

DEVELOPMENT OF A BLADE TO BLADE SOLVER FOR AXIAL
TURBOMACHINERY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST OF TECHNICAL UNIVERSITY

BY
MUSTAFA BİLGİÇ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

SEPTEMBER 2015

Approval of the thesis:

**DEVELOPMENT OF A BLADE TO BLADE SOLVER FOR AXIAL
TURBOMACHINERY**

submitted by **MUSTAFA BİLGİÇ** in partial fulfillment of the requirements for the
degree of **Master of Science in Mechanical Engineering Department, Middle
East Technical University** by,

Prof. Dr. Gülbin Dural Ünver
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Tuna Balkan
Head of Department, **Mechanical Engineering**

Prof. Dr. Mehmet Haluk Aksel
Supervisor, **Mechanical Engineering Dept., METU**

Examining Committee Members:

Prof. Dr. Kahraman Albayrak
Mechanical Engineering Dept., METU

Prof. Dr. Mehmet Haluk Aksel
Mechanical Engineering Dept., METU

Assoc. Prof. Dr. M. Metin Yavuz
Mechanical Engineering Dept., METU

Asst. Prof. Dr. Cüneyt Sert
Mechanical Engineering Dept., METU

Asst. Prof. Dr. Ö. Uğraş Baran
Mechanical Engineering Dept., TED University

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Mustafa Bilgiç

Signature :

ABSTRACT

DEVELOPMENT OF A BLADE TO BLADE SOLVER FOR AXIAL TURBOMACHINERY

Bilgiç, Mustafa

M.Sc., Department of Mechanical Engineering

Supervisor: Prof. Dr. M. Haluk Aksel

September 2015, 140 pages

In this thesis, a blade to blade solver for axial turbomachinery is developed. Actual blade to blade surface is three dimensional but it is reduced to two dimensions. This simplification is valid only for the turbomachinery having high solidity. Two different approaches are used for the solution of the two dimensional Euler equations through blade to blade streamsurface. The first one includes the solution of the steady form of the two dimensional Euler equations. The characteristics of the steady Euler equations depend on the local Mach number. Normally, the steady solver does not have capability of the solving transonic flows but implementation of the artificial compressibility to the solver provides the capability of the solving transonic flow. The flow equations are discretized on the intrinsic streamline grid so that there is no mass flux across the streamline grid. Since the grid is an actual streamline, the grid nodal displacement becomes an additional unknown to the thermodynamic variables. After discretization, Newton-Raphson linearization is applied to momentum equations which reduces the number of unknowns to two. Since the characteristic of the equation is elliptic, physical boundary conditions are applied to the inlet and outlet of the domain. The nodal displacement at the wall is set to zero as the wall boundary condition. At the stagnation streamlines, periodicity is given as the pressure

equivalence, that is, the pressure values on the upper and lower stagnation streamlines are forced to be identical. The resulting linear system is solved with modified tri-diagonal matrix algorithm. The second method includes the solution of the unsteady form of the governing Euler equations. The characteristic of the flow equation for all speed regimes is hyperbolic in time so transonic flow is covered without any additional procedure. The flow equations are discretized on a fixed structured polyhedral grid. The fluxes through the faces of the computational cells are calculated using flux difference splitting schemes. The right and left state of the cell faces, which are used in the flux calculation, are calculated using MUSCL (Monotonic Upstream-Centered Scheme for Conservation Laws) approach. Modified Van Albada limiter function is used to prevent the oscillations at the shocks for second order calculations. Characteristic inlet and outlet boundary conditions are applied at the inlet and outlet of the domain. The slip conditions are applied on the wall and ghost cell concept is used for the calculation of the fluxes at the periodic boundaries. The conservative variables at the new time level is evaluated using multistage Runge-Kutta time integration scheme. The same source terms for rotor case are implemented to both solvers. Both time marching solver and Newton solver are validated comparing with the analytical and/or experimental test case. The subsonic and transonic flow over the bump, flow inside the converging diverging nozzle, flow through turbine and compressor cascades and flow through R030 rotor blade are the test cases for the validation of the codes.

Keywords: Turbomachinery, Blade to Blade Surface, Euler Equations, Artificial Compressibility, Streamline grid, Newton Linearization, MUSCL Interpolation, Roe, RHLL, Upwind Scheme

ÖZ

İKİ KANATÇIK ARASI AKIŞI ÇÖZMEK AMACIYLA BİR ÇÖZÜCÜNÜN GELİŞTİRİLMESİ

Bilgiç, Mustafa

Yüksek Lisans, Makine Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. M. Haluk Aksel

Eylül 2015, 140 sayfa

Bu tezde iki kanatçık arası akışı çözmek amacıyla bir Euler çözücüsü geliştirilmiştir. Gerçek hesaplama yüzeyi üç boyutlu olsa da bazı varsayımlarla hesaplama yüzeyi iki boyuta indirgenmiştir. Bu varsayımlar kanatçıkların yüksek sıklıkla dizildiği dönen makinalarda geçerlidir. İki boyutlu kanatçık arası akışı çözmek için iki farklı yaklaşım kullanılmıştır. İlk yaklaşım zamandan bağımsız Euler denklemlerinin çözümünü içerir. Bu denklemlerin karakteri bölgesel Mach sayısına bağlıdır. Normalde bu tip çözücüler ses altı ve ses üstü akışların birlikte olduğu durumları çözemez ama çözücüye suni sıkıştırma eklenerek bu problemin üstesinden gelinmiştir. Akış denklemleri gerçek akış çizgilerinin oluşturduğu ağların üzerinde çözüldüğü için akış çizgilerine dik yönde kütle akısı olmamaktadır. Akış çizgileri aslında çözüm yüzeyi olduğu için ağ noktalarının pozisyonları değişken olmakta ve termodinamik bilinmeyenlere bilinmeyen olarak eklenmektedir. Ayırıklaştırma gerçekleştirildikten sonra Newton doğrusallaştırma yöntemi uygulanarak bilinmeyen sayısı ikiye düşürülmektedir. Giriş ve çıkış koşullarında fiziksel sınır koşulları uygulanmaktadır. Duvar şartı sadece duvar üzerindeki noktaların sabitlenmesi ile olmaktadır. Kanatçığın girişinde ve çıkışında periyodik sınır koşulu uygulanmaktadır. En son oluşan doğrusal denklem sistemi tri-diagonal matris algoritmasıyla çözülmektedir. İkinci yöntem

zamana baęlı Euler denklemlerinin özümünü eriyor. Denklemin karakteri hız rejiminden baęımsız olarak zamanda sürekli hiperbolik olduęu için ses altı ve ses üstü akışın beraber olduęu durumları özebilmektedir. Akış denklemleri sabit yapısal ökyüzlü aęda ayrıklaştırılıyor. Yüzeydeki akılar akı farkı ayırma yöntemlerinden biri kullanılarak hesaplanmaktadır. Yüzeydeki akıyı hesaplamak için yüzeyin saę ve sol tarafındaki durumlar MUSCL yaklaşımı kullanarak hesaplanmaktadır. Giriş ve ıkış sınır koşullarında karakteristik yöntem kullanılmaktadır. Duvarı kayma koşulu kullanılırken periyodik sınırlarda hayalet hücre kavramı kullanılmaktadır. Bir sonraki zaman dilimindeki deęişkenler Runge-Kutta zaman birleştirmesi kullanılarak hesaplanmaktadır. Kaynak terimler iki özücü içinde aynı mantıkla eklenmektedir. İki özücüde literatürdeki analitik ve deneysel sonuçlarla doęrulanmıştır.

Anahtar Kelimeler: Turbomakine, S2 Yüzeyi, Euler Denklemleri, Yapay Sıkıştırılabilirlik, Akış izgileri Aęı, Newton Doğrusallaştırması, MUSCL Ara Deęer Bulma yöntemi, Roe, RHLL, Upwind Şemaları

TO MY FAMILY

ACKNOWLEDGMENTS

I wish to thank to my supervisor Prof. Dr. M. Haluk Aksel for his guidance and support throughout my thesis and in my graduate studies.

I would like to thank to Emrah Güllü, Mert Erk, Gökhan Aran and Taylan Ercan for their technical support and advice in my thesis.

Also I would like to thanks to Cansu Yılmaz and Olcay Filiz for their endless support and friendship in my graduate study years.

Finally I would like to thank to my family for their patience, encouragement and supports throughout my educational life.

TABLE OF CONTENTS

ABSTRACT.....	v
ÖZ.....	vii
LIST OF FIGURES.....	xiii
LIST OF TABLES	xvii
LIST OF SYMBOLS	xviii
CHAPTERS	
1. INTRODUCTION.....	1
1.1. Review of Literature	6
1.2. Review of Literature	9
2. DEFINITION OF THE BLADE TO BLADE STREAM SURFACE	11
2.1. Characteristics of flow in Blade to Blade Surface.....	16
2.2. Inputs and Outputs of the Blade to Blade Solver	17
3. LINEARIZED EULER SOLVER.....	21
3.1. Geometric Parameters and Equation Discretization.....	21
3.2. Artificial Compressibility	36
3.3. Boundary Conditions.....	36
3.3.1. Duct Boundary Conditions.....	37
3.3.2. Boundary Conditions for Blade to Blade Domain	40
3.4. Global Equation and Variables.....	42
3.5. Newton Raphson Method and Linearization Procedure.....	43
3.5.1. Interior Euler Equation Linearization	45
3.5.2. Linearization of Boundary Conditions	57
3.6. Jacobian Matrix Structure.....	61

3.7.	Solution of the Newton System.....	67
3.8.	Updating Variables.....	69
4.	TIME MARCHING EULER SOLVER.....	71
4.1.	Discretization	73
4.2.	Convective Flux Calculation.....	78
4.3.	Boundary Conditions.....	85
4.3.1.	Solid Wall Boundary Condition.....	85
4.3.2.	Inlet and Outlet Boundary Conditions	92
4.3.3.	Periodic Boundary Condition	95
4.4.	Temporal Discretization.....	96
4.4.1.	Determination of Time Step.....	97
5.	TEST CASES AND SOLVER VALIDATION	99
5.1.	Mesh Independency.....	99
5.2.	Validation	104
5.2.1.	Subsonic Ni Bump	104
5.2.2.	Transonic Ni bump	107
5.2.3.	Transonic Converging Diverging Nozzle	109
5.2.4.	Hobson Cascade.....	112
5.2.5.	NACA 0012 Cascade.....	114
5.2.6.	Sanz Subcritical Turbine Cascade	117
5.2.7.	Sanz Supercritical Turbine Cascade	120
5.2.8.	Sanz Supercritical Compressor Cascade.....	122
5.2.9.	R030 Rotor.....	125
6.	CONCLUSION AND DISCUSSION.....	129
6.1.	Conclusion.....	129
6.2.	Comparison of Two Solvers.....	131

6.3. Future Work.....	132
REFERENCES.....	134

LIST OF FIGURES

FIGURES

Figure 1.1 Compressor Blade Row Flow Field Characteristic	3
Figure 1.2 Turbine Blade Row Flow Field Characteristic	3
Figure 1.3. Through Flow Stream Surface.....	5
Figure 1.4 Blade to Blade Stream Surface	5
Figure 1.5 Quasi 3-D Design Flow Chart	4
Figure 2.1 Blade to blade stream surface	11
Figure 2.2 Real and approximate blade to blade surface	12
Figure 2.3 Relation between S1 and S2 streamsurfaces	13
Figure 2.4 Relation between the tangential, axial and meridional directions	13
Figure 2.5 Definition of the Geometric Parameters	15
Figure 2.6 Decelerating and accelerating flow domains	16
Figure 2.7 Throughflow computation stations.....	17
Figure 2.8 Input and Output of the solvers	18
Figure 3.1 Structured mesh on a NACA 0012 cascade flow domain	22
Figure 3.2 Unit vectors on r-x plane	24
Figure 3.3 Typical computational cell, location of the grid nodes and face centers .	25
Figure 3.4 Definition of vectors $\vec{A}_1, \vec{A}_2, \vec{B}$	26
Figure 3.5 Location of flow variables or midpoint of the cell faces	27
Figure 3.6 Normal area vector components	30
Figure 3.7 Pressure compatibility between two adjacent streamtubes	33
Figure 3.8 Duct boundary conditions.....	37

Figure 3.9 Blade to blade boundary conditions.....	40
Figure 3.10 Indices for a computational molecule	61
Figure 3.11 Structure for Jacobian matrix implementation	63
Figure 3.12 Structures of A_1 and C_1	64
Figure 3.13 Structures of A_i and C_i ($2 \leq i \leq II - 1$)	64
Figure 3.14 Structures of Z_i and B_i ($2 \leq i \leq II - 1$)	65
Figure 3.15 Structures of Z_I and B_I	65
Figure 3.16 U_1 and G_1 matrices	66
Figure 3.17 Non-zero entries for the U and G matrices	66
Figure 4.1 Definition of the finite control volume	72
Figure 4.2 Typical cell centered control volume	73
Figure 4.3 Decomposition of face normal	81
Figure 4.4 Solution of transonic flow over NACA 0012 blade profile with and without limiter	82
Figure 4.5 Computational stencil for left and right state calculations	82
Figure 4.6 Physical and Computational Space.....	86
Figure 4.7 Stencil for the calculation of metrics	88
Figure 4.8 Periodic boundaries	95
Figure 4.9 Cell Geometry for CFL calculation	98
Figure 5.1a Flow domain with 33 x 9 number of grid points	101
Figure 5.1b Flow domain with 65 x 17 number of grid points	101
Figure 5.1c Flow domain with 129 x 33 number of grid points	101
Figure 5.2a Results of time marching method for subsonic flow with the three different mesh arrangements	102
Figure 5.2b Results of time marching method for transonic flow with the three different mesh arrangements	102

Figure 5.3a Results of Newton method for subsonic flow with the three different mesh arrangements.....	103
Figure 5.3b Results of Newton method for transonic flow with the three different mesh arrangements.....	103
Figure 5.4 Variation of wall Mach number for subsonic flow over the bump	105
Figure 5.5a Isomach lines, time marching method	106
Figure 5.5b Isomach lines, Newton method	106
Figure 5.5c Isomach lines, computational result	106
Figure 5.6 Variation of wall Mach number for transonic flow over the bump.....	108
Figure 5.7a Isomach lines, time marching solver	109
Figure 5.7b Isomach lines, Newton solver.....	109
Figure 5.7c Isomach lines, computational result	109
Figure 5.8 Converging-diverging nozzle, variation of the mid-line Mach number.	110
Figure 5.9a Isomach lines, time marching solver	111
Figure 5.9b Isomach lines, Newton solver.....	111
Figure 5.10 Variation of surface Mach number of Hobson turbine blade	112
Figure 5.11a Isomach lines, time marching solver	113
Figure 5.11b Isomach lines, Newton solver.....	113
Figure 5.12 Variation of surface Mach number of NACA 0012 profile	115
Figure 5.13a Isomach lines, Time marching solver	116
Figure 5.13b Isomach lines, Newton solver.....	116
Figure 5.13c Isomach lines, computational result	116
Figure 5.14 Variation of surface static pressure of Sanz subcritical turbine blade..	118
Figure 5.15a Static pressure contours, time marching solver	119
Figure 5.15b Static pressure contours, Newton solver	119
Figure 5.16 Variation of surface Mach number of Sanz supercritical turbine blade	120

Figure 5.17a Isopressure lines, time marching solver	121
Figure 5.17b Isopressure lines, Newton solver	121
Figure 5.18 Variation of surface pressure of Sanz subcritical compressor blade	123
Figure 5.19a Isopressure lines, time marching solver	124
Figure 5.19b Isopressure lines, Newton solver	124
Figure 5.20 Throughflow calculation stations	125
Figure 5.21 Rotor R030, comparison of the computational results with experimental results	126
Figure 5.22 Rotor R030, comparison of the computational results with results in references.....	127
Figure 5.23a Mach number contours, time marching solver	128
Figure 5.23b Mach number contours, Newton solver	128
Figure 5.23c Mach number contours, experimental result.....	128

LIST OF TABLES

Table 2.1 Name of the Parameters	15
Table 2.2 Definitions of the input and outputs of the blade to blade solver	19
Table 4.1 Inflow and Outflow Characteristic Boundary Conditions	92
Table 4.2 Multistage Schemes Constants	103
Table 5.1 Error between grid levels	104

LIST OF SYMBOLS

Chapter 2

$\vec{r}$	Radius vector
$\vec{q}$	Relative velocity
Ω	Rotational speed
x	Machine axis
$\vec{e}$	Unit vector
β	Flow angle
$\emptyset$	Streamline Curvature Angle
i	Incidence
δ	Deviance
α	Angle of Attack
c	Chord Length
t/c	Pitch to Chord Ratio
λ	Stagger Angle
q	Velocity
β	Relative Flow Angle
P	Static Pressure at Inlet and Outlet
ζ	Chamber Angle
b	Normal area
m, r, φ	Meridional, radial and tangential axis

Subscripts and Superscripts

0	Stagnation Properties
m, r, φ	Meridional, radial and tangential components

Chapter 3

ρ	Density
$\vec{q}$	Relative velocity
p	Pressure
$\vec{f}_b$	Body Force
$\hat{n}$	Unit normal
ds	Infinitesimal surface area
dV	Infinitesimal control volume
I, h	Rotaphy, Enthalpy
γ	Specific heat ratio
$\vec{\Omega}$	Rotational speed
β	Flow angle
Φ	Streamline curvature angle
r	Radius
e	Unit vector
x, y	Cartesian coordinates
$\vec{A}$	Normal cell face vector
$\vec{B}$	Streamline cell face vector
$\hat{s}$	Local unit streamwise vector
$\vec{S}$	Streamwise average vector
$\vec{N}$	Normal average vector
π	Pressure on the streamline faces
A_n	Cross sectional streamtube area
ρ_{av}	Average density at the cell center
A_c	Cell volume per unit depth
q	Average cell velocity
p_c	Pressure correction for compatibility equation
p	Average cell pressure
k	Correction factor
f	Geometric parameters for streamwise momentum

g	Geometric parameters for normal momentum
R_s	Source term for streamwise momentum
R_n	Source term for normal momentum
$\tilde{\rho}$	Upwinded density
μ	Artificial compressibility coefficient
μ_{con}	Artificial compressibility factor
μ_c	Corrected artificial compressibility factor
$\dot{m}$	Mass flow rate
$\dot{m}_{total}$	Total mass flow rate
II	Number of x division
JJ	Number of y division
Δ, δ	Linearized quantity or change of quantity
S_{in}	Inlet slope
S_{out}	Outlet slope
$A_i, B_i, C_i,$ $Z_i, G_i, U_i,$ $R_i,$	Jacobian entries for each streamwise calculation station
RF	Relaxation factor

Subscripts and Superscripts

$r, x, m \varphi$	Radial, axial, meridional and tangential direction
$1,2,3$	Calculation stations in x direction for a typical stencil (Figures 3.3 and 3.4)
$1,2$	Inlet and outlet faces of computational cell (Figure 3.5)
$- / +$	Lower / upper face of the streamline
x,y	Component in the x and y direction
t	Total (stagnation) quantity
ILE, ITE	Leading edge index, trailing edge index
n	Iteration level

Chapter 4

ρ	Density
p	Pressure
t	Time
ds	Infinitesimal surface area
$d\Omega$	Infinitesimal control volume
Q	Source term vector
$\vec{V}$	Velocity vector
W	Rotational speed
f	Body forces
E	Total energy
$\vec{W}$	Conservative variable vector
$\vec{F}_c$	Flux vector
I	Rotalphy
u	x component of the velocity
v	Y component of the velocity
q_h	Heat addition to the system
$\vec{U}$	Primitive variable vector
γ	Specific heat ratio
$\vec{\omega}$	Rotational velocity
ω	Rotational speed
r	Radius
Ω_{ij}	Volume of the cell
$\vec{n}$	Normal of the face
x,y	Cartesian coordinates
$\vec{S}$	Surface area vector
$ \vec{A}_{Roe} $	Roe Matrix
δ	Second order correction for flux calculation
a, b	Difference operators for right and left states
ϵ	Van Albada limiter factor

J	Jacobian
R	Density in computational space
V_ζ, V_η	Velocity components in computational space
u, v	Velocity components in physical plane
C	Riemann invariants
α	Runge Kutta time integration constant
CFL	Courant–Friedrichs–Lewy condition

Subscripts and Superscripts

$\sim$	Roe averages
R, L	Right and left state
x, y	Components of a quantity in x and y direction
i, j	Index for node number
w	Wall
ζ, η	Coordinates in computational space
∞	Free stream
b	Value at the boundary
d	Value coming from interior of the domain

Chapter 5

C_p	Pressure Coefficient
M_∞	Free stream Mach Number
P_∞	Free stream total pressure

CHAPTER 1

INTRODUCTION

A gas turbine is a type of internal combustion engine which is composed of five main parts. These parts are intake, compressor, combustion chamber, turbine and exhaust. Intake collects the air and creates a uniform flow for compression. After compression through the compressor, air enters the combustion chamber where air and fuel are mixed and the fuel-air mixture are burned to create high pressure, high temperature flow for work extraction through turbine. After the air passes through the turbine, it is exhausted to the atmosphere.

Compressor and turbine are the two major components of a typical gas turbine because work addition and extraction occurs within these components. Since the compressor adjusts the pressure ratio and mass flow, it has great influence on the characteristics of a gas turbine. The turbine extracts work from high pressure and high temperature air flow. The design process of the compressor and turbine starts with the cycle analysis. The output of the cycle analysis is used in the mean-line design in order to determine the initial sizing of the gas path and number of stages. The output of the mean-line analysis feeds the detailed airfoil design procedure.

The design of blade is highly complicated process because of the presence of extremely complex flows. These complex flow structures include three dimensional boundary layers, mixture of the different type of flows, hub and tip boundary layers, wakes, secondary flow vortices and unsteadiness which are shown in Figures 1.1 and 1.2. Therefore, fast but accurate blade design tools which could handle all of these flow types are needed. In the gas turbine industry, these design methodologies highly depend on two or/and three dimensional CFD analysis.

The application of the first numerical methods in the turbomachinery was seen in 1940s. The very first solution of turbomachinery flow was the solution of the three dimensional potential equation [2]. However, the potential equation does not include rotationality, non-uniformity in enthalpy and entropy generation across the shock waves. Also, use of potential function result in a boundary value problem of the second kind which is made more difficult than the boundary value problem of the first kind. All of these difficulties were eliminated by the introduction of S_1 and S_2 streamsurfaces which are shown in Figures 1.3 and 1.4 [2]. The problem was formulated using the definition of the streamfunction on the streamsurfaces for both absolute and relative frames of reference. The equation of continuity and momentum for fluid flow are combined into one principal equation. The characteristic of the principal equation depends on local Mach number. In the subsonic region the equation is elliptic and in the supersonic region it is hyperbolic. Definition of two stream surfaces, which are blade to blade (S_1) and throughflow (S_2) streamsurfaces, was regarded as the complete three dimensional solution of the problem. That is, the actual three dimensional flow can be approximated considering the combination of the solution of the two stream surfaces (Quasi Three Dimensional Flow – Q3D). Since quasi three dimensional flow is the approximation of the actual three dimensional flow, it requires some assumptions. As a first and main assumption, flow field has periodic unsteadiness leading to steady flow analysis in both absolute and relative frame of reference. Also, flow through the S_2 plane is assumed to be axisymmetric and S_1 streamsurface is assumed as the surface of revolution [3]. The Q3D method does not require too much computational power and it is quite fast. However, this method was not the complete three dimensional solution of the problem; it was only the model of real 3-D flow. Since some assumptions are included, it is necessary to check the accuracy of the method with the experimental results. The flow chart which shows whole Q3D design process is presented in Figure 1.5.

