A Numerical Investigation of Hot Streaks in Turbines

by

Björn Bengt Johan Krouthén

Civ. Ing. Engineering Physics, Uppsala University
(1985)

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

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

May 1988

Signature of Author ______________________________
Department of Aeronautics and Astronautics
May 6, 1988

Certified by ______________________________
Professor Michael B. Giles
Thesis Supervisor, Department of Aeronautics and Astronautics

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

A Numerical Investigation of Hot Streaks in Turbines

by

Björn Bengt Johan Krouthén

Submitted to the Department of Aeronautics and Astronautics, May 6, 1988 in partial
fulfillment of the requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

A numerical method is developed for simulation of hot streak redistribution in a 2-dimensional model of a turbine rotor. The flow domain is divided into a viscous region near the blade where the Reynolds-averaged, thin layer Navier-Stokes equations are solved using an implicit finite volume technique, and an inviscid core region where the Euler equations are solved using an explicit finite volume method. The computational mesh consists of an O-mesh and a H-mesh patched together smoothly to cover the domain of interest. At the interface between the inviscid and the viscous regions the numerical schemes are connected using a formula that is conservative.

Computations are performed using three different flow conditions. Hot streaks with a temperature ratio of 2.0 are used. The first test case assumes a fully turbulent boundary layer and a tangential inflow angle of 40°. The result is in agreement with data from a previous numerical investigation. The second test case is run with a fully turbulent boundary layer while the third test case is run laminar with transition on the suction surface. In the second and the third test case the tangential inflow angle is 45°. The computed solution from all three test cases predicts a migration of hot gas to the pressure surface, which also has been observed experimentally.

Thesis Supervisor: Michael B. Giles, Assistant Professor of Aeronautics and Astronautics

Jag vill här nämna de personer som gjort detta 16 månader långa äventyr möjligt. Dessa är:

Stig Lundgren, Vice GD. FFA

Mårten Landahl, Professor vid MIT

Michael Giles, Assistant Professor vid MIT

Trägen vinner.

Contents

List of Figures

List of Tables

Symbols

A	cell area
a	speed of sound
C_f	skin friction coefficient
C_p	pressure coefficient
$\bar{C}_p$	time averaged pressure coefficient
$\bar{C}_T$	time averaged temperature coefficient
c	cascade airfoil axial chord length
c_p	specific heat at constant pressure
D	fourth order dissipation function
e	total internal energy per unit volume
F	x component of inviscid flux vector
F^v	x component of viscous flux vector
G	y component of inviscid flux vector
G^v	y component of viscous flux vector
H	total enthalpy
i, j	grid coordinates
M	Mach number
$\bar{n}$	normal unit vector
p	static pressure
Pr	Prandtl number
P_s	pitch
q	state vector
r_p	exit static to inlet total pressure ratio
Re_c	Reynolds number based on axial chord length
S_k	constant in Sutherland's formula
T	static temperature
t	time
u, v	Cartesian velocity components
u^+, y^+	turbulent inner variables
V_{rot}	rotor speed
W	hot streak shape function
x, y	Cartesian coordinates
å,ä,ö	additional Swedish letters

Greek:

α	inlet angle
β	outlet angle
γ	ratio of specific heats
δ_x, δ_y	first order difference operators
$\epsilon_{ij}^{(4)}$	fourth order dissipation coefficient
ξ, η	computational coordinates
θ	spatial wave number
κ	thermal conductivity
μ	viscosity
μ_x, μ_y	averaging operators
$\bar{\pi}_k(z)$	stability polynomial
ρ	static density
τ_g	O-mesh thickness
φ	arbitrary scalar function
ϕ	flow coefficient
Φ	vector of characteristic variables
Ω	computational domain
ω	angular frequency

Subscripts:

av	averaged values
i, j	grid indices
ihs	inlet-hot streak values
is	inlet-steady values
iw	inlet-wake values
l	laminar condition
min	minimum quantities
max	maximum quantities
t	turbulent condition
$w, wall$	wall value
0	stagnation conditions

Superscripts:

n	time level

Chapter 1
Introduction

1.1 Physical Problem

To achieve a high cycle efficiency and hence a lower fuel consumption in a modern jet-engine, the gas turbine industry has moved towards designs with extremely high temperature gas leaving the combustor. It is known that the exit flow from the combustor entering a turbine stage will have wide spatial variations in temperature both radially and circumferentially. The number of burners and nozzle guide vanes are usually selected so that the locally hot gas will pass through the center of the passage between two vanes. The temperature tolerance of the guide vanes is usually based on a scaled average value of the exit combustor temperature. Due to introduction of cooling air in the nozzel, the turbine entry temperature (TET) measured behind the guide vanes is lower than the combustor exit temperature. The temperature tolerance of the blades in the first rotor row is based on a scaled,averaged TET. Recent investigations, however, have shown that hot gas migrates to the pressure surface of the rotor blade. This can lead to peak temperatures on the rotor blade that might exceed acceptable metal temperatures by as much as $250°C - 500°C$, leading to blade failures. This indicates that the scaled TET, which is presently used, is too low an estimate for the rotor surface temperature.

Different methods are presently used to keep the high blade temperatures within acceptable limits. Three methods are used for direct cooling of the turbine airfoil: Internal cooling passages, showerhead impingement cooling and external film cooling. In addition to these direct cooling methods, a more implicit method is used. By introducing dilution cooling air into the hot gas, the exit temperature distribution from the combustor can be modified so that the hot fluid which has maximum contact with the blade surfaces is cooled most. The optimization of the cooling is critically dependent on the redistribution of the hot streaks coming from the combustor. It is therefore important that the aerodynamics of the redistribution in the turbine airfoil passages is clearly understood. It would be of considerable advantage for a turbine designer to explicitly know the redistribution of both hot and cold fluid when designing a cooling scheme for the turbine.

In an earlier work by Butler et. al. [5], an experimental and analytical investigation of the redistribution process for an axial turbine stage were presented. In the experiment, a streak of hot air seeded with CO_2 was introduced at one circumferential location upstream of the inlet guide vane. The redistribution of the hot streak was determined by measuring the concentration of CO_2 inside the turbine stage. Measurements of CO_2 taken on the rotor surface indicated that hot and cold gas had been segregated with the cold gas migrating to the suction surface and the hot gas to the pressure surface. In the same paper it was postulated that the segregation effect was due to the difference in rotor relative inlet angles of the hot and the cold gases. The postulate is based on an observation by Kerrebrock and Mikolajczak [19] in their work on wake transport in compressors. The experiment also showed some significant three-dimensional effects, such as the streamline pattern on the rotor blade caused by secondary flow which, in turn, was caused by the temperature distortion.

The segregation phenomenon observed in [5] has been investigated numerically by Rai and Dring [26]. In this paper Rai uses a two-dimensional Navier-Stokes solver to simulate the redistribution of hot gas in a two-dimensional model of the turbine stage used in [5]. The paper includes a comparison between Rai's computations and Butler's measurements. The results are lacking in agreement, especially the temperature distribution on the pressure surface of the rotor. The postulated migration of hot air to the pressure surface was not predicted by the computation. The poor agreement between calculations and experiment was blamed on two basic differences in flow conditions. The first difference was that the temperature ratio between hot and cold gas was 1.2 in the calculations and 2.0 in the experiment. In a previous experiment by Stabe, Whitney and

Moffit [30] it was found that a temperature ratio of 1.2 between hot and cold gas did not effect the turbine performance. The second difference was that the flow coefficient (ratio between axial inlet velocity and circumferential velocity) was 0.78 in the calculations and 0.68 in the experiment. In [26] it was believed the lower flow coefficient would accentuate the hot streak accumulation on the rotor pressure side.

1.2 Numerical Simulations

The temperature redistribution problem described above is a very challenging problem for any numerical algorithm due to its strong non linear nature. Provided that the solution is correct a computational method has some advantages over Butlers experiment. Instead of measuring CO_2 concentration as in the experiment, the whole temperature field including the surface temperature can be obtained directly from the calculated solution. Viscous phenomenon such as growth of a boundary layer near the blade surface and vortex shedding from the trailing edge are very local phenomenon. Their redistribution and transport in large regions of the flow path is basically governed by the inviscid Euler equations. It is therefore reasonable to assume that, in a narrow region around the blade the flow is governed by the viscous Navier-Stokes equations and, elsewhere, governed by the Euler equations. Considering the high computational cost involved in the simulation of a viscous, unsteady flow it would be advantageous if the computational domain could be divided into a viscous and an inviscid region. Instead of using an expensive implicit, time accurate method in the whole domain its use could then be restricted to the viscous region close to the blade surface and a less expensive explicit method could be used in the outer, inviscid region.

In a typical turbine blade design the blade will have substantial thickness, it will be highly cambered and its leading and trailing edges will be rounded. It is therefore clear that, for a viscous calculation, an O-type structured mesh is best suited to resolve the boundary layer around the blade and especially at the leading and trailing edges where the flow gradients are large. However, to have an O-mesh in the complete computational domain around the turbine blade does have some disadvantages. Due to the spatial

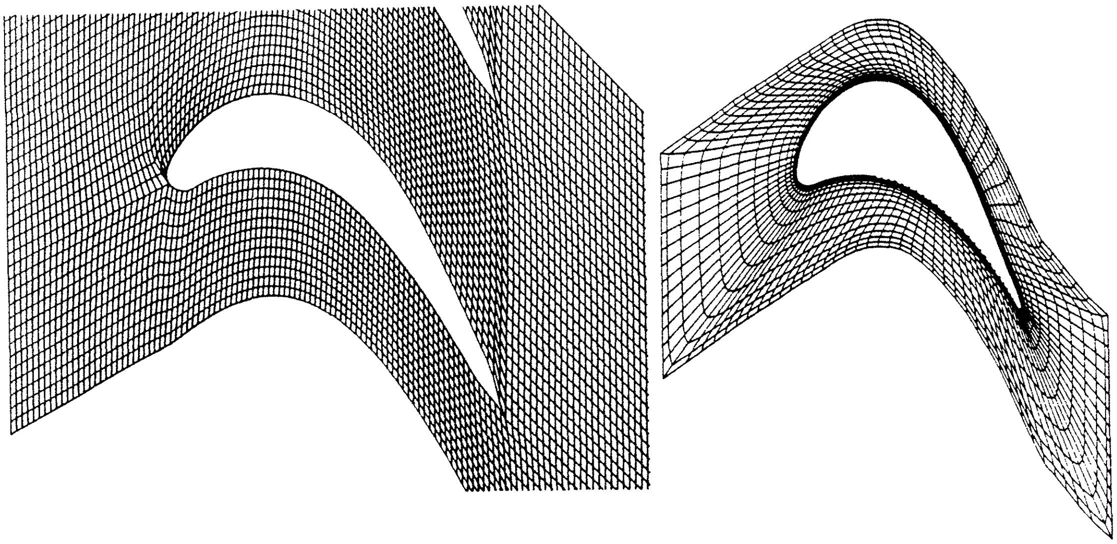

Figure 1.1: H and O-meshes in a 2-D turbine rotor

periodicity the O-mesh will inevitably become very skewed at the inflow and periodic boundaries and hence increase the numerical errors introduced in the solution. The strong non-linearity of the incoming hot streak could introduce irrecoverable errors if the cells at the inflow boundary are too skewed. An H-mesh on the other hand has very good farfield properties but has the disadvantage of resolving the leading and the trailing edges very poorly. This leads to the conclusion that for a typical turbine configuration, no single grid system offers satisfactory grid properties in the entire turbine stator or rotor passage. Figure 1.1 shows examples of H and C-meshes in a turbine.

The idea of dividing the computational mesh in a turbine passage to patched subgrids which matches the different characteristic flow regions, was introduced by Norton, Thompkins and Haimes [24]. In that paper they divided the computational mesh into an O-mesh in the viscous region close to the blade, and an H-mesh in the core flow region. As a numerical algorithm for solving the Reynolds averaged, Navier-Stokes equations, they used a cell-centered finite volume technique. To integrate the solution in time an implicit Beam-Warming time integration scheme was used. In both regions the full

Navier-Stokes equations were solved with the difference that in the inner O-mesh region a turbulence model was applied. Due to a pointer system in the solution algorithm, the scheme became grid transparent and hence very flexible. The idea of splitting the computational grid into different sub-grids was carried even further by Nakahashi and Obayashi [21]. In their paper they used a structured C-mesh in the wake and boundary layer regions and an unstructured, triangular grid in the remaining region. They applied the mesh generation to a bi-airfoil configuration. In the wake-boundary layer region the Reynolds-averaged, thin layer Navier-Stokes equations were solved using an implicit finite-difference algorithm. In the outer region the viscous terms were neglected and the Euler equations were solved with a finite element method. The method was shown to give very good results for the bi-airfoil case. Nakahashi et al. also successfully used their patched zonal method for a two dimensional turbine cascade configuration [22].

1.3 Present Work

In the present work a method to solve the unsteady Reynolds-averaged Navier-Stokes equations in a two-dimensional turbine rotor configuration is developed. A few of the ideas described in the previous section are used together with some completely new approaches. The computational domain is divided into a viscous region close to the turbine blade and a inviscid region consisting of the core flow. Consistently the computational mesh is divided into two patched grids, an O-mesh in the viscous boundary layer region and an H-mesh in the core flow region. In the viscous region an implicit finite-volume technique is used to solve the thin layer Navier-Stokes equations. A two-layer algebraic turbulence model due to Baldwin-Lomax is used to model the turbulent viscosity. The same finite-volume discretization is used for the Euler equations in the inviscid region. Since the same discretization is used in the two regions the finite-volume operator will become transparent to the mesh interface. In the inviscid region, however, an explicit Runge-Kutta time integration algorithm is used to advance the solution in time. To be able to simulate hot streaks coming in from the stator a special technique for treating the inflow and outflow boundaries is applied. The two numerical methods that are used in the work are described and evaluated in chapters 2 and 3 of the thesis. A thorough stability and accuracy analysis of the two algorithms is also presented. The generation of the complete computational grid is described in chapter 4. The generation of the mesh consists of two different techniques. The first is an algebraic technique to generate the body fitted O-mesh and the second is an elliptic PDE solver that generates an H-mesh in the core flow region. In chapter 5 the interface between the explicit and the implicit methods is discussed. The algorithm used to give a transparent connection between the two regions is described, and in particular the question of accuracy and consistency of the interface is discussed.

In chapter 6, the present numerical algorithm is used for the simulation of the temperature redistribution problem described in the beginning of the introduction. Simulations are performed under three different flow conditions in order to give results that can be compared with both previous calculations and experiment. The results include time-averaged surface temperature distributions. The temporal variation of the temperature distribution is also given to show some of the characteristics of the redistribution. Snapshots of the solution at different time levels of the cycle are given in form of iso-therm where the convection of the hot streak through the rotor can be studied. A comparisons between results from the present method, a previous computation [26] and the experiment [5] is presented for the time-averaged pressure distribution and the time-averaged temperature distribution.

Chapter 2

Thin Shear Layer Navier-Stokes Code

Since the main physical objective of this work is to study unsteady heat transfer effects and temperature redistribution in a turbine rotor with incoming hot streaks it is reasonable to assume that, in a narrow region around the blade, the streamwise viscous stresses can be neglected and so the flow is governed by the thin shear layer equations which will be referred to as the TSL equations. The TSL equations are obtained from the full Navier-Stokes equations, governing a viscous fluid flow, by neglecting shear stresses in the streamwise direction. Due to a relatively high Reynolds number and a large adverse pressure gradient on the suction surface, the flow in the viscous region close to the turbine blade will inevitably become turbulent. The numerical method used in this work will not be able to accurately resolve the unsteady turbulent length scales and therefore a turbulence model will be used.

This chapter will basically give a derivation and a validation of the numerical method used for the solution of the Reynolds-averaged TSL equations. First a derivation of the non-dimensional Navier-Stokes equations is given. In the following section the finite volume approach suggested by Swanson & Turkel [32] is described. The definition and numerical implementation of boundary conditions are given in the two following sections. Section 2.7 describes the time integration method used. Since the computations are to be done time accurately on a mesh with large differences in the cell size, an implicit Beam & Warming time integration method will be used to advance the solution in time. The time integration section is followed by a section that describes the linearization of the flux-vectors. The turbulence model used here is an algebraic method known as the Baldwin-Lomax method and is described in Section 2.9. A thorough analysis of the implicit scheme with respect to accuracy and stability is given in Section 2.10. In the last section of the chapter computational results from two different cases are presented in order to validate the code. The flat plate is used to validate both the laminar and the turbulent version of the code. The laminar model is also validated by solving the flow in a channel with a 10% thick circular bump.

2.1 Governing Equations

The system of equations governing a viscous, compressible flow is known as the Navier-Stokes equations. The full 2-D Navier-Stokes equations, expressed in Cartesian coordinates and on differential form, can be written as:

$$\frac{\partial q}{\partial t} = \frac{\partial(-F+F^v)}{\partial x} + \frac{\partial(-G+G^v)}{\partial y} \tag{2.1}$$

where:

$$q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix}, F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(e+p) \end{bmatrix}, G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(e+p) \end{bmatrix} \tag{2.2}$$

$$F^v = \begin{bmatrix} 0 \\ \lambda(u_x + v_y) + 2\mu u_x \\ \mu(u_y + v_x) \\ \mu v(u_y + v_x) + \lambda u(u_x + v_y) + 2\mu u u_x + \kappa T_x \end{bmatrix} \tag{2.3}$$

$$G^v = \begin{bmatrix} 0 \\ \mu(u_y + v_x) \\ \lambda(u_x + v_y) + 2\mu v_y \\ \mu u(u_y + v_x) + \lambda v(u_x + v_y) + 2\mu v v_y + \kappa T_y \end{bmatrix} \tag{2.4}$$

where ρ is the density, u and v are x and y velocity components respectively, p is the pressure, T is the temperature and e is the total energy per unit volume. The subscripts $_x$ and $_y$ denotes partial derivatives with respect to x and y. The coefficients of viscosity λ and μ that occur in the viscous flux-vectors can be related by the Stokes hypothesis

$$\lambda = -\frac{2}{3}\mu \tag{2.5}$$

The system of equations (2.1) is closed by two equations of state that give a relation between the thermodynamic variables. These have the form

$$p = p(e,\rho) \quad , \quad T = T(e,\rho) \tag{2.6}$$

for a stationary gas. Under the assumption that the fluid is a perfect gas the pressure relation becomes

$$p = (\gamma - 1)(e - \frac{1}{2}\rho(u^2 + v^2)) \tag{2.7}$$

and the temperature relation becomes

$$T = (\frac{\gamma}{c_p})\frac{(e - \frac{1}{2}\rho(u^2 + v^2))}{\rho} \tag{2.8}$$

where c_p is the specific heat at constant pressure and γ the ratio of specific heats. The coefficient of viscosity μ is here related to the thermodynamic variables by Sutherland's formula which states

$$\mu(T) = \mu_\infty \left(\frac{T}{T_\infty}\right)^{\frac{3}{2}} \left(\frac{T_\infty + S_k}{T + S_k}\right) \tag{2.9}$$

with S_k being a constant. The definition of the Prandtl number

$$\mathrm{Pr} = \frac{c_p \mu}{\kappa} \tag{2.10}$$

will finally give a relation between the thermal conductivity coefficient κ and μ since the Prandtl number is approximately constant for most gases.

2.2 Nondimensionalization

The following dimensional upstream quantities are used for the nondimensionalization of the fluid and thermodynamic variables: The axial chord length c, speed of sound a_∞, density ρ_∞, coefficient of viscosity μ_∞ and temperature T_∞. This gives the following expressions for the nondimensional (barred) quantities.

$$\begin{array}{lll} \bar{x} = \dfrac{x}{c}, & \bar{y} = \dfrac{y}{c}, & \bar{t} = \dfrac{t}{c/a_\infty} \\[2ex] \bar{u} = \dfrac{u}{a_\infty}, & \bar{v} = \dfrac{v}{a_\infty}, & \bar{\rho} = \dfrac{\rho}{\rho_\infty} \\[2ex] \bar{p} = \dfrac{p}{\rho_\infty a_\infty^2}, & \bar{T} = \dfrac{T}{T_\infty}, & \bar{\mu} = \dfrac{\mu}{\mu_\infty} \\[2ex] \bar{e} = \dfrac{e}{\rho_\infty a_\infty^2} & & \end{array} \tag{2.11}$$

Introducing the Reynolds number Re_c based on the upstream quantities and on the axial chord

$$\mathrm{Re}_c = \frac{\rho_\infty a_\infty c}{\mu_\infty} \tag{2.12}$$

the nondimensionalization of the Navier-Stokes equations (2.1) yields

$$\frac{\partial \bar{q}}{\partial \bar{t}} + \frac{\partial \bar{F}}{\partial \bar{x}} + \frac{\partial \bar{G}}{\partial \bar{y}} = \frac{1}{\mathrm{Re}_c}\left(\frac{\partial \bar{F}^v}{\partial \bar{x}} + \frac{\partial \bar{G}^v}{\partial \bar{y}}\right) \tag{2.13}$$

where:

$$\bar{q} = \begin{bmatrix} \bar{\rho} \\ \bar{\rho}\bar{u} \\ \bar{\rho}\bar{v} \\ \bar{e} \end{bmatrix}, \bar{F} = \begin{bmatrix} \bar{\rho}\bar{u} \\ \bar{\rho}\bar{u}^2 + \bar{p} \\ \bar{\rho}\bar{u}\bar{v} \\ \bar{u}(\bar{e} + \bar{p}) \end{bmatrix}, \bar{G} = \begin{bmatrix} \bar{\rho}\bar{v} \\ \bar{\rho}\bar{u}\bar{v} \\ \bar{\rho}\bar{v}^2 + \bar{p} \\ \bar{v}(\bar{e} + \bar{p}) \end{bmatrix} \tag{2.14}$$

$$\bar{F}^v = \begin{bmatrix} 0 \\ \bar{\mu}\frac{2}{3}(2\bar{u}_x - \bar{v}_y) \\ \bar{\mu}(\bar{u}_y + \bar{v}_x) \\ \bar{\mu}\bar{v}(\bar{u}_y + \bar{v}_x) + \bar{\mu}\bar{u}\frac{2}{3}(2\bar{u}_x - \bar{v}_y) + \frac{\bar{\mu}}{(\gamma-1)\mathrm{Pr}}\bar{T}_x \end{bmatrix} \tag{2.15}$$

$$\bar{G}^v = \begin{bmatrix} 0 \\ \bar{\mu}(\bar{u}_y + \bar{v}_x) \\ \bar{\mu}\frac{2}{3}(2\bar{v}_y - \bar{u}_x) \\ \bar{\mu}\bar{u}(\bar{u}_y + \bar{v}_x) + \bar{\mu}\bar{v}\frac{2}{3}(2\bar{v}_y - \bar{u}_x) + \frac{\bar{\mu}}{(\gamma-1)\mathrm{Pr}}\bar{T}_y \end{bmatrix} \tag{2.16}$$

The closing equations of state become

$$\bar{p} = (\gamma - 1)(\bar{e} - \frac{1}{2}\bar{\rho}(\bar{u}^2 + \bar{v}^2)) \tag{2.17}$$

and

$$\bar{T} = \gamma(\gamma - 1)\frac{(\bar{e} - \frac{1}{2}\bar{\rho}(\bar{u}^2 + \bar{v}^2))}{\bar{\rho}} = \frac{\gamma\bar{p}}{\bar{\rho}} \tag{2.18}$$

and finally the Sutherland law becomes

$$\bar{\mu} = \bar{T}^{\frac{3}{2}} \left(\frac{1 + \tilde{S}_k}{\bar{T} + \tilde{S}_k} \right) \tag{2.19}$$

where $\tilde{S}_k = \frac{S_k}{T_\infty}$. For convenience the overbar will now be dropped in all subsequent equations.

