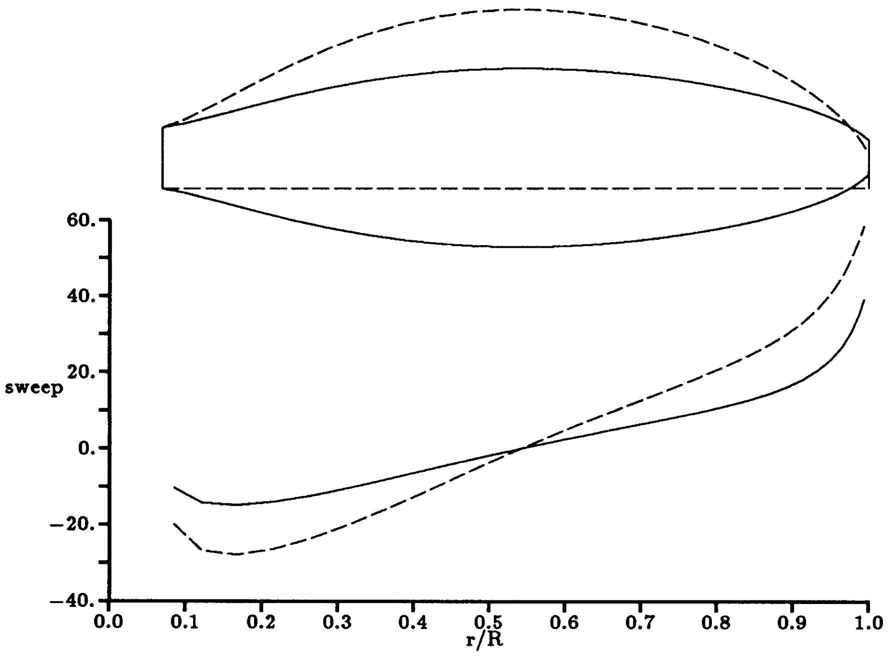
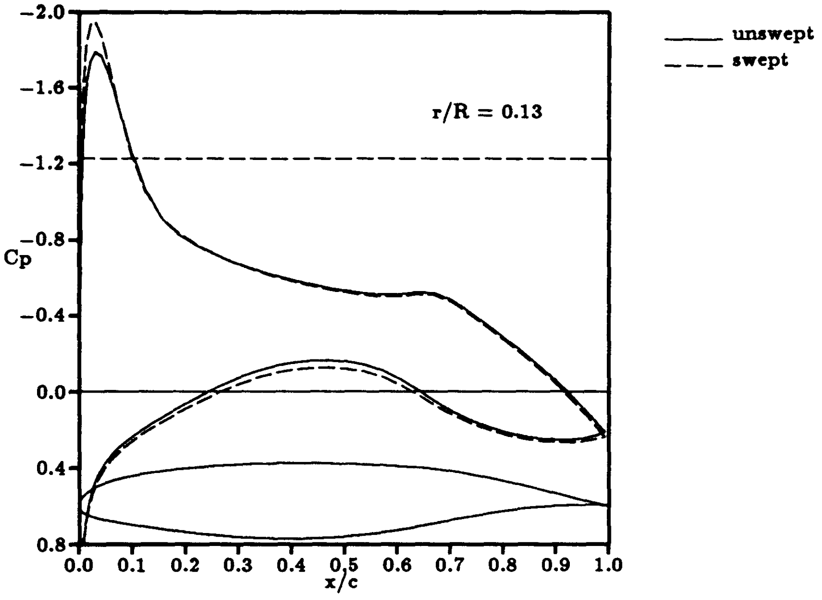


Figure 7.4: Comparison of swept and unswept planforms.



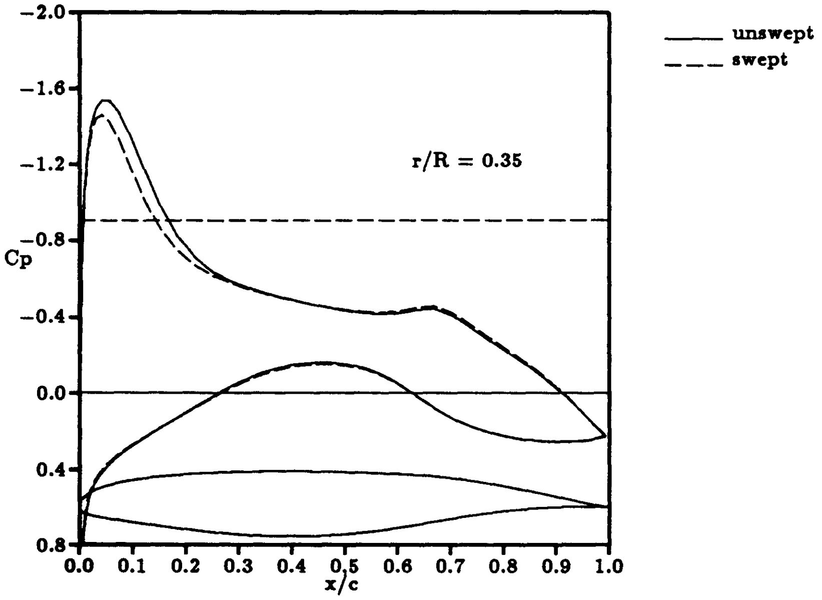


Figure 7.5: High altitude propeller analysis results, unswept vs swept; $M = 0.6$, $\lambda = 0.7$.