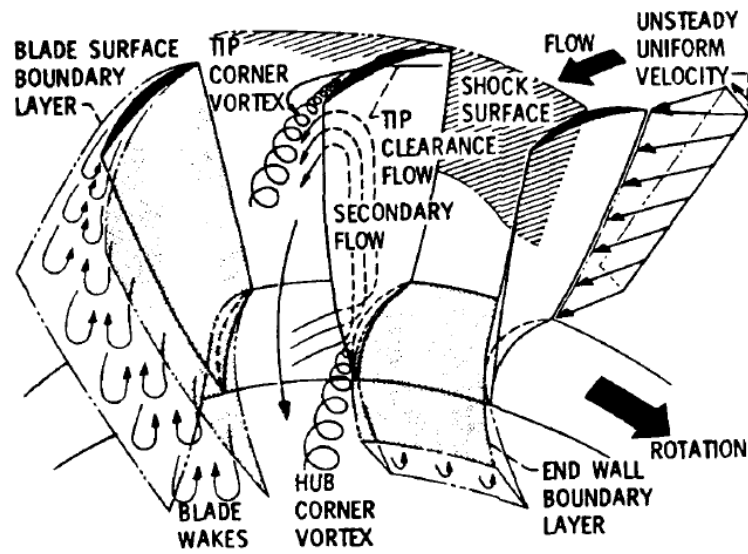


Figure 1.1 Compressor Blade Row Flow Field Characteristic [1]

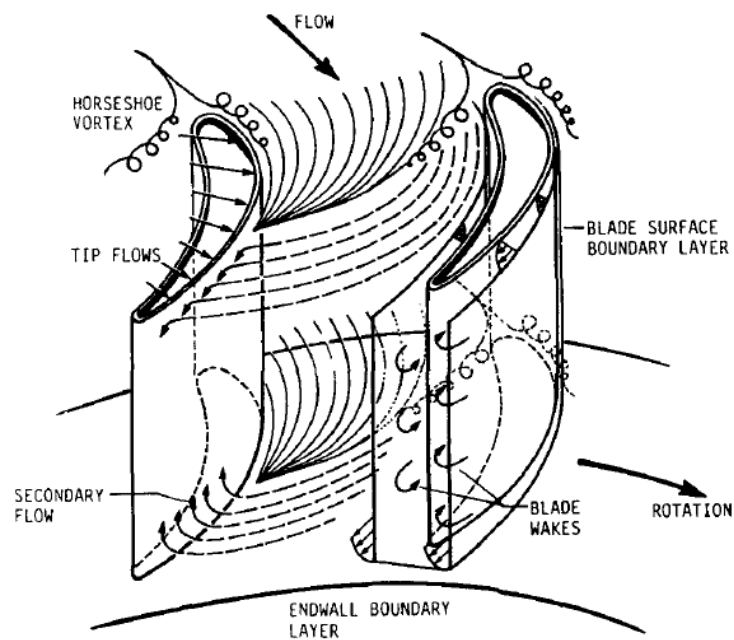


Figure 1.2 Turbine Blade Row Flow Field Characteristic [1]

The Q3D methods dominated the turbomachinery design until the early 1980's [4]. After 1980's, fully three dimensional methods first became available. These methods do not require assumptions which are needed in quasi 3-D method because it can solve complete blade rows in a single calculation. If a single blade row is to be solved, some problems arise in the implementation of the boundary conditions which are obtained from again throughflow calculations. However, today powerful computers give the possibility to solve whole stage in a single calculation which overcomes the problem of boundary condition implementation.

Modeling is only needed if the steady calculation is to be carried out. Since the real interaction of the stator and rotor is unsteady, some approximate modeling need to be made for steady calculations. However, this is not a disadvantage of the method since the error even with the modeling is still small. Unsteady solution of the complete blade row directly removes all the modelling and assumptions. The requirements of the powerful computers and computational time required to reach steady state solution can be seen as disadvantages of this method.

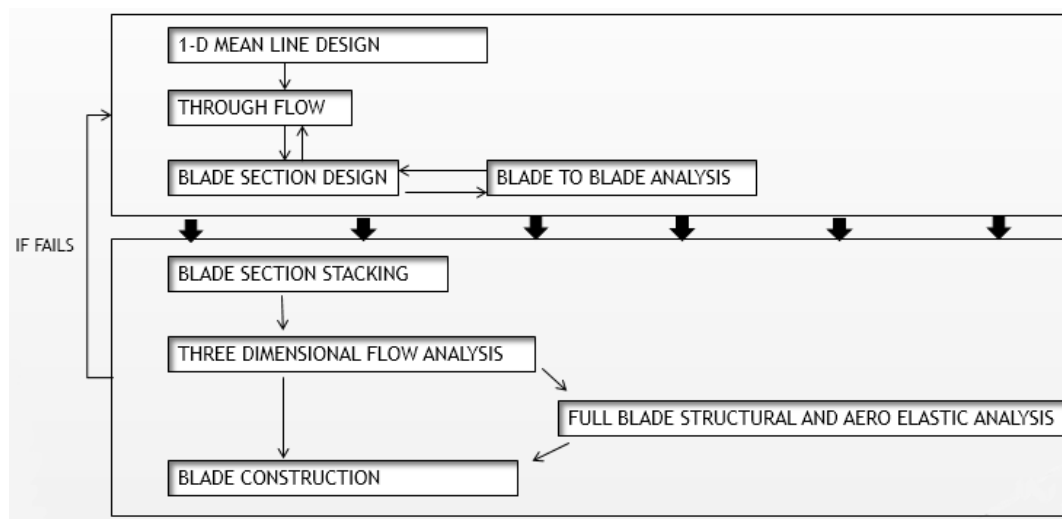


Figure 1.3 Quasi 3-D Design Flow Chart

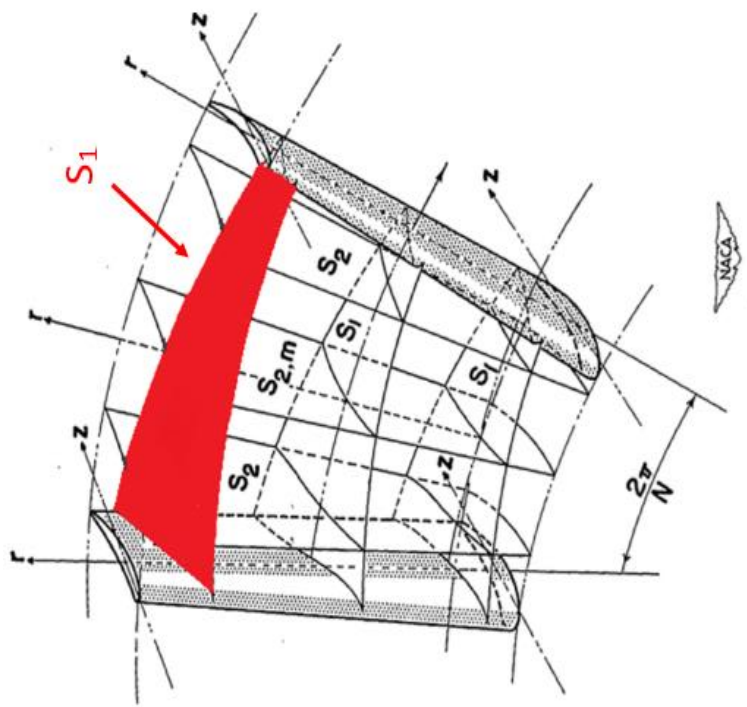


Figure 1.4 Blade to Blade Stream Surface [2]

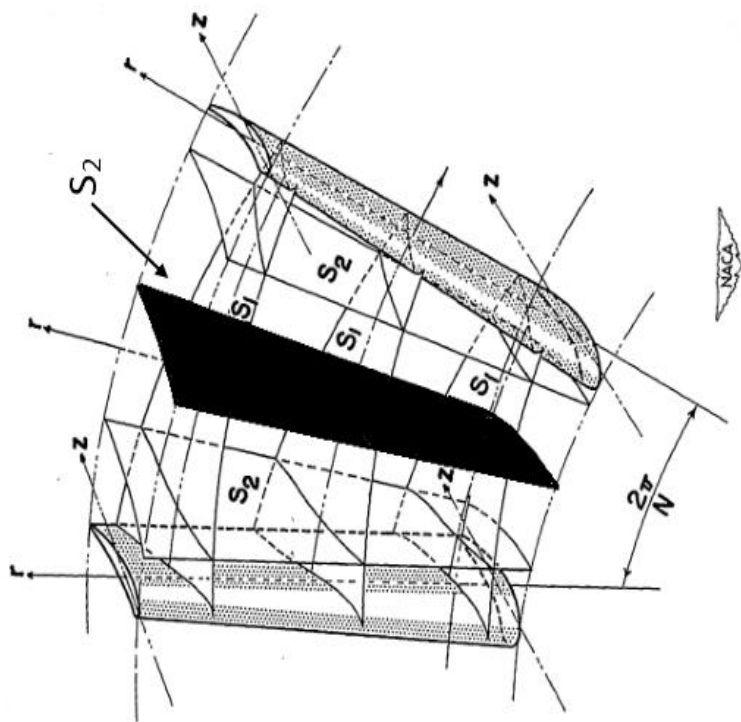


Figure 1.5. Through Flow Stream Surface [2]

1.1. Review of Literature

In the literature, various methods are available for the solution of blade to blade flow. The method of Wu [2] is a good approximate solution but it was insufficient for transonic flow calculations. Transonic principal equation is of elliptical type in subsonic region and hyperbolic type in supersonic region. In other words, the solution of the transonic flow equation includes the solutions of the both elliptic and hyperbolic equations. Since the principal equation has different characteristics for subsonic and supersonic flow, it can handle only subsonic or only supersonic flow. This actually is the general problem of the steady solvers. The characteristic of the steady flow equations depends on the speed regime. The characteristic of flow equations is elliptic in subsonic flow and hyperbolic in supersonic flow. Hyperbolic and elliptic equations are solved with different numerical methods. While relaxation methods are used in elliptic equation, space marching approach is used for hyperbolic steady flow equations. In other words, relaxation methods do not work in supersonic region and the space marching methods do not work in subsonic region. In order to overcome this difficulty, first transonic potential equation was introduced by Murman and Cole [5]. They introduced numerical viscosity in supersonic region. This method can capture the shock without any mass flux loss but momentum is not conserved due to nature of the potential flow (irrotational and isentropic). However, the integral form of the Euler and Navier-Stokes equations can describe rotational flow. It gives correct Rankine-Hugoniot relations on both sides of the shock wave. The conservative form guarantees conservation of mass, momentum and enthalpy fluxes on both sides of the shock. The approach developed by Murman and Cole leads the use of relaxation method in supersonic region. Newton's method (Newton-Raphson) [5] is the one way of interpreting the relaxation procedure. The nonlinear terms are linearized about the current solution and linearized system is solved by relaxation technique to obtain a better approximate solution. The Newton method is applied to the Euler and Navier-Stokes equations by Childs and Puaillim and Jespersen [7]. The Newton procedure worked but it does not have any advantages over the traditional time marching methods in terms of accuracy and speed. Van Leer and Mulder [8] developed an intermediate approach between time marching and Newton method. This method behaves like a time marching method for small time step while it behaves as a Newton

method for large time steps. This intermediate method has good computational efficiency but the shock locations cannot be captured accurately.

An early method used in the calculation of steady Euler equation is the streamline curvature method [9]. The one set of grid lines corresponds to streamlines of the flow so that grid points are free to move. The grid displacement is the additional unknown to the thermodynamic variables. Non-conservative form of the normal and tangential momentum equations are discretized in finite difference form and solved by the relaxation technique. The mass flux and enthalpy (or rothalpy) is conserved along the streamtubes. Although there are certain applications of streamline curvature method for supersonic flow calculations, instabilities arise at shock waves [4]. This method is popular for subsonic turbomachinery calculations and it is widely used in throughflow calculations.

Giles [6] has combined the streamline grids of the streamline curvature method and conservative formulation of finite volume method. Steady state integral form of the Euler equations are applied to the quadrilateral cells. The discretized Euler equations are linearized using Newton-Raphson linearization technique. The system of linearized equations are solved using relaxation technique. Originally, it is capable of solving the subsonic flow only. However, in order to cover transonic flow, artificial compressibility concept was implemented by Giles [6] similar to the application for the potential equation [5]. With the implementation of the artificial compressibility, supersonic flow is covered without loss of accuracy at the shock waves. In his work, Giles used a discretization technique which combines the conservative formulation of the finite volume method with the intrinsic streamline grid of streamline curvature method. Since the Euler equation is solved on actual streamline grids, the mass flux across streamlines is automatically eliminated. The only contribution coming from the streamline surfaces to the Euler equation is the pressure term. Along the streamline, with the aid of continuity and energy equations, mass flux and enthalpy (or rothalpy) are conserved. Also, grid node displacement is treated as primitive variable and solved together with the flow variables. It may be seen that this approach increases the number of unknowns leading to additional computational effort. However, this difficulty is eliminated using the Newton linearization. Giles [6] has applied Newton

linearization and manipulated the flow equation in order to obtain more simplified form of the original equations. Finally, all the flow variables are written in terms of variation of nodal displacement normal to the streamline and variation of density of the fluid. That is, the total unknown is reduced from five (δ_n , $\delta\rho$, δu , δv and δp) to two (δ_n and $\delta\rho$). The resultant system of equations can be solved either by classical Gauss elimination method or an iterative method.

The latest and most advanced solution method of Euler equations is the time marching method. Unsteady form of the Euler equation is of hyperbolic type in time so that flow variables can be marched in time. Since the characteristic of unsteady form of the Euler equation is of hyperbolic type for all flow regimes (subsonic, transonic, supersonic), transonic flow can be solved without additional treatment such as artificial compressibility. The schemes used in the time marching methods can be classified as central methods and characteristic based methods (upwind methods) in the literature. The Lax Wendroff method [13] which is second order accurate in time and space is one of the well-known central schemes in the literature. Another approach developed by the MacCormack [14] which is a different version of the Lax Wendroff method. This is also second order accurate in time and space. Ni [15] has developed a multiple grid algorithm in order to accelerate the solution procedure. Although this algorithm is based on central scheme, it shows characteristic behavior and can easily be implemented for finite volume integration. In general, the central schemes have instabilities at the discontinuities and they cannot suppress odd-even decoupling [12]. In order to solve this problem, artificial viscosity (dissipation) is implemented to the original central schemes. The artificial dissipation procedure includes the artificial smoothing factor which is determined by the trial error process [16], [17].

The characteristic based spatial discretization schemes are more advanced methods as compared to the central schemes because the physical behavior of the gas flow (wave propagation direction) is considered in the flux calculation. Roughly, the upwind methods can be divided into four group including flux vector splitting, flux difference splitting, total variation diminishing (TVD) and fluctuating splitting schemes [12]. The first flux vector splitting method is introduced by Steger and Warming [20]. They decomposed the vector of the convective fluxes into two parts according to sign of the

characteristic variables. The second group of the flux vector splitting decomposes the flux vectors into two parts: convective and pressure (acoustic) parts. The popular second group of methods are AUSM (Advection Upstream Splitting Method) [21] and CUSP (Convective Upwind Split Pressure) [22]. The first idea for flux difference splitting schemes was introduced by the Godunov [23] including locally one dimensional solutions of the Riemann problem. Later Roe [24] and Osher [25] has developed approximate Riemann solver in order to reduce the computational effort. The idea of TVD schemes was first introduced by Harten [26]. The idea for the TVD schemes is that the maxima must be non-increasing and minima must be non-decreasing. This idea is also called monotonicity preserving. The TVD schemes are generally include averaging of the convective fluxes of either side of the face with additional dissipation terms. These dissipation terms can either be characteristic based or symmetric type. Finally, Fluctuating Splitting Schemes split the equations according to the orientation of the grid cells [27]. For this method, the flow variables are associated at the grid nodes.

1.2. Present Study

In this thesis, two different approaches are used for the solution of the blade to blade flow. The first one is the steady solution of two dimensional Euler equations [6]. Integral form of the two dimensional Euler equations in a rotating frame of reference are solved by using cell centered finite volume method. The flow domain is discretized on the intrinsic streamline grid. Initially Euler equation is discretized as x and y momentum equations. After some manipulation, the streamwise and normal momentum equations are obtained from momentum equations in x and y directions. For transonic flow solution, the artificial compressibility is implemented and shocks are captured accurately. After the discretization is completed, the Newton-Raphson method is applied to the discrete flow equations. Together with the linearization, some manipulation is applied using continuity and energy equations in order to reduce the number of unknowns to two. After the linearization is applied to all discrete control volumes, a huge linear system of equations is formed. For the solution of linear system of equations, both direct and iterative approaches may be used. Giles [6] has stated

that the iterative solver has no superiority on the direct method, so in this thesis, direct method is preferred. Direct method is the solution of the resultant linear system of equations using modified tri-diagonal algorithm [10] which was developed for the block matrix systems. The density and nodal coordinates are updated for next iteration level using under relaxation. The relaxation factor is used in order to prevent excess changes in density and nodal coordinates.

The second approach for the solution of the Euler equation is the time marching finite volume method. The integral form of the unsteady Euler equations are discretized on a polyhedral control volume which does not change in time. The flow variables are stored at the centers of the control volumes (cell centered scheme). Fluxes are evaluated at the faces of the each cell considering left and right states of the faces. The left and right states of the faces are determined using MUSCL approach with modified Van Albada Limiter function [12], [43]. Two different flux difference splitting methods (Roe and RHLL) are used in order to determine the fluxes at the faces of the computational cells. The time integration is made explicitly applying Runge-Kutta time integration [12], [45]. In order to accelerate the convergence, local time stepping is applied.

In chapter two, some definitions about the blade to blade surface are presented. In chapter three, details of the newton solver are presented. In chapter four, details of the time marching solver are presented. Test cases, output of both solvers are presented in chapter five. The discussion and conclusion are presented in chapter 6.

CHAPTER 2

DEFINITION OF THE BLADE TO BLADE STREAM SURFACE

In this chapter, details of the blade to blade flow domain are presented. Previously, S_1 and S_2 streamsurfaces are presented in Figure 1.4 and Figure 1.5. For further visualization, Figure 2.1 shows the three dimensional view of the blade to blade streamsurface.

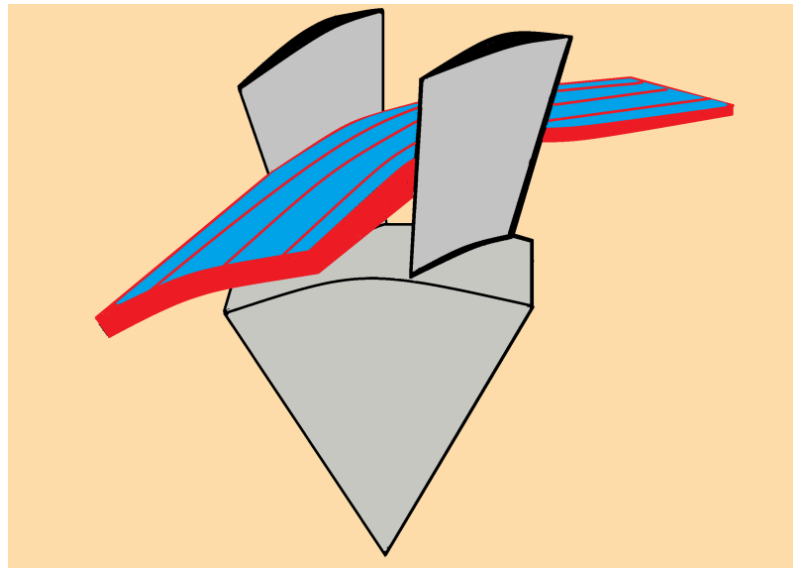


Figure 2.1 Blade to blade stream surface [29]

This streamsurface is solved by making some simplifications. First, the cylindrical surface of the blade to blade domain is approximated as flat surface which is shown in Figure 2.2. This is required for the solution of the flow equation in the two dimensional space. In the Figure 2.2, while PQ is the actual surface, the P'Q' shows the approximated surface. The only requirement for this approach is the high solidity. For low solidity case, the deviation of the approximate streamsurface from actual streamsurface increases and, hence, the accuracy of the results decreases.

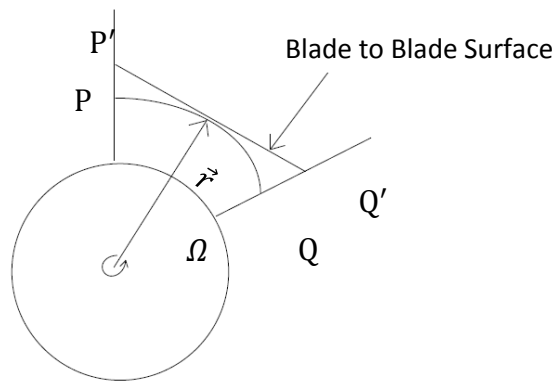


Figure 2.2 Real and approximate blade to blade surface [30]

The relation between the throughflow and blade to blade streamsurfaces is shown in Figure 2.3 [31]. In quasi three-dimensional method, the thorough flow and blade to blade calculation procedures are coupled. The streamsurface definition used in blade to blade calculations comes from the throughflow calculations. The streamline curvature angles and inlet-outlet radius of the streamtube are obtained from the definition of the streamlines. These streamlines are determined from the throughflow calculations.