2.3 Spatial Discretization

To be able to solve the Navier-Stokes equations numerically some sort of spatial discretization has to be performed. Following a standard approach of a finite volume technique (cf. [10]) the computational region, denoted Ω, is divided into a number of quadrilaterals forming a computational mesh. Integrating equation (2.13) over Ω will give the Navier-Stokes equations in integral form. If in addition Greens theorem is used this integral form becomes

$$\int_{\Omega} \frac{\partial q}{\partial t} dV + \oint_{\partial\Omega} (F dy - G dx) = \frac{1}{\mathrm{Re}_c} \oint_{\partial\Omega} (F^v dy - G^v dx) \tag{2.20}$$

where the line integrals are evaluated in a counter-clockwise direction. This integral relation holds on each quadrilateral subdomain ω_{ij} as well as on the entire region Ω. If it is assumed that the state vector q is piecewise constant on each cell and the cell shape is fixed the integral can be written for each cell as

$$\frac{dq_{ij}}{dt} A_{ij} + \oint_{\partial\omega_{ij}} (F dy - G dx) = \frac{1}{\mathrm{Re}_c} \oint_{\partial\omega_{ij}} (F^v dy - G^v dx) \tag{2.21}$$

where A_{ij} denotes the area of cell i, j. The approximation of the line integral proposed by Swanson & Turkel [32] can be used under the assumption that the flux-vectors F, G, F^v and G^v are piecewise constant on the cell boundary. The semi-discrete formulation of the integral equation will become (see fig. 2.1 for notations)

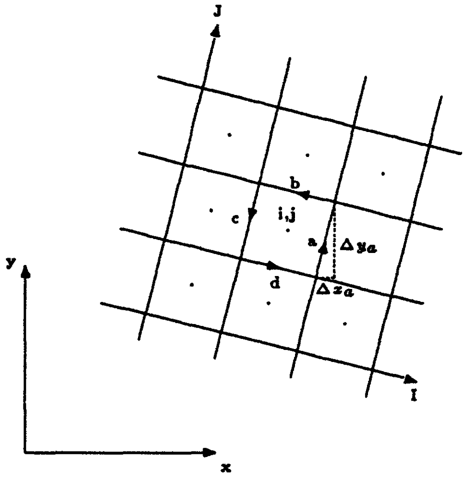

Figure 2.1: Identifying computational mesh.

$$
\begin{aligned}
\left(\frac{\partial q}{\partial t}\right)_{ij} &= \frac{1}{2A_{ij}}[-(F_{i+1j}+F_{ij})\Delta y_a - (F_{i-1j}+F_{ij})\Delta y_c \\
&\quad +(G_{i+1j}+G_{ij})\Delta x_a + (G_{i-1j}+G_{ij})\Delta x_c] \\
&+ \frac{1}{2A_{ij}}[-(F_{ij+1}+F_{ij})\Delta y_b - (F_{ij-1}+F_{ij})\Delta y_d \\
&\quad +(G_{ij+1}+G_{ij})\Delta x_b + (G_{ij-1}+G_{ij})\Delta x_d] \\
&+ \frac{1}{\mathrm{Re}_c A_{ij}}(+F_b^v\Delta y_b + F_d^v\Delta y_d - G_b^v\Delta x_b - G_d^v\Delta x_d) \\
&= -\mathbf{R}(q_{i-1j}, q_{ij-1}, q_{ij}, q_{i+1j}, q_{ij+1}) \qquad (2.22)
\end{aligned}
$$

where Δx and Δy are x and y components of the tangential vector defined on each cell face (see Fig. 2.1). Note that because of the TSL assumption F^v and G^v are assumed to be zero on faces a and c. The method of describing the gradients of a function in a finite volume manner was proposed by Peyret & Taylor [25]. The method has successfully been used by Swanson & Turkel [32] and by Müller & Rizzi [20] . The method can be described as follows. The gradient of a arbitrary scalar function $\varphi = \varphi(x, y)$ taken at point p' in the volume V' can be expressed by the following approximation (see fig. 2.2 for notations)

$$(\varphi_x, \varphi_y)_{p'} = \nabla\varphi \simeq \frac{\int_{V'} \nabla\varphi dV}{V'} = \frac{\oint_{\partial V'} \varphi d\vec{A}}{V'} \simeq \frac{1}{V'}\sum_{k=1}^{4}\varphi_k \vec{n}_k \qquad (2.23)$$

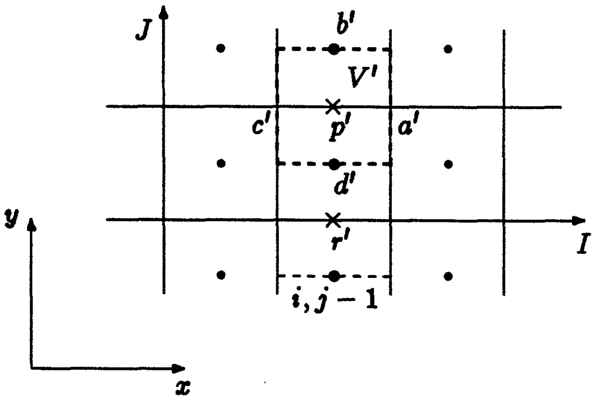

Figure 2.2: Identifying viscous cells in the computational mesh.

where the line integral is evaluated counter-clockwise and thus $\vec{n}$ in the sum are the normal face vectors. φ is assumed to be constant along each cell face. In the TSL limit the cell faces a' and c' becomes small compared to b' and d'. Figure 2.3 shows the computational mesh in the boundary layer region close to the wall. With the TSL assumptions the approximation of the line integral becomes:

$$\frac{\oint_{\partial V'} \varphi d\vec{A}}{V'} \simeq \frac{1}{V'}(\varphi_{b'}\Delta y_{b'} + \varphi_{d'}\Delta y_{d'} \ , \ -\varphi_{b'}\Delta x_{b'} - \varphi_{d'}\Delta x_{d'}) \tag{2.24}$$

where again Δx and Δy are the x and y components of the tangential vector defined on the two cell faces. This will give the following approximation of the gradient at point p' :

$$(\varphi_x)_{p'} = \frac{2}{A_{ij} + A_{ij+1}}(\varphi_{b'}\Delta y_{b'} + \varphi_{d'}\Delta y_{d'}) \tag{2.25}$$

$$(\varphi_y)_{p'} = \frac{2}{A_{ij} + A_{ij+1}}(-\varphi_{b'}\Delta x_{b'} - \varphi_{d'}\Delta x_{d'}) \tag{2.26}$$

where A_{ij} is the area of cell ij. There is, however, a problem with the above formulation of the viscous terms. If the flow is stagnant or has uniform velocity and the mesh is non-orthogonal the viscous terms are non-zero. To get around this problem the following approximation is made. Instead of taking both the sides in the computational cell into account in the line integral the averaged value is used. Equations (2.25) and (2.26) willhence become

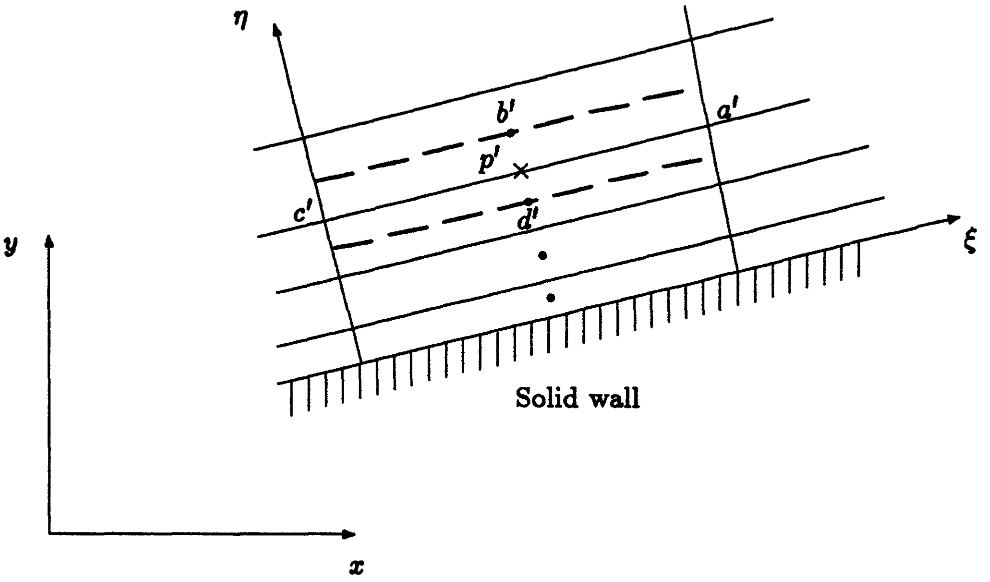

Figure 2.3: Boundary layer cells with large aspect ratios, i.e. $a', c' \ll b', d'$

$$(\varphi_x)_{p'} = \frac{2}{A_{ij} + A_{ij+1}}(-\varphi_b + \varphi_d)\,\Delta y_{p'} \tag{2.27}$$

$$(\varphi_y)_{p'} = \frac{2}{A_{ij} + A_{ij+1}}(\varphi_b - \varphi_d)\,\Delta x_{p'} \tag{2.28}$$

Using these expressions for the gradients one can observe that the four viscous terms in Eq. (2.22) are functions only of their two closest mesh points and therefore the following holds:

$$\begin{aligned}
F_b^v &= F_b^v(q_{ij+1}, q_{ij}) \\
F_d^v &= F_d^v(q_{ij}, q_{ij-1}) \\
G_b^v &= G_b^v(q_{ij+1}, q_{ij}) \\
G_d^v &= G_d^v(q_{ij}, q_{ij-1})
\end{aligned}$$

Using the notations in figure 2.4, an explicit expression of the discretized viscous flux-vector components can be derived. In the figure the indices N and S indicates the state vector values north and south of the centerline C. The averaging procedure is performed so that the viscous flux vectors are evaluated at point p on the centerline C.

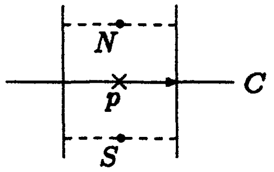

Figure 2.4: The face between two cells where the viscous flux vectors are evaluated (point p).

Using Equations (2.27) and (2.28) the F^v terms at interface C can be written as

$$F_1^v(N,S) = 0 \tag{2.29}$$

$$F_2^v(N,S) = \left(\frac{\mu_N + \mu_S}{A_N + A_S}\right) \frac{2}{3} (2(-u_N + u_S)\,\Delta y_p - (v_N - v_S)\,\Delta x_p) \tag{2.30}$$

$$F_3^v(N,S) = \left(\frac{\mu_N + \mu_S}{A_N + A_S}\right) ((-v_N + v_S)\,\Delta y_p + (u_N - u_S)\,\Delta x_N) \tag{2.31}$$

$$\begin{aligned} F_4^v(N,S) &= \left(\frac{\mu_N + \mu_S}{A_N + A_S}\right) \left[\left(\frac{v_N + v_S}{2}\right) ((-v_N + v_S)\,\Delta y_p + (u_N - u_S)\,\Delta x_p)\right. \\ &+ \frac{2}{3}\left(\frac{u_N + u_S}{2}\right) (2(-u_N + u_S)\,\Delta y_p - (v_N - v_S)\,\Delta x_p) \\ &+ \left.\frac{1}{(\gamma - 1)\mathrm{Pr}}(-T_N + T_S)\,\Delta x_p\right] \end{aligned} \tag{2.32}$$

Similarly the viscous G^v terms can be written as:

$$G_1^v(N,S) = 0 \tag{2.33}$$

$$G_2^v(N,S) = \left(\frac{\mu_N + \mu_S}{A_N + A_S}\right) ((-v_N + v_S)\,\Delta y_p + (u_N - u_S)\,\Delta x_p) \tag{2.34}$$

$$G_3^v(N,S) = \left(\frac{\mu_N + \mu_S}{A_N + A_S}\right) \frac{2}{3} (2(v_N - v_S)\,\Delta x_p - (-u_N + u_S)\,\Delta y_p) \tag{2.35}$$

$$\begin{aligned} G_4^v(N,S) &= \left(\frac{\mu_N + \mu_S}{A_N + A_S}\right) \left[\left(\frac{u_N + u_S}{2}\right) ((-v_N + v_S)\,\Delta y_p + (u_N - u_S)\,\Delta x_p)\right. \\ &+ \frac{2}{3}\left(\frac{v_N + v_S}{2}\right) (2(v_N - v_S)\,\Delta x_p - (-u_N + u_S)\,\Delta y_p) \\ &+ \left.\frac{1}{(\gamma - 1)\mathrm{Pr}}(T_N - T_S)\,\Delta x_p\right] \end{aligned} \tag{2.36}$$

Here A_N is the area of the north cell and A_S is the area of the south cell. These expressions are, however, only valid for interior points. At the boundaries one has to apply the methods described in section 2.6.

2.4 Artificial Dissipation

It is well known that the cell centered scheme Equation (2.22) is by itself nondissipative and hence not stable upon integration in time. To obtain stability and uniqueness some extra dissipation has to be added. The fourth order dissipative terms given by Jameson [16] and Eriksson [10] will be used. With the dissipative terms added to the fluxes in Eq. (2.22) the semi-discrete formulation becomes

$$\begin{aligned}\left(\frac{dq}{dt}\right)_{ij} A_{ij} &= \frac{1}{2}[-(F_{i+1j}+F_{ij})\Delta y_a - (F_{i-1j}+F_{ij})\Delta y_c \\ &\quad +(G_{i+1j}+G_{ij})\Delta x_a + (G_{i-1j}+G_{ij})\Delta x_c] \\ &+ \frac{1}{2}[-(F_{ij+1}+F_{ij})\Delta y_b - (F_{ij-1}+F_{ij})\Delta y_d \\ &\quad +(G_{ij+1}+G_{ij})\Delta x_b + (G_{ij-1}+G_{ij})\Delta x_d] + D_{ij} \\ &= -\mathbf{R}_{ij} \qquad (2.37)\end{aligned}$$

where D_{ij} denotes the dissipative terms and $\mathbf{R}_{ij}$ the residual. The dissipation factor $D_{ij}(q)$ has the form

$$D_{ij}(q) = d_{i+\frac{1}{2},j} - d_{i-\frac{1}{2},j} + d_{i,j+\frac{1}{2}} - d_{i,j+\frac{1}{2}} \qquad (2.38)$$

where

$$d_{i+\frac{1}{2},j} = \epsilon^{(4)}_{i+\frac{1}{2},j}(q_{i+2j} - 3q_{i+1j} + 3q_{ij} - q_{i-1j}) \qquad (2.39)$$

$$d_{i,j+\frac{1}{2}} = \epsilon^{(4)}_{i,j+\frac{1}{2}}(q_{ij+2} - 3q_{ij+1} + 3q_{ij} - q_{ij-1}) \qquad (2.40)$$

The coefficients $\epsilon^{(4)}_{i+\frac{1}{2},j}$ and $\epsilon^{(4)}_{i,j+\frac{1}{2}}$ are normalized by a factor proportional to the size of the cell. This fourth order dissipation provide background smoothing of the solution throughout the domain. The smoothing term acts as a global filter that suppresses spurious saw-tooth error waves.

It was pointed out by Agarwal and Deese [1] that the added dissipation in the body normal direction tends to mask the real viscosity, especially if a turbulence model is used. Following their concept the coefficient $\epsilon^{(4)}$ in the normal direction were set to zero in the close neighborhood of the blade. For the boundary cells an alternate formula is used instead of the one described above. The theory due to Eriksson [11] based on energy-decreasing boundary damping terms is used. In this theory the i-directional damping terms at $i = 1$ and $i = 2$ are the same as in Eq. (2.38) and the d's are modified to

$$d_{\frac{1}{2},j} = 0 \qquad (2.41)$$

$$d_{\frac{3}{2},j} = \epsilon^{(4)}_{\frac{3}{2},j} (q_{3j} - 2q_{2j} + q_{1j}) \qquad (2.42)$$

The same holds for the terms at the other boundaries.

2.5 Boundary Conditions

For the TSL code described here applied to channel flows there are two kind of boundary conditions. These are solid wall boundary conditions and inflow/outflow boundary conditions. Later on when connecting the TSL code to the explicit Euler code the inflow/outflow boundary conditions will be replaced by interface boundary conditions. Which will be described in detail in chapter 5.

2.5.1 Solid Wall Boundary

The no-slip condition on a solid, impermeable wall requires

$$u_w = v_w = 0 \qquad (2.43)$$

where the subscript $_w$ denotes the wall value. The temperature boundary condition can be of two different types. It can be either a specified wall temperature as a function of the arc length

$$T_w = T_w(l) \qquad (2.44)$$

or a zero heat-flux condition for an adiabatic wall

$$\left(\frac{\partial T}{\partial n}\right)_w = 0 \tag{2.45}$$

where n denotes the wall-normal.

2.5.2 Inflow/outflow Boundary condition

The test cases run in section 2.11 require boundary conditions on the inflow and outflow boundaries respectively. These are not physical boundary conditions but just imposed mathematical boundaries to restrict the computational area. The following choice of boundary conditions is consistent with the hyperbolic system eq. (2.1). At the upstream boundary (inflow) the stagnation pressure P_0, stagnation temperature T_0 and the flow angle α are specified and the static pressure p is extrapolated from the interior. At the downstream boundary (outflow) the static pressure p is specified and the other flow variables are extrapolated from the interior.

2.6 Numerical Implementation of Boundary Conditions

2.6.1 Solid wall

The contribution to the line integrals in eq. (2.13) at the solid wall can be written as

$$I(q) = \int_{wall} \left(\left[F - \frac{1}{\mathrm{Re}_c} F^v, G - \frac{1}{\mathrm{Re}_c} G^v \right] \cdot \bar{n} \right) ds \tag{2.46}$$

where $\bar{n}$ is the normal at the solid wall. If the dot product is carried out this integral reads

$$I(q) = \int_{wall} \left(\left[\begin{array}{c} 0 \\ pn_x + \frac{1}{\mathrm{Re}_c} \left(\mu \frac{2}{3} (2u_x - v_y) n_x + \mu (u_y + v_x) n_y \right) \\ pn_y + \frac{1}{\mathrm{Re}_c} \left(\mu (u_y + v_x) n_x + \mu \frac{2}{3} (2v_y - u_x) n_y \right) \\ \frac{\mu}{(\gamma - 1)\mathrm{Pr}} \frac{\partial T}{\partial n} \end{array} \right] \right) ds \tag{2.47}$$

Using the finite volume approximation eq. (2.22) that assumes that the variables are piecewise constant on each finite segment along the boundary the integral can be approximated by

$$I(q) \simeq \left[\begin{array}{c} 0 \\ p n_x + \frac{1}{\mathrm{Re}_c} \left(\mu \frac{2}{3}(2u_x - v_y) n_x + \mu (u_y + v_x) n_y \right) \\ p n_y + \frac{1}{\mathrm{Re}_c} \left(\mu (u_y + v_x) n_x + \mu \frac{2}{3}(2v_y - u_x) n_y \right) \\ \frac{\mu}{(\gamma-1)\mathrm{Pr}} \frac{\partial T}{\partial n} \end{array} \right]_{wall} \Delta s \qquad (2.48)$$

This indicates that we have to evaluate the pressure, velocity gradients, viscosity coefficient and temperature gradients at the wall. The wall pressure is obtained by assuming the boundary-layer approximation for the pressure to hold on the body contour. That is, the normal pressure gradient is assumed to be zero at the wall, and hence the wall pressure is taken to be equal to the pressure in the first cell.

$$p_w = p_{k,1} \qquad (2.49)$$

were $k, 1$ indicates the center of the solid computational cell showed in Fig. 2.5, indicated by a dot. The velocity gradients are obtained by linearly extrapolating the gradients in the two cells closest to the solid wall. In figure 2.5 these two cells are indicated as dashed cells with index $i, 1$ and $i, 2$. By using eq. (2.23) on these two cells , with $u_w = v_w = 0$, two values on the gradient are obtained. Finally the value on the wall is computed by

$$\nabla \varphi_w = \frac{4}{3} \nabla \varphi_{i,1} - \frac{1}{3} \nabla \varphi_{i,2} \qquad (2.50)$$

The temperature boundary condition can either be a specified wall temperature or an adiabatic wall. In the first case there are no problems. The temperature gradient is calculated in the same way as the velocity gradient. Since the temperature is specified at the wall the viscosity coefficient μ is given by Sutherlands formula using the specified wall temperature. If the adiabatic wall condition is used, the temperature gradient is by definition equal to zero, and the wall temperature used in Sutherlands formula for μ is simply taken to be the temperature in the first cell

$$T_w = T_{i,1} \qquad (2.51)$$

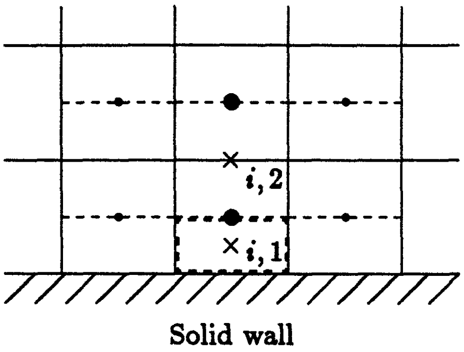

Figure 2.5: Identifying viscous cells in the computational mesh close to a solid wall

2.6.2 Inlet/outlet

At the far field boundary the extrapolated flow variables are given by

$$\varphi_{face} = \frac{3}{2}\varphi_{i/o} - \frac{1}{2}\varphi_{i/o-1} \tag{2.52}$$

where the values i/o are taken at the cell closest to the boundary and $i/o-1$ at the cell next closest to the boundary. At the inlet the state vector components are determined from:

$$\begin{aligned} p_{in} &= p_{extr} && (2.53)\\ \rho_{in} &= \left(\frac{\gamma P_0}{(\gamma-1)H_0}\right)\left(\frac{p_{in}}{P_0}\right)^{\frac{1}{\gamma}} && (2.54)\\ (\rho u)_{in} &= \rho_{in}\sqrt{2\left(H_0 - \left(\frac{\gamma p_{in}}{(\gamma-1)\rho_{in}}\right)\right)}\cos(\alpha) && (2.55)\\ (\rho v)_{in} &= (\rho u)_{in}\tan(\alpha) && (2.56)\\ e_{in} &= \frac{p_{in}}{\gamma-1} + \frac{1}{2}\left[(\rho u)^2_{in} + (\rho v)^2_{in}\right]\frac{1}{\rho_{in}} && (2.57) \end{aligned}$$

where p_{extr} is the static pressure obtained from the above described extrapolation formula, P_0 is the inlet total pressure, H_0 is the inlet total enthalpy and α is the inlet flow angle.

At the outlet the state vector components are determined from:

$$p_{out} = p_{spec} \tag{2.58}$$

$$\rho_{out} = \rho_{extr} \tag{2.59}$$

$$(\rho u)_{out} = (\rho u)_{extr} \tag{2.60}$$

$$(\rho v)_{out} = (\rho v)_{extr} \tag{2.61}$$

$$e_{out} = \frac{p_{out}}{\gamma - 1} + \frac{1}{2}\left[(\rho u)^2_{out} + (\rho v)^2_{out}\right]\frac{1}{\rho_{out}} \tag{2.62}$$

2.7 Time Integration

Since the TSL equation governs a viscous flow it is necessary to accurately resolve the boundary layer. This calls for clustering of grid points in the normal direction near the solid wall, and will inevitably lead to mesh cells with large aspect ratios. In performing time accurate calculations on meshes with large differences in cell size, it is not advisable to use an explicit scheme since the size of the cell governs the stability criterion. An implicit scheme, however, in general does not have the same stability restriction as an explicit scheme. Hence, even if an implicit scheme is more computationally expensive it is preferable to the explicit scheme.

It was established in Section 2.3, Eq. (2.22) that the semi-discretized Navier-Stokes equations could be written as

$$\left(\frac{dq}{dt}\right)_{ij} = -\mathbf{R}(q_{i-1j}, q_{ij-1}, q_{ij}, q_{i+1j}, q_{ij+1}) \tag{2.63}$$

A solution to this equation can be found by integrating in time. In this section a numerical time-integration scheme for solving eq. (2.63) will be derived. Consider the discretization of the time derivative given by

$$\frac{dq^{n+1}}{dt} = \frac{1+\psi}{\Delta t}\Delta q^n - \frac{\psi}{\Delta t}\Delta q^{n-1} \tag{2.64}$$

Combining equations (2.63) and (2.64) gives

$$\frac{1+\psi}{\Delta t}\Delta q^n - \frac{\psi}{\Delta t}\Delta q^{n-1} = -\mathbf{R}^{n+1} \tag{2.65}$$

Since the left hand side is at time level $n+1$ this is an implicit scheme, and can be shown to be identified as

- The 1st order Backward Euler formula for $\psi = 0$
- The 2st order Backwards difference formula for $\psi = \frac{1}{2}$

Since the term $\mathbf{R}^{n+1}$ is non-linear, it must be approximated by a Taylor series expansion:

$$\mathbf{R}^{n+1} \simeq \mathbf{R}^n + \left(\frac{\partial \mathbf{R}}{\partial q}\right)^n \Delta q^n \tag{2.66}$$

where $\frac{\partial \mathbf{R}}{\partial q}$ is the Jacobian matrix and $\Delta q^n = q^{n+1} - q^n$. Inserted in Eq. (2.65) and in an abbreviated form the scheme will look like:

$$\left(\mathbf{I} + \frac{\Delta t}{\psi + 1}\left(\frac{\partial \mathbf{R}}{\partial q}\right)^n\right)\Delta q^n = -\frac{\Delta t}{\psi + 1}\mathbf{R}^n + \frac{\psi}{\psi + 1}\Delta q^{n-1} \tag{2.67}$$

The Jacobian matrix will split up into six different terms to become:

$$\begin{aligned} \left(\frac{\partial \mathbf{R}}{\partial q}\right)^n \Delta q &= \left(\frac{\partial \mathbf{R}}{\partial q_{i-1j}}\right)^n \Delta q_{i-1j} + \left(\frac{\partial \mathbf{R}}{\partial q_{ij}}\right)^n_{a,c} \Delta q_{ij} + \left(\frac{\partial \mathbf{R}}{\partial q_{i+1j}}\right)^n \Delta q_{i+1j} \\ &+ \left(\frac{\partial \mathbf{R}}{\partial q_{ij-1}}\right)^n \Delta q_{ij-1} + \left(\frac{\partial \mathbf{R}}{\partial q_{ij}}\right)^n_{b,d} \Delta q_{ij} + \left(\frac{\partial \mathbf{R}}{\partial q_{ij+1}}\right)^n \Delta q_{ij+1} (2.68) \end{aligned}$$

Using the explicit expression of the right hand side eq. (2.22), the terms in the split Jacobian becomes

$$\begin{aligned} \left(\frac{\partial \mathbf{R}}{\partial q_{i-1j}}\right) &= -\frac{1}{2A_{ij}}\left(\frac{\partial F}{\partial q}\Delta y_c - \frac{\partial G}{\partial q}\Delta x_c\right)_{i-1j} \\ \left(\frac{\partial \mathbf{R}}{\partial q_{i+1j}}\right) &= -\frac{1}{2A_{ij}}\left(\frac{\partial F}{\partial q}\Delta y_a - \frac{\partial G}{\partial q}\Delta x_a\right)_{i+1j} \\ \left(\frac{\partial \mathbf{R}}{\partial q_{ij-1}}\right) &= -\frac{1}{2A_{ij}}\left(\frac{\partial F}{\partial q}\Delta y_d - \frac{\partial G}{\partial q}\Delta x_d\right)_{ij-1} - \frac{1}{A_{ij}}\left(-\frac{\partial F^v_d}{\partial q_{ij-1}}\Delta y_d + \frac{\partial G^v_d}{\partial q_{ij-1}}\Delta x_d\right) \\ \left(\frac{\partial \mathbf{R}}{\partial q_{ij+1}}\right) &= -\frac{1}{2A_{ij}}\left(\frac{\partial F}{\partial q}\Delta y_b - \frac{\partial G}{\partial q}\Delta x_b\right)_{ij+1} - \frac{1}{A_{ij}}\left(-\frac{\partial F^v_b}{\partial q_{ij+1}}\Delta y_b + \frac{\partial G^v_b}{\partial q_{ij+1}}\Delta x_b\right) \\ \left(\frac{\partial \mathbf{R}}{\partial q_{ij}}\right)_{ac} &= -\frac{1}{2A_{ij}}\left(\frac{\partial F}{\partial q}(\Delta y_a + \Delta y_c) - \frac{\partial G}{\partial q}(\Delta x_a + \Delta x_c)\right)_{ij} \\ \left(\frac{\partial \mathbf{R}}{\partial q_{ij}}\right)_{bd} &= -\frac{1}{2A_{ij}}\left(\frac{\partial F}{\partial q}(\Delta y_b + \Delta y_d) - \frac{\partial G}{\partial q}(\Delta x_b + \Delta x_d)\right)_{ij} \end{aligned}$$

$$- \quad \frac{1}{A_{ij}}\left(-\frac{\partial F^v_d}{\partial q_{ij}}\Delta y_d + \frac{\partial G^v_d}{\partial q_{ij}}\Delta x_d - \frac{\partial F^v_b}{\partial q_{ij}}\Delta y_b + \frac{\partial G^v_b}{\partial q_{ij}}\Delta x_b\right) \tag{2.69}$$

When inserting eq. (2.68) in eq. (2.67) the time integration becomes:

$$\begin{aligned}\Delta q^n_{ij} \quad &+ \quad \frac{\Delta t}{\psi+1}\left[\left(\frac{\partial \mathbf{R}}{\partial q_{i-1j}}\right)^n \Delta q_{i-1j} + \left(\frac{\partial \mathbf{R}}{\partial q_{ij}}\right)^n_{a,c} \Delta q_{ij} + \left(\frac{\partial \mathbf{R}}{\partial q_{i+1j}}\right)^n \Delta q_{i+1j}\right] \\ &+ \quad \frac{\Delta t}{\psi+1}\left[\left(\frac{\partial \mathbf{R}}{\partial q_{ij-1}}\right)^n \Delta q_{ij-1} + \left(\frac{\partial \mathbf{R}}{\partial q_{ij}}\right)^n_{b,d} \Delta q_{ij} + \left(\frac{\partial \mathbf{R}}{\partial q_{ij+1}}\right)^n \Delta q_{ij+1}\right] \\ &= \quad -\frac{\Delta t}{\psi+1}\mathbf{R}^n + \frac{\psi}{\psi+1}\Delta q^{n-1} \end{aligned} \tag{2.70}$$

Abbreviated, this becomes:

$$\left(\mathbf{I} + \frac{\Delta t}{\psi+1}\delta_i\mathbf{A} + \frac{\Delta t}{\psi+1}\delta_j\mathbf{B}\right)\Delta q = -\frac{\Delta t}{\psi+1}\mathbf{R}^n + \frac{\psi}{\psi+1}\Delta q^{n-1} \tag{2.71}$$

Using the approximate factorization suggested by Beam and Warming [3] this scheme becomes

$$\left(\mathbf{I} + \frac{\Delta t}{\psi+1}\delta_j\mathbf{B}\right)\left(\mathbf{I} + \frac{\Delta t}{\psi+1}\delta_i\mathbf{A}\right)\Delta q = -\frac{\Delta t}{\psi+1}\mathbf{R}^n + \frac{\psi}{\psi+1}\Delta q^{n-1} \tag{2.72}$$

These equations are solved in three steps

- Step 1: Calculate $\Delta\tilde{q}$ from $\left(\mathbf{I}+\frac{\Delta t}{\psi+1}\delta_i\mathbf{B}\right)\Delta\tilde{q}=-\frac{\Delta t}{\psi+1}\mathbf{R}^n+\frac{\psi}{\psi+1}\Delta q^{n-1}$.
- Step 2: Calculate Δq from $\left(\mathbf{I}+\frac{\Delta t}{\psi+1}\delta_j\mathbf{A}\right)\Delta q=\Delta\tilde{q}$.
- Step 3: Update solution by $q^{n+1}=q^n+\Delta q$.