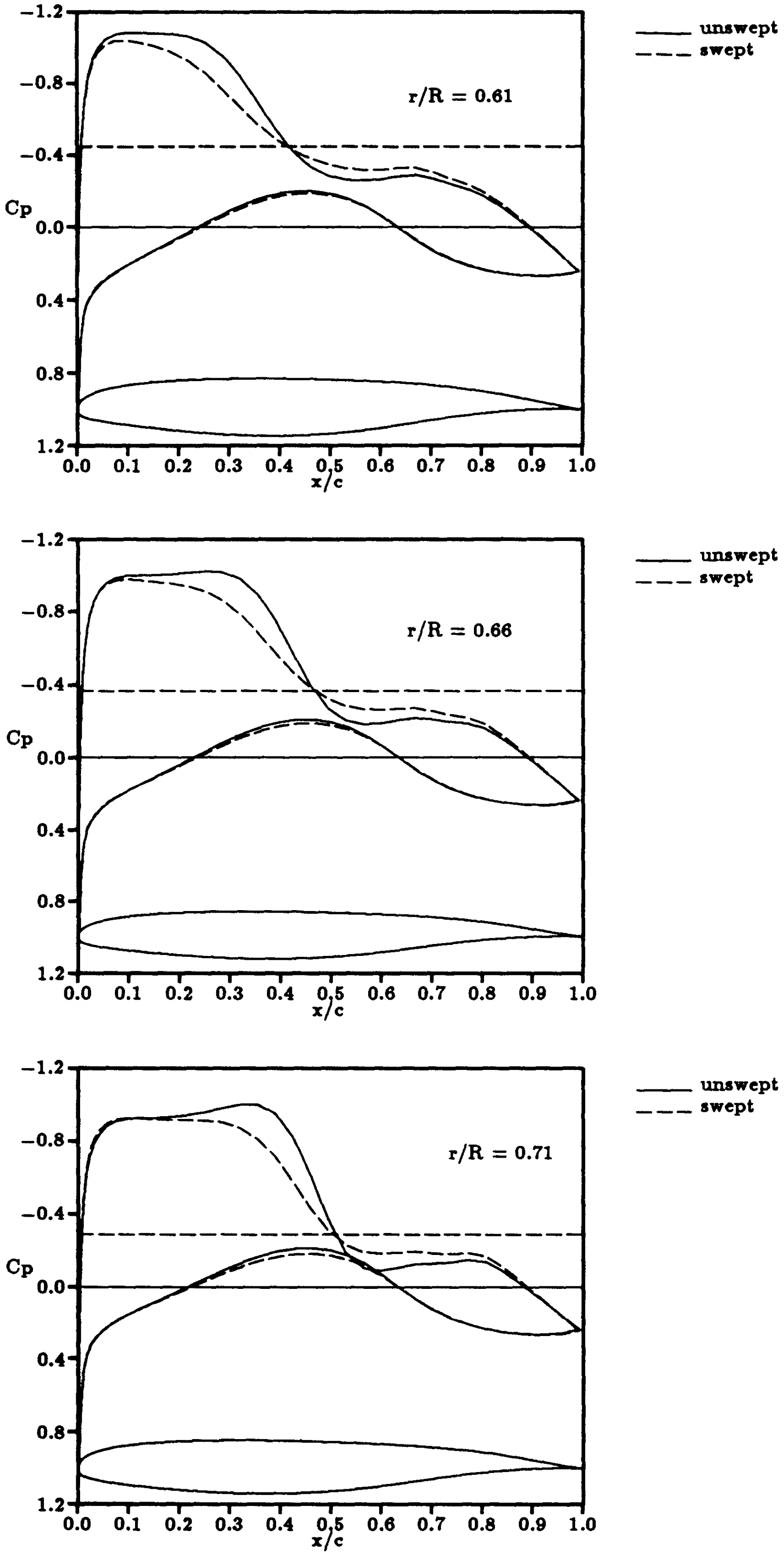


Figure 7.5: (continued).