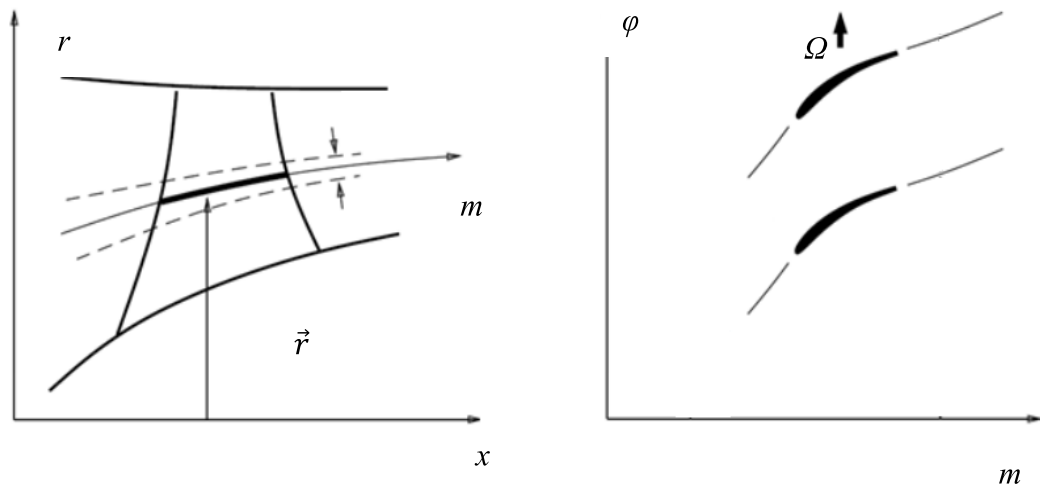


Figure 2.3 Relation between S_1 and S_2 streamsurfaces [31]

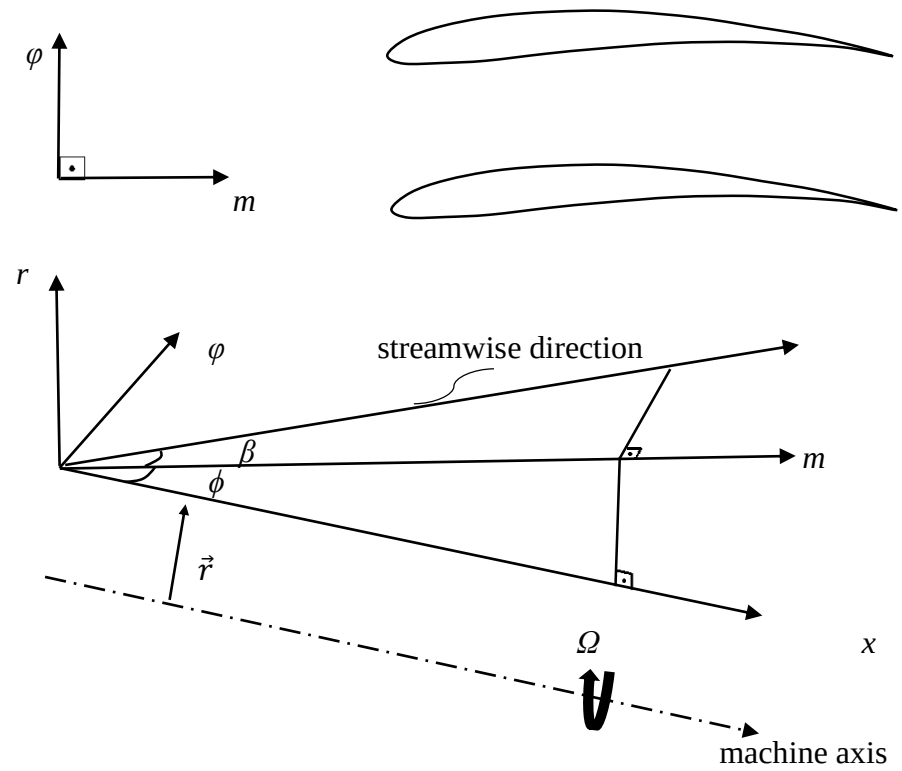


Figure 2.4 Relation between the tangential, axial and meridional directions [32].

Also the relation between the axial direction (x), tangential direction (ϕ) and radial direction is shown in the Figure 2.4.

According to Figure 2.4, the relative velocity, position and angular velocity vectors can be written as

$$\vec{q} = q \cos(\beta) \cos(\phi) \vec{e}_x + q \cos(\beta) \sin(\phi) \vec{e}_r + q \sin(\beta) \vec{e}_\phi \quad (2.1)$$

$$\vec{\Omega} = \Omega \vec{e}_x \quad (2.2)$$

$$\vec{r} = r \vec{e}_r \quad (2.3)$$

where q , Ω and r are the magnitudes of the velocity, angular velocity and position vectors [30], [32]. $\vec{e}_x$, $\vec{e}_r$ and $\vec{e}_\phi$ are the unit vectors along x , r and ϕ directions as shown in Figure 2.4. The angle β is the flow angle defined on the m - ϕ plane and the angle ϕ is the streamline curvature angle defined on the r - x plane.

The solution of the governing equations is carried out on the $m - \phi$ streamsurface, therefore all the variables must be defined on this surface. The unit vector $\vec{e}_m$ can be written as [30]

$$\vec{e}_m = \cos(\phi) \vec{e}_x + \sin(\phi) \vec{e}_r \quad (2.4)$$

This equation is especially required for the treatment of the source terms which will be explained in Chapter 3.

Some other parameters which are shown in Figure 2.5 are defined for the cascade domain.

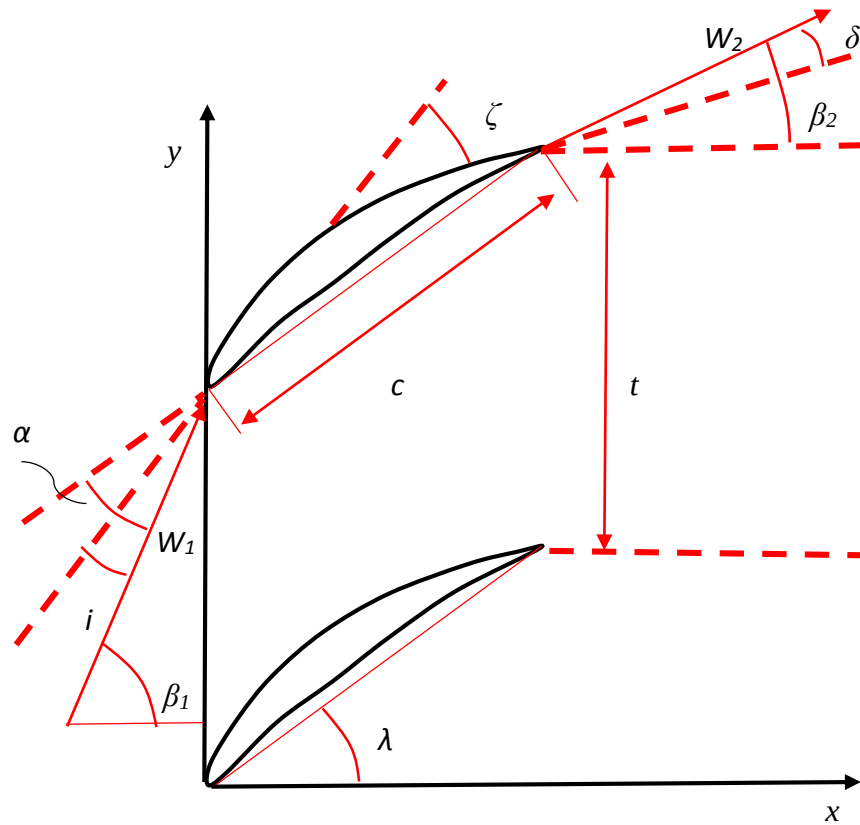


Figure 2.5 Definition of the Geometric Parameters [29]

Table 2.1 Definition of the geometric parameters in figure 2.5 [29]

α	Angle of Attack
c	Chord Length
t/c	Pitch to Chord Ratio
λ	Stagger Angle
W_1, W_2	Inlet and Outlet Velocity
β_1, β_2	Relative Flow Angle at the Inlet
i	Incidence
δ	Deviation
P_1, P_2	Static Pressure at Inlet and Outlet
P_1^0, P_2^0	Stagnation Pressure at Inlet and Outlet
ζ	Chamber Angle

2.1. Characteristics of flow in Blade to Blade Surface

The flow characteristic for blade to blade flow can be grouped into two for accelerating and decelerating flow. Typical accelerating and decelerating flow domains are shown in figure 2.6.

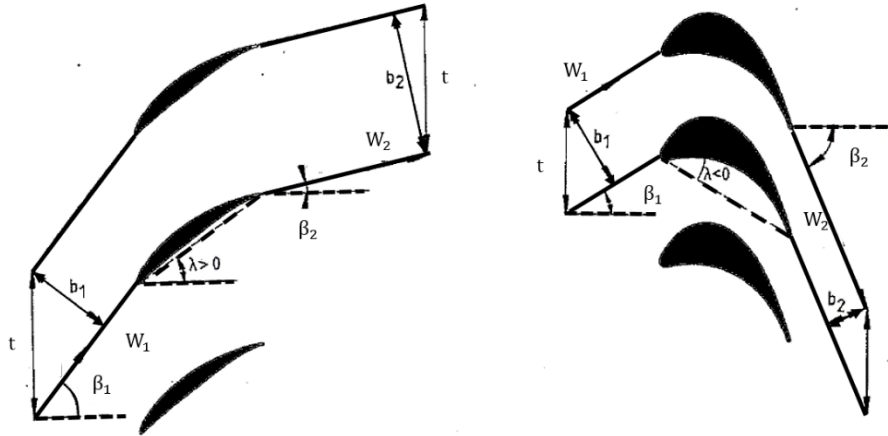


Figure 2.6 Decelerating and accelerating flow domains [29]

In order to understand the acceleration and deceleration in the flow field, the continuity equation can be written for the domains which are shown in Figure 2.6 as

$$\rho_1 W_1 b_1 = \rho_2 W_2 b_2 \quad (2.5)$$

In gas turbines, the flow is absolutely compressible but in order to understand the relation between the magnitude of the velocity and angles, following relation can be written neglecting the density change.

$$W_1 t \cos \beta_1 = W_2 t \cos \beta_2 \quad (2.6)$$

or

$$\frac{W_1}{W_2} = \frac{\cos \beta_2}{\cos \beta_1} \quad (2.7)$$

A decrease of the absolute value of the β results in a decelerating flow so that the blade passage compresses the fluid (compressor). An absolute increase of the value of the β results in accelerating flow indicating that the blade passage expands the fluid (turbine).

2.2. Inputs and Outputs of the Blade to Blade Solver

In quasi 3-D design method, the blade to blade solver and throughflow solver is coupled in order to obtain 3-D blade shape. The results of the throughflow solver are the inputs for the blade to blade solver. The flow on the S_2 plane is solved along a number of streamlines which are specified by the designer. The Figure 2.7 shows the solution stations and streamlines of the throughflow domain.

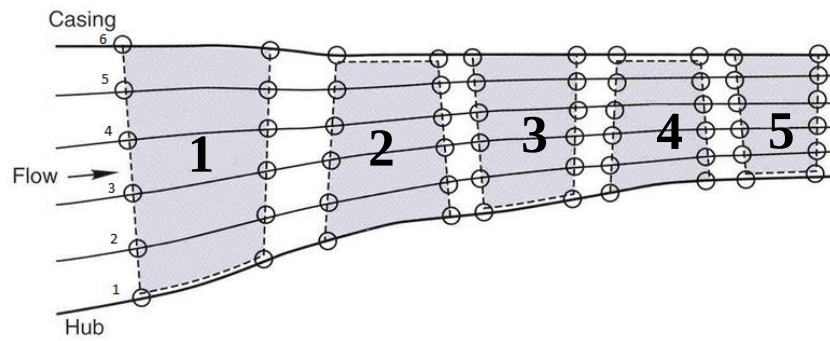


Figure 2.7 Throughflow computation stations

In Figure 2.7, there are 6 calculation stations for blade to blade solver. For example, when the first blade is solved, six different blade profiles will be obtained. After these

profiles are determined, the stacking process starts and complete 3-D blade profile is obtained.

Above explanations briefly indicate the quasi 3-D design process and the role of the blade to blade solver. The detailed input output relations are shown in Figure 2.8 and Table 2.2.

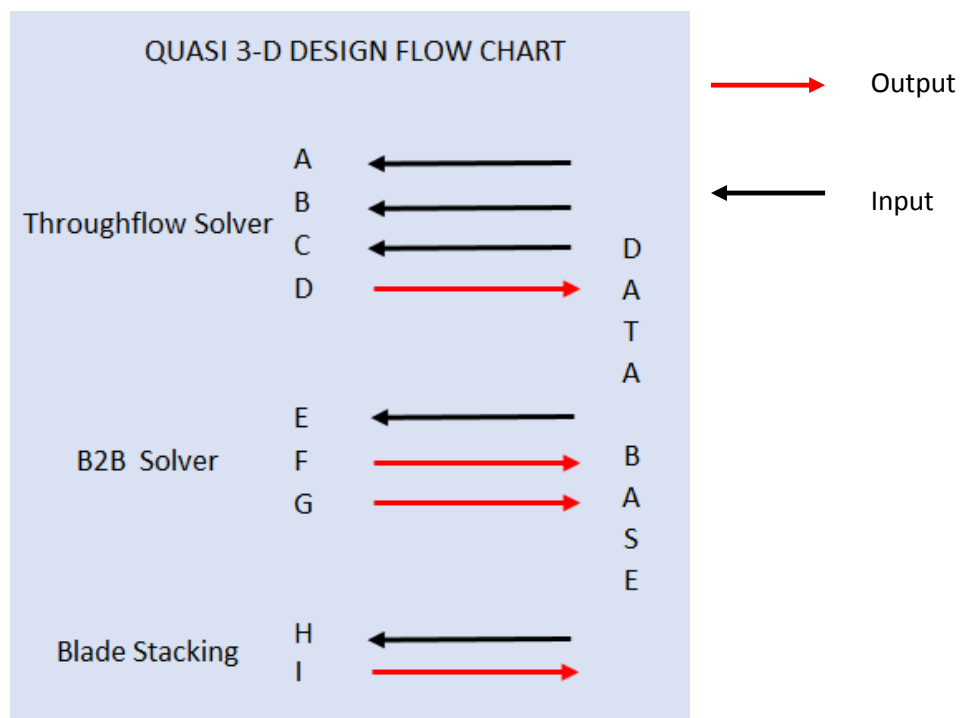


Figure 2.8 Input and Output of the solvers [1]

Table 2.2 Definitions of the input and outputs in Figure 2.8 [1]

A	Annulus information
	Blade row exit information
	Inlet profiles of pressure, temperature and velocity
	Inlet mass flow and rotational speed of the rotor
B	Blade geometry and loss distribution
C	Passage averaged perturbation time
D	Blade row inlet and exit conditions
	Streamline definition and streamtube height
E	Blade geometry, Rotational speed
	Inlet and exit boundary conditions
	Streamline definition and streamtube height
F	Surface velocity distribution
	Profile and overall loss
G	Passage averaged perturbation terms
	Loss distribution
H	Blade section geometry
	Stacking points
I	Three dimensional blade geometry

From Figure 2.9 and Table 2.2, it is clearly seen that all the inputs for the blade to blade solver come from the throughflow solver. The blade to blade solver should give surface velocity distribution or surface pressure distribution and overall loss including stagnation pressure loss and viscous losses. Since the solvers developed in this thesis are inviscid, specification of the viscous losses cannot be determined. Implementation of the viscous behavior into the current computer programs is the future work of this study. At this point, only the stagnation pressure loss due to shocks and velocity distribution on the blade surface will be presented as output of the blade to blade solver.

CHAPTER 3

LINEARIZED EULER SOLVER

The method includes the solution of the integral form of Euler equations with the cell centered finite volume approach. The densities, relative velocities and pressures are stored at the cell faces. The momentum equation is written in x and y directions, then it is converted streamwise and normal momentum equations applying some manipulation. Since the grid actually represents the streamline of the flow, the only contribution coming from the streamline surfaces are the pressure values (no mass flow exists across the streamline). Also with the linearization process the number of unknowns are reduced to two which are density change ($\delta\rho$) and nodal displacement change (δn). The linear system of equations is solved using modified tri-diagonal algorithm. At each iteration, x and y coordinates of nodes are updated together with the density values at cell faces.

3.1. Geometric Parameters and Equation Discretization

Figure 3.1 shows typical structured mesh applied to NACA 0012 cascade. Each horizontal line shown in Figure 3.1 represents the streamlines of the flow. For each control volume shown in the Figure 3.1, the integral form of steady state Euler equations can be written as

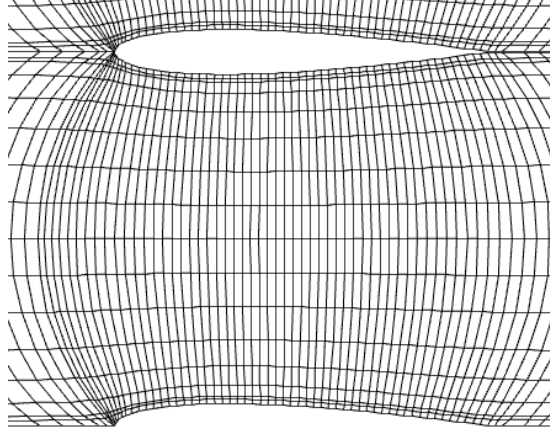


Figure 3.1 Structured mesh on a NACA 0012 cascade flow domain

$$\oint \rho \vec{q} \cdot \hat{n} ds = 0 \quad (3.1)$$

$$\oint (\rho(\vec{q} \cdot \hat{n}) \vec{q} + p \hat{n}) ds = \int \rho \vec{f}_b dV \quad (3.2)$$

$$\oint \rho \vec{q} \cdot \hat{n} l ds = 0 \quad (3.3)$$

In rotating frame of reference, instead of total enthalpy, rothalpy is conserved and defined as

$$I = \frac{\gamma}{\gamma - 1} \left(\frac{p}{\rho} \right) + \frac{q^2}{2} - \frac{(\Omega r)^2}{2} \quad (3.4)$$

When the rotational speed is set to zero, the rotating frame of reference automatically turns into the stationary frame of reference. The relative velocity “ q ” becomes the absolute velocity and rothalpy turns out to be total enthalpy.

The body forces due to rotation are the Coriolis and centrifugal body forces. These forces can be written in general form as

$$\vec{f}_b = -\vec{\Omega} \times (\vec{\Omega} \times \vec{r}) - 2\vec{\Omega} \times \vec{q} \quad (3.5)$$

In Equation (3.5), the first term and second terms represent the centrifugal and Coriolis forces respectively. Referring to Figure 2.4 in Chapter 2, the body forces can be written as

$$\vec{f}_b = 2\Omega q \cos\beta \sin\phi \vec{e}_\varphi - (2\Omega q \sin\beta - \Omega^2 r) \vec{e}_r \quad (3.6)$$

In order to calculate the effect of these body forces on the meridional-tangential (m- φ) plane, the dot product of Equation (3.6) with $\vec{e}_m$ and $\vec{e}_\varphi$ are required. Normally, $\vec{e}_m$ does not exist but it can be easily derived from axial and radial unit vectors.

From Figure 3.2, the meridional unit vector can be written as

$$\vec{e}_m = \cos\phi \vec{e}_x + \sin\phi \vec{e}_r \quad (3.7)$$

In the meridional plane, the body forces per unit mass can now be defined as

$$\vec{f}_b \cdot \vec{e}_m = -2\Omega q_{av} \sin\beta \sin\phi + \Omega^2 r \sin\phi \quad (3.8)$$

$$\vec{f}_b \cdot \vec{e}_\varphi = 2\Omega q_{av} \cos\beta \sin\phi \quad (3.9)$$

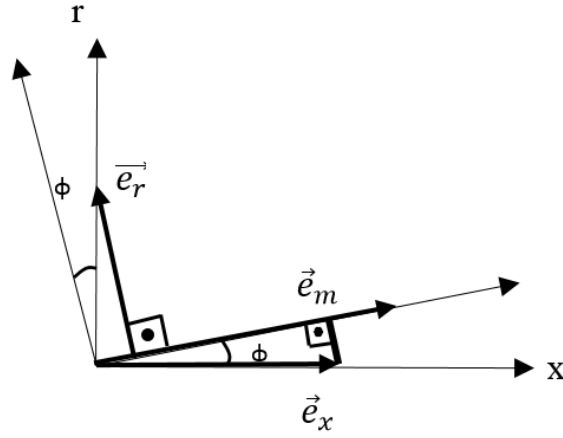


Figure 3.2 Unit vectors on r-x plane [32]

Now all the variables in Equations (3.1), (3.2) and (3.3) are clearly written in m - φ plane so that the discrete form of the governing equations can be defined fully. Before the discretization of the governing equation, geometric parameters for the computational cell will be defined.

For geometric parameters, the notations which are used in references [3], [6], [30], [32], [34] and [35] are also used in this thesis. The typical computational cell along the streamtube is shown in Figure 3.3. The crosses and circles represent position of the node and cell face midpoints, respectively. This convention is different from the classical structured, cell centered finite volume method because the cell faces normally combines two adjacent nodes but in this method cell faces are located between the midway of the nodes.