This scheme was originally suggested and developed by Beam and Warming [3] and is often referred to as the Beam-Warming scheme. A rigorous accuracy and stability analysis of this scheme is given in Section 2.10.

2.8 Linearization of Flux Vectors

The Taylor expansion eq. (2.66) yielded a Jacobian matrix. Equation (2.69) gives the different terms that the Jacobian consists of. It is easily seen that four Jacobians are needed. These are: $\frac{\partial F}{\partial q}$, $\frac{\partial G}{\partial q}$, $\frac{\partial F^v}{\partial q}$ and $\frac{\partial G^v}{\partial q}$. The two first are the derivatives of the inviscid flux-vectors. These are obtained by using the chain rule

$$\frac{\partial F_i}{\partial q_j}=\frac{\partial F_i}{\partial v_k}\frac{\partial v_k}{\partial q_j}\quad 1\leq i,j,k\leq 4 \tag{2.73}$$

and

$$\frac{\partial G_i}{\partial q_j}=\frac{\partial G_i}{\partial v_k}\frac{\partial v_k}{\partial q_j}\quad 1\leq i,j,k\leq 4 \tag{2.74}$$

where the F and G are the flux vectors, q is the state vector and $\vec{v}$ is an intermediate vector defined as:

$$\vec{v}=\begin{bmatrix}\rho\\ u\\ v\\ p\end{bmatrix}=\begin{bmatrix}\rho\\ \frac{\rho u}{\rho}\\ \frac{\rho v}{\rho}\\ (\gamma-1)(e-\frac{1}{2}\frac{(\rho u)^2+(\rho v)^2}{\rho})\end{bmatrix} \tag{2.75}$$

The viscous terms are handled in a similar manner. Here, however, the intermediate vector $\vec{v}$ is defined as:

$$\vec{v}=\begin{bmatrix}\mu\\ u\\ v\\ T\end{bmatrix}=\begin{bmatrix}T^{\frac{3}{2}}\left(\frac{1+\tilde{S}_k}{T+\tilde{S}_k}\right)\\ \frac{\rho u}{\rho}\\ \frac{\rho v}{\rho}\\ \gamma(\gamma-1)(\frac{e}{\rho}-\frac{1}{2}\frac{(\rho u)^2+(\rho v)^2}{\rho^2})\end{bmatrix} \tag{2.76}$$

It is convenient to introduce an intermediate vector to make it easier to work out the complicated chain rules involved. Considering the discretized viscous terms it can be seen that the important flow and thermodynamic variables in the viscous terms is the temperature T, the x- and y-velocities u and v and the coefficient of viscosity μ and therefore a natural choice of intermediate vector would be:

$$\begin{bmatrix} T \\ u \\ v \\ \mu \end{bmatrix} \tag{2.77}$$

Since μ is dependent of the temperature T through Sutherland formula it is convenient to calculate $\frac{\partial \mu}{\partial T}$ as a middle step. This will give

$$\frac{\partial F_i^v}{\partial \mu}\frac{\partial \mu}{\partial q_j} = \frac{\partial F_i^v}{\partial \mu}\frac{\partial \mu}{\partial T}\frac{\partial T}{\partial q_j} \tag{2.78}$$

First define the temperature as a function of the state vector variables:

$$T = \gamma(\gamma - 1)\left(\frac{e}{\rho} - \frac{1}{2}\frac{(\rho u)^2 + (\rho v)^2}{\rho^2}\right) \tag{2.79}$$

which gives us the first partial derivatives needed:

$$\begin{aligned}
\frac{\partial T}{\partial \rho} &= -\gamma(\gamma - 1)\left(\frac{e}{\rho^2} - \frac{(\rho u)^2 + (\rho v)^2}{\rho^3}\right) \\
\frac{\partial T}{\partial \rho u} &= -\gamma(\gamma - 1)\frac{\rho u}{\rho^2} \\
\frac{\partial T}{\partial \rho v} &= -\gamma(\gamma - 1)\frac{\rho v}{\rho^2} \\
\frac{\partial T}{\partial e} &= \gamma(\gamma - 1)\frac{1}{\rho}
\end{aligned} \tag{2.80}$$

Next intermediate variable is u or in terms of the state vector $\frac{\rho u}{\rho}$. The derivative of this with respect to the state vector becomes:

$$\begin{aligned}
\frac{\partial u}{\partial \rho} &= -\frac{\rho u}{\rho^2} \\
\frac{\partial u}{\partial \rho u} &= \frac{1}{\rho} \\
\frac{\partial u}{\partial \rho v} &= 0 \\
\frac{\partial u}{\partial e} &= 0
\end{aligned} \tag{2.81}$$

and similarly for the v variable:

$$\begin{aligned}\frac{\partial v}{\partial \rho} &= -\frac{\rho v}{\rho^2} \\ \frac{\partial v}{\partial \rho u} &= 0 \\ \frac{\partial v}{\partial \rho v} &= \frac{1}{\rho} \\ \frac{\partial v}{\partial e} &= 0 \end{aligned} \tag{2.82}$$

Sutherland's law gives μ as a function of the temperature T see Eq. (2.9) . The derivative of μ w.r.t. T can be written:

$$\frac{\partial \mu}{\partial T} = \mu \left(\frac{2}{3}\frac{1}{T} - \frac{1}{(T - \tilde{S}_k)^2} \right) \tag{2.83}$$

Using this derivative and the previous information about the derivative of T the derivative of μ w.r.t. the state vector becomes:

$$\begin{aligned}\frac{\partial \mu}{\partial \rho} &= \frac{\partial \mu}{\partial T}\frac{\partial T}{\partial \rho} \\ \frac{\partial \mu}{\partial \rho u} &= \frac{\partial \mu}{\partial T}\frac{\partial T}{\partial \rho u} \\ \frac{\partial \mu}{\partial \rho v} &= \frac{\partial \mu}{\partial T}\frac{\partial T}{\partial \rho v} \\ \frac{\partial \mu}{\partial e} &= \frac{\partial \mu}{\partial T}\frac{\partial T}{\partial e} \end{aligned} \tag{2.84}$$

These expressions can now be used in equations (2.73) and (2.74).

2.9 Turbulence Model

It was noticed that in the case of a laminar flow solution around a turbine blade, a large non-physical separation bubble appeared on the suction surface of a turbine blade. To be able to suppress this separation the flow has to become turbulent. However, the method described above is unable to predict turbulent flow and hence a turbulence model has to be introduced. The relatively simple two-layer algebraic model described by Baldwin and Lomax [2] is used. The simplicity lays in the fact that it does not add any new equations to the original Navier-Stokes equations and one does not have to find the edge of the boundary layer.

Following the idea of Baldwin and Lomax the turbulence is simulated by a eddy viscosity coefficient μ_t that is added to the molecular viscosity μ. The Reynolds heat flux terms are approximated using the constant Prandtl number assumption. The model gives the following expression for the viscosity coefficient μ and heat flux coefficient κ

$$\mu = \mu_l + \mu_t \tag{2.85}$$

$$\kappa = c_p\left[\left(\frac{\mu}{\Pr}\right)_l + \left(\frac{\mu}{\Pr}\right)_t\right] \tag{2.86}$$

where the subscripts $_l$ and $_t$ refers to laminar and turbulent quantities. The main idea of the two layer model is that the eddy viscosity μ_t is determined by

$$\mu_t = \left[\begin{array}{ll} (\mu_t)_{\text{inner}}, & y \leq y_{\text{crossover}} \\ (\mu_t)_{\text{outer}}, & y > y_{\text{crossover}} \end{array}\right. \tag{2.87}$$

where y is the distance from the wall and $y_{\text{crossover}}$ is the smallest value on y where the both expressions are equal. The inner viscous term is determined using the Prandtl-Van Driest formulation and the outer decaying viscous term is governed by the Klebanoff intermittency factor. A fully detailed and easy to read description of the model, with given values on various constants, can be found in reference [2].

2.10 Accuracy and Stability

2.10.1 Spatial Part

On an equidistant mesh (ξ, η) with grid spacing $\Delta\xi$ and $\Delta\eta$ in each direction, one can show that the finite volume formulation is equivalent to the central finite difference formulation. This is convenient since both accuracy and stability analysis are usually performed using the finite difference formulation of operators such as the average and difference operators. To prove the equivalence between the two methods consider the

Euler equation in differential form

$$\frac{\partial q}{\partial t} + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} = 0 \tag{2.88}$$

The semi-discrete formulation of this equation using finite differences can be written as

$$\frac{dq_{ij}}{dt} + \frac{1}{\Delta\xi}\mu_\xi^+\delta_\xi^- F(q_{ij}) + \frac{1}{\Delta\eta}\mu_\eta^+\delta_\eta^- G(q_{ij}) = 0 \tag{2.89}$$

where the standard notation δ^- for backwards difference operator and μ^+ for forward average operator is used. These operators applied to some function φ_{ij} are defined as

$$\begin{aligned}
\delta_\xi^-\varphi_{ij} &= (\varphi_{ij} - \varphi_{i-1j}) \\
\mu_\xi^+\varphi_{ij} &= \frac{1}{2}(\varphi_{i+1j} + \varphi_{ij}) \\
\delta_\eta^-\varphi_{ij} &= (\varphi_{ij} - \varphi_{ij-1}) \\
\mu_\eta^+\varphi_{ij} &= \frac{1}{2}(\varphi_{ij+1} + \varphi_{ij})
\end{aligned} \tag{2.90}$$

In the case of the finite volume formulation, consider the inviscid form of the operator given in Equation (2.22) ($F^v = G^v = 0$) given on an equidistant, orthogonal (ξ, η) mesh. Note that on this mesh $A_{ij} = \Delta\xi\,\Delta\eta$. The following identity can be observed

$$\begin{aligned}
& \frac{1}{\Delta\xi\,\Delta\eta}\left(\frac{1}{2}(F(q_{i+1j}) - F(q_{i-1j}))\,\Delta\eta + \frac{1}{2}(G(q_{ij+1}) - G(q_{ij-1}))\,\Delta\xi\right) \\
= \; & \frac{1}{\Delta\xi}\mu_\xi^+\delta_\xi^- F(q_{ij}) + \frac{1}{\Delta\eta}\mu_\eta^+\delta_\eta^- G(q_{ij}) \quad \square
\end{aligned}$$

which shows that the finite volume operator is identical to the central finite difference operator.

To study the accuracy of the spatial finite volume operator, a simplified model equation will be used. In a general curvilinear coordinate system (ξ, η) the model equation is defined as

$$\frac{\partial u}{\partial t} + \lambda_\xi\frac{\partial u}{\partial \xi} + \lambda_\eta\frac{\partial u}{\partial \eta} = \nu_\eta\frac{\partial^2 u}{\partial \eta^2} \tag{2.91}$$

where

$$\lambda_\xi, \lambda_\eta \in \mathbf{R}, \quad \nu_\eta \geq 0 \tag{2.92}$$

This scalar, model equation is closely related to the linearized momentum and energy part of the TSL equation. If the coefficient $\nu_\eta = 0$, then the equivalent Euler equation is recovered. Since it was shown above that finite volume and finite difference formulations were identical following discretization will be used

$$\begin{aligned}
\frac{\partial u_{ij}}{\partial \xi} &= \frac{1}{\Delta\xi}\mu_\xi^+\delta_\xi^- u_{ij} \;+\; O(\Delta\xi^2) \\
\frac{\partial u_{ij}}{\partial \eta} &= \frac{1}{\Delta\eta}\mu_\eta^+\delta_\eta^- u_{ij} \;+\; O(\Delta\eta^2) \\
\frac{\partial^2 u_{ij}}{\partial \eta^2} &= \frac{1}{\Delta\eta^2}\delta_\eta^+\delta_\eta^- u_{ij} \;+\; O(\Delta\eta^2)
\end{aligned} \tag{2.93}$$

Inserted in the model equation will give the semi discrete equation

$$\frac{\partial u_{ij}}{\partial t} + \left[\frac{\lambda_\xi}{\Delta\xi}\mu_\xi^+\delta_\xi^- + \frac{\lambda_\eta}{\Delta\eta}\mu_\eta^+\delta_\eta^- - \frac{\nu_\eta}{\Delta\eta^2}\delta_\eta^+\delta_\eta^-\right] u_{ij} = 0 \tag{2.94}$$

Using a straight forward Taylor expansion will finally give following expression

$$\frac{\partial u}{\partial t} + \lambda_\xi\frac{\partial u}{\partial \xi} + \lambda_\eta\frac{\partial u}{\partial \eta} - \nu_\eta\frac{\partial^2 u}{\partial \eta^2} \;=\; O(\Delta\xi^2, \Delta\eta^2) \tag{2.95}$$

where the right hand side is the size of the truncation error which indicates that the spatial finite volume discretization is of global second order accuracy.

2.10.2 Temporal Part

The implicit time integration algorithm (Eq. (2.65)) introduced in Section 2.6 can be identified as a linear multi-step differentiation formula. The original formula is

$$\frac{1+\psi}{\Delta t}\Delta q^n \;-\; \frac{\psi}{\Delta t}\Delta q^{n-1} \;=\; -\mathbf{R}^{n+1} \tag{2.96}$$

Rewriting this formula using the standard notation $k = \Delta t$ will give

$$q^{n+1} - \left(\frac{1+2\psi}{1+\psi}\right) q^n + \left(\frac{\psi}{1+\psi}\right) q^{n-1} \;=\; -k\left(\frac{1}{1+\psi}\right)\mathbf{R}^{n+1} \tag{2.97}$$

which is a standard form of writing linear multistep formulas. The characteristic polynomials for this specific formula are

$$\rho(z) \;=\; z^2 + \left(\frac{1+2\psi}{1+\psi}\right) z + \left(\frac{\psi}{1+\psi}\right) \tag{2.98}$$

$$\sigma(z) \;=\; \left(\frac{1}{1+\psi}\right) z^2 \tag{2.99}$$

Accuracy

The (p) Padé approximant to $\log z$ at $z = 1$ is defined as the unique rational function $\rho(z)/\sigma(z)$ that satisfies

$$\begin{aligned}\frac{\rho(z)}{\sigma(z)} &= \log z + O\left((z-1)^{p+1}\right) \\ &= \left[(z-1) - \frac{1}{2}(z-1)^2 + \frac{1}{3}(z-1)^3 - \ldots\right] + O\left((z-1)^{p+1}\right) \qquad (2.100)\end{aligned}$$

By definition a linear multistep formula has order of accuracy p if and only if its characteristic polynomial satisfies Equation (2.100). For the backward differentiation formula defined above, the ratio of the characteristic polynomial can, after Taylor expanding $\sigma(z)^{-1}$, be written as

$$\frac{\rho(z)}{\sigma(z)} = \left[(z-1) - (\psi - 1)(z-1)^2 + (1-2\psi)(z-1)^3 + O\left((z-1)^{p+1}\right)\right] \qquad (2.101)$$

and hence for $\psi = \frac{1}{2} \Rightarrow p = 2$ gives second order accuracy and for $\psi = 0 \Rightarrow p = 1$ gives first order accuracy.

Stability

The time stability criterion for the second-order backwards differentiation formula is obtained by analyzing the scheme applied to the model equation

$$\frac{dq}{dt} = \alpha q \qquad (2.102)$$

where α is some complex constant. Applied to this equation, the associated stability polynomial $\pi_{\bar{k}}(z)$ for the scheme can be defined as

$$\pi_{\bar{k}}(z) = \rho(z) - \bar{k}\,\sigma(z) \qquad (2.103)$$

where again $\bar{k} = \Delta t\,\alpha$. For the present scheme the stability polynomial will look like

$$\pi_{\bar{k}}(z) = \left(1 - \frac{\bar{k}}{\psi + 1}\right) z^2 - \left(\frac{1+2\psi}{1+\psi}\right) z + \left(\frac{1}{1+\psi}\right) \qquad (2.104)$$

where $\psi = \frac{1}{2}$. According to Theorem 1.9 in [34] a linear multistep formula is time-stable for a particular value $\bar{k} = \Delta t\,\alpha$ if and only if all the roots of $\pi_{\bar{k}}(z)$ satisfy $|z| \leq 1$, and any root with $|z| = 1$ is simple. With $\psi = \frac{1}{2}$ in Equation (2.104) the roots to the stability polynomial are found to be

$$|z| = \frac{1}{2}\left|\frac{2 \pm \sqrt{1+2\bar{k}}}{\frac{3}{2} - \bar{k}}\right| \tag{2.105}$$

By definition the stability region of a linear multistep formula is the set of all $\bar{k} \in \mathbf{C}$ for which the formula is time-stable. Figure 2.6 shows the stability region for the second-order backwards differentiation formula. For each value of $\bar{k}$ the maximum $|z|$ is found and referring to the stability plot the scheme is shown to be at least A-stable i.e. stable in the complete left half plane. The model Equation (2.91) will be used to investigate the stability of the full implicit scheme. The solution to the semi-discrete version of this model equation can be written in terms of its discrete Fourier modes, which inserted in the semi-discrete model equation will give the following equation

$$\frac{dq_{mn}}{dt} = \varphi\, q_{mn} \tag{2.106}$$

where

$$\varphi = -\frac{\nu_\eta}{2\Delta\eta^2}(1 - \cos(\theta_\eta)) - i\left(\frac{\lambda_\xi}{\Delta\xi}\sin(\theta_\xi) + \frac{\lambda_\eta}{\Delta\eta}\sin(\theta_\eta)\right) \tag{2.107}$$

In Figure 2.6 it can be seen that there is no restriction on the imaginary part of φ as long as the real part is < 0. By definition $\nu_\eta \geq 0, \epsilon_\xi, \epsilon_\eta \leq 0$ and therefore φ will always be in the left half plane. Since the time integration scheme it at least A-stable and the eigenvalues are in the left half-plane, the full scheme is unconditionally stable. For the TSL Navier-Stokes equations ν_η can be identified as the coefficient of viscosity in the momentum and energy parts of the equation.

Here it has been shown that the Beam-Warming scheme is unconditionally stable for a linear hyperbolic scalar model equation. In practice, however, the approximate factorization of the scheme and the fact that the Euler equations are non-linear will yield a stability criterion. Depending on mesh geometry and flow conditions this scheme has been shown to have an optimal CFL number ranging from 2 up to 100.

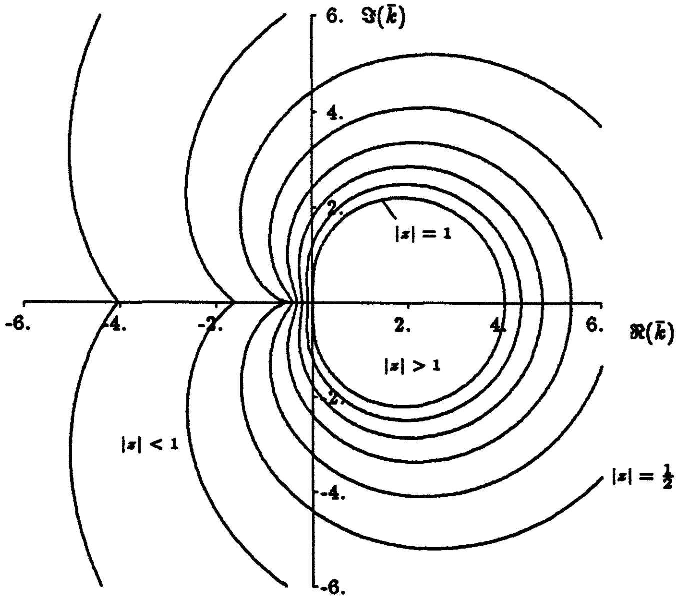

Figure 2.6: Stability region for second order backwards differentiation formula (exterior of innermost closed contour).

2.11 Code Validation

Although based on known methods such as the finite volume technique and the Beam and Warming time integration scheme the present method has been developed entirely from scratch. As always is the case when developing a numerical code a number of test cases must be run to check the accuracy of the code. In order to validate the present viscous code three test cases have been run. The test cases are:

- Subsonic, laminar, low Reynolds number flow over a flat plate.
- Subsonic, turbulent, high Reynolds number flow over a flat plate.
- Subsonic, laminar, low Reynolds number flow over a circular arc bump in a channel.

These three cases should give enough information about the accuracy of the code. For all three cases there are a large number of results available in the literature for comparisons.

Below are presented some of the numerical results obtained from the two different cases.

2.11.1 Flat Plate

The flat plate is a good test that gives information about the accuracy of both laminar and turbulent viscous terms in the code. The numerical results obtained for the laminar case may be compared with the Blasius solution (cf. Schlichting [28]). Although the Blasius solution is obtained under the assumption that the flow is incompressible, the results from a compressible calculation should be in good agreement with the Blasius solution if the free stream Mach number is chosen sufficiently low .

The same as above is true for the turbulent flow over a flat plate. That is, if the Mach number is low enough the results from a compressible computation should agree with the ones obtained with an incompressible assumption. The results can be compared with the results from a power-law solution given in White [35].

Laminar

For the laminar case the following flow conditions were chosen to give approximately incompressible conditions:

$$\begin{aligned} M_{\infty} &= 0.2 \\ \mathrm{Re} &= 10^4 \\ \mathrm{Pr} &= 0.72 \\ \tilde{S}_k &= 0.4 \end{aligned}$$

The plate was also assumed to be adiabatic. The solution was obtained on a 65 × 33 mesh with the upper boundary $2c$ away from the plate which gives a mesh stretching ratio of 1.15. Comparisons between the computed results obtained with the present method, and the Blasius solution taken from [28], is shown in Fig. 2.7 and Fig. 2.8. To avoid influence from the singularity at the leading edge of the plate, comparisons of the velocity profile shown in Fig. 2.7 is taken at 50% of the chord. The numerical result is in very good agreement with the Blasius solution. In Fig. 2.8 the skin friction along the plate is presented, and it shows good agreement with the Blasius solution.

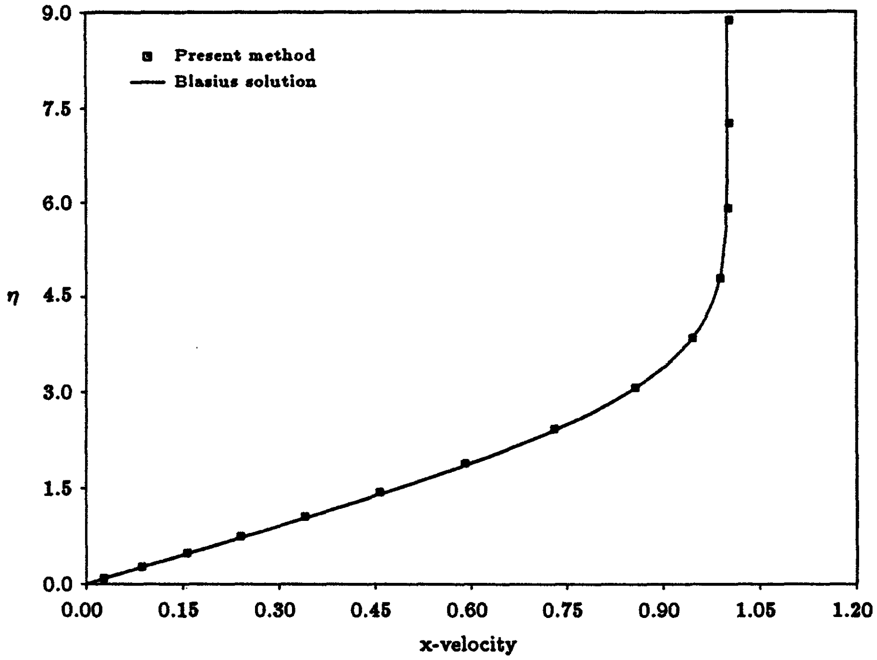

Figure 2.7: Velocity profile at 50% chord of the flat plate

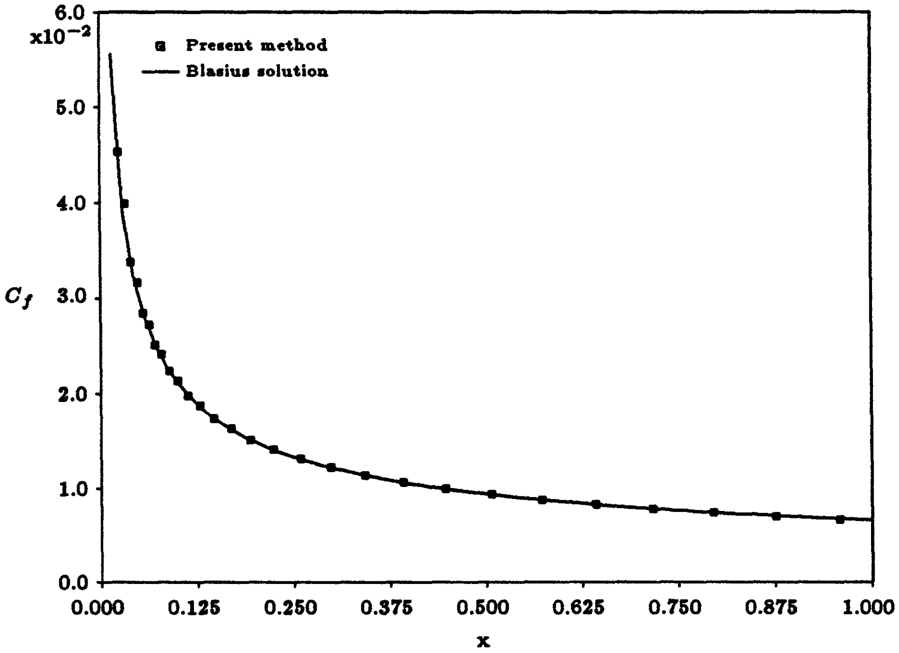

Figure 2.8: Skin friction along the flat plate

Turbulent

For the turbulent case a 65×61 size computational mesh was used where 33 points was used on the flat plate. The mesh stretching in the normal direction was 1.1 and the upper boundary was located 20δ away from the flat plate, where δ was measured at the outflow boundary. The following flow conditions were used to give approximately the same conditions as in an incompressible case.

$$\begin{aligned} M_\infty &= 0.5 \\ \mathrm{Re} &= 10^6 \\ \mathrm{Pr} &= 0.72 \\ \mathrm{Pr}_t &= 0.9 \\ \tilde{S}_k &= 0.4 \end{aligned}$$

As in the laminar case the plate was assumed to be adiabatic. The transition point was set at 30% of the chord where Re_θ was approximately 400. Figure. 2.9 shows the skin friction C_f as a function of x/c. The symbols represent the computational result which shows a smooth transition to turbulence at 30% chord. Indicated in Fig. 2.9 is also the laminar Blasius solution and the power-law estimate

$$C_f \simeq 0.026\,\mathrm{Re}_x^{-1/7} \tag{2.108}$$

given in [35]. Figure 2.10 shows the inner variable $u^+ = u/v^*$ as a function of the inner variable $y^+ = y\,v^*/\nu$ where $v^* = \sqrt{\tau_w/\rho}$ is the wall-friction velocity. The computed result (symbols) are compared with the inner law sublayer

$$u^+ = y^+ \tag{2.109}$$

and the logarithmic overlap layer

$$u^+ = \frac{1}{0.4}\ln y^+ + 5.0 \tag{2.110}$$

also obtained from White [35]. For both the skin friction and the velocity profile cases the agreement between computations and theory is fully satisfactorily.