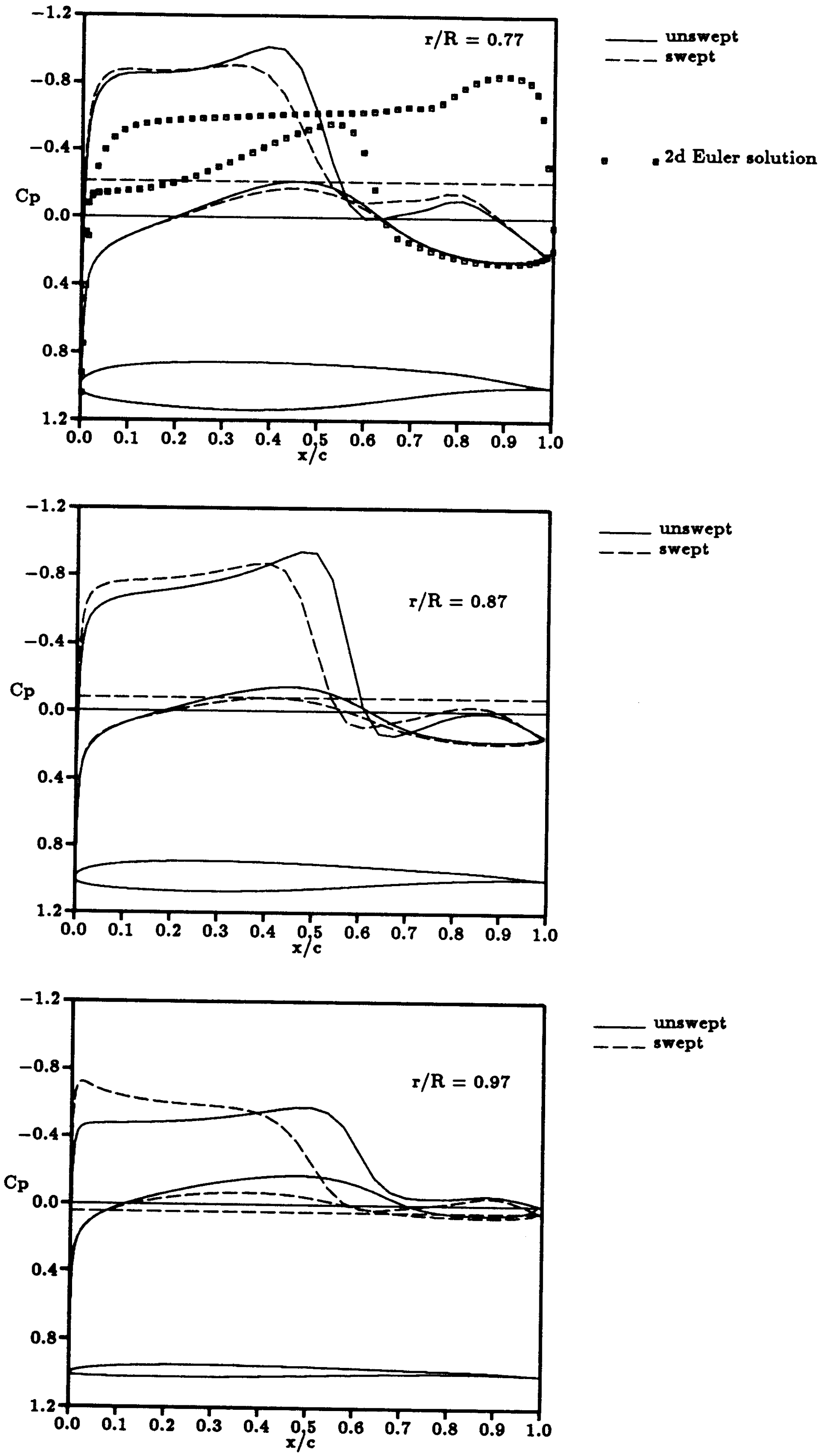


Figure 7.5: (continued).

The swept propeller blade was used to test the design method described in the previous section. Three-dimensional geometric modes were created from the basic mode shapes presented in Figure 6.2. The design region was restricted to the upper surface and 7 chordwise modes were combined with 4 spanwise modes to make a set of 28 total mode shapes.

Figure 7.6 presents the results from a sample redesign. In the spanwise region where the shock begins to form (between 60 and 75% span), the target pressure distribution was specified to extend the sonic region and very gradually compress to subcritical flow. For the outer sections, the target was not to eliminate the shock wave but to move it aft so as to increase the local section lift coefficient. The agreement between the design and target pressure distributions is quite good. The section mid-chord thickness was increased between 60 and 70% span to smooth out the recompression of flow. Note also that the suction peak was lowered slightly in this region. This result suggests that lowering the suction peak would help in smoothing out the recompression to subcritical flow. Of course this approach will only help in the spanwise regions where the oncoming relative Mach number is low enough to permit an isentropic recompression. Outboard of 70% span, the increase in thickness tends to move aft as the shock wave moves aft.

Calculating $\frac{\partial \phi}{\partial A_\ell}$ for the 28 design modes required slightly less CPU time than the direct analysis solution for the unswept blade (the calculation for the swept blade was initialized with the solution for the unswept blade and therefore required little additional time to reconverge). Once the solution sensitivities were calculated and the target pressures specified, the linearly perturbed solution was obtained in about 1/10 the time of a direct calculation.