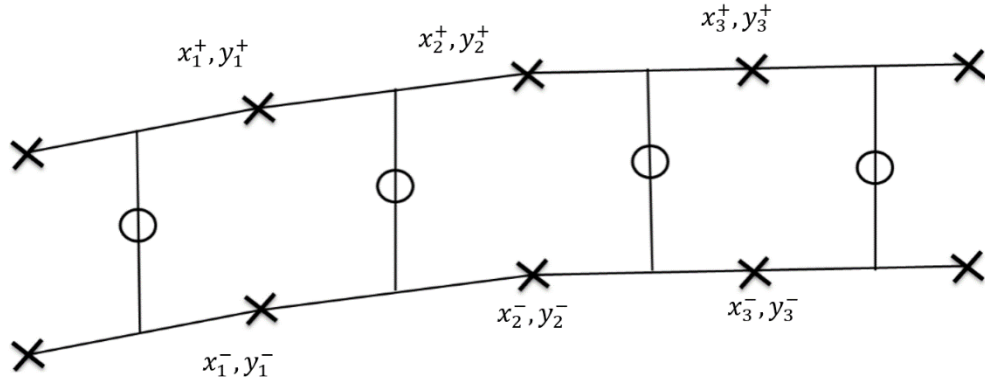


Figure 3.3 Typical computational cell, location of the grid nodes and face centers [3]

Four vectors are need to be defined along the cell and streamline faces, which are illustrated in Figure 3.4.

$$A_{x1} = \frac{1}{2}(x_1^+ + x_2^+) - \frac{1}{2}(x_1^- + x_2^-) \quad (3.10a)$$

$$A_{x2} = \frac{1}{2}(x_2^+ + x_3^+) - \frac{1}{2}(x_2^- + x_3^-) \quad (3.10b)$$

$$A_{y1} = \frac{1}{2}(y_1^+ + y_2^+) - \frac{1}{2}(y_1^- + y_2^-) \quad (3.11a)$$

$$A_{y2} = \frac{1}{2}(y_2^+ + y_3^+) - \frac{1}{2}(y_2^- + y_3^-) \quad (3.11b)$$

$$B_x^- = \left[\frac{1}{2}(x_3^- + x_2^-) - x_2^- \right] + \left[x_2^- - \frac{1}{2}(x_2^- + x_1^-) \right] = \frac{1}{2}(x_3^- - x_1^-) \quad (3.12)$$

$$B_y^- = \left[\frac{1}{2}(y_3^- + y_2^-) - y_2^- \right] + \left[y_2^- - \frac{1}{2}(y_2^- + y_1^-) \right] = \frac{1}{2}(y_3^- - y_1^-) \quad (3.13)$$

$$B_x^+ = \left[\frac{1}{2}(x_3^+ + x_2^+) - x_2^+ \right] + \left[x_2^+ - \frac{1}{2}(x_2^+ + x_1^+) \right] = \frac{1}{2}(x_3^+ - x_1^+) \quad (3.14)$$

$$B_y^+ = \left[\frac{1}{2}(y_3^+ + y_2^+) - y_2^+ \right] + \left[y_2^+ - \frac{1}{2}(y_2^+ + y_1^+) \right] = \frac{1}{2}(y_3^+ - y_1^+) \quad (3.15)$$

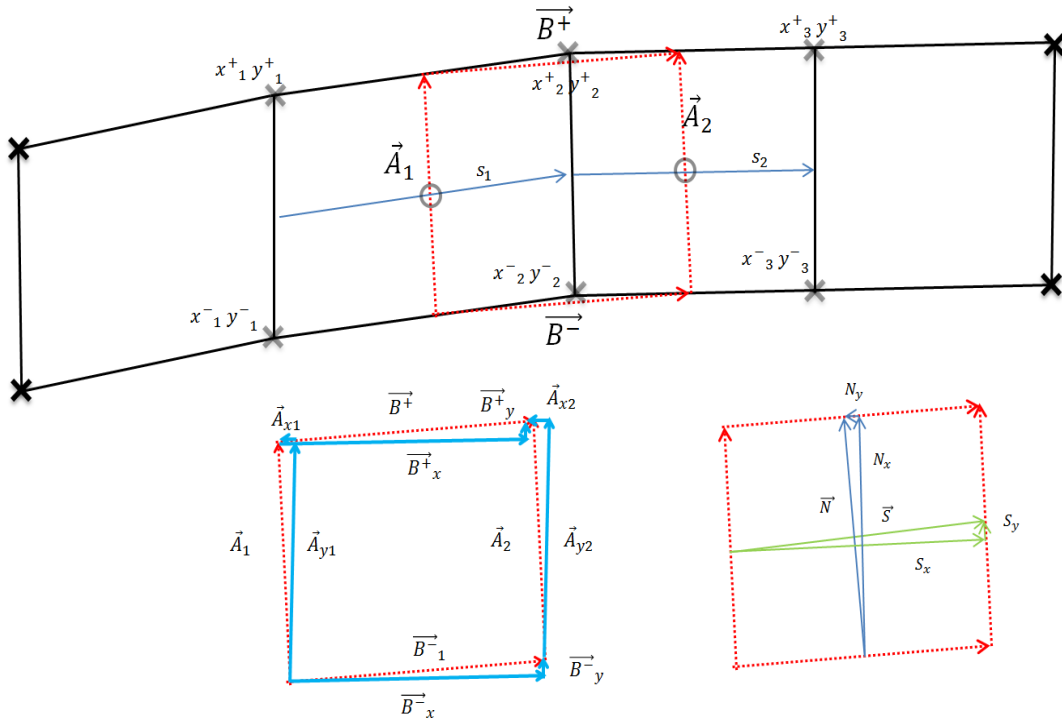


Figure 3.4 Definition of vectors $\vec{A}_1, \vec{A}_2, \vec{B}^-, \vec{B}^+$ [6]

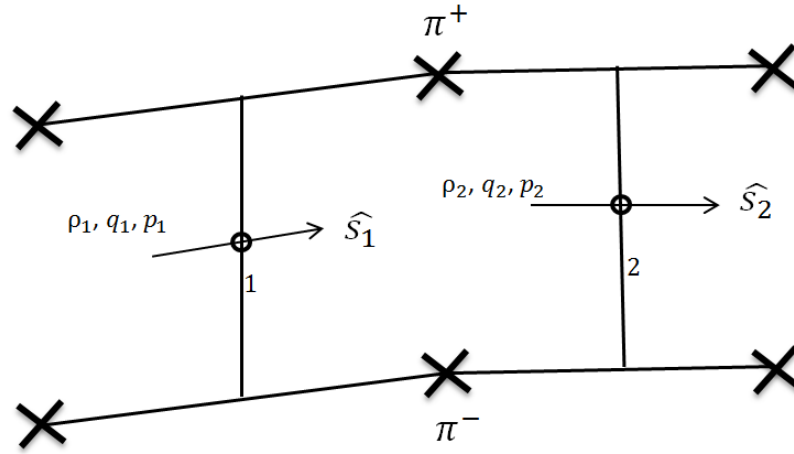


Figure 3.5 Location of flow variables or midpoint of the cell faces [28]

Additional vector quantities which are the averages of $\vec{B}$ and $\vec{A}$ are defined as

$$S_x = \frac{1}{2}(B_x^- + B_x^+) \quad (3.16a)$$

$$S_y = \frac{1}{2}(B_y^- + B_y^+) \quad (3.16b)$$

$$N_x = \frac{1}{2}(A_{x1} + A_{x2}) \quad (3.17a)$$

$$N_y = \frac{1}{2}(A_{y1} + A_{y2}) \quad (3.17b)$$

Since the horizontal lines represent the edges of the streamtube, the unit vector which indicates the flow direction at the face of the cells should be defined.

This unit vectors are defined as lowercase s which are shown below as

$$s_{x1} = \frac{1}{2}(x_2^- + x_2^+) - \frac{1}{2}(x_1^- + x_1^+) \quad (3.18a)$$

$$s_{y1} = \frac{1}{2}(y_2^- + y_2^+) - \frac{1}{2}(y_1^- + y_1^+) \quad (3.18b)$$

$$s_1 = (s_{x1}^2 + s_{y1}^2)^{\frac{1}{2}} \quad (3.19a)$$

$$\hat{s}_{x1} = \frac{s_{x1}}{s_1} \quad (3.19b)$$

$$\hat{s}_{y1} = \frac{s_{y1}}{s_1} \quad (3.19c)$$

$$s_{x2} = \frac{1}{2}(x_3^- + x_3^+) - \frac{1}{2}(x_2^- + x_2^+) \quad (3.20a)$$

$$s_{y2} = \frac{1}{2}(y_3^- + y_3^+) - \frac{1}{2}(y_2^- + y_2^+) \quad (3.20b)$$

$$s_2 = (s_{x2}^2 + s_{y2}^2)^{\frac{1}{2}} \quad (3.21a)$$

$$\hat{s}_{x2} = \frac{s_{x2}}{s_2} \quad (3.21b)$$

$$\hat{s}_{y2} = \frac{s_{y2}}{s_2} \quad (3.21c)$$

In two dimensional analysis, the normal flow area (A_n) becomes length of the face. The normal area definitions are

$$A_{n1} = s_{x1}A_{y1} - s_{y1}A_{x1} \equiv |\hat{s}_1 \times \vec{A}_1| \quad (3.22a)$$

$$A_{n2} = s_{x2}A_{y2} - s_{y2}A_{x2} \equiv |\hat{s}_2 \times \vec{A}_2| \quad (3.22b)$$

The schematic representation of the normal area is shown in Figure 3.6.

The operator “|...|” is the only magnitude of the cross product in two dimensions (in three dimensions, it is a vector). Also this operator does not indicate the absolute value of a quantity. The result both can be negative or positive depending on the flow conditions. However, the normal area cannot be negative, in a normal operation this always results in a positive value.

The flow variables density, ρ , speed, q , and pressure, p are located at the midpoint of computational faces, as shown in Figure 3.5. Since no mass flux passes across the streamlines, pressure is the only contribution to the Euler equations from the streamlines. In order not to confuse with the pressure values stored at the midpoint of the faces, the symbol “ π ” is used for the pressure on the streamline faces.

Continuity equation is the simple expression indicating the conservation of mass flux within the streamtube. Referring to Figure 3.5, the mass flux entering to the inlet face (1), is equal to the mass flux leaving out through the outlet face (2). The continuity equation is expressed as

$$\dot{m} = \rho_1 q_1 A_{n1} = \rho_2 q_2 A_{n2} \quad (3.23)$$

From now on, for convenience m - ϕ plane is treated as x - y plane. Then, the integral form of the momentum equation is discretized referring to Figures 3.3, 3.4, 3.5 and 3.6 as follow

$$\begin{aligned} & \rho_1 q_1^2 A_{n1} \hat{s}_{x1} - \rho_2 q_2^2 A_{n2} \hat{s}_{x2} + p_1 A_{y1} - p_2 A_{y2} + \pi^+ B_y^+ - \pi^- B_y^- \\ & = \rho_{av} A_c (-2\Omega q_{av} \sin\beta \sin\phi + \Omega^2 r \sin\phi) \end{aligned} \quad (3.24)$$

and

$$\begin{aligned} \rho_1 q_1^2 A_{n1} \hat{s}_{y1} - \rho_2 q_2^2 A_{n2} \hat{s}_{y2} - p_1 A_{x1} + p_2 A_{x2} - \pi^+ B_x^+ + \pi^- B_x^- \\ = \rho_{av} A_c (2\Omega q_{av} \cos\beta \sin\phi) \end{aligned} \quad (3.25)$$

in x and y directions, respectively.

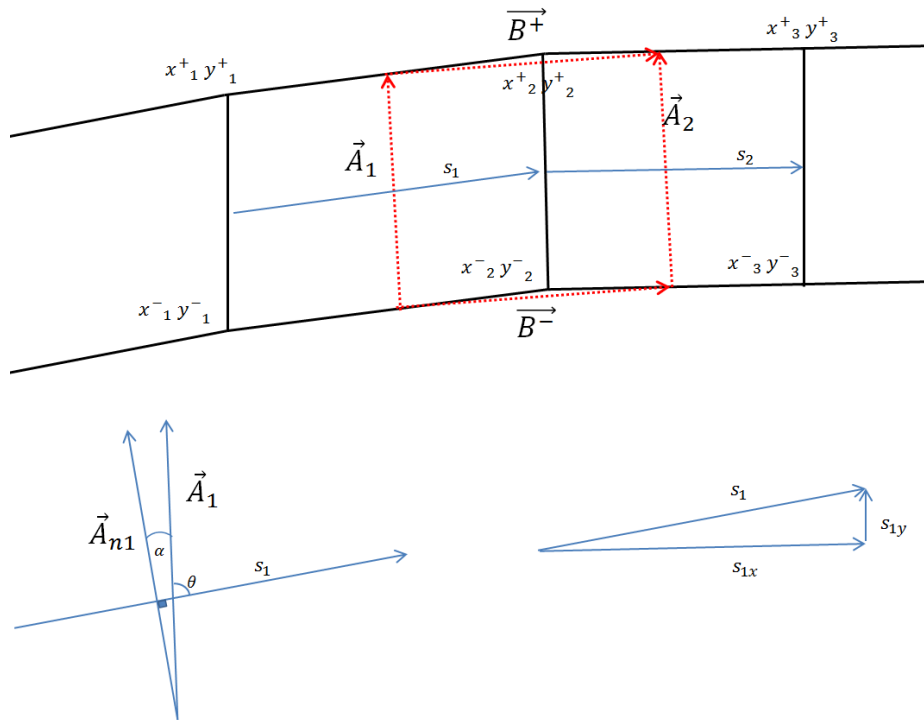


Figure 3.6 Normal area vector components [3]

Referring to Figure 3.4 and using the identity $\vec{B}^+ - \vec{B}^- = \vec{A}_2 - \vec{A}_1$, the x and y momentum equations can be rewritten as

$$\begin{aligned} \rho_1 q_1^2 A_{n1} \hat{s}_{x1} - \rho_2 q_2^2 A_{n2} \hat{s}_{x2} + (p_1 - p_2) N_y + (\pi^+ - \pi^-) S_y \\ + \frac{1}{2} (\pi^+ + \pi^- - p_1 - p_2) (B_y^+ - B_y^-) \\ = \rho_{av} A_c (-2\Omega q \sin\beta \sin\phi + \Omega^2 r \sin\phi) \end{aligned} \quad (3.26)$$

and

$$\begin{aligned}
& \rho_1 q_1^2 A_{n1} \hat{s}_{y1} - \rho_2 q_2^2 A_{n2} \hat{s}_{y2} - (p_1 - p_2) N_x - (\pi^+ - \pi^-) S_x \\
& - \frac{1}{2} (\pi^+ + \pi^- - p_1 - p_2) (B_x^+ - B_x^-) \\
& = \rho_{av} A_c (2\Omega q \cos\beta \sin\phi)
\end{aligned} \tag{3.27}$$

respectively.

In the momentum equations (3.26) and (3.27), at the left hand side of the equations, first two terms are the momentum fluxes which enters and go out from a computational cell. The remaining terms are the surface pressure forces coming from cell faces and streamlines. It can be easily seen that the only contribution comes from the streamlines are the pressure forces which are the $\pi^+ B_x^+$ and $\pi^- B_x^-$ terms. The right hand side of the equations includes the body forces which were explained in detail previously. The important thing about the body forces is that they affect the whole computational volume. Therefore, this volumetric effect can be given to the momentum equation by calculating the body force at the cell centers. That is why, the body forces are multiplied by average density (ρ_{av}) and computational cell volume (A_c). By referring to Figure 3.5 the average quantities in the body force terms can be evaluated at the cell centers as

$$\rho_{av} = \frac{1}{2} (\rho_1 + \rho_2) \tag{3.28}$$

$$q_{av} = \frac{1}{2} (q_1 + q_2) \tag{3.29}$$

$$A_c = \frac{1}{2} [|\vec{B}^- \times \vec{A}_1| + |\vec{B}^+ \times \vec{A}_2|] \tag{3.30}$$

Together with the flow variables which are evaluated at the cell faces, the pressure values on the streamlines (π pressure values) need to be calculated. The equation which combines the face pressure values (p_1 and p_2) and π pressure values is called “auxiliary pressure relation”. The auxiliary pressure equation can be written as

$$\pi^- + \pi^+ = p_1 + p_2 + 2p_c \quad (3.31)$$

where, p_c is the correction value. The correction value is necessary because it prevents the sawtooth type grid oscillations. The details about pressure correction are available in reference [6]. The relation which gives the correction term can be given as

$$p_c = \begin{cases} k\gamma p' M^2 (1 - M^2) \frac{|\vec{s}_1^- \times \vec{s}_2^-| - |\vec{s}_1^+ \times \vec{s}_2^+|}{2|\vec{S} \times \vec{N}|}, & M^2 < 1 \\ 0, & M^2 \geq 1 \end{cases} \quad (3.32)$$

where,

$$p' = \frac{1}{2} (p_1 + p_2) \quad (3.33)$$

$$M^2 = \frac{1}{2} (M_1^2 + M_2^2) \quad (3.34)$$

$$0.05 < k < 0.20 \quad (3.35)$$

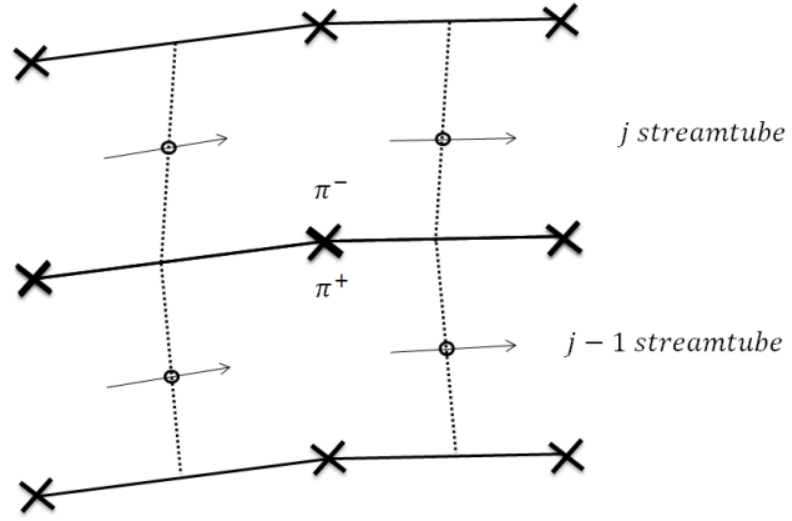


Figure 3.7 Pressure compatibility between two adjacent streamtubes [6]

Since the grid is intrinsic streamline grid and the horizontal tubes are actually the streamtubes of the flow, the x and y momentum equations are written in terms of streamwise and normal momentum equation in $\vec{S}$ and $\vec{N}$ direction respectively. In order to obtain S momentum the following manipulation is made:

$$[(S_x) (x_momentum\ equation) + (S_y) (y_momentum\ equation)] / (|\vec{S} \times \vec{N}|)$$

The S momentum equation can be written as

$$\begin{aligned} \rho_1 q_1^2 f_1 - \rho_2 q_2^2 f_2 + p_1 - p_2 + P_c f_3 \\ + \rho_{av} f_4 (2\Omega q \sin\beta \sin\phi S_x - 2\Omega q \cos\beta \sin\phi S_y \\ - \Omega^2 r \sin\phi S_x) = 0 \end{aligned} \quad (3.36)$$

where

$$f_1 = \frac{A_{n1} \hat{s}_1 \cdot \vec{S}}{|\vec{S} \times \vec{N}|} \quad (3.37)$$

$$f_2 = \frac{A_{n2} \hat{S}_2 \cdot \vec{S}}{|\vec{S} \times \vec{N}|} \quad (3.38)$$

$$f_3 = \frac{|\vec{B}^- + \vec{B}^+|}{|\vec{S} \times \vec{N}|} \quad (3.39)$$

$$f_4 = \frac{A_c}{|\vec{S} \times \vec{N}|} \quad (3.40)$$

The right hand side of the Equation (3.36) is the source term and this source term can further be simplified by using below relation

$$\cos \beta = \frac{S_y}{S_x} \quad (3.41)$$

Then, the new S momentum equation becomes

$$\rho_1 q_1^2 f_1 - \rho_2 q_2^2 f_2 + p_1 - p_2 + P_c f_3 + R_s f_4 = 0 \quad (3.42)$$

where R_s is the new source term and it is written as

$$R_s = -\rho_{av} \Omega^2 r \sin \phi S_x \quad (3.43)$$

Similar approach used for derivation of S momentum equation is used in the derivation of the N momentum equation. This time x momentum equation is multiplied by N_x and y momentum equation is multiplied by N_y then both of them is divided by $|\vec{S} \times \vec{N}|$. The resulting equation is called normal momentum equation and is written as

$$\rho_1 q_1^2 g_1 - \rho_2 q_2^2 g_2 - \pi^+ + \pi^- + P_c g_3 + R_n g_4 = 0 \quad (3.44)$$

where

$$g_1 = \frac{A_{n1} \hat{s}_1 \cdot \vec{N}}{|\vec{S} \times \vec{N}|} \quad (3.45)$$

$$g_2 = \frac{A_{n2} \hat{s}_2 \cdot \vec{N}}{|\vec{S} \times \vec{N}|} \quad (3.46)$$

$$g_3 = \frac{|\vec{A}_1 + \vec{A}_2|}{|\vec{S} \times \vec{N}|} \quad (3.47)$$

$$g_4 = \frac{A_c}{|\vec{S} \times \vec{N}|} \quad (3.48)$$

$$R_n = \rho_{av} (2\Omega q_{av} \sin\beta \sin\phi N_x - 2\Omega q_{av} \cos\beta \sin\phi N_y - \Omega^2 r \sin\phi N_x) \quad (3.49)$$

The N momentum equation is solved through two adjacent streamtubes using the pressure compatibility relation. That is, since the pressure values on both sides of the edge of the streamlines are equal, the momentum equation between two adjacent streamlines can be solved together. Therefore, referring to Figure 3.7, combination of the pressure compatibility relation (3.31) with the normal momentum Equation (3.44) gives

$$\begin{aligned} 2\pi_{i,j-1}^+ - 2\pi_{i,j}^- &= (\rho_1 q_1^2 g_1 - \rho_2 q_2^2 g_2 - p_1 - p_2 + P_c g_3 + R_n g_4 - \\ &2p_c)_j + (\rho_1 q_1^2 g_1 - \rho_2 q_2^2 g_2 + p_1 + p_2 + P_c g_3 + R_n g_4 - 2p_c)_{j-1} = 0 \end{aligned} \quad (3.50)$$

Finally, Equations (3.42) and (3.50) are the main equations to be solved and they will be linearized later in this chapter.

3.2. Artificial Compressibility

The compressible flow equations are elliptic in the subsonic region and hyperbolic in the supersonic region. When the flow is completely subsonic or completely supersonic, there is no problem with the solution technique. However, if the flow is transonic, some part of the flow is subsonic (elliptic) and some part of the flow is supersonic (hyperbolic). Artificial compressibility concept makes the transonic flow solution possible for elliptic flow equations. In the Newton's method, the Euler equations are essentially elliptic. In order to have upstream dependence in the supersonic region (or close to supersonic), artificial compressibility is used. In reference [6], the density is modified in the supersonic region using upwind approach. Therefore, density will be modified in the supersonic region as follows:

$$\tilde{\rho} = \rho_2 - \mu(\rho_2 - \rho_1) + \mu_c(\rho_1 - \rho_0) \quad (3.51)$$

where

$$\mu = \begin{cases} 0, & M < M_{cr} \\ \mu_{con} \left(\frac{M^2 - M_{cr}^2}{(\gamma + 1)M^2} \right), & M \geq M_{cr} \end{cases} \quad (3.52)$$

Detailed information about μ , μ_{con} and μ_c can be found in references [3], [6], [34] and [35].

3.3. Boundary Conditions

The program includes two type of boundary conditions which are duct boundary conditions and cascade (or blade to blade) boundary conditions. The physical boundary conditions are used at inlet and outlet of the domain. On the wall, the node displacement is set to zero and no thermodynamic variable is specified on the wall. Periodicity is given as equivalence of pressure and grid nodal displacement on the stagnation streamlines for blade to blade domain.

The boundary conditions at the inlet and outlet are modified for the choked flow. For choked flow, the unique mass flow is determined using inlet total density, inlet total temperature and throat area. At the inlet, total enthalpy, mass flow rate and inlet flow angle are given. The behavior of the flow after the throat is determined by the back pressure. Therefore, it is necessary to specify mass averaged total density or back pressure at the exit plane of the domain.

3.3.1. Duct Boundary Conditions

The duct boundary conditions consist of simply inlet boundary, wall boundary and outlet boundary as shown in Figure 3.8.