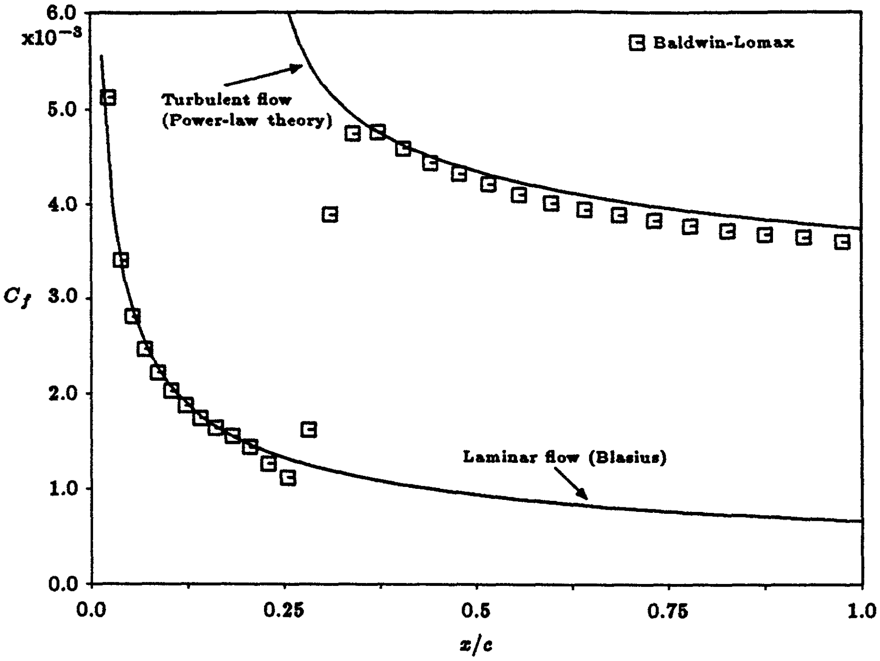

Figure 2.9: Skin friction along the flat plate with turbulent transition at 30% chord

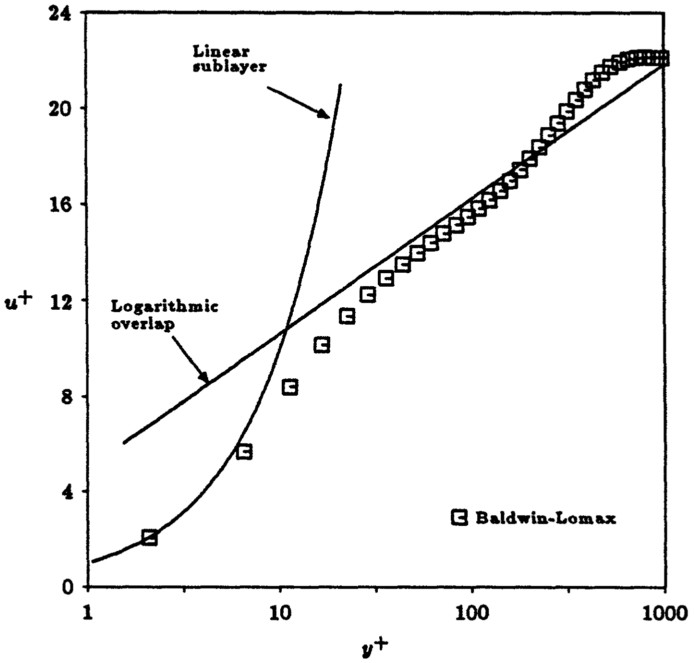

Figure 2.10: Turbulent-boundary-layer velocity profile compared with analytical expressions

2.11.2 Circular Arc

The circular arc bump in a channel is a commonly used test case both for viscous and inviscid computational methods. The geometry is often referred to in the literature as the Ni-bump since Ni was one of the first to use it as a test case to validate his inviscid Euler solver [23]. For moderate Mach and Reynolds numbers Chima [6], Davis [7] and Kallinderis [18] have performed viscous calculations both for laminar and turbulent flow on this geometry.

The computational mesh used consisted of 65 × 33 grid points which were clustered in the region of the boundary layer and at the leading and trailing parts of the bump. The width of the channel is equal to the length of the bump and the thickness to chord ratio of the bump is 10%. The geometry and the mesh size are identical to the ones used in [6] and [7]. This test case was run at inlet Mach number $M_\infty = 0.5$ and Reynolds number 8000. The temperature was specified on the bump and on the aft-wall to be equal to the inlet stagnation temperature. On this boundary the non slip condition, $u = v = 0$, was also imposed. At the inflow boundary the total pressure P_0, the total enthalpy H_0 and the flow angle α were specified. The pressure was extrapolated from the interior. At the outflow the static pressure p was specified and the density ρ and the velocity components u, v where extrapolated. At the upper boundary an inviscid, symmetry condition was imposed. The same boundary condition was imposed before the bump. All flow conditions and boundary conditions are the same as in refs. [7] and [6]. The mesh topology and the different flow conditions imposed at the different boundaries are shown in Figure 2.11.

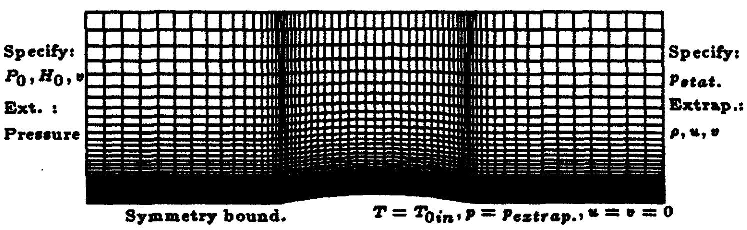

Figure 2.11: Computational mesh and flow conditions for the 10% circular arc cascade

In Figure 2.12 a comparison of the wall pressure obtained by different numerical schemes is shown. Figure 2.13 shows a comparison of the skin friction along the bump. The overall agreement between the different methods is good. As one would expect the major discrepancies between the results are in the region where the bump ends and the straight slit starts. As can be seen in the skin friction plot the flow is separated in this region. The cell centered method (present method) predicts a slightly larger separated region than the node based method ([7]). Other numerical experiments on the same geometry showed that the cell based scheme was more sensitive to grid discontinuities than the node based.

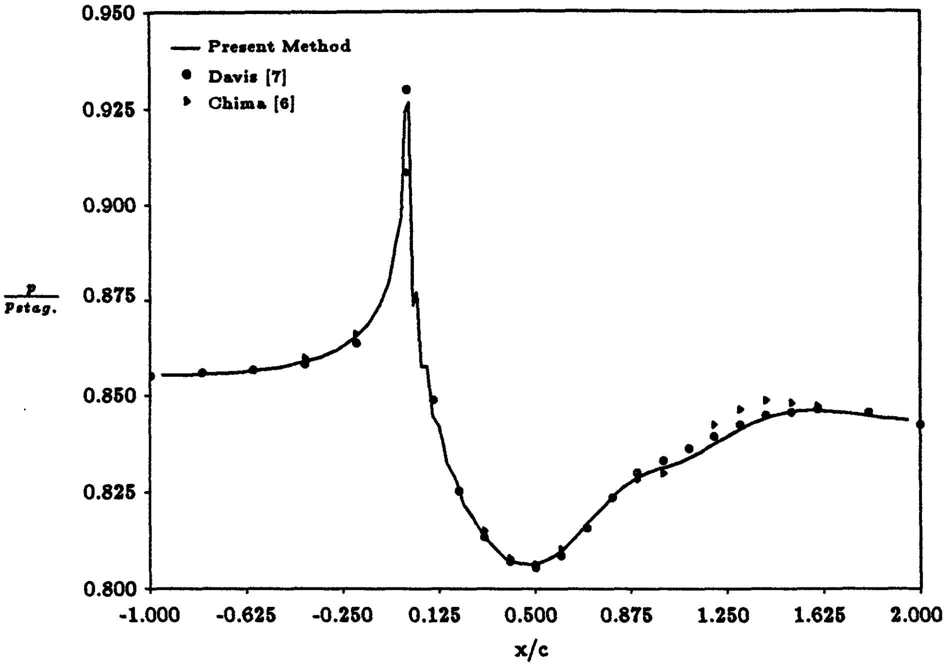

Figure 2.12: Wall pressure normalized by inlet stagnation pressure for the circular arc cascade

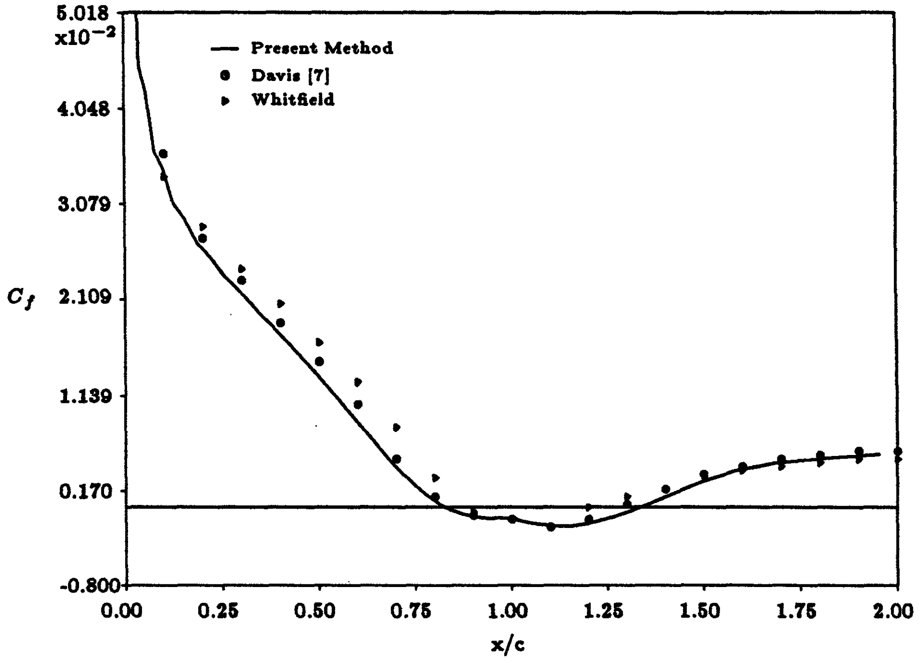

Figure 2.13: Skin friction along the circular arc cascade

Chapter 3
Unsteady Inviscid Code

Numerical simulation of unsteady effects in cascades such as wakes and hot streaks, calls for a time accurate method. The methods used for advancing a flow solution in time can be divided into two groups, namely the explicit and the implicit methods. The Beam-Warming time integration method described in the previous chapter is an example of an implicit method. For the purpose of studying unsteady flow in cascades it would, however, be very expensive computationally to apply an implicit method in the whole domain. The main numerical idea in this work is to combine a fast explicit method and a non CFL restricted implicit method. The explicit method has the advantage of being fast but with the drawback of being restricted by the CFL criterion. In the major part of the cascade region the flow is inviscid and hence governed by the Euler equations. From a computational point of view an inviscid/Euler region is well suited for an explicit method. The cells in the computational mesh in an inviscid region may have uniform size which optimizes the CFL criterion and hence the allowable time step for the explicit method.

In this work an explicit centered finite volume method will be used to solve the unsteady Euler equations in the inviscid regions of the cascade. The governing equations will be described in the first section. This section also gives the non-dimensionalization of the equations. In the second section the finite volume method is described. The one step, three stage Runge-Kutta type time integration algorithm is described and analyzed in the following section. In the last section the unsteady, non-reflecting inflow/outflow boundary conditions are described.

3.1 Governing Equations

An inviscid, compressible flow is governed by the Euler equations. The Euler equations can be derived from the Navier-Stokes equations by neglecting the viscous terms. In two-dimensions these equations can be written in conservative form as

$$\frac{\partial q}{\partial t} = -\left(\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y}\right) \tag{3.1}$$

where the vectors q, F and G is given by

$$q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix}, F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(e+p) \end{bmatrix}, G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(e+p) \end{bmatrix} \tag{3.2}$$

where ρ is density, u and v are cartesian velocity components, p is pressure and e energy per unit volume. To close system Eq. (3.1) an equation of state is needed. Assuming that the gas is perfect the equation of state becomes

$$e = \frac{p}{\gamma - 1} + \frac{1}{2}\rho\left(u^2 + v^2\right) \tag{3.3}$$

where γ is the ratio of specific heats.

The variables used with the Euler equations must match the Navier-Stokes variables. Therefore the Euler equations are non-dimensionalized using the axial chord c, upstream speed of sound a_∞ and upstream density ρ_∞. Using this choice of variables the non-dimensional quantities become

$$\begin{gathered} \bar{x} = \frac{x}{c}, \quad \bar{y} = \frac{y}{c}, \quad \bar{t} = \frac{t}{c/a_\infty} \\ \bar{u} = \frac{u}{a_\infty}, \quad \bar{v} = \frac{v}{a_\infty}, \quad \bar{\rho} = \frac{\rho}{\rho_\infty} \\ \bar{p} = \frac{p}{\rho_\infty a_\infty^2}, \quad \bar{e} = \frac{e}{\rho_\infty a_\infty^2} \end{gathered} \tag{3.4}$$

3.2 Finite Volume Method

For the numerical solution of the spatial part of the Euler equations Eq. (3.1), a cell centered finite volume method is used, which is identical to the one used for the viscous Navier-Stokes equations in the previous chapter. The method has been used successfully on the Euler equations by Eriksson [10], who used it for two-dimensional cascade flow. Following the same idea as in the Navier-Stokes case, the Euler equations are put in integral form. Using Green's theorem the equations are

$$\int_{\Omega} \frac{\partial q}{\partial t} dV + \oint_{\partial\Omega} (F dy - G dx) = 0 \tag{3.5}$$

The first approximation is to divide the domain Ω into a number of quadrilateral subdomains $\omega_{i,j}$ forming a structured computational mesh. The integral form of the Euler equations Eq. (3.3) is valid on each quadrilateral subdomain ω_{ij} as well as on the entire region Ω. If it is assumed that the state vector q is piecewise constant on each quadrilateral the integral can be written for each cell as

$$\frac{dq_{ij}}{dt} A_{ij} + \oint_{\partial\omega_{ij}} (F dy - G dx) = 0 \tag{3.6}$$

where A_{ij} denotes the area of cell i, j. The line integral approximation can be used under the assumption that the flux-vectors F and G are piecewise constant on the cell boundary. The semi-discrete formulation of the integral equation will become (see fig. 3.1 for notations)

$$\begin{aligned}
\left(\frac{dq}{dt}\right)_{ij} &= \frac{1}{2A_{ij}}[-(F_{i+1j} + F_{ij})\Delta y_a + (F_{i-1j} + F_{ij})\Delta y_c \\
&\quad +(G_{i+1j} + G_{ij})\Delta x_a - (G_{i-1j} + G_{ij})\Delta x_c] \\
&+ \frac{1}{2A_{ij}}[(F_{ij+1} + F_{ij})\Delta y_b - (F_{ij-1} + F_{ij})\Delta y_d \\
&\quad -(G_{ij+1} + G_{ij})\Delta x_b + (G_{ij-1} + G_{ij})\Delta x_d] \\
&= \mathbf{R}(q_{i-1j}, q_{ij-1}, q_{ij}, q_{i+1j}, q_{ij+1})
\end{aligned} \tag{3.7}$$

where the solution vector q is given in the center of the cell. The scheme is conservative and consistent. Referring to Section 2.10 the spatial operator is second order accurate on an equally distant mesh. The cell centered scheme Equation (3.6) is by itself

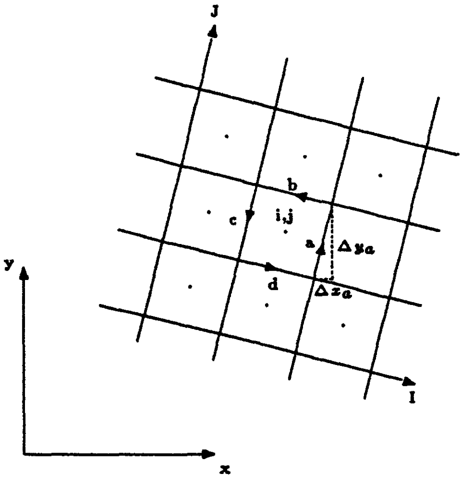

Figure 3.1: Computational cell ω_{ij}. Δx_a and Δy_a being the components of the tangential face vector a

non-dissipative and hence not stable when integrated in time. To obtain stability and uniqueness some extra dissipation has to be added. For consistency the same fourth order dissipative term given in Section 2.4 will also be used here.

Together with proper boundary conditions on both the flux vectors and the dissipation operator, Equation (3.7) forms a large system of non-linear ordinary differential equations. This system will be solved using an explicit time integration algorithm.

3.3 Explicit Time Integration

Since the work concerns unsteady calculations it is important that any numerical algorithms used to solve Equation (3.7) is of a high order of accuracy. Runge-Kutta type multistage schemes is a class of explicit schemes that are widely used for numerical solution of ODE's. These methods are designed to give high accuracy. In this work a second order accurate 1-step, 3-stage scheme will be used. The algorithm has been used extensively by Rizzi and Eriksson [9] to achieve steady solutions to the Euler equations.

The system of ordinary differential equations Eq. (3.7) with additional boundary conditions defines an initial value problem which can be written as

$$\begin{aligned} \frac{dq}{dt} &= \mathbf{R}(q) \\ q(0) &= q_0 \end{aligned} \tag{3.8}$$

Numerically this initial value problem will be solved using a 1-step, 3-stage scheme defined by

$$\begin{aligned} q(t_n) & \quad \text{given} \\ q^*(t_{n+1}) &= q(t_n) + \Delta t\, \mathbf{R}(q(t_n)) \\ q^{**}(t_{n+1}) &= q(t_n) + \frac{1}{2}\Delta t\, \mathbf{R}(q(t_n)) + \frac{1}{2}\Delta t\, \mathbf{R}(q^*(t_{n+1})) \\ q(t_{n+1}) &= q(t_n) + \frac{1}{2}\Delta t\, \mathbf{R}(q(t_n)) + \frac{1}{2}\Delta t\, \mathbf{R}(q^{**}(t_{n+1})) \end{aligned} \tag{3.9}$$

Accuracy

To show the accuracy of the scheme (3.9) a Taylor expansion has to be performed. Let the notation q^n be the usual notation $q^n = q(t_n)$. By Taylor expanding q and $\mathbf{R}$ the following is obtained

$$q^{n+1} = q^n + \left(\frac{dq}{dt}\right)^n \Delta t + \frac{1}{2}\left(\frac{d^2q}{dt^2}\right)^n \Delta t^2 + \cdots \tag{3.10}$$

$$\mathbf{R}(q^*) = \mathbf{R}(q^n) + \left(\frac{d\mathbf{R}}{dq}\right)^n (u^* - u^n) + \frac{1}{2}\left(\frac{d^2\mathbf{R}}{dq^2}\right)^n (u^* - u^n)^2 + \cdots \tag{3.11}$$

$$\mathbf{R}(q^{**}) = \mathbf{R}(q^n) + \left(\frac{d\mathbf{R}}{dq}\right)^n (u^{**} - u^n) + \frac{1}{2}\left(\frac{d^2\mathbf{R}}{dq^2}\right)^n (u^{**} - u^n)^2 + \cdots \tag{3.12}$$

The following chain rule relation is also needed to show the order of accuracy of the scheme

$$\frac{d^2q}{dt^2} = \frac{d\mathbf{R}(q)}{dt} = \frac{d\mathbf{R}(q)}{dq}\frac{dq}{dt} \tag{3.13}$$

Inserting these relations in Equation (3.9) and skipping all the tedious algebra involved, one ends up with following expression

$$\frac{dq}{dt} - \mathbf{R}(q) = O(\Delta t^2) \tag{3.14}$$

which shows that the truncation error is of order Δt^2 and hence the integration scheme is second order accurate.

Stability

The stability region for the Runge-Kutta scheme is obtained by analyzing the scheme with the equation

$$\frac{dq}{dt} = \alpha q \tag{3.15}$$

where α is some complex constant. Applying the Runge-Kutta scheme to this equation gives the following relation

$$q^{n+1} = (1 \; + \; \Delta t \, \alpha \; + \; \frac{1}{2}\Delta t^2 \, \alpha^2 \; + \; \frac{1}{4}\Delta t^3 \, \alpha^3) \, q^n \tag{3.16}$$

If $\bar{k} = \Delta t \, \alpha$ the stability region is defined as

$$D = \left[\text{complex } \bar{k} : |1 + \bar{k} + \frac{1}{2}\bar{k}^2 + \frac{1}{4}\bar{k}^3| \leq 1\right] \tag{3.17}$$

and is plotted in Figure 3.2.

To investigate the stability of the full explicit scheme, the model Equation (2.91) given in chapter 2 , will be used with $\nu_\eta = 0$. Since this is a hyperbolic equation the solution can be written in terms of its Fourier modes. The semi-discrete version of this model equation also has a solution consisting of discrete Fourier modes. In two space dimensions a general mode of the discrete solution can be written as

$$q(\xi_m, \eta_n, t) \; = \; \tilde{q}(t) \, e^{i(k_\xi m \Delta \xi + k_\eta n \Delta \eta)} \; = \; \tilde{q}(t) \, e^{i(m\theta_\xi + n\theta_\eta)} \tag{3.18}$$

Inserted in the semi-discrete model equation this gives the following equation

$$\frac{dq_{mn}}{dt} = \varphi \, q_{mn} \tag{3.19}$$

where

$$\varphi = -i \left(\frac{\lambda_\xi}{\Delta \xi} \sin(\theta_\xi) \; + \; \frac{\lambda_\eta}{\Delta \eta} \sin(\theta_\eta) \right) \tag{3.20}$$

The stability region (Figure 3.2) has as a bound on the imaginary axis $|\Im(\bar{k})| \leq 2$ where $\Im$ denotes the imaginary part and, as defined above, $\bar{k} = \Delta t \varphi$. The bound on the real

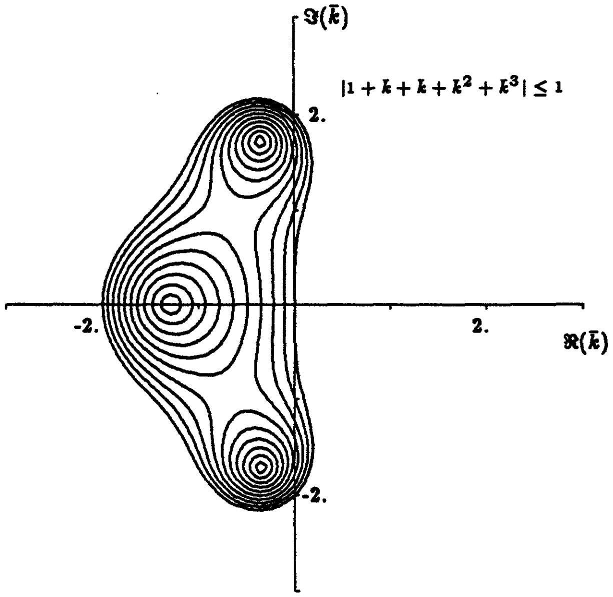

Figure 3.2: Stability region for the 3-stage Runge-Kutta method

axis is $-2 \leq \Re(\bar{k}) \leq 0$ where $\Re$ denotes the real part. Choosing $|\lambda_\xi|/\Delta\xi$ and $|\lambda_\eta|/\Delta\eta$ to be the largest, modulus of the eigenvalues, of Jacobian matrix in the Euler equations (see ref. [27]) will give a conservative estimate on the largest possible time step. Using these eigenvalues will give the following CFL restriction

$$\left|\frac{|\vec{u}\cdot\vec{l}| + |\vec{u}\cdot\vec{m}| + a(|\vec{l}| + |\vec{m}|)}{\Delta\xi\,\Delta\eta}\right| \leq 2 \tag{3.21}$$

where $\vec{u}$ is the total velocity, $\vec{l}$ and $\vec{m}$ the normal face vectors in the i and j directions and a the local speed of sound. If $\epsilon_\xi = \epsilon_\eta \leq 0.01$ are chosen, the real part of $\bar{k}$ is going to be well within the stability region. The imaginary part of the stability region even gets larger for small negative values on $\bar{k}$ (see Figure 3.2).

3.4 Solid Wall Boundary Condition

At a solid wall the normal mass flux is of course zero but the unsteady pressure at the wall will contribute to the momentum flux. If $\vec{n}\cdot\vec{v} = 0$ ($\vec{n}$ being the normal vector) at some part ∂Solid of the integration path $\partial\omega$ in Eq. (3.6) it is easy to see that the contribution to the path integral from the ∂Solid part becomes

$$\int_{\partial\text{Solid}} (F dy - G dx) \simeq \begin{bmatrix} 0 \\ \Delta y_{i,w}\, p_{i,w} \\ -\Delta x_{i,w}\, p_{i,w} \\ 0 \end{bmatrix} \tag{3.22}$$

The pressure at the wall $p_{i,w}$ is obtained by a linear interpolation procedure. Again using the index notation in Fig. 3.3 the wall pressure is given by

$$p_{i,w} = \frac{3}{2} p_{i,1} - \frac{1}{2} p_{i,2} \tag{3.23}$$

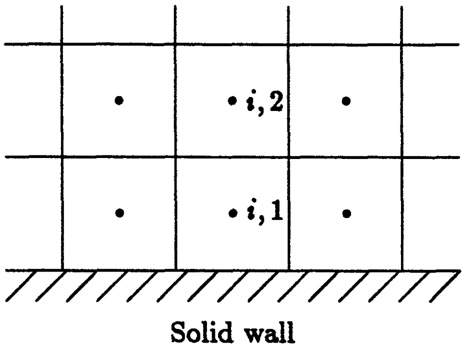

Figure 3.3: The computational mesh close to a solid wall

3.5 Unsteady Inflow/Outflow Boundary Conditions

At the inflow and outflow boundaries the method described by Giles [12] will be used. The method is based on an analysis of the unsteady 1-D Euler equations. The Euler equations can be written in non-conservative form as

$$\frac{\partial q}{\partial t} + A \frac{\partial q}{\partial x} = 0 \tag{3.24}$$

where

$$q = \begin{bmatrix} \rho \\ u \\ p \end{bmatrix} \quad , \quad A = \begin{bmatrix} u & \rho & 0 \\ 0 & u & \frac{1}{\rho} \\ 0 & \rho a^2 & u \end{bmatrix} \tag{3.25}$$

where a is the local speed of sound. Only linear perturbations from the uniform flow q_0 will be considered and hence A_0 is a function only of the steady flow and $\Delta q = q - q_0$ is the unsteady fluctuation. The perturbation form of the Euler equations is now written as

$$\frac{\partial \Delta q}{\partial t} + A_0 \frac{\partial \Delta q}{\partial x} = 0 \tag{3.26}$$

A_0 can be diagonalized by a similarity transform since A_0 is nondefective, i.e. multiple eigenvalues of A_0 corresponds to linearly independent eigenvectors. A theorem from linear algebra states that (see reference [31]) A_0 is nondefective if and only if there is a non-singular matrix T_0 such that

$$T_0^{-1} A_0 T_0 = \begin{bmatrix} u & 0 & 0 \\ 0 & u+c & 0 \\ 0 & 0 & u-c \end{bmatrix} = \Lambda_0 \tag{3.27}$$

where the diagonal elements are A_0's eigenvalues. It can also be noted that the ith column of T_0 is an eigenvector corresponding to λ_i, and the ith row of T_0^{-1} is a left eigenvector corresponding to λ_i. Since A_0 is a constant matrix, T_0 and T_0^{-1} are constant as well. Multiplying Eq. (3.26) from the left by T_0^{-1} gives the characteristic equation

$$\frac{\partial \Phi}{\partial t} + \Lambda_0 \frac{\partial \Phi}{\partial x} = 0 \tag{3.28}$$

where $\Phi = T_0^{-1} \Delta q$. The vector Φ is known as the linear characteristic variables which in detail can be written as

$$\begin{aligned} \phi_1 &= (\Delta\rho) - \frac{\Delta p}{c_0^2} \\ \phi_2 &= (\Delta u) + \frac{\Delta p}{\rho_0 c_0} \\ \phi_3 &= -(\Delta u) + \frac{\Delta p}{\rho_0 c_0} \end{aligned} \tag{3.29}$$

By assuming no variation in the y-direction, the fourth equation in the 2-D case is given by

$$\frac{\partial \phi_4}{\partial t} + u \frac{\partial \phi_4}{\partial x} = 0 \tag{3.30}$$

where

$$\phi_4 = \Delta v \tag{3.31}$$

By checking the eigenvalues for a subsonic flow one can see that ϕ_1, ϕ_2 and ϕ_4 are propagating downstream and ϕ_3 is propagating upstream. In order to prevent spurious reflection of waves at the inflow boundary three characteristic variables, ϕ_1, ϕ_2 and ϕ_4, are specified to be zero and one, ϕ_3, is extrapolated from the interior nodes. At the outflow boundary one characteristic variable, ϕ_3 is specified to be zero and the other three are extrapolated.