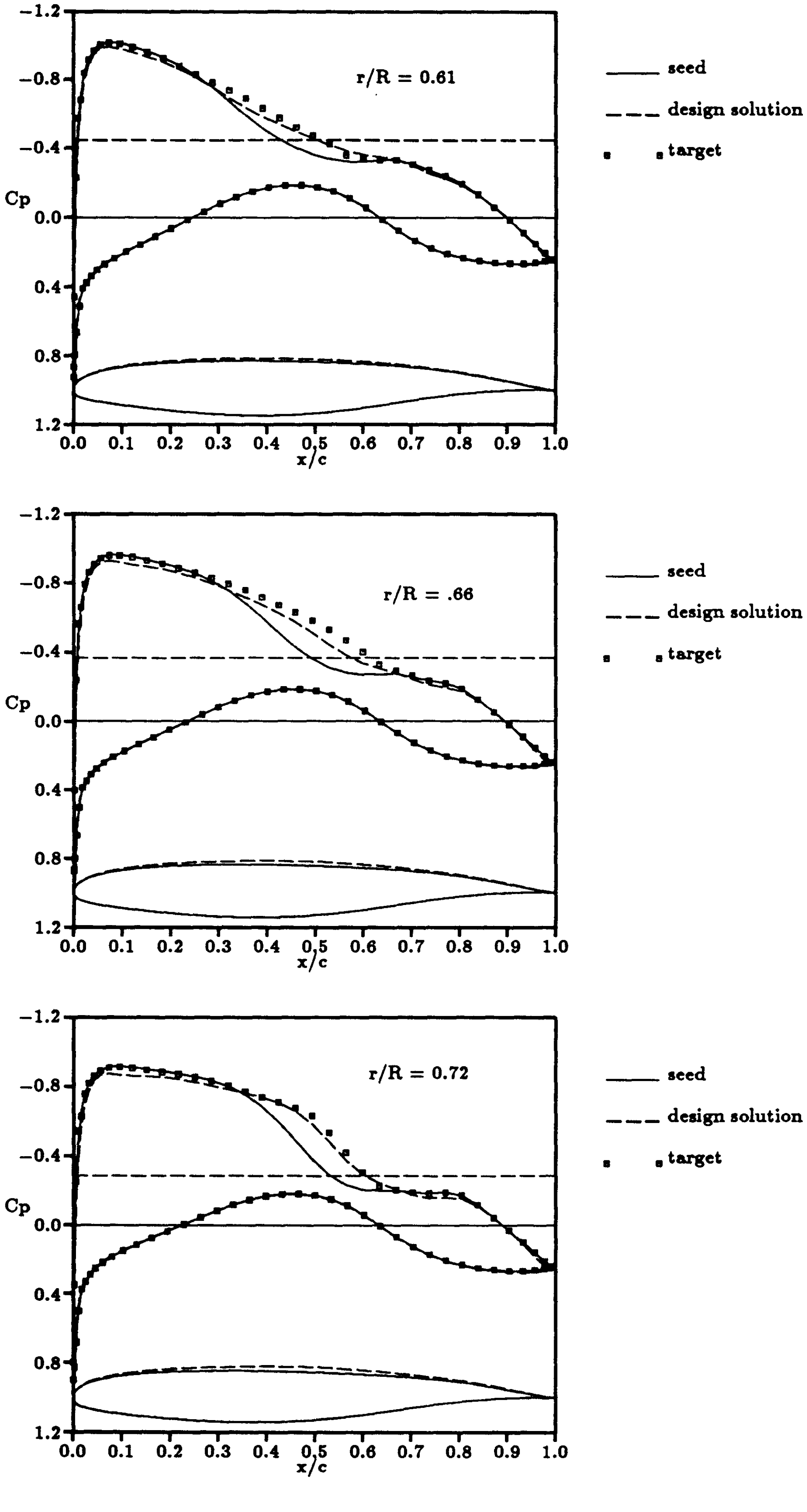


Figure 7.6: Sample redesign; $M = 0.6$, $\lambda = 0.7$.

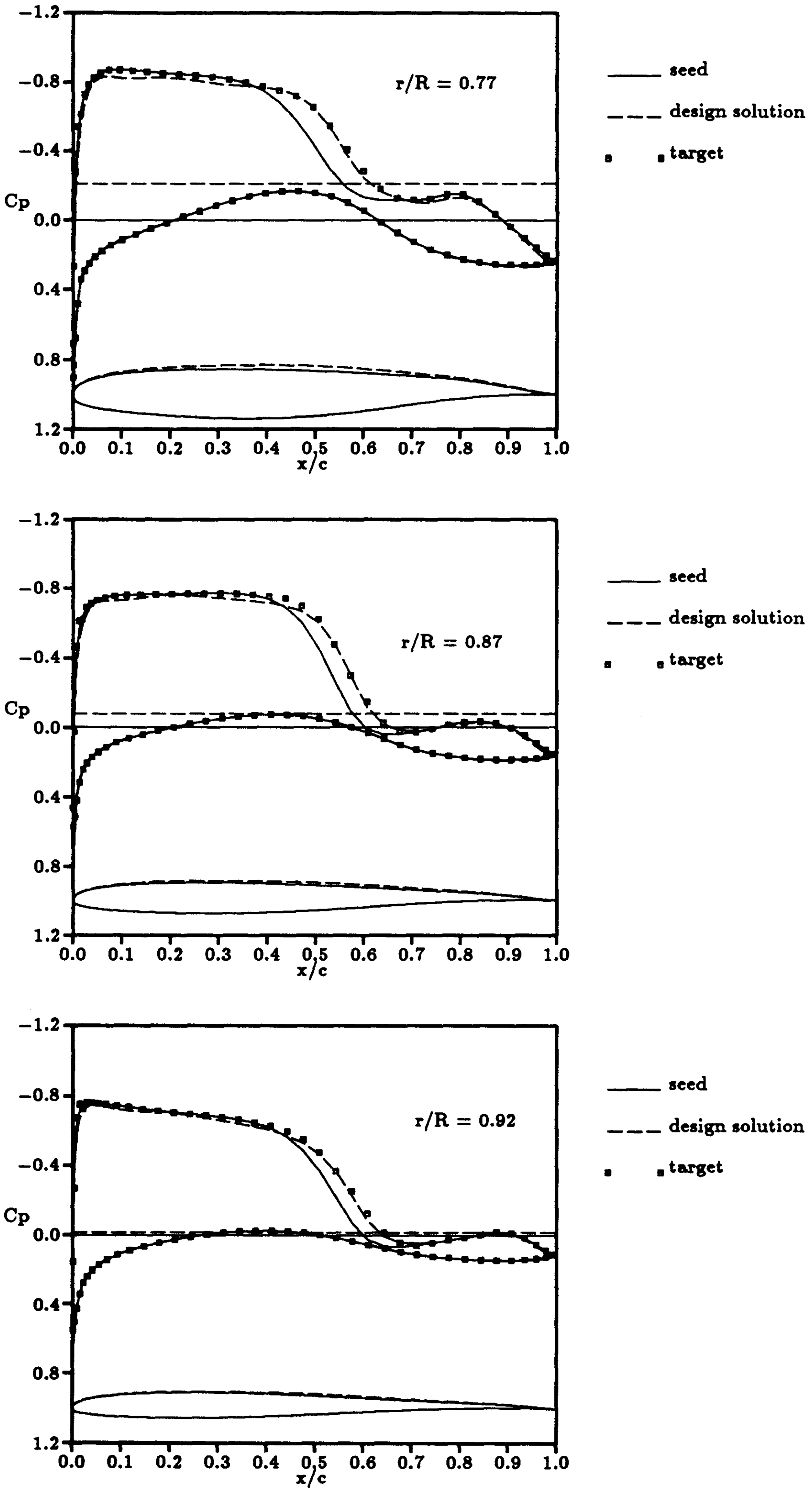


Figure 7.6: (continued).

Chapter 8
Conclusions and Recommendations

This thesis has presented an analysis and design method for transonic propellers based on the full potential equation. The method is based on a finite element discretization of the full potential equation and solution by the Newton-Raphson method for non-linear equation systems. The design method makes use of the Newton solution approach to obtain sensitivies to global unknown mode coefficients that describe perturbations to the initial geometry. The technique can quickly obtain the solution to the linearly perturbed geometry within each design cycle without solving the full non-linear equations. The result is a design method that, for the first target pressure distribution, requires approximately the same amount of time as a single direct calculation. The linearly perturbed solution for subsequent target pressure distributions can be obtained in much less time (at least 1 order of magnitude) than a direct calculation.