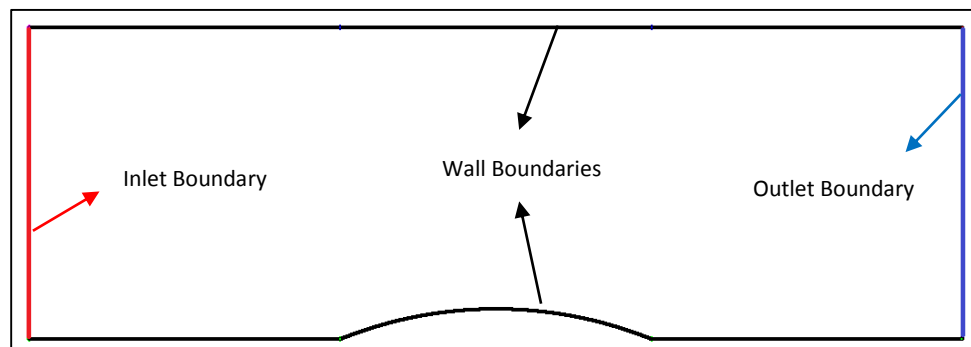


Figure 3.8 Duct boundary conditions

There are three flow quantities at the inlet. The mass flow rate and stagnation enthalpy is specified through each streamtube at the inlet of the domain. Third quantity at the inlet can be selected either total density or inlet Mach number. In duct boundary condition section, inlet total density condition will be given. Inlet Mach number condition will be explained in cascade boundary condition section. Since discrete Euler equation includes the displacement of the node, a boundary condition is required for the position of the nodes at the inlet. The inlet mass flow, total density, inlet stagnation enthalpy and geometric boundary conditions are specified as

$$\dot{m}_j = \dot{m}_{total} \frac{(Area)_j}{\sum_{j=1}^{JJ-1} (Area)_j}, \quad 1 \leq j \leq JJ - 1 \quad (3.53)$$

$$\begin{aligned} \rho_{tj} &= \rho_1 \left(1 + \frac{\gamma - 1}{2} M^2 \right)^{\frac{1}{\gamma-1}} \\ &= \rho_1 \left(1 - \frac{q_1^2}{2h_t} \right)^{-\frac{1}{\gamma-1}}, \end{aligned} \quad 1 \leq j \leq JJ - 1 \quad (3.54)$$

$$(h_t)_j = (h_{t_{spec}})_j, \quad 1 \leq j \leq JJ - 1 \quad (3.55)$$

$$Y_{1,j+1} - Y_{1,j} = (Y_{1,JJ} - Y_{1,1}) \frac{\dot{m}_j}{\dot{m}_{total}}, \quad 1 \leq j \leq JJ - 1 \quad (3.56)$$

respectively. The indices “ j ” represent the stations of the streamtubes of the flow. The final boundary condition (3.56) is for the nodal displacement. Since both $Y_{1,JJ} - Y_{1,1}$ and $\frac{\dot{m}_j}{\dot{m}_{total}}$ are specified at the inlet, the area of the each streamtube is constant. That is, the node at the inlet plane does not move in any direction.

At the outlet, no thermodynamic variable is specified otherwise the system becomes overdefined. However, the boundary condition for nodal displacement is required. At the outlet, same approach used at the inlet cannot be used since the mass flow may not be uniform due to entropy production at the shocks. The appropriate boundary condition for the outlet is that the area of the I^{th} and $(I-1)^{th}$ calculation station are the same. That is,

$$(Y_{i,j+1} - Y_{1,j}) - (Y_{i-1,j+1} - Y_{i-1,j}) = 0, \quad 1 \leq j \leq JJ - 1 \quad (3.57)$$

The solid wall boundary condition is very simple. Only the position of the grid nodes are specified. For wall boundary condition, the displacement of the wall nodes are set to zero.

$$\delta n_{i,1} = \delta n_{i,JJ} = 0, \quad 1 \leq j \leq II \quad (3.58)$$

Choked Boundary Condition Modification

The detailed derivation of the choked boundary condition is available in reference [6]. Here, only the result will be presented. The Equation (3.54) need to be modified for choking conditions such that

$$(\rho_t)_{j+1} - (\rho_t)_{j+1} = (\Delta\rho_t)_{specified}, \quad 1 \leq j \leq JJ - 1 \quad (3.59)$$

Generally, at the inlet there is no entropy generation so $(\Delta\rho_t)_{specified}$ is set to zero. At the outlet, average stagnation density must be specified such that

$$\sum_{j=1}^{j-1} \frac{m_j \rho_{tj}}{m_{tot}} = \rho_{t_{specified}}, \quad 1 < j < JJ - 1 \quad (3.60)$$

For choking flow, the mass flow is determined uniquely by the inlet conditions and the geometry of the domain such that

$$\dot{m}_{choke} = \rho_{t_{inlet}} \sqrt{\gamma R T_{t_{inlet}}} A_{n_{inlet}} \frac{M_{inlet}}{\left(1 + \frac{\gamma - 1}{2} M_{inlet}^2\right)^{\frac{\gamma + 1}{2(\gamma - 1)}}} \quad (3.61)$$

If the inlet mass flow is assumed different than that of the choking condition, the solution does not converge. Therefore, for a fixed total inlet density and temperature,

there is a unique Mach number at the inlet. For example, if the flow is choked at the throat of converging-diverging nozzle, there must be only one mass flow rate that satisfies the choking mass flow.

3.3.2. Boundary Conditions for Blade to Blade Domain

In addition to the boundary conditions specified for duct solver, the periodic boundary condition must be implemented for blade to blade domain.

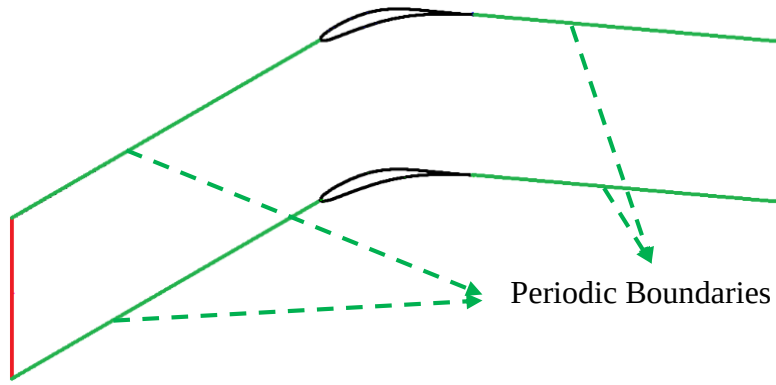


Figure 3.9 Blade to blade boundary conditions

At the inlet, through each streamtube, Mach number (or total density), rotationality, mass flow rate are specified as a thermodynamic boundary conditions. In the case of stator problem, rotationality automatically reduces to total enthalpy.

$$M_{1,j} = M_{1spec,j} , \quad 1 \leq j \leq JJ - 1 \quad (3.62)$$

$$(I)_j = (I_{spec})_j , \quad 1 \leq j \leq JJ - 1 \quad (3.63)$$

$$\dot{m} = \rho_{t_{in}} \sqrt{\gamma R T_{t_{in}}} A_{in} \frac{M_{in}}{\left(1 + \frac{\gamma-1}{2} M_{in}^2\right)^{\frac{\gamma+1}{2(\gamma-1)}}} \quad 1 \leq j \leq JJ \quad (3.64)$$

Additionally, the inlet slope need to be fixed in order to be compatible with the slope coming from throughflow solver. Therefore, inlet slope should be specified as an inlet boundary condition.

$$s_{in} = s_{in_{spec}} \quad (3.65)$$

where

$$s_{in} = \frac{y_{2,j} - y_{1,j}}{x_{2,j} - x_{1,j}} \quad 1 \leq j \leq JJ - 1 \quad (3.66)$$

At the outlet, no thermodynamic variable is specified. The only boundary condition is related to the exit slope. Specifying the exit slope is not physical because the exit flow angle will be determined with respect to flow conditions and blade geometry. Therefore, the exit flow angle is determined by the trailing edge Kutta condition and this will be explained in global equation section.

The wall boundary condition is the same as that of duct domain. The only difference is that Equation (3.58) is applied in the interval of $[ILE, ITE]$. In addition to the wall specified in duct boundary conditions, periodicity must be given to the equations upstream and downstream of the blade passage. There are two parameters which defines periodicity in the flow domain. Firstly, nodal displacement on upper and lower stagnation streamlines are forced to be equal such that the stagnation streamlines remain parallel each other.

$$\delta n_{i,1} = \delta n_{i,II} \quad 2 < i < ILE \text{ and } ITE + 1 < i < II - 1 \quad (3.67)$$

The second parameter is the pressure on the stagnation streamlines. The pressure values on the same grid must be equal. Therefore,

$$\pi_{i,1}^- = \pi_{i,II-1}^+ \quad 2 < i < ILE \text{ and } ITE + 1 < i < II - 1 \quad (3.68)$$

Choked boundary condition modification is exactly the same as that of duct flow. Equations (3.59), (3.60) and (3.61) are applicable also for blade to blade flow domain.

3.4. Global Equation and Variables

Since it includes only nodal and density displacements, any other variable cannot be located into the momentum equation. Some of the variables calculated separately from the discretized flow equations. These variables are inlet slope S_{inlet} , outlet slope, S_{out} and leading edge movement, S_{Le} . Also, there are some constraints for the global variables and known as global constraints. For the inlet slope, the global equation and global constraints are the Equation (3.65) and (3.66) respectively.

The outlet slope can be modelled similar to that of inlet slope. However, specification of inlet and outlet slope together is not physical and may cause pressure jump across the trailing edge. Therefore, instead of specifying the outlet slope, trailing edge Kutta condition is applied. The outlet slope is determined by the flow condition and blade geometry. Outlet slope and trailing edge Kutta condition is given as

$$S_{out} = \frac{y_{II,j} - y_{II-1,j}}{x_{II,j} - x_{II-1,j}}, \quad 1 \leq j \leq II \quad (3.69)$$

$$\pi_{ITE,1}^- = \pi_{ITE,II-1}^+ \quad (3.70)$$

The final global equation is related to the leading edge movement of the blade with blunt leading edge. For a blade with blunt leading edge, the location of the stagnation point is part of the solution. The stagnation point is initially assumed and it will be updated at each iteration step. The movement of the leading edge is the additional unknown (δs_{le}) and treated as a global variable. This variable is constrained by the leading edge Kutta conditions and it is defined as

$$\pi_{ILE,1}^- = \pi_{ILE,JJ-1}^+ \quad (3.71)$$

Detailed information about the implementation of the global equations and variables are available in the code listing section of reference [6].

3.5. Newton Raphson Method and Linearization Procedure

In order to minimize the computational cost, it is desirable to use the smallest possible number of unknowns. The discretized momentum equations together with boundary condition and global equations are linearized using the Newton Raphson method. Then, using discrete mass equation, ideal gas law and energy equation, the momentum equations are manipulated. After linearization is applied, the numbers of unknowns are reduced to two.

The Newton Raphson method is well known method for the solution of the elliptic equations. For a scalar system, the Newton Raphson [45] method can be explained as

$$z(u) = 0 \quad (3.72)$$

where, z is the nonlinear function. The variable u is the solution to be obtained after convergence is satisfied. Since the solution procedure is iterative, the process starts with an initial guess. The available approximate value is represented by subscript “ n ” and updated approximate value is denoted by superscript “ $n+1$ ”.

$$u^{n+1} = u^n + \Delta u^n \quad (3.73)$$

It is necessary to find an expression for Δu^n in the above Equation (3.73). The function z for the next value is

$$z(u^{n+1}) \equiv z(u^n + \Delta u^n) \quad (3.74)$$

The Newton Raphson method forces the above equation to zero. The change in the variable “ u ” can be obtained as

$$z(u^{n+1}) \equiv z(u^n + \Delta u^n) \equiv z(u^n) + \frac{\partial z}{\partial u} \Delta u^n \equiv 0$$

$$\Delta u^n = \frac{-z(u^n)}{\left(\frac{dz}{du}\right)^n} \quad (3.75)$$

It is easily seen that, the change in the variable u at the new level is determined from the known state. This procedure continues until Δu is reduced to prescribed accuracy level.

Similar procedure can also be applied to a system of M nonlinear equations with M unknowns.

$$\vec{Z}(\vec{u}) = 0 \quad (3.76)$$

$$\vec{u}^{n+1} = \vec{u}^n + \Delta \vec{u}^n \quad (3.77)$$

$$\vec{Z}(\vec{u}^{n+1}) \equiv \vec{Z}(\vec{u}^n + \Delta \vec{u}^n) \quad (3.78)$$

$$\begin{aligned}\vec{Z}(\vec{u}^{n+1}) &\equiv \vec{Z}(\vec{u}^n + \Delta\vec{u}^n) \equiv \vec{Z}(\vec{u}^n) + \left[\frac{\partial \vec{Z}}{\partial \vec{u}}\right]_{M \times M}^n \Delta\vec{u}^n \equiv 0 \\ \Delta\vec{u}^n &= \frac{-\vec{Z}(\vec{u}^n)}{\left[\frac{\partial \vec{Z}}{\partial \vec{u}}\right]_{M \times M}^n}\end{aligned}\tag{3.79}$$

$\left[\frac{\partial \vec{Z}}{\partial \vec{u}}\right]_{M \times M}^n$ is called Jacobian matrix which is evaluated at the present calculation step. $\vec{Z}(\vec{u}^n)$ is called residual vector since approximate solution $\vec{u}^n$ does not satisfy Equation (3.76) directly. In the next three sections, Newton's procedure will be applied interior Euler equation, boundary conditions and global equations.

3.5.1. Interior Euler Equation Linearization

Before linearization, functional dependencies are required in order to clarify the linearization procedure. The functional dependencies of the geometric variables are written as

$$S = S(x(n), y(n))\tag{3.80a}$$

$$N = N(x(n), y(n))\tag{3.80b}$$

$$s = s(x(n), y(n))\tag{3.80c}$$

$$A = A(x(n), y(n))\tag{3.80d}$$

The functional dependencies of the thermodynamic variables can be written as

$$q = q(\rho, A)\tag{3.81a}$$

$$p = p(\rho, q)\tag{3.81b}$$

The Equations (3.42) and (3.50) are linearized as

$$\begin{aligned}
& q_1^2 f_1 \delta \rho_1 + 2\rho_1 q_1 f_1 \delta q_1 + \rho_1 q_1^2 \delta f_1 - q_2^2 f_2 \delta \rho_2 - 2\rho_2 q_2 f_2 \delta q_2 \\
& - \rho_2 q_2^2 \delta f_2 + \delta p_1 - \delta p_2 + f_3 \delta P_c + P_c \delta f_3 + f_4 \delta R_s \\
& + R_s \delta f_4 \\
& = -\rho_1 q_1^2 f_1 + \rho_2 q_2^2 f_2 - p_1 + p_2 - P_c f_3 - R_s f_4
\end{aligned} \tag{3.82}$$

$$\begin{aligned}
& [q_1^2 g_1 \delta \rho_1 + 2\rho_1 q_1 g_1 \delta q_1 + \rho_1 q_1^2 \delta g_1 - q_2^2 g_2 \delta \rho_2 - 2\rho_2 q_2 g_2 \delta q_2 - \\
& \rho_2 q_2^2 \delta g_2 - \delta p_1 - \delta p_2 + (g_3 - 2) \delta P_c + P_c \delta g_3 + g_4 \delta R_n + \\
& R_n \delta g_4]_j - [q_1^2 g_1 \delta \rho_1 + 2\rho_1 q_1 g_1 \delta q_1 + \rho_1 q_1^2 \delta g_1 - q_2^2 g_2 \delta \rho_2 - \\
& 2\rho_2 q_2 g_2 \delta q_2 - \rho_2 q_2^2 \delta g_2 + \delta p_1 + \delta p_2 + (g_3 + 2) \delta P_c + P_c \delta g_3 + \\
& g_4 \delta R_n + R_n \delta g_4]_{j-1} = 2\pi_{i,j}^- - 2\pi_{i,j-1}^+
\end{aligned} \tag{3.83}$$

where, δp and δn are the primitive unknowns in the linearized equations. Equations (3.82) and (3.83) are the linearized form of the S momentum and N momentum equations respectively. The remaining linearized terms (δq , δp , δA ...) are manipulated using continuity and energy equations. In order to linearize the pressure term the energy equation will be used

$$I_t = \frac{\gamma}{\gamma - 1} \frac{p}{\rho} + \frac{1}{2} q^2 - \frac{1}{2} (\Omega r)^2 = \text{constant} \tag{3.84}$$

From Equation (3.84), the pressure relation can be obtained as

$$p = \left(I_t + \frac{1}{2} (\Omega r)^2 - \frac{1}{2} q^2 \right) \frac{\rho(\gamma - 1)}{\gamma} \tag{3.85}$$

Using Equations (3.81) and (3.85) the pressure term is linearized as

$$\delta p = \frac{\partial p}{\partial \rho} \delta \rho + \frac{\partial p}{\partial q} \delta q \quad (3.86)$$

where

$$\frac{\partial p}{\partial \rho} = \frac{p}{\rho} \quad (3.87a)$$

$$\frac{\partial p}{\partial q} = -\frac{\gamma - 1}{\gamma} \rho q \quad (3.87b)$$

The relative velocity q is linearized using the continuity equation. Since the mass flux through each streamtube is constant, the relative velocity can be written as

$$q = \frac{\dot{m}}{\rho A_n} \quad (3.88)$$

Using the Equations (3.81) and (3.88), the relative velocity is linearized as

$$\delta q = \frac{\partial q}{\partial \rho} \delta \rho + \frac{\partial q}{\partial A_n} \delta A_n \quad (3.89)$$

where

$$\frac{\partial q}{\partial \rho} = \frac{q}{\rho} \quad (3.90a)$$

$$\frac{\partial q}{\partial A_n} = -\frac{q}{A_n} \quad (3.90b)$$

Definitions of the geometric variables and terms corresponding to the computational stencil (see Figure 3.6) were given before. The change in the normal area can be obtained considering functional dependencies and chain rule as

$$\delta A_{n1} = \frac{\partial A_n}{\partial n_1^+} \delta n_1^+ + \frac{\partial A_n}{\partial n_1^-} \delta n_1^- + \frac{\partial A_n}{\partial n_2^+} \delta n_2^+ + \frac{\partial A_n}{\partial n_2^-} \delta n_2^- \quad (3.91)$$

The linearization of the area which is normal to the relative flow direction is a quite long expression. Therefore, each term in Equation (3.91) will be explained term by term.

At this point, the relation between normal and the local grid coordinates (x, y) must be established. Each grid node is constrained to move in the direction of unit vector. This unit vector is taken perpendicular to the streamline. Therefore, the changes δx and δy are calculated as

$$\delta x = \frac{\partial x}{\partial n} \delta n = n_x \delta n \quad \text{and} \quad \delta y = \frac{\partial y}{\partial n} \delta n = n_y \delta n \quad (3.92)$$

The values of n_x and n_y are calculated before each iteration so they are known constants. Then, the first term of the Equation (3.91) can be written as

$$\frac{\partial A_n}{\partial n_1^+} = \frac{\partial A_n}{\partial x_1^+} \delta x_1^+ + \frac{\partial A_n}{\partial y_1^+} \delta y_1^+ = \frac{\partial A_n}{\partial x_1^+} n_{x1}^+ \delta n_1^+ + \frac{\partial A_n}{\partial y_1^+} n_{y1}^+ \delta n_1^+ \quad (3.93)$$

The remaining terms $\frac{\partial A_n}{\partial n_1^-}$, $\frac{\partial A_n}{\partial n_2^-}$ and $\frac{\partial A_n}{\partial n_2^+}$ can be linearized in a similar way.

The other geometric linearized terms appearing in Equations (3.82) and (3.83) are δf_1 , δf_2 , δf_3 , δg_1 , δg_2 and δg_3 . These are all similar expressions and their linearization is long but straight forward procedure. Here, only one of them will be explained in detail and others are linearized in a similar way.

$$\delta f_1 = \frac{\partial f_1}{\partial A_{n1}} \delta A_{n1} + \frac{\partial f_1}{\partial (\vec{S} \cdot \hat{s}_1)} \delta (\vec{S} \cdot \hat{s}_1) + \frac{\partial f_1}{\partial (\vec{S} \times \vec{N})} \delta (\vec{S} \times \vec{N}) \quad (3.94)$$

In the above equation, the term δA_{n1} was already calculated. The remaining terms can be linearized as

$$\frac{\partial f_1}{\partial A_{n1}} = \frac{\hat{s}_1 \cdot \vec{S}}{|\vec{S} \times \vec{N}|} \quad (3.95a)$$

$$\frac{\partial f_1}{\partial (\vec{S} \cdot \hat{s}_1)} = \frac{A_{n1}}{|\vec{S} \times \vec{N}|} \quad (3.95b)$$

$$\frac{\partial f_1}{\partial (\vec{S} \times \vec{N})} = -\frac{f_1}{|\vec{S} \times \vec{N}|} \quad (3.95c)$$

The term $(\vec{S} \cdot \hat{s}_1)$ should be expanded before the linearization. The dot product of the each vector using the definitions of the $\vec{S}$ and $\hat{s}_1$ gives

$$\begin{aligned} \vec{S} \cdot \hat{s}_1 &= S_x \hat{s}_x + S_y \hat{s}_y \\ &= \frac{\left\{ \left[\frac{1}{4} (x_3^+ - x_1^+ + x_3^- - x_1^-) \right] \left[\frac{1}{2} (x_2^- + x_2^+) - \frac{1}{2} (x_1^- + x_1^+) \right] \right\}}{s_1} \\ &\quad + \frac{\left\{ \left[\frac{1}{4} (y_3^+ - y_1^+ + y_3^- - y_1^-) \right] \left[\frac{1}{2} (y_2^- + y_2^+) - \frac{1}{2} (y_1^- + y_1^+) \right] \right\}}{s_1} \end{aligned} \quad (3.96)$$

From the above equation, the functional dependency of the $\vec{S} \cdot \hat{s}_1$ can be written as

$$\vec{S} \cdot \hat{s}_1 = f(x_1^+, x_1^-, x_2^+, x_2^-, x_3^+, x_3^-, y_1^+, y_1^-, y_2^+, y_2^-, y_3^+, y_3^-) \quad (3.97)$$

Then the term $\delta(\vec{S} \cdot \hat{s}_1)$ can be expressed as

$$\begin{aligned}
\delta(\vec{S} \cdot \hat{s}_1) = & \left(\frac{\partial(\vec{S} \cdot \hat{s}_1)}{\partial x_1^+} n_{x_1}^+ + \frac{\partial(\vec{S} \cdot \hat{s}_1)}{\partial y_1^+} n_{y_1}^+ \right) \delta n_1^+ \\
& + \left(\frac{\partial(\vec{S} \cdot \hat{s}_1)}{\partial x_1^-} n_{x_1}^- + \frac{\partial(\vec{S} \cdot \hat{s}_1)}{\partial y_1^-} n_{y_1}^- \right) \delta n_1^- \\
& + \left(\frac{\partial(\vec{S} \cdot \hat{s}_1)}{\partial x_2^+} n_{x_2}^+ + \frac{\partial(\vec{S} \cdot \hat{s}_1)}{\partial y_2^+} n_{y_2}^+ \right) \delta n_2^+ \\
& + \left(\frac{\partial(\vec{S} \cdot \hat{s}_1)}{\partial x_2^-} n_{x_2}^- + \frac{\partial(\vec{S} \cdot \hat{s}_1)}{\partial y_2^-} n_{y_2}^- \right) \delta n_2^- \\
& + \left(\frac{\partial(\vec{S} \cdot \hat{s}_1)}{\partial x_3^+} n_{x_3}^+ + \frac{\partial(\vec{S} \cdot \hat{s}_1)}{\partial y_3^+} n_{y_3}^+ \right) \delta n_3^+ \\
& + \left(\frac{\partial(\vec{S} \cdot \hat{s}_1)}{\partial x_3^-} n_{x_3}^- + \frac{\partial(\vec{S} \cdot \hat{s}_1)}{\partial y_3^-} n_{y_3}^- \right) \delta n_3^- \\
& + \frac{\partial(\vec{S} \cdot \hat{s}_1)}{\partial s_1} \left[\left(\frac{\partial s_l}{\partial x_l^+} n_{x_l}^+ + \frac{\partial s_l}{\partial y_l^+} n_{y_l}^+ \right) \delta n_1^+ \right. \\
& + \left(\frac{\partial s_1}{\partial x_1^-} n_{x_1}^- + \frac{\partial s_1}{\partial y_1^-} n_{y_1}^- \right) \delta n_1^- \\
& + \left(\frac{\partial s_1}{\partial x_2^+} n_{x_2}^+ + \frac{\partial s_1}{\partial y_2^+} n_{y_2}^+ \right) \delta n_2^+ \\
& \left. + \left(\frac{\partial s_1}{\partial x_2^-} n_{x_2}^- + \frac{\partial s_1}{\partial y_2^-} n_{y_2}^- \right) \delta n_2^- \right]
\end{aligned} \tag{3.98}$$