To allow for specified unsteadiness at the inflow boundary such as velocity defects and hot streaks, requires modification of the characteristic variables $\phi_1, \phi_2, \phi_3, \phi_4$ so that these are perturbations of some known inlet disturbance.

$$\begin{aligned}
\phi_1 &= (\Delta\rho) - \frac{\Delta p}{c_{is}^2} \\
\phi_2 &= (\Delta u) + \frac{\Delta p}{\rho_{is} c_{is}} \\
\phi_3 &= -(\Delta u) + \frac{\Delta p}{\rho_{is} c_{is}} \\
\phi_4 &= \Delta v
\end{aligned} \tag{3.32}$$

where $\Delta\rho = \rho - \rho_{inl}, \Delta u = u - u_{inl}, \Delta v = v - v_{inl}$ and $\Delta p = p - p_{inl}$. The subscript $_{inl}$ refers to the known inlet value and the subscript $_{is}$ refers to the steady flow variables in the stator frame.

3.6 Wake Models

Two different kinds of inflow disturbances will be used in this work. The first is a sinusoidal velocity perturbation that is used as a test case for the unsteady Euler code. The velocity defect model was suggested and used by Giles [13]. The second is a temperature excess simulating a hot streak from a burner in a jet-engine.

3.6.1 Velocity Perturbation

In the velocity wake model used for the test case it is assumed that the wake flow is parallel, that the static pressure is uniform and that the total enthalpy is uniform. To be able to compare computed results with those from a linear method, the velocity defect is defined to be sinusoidal. The flow variables are given by

$$\begin{aligned} p_{iw} &= p_{is} \\ u_{iw} &= u_{is}\,(1 - D\sin(2\pi\eta)) \\ v_{iw} &= v_{is}\,(1 - D\sin(2\pi\eta)) \\ \rho_{iw} &= \frac{\gamma}{\gamma-1}\frac{p_{is}}{\left(H_{is} - \frac{1}{2}(u_{iw}^2 + v_{iw}^2)\right)} \end{aligned} \tag{3.33}$$

where the subscript $_{is}$ denotes inlet-steady values and $_{iw}$ denotes the inlet-wake values. The constant D is the fractional velocity defect. H_{is} is the inlet-steady total enthalpy related to the other quantities by

$$H = e + \frac{p}{\rho} \tag{3.34}$$

The parameter η is defined by

$$\eta = \frac{y + V_{rot.}t - \tan(\alpha_{is})x}{P_s} \tag{3.35}$$

where $V_{rot.}$ is the rotational speed, α_{is} is the inlet-steady flow angle and P_s is the pitch. The final inlet flow in the rotor frame is obtained by applying a Lagrangian transformation

$$\begin{aligned} \rho_{inl}(x,y,t) &= \rho_{iw}(\eta) \\ u_{inl}(x,y,t) &= u_{iw}(\eta) \\ v_{inl}(x,y,t) &= v_{iw}(\eta) - V_{rot.} \\ p_{inl}(x,y,t) &= p_{iw}(\eta) \end{aligned} \tag{3.36}$$

3.6.2 Hot Streak

A model similar to the one used for the velocity perturbation is used for the temperature excess. To be consistent with the physical problem it is assumed that the flow is parallel along the inflow boundary. It is also assumed that the Mach number, static and total pressures in the hot streak are the same as in the free flow. The initial temperature excess given in [5] is more or less a square well. This distribution will, after convection through the stator row, be assumed to have been diffused slightly. The smoothed distribution is given by a cubic spline function $W(\xi)$ which has a smooth transition region spanning over 10% of the pitch. Under these assumptions the inlet flow quantities are given by

$$\begin{aligned} p_{ihs} &= p_{in} \\ T_{ihs} &= T_{in}\,(1 + D\,W(s(\eta))) \\ \rho_{ihs} &= \rho_{in}\,(1 + D\,W(s(\eta)))^{-1} \\ u_{ihs} &= u_{in}\,[1 + D\,W(s(\eta))\,]^{\frac{1}{2}} \\ v_{ihs} &= v_{in}\,[1 + D\,W(s(\eta))\,]^{\frac{1}{2}} \end{aligned} \qquad (3.37)$$

As before the subscript $_{ihs}$ denotes inlet-hot streak values and $_{in}$ denotes the inlet-steady values obtained from a steady-state calculation in the rotor frame. M_{in} is the inlet Mach number, D is the fractional temperature excess and W is the distribution function shown in Figure 3.4.

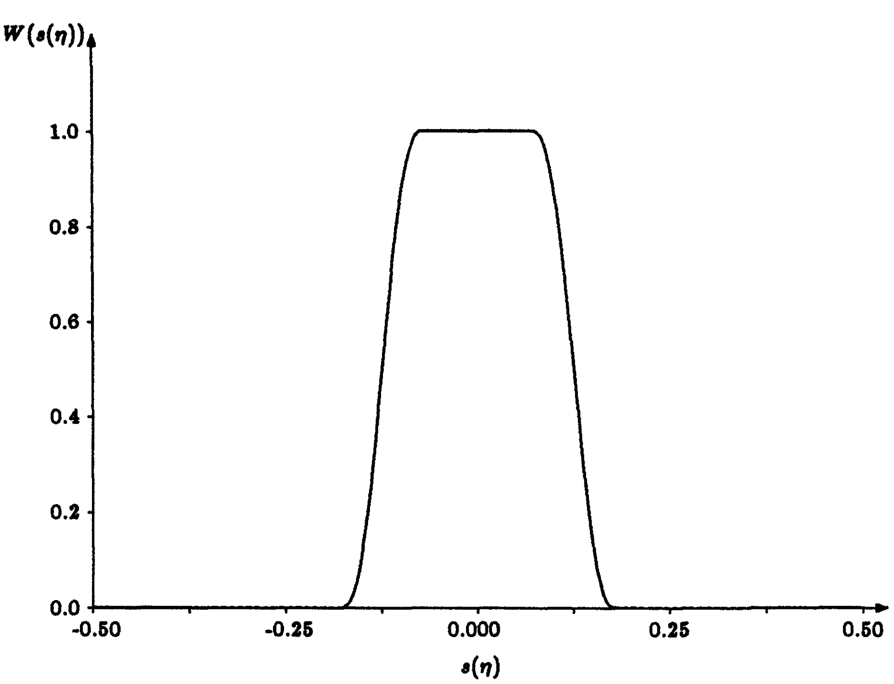

Figure 3.4: Continuous transition function simulating the hot streak

Periodicity is enforced through the function $s(\eta)$ in Equation (3.37) which is a periodic sawtooth function defined by

$$s(\eta) = \eta - n, \qquad n - \frac{1}{2} < \eta < n + \frac{1}{2} \tag{3.38}$$

and graphically shown in Figure 3.5. The parameter η is defined by

$$\eta = \frac{y + V_{rot}.t - \tan(\alpha_{is})x}{P_s} \tag{3.39}$$

Once again the final inlet flow in the rotor frame is obtained by applying a Lagrangian transformation

$$\begin{aligned}
\rho_{inl}(x,y,t) &= \rho_{ihs}(\eta) \\
u_{inl}(x,y,t) &= u_{ihs}(\eta) \\
v_{inl}(x,y,t) &= v_{ihs}(\eta) - V_{rot}. \\
p_{inl}(x,y,t) &= p_{ihs}(\eta)
\end{aligned} \tag{3.40}$$

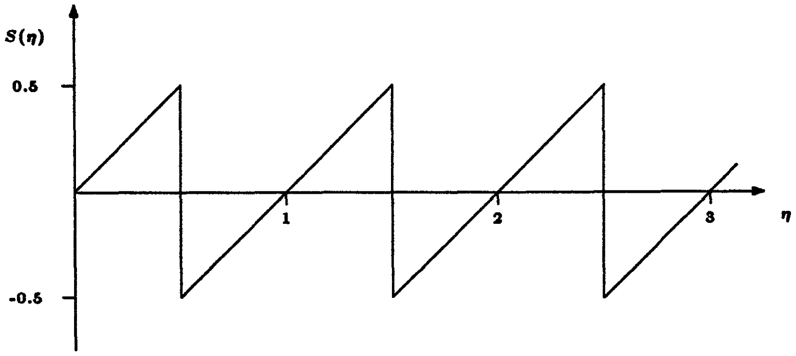

Figure 3.5: Sawtooth function $s(\eta)$

3.7 Flat Plate Test Case

In order to validate the unsteady Euler code quantitatively a very simple linear test case is considered. The test case has previously been used by Giles [13] to validate an unsteady Euler code. The flow situation is a low amplitude sinusoidal gust entering a flat plate cascade. As pointed out by Giles, this flow case can be analyzed using the program LINSUB, developed by D. Whitehead [36] based on the linear perturbation theory of S. Smith [29].

Table 3.1 lists the relevant flow and geometry parameters. The values chosen are exactly the same used by Giles. The wake defect was chosen so that it was big enough to avoid machine accuracy problems and small enough to ensure the solution to be linear.

Table 3.1: Parameters for flat plate test case

Pitch/chord (P_s/c)	0.57735
Stagger angle	30°
Mach number (steady flow)	0.7
Wake flow angle	−30°
Wake velocity defect	0.05
Reduced frequency $k = (\omega c/U_{\text{axial}})$	12.5664

The verification of linearity of the solution was done by Fourier transforming (DFT) the resultant unsteady pressure distribution on the blade, and checking that the second Fourier mode of both lift and moment was less than 1% of the first linear Fourier mode.

Computational results were obtained on three different grids. The basic coarse grid dimensions were 124 × 11. Doubling these dimensions gives the medium grid 247 × 21 and doubling again gives the fine grid 493 × 41. All three grids extended one chord length upstream and down stream from the blade row respectively. The calculations were carried out long enough to ensure that the lift were converged within 1% . The calculations on the coarse grid were started from a uniform flow and required about 25 cycles for convergence. The calculations on the finer meshes were started from an interpolated solution of the previous solution and were run about 15 cycles each.

The unsteady pressure distribution on the flat plate was Fourier transformed by a DFT-program and then non-dimensionalized as specified by Whitehead, to be able to compare it with LINSUB. Figs. 3.6 and 3.7 shows the real and imaginary components of the complex amplitude of the first Fourier mode of the pressure jump across the flat plate. The symbols are the solution obtained by LINSUB. The agreement between the two solutions is very good. The major discrepancies are in the imaginary component at the trailing edge which can be explained by the fact that the method is cell centered and hence does not have a computational point exactly at the trailing edge. The $1/\sqrt{x}$ singularity at the leading edge is surprisingly well resolved, even if one can trace some minor oscillations.

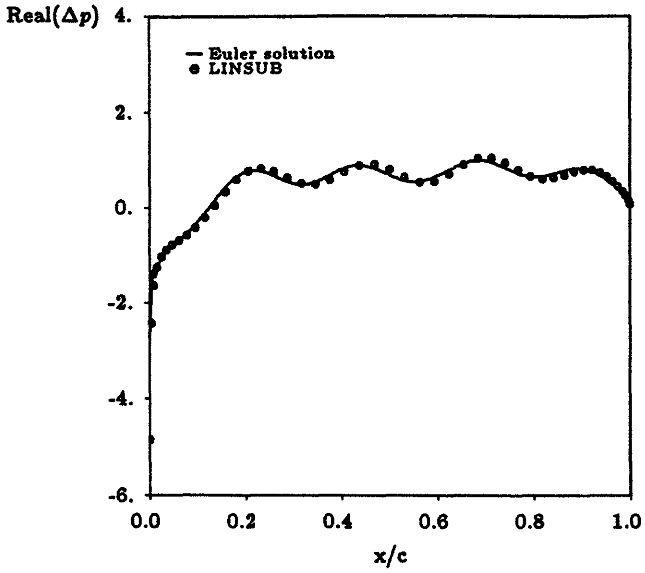

Figure 3.6: Real part of complex pressure jump over the flat plate

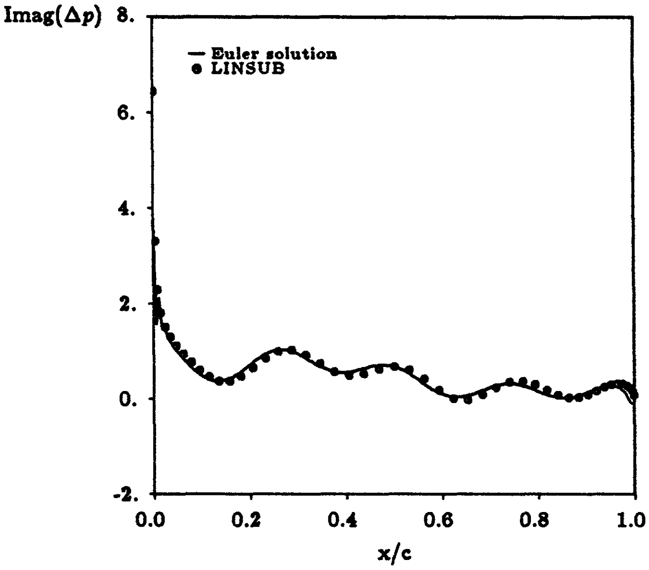

Figure 3.7: Imaginary part of complex pressure jump over the flat plate

Table 3.2 shows a comparison of the unsteady lift and moment coefficients obtained on the three different grids. The solution on the coarse grid is very far from the one obtained by LINSUB. The solutions obtained on the finer grids are much better. On the finest grid the error is approximately 3%. As was observed by Giles, a Richardson extrapolation is possible since the leading error is proportional to the mesh spacing. Using a Richardson extrapolation formula an even better result were obtained. The extrapolation is given by

$$p^{exact} \simeq p^{fine} + \frac{1}{3}(p^{fine} - p^{coarse}) \tag{3.41}$$

The coarse solution is here taken to be the 247 × 21 solution and the fine solution the 493 × 41 solution. The results are given in Table 3.2. The extrapolation reduced the error to within 1%.

Table 3.2: Unsteady lifts and moments for flat plat cascade

Code	mesh	lift		Moment	
	size	Real	Imag	Real	Imag
Present Euler	124 × 11	−0.356	−0.255	−0.226	−0.061
	247 × 21	−0.510	−0.463	−0.343	−0.139
	493 × 41	−0.503	−0.500	−0.345	−0.160
	extrap.	−0.501	−0.512	−0.346	−0.167
LINSUB		−0.501	−0.517	−0.347	−0.166

Chapter 4
Mesh Generation

One crucial component in the numerical simulation of a fluid flow using the two previously defined methods is the computational mesh. The cell-centered, central difference methods described in chapters 2 and 3 are shown to be sensitive to mesh discontinuities such as sharp kinks in mesh lines. These types of discontinuities will especially give rise to problems if they appear in regions where the flow is subject to large gradients. One of the novel features of this work is that the mixed viscous-inviscid numerical algorithm is run on a mixed viscous-inviscid grid. In the mesh strategy suggested in this chapter the total computational mesh will consist of two non-overlapping sub-grids and hence it is important that the interface between the two meshes is smooth. It is also desirable that the mesh generation algorithm is sufficiently flexible so that it can be used for a variety of different geometries. Some properties such as mesh stretching, clustering of grid points etc. must be easy to define and change.

In this chapter two different mesh generation techniques are described. In the first section an algebraic technique that generates an O-mesh close to a cascade blade is defined. The next section shows a technique based on the solution of an elliptic PDE. The PDE method is used to generate a mesh in the domain between two cascade blades. Finally the algorithm for generating the complete computational mesh is given.

4.1 Viscous Mesh

The first grid of the two is defined as the inner viscous grid. This grid will be defined close to the blade where the flow is highly viscous and hence governed by the Navier-Stokes equations. The inner grid must have the following properties

- It must be dense enough in the direction normal to the blade to be able to resolve the boundary-layer around the blade.
- The stretching in the direction normal to the blade must be within a certain limit for the numerical simulation of the viscous stresses to be at least first order accurate.
- The region around the leading and trailing edges must be sufficiently resolved.

These properties indicate that the O-mesh structure will be best suited for the inner mesh. The O-mesh structure is defined as a structured grid that has one parameter line runing around the profile and one runing in the normal direction to the profile (see Fig. 4.1). Consequently the mesh only has two boundaries.

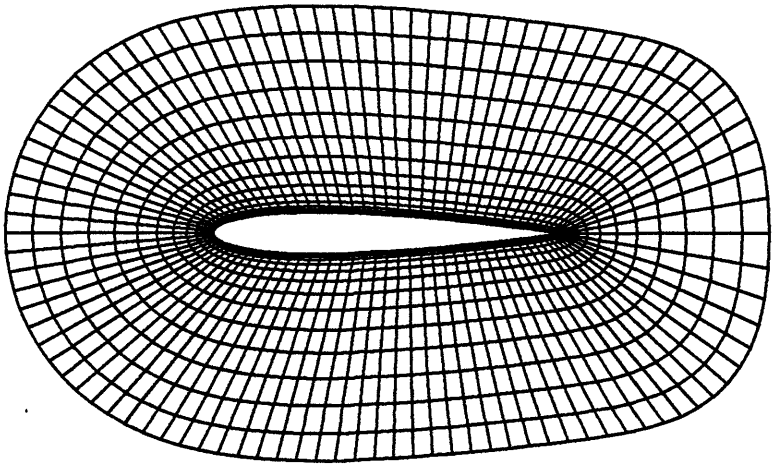

Figure 4.1: O-mesh structure around a NACA0012 profile

4.1.1 O-mesh Generation

In this work a very simple algebraic method is used for the O-mesh generation. The generator is essentially based on two basic spline routines, a cubic spline routine for the basic grid point distribution and a penta spline routine for the collocation functions.

By changing parameters in these routines it is easy to cluster points in areas where gradients are expected to be large.

Numerical procedure

Given a number of points defining the pressure and suction surfaces of the blade, a standard cubic spline routine is used to parameterize the complete surface. The routine gives as an output besides position of a surface point also the 1st and 2nd derivative at that point. Since it is a cubic spline routine these derivatives are both continuous. For the mesh generation the normal derivatives are needed. It is easy to transform the derivatives obtained from the spline functions to the desired normal derivatives. Using a penta spline collocation function, mesh points are distributed along the blade surface. This function can cluster points in the region of interest such as the leading and trailing edges. The set of points generated by the spline functions will now define the inner boundary of the O-mesh. An excellent description of general spline functions can be found in [4].

The rest of the O-mesh is now built up from the points on the inner boundary. By using the constant r_g defined as the mesh thickness, and by using the normal derivatives given by the spline function the outer boundary can be defined. By combining corresponding points on the inner and outer boundary linearly, the normal grid lines are defined. Let n_x and n_y be the components of the normal vector on the blade. Let further (x_s, y_s) be some point on the surface. The corresponding point on the outer mesh boundary is the given by the linear expression

$$\left[\begin{array}{lcl} x_0 & = & x_s + r_g\, n_x \\ y_0 & = & y_s + r_g\, n_y \end{array} \right. \tag{4.1}$$

where the thickness parameter r_g is given as percent of axial chord length. Grid points can now easily be distributed along each normal grid line.

The ratio between the height of two consecutive computational cells is known as the stretching. One can show that the stretching in the normal direction puts a restriction on the accuracy of the numerical evaluation of the viscous stresses. To be able to globally control the stretching of the mesh in the direction normal to the blade, the normal distribution function is defined as follows. If the parameter r_s defines the stretching in the normal direction, the first mesh point lays on a distance Δ_1 from the blade surface. The first spacing Δ_1 is defined by

$$\Delta_1 = \frac{r_g}{\sum\limits_{n=1}^{N-1} r_s^n} \tag{4.2}$$

where N is the number of grid points in the normal direction. The j^{th} grid point on a normal mesh line is given by the following collocation formula

$$\left[\begin{array}{lcl} x_j & = & x_s + \Delta_1 \sum\limits_{i=1}^{j-1} r_s^i \, n_x \\ y_j & = & y_s + \Delta_1 \sum\limits_{i=1}^{j-1} r_s^i \, n_y \end{array} \right. \tag{4.3}$$

Figure 4.2 shows an O-mesh generated by this method around a typical turbine blade. Figure 4.3 shows an enlargement of the mesh at the leading edge.

This very simple mesh generation algorithm gives surprisingly good results. The method gives mesh lines that are normal to the blade surface which minimizes the skewness of the mesh in the boundary-layer region. Non-skewness is a desirable property especially if the boundary layer flow is turbulent. Due to its simplicity this method suffers from one major drawback. For highly cambered blades the mesh lines might cross each other. However, for the geometries considered in this work crossing mesh lines have not been a problem.

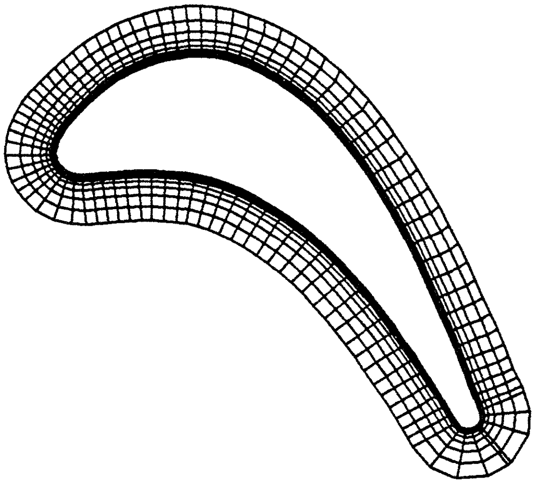

Figure 4.2: Viscous O-mesh generated around a typical turbine blade

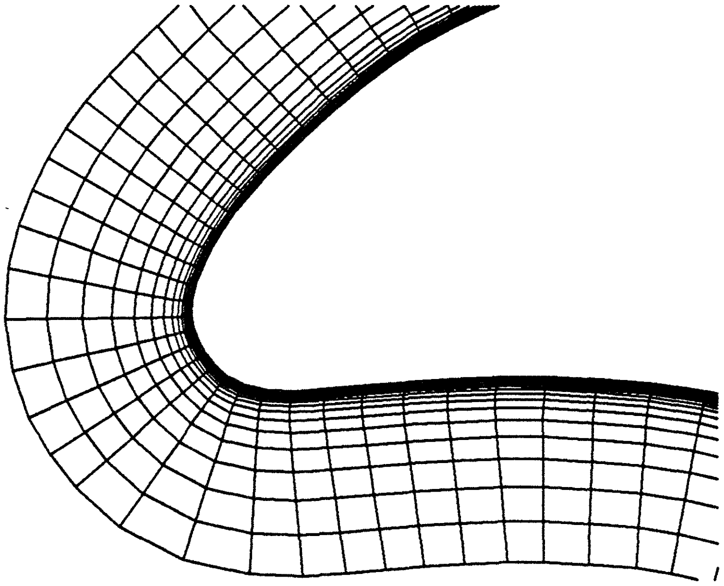

Figure 4.3: Enlargement of the mesh in the leading edge region of the blade shown in previous figure

4.2 Inviscid Mesh

The second mesh is defined as the outer inviscid grid. It is defined in the region where the flow is basically governed by the inviscid Euler equations. This grid as well as the previous O-mesh, is subject to some basic constraints

- The cell size must be as uniform as possible to ensure no severe CFL restrictions on the Euler solver.
- The grid must be dense enough in the direction normal to the flow to be able to resolve incoming wakes and temperature defects.

The best mesh structure satisfying these constraints is the H-mesh structure. This mesh can be described as having one parameter line running in the streamwise direction and one in the direction normal to the flow. However, the mesh lines do not necessarily have to be in these directions even if they usually are. The basic thing about the mesh is that it globally forms a logical quadrilateral. The mesh used in the channel flow problem in section 2.8, fig. 2.11 is an example of a H-mesh.

For generating the inviscid outer grid a method based on the solution of an elliptic PDE will be used. This method was first suggested and developed by Thompson [33]. The most commonly used method for controlling the grid point distribution, using the method of PDE, is solving the Poisson equation where the non-homogeneous source terms are the mesh-controlling terms. Giles [14], however, has suggested an algebraic method for the grid control which will be briefly described below. The algebraic method increases the flexibility of the grid point distribution and is also shown to be more robust than the Poisson method.

Poisson's Method

Let the coordinates (x, y) define the physical space and let the coordinates (ξ, η) define a transformed computational space. The transformation is constructed by specifying the desired grid points on the boundary of the physical domain. The distribution of grid points on the interior is then determined by solving

$$\xi_{xx} + \xi_{yy} = P(\xi, \eta) \tag{4.4}$$

$$\eta_{xx} + \eta_{yy} = Q(\xi, \eta) \tag{4.5}$$

where P and Q are the source terms that control the grid point distribution. Equations (4.4) and (4.5) are then transformed to computational space by interchanging the roles of independent and dependent variables. This yields a system of two elliptic equations of the form

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = -J^2(Px_\xi + Qx_\eta) \tag{4.6}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = -J^2(Py_\xi + Qy_\eta) \tag{4.7}$$

where

$$\begin{aligned}
\alpha &= x_\eta^2 + y_\eta^2 \\
\beta &= x_\xi x_\eta + y_\xi y_\eta \\
\gamma &= x_\xi^2 + y_\xi^2 \\
J &= x_\xi y_\eta - x_\eta y_\xi
\end{aligned} \tag{4.8}$$

These equations are solved on an uniformly spaced grid in the computational plane and P and Q control the grid point distribution. This transformation gives a one-one correspondence between the points in the two spaces. The choice of the functions P and Q is not a trivial task. Badly chosen source terms might even give rise to convergence problem in the corresponding numerical method.

Algebraic Method

The algebraic method suggested by Giles [14] is based on the solution of the Laplace's equation rather than the Poisson's equation. Following the same concept as in the Poisson method the transformed set of coordinates (ξ, η) is defined by the equations

$$\xi_{xx} + \xi_{yy} = 0 \tag{4.9}$$

$$\eta_{xx} + \eta_{yy} = 0 \tag{4.10}$$

After changing variables the equations become

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = 0 \tag{4.11}$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = 0 \tag{4.12}$$

where α, β and γ are defined by Equation (4.8). Instead of solving Equations (4.11) and (4.12) on an uniformly-spaced computational grid, algebraic manipulations are performed directly on the computational grid. This method gives a good direct control possibility on the mesh in physical space. Again the (x, y) coordinates for the boundary nodes in the physical space are specified. Equations (4.11) and (4.12) are solved numerically using finite differences and iterative SLOR methods.

To illustrate the method a mesh generated in a channel with a bump. The bump thickness was 50% of the channel height. Figs. 4.4 and 4.5 shows the mesh in the computational domain and the resulting mesh in the physical space. As can be seen in these figures the resolution at the leading and trailing edges is very bad. The cells closest to the lower wall are much to large. To improve the mesh some manipulations were made on the computational mesh. Figure 4.6 shows the new computational mesh. A function that increased the stretching in the normal direction was applied in the neighborhood of the leading and trailing edges. The resulting mesh in the physical domain, shown in Figure 4.7, indeed shows great improvements.