Results were presented that verify the analysis capability of the present method within the limits of the approximations made in using the full potential equation. Results presented for the design method showed that the linearly perturbed solution is a good approximation to the full direct calculation even for relatively large geometry perturbations. A simple redesign was presented for a propeller designed for a high altitude atmospheric research aircraft that included sweeping the blade to reduce compressibility losses. The propeller sweep had a significant effect on the local section loading that cannot be predicted with a two-dimensional cascade algorithm. The swept blade was also used in the design method to make small modifications to the pressure distribution. A set of 28 geometric modes (7 chordwise modes combined with 4 spanwise modes) provided an effective set for redesigning the upper surface of the blade.

Several ideas were mentioned for increasing the capability of the present method.

First, the most time-consuming task involves the inversion of a large ($\sim 10^5$) algebraic system of equations using SLOR. Great time savings could be realized by incorporating an acceleration technique such as multi-grid.

Second, an unstructured grid would increase the geometry capabilities of the present code. Although all cases presented in this thesis were adequately modeled using a structured grid, many more complicated propellers could not be analyzed with the present code simply because a satisfactory grid cannot be constructed. Geometric complications such as excessive blade sweep and very low blade pitch result in structured grids with highly skewed cells or cells with negative volumes. The finite element method is by nature an unstructured approach to discretizing equations. The present code could be easily modified to handle an unstructured grid. Of course, there are performance penalties (in both memory requirements and computational requirements) when using an unstructured grid. However, such a grid may be necessary for more complicated geometries.

Finally, the design method presented minimizes a function that effectively results in the best fit pressure distribution to a given target. Other functions are possible and could be useful. Any function that can be expressed as a function of the unknown potential in the domain can be minimized. In particular, one useful function would be one that results in the best fit of radial circulation distribution. Such a function, coupled with a set of design modes that effectively modified the radial variation of pitch could be very useful in designing the twist distribution for a propeller blade.

Bibliography

[1] R. J. Boswell and M. L. Miller. Unsteady propeller loading–measurement, correction, with theory and parametric study. Technical Report DTNSRDC 2625, October 1968.

[2] NASA Ames Research Center. Global stratospheric change: Requirements for a very-high-altitude aircraft for atmospheric research. NASA Conference Publication 10041, 1989.

[3] H. Deconinck and Ch. Hirsch. Finite element methods for transonic blade-to-blade calculation in turbomachines. *Journal of Engineering for Power*, Vol. 103:665–677, October 1981.

[4] M. Drela. *Two-Dimensional Transonic Aerodynamic Design and Analysis using the Euler Equations.* PhD thesis, Massachusetts Institute of Technology, December 1985.

[5] M. B. Giles. *Newton Solution of Steady Two-Dimensional Transonic Flow.* PhD thesis, Massachusetts Institute of Technology, June 1985.

[6] W. G. Habashi and M. M. Hafez. Finite element solutions of transonic flow problems. AIAA-81-1472, 1981.

[7] C. Y. Hsin. *Development and Analysis of Panel Methods for Propellers in Unsteady Flow.* PhD thesis, Massachusetts Institute of Technology, September 1990.

[8] J. T. Lee. *A Potential Based Panel Method for the Analysis of Marine Propellers in Steady Flow.* PhD thesis, Massachusetts Institute of Technology, August 1987.

[9] M. J. Lighthill. A new method of two-dimensional aerodynamic design. R&M 2112, Aeronautical Research Council, June 1945.

[10] R. C. Lock. Aerodynamic design methods for transonic wings. *Aeronautical Journal*, January 1990.

[11] Tad McGeer, John S. Langford, James G. Anderson, and Mark Drela. Theseus: A high-altitude aircraft for atmospheric science. Aurora Report 905, November 1990.

[12] C. Rohrbach, F. B. Metzger, D. M. Black, and R. M. Ladden. Evaluation of wind tunnel performance testings of an advanced 45° swept eight-bladed propeller at mach numbers from 0.45 to 0.85. NASA Contractor Report 3505, March 1982.

[13] M. P. Scully. *Computation of Helicopter Rotor Wake Geometry and its Influence on Rotor Harmonic Airloads.* PhD thesis, Massachusetts Institute of Technology, 1975.

Appendix A
Linearization of Discrete Equations

This appendix is a summary of equations useful in the development of the present method. Analytic and discrete expressions are listed as well as derivatives that are used in linearizing the discrete equations as required by the Newton-Raphson method. It is important to remember that ϕ is stored at the grid nodes while any quantity based on the gradient of ϕ (e.g. $\vec{W}, \bar{\rho}$) is calculated at the centroid of the hexahedral elements. Gradients can also be calculated at the centroid of element faces if the component of the gradient normal to the face is known.

A.1 Velocity at Element Centroid

$$\begin{aligned}
\vec{W} &= \nabla\phi - \vec{V}_{adv} \\
&= \sum_{j=1}^{8} \nabla N_j \phi_j - \vec{V}_{adv} \\
|\vec{W}|^2 &= \left(\sum_{j=1}^{8} N_{j_x}\phi_j - V_{adv_{(x)}}\right)^2 + \left(\sum_{j=1}^{8} N_{j_y}\phi_j - V_{adv_{(y)}}\right)^2 + \\
&\quad \left(\sum_{j=1}^{8} N_{j_z}\phi_j - V_{adv_{(z)}}\right)^2 \\
\frac{\partial|\vec{W}|^2}{\partial\phi_j} &= 2\vec{W}\cdot\nabla N_j
\end{aligned}$$