In Equation (3.98), the term $\frac{\partial(\vec{S} \cdot \hat{s}_1)}{\partial s_1}$ can be evaluated as

$$\frac{\partial(\vec{S} \cdot \hat{s}_1)}{\partial s_1} = - \frac{\vec{S} \cdot \hat{s}}{s_1} \tag{3.99}$$

For the last term in Equation (3.94), the term $\vec{S} x \vec{N}$ is written as

$$\begin{aligned}
\vec{S}x\vec{N} &= S_x N_y + S_y N_x \\
&= \frac{1}{4} [(x_3^+ - x_1^+ + x_3^- - x_1^-)] \frac{1}{4} [(y_1^+ - y_1^- + 2y_2^+ \\
&\quad - 2y_2^- + y_3^+ + y_3^-)] \\
&\quad + \frac{1}{4} [(y_3^+ - y_1^+ + y_3^- - y_1^-)] \frac{1}{4} [(x_1^+ - x_1^- + 2x_2^+ \\
&\quad - 2x_2^- + x_3^+ + x_3^-)]
\end{aligned} \tag{3.100}$$

From the above equation, the term $\vec{S}x\vec{N}$ depends on

$$\vec{S}x\vec{N} = f(x_1^+, x_1^-, x_2^+, x_2^-, x_3^+, x_3^-, y_1^+, y_1^-, y_2^+, y_2^-, y_3^+, y_3^-) \tag{3.101}$$

Finally, the term $\delta(\vec{S}x\vec{N})$ is linearized as

$$\begin{aligned}
\delta(\vec{S}x\vec{N}) &= \left(\frac{\partial(\vec{S} \times \vec{N})}{\partial x_1^+} n_{x_1^+} + \frac{\partial(\vec{S} \times \vec{N})}{\partial y_1^+} n_{y_1^+} \right) \delta n_1^+ \\
&\quad + \left(\frac{\partial(\vec{S} \times \vec{N})}{\partial x_1^-} n_{x_1^-} + \frac{\partial(\vec{S} \times \vec{N})}{\partial y_1^-} n_{y_1^-} \right) \delta n_1^- \\
&\quad + \left(\frac{\partial(\vec{S} \times \vec{N})}{\partial x_2^+} n_{x_2^+} + \frac{\partial(\vec{S} \times \vec{N})}{\partial y_2^+} n_{y_2^+} \right) \delta n_2^+ \\
&\quad + \left(\frac{\partial(\vec{S} \times \vec{N})}{\partial x_2^-} n_{x_2^-} + \frac{\partial(\vec{S} \times \vec{N})}{\partial y_2^-} n_{y_2^-} \right) \delta n_2^- \\
&\quad + \left(\frac{\partial(\vec{S} \times \vec{N})}{\partial x_3^+} n_{x_3^+} + \frac{\partial(\vec{S} \times \vec{N})}{\partial y_3^+} n_{y_3^+} \right) \delta n_3^+ \\
&\quad + \left(\frac{\partial(\vec{S} \times \vec{N})}{\partial x_3^-} n_{x_3^-} + \frac{\partial(\vec{S} \times \vec{N})}{\partial y_3^-} n_{y_3^-} \right) \delta n_3^-
\end{aligned} \tag{3.102}$$

The linearization of the corrected pressure is again straight forward but quite a long expression. First of all, the functional dependencies of the corrected pressure term is

$$P_c = f(p_1, p_2, q_1, q_2, \rho_1, \rho_2, x_1^+, x_1^-, x_2^+, x_2^-, x_3^+, x_3^-, y_1^+, y_1^-, y_2^+, y_2^-, y_3^+, y_3^-) \quad (3.103)$$

Therefore, linearized form of the corrected pressure term is

$$\begin{aligned} \delta P_c = & \frac{\partial P_c}{\partial p_1} \delta p_1 + \frac{\partial P_c}{\partial p_2} \delta p_2 + \frac{\partial P_c}{\partial q_1} \delta q_1 + \frac{\partial P_c}{\partial q_2} \delta q_2 + \frac{\partial P_c}{\partial \rho_1} \delta \rho_1 + \frac{\partial P_c}{\partial \rho_2} \delta \rho_2 \\ & + \frac{\partial P_c}{\partial x_1^+} n_{x_1^+} \delta n_1^+ + \frac{\partial P_c}{\partial x_1^-} n_{x_1^-} \delta n_1^- + \frac{\partial P_c}{\partial x_2^+} n_{x_2^+} \delta n_2^+ \\ & + \frac{\partial P_c}{\partial x_2^-} n_{x_2^-} \delta n_2^- + \frac{\partial P_c}{\partial x_3^+} n_{x_3^+} \delta n_3^+ + \frac{\partial P_c}{\partial x_3^-} n_{x_3^-} \delta n_3^- \end{aligned} \quad (3.104)$$

The above equation is a very long expression and all linearized terms will not be written explicitly. The full details are available in the code listing part of reference [6].

The only remaining linearized parts are related to the source terms which are R_n and R_s . Then, functional dependencies are written as

$$R_s = f(\rho_1, \rho_2, x_1^+, x_1^-, x_3^+, x_3^-, y_1^+, y_1^-, y_3^+, y_3^-) \quad (3.105)$$

Then the linearization is carried out as

$$\begin{aligned}
\delta R_s = & \frac{\partial R_s}{\partial \rho_1} \delta \rho_1 + \frac{\partial R_s}{\partial \rho_2} \delta \rho_2 \\
& + \frac{\partial R_s}{\partial S_x} \left(\frac{\partial S_x}{\partial x_1^+} n_{x_1^+} \delta n_1^+ + \frac{\partial S_x}{\partial x_1^-} n_{x_1^-} \delta n_1^- + \frac{\partial S_x}{\partial x_3^+} n_{x_3^+} \delta n_3^+ \right. \\
& \left. + \frac{\partial S_x}{\partial x_3^-} n_{x_3^-} \delta n_3^- \right)
\end{aligned} \quad (3.106)$$

Applying a similar procedure, the linearized form of the R_n is written as

$$R_n = f(\rho_1, \rho_2, x_1^+, x_1^-, x_2^+, x_2^-, x_3^+, x_3^-, y_1^+, y_1^-, y_2^+, y_2^-, y_3^+, y_3^-) \quad (3.107)$$

$$\begin{aligned}
\delta R_n = & \frac{\partial R_n}{\partial \rho_1} \delta \rho_1 + \frac{\partial R_n}{\partial \rho_2} \delta \rho_2 \\
& + \frac{\partial R_n}{\partial N_x} \left(\frac{\partial N_x}{\partial x_1^+} n_{x_1^+} \delta n_1^+ + \frac{\partial N_x}{\partial x_1^-} n_{x_1^-} \delta n_1^- + \frac{\partial N_x}{\partial x_2^+} n_{x_2^+} \delta n_2^+ \right. \\
& + \frac{\partial N_x}{\partial x_2^-} n_{x_2^-} \delta n_2^- + \frac{\partial N_x}{\partial x_3^+} n_{x_3^+} \delta n_3^+ + \frac{\partial N_x}{\partial x_3^-} n_{x_3^-} \delta n_3^- \left. \right) \\
& + \frac{\partial R_n}{\partial N_y} \left(\frac{\partial N_y}{\partial y_1^+} n_{y_1^+} \delta n_1^+ + \frac{\partial N_y}{\partial y_1^-} n_{y_1^-} \delta n_1^- + \frac{\partial N_y}{\partial y_2^+} n_{y_2^+} \delta n_2^+ \right. \\
& + \frac{\partial N_y}{\partial y_2^-} n_{y_2^-} \delta n_2^- + \frac{\partial N_y}{\partial y_3^+} n_{y_3^+} \delta n_3^+ + \frac{\partial N_y}{\partial y_3^-} n_{y_3^-} \delta n_3^- \left. \right)
\end{aligned} \quad (3.108)$$

As an example, only Equation (3.106) will be expanded explicitly term by term. Then,

$$\frac{\partial R_s}{\partial \rho_1} = \frac{\partial R_s}{\partial \rho_{av}} \frac{\partial \rho_{av}}{\partial \rho_1} \delta \rho_1 \quad (3.109)$$

$$\frac{\partial R_s}{\partial \rho_2} = \frac{\partial R_s}{\partial \rho_{av}} \frac{\partial \rho_{av}}{\partial \rho_2} \delta \rho_2 \quad (3.110)$$

where,

$$\frac{\partial R_s}{\partial \rho_{av}} = -\Omega^2 r \sin(\phi) S_x \quad (3.111)$$

and

$$\frac{\partial \rho_{av}}{\partial \rho_1} = \frac{\partial \rho_{av}}{\partial \rho_2} = \frac{1}{2} \quad (3.112)$$

The geometric part of the Equation (3.106) is written as

$$\frac{\partial R_s}{\partial S_x} = -\rho_{av} \Omega^2 r \sin(\phi) \quad (3.113)$$

and

$$\frac{\partial S_x}{\partial x_1^-} = -\frac{1}{4}, \quad \frac{\partial S_x}{\partial x_1^+} = -\frac{1}{4}, \quad \frac{\partial S_x}{\partial x_3^+} = \frac{1}{4}, \quad \frac{\partial S_x}{\partial x_3^-} = \frac{1}{4} \quad (3.114)$$

The final linearized term is the δf_4 which is common for both R_n and R_s . Then,

$$\delta f_4 = \frac{\partial f_4}{\partial A_c} \delta A_c + \frac{\partial f_4}{\partial (\vec{S} \times \vec{N})} \delta (\vec{S} \times \vec{N}) \quad (3.115)$$

where

$$\frac{\partial f_4}{\partial A_c} = \frac{1}{|\vec{S} \times \vec{N}|} \quad (3.116a)$$

and

$$\frac{\partial f_4}{\partial (\vec{S} \times \vec{N})} = -\frac{f_4}{|\vec{S} \times \vec{N}|} \quad (3.116b)$$

The term $\delta(\vec{S} \times \vec{N})$ was explained before and the term δA_c will now be expanded. The functional dependency of the A_c is written as

$$A_c = f(B_x^-, B_y^-, B_x^+, B_y^+, A_{1x}, A_{1y}, A_{2x}, A_{2y}) \quad (3.117a)$$

$$B_x^- = f(x_1^-, x_3^-) \quad (3.117b)$$

$$B_x^+ = f(x_1^+, x_3^+) \quad (3.117c)$$

$$B_y^- = f(y_1^-, y_3^-) \quad (3.117d)$$

$$B_y^+ = f(y_1^+, y_3^+) \quad (3.117e)$$

$$A_{1x} = f(x_1^+, x_1^-, x_2^+, x_2^-) \quad (3.117f)$$

$$A_{1y} = f(y_1^+, y_1^-, y_2^+, y_2^-) \quad (3.117g)$$

$$A_{2x} = f(x_2^+, x_2^-, x_3^+, x_3^-) \quad (3.117h)$$

$$A_{2y} = f(y_2^+, y_2^-, y_3^+, y_3^-) \quad (3.117i)$$

then,

$$\begin{aligned}
\delta A_c = & \frac{\partial A_c}{\partial B_x^-} \left(\frac{\partial B_x^-}{\partial x_1^-} n_{x_1}^- \delta n_1^- + \frac{\partial B_x^-}{\partial x_3^-} n_{x_3}^- \delta n_3^- \right) \\
& + \frac{\partial A_c}{\partial B_x^+} \left(\frac{\partial B_x^+}{\partial x_1^+} n_{x_1}^+ \delta n_1^+ + \frac{\partial B_x^+}{\partial x_3^+} n_{x_3}^+ \delta n_3^+ \right) \\
& + \frac{\partial A_c}{\partial B_y^-} \left(\frac{\partial B_y^-}{\partial y_1^-} n_{y_1}^- \delta n_1^- + \frac{\partial B_y^-}{\partial y_3^-} n_{y_3}^- \delta n_3^- \right) \\
& + \frac{\partial A_c}{\partial B_y^+} \left(\frac{\partial B_y^+}{\partial y_1^+} n_{y_1}^+ \delta n_1^+ + \frac{\partial B_y^+}{\partial y_3^+} n_{y_3}^+ \delta n_3^+ \right) \\
& + \frac{\partial A_c}{\partial A_{1x}} \left(\frac{\partial A_{1x}}{\partial x_1^-} n_{x_1}^- \delta n_1^- + \frac{\partial A_{1x}}{\partial x_1^+} n_{x_1}^+ \delta n_1^+ + \frac{\partial A_{1x}}{\partial x_2^-} n_{x_2}^- \delta n_2^- \right. \\
& \left. + \frac{\partial A_{1x}}{\partial x_2^+} n_{x_2}^+ \delta n_2^+ \right) \\
& + \frac{\partial A_c}{\partial A_{2x}} \left(\frac{\partial A_{2x}}{\partial x_2^-} n_{x_2}^- \delta n_2^- + \frac{\partial A_{2x}}{\partial x_2^+} n_{x_2}^+ \delta n_2^+ + \frac{\partial A_{2x}}{\partial x_3^-} n_{x_3}^- \delta n_3^- \right. \\
& \left. + \frac{\partial A_{2x}}{\partial x_3^+} n_{x_3}^+ \delta n_3^+ \right) \\
& + \frac{\partial A_c}{\partial A_{1y}} \left(\frac{\partial A_{1y}}{\partial y_1^-} n_{y_1}^- \delta n_1^- + \frac{\partial A_{1y}}{\partial y_1^+} n_{y_1}^+ \delta n_1^+ + \frac{\partial A_{1y}}{\partial y_2^-} n_{y_2}^- \delta n_2^- \right. \\
& \left. + \frac{\partial A_{1y}}{\partial y_2^+} n_{y_2}^+ \delta n_2^+ \right) \\
& + \frac{\partial A_c}{\partial A_{2y}} \left(\frac{\partial A_{2y}}{\partial y_2^-} n_{y_2}^- \delta n_2^- + \frac{\partial A_{2y}}{\partial y_2^+} n_{y_2}^+ \delta n_2^+ + \frac{\partial A_{2x}}{\partial y_3^-} n_{y_3}^- \delta n_3^- \right. \\
& \left. + \frac{\partial A_{2x}}{\partial y_3^+} n_{y_3}^+ \delta n_3^+ \right)
\end{aligned} \tag{3.118}$$

The above expression completes the linearization of the interior Euler equations. Next, boundary conditions will be linearized.

3.5.2. Linearization of Boundary Conditions

3.5.2.1. Linearization of the Duct Boundary Condition

For the inlet total density condition, the linearization is similar to the interior Euler equation linearization and given in reference [6]. Recall the inlet total density condition as

$$\rho_t = \rho_1 \left(1 + \frac{\gamma - 1}{2} M^2 \right)^{\frac{1}{\gamma - 1}} = \rho_1 \left(1 - \frac{q_1^2}{2h_t} \right)^{-\frac{1}{\gamma - 1}} \quad (3.54)$$

Linearization yields

$$\frac{\rho_t}{\rho_1} \delta \rho_1 + \frac{q_1 \rho_t}{(\gamma - 1)(h_t - \frac{q_1^2}{2})} \delta q_1 = \text{Residual} = (\rho_{t_{\text{specified}}}) - \rho_t \quad (3.119)$$

The linearized form of the inlet condition

$$Y_{1,j+1} - Y_{1,j} = (Y_{1,JJ} - Y_{1,1}) \frac{\dot{m}_j}{\dot{m}_{total}}, \quad 1 \leq j \leq JJ - 1 \quad (3.56)$$

is written as

$$\delta n_{1,j+1} - \delta n_{1,j} = 0, \quad 1 \leq j \leq JJ - 1 \quad (3.120)$$

For unchoked flow, there is no thermodynamic boundary condition for the exit plane. For choking condition, the following equation

$$(\rho_t)_{j+1} - (\rho_t)_j = (\Delta\rho_t)_{specified} \quad (3.59)$$

is linearized as

$$\begin{aligned} & \left(\frac{\rho_t}{\rho_1} \delta\rho_1 + \frac{q_1\rho_t}{(\gamma-1)\left(h_t - \frac{q_1^2}{2}\right)} \delta q_1 \right)_{j+1} \\ & - \left(\frac{\rho_t}{\rho_1} \delta\rho_1 + \frac{q_1\rho_t}{(\gamma-1)\left(h_t - \frac{q_1^2}{2}\right)} \delta q_1 \right)_j \\ & = (\Delta\rho_t)_{specified} - [(\rho_t)_{j+1} - (\rho_t)_j] \end{aligned} \quad (3.121)$$

The geometric exit boundary condition Equation (3.57)

$$(Y_{i,j+1} - Y_{1,j}) - (Y_{1-1,j+1} - Y_{i-1,j}) = 0, \quad 1 \leq j \leq JJ - 1 \quad (3.57)$$

is linearized as

$$(\delta n_{II,j+1} - \delta n_{II,j}) - (\delta n_{II-1,j+1} - \delta n_{II-1,j}) = 0, \quad 1 \leq j \leq JJ - 1 \quad (3.122)$$

Finally linearized form of the solid wall boundary condition is simply

$$\delta n_{i,1} = \delta n_{i,JJ} = 0, \quad 1 < i < II \quad (3.123)$$

3.5.2.2. Linearization of the Blade to Blade Boundary Conditions

The inlet Mach number condition

$$M_{1,j} = M_{1spec,j} , \quad 1 \leq j \leq JJ - 1 \quad (3.61)$$

is linearized as

$$\frac{q_1^2}{\gamma p_1} \delta \rho_1 + \frac{2\rho_1 q_1}{\gamma p_1} \delta q_1 - \frac{\rho_1 q_1^2}{\gamma p_1^2} \delta p_1 = -(M_1^2 - M_{1specified}^2) \quad (3.124)$$

At the inlet, inlet slope must be specified as a geometric boundary condition. The inlet slope condition (3.66)

$$s_{in} = \frac{y_{2,j} - y_{1,j}}{x_{2,j} - x_{1,j}} \quad 1 \leq j \leq JJ \quad (3.66)$$

is linearized as

$$\begin{aligned} \delta y_{2,j} - \delta y_{1,j} - s_{inl}(\delta x_{2,j} - \delta x_{1,j}) - (x_{2,j} - x_{1,j})\delta s_{inl} \\ = s_{inl}(x_{2,j} - x_{1,j}) + y_{1,j} - y_{2,j} \end{aligned} \quad (3.125)$$

At the inlet, there is one more boundary condition in the case of choking condition. Modification for the choking flow at the inlet boundary conditions of cascade domain is the same modification that of duct boundary condition.

The geometric periodicity is given by Equation (3.67)

$$\delta n_{i,1} = \delta n_{i,JJ} \quad 2 < i < ILE \text{ and } ITE + 1 < i < II - 1 \quad (3.67)$$

is linearized to

$$\delta n_{i,1} - \delta n_{i,JJ} = \Delta n_i \quad (3.126)$$

where, Δn_i is the gap between the 1^{st} and JJ^{th} streamline. The pressure periodicity given by the Equation (3.68)

$$\pi_{i,1}^- = \pi_{i,JJ-1}^+ \quad 2 < i < ILE \text{ and } ITE + 1 < i < II - 1 \quad (3.68)$$

is linearized to

$$\delta \pi_{i,1}^- - \delta \pi_{i,JJ-1}^+ = \pi_{i,JJ-1}^+ - \pi_{i,1}^- \quad (3.127)$$

At the outlet, there is no geometric boundary condition since the exit slope is determined by the trailing edge Kutta condition. Linearization of the Kutta condition is carried out in a similar to the linearization of N momentum. That is, the N momentum equation is applied for all vertical stations through computational domain while trailing edge Kutta is applied only for trailing edge. The following equation explains the linearization of the Kutta condition:

$$\begin{aligned}
& [q_1^2 g_1 \delta \rho_1 + 2\rho_1 q_1 g_1 \delta q_1 + \rho_1 q_1^2 \delta g_1 - q_2^2 g_2 \delta \rho_2 \\
& \quad - 2\rho_2 q_2 g_2 \delta q_2 - \rho_2 q_2^2 \delta g_2 - \delta p_1 - \delta p_2 \\
& \quad + (g_3 - 2)\delta P_c + P_c \delta g_3 + g_4 \delta R_n \\
& \quad + R_n \delta g_4]_1 \\
& - [q_1^2 g_1 \delta \rho_1 + 2\rho_1 q_1 g_1 \delta q_1 + \rho_1 q_1^2 \delta g_1 \quad i=ITE \\
& \quad - q_2^2 g_2 \delta \rho_2 - 2\rho_2 q_2 g_2 \delta q_2 - \rho_2 q_2^2 \delta g_2 \\
& \quad + \delta p_1 + \delta p_2 + (g_3 + 2)\delta P_c + P_c \delta g_3 \\
& \quad + g_4 \delta R_n + R_n \delta g_4]_{JJ-1} \\
& = 2\pi_{i,1}^- - 2\pi_{i,JJ-1}^+
\end{aligned} \tag{3.128}$$

For a blunt leading edge, the leading edge Kutta again must be applied. The above formulation will be applied for leading edge ($i=ILE$) only.

3.6. Jacobian Matrix Structure

In the complete system, there are $(II-2) \times (JJ-1)$ S momentum and $(II-2) \times (JJ-2)$ reduced N momentum equations. Indices for both S momentum and reduced N momentum are shown below Figure 3.10.