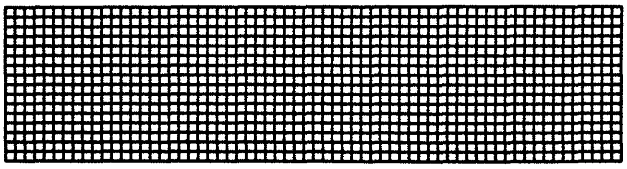

Figure 4.4: Uniformly spaced computational grid

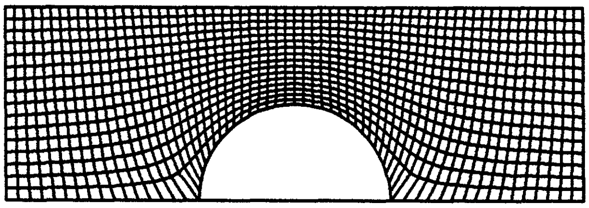

Figure 4.5: Pure Laplace mesh in a channel with a 50% thick bump

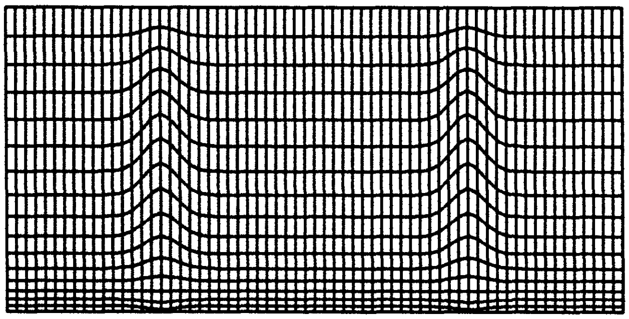

Figure 4.6: Non-uniform computational mesh where the kinks indicates the leading and trailing edges

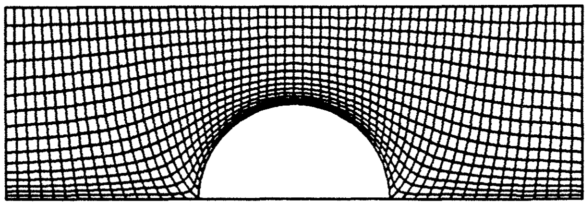

Figure 4.7: Improved mesh in the channel

4.3 Final Mesh

In the two previous sections, two different mesh generating techniques have been defined individually. In order to create a complete, mixed computational mesh these two methods must be combined together. The complete mesh is to be used in a two-dimensional cascade row, simulating a rotor in a turbine. For the general flow case in a cascade the computational domain could be very large. The problem can, however, be simplified considerably by observing that a cascade row consists of an infinite number of blades stacked periodically and by assuming that incoming disturbances in the flow are spatially periodic with the same period as the blade spacing. Using this assumption, the computational domain required is limited to the domain between two blades and therefore only two meshes are needed. These are, as defined in the previous sections, the O-mesh around a blade and the H-mesh in the region between two blades.

The first step is to create the O-mesh around the blade. Assuming that one has a rough idea of the thickness of the boundary layer, the outer mesh boundary is defined to be well outside the thickest part of the boundary layer (the boundary layer at the circular trailing edge not included). The mesh spacing, and number of mesh points in the normal direction is dependent on the current flow condition i.e. basically the Reynolds number and whether the flow is turbulent or laminar. Having defined the outer mesh boundary, the number of mesh points in normal and streamwise directions and the normal mesh stretching the O-mesh can be generated.

The second step is to generate the H-mesh in between the blades. Since the H-mesh is generated by an elliptic PDE method, all the boundaries around the domain must be defined. The first obvious boundary is the outer boundary of the O-mesh which defines parts of the upper and lower H-mesh boundary. Even if non-reflecting boundary conditions are used the computational domain is extended one chord length downstream and upstream respectively. Two lines having the same length as the blade spacing (pitch), are defined one chord length upstream and downstream. These lines are the left and right boundary of the H-mesh. To get a reasonably smooth transition between the O and the H meshes, two angles α and β are defined. The rays connecting the O-mesh boundary and the left and right boundaries makes angles α and β with the horizontal axis and hence the position of the left and right boundaries are fixed. The domain and the different boundaries are shown in Figure 4.8. Once the outer boundary is fixed the H-mesh can be generated.

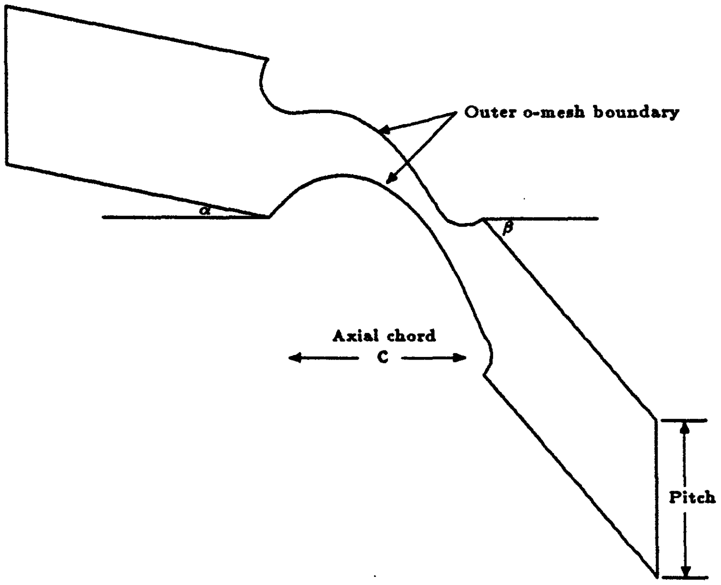

Figure 4.8: Geometry description of H-mesh outer boundary

The following algorithm is used to generate the full computational mesh

- Define the blade geometry and give O-mesh thickness r_g.
- Generate the O-mesh.
- Define leading and trailing edge angles α and β.
- Use the outer boundary of the O-mesh, α, β, chord length c and pitch p to define the H-mesh boundary in the physical space (see Figure 4.8).
- Generate the H-mesh.

In Figure 4.9 is shown a full mesh generated by this method around a typical turbine rotor cascade. In this figure it can be seen that there are kinks in the interface region between the two meshes. Since the numerical methods used here are sensitive to discontinuities these kinks have to be smoothed out. An easy way of smoothing these kinks is to run a redistribution program based on splines along the interface line. The program redefines the position of the mesh points in a predefined region around the interface line. A smoothing program with 6 mesh points bandwidth was run on the mesh shown in Figure 4.9. The result after the smoothing is shown in Figure 4.10.

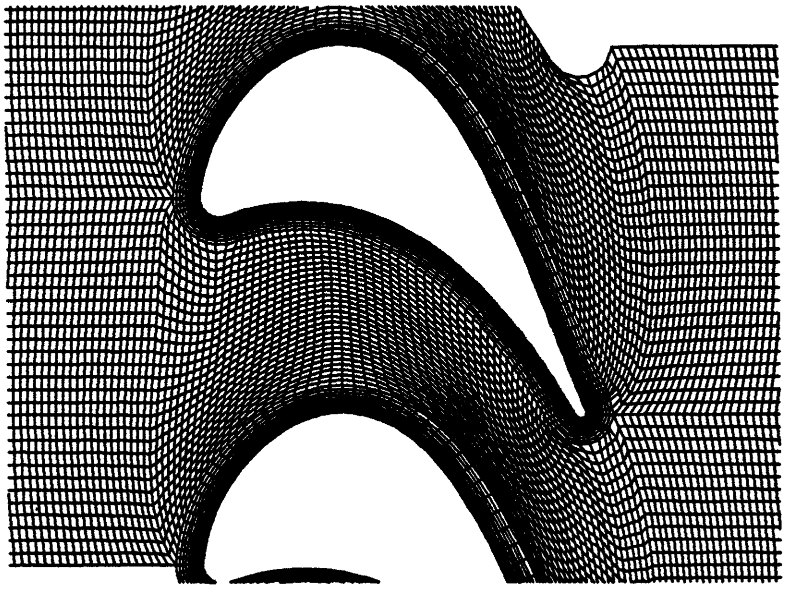

Figure 4.9: Complete mesh around a cascade without interface smoothing

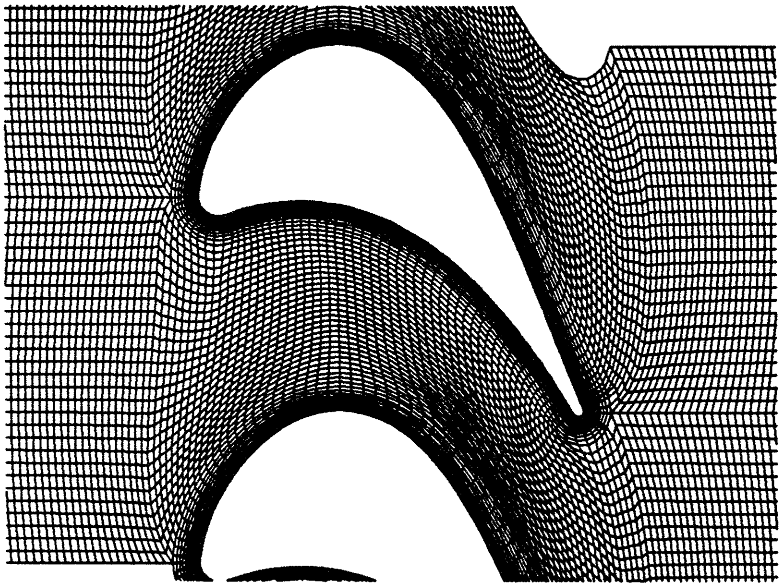

Figure 4.10: Complete mesh around a cascade with interface smoothing

Chapter 5

Viscous-Inviscid Interface

The main theme in this chapter is to describe the connection of the two different numerical methods described in chapters 2 and 3. As indicated earlier in the introduction, the two methods are to be used simultaneously on two different meshes that are defined on two regions with different flow structures. The complete combined region is the flow region in a two-dimensional cascade configuration. The two sub-domains being, a viscous boundary layer region where an implicit scheme is used and a inviscid channel region where an explicit scheme is used. As was shown, these two methods are second order accurate both in space and time. The main concern when these methods are combined together is to be able to maintain global second order accuracy even if the methods are locally first order accurate. Since the two schemes use the same spatial discretization method, the spatial operator can be considered to be working on the complete computational domain, unrestricted by the fact that the complete mesh consists of two different mesh topologies. For consistency, in performing an unsteady calculation, it is important that the fluxes between the two regions are calculated at the same time level. It is also important that the spatial smoothing is allowed to operate freely over the interface at the accurate time level.

The first section of this chapter reviews the discrete, spatial finite volume operator derived in chapters 2 and 3. The operator is given in a form that emphasizes the flux properties of the schemes. Using this flux form of the spatial operator makes it easier to point out the interface parts of the operator when studying the temporal algorithms. The second section describes the temporal interface between the implicit and the explicit methods.

5.1 Spatial Interface

The spatial discretization of the Euler and Navier-Stokes equations given in the viscous and inviscid chapter, chapters 2 and 3, is known as the finite volume method. The spatial operator associated with the finite volume method was introduced in Eq. (2.22) and in Eq. (3.7). Using a more compact form this operator can be rewritten as

$$\frac{1}{A_{ij}} \oint_{\partial\omega_{ij}} (Fdy - Gdx) \simeq \frac{1}{A_{ij}} \left(\sum_{k=i-1}^{i} \tilde{\mu}_{\xi}^{+} \left[\mathbf{H}(q_{kj})\right] \cdot \vec{l}_k + \sum_{k=j-1}^{j} \tilde{\mu}_{\eta}^{+} \left[\mathbf{H}(q_{ik})\right] \cdot \vec{m}_k \right) \tag{5.1}$$

where μ_k is the averaging operator defined by

$$\begin{aligned} \tilde{\mu}_{\xi}^{+} \varphi_{ij} &= \frac{1}{2}(\varphi_{i+1j} + \varphi_{ij}) \\ \tilde{\mu}_{\eta}^{+} \varphi_{ij} &= \frac{1}{2}(\varphi_{ij+1} + \varphi_{ij}) \end{aligned} \tag{5.2}$$

and the matrix $\mathbf{H}(q_{ij})$ has the flux vectors F and G as columns thus

$$\mathbf{H}(q_{ij}) = (F(q_{ij})\,\hat{\subset} \;+\; G(q_{ij})\,\hat{\supset}) \tag{5.3}$$

The vectors $\vec{l}$ and $\vec{m}$ are the face vectors of the cell ω_{ij} in ξ and η directions respectively (see Figure 5.1).

Interface Region

As long as each cell in the computational domain has only four closest neighbor cells, the previously described spatial finite volume operator will always be consistent which implies at least first order accuracy. The most critical point along the intersection between the two mesh topologies is the point where the H-mesh splits and is joined by the O-mesh. As can be seen in Figure 5.2 all cells in this critical region have only four faces and hence only four closest neighbors. The spatial operator along the interface line will always have four points in one region and one point in the other. Provided that the interface line is defined to be located outside the boundary layer, i.e., the viscous terms in the last layer of cells in the O-mesh can be neglected, the steady state spatial operator is consistent along this line. It was found that the numerical scheme became

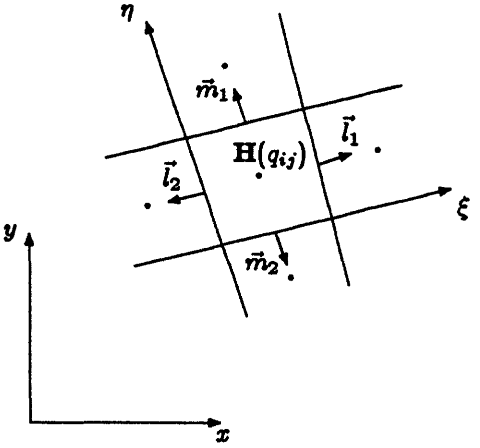

Figure 5.1: Computational mesh in ξ and η direction

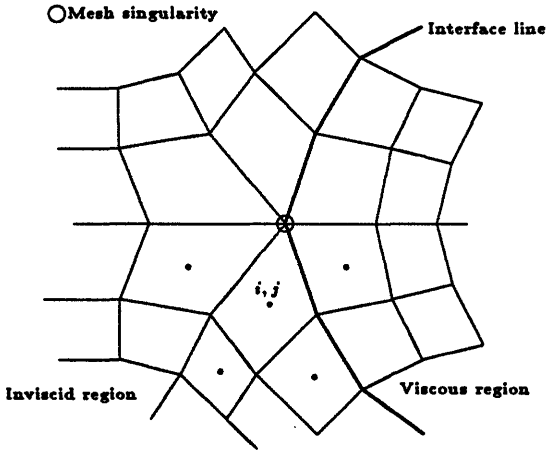

Figure 5.2: Points in the spatial finite volume operator and mesh singularities

unstable if the singular point (six joining mesh lines) was inside the strong gradient region of the leading edge. Fortunately when simulating the actual physical problem with hot streaks in the inflow, the periodically passing hot streaks did not give rise to any problem.

5.2 Temporal Interface

It was shown in the second chapter that the spatial finite volume operator was of global second order accuracy. It was also shown that both the implicit and the explicit time integration algorithms were second order accurate individually. For the complete mixed temporal scheme no proof will be given that it is second order accurate. The full temporal scheme might in fact be less than second order accurate along the interface line but it is definitely consistent and hence at least first order accurate. In this section a description of the temporal algorithm along the interface line is given. Some persuasive arguments are given that the possible lack of accuracy in the interface region does not affect the global accuracy.

In Figure 5.3 a graphical representation of the procedure at the interface is given. The arrows between the inviscid and viscous regions indicates the fluxes calculated at each stage. Note that the extrapolation update done in the viscous region is *not* used as a permanent update but just to enforce consistency in the inviscid region. The temporal matching between the viscous and inviscid regions can be described as follows

1. Calculate the flux through the interface between the inviscid and viscous region at time level n, i.e. $F^n_{int} = \frac{1}{2}(F^n_{visc} + F^n_{inv})$
2. Make a predictor step with the Runge-Kutta formula throughout the inviscid mesh. $q^* = q^n + \Delta t\, \mathbf{R}^n$ (Euler forward step)
3. Update the boundary cells in the viscous region using the extrapolation formula $q^* = q^n + \Delta q^{n-1}$
4. Calculate the flux through the interface between the inviscid and viscous region

at time level $n+1$, i.e. $F^*_{int} = \frac{1}{2}(F^*_{visc} + F^*_{inv})$

5. Make a corrector step with the Runge-Kutta formula through out the inviscid mesh.$q^{**} = q^n + \frac{1}{2}\Delta t\ (\mathbf{R}^* + \mathbf{R}^n)$
6. Calculate the flux through the interface between the inviscid and viscous region at time level $n+1$, i.e. $F^{**}_{int} = \frac{1}{2}\,(F^{**}_{visc} + F^{**}_{inv})$
7. Make a corrector step with the Runge-Kutta formula through out the inviscid mesh.$q^{n+1} = q^n + \frac{1}{2}\Delta t\ (\mathbf{R}^{**} + \mathbf{R}^n)$
8. Save the flux through the interface calculated by $F^{n+1}_{int} = \frac{1}{2}\,(F^n_{int} + F^{**}_{int})$
9. Make an implicit step through out the viscous region using the interface flux calculated in the explicit step

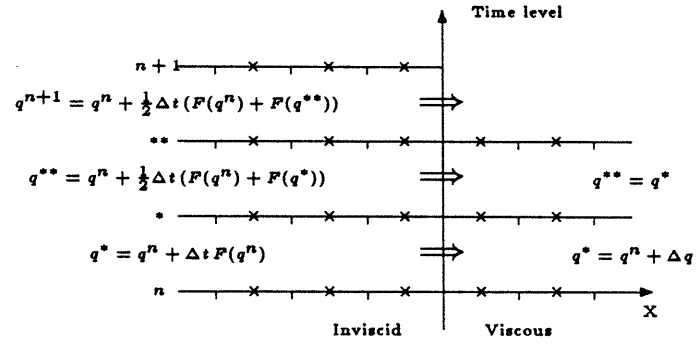
Time level
$n+1$
$q^{n+1} = q^n + \frac{1}{2}\Delta t\,(F(q^n) + F(q^{**}))$
$**$
$q^{**} = q^n + \frac{1}{2}\Delta t\,(F(q^n) + F(q^*))$
$q^{**} = q^*$
$*$
$q^* = q^n + \Delta t\, F(q^n)$
$q^* = q^n + \Delta q$
n
X
Inviscid
Viscous

Figure 5.3: Three stages in the Runge-Kutta time stepping scheme. Viscous and inviscid part

The fourth order smoothing (artificial dissipation) of the fluxes described in chapter 2 is imposed at each stage of the explicit scheme and in the beginning of the implicit step. To smooth the last two cells before the interface in both the regions, two points in the other region are needed for the five-point smoothing stencil. Two layers of cells next to the interface in each region are buffered and stored independently to be used for smoothing. The buffer values in the viscous region follows the updating described above and therefore the smoothing is always performed at the correct time level.

The algorithm described above is second order accurate everywhere except in the interface region. In the interface region, however, the scheme is consistent since the intermediate extrapolation update ($q^{n+1} = q^n + \Delta q^{n-1}$) in the viscous region is consistent. Consistency indicates that the scheme is at least first order accurate. The algorithm presented above is adequate in the interface region up to the second corrector step where the fluxes through the interface of the boundary cell are based on the predicted value in the viscous region rather then the corrected values. The first step is identical on both sides of the interface boundary since the Euler step and the extrapolation step are both first order accurate. Considering the spatial operator at the last corrector level $()^{**}$

$$\frac{1}{A_{ij}} \left(\sum_{k=i-1}^{i} \tilde{\mu}_{\xi}^{+} \left[\mathbf{H}(q_{kj}^{**}) \right] \cdot \vec{l}_k + \tilde{\mu}_{\eta}^{+} \left[\mathbf{H}(q_{ij-1}^{**}) \right] \cdot \vec{m}_{j-1} + \frac{1}{2} \left(\mathbf{H}(q_{ij}^{**}) + \underline{\mathbf{H}(q_{visc}^{*})} \right) \cdot \vec{m}_j \right) \tag{5.4}$$

where the only difference is in the averaged flux on the boundary (underlined term). However, the disturbance introduced to the scheme by averaging at different stages that might give local first order accuracy are not going to effect the global second order accuracy of the full scheme. This rather controversial statement is based on a theorem by Gustafsson [15] which states that for convergence, the accuracy of the boundary conditions to an interior scheme can be of one order less than the scheme it self.

Chapter 6
Hot Streak Simulations

The two-dimensional numerical simulation algorithm described in the previous chapters is used to simulate the redistribution of hot gas streaks in the rotor of a turbine stage. The hot streaks in the rotor originate from the combustor in a jet engine. They are convected through the stator with a minimum of mixing i.e. maintaining most of their original shape. The streak will not affect the flow features in the stator or along its boundary. The streaks leave the stator at a high angle relative to the rotational axis and enter the rotor where the computational domain is defined. The primary objective of the present numerical investigation is to try to predict and understand the observed migration of hot gas to the pressure surface of the rotor blade [5]. The migration phenomenon has previously been investigated numerically by Rai [26] using a full Navier-Stokes code in a complete two-dimensional turbine stage (stator + rotor) but with flow conditions differing from the ones used in the actual experiment. The intention of this work is to match the experimental conditions closer than the previous numerical investigation. This chapter will present results from three different computations.

The configuration in which the calculations are performed is taken to be a two-dimensional version of the experimental setup used in reference [5], the mid span section of the rotor in the full turbine stage. The actual experiment was conducted at United Technologies Research Center (UTRC) in their large scale rotating rig so the rotor will henceforth be referred to as the UTRC rotor. The original experiment was run at almost incompressible conditions with the outflow Mach number from the rotor being approximately 0.2. The steady inflow and outflow boundary conditions, upon which the unsteady boundary conditions are based, were obtained by performing a steady computation using the inviscid Euler solver UNSFLO developed by Giles [12]. For the UNSFLO computation the inflow angle α, inflow stagnation enthalpy and outlet static to

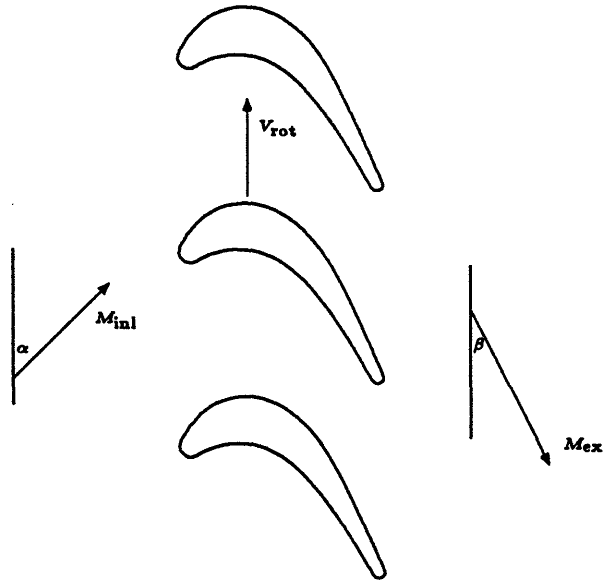

Figure 6.1: two-dimensional rotor geometry with inlet and exit flow conditions

inlet total pressure ratio were specified. The steady calculations yield the inflow/outflow boundary values for the Mach number and the flow angles (α and β). The geometry and the inlet/exit flow conditions are shown in Fig. 6.1. The unsteady computations were started from a field initialized by averaging the inflow and outflow steady conditions.

The dependent variables were, as described in chapter 2 and 3, nondimensionalized with respect to the axial chord c, the steady inflow values of density ρ_{∞}, the speed of sound c_{∞}, the temperature T_{∞} and the coefficient of viscosity μ_{∞}. In the rotating frame the unsteady inlet boundary conditions for the unheated gas become

$$\begin{aligned} u_{inl} &= M_{inl}\sin(\alpha) \\ v_{inl} &= M_{inl}\cos(\alpha) \\ \rho_{inl} &= 1. \\ p_{inl} &= \frac{1}{\gamma} \end{aligned}$$

At the exit the boundary conditions become

$$\begin{aligned} u_{ex} &= M_{ex}\sin(\beta) \\ v_{ex} &= M_{ex}\cos(\beta) \\ \rho_{ex} &= \frac{\rho_{inl}u_{inl}}{u_{ex}} \\ p_{ex} &= p_{inl}\left(1+\frac{\gamma-1}{2}M^2\right)^{\frac{\gamma-1}{\gamma}} r_p \end{aligned}$$

where the subscripts *inl* and *ex* refer to the inlet conditions and outflow conditions respectively and where r_p is the specified ratio between exit static pressure and the inlet stagnation pressure.

$$r_p = \frac{p_{ex}}{P_{0\,inl}}$$

The values in the stator frame are obtained by simply adding the rotational speed of the rotor V_{rot} to the y-component of the velocity v_{inl}. The rotational speed is given by

$$V_{rot} = \frac{u_{inl}}{\tan(\beta_{\text{stator}})} - v_{inl}$$

where β_{stator} is the exit flow angle from the stator in the stator frame. To specify the boundary values in the hot gas, the assumptions introduced in chapter 3 will be used. These were, the static and total pressure in the hot streak are the same as in the unheated gas and hence the Mach number is also the same. Let λ define the ratio between the hot and the cold gas then the following holds for the hot streak in the stator frame

$$\begin{aligned} u_{hs} &= u_{inl}\sqrt{\lambda} \\ v_{hs} &= (v_{inl}+V_{rot})\sqrt{\lambda} \\ \rho_{hs} &= \frac{\rho_{inl}}{\lambda} \\ p_{hs} &= p_{inl} \end{aligned}$$

where the subscript *hs* represents the hot streak. The final boundary conditions are defined and implemented using Equations (2.72) and (2.73).

To be able to compare the results both with the experimental data and with previous calculations, three different simulations were performed. The test cases were run at two

Table 6.1: Inflow/outflow boundary values for different flow coefficients

	$\phi = 0.78$	$\phi = 0.68$
Inlet Mach no. (M_{inl})	0.12	0.11
Exit Mach no. (M_{ex})	0.2	0.2
Inlet flow angle (α)	41°	46°
Exit flow angle (β)	24.33°	24.03°
Pressure ratio (r_p)	0.9725	0.9725

different flow coefficients, where the flow coefficient is defined as

$$\phi = \frac{u_{inlet}}{V_{rot}}$$

and with two different turbulence assumptions. In [26] Rai used the flow coefficient $\phi = 0.78$ and the temperature ratio $\lambda = 1.2$. In order to see what a higher temperature in the hot streak does to the flow field and the temperature distribution along the blade, the temperature ratio will be held at $\lambda = 2.0$ in all three test cases. In the first test case the flow coefficient will be $\phi = 0.78$ and the boundary layer will be assumed to be fully turbulent. In the actual experiment ϕ was set to 0.68 and λ was set to 2 and therefore, for the second and third computation, the flow coefficient will be 0.68. As in the first test case, the boundary layer in the second test case is assumed to be fully turbulent. To investigate the effects of a laminar or a turbulent boundary layer and the effects of transition the third test case is run with a laminar pressure surface and a partly laminar suction surface. Table 6.1 shows the inflow and outflow conditions obtained by UNSFLO for $\phi = 0.78$ and for $\phi = 0.68$. It can be seen that the flow conditions do not differ much between the two cases with the exception of the inflow angle α. In flow situations like those considered in this work it is reasonable to assume that the blade wall is adiabatic i.e. there is no heat flux at the blade surface. In both computations the Reynolds number, based on axial chord and inlet speed of sound, was chosen to be 10^6 to match the experimental data as well as Rai's computational data.