Evaluating ∇N_j at the element centroid is done as follows:

$$\nabla N_j = \left\{ \begin{array}{c} N_{j_x} \\ N_{j_y} \\ N_{j_z} \end{array} \right\} = \left[\begin{array}{ccc} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ x_\zeta & y_\zeta & z_\zeta \end{array} \right]^{-1} \left\{ \begin{array}{c} N_{j_\xi} \\ N_{j_\eta} \\ N_{j_\zeta} \end{array} \right\} \tag{A.1}$$

where

$$x_\xi = \sum_{j=1}^{8} N_{j_\xi} x_j, \qquad \text{etc.}$$

and

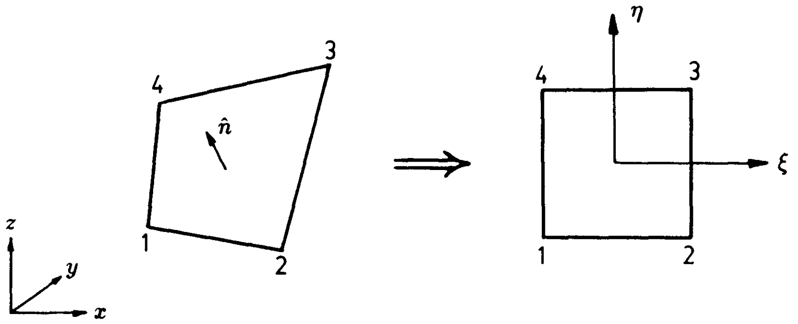
$$\begin{aligned} N_{j_\xi} &= \frac{\xi_j}{8}(1+\eta\eta_j)(1+\zeta\zeta_j) \\ N_{j_\eta} &= \frac{\eta_j}{8}(1+\xi\xi_j)(1+\zeta\zeta_j) \\ N_{j_\zeta} &= \frac{\zeta_j}{8}(1+\xi\xi_j)(1+\eta\eta_j) \end{aligned}$$

At the element centroid $(\xi, \eta, \zeta) = (0, 0, 0)$ so

$$\begin{aligned} N_{j_\xi} &= \frac{\xi_j}{8} \\ N_{j_\eta} &= \frac{\eta_j}{8} \\ N_{j_\zeta} &= \frac{\zeta_j}{8} \end{aligned}$$

A.2 Velocity at Face Centroid

The 3 components of velocity can be determined from 4 nodal values of ϕ for an element face if the component of velocity normal to the face is known. For a face that lies on a solid surface, the condition of flow tangency provides the required normal component of velocity. A general face is illustrated below.



A set of bilinear shape functions is defined for the two-dimensional face as

$$N_j^{(2)} = \frac{1}{4}(1+\xi\xi_j)(1+\eta\eta_j) \qquad j = 1,\ldots,4$$

The problem is to find $\nabla\phi$ given that $\nabla\phi\cdot\hat{n}$ is known. A matrix equation analogous to Equation A.1 for the two-dimensional problem is

$$\begin{Bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{Bmatrix} = \begin{bmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ x_{\hat{n}} & y_{\hat{n}} & z_{\hat{n}} \end{bmatrix}^{-1} \begin{Bmatrix} \phi_\xi \\ \phi_\eta \\ \nabla\phi\cdot\hat{n} \end{Bmatrix}$$

$$= \begin{bmatrix} \xi_x & \eta_x & \hat{n}_x \\ \xi_y & \eta_y & \hat{n}_y \\ \xi_z & \eta_z & \hat{n}_z \end{bmatrix} \begin{Bmatrix} \sum_{j=1}^{4} N_{j_\xi}^{(2)}\phi_j \\ \sum_{j=1}^{4} N_{j_\eta}^{(2)}\phi_j \\ \nabla\phi\cdot\hat{n} \end{Bmatrix}$$

where

$$x_\xi = \sum_{j=1}^{4} N_{j_\xi}^{(2)} x_j, \qquad \text{etc.}$$

and

$$\begin{aligned} N_{j_\xi}^{(2)} &= \frac{\xi_j}{4}(1+\eta\eta_j) \\ N_{j_\eta}^{(2)} &= \frac{\eta_j}{4}(1+\xi\xi_j) \end{aligned}$$

and $<x_{\hat{n}}, y_{\hat{n}}, z_{\hat{n}}>$ are simply the x, y, z components of the unit normal vector $\hat{n}$. Writing out the equation for ϕ_x gives

$$\begin{aligned} \phi_x &= \xi_x \sum_{j=1}^{4} N_{j_\xi}^{(2)} \phi_j + \eta_x \sum_{j=1}^{4} N_{j_\eta}^{(2)} \phi_j + \hat{n}_x(\nabla\phi \cdot \hat{n}) \\ &= \sum_{j=1}^{4} \left(N_{j_\xi}^{(2)} \xi_x + N_{j_\eta}^{(2)} \eta_x \right) \phi_j + \hat{n}_x(\nabla\phi \cdot \hat{n}) \end{aligned}$$

Let

$$\begin{aligned} N_{j_x}^{(2)} &\equiv N_{j_\xi}^{(2)} \xi_x + N_{j_\eta}^{(2)} \eta_x \\ N_{j_y}^{(2)} &\equiv N_{j_\xi}^{(2)} \xi_y + N_{j_\eta}^{(2)} \eta_y \\ N_{j_z}^{(2)} &\equiv N_{j_\xi}^{(2)} \xi_z + N_{j_\eta}^{(2)} \eta_z \end{aligned}$$