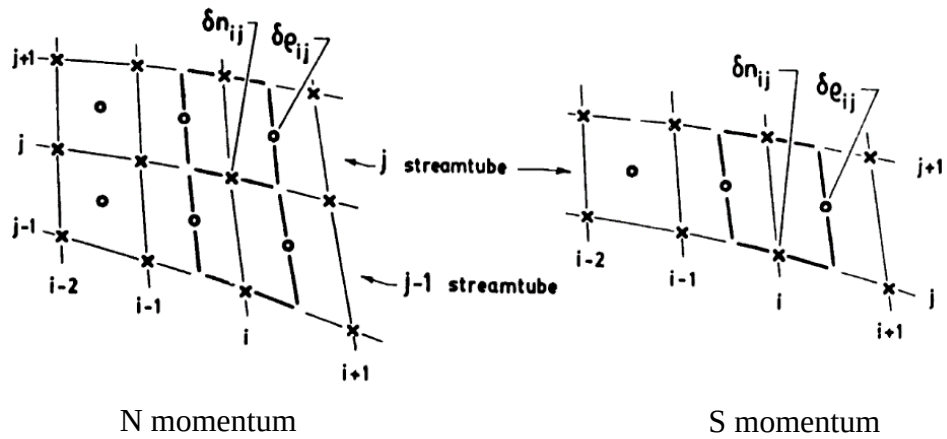


Figure 3.10 Indices for a computational molecule [34]

The term δ denotes the vector of unknowns for each streamwise station including δn and $\delta \rho$ while δg denotes the vector of global unknowns such as δs_{inl} . The structure of the matrices A, B, C, Z, U, G, δ and R will be explained in detail. The matrices A, B, C, and Z have the dimension of $(2JJ-1) \times (2JJ-1)$ and δ and R have the dimensions of $(2JJ-1)$. The remaining matrices are very small matrices when compared to the others since they include variables coming from global equations such as the inlet slope and trailing edge Kutta condition. At the boundary, structure of the matrices changes due to boundary condition implementation. Therefore, these matrices will be examined in three parts: inlet, interior and exit.

$$\left[\begin{array}{cccc|cccccc} A_1 & C_1 & & & & & & & U_1 \\ B_2 & A_2 & C_2 & & & & & & U_2 \\ Z_3 & B_3 & A_3 & C_3 & & & & & U_3 \\ & . & . & . & . & & & & . \\ & & Z_i & B_i & A_i & C_i & & & U_i \\ & & & . & . & . & . & & . \\ & & & & . & . & . & . & . \\ & & & & & Z_{II-1} & B_{II-1} & A_{II-1} & C_{II-1} & U_{II-1} \\ & & & & & & Z_{II} & B_{II} & A_{II} & U_{II} \\ \hline G_1 & G_2 & G_3 & . & . & . & . & . & G_{II} & G_g \end{array} \right] = \left[\begin{array}{c} \delta_1 \\ \delta_2 \\ \delta_3 \\ . \\ \delta_i \\ . \\ . \\ \delta_{II-1} \\ \delta_{II} \\ \delta_g \end{array} \right] = \left[\begin{array}{c} R_1 \\ R_2 \\ R_3 \\ . \\ R_i \\ . \\ . \\ R_{II-1} \\ R_{II} \\ R_g \end{array} \right] \quad (3.129)$$

62

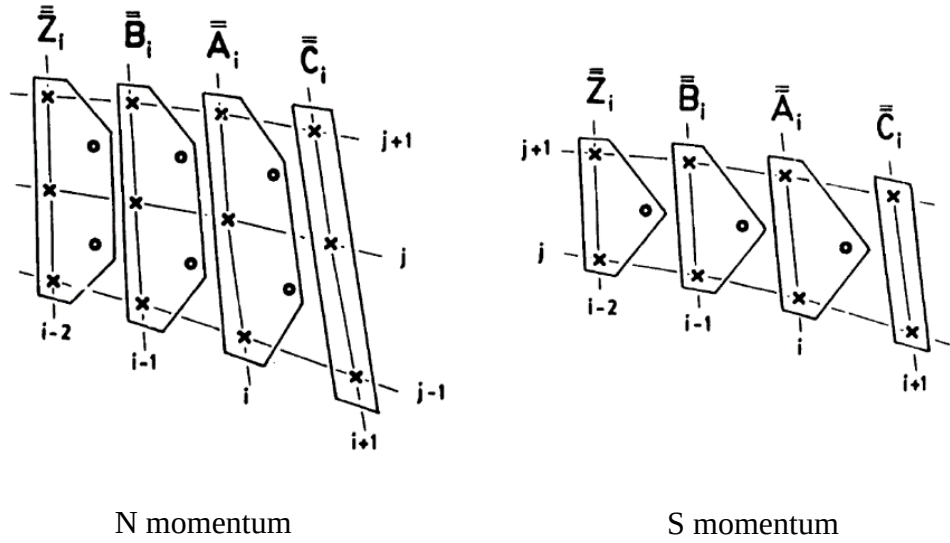


Figure 3.11 Structure for Jacobian matrix implementation [34]

The circles in the above figure shows the Jacobian entries coming from the coefficients of the density change while the cross represents the Jacobian entries coming from the coefficients of node normal displacements. If there is no artificial compressibility, the matrix Z is completely empty because this matrix is used only for artificial compressibility modification.

At the inlet streamwise station ($i=1$), Z and B do not exist. The upper part of A and B at the inlet are filled with the inlet slope equations (3.124). The lower part of the A and B are filled with the coefficients coming from the inlet total density or inlet stagnation density conditions. The crosses (x) and circles (o) represent Jacobian entry for node normal displacement and density change, respectively.

At the interior ($2 < i < II-1$), the structure of the matrices Z , B , A and C are shown Figure 3.13 and 3.14. The first and last rows of the upper halves of these matrices include the boundary conditions of solid wall and periodicity. These parts of the matrices depend on the selection of the boundary condition (i.e. duct or blade to blade). Remaining parts between the first and last rows of upper halves are filled by the

reduced N momentum equations. The lower halves of the matrices are filled by the S momentum equations.

At the exit ($i=II$), the upper parts of the matrices B and A include the outlet flow angle conditions. The lower part of the matrices are dummy because the density at $i=II$ does not exist.

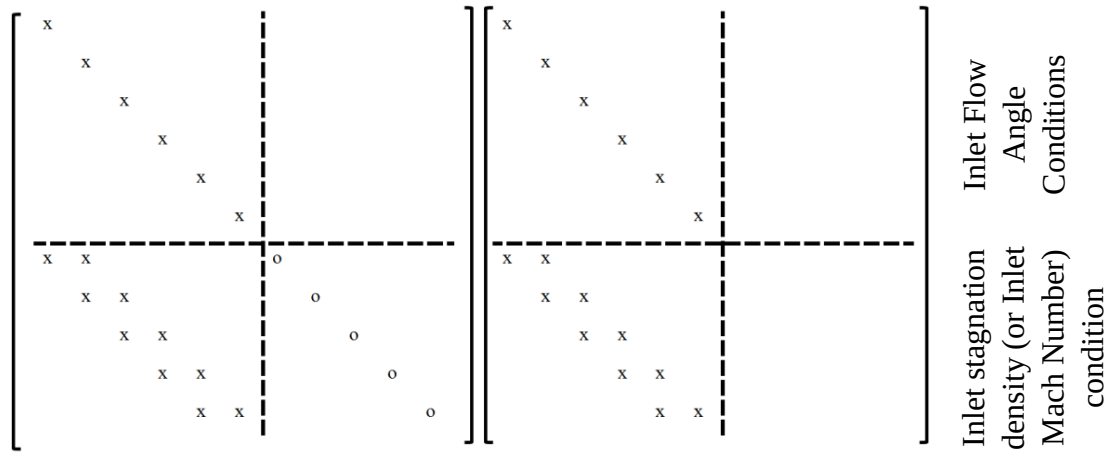


Figure 3.12 Structures of A_1 and C_1

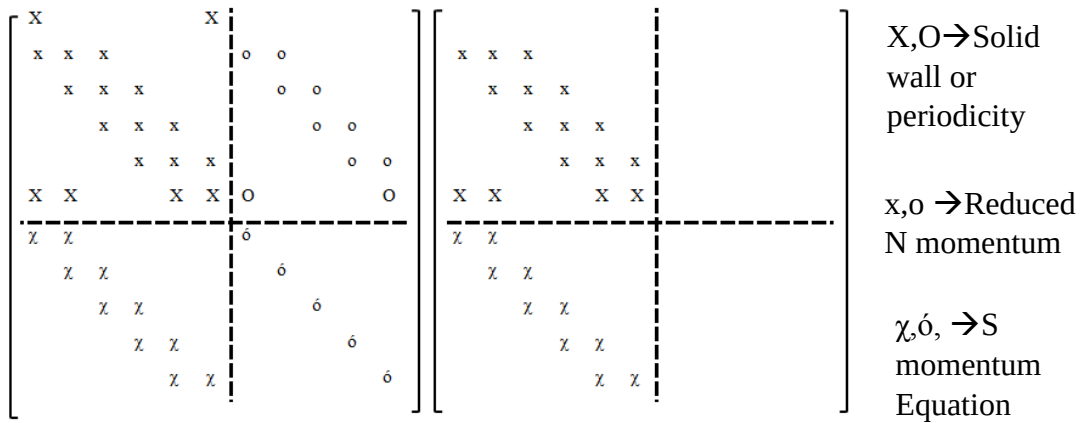


Figure 3.13 Structures of A_i and C_i ($2 \leq i \leq II - 1$)

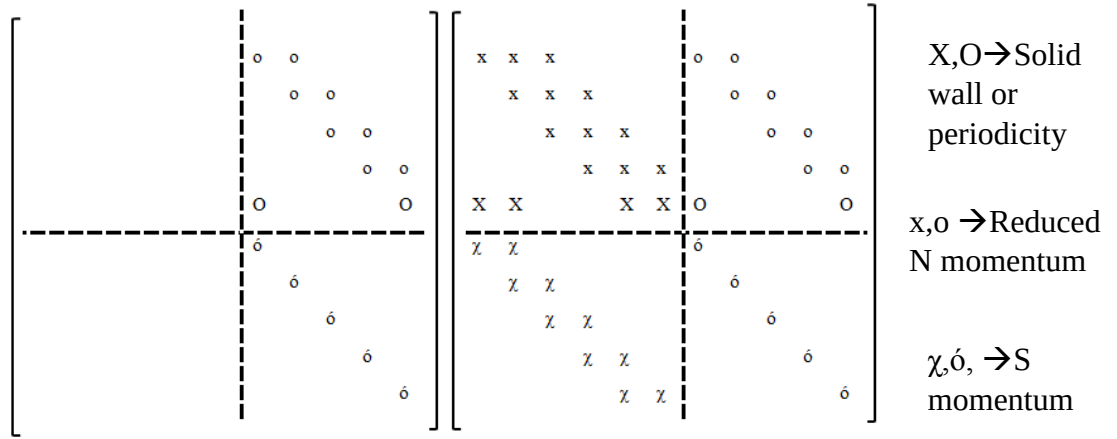


Figure 3.14 Structures of Z_i and B_i structure ($2 \leq i \leq II - 1$)

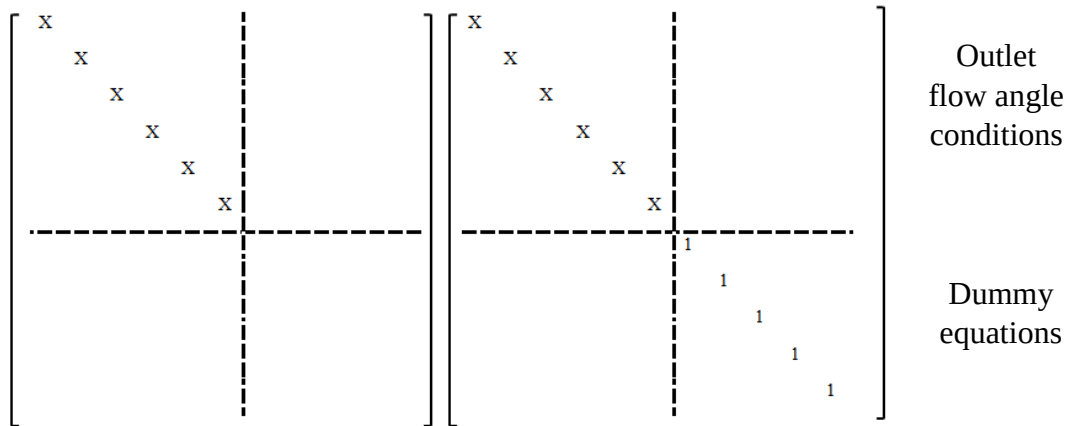


Figure 3.15 Structures of A_I and B_I

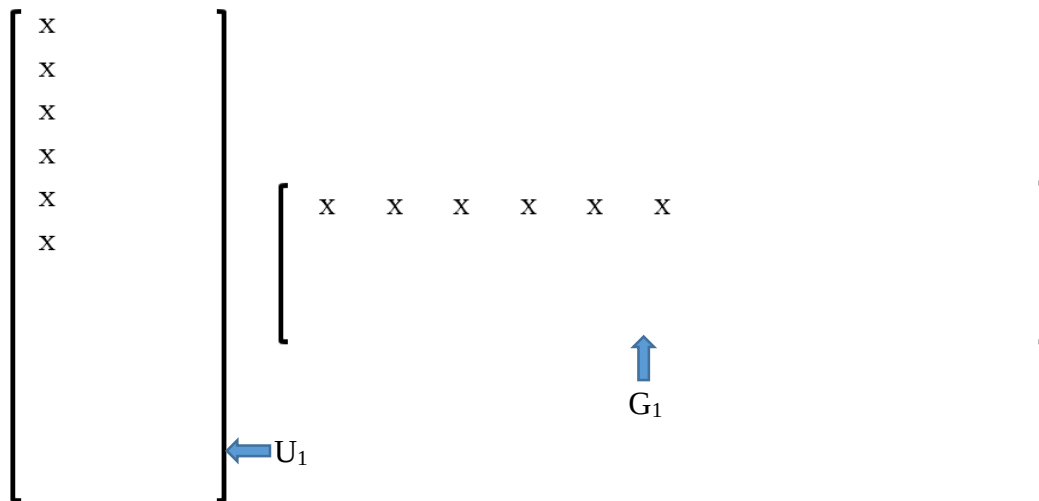


Figure 3.16 U_1 and G_1 matrices

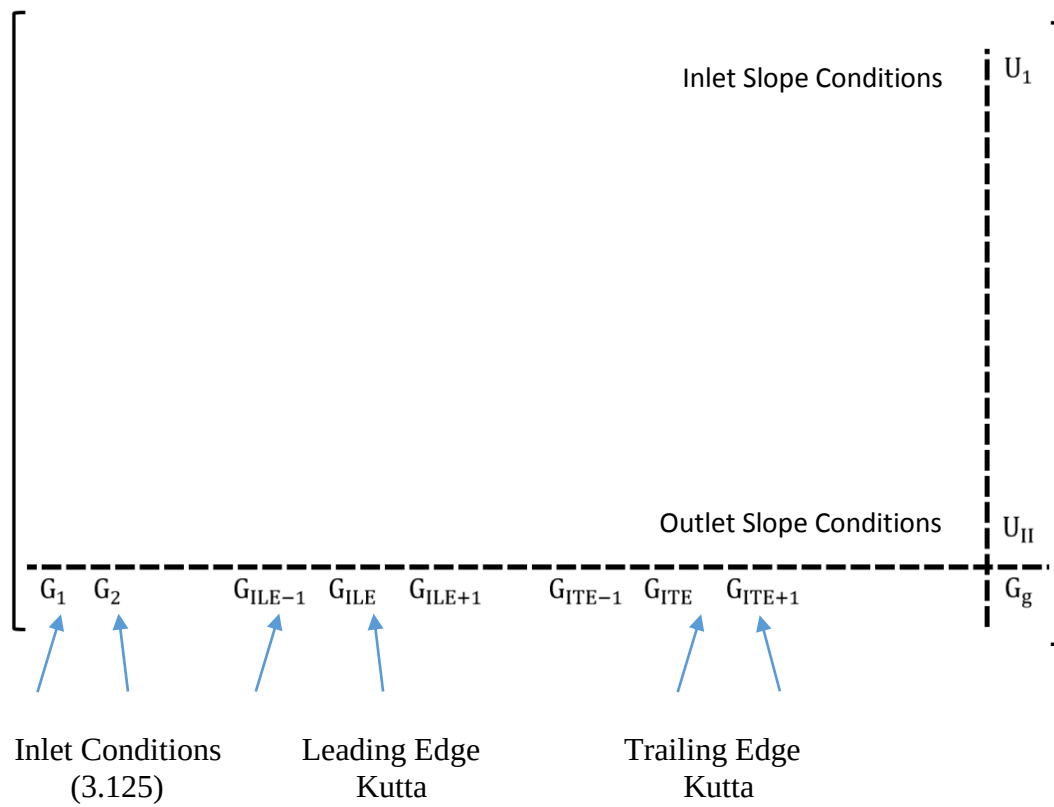


Figure 3.17 Non-zero entries for the U and G matrices

The global equation entries are stored in U and G matrices which are shown in Figures 3.16 and 3.17. U matrices contain entries for the global unknowns. For example, first column of the U_1 matrix is filled with the Jacobian entries for global unknown δ_{Sinlet} (note that these entries do not come from the global equation, they come from inlet boundary condition). The entries for δn and δp , coming from the global equations are stored in G matrices. The G_g matrix includes the entries for global unknown coming from the global equations. The non-zero entries for U and G matrices are shown below and empty spaces are all zeros. In order to clarify the global entries, only the U_1 and G_1 is presented in Figure 3.16.

3.7. Solution of the Newton System

The formed block matrix can be solved by either iterative method or Gauss elimination method. Giles [6] has stated that the iterative method has no superiority on the direct Gauss elimination method. That is why; the direct method is preferred in this thesis.

Equation (3.129) still can be handled by the direct Gauss elimination method but it is not in the most convenient form. Equation (3.129) without the global entries constitutes a tri-diagonal block matrix. Therefore, Equation (3.129) will be divided into two parts: tri diagonal block matrix part and global equation part. Before explanation of the solution procedure, the tri-diagonal matrix algorithm will be presented. Giles [6] has explained the solution of the modified block tri-diagonal algorithm. The procedure is defined as

Forward Sweep

$$\begin{array}{llll}
 i=1 \text{ to } II & B'_i = B_i - Z_i C'_{i-2}, & R'_i = R_i - Z_i R''_{i-2} & \text{Eliminating } Z_i \\
 & A'_i = A_i - B'_i C'_{i-1}, & R''_i = R'_i - B'_i R'''_{i-1} & \text{Eliminating } B_i' \\
 & C'_i = A'^{-1}_i - C_i, & R'''_i = A'^{-1}_i R''_i & \text{Normalize row}
 \end{array}$$

Backward Sweep

$$i=II \text{ to } 1 \quad \delta_i = R'''_i - C'_i D_{i+1}$$

The terms such as C_{-1}, C_0 appearing in this algorithm is not a problem because all of such terms are assumed to be zero.

As mentioned above, Equation (3.129) is solved in two parts

$$\begin{bmatrix} A_1 & C_1 & & & & & & & \\ B_2 & A_2 & C_2 & & & & & & \\ Z_3 & B_3 & A_3 & C_3 & & & & & \\ & \cdot & \cdot & \cdot & \cdot & & & & \\ & & Z_i & B_i & A_i & C_i & & & \\ & & & \cdot & \cdot & \cdot & \cdot & & \\ & & & & \cdot & \cdot & \cdot & \cdot & \\ & & & & & Z_{II-1} & B_{II-1} & A_{II-1} & C_{II-1} \\ & & & & & Z_I & B_{II} & A_{II} & \\ & & & & & & & & \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \cdot \\ \delta_i \\ \cdot \\ \cdot \\ \delta_{II-1} \\ \delta_{II} \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ \cdot \\ R_i \\ \cdot \\ \cdot \\ R_{II-1} \\ R_I \end{bmatrix} - \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ \cdot \\ U_i \\ \cdot \\ \cdot \\ U_{II-1} \\ U_{II} \end{bmatrix} \delta_g \quad (3.130)$$

$$G_1\delta_1 + G_2\delta_2 + G_3\delta_3 + \cdots G_i\delta_i + \cdots G_{II}\delta_{II} + G_g\delta_g = R_g \quad (3.131)$$

Equation (3.130) now constitutes standard tri-diagonal block matrix with multiple right hand side which is solved by standard tri-diagonal algorithm described by Giles [6]. The resulting solution of Equation (3.130) has the following form

$$\delta_i = r_i - u_i\delta_g \quad (3.132)$$

Substitution of Equation (3.132) into Equation (3.131) gives a system which involves only the global unknowns as

$$\begin{aligned} (G_g - G_1u_1 - G_2u_2 - G_3u_3 + \cdots G_iu_i + \cdots G_{II}u_{II})\delta_g \\ = (R_g - G_1r_1 + G_2r_2 + G_3r_3 + \cdots G_ir_i + \cdots G_{II}r_{II}) \end{aligned} \quad (3.133)$$

The above equation includes only the global variables so it is a very small linear system. For classical cascade problem, there are two global variables including the

inlet and outlet slope. These variables are constrained by the specified inlet slope and trailing edge Kutta condition. For cascade problem, Equation (3.133) becomes

$$\begin{bmatrix} x & \dots & x & \dots \\ x & \dots & x & \dots \end{bmatrix}_{2 \times 2} \begin{bmatrix} \delta_{s_{inlet}} \\ \delta_{s_{outlet}} \end{bmatrix}_{2 \times 1} = \begin{bmatrix} x & \dots \\ x & \dots \end{bmatrix}_{2 \times 1} \quad (3.134)$$

3.8. Updating Variables

After changes in the variables are determined, the density and coordinates are updated. The updating is simply given by

$$\begin{aligned} \rho_{i,j}^{n+1} &= \rho_{i,j}^n + RF(\delta\rho_{i,j}) \\ x_{i,j}^{n+1} &= x_{i,j}^n + RF \hat{n}_{x_{i,j}} \delta n_{i,j} \\ y_{i,j}^{n+1} &= y_{i,j}^n + RF \hat{n}_{y_{i,j}} \delta n_{i,j} \end{aligned} \quad (3.135)$$

The relaxation factor can be selected as “1” for the standard Newton iteration procedure. However, the value of r must be chosen so that the density never changes by more than a factor of 2.

CHAPTER 4

TIME MARCHING EULER SOLVER

In this thesis, another solution method developed for the blade to blade flow is the time marching method. The governing unsteady fluid flow equations are hyperbolic in time, so flow variables are marched in time starting from known initial conditions. As discussed in the previous chapter, solution of the steady flow equations for transonic flow requires some additional procedures like artificial compressibility. However, since the unsteady Euler equations are hyperbolic in time for all type of flow regime, it is applicable to transonic flow without any restrictions.

The dynamic behavior of a fluid element is determined by the conservation of mass, momentum and energy. Since the aim is to develop an inviscid solver in this thesis, the viscous terms and heat conduction terms are set to zero. Therefore, typical control volume for the Euler equations is shown in figure 4.1.

The general integral form of the Euler equations [36] is written as

$$\frac{D}{dt} \int \rho \eta d\Omega = \int \frac{\partial}{\partial t} (\rho \eta) d\Omega + \int \rho \eta (\vec{V} \cdot \hat{n}) dS = \int Q_v d\Omega + \int (\vec{Q}_s \cdot \hat{n}) dS \quad (4.1)$$

where η is any property (vector and scalar) and right hand side of the equation is the source terms.