The three computations were performed on the same computational mesh. The inviscid H-mesh consisted of 193 × 51 mesh points and the viscous O-mesh consisted of

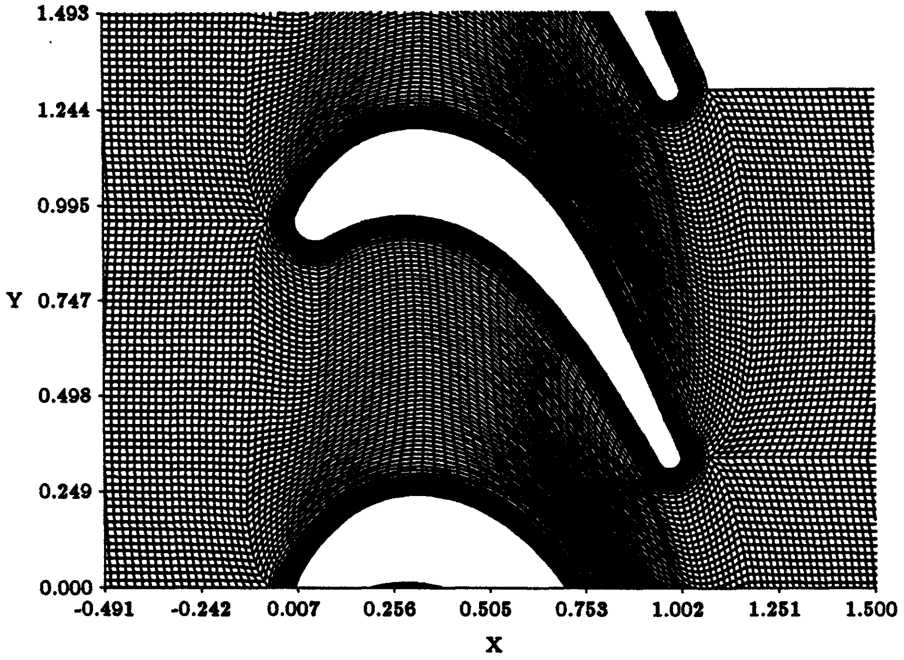

Figure 6.2: Computational mesh in the UTRC rotor

129×21 mesh points which gives a total of 12552 mesh points. For the O-mesh the stretching factor in the normal direction was set to 1.1 . To get a good resolution of the trailing edge, 15 mesh points were specified around the trailing edge circle. The full mesh is shown in Fig. 6.2, and enlarged details of the leading and trailing edge regions are shown in Figs. 6.3 and 6.4. To get a good temporal resolution, 2000 iterations were performed for each cycle (blade passing). For all test cases, this was shown to give a CFL restriction corresponding to $CFL = 1.65$ in the inviscid region and $CFL = 6.50$ in the viscous region. 7 cycles were needed to obtain a periodically converged solution from the free stream condition. All computations were performed on an 3-processor Alliant FX-8 computer for which the program has been optimized. The computational cost per iteration was computed to be 8.1 CPU seconds which gives a total of 4.5 CPU hours per cycle. It was found that the solution of the two sets of tri-diagonal equation systems took approximately 80% of the total computational time.

Even if the intention of the present simulation is to be as close to the experimental setup as possible there are still some major differences. One must bear these differences

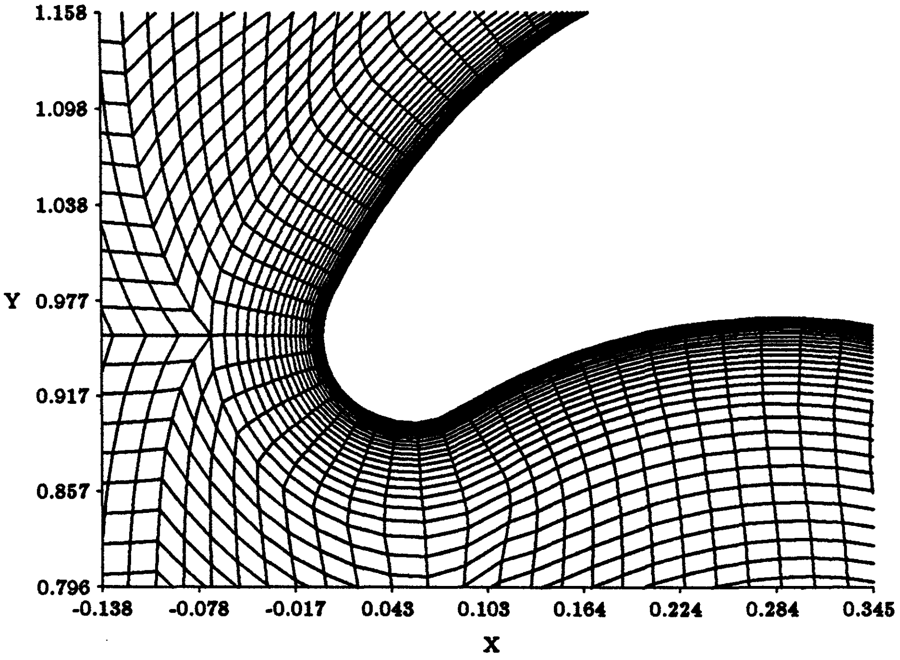

Figure 6.3: Computational mesh around the leading edge of the UTRC rotor

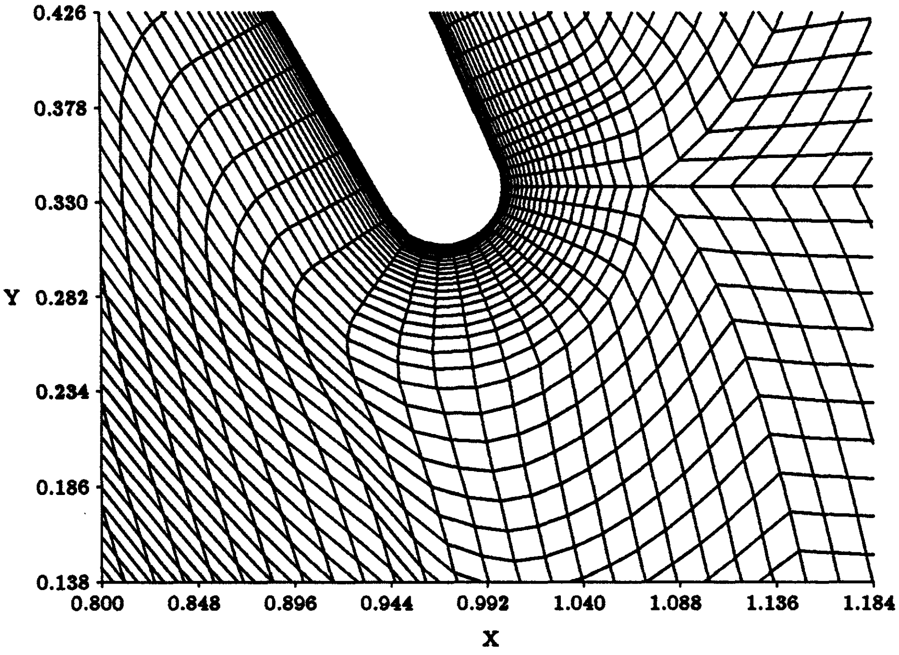

Figure 6.4: Computational mesh around the trailing edge of the UTRC rotor

in mind in the process of analyzing the computational results and comparing them to the experimental data. As pointed out by Rai and Dring in ref. [26], the most important differences between the calculations and the experiment are:

1. The actual flow is three-dimensional in nature whereas the present calculation is only two-dimensional.
2. The experimental configuration has 22 stator airfoils and 28 rotor airfoils and therefore the incoming wakes does not have the same spatial period as the rotor spacing.
3. In the experiment, the hot gas entered the system through only one stator passage. The periodicity condition used in the calculation produces one hot streak per rotor blade thus resulting in a much greater amount of hot gas entering the rotor. Another factor is that the hot streak in the experiment only existed over one third of the span while the two-dimensional code more closely simulates a hot streak existing over the whole span.

6.1 Fully turbulent flow with flow coefficient $\phi = 0.78$

To be consistent with the computation performed by Rai, the first test case is run with a flow coefficient $\phi = 0.78$. It is also assumed, to be consistent with Rai, that the flow around the blade is fully turbulent. This can be justified by the fact that the flow originates from the combustor and hence contains a high level of free stream turbulence. By considering the time averaged wall pressure distribution in [26] it is reasonable to assume that, even without the incoming hot streaks, the laminar boundary layer will become turbulent very quickly due to an adverse pressure gradient on the suction side and a pressure peak on the pressure side.

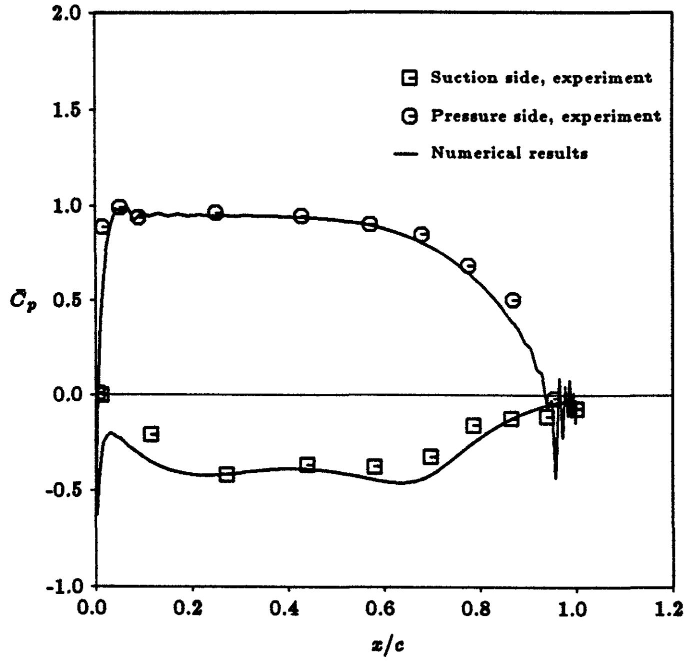

Figure 6.5: Time averaged surface pressure coefficient, $\phi = 0.78$

Pressure distribution

Figure 6.5 shows the time-averaged pressure coefficient ($\bar{C}_p$) as a function of the normalized axial distance x/c. The pressure distribution is defined as

$$\bar{C}_p = \frac{(p_{ave} - p_{ex})}{P_{0\,inl} - p_{ex}} \tag{6.1}$$

where p_{ave} is the static pressure time-averaged over one cycle. Fig. 6.5 also shows the experimentally obtained pressure distribution, which was obtained without any incoming hot streaks. The general agreement between computation and experiment is good. There are, however, some discrepancies in the leading edge region. The differences can be explained by the fact that the rotor relative flow angle is different in the hot and cold fluid. This can be understood by considering the velocity triangles shown in Fig. 6.6. In the figure the subscripts *hs* and *col* correspond to hot and cold fluid respectively and the superscripts *stat* and *rot* indicates stator and rotor relative velocities. Also indicated is the difference in flow angle $\Delta\alpha$. A different time averaged angle of attack will give rise to a different pressure distribution. The random fluctuations in the pressure at the

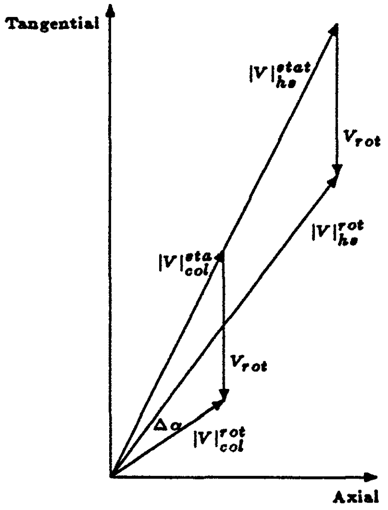

Figure 6.6: Velocity triangles for hot and cold fluid show difference in rotor relative flow angle

end of the blade correspond to the points on the trailing edge circle and are probably caused by numerical errors.

Temperature distribution

In Figure 6.7 is shown the time-averaged temperature coefficient $\bar{C}_T$ along the blade surface. The surface length is nondimensionalized with respect to the axial chord c. The temperature coefficient is defined as

$$\bar{C}_T = \frac{T - T_\infty}{T_{av} - T_\infty} \tag{6.2}$$

where T_{av} is the area averaged inlet temperature and T_∞ is the temperature in the unheated free stream. It can be seen that the temperature coefficient varies around 1 as expected. A peak in the temperature appears in the beginning of the pressure surface where the hot streak intersects with the blade. Following the pressure surface, the average temperature slowly rises from 1.15 to about 1.4 and finally collapses under 1 at the trailing edge. On the suction surface the average temperature reaches a maximum of about 1.15 at the crest of the blade and then linearly decreases to a value of 0.9 at the trailing edge.

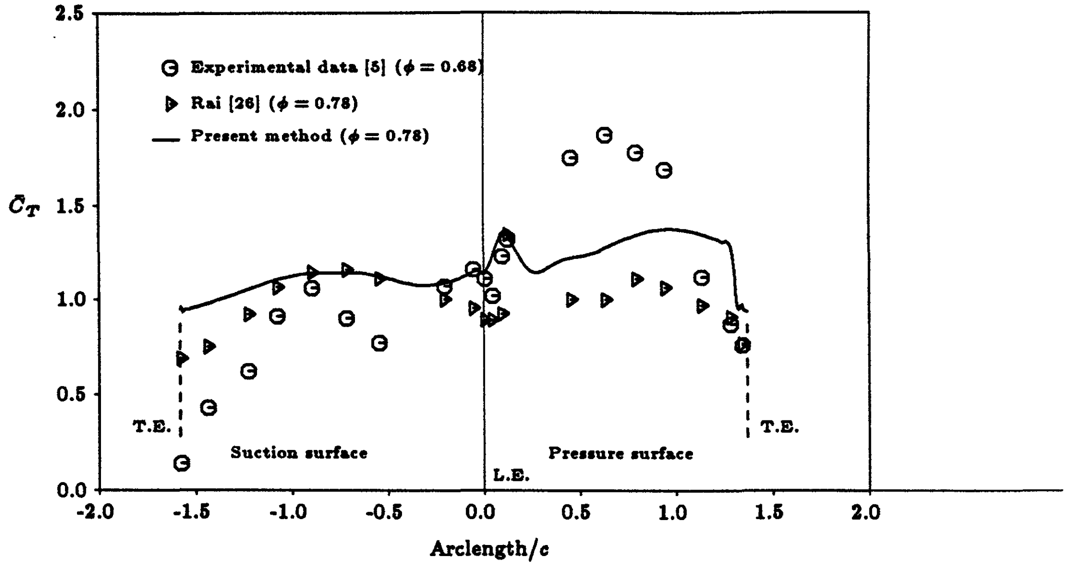

Figure 6.7: Time averaged surface temperature distribution, $\phi = 0.78$

Figure 6.7 also shows the numerical results of Rai and the experimental results. The results obtained with the present method are, on the suction surface, in close agreement with Rai's data. On the pressure surface, however, the present solution predicts a higher average temperature which indicates that a higher temperature ratio in the hot streak does indeed effect the computed flow field solution around the cascade. In comparison with the experiment, the present numerical solution show some major differences. With the flow conditions specified for this test case, the present method did not predict the same amount of migration as was experimentally observed.

Unsteady temperature distribution

The unsteady temperature distribution along the blade surface is shown in Figure 6.8. The figure shows the time evolution of the temperature distribution over two blade passages. In the figure the time is nondimensionalized with the cycle time. It can be seen that the variation between high and low temperature is more accentuated on the suction surface than on the pressure surface. This was also observed by Rai as he considered the absolute difference between max and min temperature $|T_{max} - T_{min}|$ as

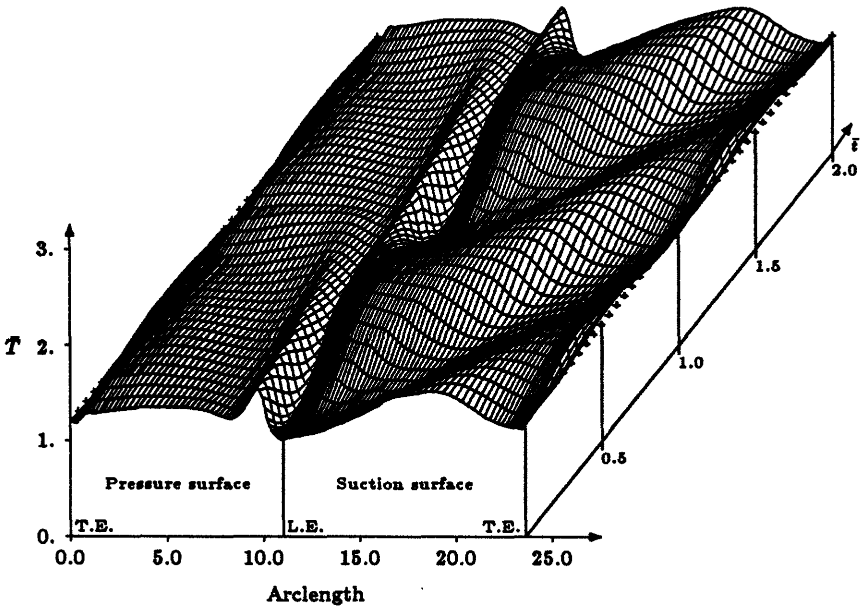

Figure 6.8: Time evolution of the surface temperature distribution, $\phi = 0.78$

a function of the blade surface. This phenomenon can be explained by the fact that the hot streak is convected much faster on the suction side than on the pressure side. Due to the fact that the diffusive time scale is much smaller than the convective, the boundary layer on the suction side can be considered to be quasi-steady and hence the temperature on the blade surface will be the same as the temperature at the edge of the boundary layer.

The hot streaks seems to attach at the leading edge and be slowly convected downstream on the pressure surface. This will force the hot streak outside the boundary layer to wrap around the blade and eventually mix with the cool fluid along the pressure side. This will account for the relatively steady temperature distribution found on the pressure surface. The way the temperature peak at the leading edge on the pressure surface decays in time, indicates that the hot streak is being diffused rapidly in the turbulent boundary layer. This is quite in order since the turbulence model will increase the thermal diffusion.

6.2 Fully turbulent flow with flow coefficient $\phi = 0.68$

In order to match the actual experiment more closely, the flow coefficient for this second test case is set to be 0.68 which corresponds to a higher rotational speed and hence a higher tangential inflow angle.

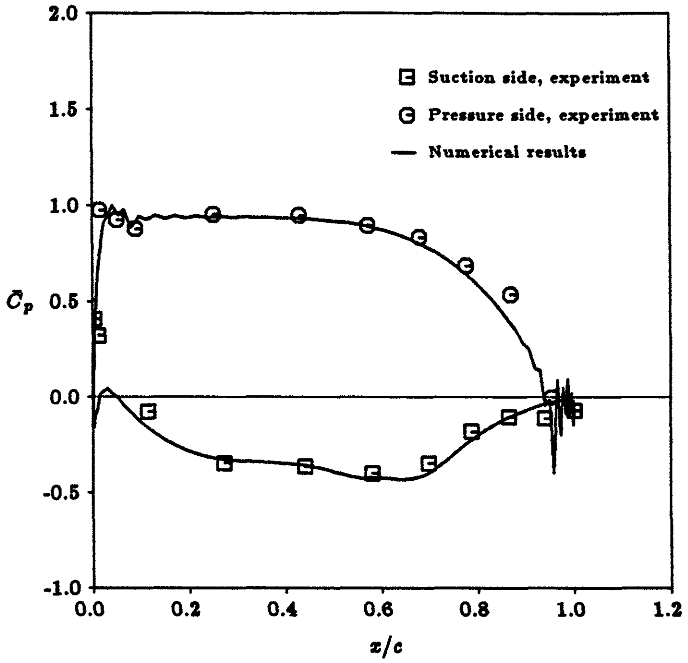

Figure 6.9: Time averaged surface pressure distribution, $\phi = 0.68$ fully turbulent

Pressure distribution

A comparison between the numerically and the experimentally obtained time-averaged pressure coefficient $\bar{C}_p$ is shown in Figure 6.9. The pressure coefficient was defined in Eq. (6.1). The experimental data was, as for the case with $\phi = 0.78$, obtained without incoming hot streaks. The agreement between experimental and computational data is good with a small difference in the leading edge region. As in the previous test case the differences can be explained by the differences in relative flow angles shown in Figure 6.6. The wiggles in the leading edge region of the pressure surface is due to a combination of poor mesh resolution and a minimum of artificial smoothing in the streamwise direction in the boundary layer.

Temperature distribution

Figure 6.10 shows the averaged temperature coefficient as a function of the rotor blade surface length. Near the leading edge on the pressure surface there is a peak in the temperature indicating the position where the hot streak first hits the blade. From this peak, the temperature increases almost linearly along the pressure surface. It starts at a value of ~ 1.25 behind the peak and ends at ~ 1.4 at the trailing edge circle where it drops to 1.0. On the suction surface between the leading edge and the crest of the blade, the temperature rises from about 1.10 to 1.15. From the crest, the temperature decreases linearly down to the trailing edge where it takes on the value of ~ 0.9.

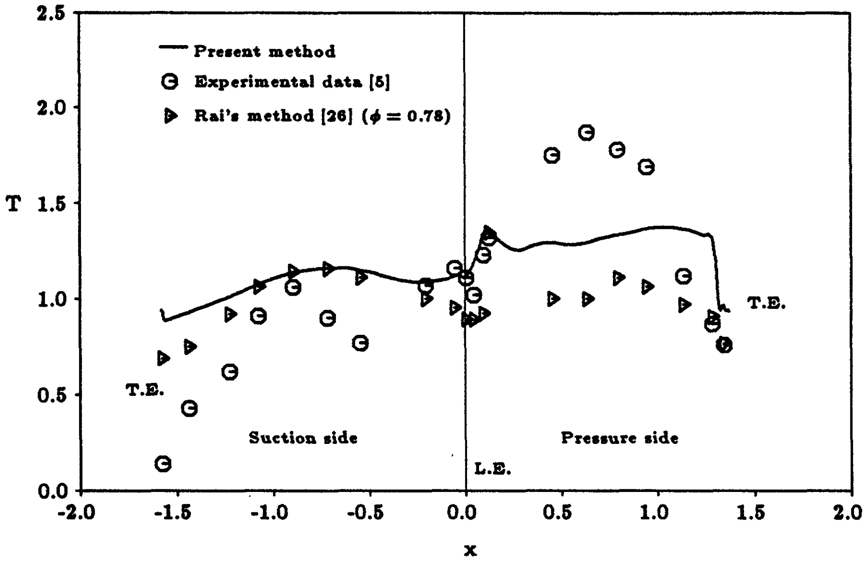

Figure 6.10: Time averaged surface temperature distribution, $\phi = 0.68$ fully turbulent

In Figure 6.10 is also shown the experimental data. The agreement between the experimental data and the computational results is, as in the previous case, still quite poor. The peak averaged temperature does not reach the same high value in the computation as in the experiment. However, in Fig. 6.10 one can notice an indication of more

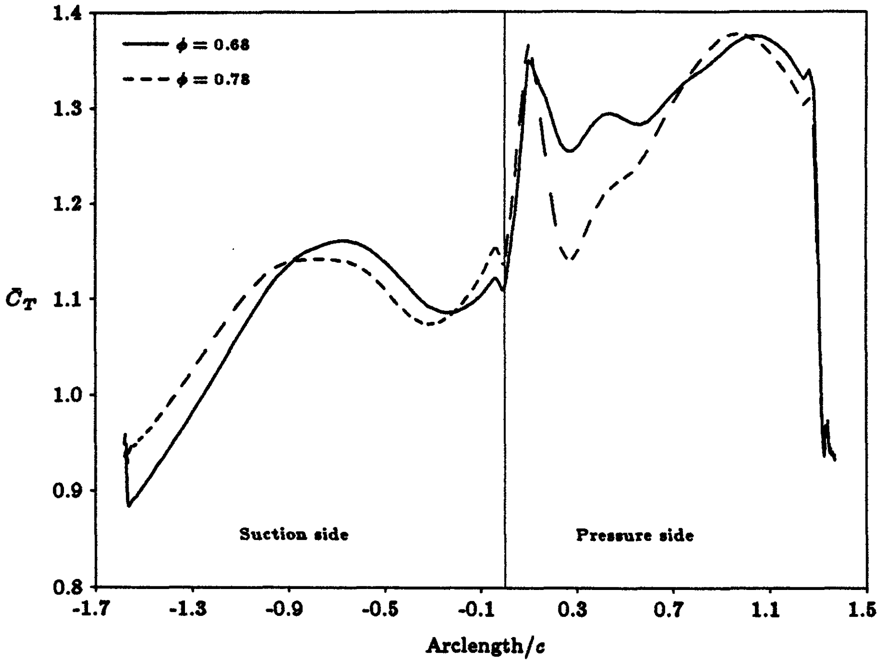

Figure 6.11: Time averaged surface temperature distribution

heat accumulation on the pressure surface. As pointed out also by Rai, the higher axial flow angle α for $\phi = 0.68$ will accentuate the the segregation of hot and cold gas between the pressure and suction surfaces. The differences between the two computations can clearly be seen in Figure 6.11 where an enlargement of the the average temperature distribution from the two cases is shown.

6.3 Transitional flow with flow coefficient $\phi = 0.68$

The effect of isolated wakes on a laminar and transitional boundary layer was investigated in an experiment by Doorly, Oldfield and Scrivner [8]. It was found that an isolated passing wake induced a rapid and complete transition of an otherwise laminar boundary layer to a temporarily fully turbulent state. The level of freestream turbulence in that specific test case was very low. This is of interest to the hot streak investigation since the original experiment by Butler only had one hot streak coming in from the stator and probably a very small amount of freestream turbulence. It is possible that, since the pressure peak is much weaker for this inflow angle, the flow on the pressure surface is laminar. It is also possible that the flow on the suction surface is transitional due to the strong pressure gradient.

In a second experiment, however, the freestream turbulence was increased to 2% intensity and the wakes were more closely spaced. It was found that the successive turbulent patches eventually merge to form a continuously turbulent boundary layer. This experiment is closer to a real flow situation which indicates that the fully turbulent assumption made in Section 6.1 and Section 6.2 is more physically correct.

Considering the time averaged pressure distribution obtained from the experiment, the transition point on the suction surface should be located about 65-70% of the axial chord from the leading edge. At this point the pressure gradient changes from favorable and becomes adverse. The transition point on the pressure surface was chosen just before the trailing edge circle in order for the flow to be turbulent around the trailing edge.

Temperature distribution

Figure 6.12 shows the averaged temperature coefficient as a function of the rotor blade surface length. Near the leading edge on the pressure surface there is a peak in the temperature indicating the position where the hot streak first hits the blade. From this

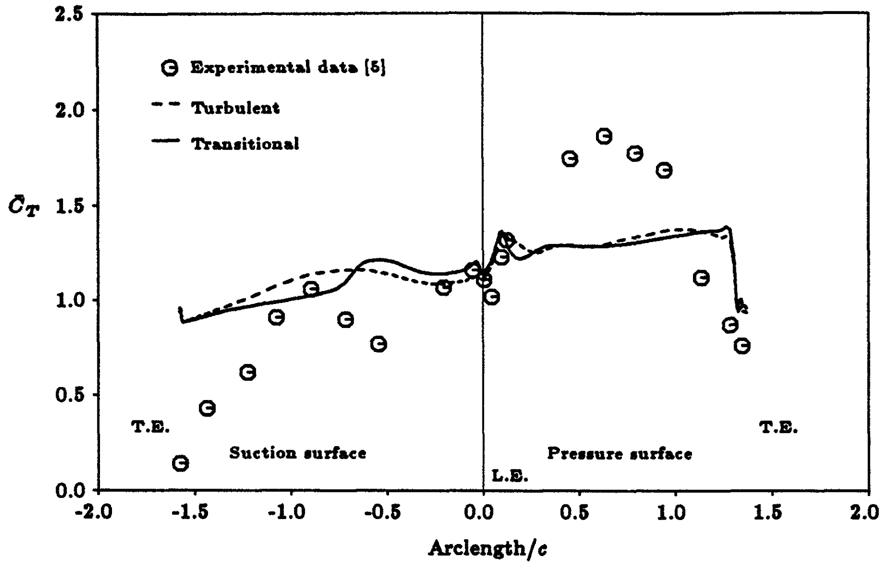

Figure 6.12: Time averaged surface temperature distribution for $\phi = 0.78$

peak, the temperature increases almost linearly along the pressure surface. It starts at a value of $\sim$ 1.25 behind the peak and ends at $\sim$ 1.4 at the trailing edge circle where it drops to 1.0. On the suction surface between the leading edge and the transition point the temperature increases from about 1.15 to 1.2. At the transition point the temperature coefficient suddenly drops to about 1.0 and then linearly decreases down to the trailing edge where it is $\sim$ 0.9.