Then

$$\nabla\phi = \left\{ \begin{array}{c} \phi_x \\ \phi_y \\ \phi_z \end{array} \right\} = \sum_{j=1}^{4} \nabla N_j^{(2)} \phi_j + \left\{ \begin{array}{c} \hat{n}_x \\ \hat{n}_y \\ \hat{n}_z \end{array} \right\} \nabla\phi \cdot \hat{n}$$

with $\nabla\phi \cdot \hat{n}$ known. The velocity at the centroid of an element face, then, is simply

$$\begin{aligned}
\vec{W} &= \nabla\phi - \vec{V}_{adv} \\
&= \sum_{j=1}^{4} \nabla N_j^{(2)}\phi_j + \left\{ \begin{array}{c} \hat{n}_x \\ \hat{n}_y \\ \hat{n}_z \end{array} \right\} \nabla\phi\cdot\hat{n} - \vec{V}_{adv} \\
|\vec{W}|^2 &= \left(\sum_{j=1}^{4} N_{j_x}^{(2)}\phi_j + \hat{n}_x(\nabla\phi\cdot\hat{n}) - V_{adv_{(x)}} \right)^2 + \\
&\quad \left(\sum_{j=1}^{4} N_{j_y}^{(2)}\phi_j + \hat{n}_y(\nabla\phi\cdot\hat{n}) - V_{adv_{(y)}} \right)^2 + \\
&\quad \left(\sum_{j=1}^{4} N_{j_z}^{(2)}\phi_j + \hat{n}_z(\nabla\phi\cdot\hat{n}) - V_{adv_{(z)}} \right)^2 \\
\frac{\partial|\vec{W}|^2}{\partial\phi_j} &= 2\vec{W}\cdot\nabla N_j^{(2)}
\end{aligned}$$

A.3 Mach Number

$$\begin{aligned}
M^2 &= M_\infty^2 \frac{|\vec{W}|^2}{V_\infty^2} \left[1 + \frac{\gamma-1}{2} M_\infty^2 \left(1 - \frac{|\vec{W}|^2}{V_\infty^2} + \frac{|\Omega|^2 r^2}{V_\infty^2} \right) \right]^{-1} \\
\frac{\partial M^2}{\partial\phi} &= \frac{\partial M^2}{\partial|\vec{W}|^2} \frac{\partial|\vec{W}|^2}{\partial\phi} \\
\frac{\partial M^2}{\partial|\vec{W}|^2} &= M_\infty^2 \left[1 + \frac{\gamma-1}{2} M^2 \right] \left[1 + \frac{\gamma-1}{2} M_\infty^2 \left(1 - \frac{|\vec{W}|^2}{V_\infty^2} + \frac{|\Omega|^2 r^2}{V_\infty^2} \right) \right]^{-1}
\end{aligned}$$

A.4 Density

Density is upwinded in regions where the flow is supersonic. The amount of upwinding is controlled by the factor μ which itself takes on three different forms depending on the local flow conditions. The exact equation for density and its derivatives, then, has three different forms. In the following equations, j indexes over the nodes of the current element and k indexes over the nodes of the upstream element. The current element is labeled i and the upstream element, $i-1$.

$$\tilde{\rho} = \bar{\rho}_i - \mu(\bar{\rho}_i - \bar{\rho}_{i-1})$$

$$\begin{aligned}
\bar{\rho} &= \frac{\rho}{\rho_\infty} = \left[1 + \frac{\gamma - 1}{2} M_\infty^2 \left(1 - \frac{|\vec{W}|^2}{V_\infty^2} + \frac{|\vec{\Omega}|^2 r^2}{V_\infty^2}\right)\right]^{\frac{1}{\gamma-1}} \\
\frac{\partial\bar{\rho}}{\partial\phi} &= \frac{\partial\bar{\rho}}{\partial|\vec{W}|^2}\frac{\partial|\vec{W}|^2}{\partial\phi} \\
\frac{\partial\bar{\rho}}{\partial|\vec{W}|^2} &= -\frac{M_\infty^2}{V_\infty^2}\left[1 + \frac{\gamma - 1}{2} M_\infty^2 \left(1 - \frac{|\vec{W}|^2}{V_\infty^2} + \frac{|\vec{\Omega}|^2 r^2}{V_\infty^2}\right)\right]^{\frac{2-\gamma}{\gamma-1}}
\end{aligned}$$

$$\begin{aligned}
\mu &= \max\left[0,\ \mu_c\left(1 - \frac{M_{crit}^2}{M_i^2}\right),\ \mu_c\left(1 - \frac{M_{crit}^2}{M_{i-1}^2}\right)\right] \\
\frac{\partial\mu}{\partial\phi} &= \frac{\partial\mu}{\partial M^2}\frac{\partial M^2}{\partial\phi} \\
\frac{\partial\mu}{\partial M^2} &= \frac{\mu_c M_{crit}^2}{M^4}
\end{aligned}$$

Three cases must be considered to evaluate $\frac{\partial\tilde{\rho}}{\partial\phi}$:

Case 1: flow is subsonic in current and upstream elements.

$$\begin{aligned}
\frac{\partial\tilde{\rho}}{\partial\phi_j} &= \frac{\partial\bar{\rho}_i}{\partial\phi_j} \\
\frac{\partial\tilde{\rho}}{\partial\phi_k} &= 0
\end{aligned}$$

Case 2: flow is supersonic and accelerating

$$\begin{aligned}
\frac{\partial\tilde{\rho}}{\partial\phi_j} &= \frac{\partial\bar{\rho}_i}{\partial\phi_j} - \left[\frac{\partial\mu}{\partial\phi_j}(\bar{\rho}_i - \bar{\rho}_{i-1}) + \mu\frac{\partial\bar{\rho}_i}{\partial\phi_j}\right] \\
\frac{\partial\tilde{\rho}}{\partial\phi_k} &= \qquad - \left[\qquad\qquad\qquad -\mu\frac{\partial\bar{\rho}_{i-1}}{\partial\phi_k}\right]
\end{aligned}$$