In Figure 6.12 is also shown the experimental data and the results from the fully turbulent calculation. There is a significant difference between the two computational results. On the suction surface, the temperature is higher for the laminar case up to the transition point where it drops and becomes lower than in the turbulent case. The peak averaged temperature does not reach the same high value in the computation as in the experiment. It is remarkable to note that the temperature distributions on the pressure surface are so similar for the laminar and turbulent cases. Further, it is interesting to notice the drastic drop in temperature when the flow changes from laminar to turbulent. This numerically simulated temperature decrease due to transition, makes it reasonable to believe that it is transition that causes the experimentally observed temperature decrease at 0.2 arclength from the leading edge on the suction surface.

It is now clear that the turbulence model is a crucial part of the simulation and that the model affects the flow features. The choice of transition point is also of great importance. It can be discussed whether transition should be enforced or if the boundary layer should be fully turbulent. Based on the fact that the actual flow from a combustor contains a high level of freestream turbulence it is reasonable to believe that the boundary layer is fully turbulent. However, the combustor exit flow turbulence will not be homogeneous so it may only be possible to model it effectively by an true unsteady method.

Unsteady temperature distribution

In Figure 6.13 the time evolution of the surface temperature distribution is shown. At the leading edge can be observed the periodic impact of the hot streak on the blade. The behavior of the temperature is more dramatic on the suction surface than on the pressure surface. Due to a higher convection speed, the differences between maximum and minimum temperatures are larger on the suction surface. It can be seen that the temperature differences are much smaller at the transition point on the suction side. The thermal diffusivity increases after the transition point and the high and low temperature parts of the gas mix faster which is in agreement with the theory. It is interesting to note that, with the exception of the leading edge region, the temperature on the pressure surface stays almost constant during the whole cycle.

Temperature contours

One of the big advantages of using a computational approach is that it yields, as a part of the calculation, not only surface temperatures but also the temperature distribution in the entire computational region. The figure sequence 6.15 - 6.18 shows four snapshots of the temperature field in the rotor at four different time stages in the cycle. These are at 25, 50, 75 and 100 % of the cycle time. In all figures the temperature is represented by iso-clines where the lower temperature levels have been filtered out. The hot streaks

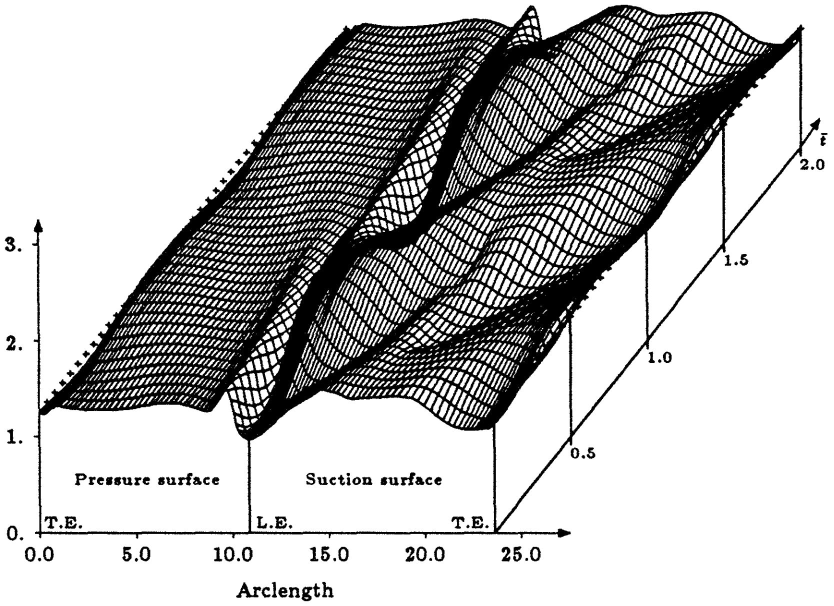

Figure 6.13: Time evolution of surface temperature, $\phi = 0.68$

are in between the two narrow bands of iso-clines representing the transition between hot and cold gas.

Figure 6.18 shows the solution at time level 1.0 which is equal to the solution at time level 0.0. Five different phases of the hot streak redistribution can be identified. The first is the left most stage in the inflow where the hot streak is being convected towards the blade row, indicated by A in the figure. The second stage is the one close to the leading edge where the hot streaks have started to interact with the pressure perturbation from the blade. A first indication of bending of the streak can be noticed at station B in the figure. In the third stage the hot streak has impinged on the pressure surface of the blade (station C) and is slowly being convected along the surface. Due to the acceleration gradient in the high speed region between the pressure and suction surface of two consecutive blade, the streaks starts to deform and form a V shape (D) which is referd to as bowing. In stage four the hot streak has been wrapped around the blade and the outer contours downstream of the trailing edge are being effected

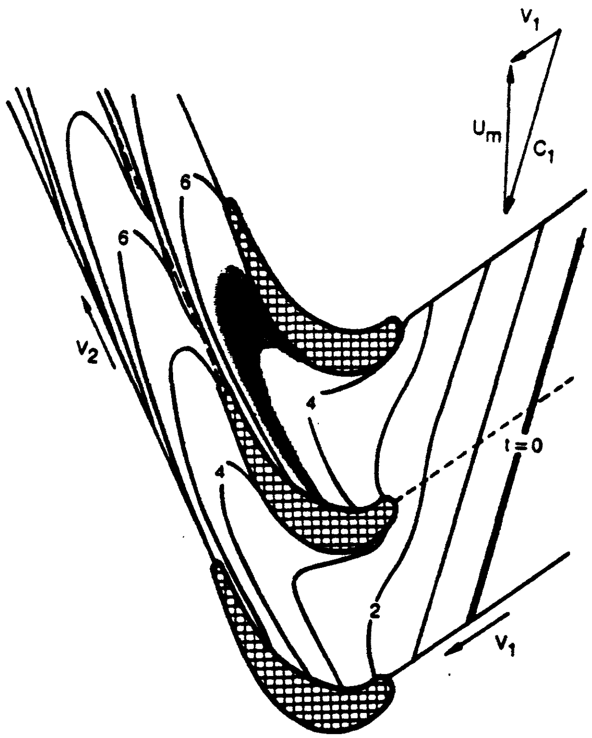

Figure 6.14: Inviscid wake transport in the UTRC rotor at $\phi = 0.34$

by the vortex shedding at the trailing edge (E). At this stage it can be seen that the redistribution of the hot streak is dominated by convection on the suction surface and by a combination of convection and heat diffusion on the pressure surface. Finally in stage five the chopped remainder of the hot streak is being convected downstream and out of the computational domain (F).

An inviscid wake transport phenomenon was investigated by Joslyn, Caspar and Dring [17]. They used the same geometry as in this work, to investigate the inviscid transport of an infinitesimal wake velocity defect through the rotor. The results were compared to a smoke trace experiment and showed good agreement. The flow coefficient in their computation was much smaller than in the present calculation ($\phi = 0.34$). However, the same qualitative flow features such as wake chopping and wake bowing were observed in both Joslyn's calculation and the present calculation. Figure 6.14 that was taken out of reference [17] shows Joslyn's calculated wake transport. In comparison with the figure series 6.15 to 6.18 it can be seen that the wake on the pressure surface is convected much faster in the inviscid case than in the viscous.

By considering the sequence of solutions illustrated in Figures 6.15 - 6.18 it is easy to see that the hot streaks are being dissipated as they convect downstream even in regions where viscous effects were assumed to be negligible. This indicates that the central, finite volume scheme strongly dissipates the solution in flow regions with strong gradients. The dissipation of the hot streak leads to a decrease in peak temperature in the hot streak. The fact that the hot/cold temperature ratio was not constant at 2.0 in the upstream region, might contribute to the absence of the experimentally measured temperature peak on the pressure surface (see Figure 6.12). In a real flow situation, however, the high level of freestream turbulence will indeed cause dissipation of the hot streak.

A considerable amount of smoothing and too few mesh points around the trailing edge is the most probable reason that the vortex shedding in this simulation is weaker than the one observed by Rai.

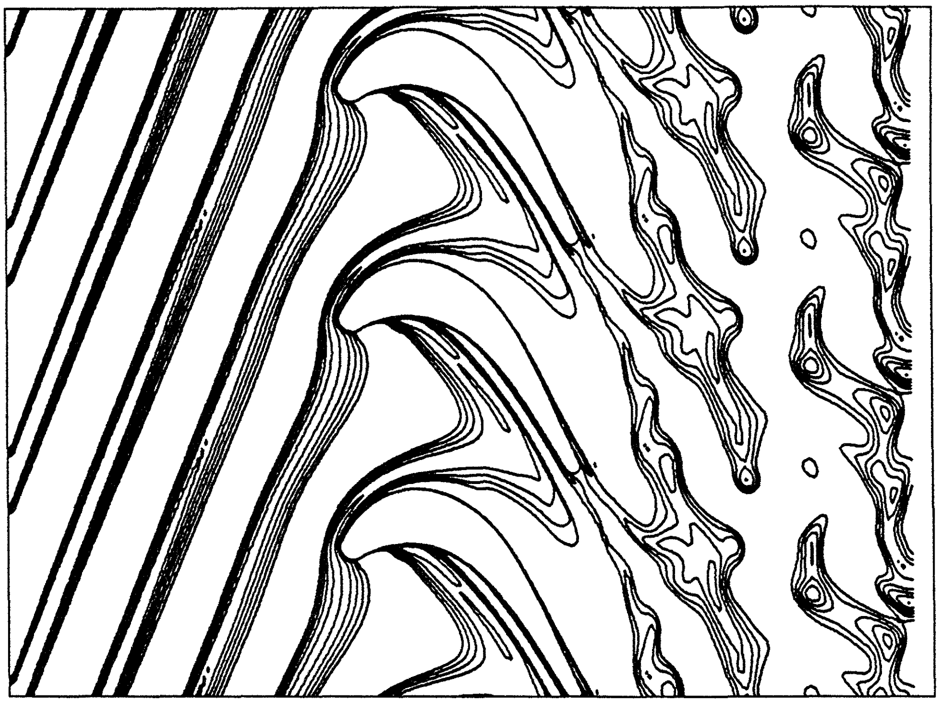

Figure 6.15: Temperature contours, $t = 0.25$

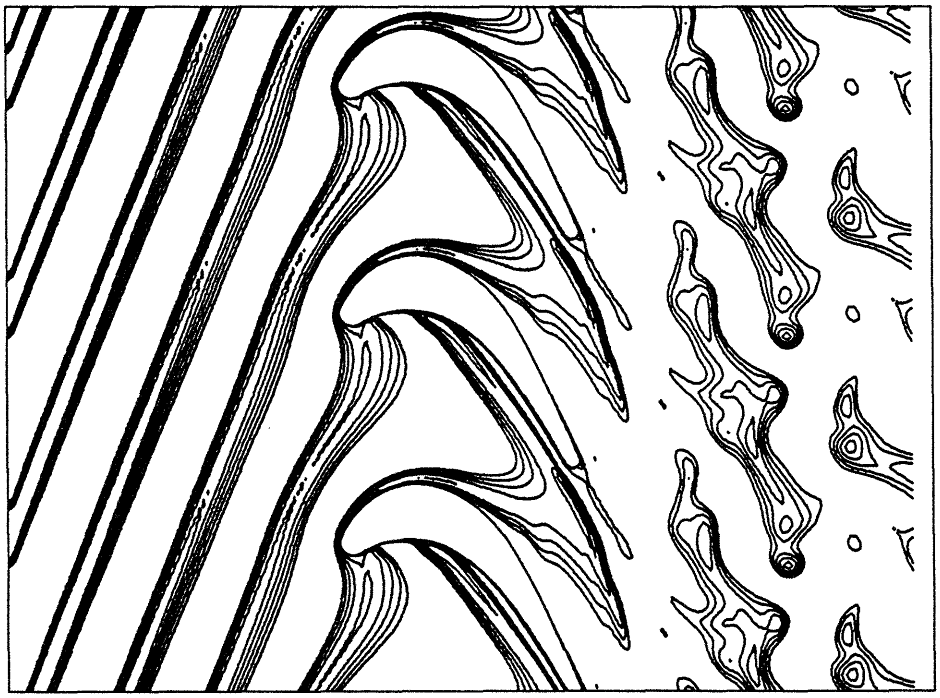

Figure 6.16: Temperature contours, $t = 0.5$

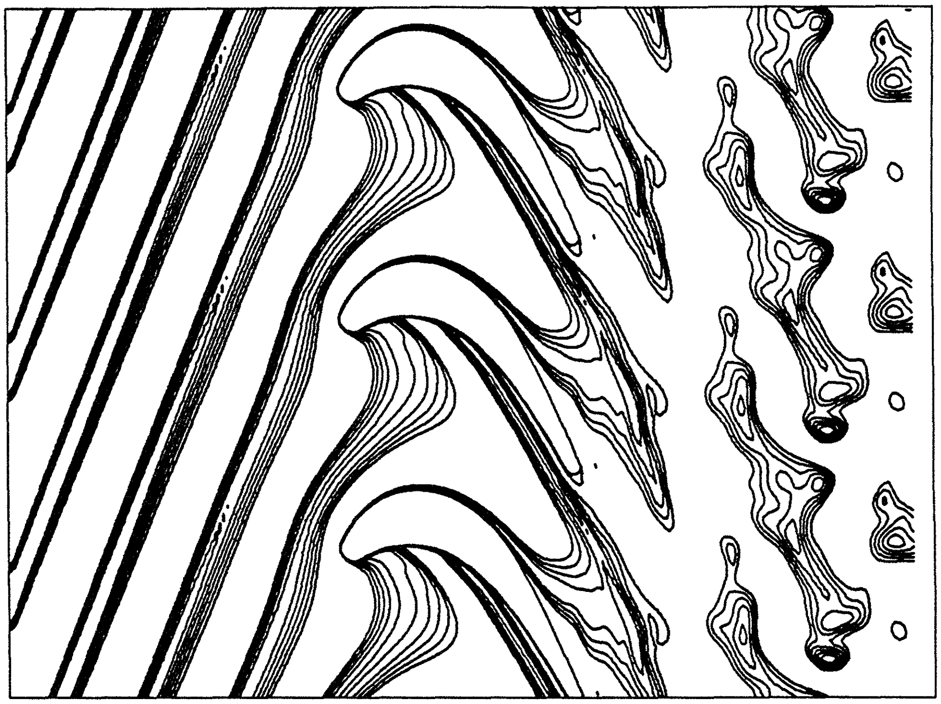

Figure 6.17: Temperature contours, $t = 0.75$

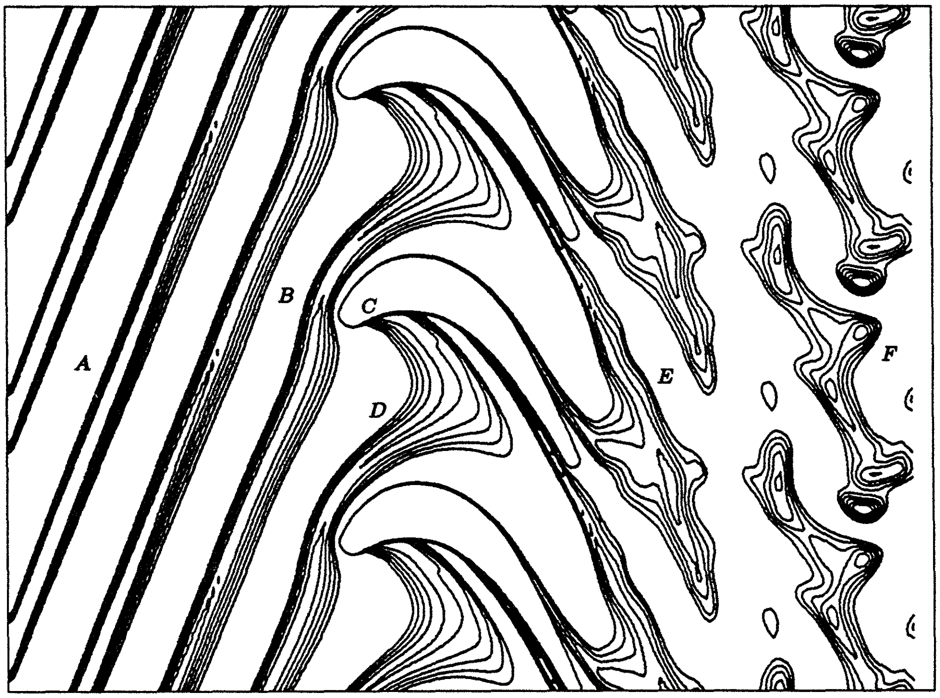

Figure 6.18: Temperature contours, $t = 1.0$

Chapter 7
Conclusions

In this work a numerical algorithm for simulating an unsteady, two dimensional flow in a turbine rotor have been developed. The basic philosophy in the development of the code was to divide the flow region into sub-regions with different flow characteristics. In regions where viscous effects dominated the flow, the unsteady Reynolds-averaged thin layer Navier-Stokes equations are solved. In turbulent regions a Baldwin-Lomax turbulence model is used to model the turbulent flow characteristics. The unsteady Euler equations are solved in regions where viscous effects were assumed to be negligible. The objective of this study was two-fold:

- To assess the possibilities of using a mixed viscous-inviscid, implicit-explicit numerical scheme for unsteady flow simulations on patched grids.
- To study the redistribution of hot streaks in turbines with emphasis on the migration of hot gas to the pressure side of the rotor blade.

On each different flow region a structured computational mesh is defined. The different meshes are patched together to create a complete computational mesh. A standard finite volume technique is used to solve the spatial part of the governing equations both in the inviscid and the viscous regions. In the viscous region an implicit Beam-Warming time-integration scheme is used to accurately advance the flow solution in time. An explicit three-stage Runge-Kutta method is used for time integration in the inviscid region. An accuracy study shows that the two numerical methods are both spatially and temporally second order accurate. A Von-Neumann stability analysis shows that the implicit scheme is unconditionally stable for a mixed hyperbolic/parabolic model equation, and that the explicit scheme is conditionally stable with a Courant number of 2 for a hyperbolic model equation. Unsteady non-reflecting boundary conditions are used in order to allow waves to pass through computational boundaries without being reflected. These boundary conditions also allow prescribed non-uniformities such as velocity wakes and hot gas streaks to enter the computational domain. Calculations on different test geometries under different flow conditions gives results which are in excellent agreement with experiments, previous calculations and analytic solutions.

For a turbine rotor the two dimensional computational region is divided into a viscous region close to the blade where an O-mesh is generated, and an inviscid region in the through flow region where a H-mesh is generated. The H-mesh is created using an elliptic grid generator and the O-mesh using an algebraic technique based on splines. The two meshes are patched together and the interface region is smoothed in order to prevent numerical errors.

The two outermost layers of computational cells in each mesh configuration constitute the interface region between the viscous and the inviscid region. The interface is transparent to the finite volume operator since the structure of the operator is identical on the different meshes. The interface is likewise transparent to the fourth-order dissipation operator. To achieve time accuracy the flow field is solved simultaneously on the different regions. In the interface region the temporal integration is locally first order accurate. This, however, does not affect the global second order accuracy of the scheme.

The numerical method is used to investigate the redistribution of hot streaks in a turbine rotor. A two dimensional model of a UTRC turbine rotor is used as a test geometry. An experiment and a previous calculation have been performed on this geometry, regarding the transport of hot streaks. In the first calculation the flow coefficient was $\phi = 0.76$, the temperature ratio between hot and cold gas was 2.0 and the flow around the rotor blade was assumed to be fully turbulent. A comparison between results obtained from a previous calculation and results generated by the present code showed favorable agreement. The time-averaged pressure distribution calculated for this case was correctly predicted by the code. The migration of hot gas to the pressure surface is predicted for these flow conditions.

To match the original experimental setup more closely the second and third test cases are run with a flow coefficient of $\phi = 0.68$ and a temperature ratio of 2.0. The prediction of a larger hot gas migration due to a higher axial flow angle was verified by the two calculations. As in the previous case, the pressure distribution along the blade is correctly predicted.

In the third test case the effect of transition was studied. The pressure surface was forced to be laminar and on the suction surface a transition point was chosen at a position where the pressure gradient becomes adverse. At the transition point the temperature decreases rapidly which is in agreement with the experiment. It is the author's opinion that the magnitude of hot gas migration to the pressure surface is significantly dependent on the differences in flow coefficient. Whether the boundary layer is fully turbulent or transitional is precluded from playing any significant role of the migration. It is also believed that some of the differences between experiment and computation are due to the three dimensional nature of the flow which can not be simulated by a 2D code.

For future work the following items are suggested:

- Development and implementation of a more accurate turbulence model.
- Perform an inviscid calculation under the same conditions to establish the importance of viscous effects on the migration of hot gas.
- Extension of the numerical code to be able cover a full turbine stage with stator and rotor.

Bibliography

[1] Agarwal, R. K. and Deese, J. E. "Computation of Transonic Viscous Airfoil, Inlet and Wing Flowfields", AIAA-84-1551 Jun. 1984.

[2] Baldwin, B.S. and Lomax, H. "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows", AIAA-78-257 Jan. 1978

[3] Beam, R. and Warming, R. "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations", AIAA Journal, Vol. 16, No 4, April 1978

[4] de Boor, C. "A Practical Guide to Splines", Springer Verlag, 1978

[5] Butler, T.L., Sharma, O.P., Joslyn,H.D. and Dring, R.P. "Redistribution of an Inlet Temperature Distortion in an Axial Flow Turbine Stage", AIAA-86-1468 June 1986

[6] Chima, R. V. "Efficient Solutions of the Euler and Navier-Stokes Equations with a Vectorized Multiple-Grid Algorithm", AIAA Journal, 23:23-32, Jan. 1985

[7] Davis, .R, Ni, R-H., and Carter, J "Cascade Viscous Flow Analysis Using the Navier-Stokes Equations", AIAA-86-0033 Jan. 1986

[8] Doorly, D. J., Oldfield, M. L. G. and Scrivener, C. T. J. "Wake Passing in a Turbine Rotor Cascade" AGARD CP-390, May, 1985

[9] Eriksson, L.-E. and Rizzi, A. W. "Computation of Vortex Flow Around Wings Using the Euler Equations", ed. H. Viviand, Proc. of the Fourth GAMM-Conf. on Numerical Meth. in Fluid Mech., Vieweg Verlag, 1982

[10] Eriksson, L-E. "Simulation of Inviscid Flow Around Airfoils and Cascades Based on the Euler Equations", FFA TN 1985-20, 1985

[11] Eriksson, L-E. "Transfinite Mesh Generation and Computer-Aided Analysis of Mesh Effects", PhD. Thesis, Uppsala University, Department of Computer Sciences, March 1984

[12] Giles, M. B. "UNSFLO: A Numerical Method For Unsteady Inviscid Flow in Turbomachinery", MIT Report, CFDL-TR-86-6, Dec. 1986

[13] Giles, M. B. "Calculation of Unsteady Wake/Rotor Interactions", AIAA-87-0006 Jan. 1987

[14] Giles, M. B. "Newton Solution of Steady Two-Dimensional Transonic Flow", PhD. Thesis, Dep. of Aeronautics & Astronautics, MIT, June 1985

[15] Gustafsson, B. "The Convergence Rate for Difference Approximations to Mixed Initial Boundary Value Problems", Math. Comp., Vol. 29, No. 130, pp 396-406, Apr. 1975

[16] Jameson, A. "Solution of the Euler Equations for Two Dimensional Transonic Flow by a Multigrid Method", Princeton University MAE Report No. 1613, June 1983

[17] Joslyn, H.D.,Caspar, J.R. and Dring, R.P. "Inviscid Modelling of Turbomachinery Wake Transport" ,J. of Propulsion and Power, vol 2, no.2, March-April 1986

[18] Kallinderis, Y. and Baron, J. "Adaptation Methods for a New Navier-Stokes Algorithm", AIAA 87-1167-CP, June 1987

[19] Kerrebrock, J. L. and Mikolajczak, A.A. "Intra-Stator Transport of Rotor Wakes and its Effect on Compressor Performance", ASME Journal of Engineering for Power, Oct. 1970

[20] Müller, B. and Rizzi, A.W. "Runge-Kutta Finite-Volume Simulations of Laminar Transonic Flow Over a NACA0012 Airfoil Using the Navier-Stokes Equations", FFA TN 1986-60, 1986

[21] Nakahashi, K. and Obayashi, S. "Viscous Flow Computations Using a Composit Grid", AIAA paper AIAA-87-1128, June 1987

[22] Nakahashi, K., Nozaki, O., Kikuchi, K. and Atsuhiro, T. "Navier-Stokes Computations of Two- and Three Dimensional Cascade Flow Fields", AIAA paper AIAA-87-1315, June 1987

[23] Ni, R. H. "A Multiple-Grid Scheme for Solving the Euler Equations", AIAA J., Vol 20, No 11, Nov 1981, pp 1565-1571

[24] Norton, R. J. G., Thompkins, W. T. and Haimes, R. "Implicit Finite Difference Schemes with Non-Simply Connected Grids - A Novel Approach", AIAA paper AIAA-84-0003, January 1984

[25] Peyret, R and Taylor, T.D. "Computational Methods for Fluid Flow", Springer-Verlag, 1983

[26] Rai, M. M. and Dring, R.P. "Navier-Stokes Analyses of the Redistribution of Inlet Temperature Distortions in a Turbine", AIAA-87-2146, 23rd Joint Propulsion conference, 1987.

[27] Rizzi, A. and Eriksson, L.-E. "Computation of Flow Around Wings Based on the Euler Equations", J. Fluid Mech., Vol. 148, pp. 45-71, 1984.

[28] Schlichting, H. "Boundary-Layer Theory", seventh edition, McGraw-Hill, 1979

[29] Smith, S., N. "Discrete Frequency Sound Generation in Axial Flow Turbomachines", University of Cambridge, Department of Engineering Report CUED/ATurbo/TR 29, 1971

[30] Stabe, R. G., Whitney, W. J. and Moffitt, T. P. "Performance of a High Work Low Aspect Ratio Turbine Tested with a Realistic Inlet Radial Temperature Profile", AIAA-84-1161, June 1984

[31] Stewart, G., W. "Introduction to Matrix Computation", Academic Press, 1973.

[32] Swanson, C. and Turkel, E. "A Multi-stage Time-stepping Scheme for the Navier-Stokes Equations", AIAA-85-0035, Jan 1985

[33] Thompson, J. F., Thames, F. C. and Mastin, C. W. "Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies", Journal of Computational Physics, Vol. 15, pp 299-319,1974

[34] Trefethen, L., N. "A Course in Finite Difference and Spectral Methods", To be published,

[35] White, F.M. "Viscous Fluid Flow", McGraw-Hill, 1974

[36] Whitehead, D., S. "LINSUB User's Guide"