Case 3: flow is supersonic and decelerating

$$\begin{aligned}
\frac{\partial \tilde{\rho}}{\partial \phi_j} &= \frac{\partial \bar{\rho}_i}{\partial \phi_j} - \left[\qquad\qquad\qquad\qquad \mu \frac{\partial \bar{\rho}_i}{\partial \phi_j} \right] \\
\frac{\partial \tilde{\rho}}{\partial \phi_k} &= \qquad - \left[\frac{\partial \mu}{\partial \phi_k} (\bar{\rho}_i - \bar{\rho}_{i-1}) - \mu \frac{\partial \bar{\rho}_{i-1}}{\partial \phi_k} \right]
\end{aligned}$$

A.5 Pressure Coefficient

Calculation of the pressure coefficient and its derivatives is necessary in the design procedure of the current method. Two different reference velocities for the pressure coefficient are useful. The local pressure coefficient (C_p) uses the advance velocity at the current radius as the reference velocity. C_p is useful for comparing section loads to two dimensional airfoil data. The global pressure coefficient (C_{p_g}) uses V_∞ as the reference velocity. C_{p_g} is useful in calculating total blade data (thrust, torque, etc) since all pressures are referenced to the same velocity. In the following expressions, V_{ref} can be replaced by V_∞ or V_{adv_r} as required.

$$\begin{aligned}
C_p &= \frac{\frac{p}{\rho_\infty} - \frac{p_\infty}{\rho_\infty}}{\frac{1}{2} V_{ref}^2} \\
& \frac{p_\infty}{\rho_\infty} = \frac{V_\infty^2}{\gamma M_\infty^2} \\
& \frac{p}{\rho_\infty} = \frac{\gamma - 1}{\gamma} \bar{\rho} \left(I - \frac{|\vec{W}|^2}{2} + \frac{\omega^2 r^2}{2} \right) \\
\frac{\partial C_p}{\partial \phi} &= \frac{\gamma - 1}{\gamma} \frac{1}{\frac{1}{2} V_{ref}^2} \left[\frac{\partial \bar{\rho}}{\partial \phi} \left(I - \frac{|\vec{W}|^2}{2} + \frac{\omega^2 r^2}{2} \right) - \frac{\bar{\rho}}{2} \frac{\partial |\vec{W}|^2}{\partial \phi} \right]
\end{aligned}$$

A.6 The Discrete Equations

For a general interior node, the discrete equation as presented in Chapter 3 is

$$R_\ell = \sum_{e=1}^{8} \left\{ \tilde{\rho}^e \left(\sum_{j=1}^{8} \phi_j^e k_{\ell j}^e \right) - \tilde{\rho}^e f_{v_\ell}^e \right\} = 0$$

The discrete equation is linearized so that

$$R_\ell^{n+1} = R_\ell^n + \left.\frac{\partial R_\ell}{\partial \phi}\right|^n \Delta\phi$$

Expressions for $\frac{\partial R_\ell}{\partial \phi}$ are listed below. In these expressions, j indexes over the 8 nodes of the current element in the summation and k indexes over the element upstream of the current element. In general, R_ℓ depends on 36 nodal values of ϕ.

$$\begin{aligned}
\frac{\partial R_\ell}{\partial \phi_j} &= \sum_{e=1}^{8} \left\{ \frac{\partial \tilde{\rho}^e}{\partial \phi_j} \left(\sum_{jj=1}^{8} \phi_{jj}^e k_{\ell jj}^e \right) + \tilde{\rho}^e k_{\ell j}^e - \frac{\partial \tilde{\rho}^e}{\partial \phi_j} f_{v_\ell}^e \right\} \\
\frac{\partial R_\ell}{\partial \phi_k} &= \sum_{e=1}^{8} \left\{ \frac{\partial \tilde{\rho}^e}{\partial \phi_k} \left(\sum_{jj=1}^{8} \phi_{jj}^e k_{\ell jj}^e \right) - \frac{\partial \tilde{\rho}^e}{\partial \phi_k} f_{v_\ell}^e \right\}
\end{aligned}$$

The discrete equations for boundary nodes were presented in Table 3.1. These equations are linearized by calculating simple derivatives as shown above for the interior discrete equation. The only difference is that there is an added dependence on Γ for nodes on the wake and nodes on the farfield boundaries.

On the wake $\qquad \frac{\partial R_\ell}{\partial \Gamma} = -1$

On the inflow farfield boundary $\qquad \frac{\partial R_\ell}{\partial \Gamma} = -\frac{\partial \phi_{ff}}{\partial \Gamma}$

On the tip farfield boundary $\qquad \frac{\partial R_\ell}{\partial \Gamma} = \sum_{e=1}^{\leq 4} \left\{ -\tilde{\rho}^e \oint_{S_e} N_\ell^e \left(\frac{\partial \vec{W}_{ff}}{\partial \Gamma} \cdot \hat{n} \right) \, dS \right\}$

For the design method, the derivative $\frac{\partial R}{\partial A}$ is required. As presented in Section 6.2, the residual equation for nodes on the design surface is modified so that

$$R_d = \ldots \sum_{ef=1}^{\leq 4} (\rho V_w)^{ef} \int_{S_{ef}} N_d \, dS$$

Then

$$\frac{\partial R_d}{\partial A_\ell} = \sum_{ef=1}^{\leq 4} \frac{\partial (\rho V_w)^{ef}}{\partial A_\ell} \int_{S_{ef}} N_d \, dS$$

where

$$\begin{aligned} (\rho V_w)^{ef} &= (\tilde{\rho} \vec{W})^{ef} \cdot \nabla \tilde{\Delta} \\ \frac{\partial (\rho V_w)^{ef}}{\partial A_\ell} &= (\tilde{\rho} \vec{W})^{ef} \cdot \nabla \frac{\partial \tilde{\Delta}}{\partial A_\ell} \\ \tilde{\Delta} &= \sum_{\ell=1}^{L} A_\ell f_\ell \\ \frac{\partial \tilde{\Delta}}{\partial A_\ell} &= f_\ell \end{aligned}$$