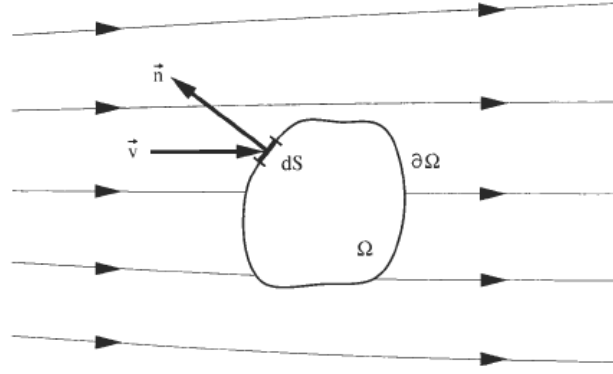


Figure 4.1 Definition of the finite control volume [12]

By setting the η equal to “1” and setting all source terms to “0”, conservation of mass equation is obtained and it is written as

$$\int \frac{\partial}{\partial t} \rho d\Omega + \int \rho (\vec{V} \cdot \hat{n}) dS = 0 \quad (4.2)$$

Setting η equal to $\vec{V}$, momentum equation is obtained and it is written as

$$\int \frac{\partial}{\partial t} \rho \vec{V} d\Omega + \int \rho \vec{V} (\vec{V} \cdot \hat{n}) dS = \int \rho \vec{f}_e d\Omega - \int p \vec{n} dS \quad (4.3)$$

where Coriolis and centrifugal forces are the volumetric body forces while pressure forces are surface body forces. These terms appear in the right hand side of Equation (4.3). Finally, setting the η equal to E , which is total energy per unit mass, the energy equation is obtained as

$$\int \frac{\partial}{\partial t} \rho E d\Omega + \int \rho E (\vec{V} \cdot \hat{n}) dS = \int \rho \vec{f}_e \cdot \vec{V} d\Omega - \int p (\vec{V} \cdot \vec{n}) dS \quad (4.4)$$

The work done by the external body forces and work done by the surface pressure appears on the right hand side of the equation.

4.1. Discretization

In this thesis, cell centered finite volume approach is used where the flow quantities are stored at the centroids of the grid cell so the control volumes are identical to the grid cells. The typical control volume is shown in Figure 4.1.

The vector form of the Euler equations including Equations (4.2), (4.3) and (4.4) can be written as

$$\frac{\partial}{\partial t} \int \vec{W} d\Omega + \oint \vec{F}_c dS = \int \vec{Q} d\Omega \quad (4.5)$$

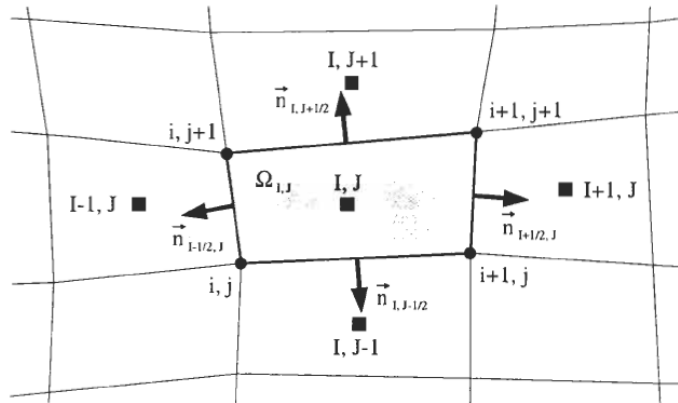


Figure 4.2 Typical cell centered control volume [12]

The conservative variable vector in two dimensions consists of the following four components:

$$\vec{W} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix} \quad (4.6)$$

where

$$E = \frac{1}{\gamma - 1} \frac{P}{\rho} + \frac{1}{2} (u^2 + v^2) - \frac{1}{2} (\omega r)^2 \quad (4.7)$$

Convective fluxes can be written as

$$\vec{F}_c = \begin{bmatrix} \rho(un_x + vn_y) \\ \rho u(un_x + vn_y) + Pn_x \\ \rho v(un_x + vn_y) + Pn_y \\ \rho I(un_x + vn_y) \end{bmatrix} \quad (4.8)$$

In Equation (4.8), “ I ” represents rothalpy and can be written as

$$I = \frac{\gamma}{\gamma - 1} \frac{P}{\rho} + \frac{1}{2} (u^2 + v^2) - \frac{1}{2} (\omega r)^2 \quad (4.9)$$

The right most term in Equation (4.9) is the modification due to the angular speed of the rotor. In the case of stator, the term becomes automatically zero and I turns out to be total enthalpy.

Finally, the source terms can be written as

$$\vec{Q} = \begin{bmatrix} 0 \\ 0 \\ \rho(W^2 y + 2Ww) \\ q_h \end{bmatrix} \quad (4.10)$$

where the q_h is the heat addition to the system. The other terms will be explained in the next section.

In addition to the above definitions, the vector of primitive variables can be given as

$$\vec{U} = \begin{bmatrix} \rho \\ u \\ v \\ E \end{bmatrix} \quad (4.11)$$

The conversion between conservative and primitive variables is necessary in the computer program. Therefore, the conversion from conservative variables to the primitive variables referring to Equation (4.6), (4.10) and (4.11) can be written as

$$\vec{U} = \begin{bmatrix} W_1 \\ \frac{W_2}{W_1} \\ \frac{W_3}{W_1} \\ (\gamma - 1) \left[W_4 + \frac{1}{2} W_1 (\omega r)^2 - \frac{1}{2} W_1 \left(\left(\frac{W_2}{W_1} \right)^2 + \left(\frac{W_3}{W_1} \right)^2 \right) \right] \end{bmatrix} \quad (4.12)$$

The conversion from primitive to conservative variables referring to Equations (4.6), (4.10) and (4.11) is written as

$$\vec{W} = \begin{bmatrix} U_1 \\ U_1 U_2 \\ U_1 U_2 \\ \left(\frac{1}{\gamma - 1} \right) U_4 + \frac{1}{2} U_1 (U_2^2 + U_3^2) \end{bmatrix} \quad (4.13)$$

For a particular stationary control volume, the time derivative of the conservative variables can be written as

$$\frac{\partial}{\partial t} \int \vec{W} d\Omega = \Omega \frac{\partial \vec{W}}{\partial t} \quad (4.14)$$

Equation (4.5) can be rewritten as using above definition as

$$\frac{\partial \vec{W}}{\partial t} = -\frac{1}{\Omega} \left[\oint \vec{F}_c dS - \int \vec{Q} d\Omega \right] \quad (4.15)$$

Since the control volume is definite, Equation (4.15) can be written as

$$\frac{\partial \vec{W}_{i,j}}{\partial t} = -\frac{1}{\Omega_{i,j}} \left[\sum_{m=1}^{\#Face} \vec{F}_c \Delta S_m - (\vec{Q}\Omega)_{i,j} \right] \quad (4.16)$$

The terms inside the square brackets on right hand side of Equation (4.16) is called as residual. Then, the Equation (4.16) is rewritten as

$$\frac{\partial \vec{W}_{i,j}}{\partial t} = -\frac{1}{\Omega_{i,j}} R_{i,j} \quad (4.17)$$

Equation (4.17) is written for all control volumes and a system of ordinary differential equation is obtained.

It is necessary to evaluate the geometric parameters in order to calculate residuals corresponding to each computational cell. Figure 4.2 shows the typical volume, surface and unit normal definitions

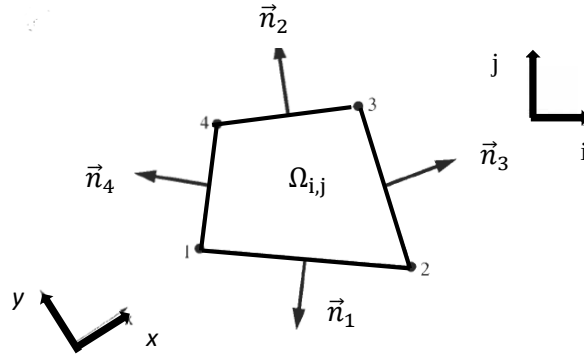


Figure 4.3 Control volume, control surfaces and unit normal vectors [12]

Referring to Figure 4.2, the volume of the computational cell per unit width is evaluated as

$$\Omega_{i,j} = [(x_1 - x_3)(y_2 - y_4) + (x_4 - x_2)(y_1 - y_3)] \quad (4.18)$$

Referring to Figure 4.2, the surface vector and normal of the faces are evaluated as

$$\vec{S}_m = \begin{bmatrix} S_{x_m} \\ S_{y_m} \end{bmatrix} = \vec{n}_m \Delta S_m \quad (4.19)$$

$$\vec{S}_1 = \begin{bmatrix} y_2 - y_1 \\ x_1 - x_2 \end{bmatrix} \quad (4.20)$$

$$\vec{S}_2 = \begin{bmatrix} y_3 - y_2 \\ x_2 - x_3 \end{bmatrix} \quad (4.21)$$

$$\vec{S}_3 = \begin{bmatrix} y_4 - y_3 \\ x_3 - x_4 \end{bmatrix} \quad (4.22)$$

$$\vec{S}_4 = \begin{bmatrix} y_1 - y_4 \\ x_4 - x_1 \end{bmatrix} \quad (4.23)$$

The unit normal vectors are evaluated as

$$\vec{n}_m = \frac{\vec{S}_m}{\Delta S_m} \quad (4.24)$$

where

$$\Delta S_m = \sqrt{S_x^2 + S_y^2} \quad (4.25)$$

4.2. Convective Flux Calculation

Basically, in the literature there are two types of methods for the calculation of the fluxes at the cell faces including central schemes with artificial dissipation and characteristic based schemes. Since the central schemes have some instability problems, they are not preferred in this thesis. Instead, characteristic based approaches including Roe approximate Riemann solver and rotated RHLL (Roe, Harten, Lax, Leer) are used for convective flux calculation.

According to references [37], [38], [39] and [40] the numerical experiments has shown that HLL (Harten, Lax, Leer) Riemann solvers are carbuncle free but may sometimes show unacceptable solutions due to high numerical diffusion. Some other schemes such as AUSM, HLLE and Roe approximate Riemann solver are accurate, less dissipative but they generate carbuncle phenomenon. The RHLL is the combination of the Roe and HLL schemes. According to the reference [38], the accuracy of the schemes are not improved impressively but robustness is increased significantly.

In this thesis, both Roe approximate Riemann solver and rotated RHLL are implemented to the computer program. In this chapter, the Roe flux calculation will be examined in detail but modification for rotated RHLL will not be given. The details are available in reference [37], [38] and [41] for rotated RHLL.

Roe's method which is first introduced by the Godunov is based on the decomposition of the flux difference over a face of control volume considering the directions of characteristic waves. Referring to the figure 4.2

$$(\vec{F}_c)_{face} = \frac{1}{2} [\vec{F}_c(\vec{W}_R) + \vec{F}_c(\vec{W}_L) - |\vec{A}_{Roe}|_{face} (\vec{W}_R - \vec{W}_L)] \quad (4.26)$$

where R and L represents the right and left states of the face. The product of $|\vec{A}_{Roe}|$ and the difference of the left and right states [12] can be evaluated as

$$|\vec{A}_{Roe}|_{face} (\vec{W}_R - \vec{W}_L) = |\Delta\vec{F}_1| + |\Delta\vec{F}_2| + |\Delta\vec{F}_3| \quad (4.27)$$

where

$$|\Delta\vec{F}_1| = |\tilde{V} - \tilde{c}| \left(\frac{\Delta P - \tilde{\rho}\tilde{c}\Delta V}{2\tilde{c}^2} \right) \begin{bmatrix} 1 \\ \tilde{u} - \tilde{c}n_x \\ \tilde{v} - \tilde{c}n_y \\ \tilde{H} - \tilde{c}\tilde{V} \end{bmatrix} \quad (4.28)$$

$$|\Delta\vec{F}_2| = |\tilde{V}| \left\{ \left(\Delta\rho - \frac{\Delta P}{\tilde{c}^2} \right) \begin{bmatrix} 1 \\ \tilde{u} \\ \tilde{v} \\ \frac{\tilde{q}^2}{2} \end{bmatrix} + \tilde{\rho} \begin{bmatrix} 0 \\ \Delta u - \Delta V n_x \\ \Delta v - \Delta V n_y \\ \tilde{u}\Delta u + \tilde{v}\Delta v - \tilde{V}\Delta V \end{bmatrix} \right\} \quad (4.29)$$

$$|\Delta \vec{F}_3| = |\tilde{V} + \tilde{c}| \left(\frac{\Delta P + \tilde{\rho} \tilde{c} \Delta V}{2 \tilde{c}^2} \right) \begin{bmatrix} 1 \\ \tilde{u} + \tilde{c} n_x \\ \tilde{v} + \tilde{c} n_y \\ \tilde{H} + \tilde{c} \tilde{V} \end{bmatrix} \quad (4.30)$$

In Equations (4.27), (4.28), (4.29) and (4.30), while "Δ" represents the difference between right and left states, the symbol “~” represents the Roe averages. The Roe averages can be calculated as

$$\tilde{\rho} = \sqrt{\rho_L \rho_R} \quad (4.31)$$

$$\tilde{u} = \frac{u_L \sqrt{\rho_L} + u_R \sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \quad (4.32)$$

$$\tilde{v} = \frac{v_L \sqrt{\rho_L} + v_R \sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \quad (4.33)$$

$$\tilde{H} = \frac{H_L \sqrt{\rho_L} + H_R \sqrt{\rho_R}}{\sqrt{\rho_L} + \sqrt{\rho_R}} \quad (4.34)$$

$$\tilde{c} = \sqrt{(\gamma - 1) \left(\tilde{H} - \frac{\tilde{q}^2}{2} \right)} \quad (4.35)$$

$$\tilde{V} = \tilde{u} n_x + \tilde{v} n_y \quad (4.36)$$

$$\tilde{q}^2 = \tilde{u}^2 + \tilde{v}^2 \quad (4.37)$$

Normally, the Roe's approximate Riemann solver produces unphysical expansion shock in the case of stationary expansion. In order to deal with this problem, Harten's entropy correction is applied. The detailed explanation about this modification is available in reference [12].

Another flux calculation method used in this thesis is rotated-RHLL solver (Roe, Harten, Lax, Leer). The normal of the face is divided into two orthogonal normal directions as shown in Figure 4.3. Then, along each orthogonal normal direction, different flux calculation techniques are used. In the direction of the n_1 , the HLL and in the direction of the n_2 , Roe flux functions are used. It is easy to implement rotated RHLL onto an existing Roe flux function. The details about the implementation of the method is available in reference [37], [38] and [41].

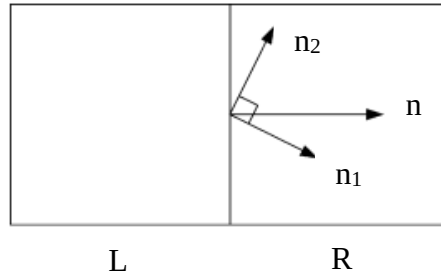


Figure 4.4 Decomposition of face normal [38]

The flow properties at the right and left states are evaluated using non-symmetric formulas. The interpolation formulas are mostly based on Van Leer's MUSCL (Monotone Upstream Centered Schemes for Conservation laws) approach [12].

In the MUSCL scheme, in order to prevent high oscillations in the large gradient region (such as shock waves), limiters are used. As shown in Figure 4.4, it is shown that without limiter function there are large oscillations in the case of shock wave without limiter function.

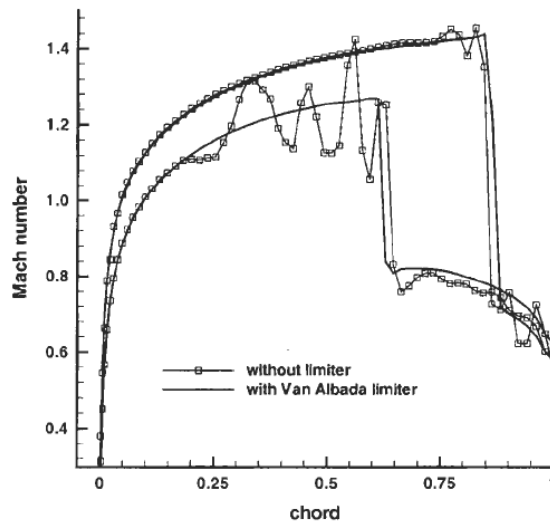


Figure 4.5 Solution of transonic flow over NACA 0012 blade profile with and without limiter [12]

Therefore, the implementation of a limiter while calculating right and left states is strongly needed for transonic flow calculations. However, the implementation of the limiters generally stalls the convergence and residuals cannot descend enough to reach steady state solution. In order to deal with this problem, modified Van Albada limiter [44] is used for calculation of the right and left states. The computational stencil for the right and left states is given in Figure 4.5.

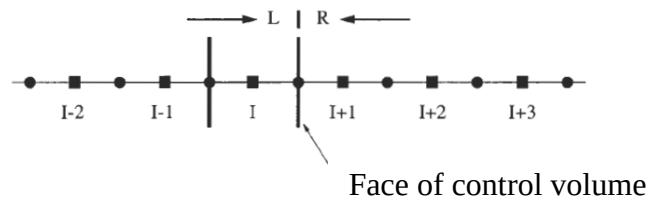


Figure 4.6 Computational stencil for left and right state calculations

Referring to Figure 4.5, the right and left states can be calculated as

$$U_R = U_{I+1} - \frac{1}{2}\delta_R \quad (4.38)$$

$$U_L = U_I + \frac{1}{2}\delta_L \quad (4.39)$$

where

$$\delta = \frac{a(b^2 + \epsilon) + b(a^2 + \epsilon)}{a^2 + b^2 + 2\epsilon} \quad (4.40)$$

The coefficient a and b can be evaluated for left and right states as

$$a_R = U_{I+2} - U_{I+1} \quad \text{and} \quad a_L = U_{I+1} - U_I \quad (4.41)$$

$$b_R = U_{I+1} - U_I \quad \text{and} \quad b_L = U_I - U_{I-1}$$

Selecting $\epsilon = 0$ represents the original MUSCL scheme with Van Albada limiter. In the reference [43], after several computational experiments are carried out, in order to reach desired convergence rate the ϵ is calculated as

$$\epsilon = \sqrt{(Local\ Grid\ Size)^3} \quad (4.42)$$

This characteristic length for the structured grid is selected as the length of the corresponding computational cell face.

4.3. Calculation of the Source Terms

The calculation of the source term is quite simple as compared to that of convective flux. The source term is constant inside the control volume. Therefore, the source term in Equation (4.16) is evaluated as

$$(\vec{Q}\Omega)_{i,j} = \vec{Q}\Omega_{i,j} \quad (4.43)$$

In other words, the source terms are calculated in each control volume and multiplied by cell volume. The resultant term is summed with convective fluxes and finally, residuals in Equation (4.17) are determined.

The definition of the source terms is similar to that of Newton solver. In the previous chapter, Equations (3.8) and (3.9) are the components of the Coriolis and centrifugal forces projected onto the m-φ plane. The effects of the Coriolis and centrifugal forces on m-φ plane can be written as

$$\vec{Q}_{i,j} = \rho_{i,j} \begin{bmatrix} 0 \\ 0 \\ (W^2 y_{mid} + 2Ww) \\ 0 \end{bmatrix} \quad (4.44)$$

where,

$$y_{mid} = \frac{y_{top_{inlet}} + y_{bottom_{inlet}}}{2} \quad (4.45)$$

and w is the velocity component in z direction which normally does not exist in the calculation domain. The value of w at the inlet is used for the source term calculation within the passage. The term W is the rotational speed of the rotor. The value of y in Equation (4.44) is taken as midline value of y axis at the inlet.

4.4. Boundary Conditions

The boundary conditions of the time marching solver are more complicated than that of Newton's solver since characteristic boundary conditions are required at the inlet and exit of the domain. The boundary conditions for the duct and blade to blade solvers can be classified as inviscid wall boundary, inlet/exit boundary and periodic boundary.

4.4.1. Solid Wall Boundary Condition

Since the solver is an inviscid solver, the slip condition must be given to the wall boundaries. Since there is no friction, the velocity vector on the wall must be tangent to the wall surface, i.e.,

$$\vec{V} \cdot \hat{n} = 0 \quad (4.46)$$

Then, the convective flux vector on the wall can be written as

$$\vec{F}_{cw} = \begin{bmatrix} 0 \\ Pn_x \\ Pn_y \\ 0 \end{bmatrix}_w \quad (4.47)$$

Only the wall pressure need to be determined. There are different approaches for the calculation of the wall pressure in the literature but basically they can be grouped in two: solution of normal and tangential momentum equation on the wall and extrapolation method [11]. Extrapolation method is quite simple and it can be found in references [11], [12]. The solution of the momentum equation will be presented here. Hirsh [11] suggests a method including the reflected cell concept but it is not applied due to complexity and necessity of the calculation of the radius of the curvature. Instead, the method which is also available in references [24], [25], [26], is used for the wall pressure calculation.

The method mentioned above requires the grid transformation from physical plane to computational plane as shown in Figure 4.6. This transformation guarantees the flow tangency because after transformation the velocity corresponding to η direction (V_η) is set to zero. Before the transformation of the primitive variables, some definitions will be presented. Complete transformation procedure is presented in references [24], [25] and [26].

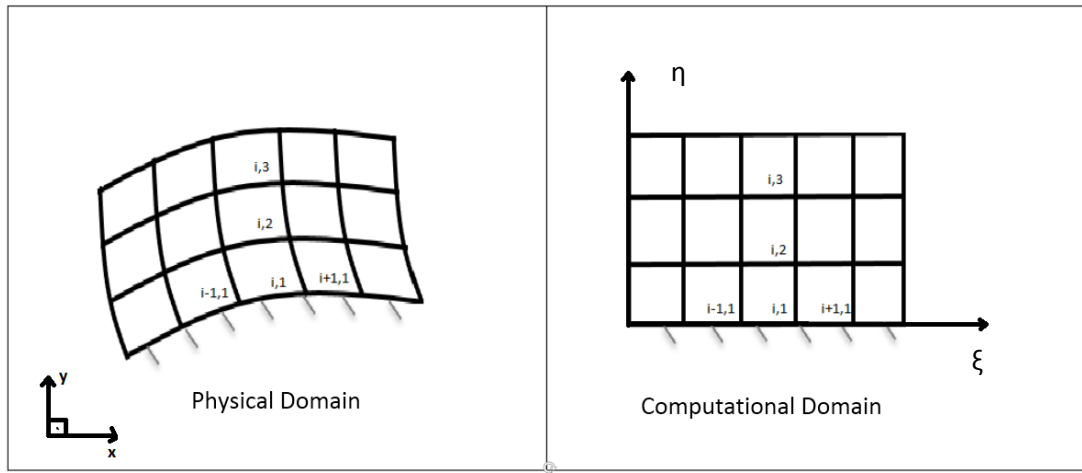


Figure 4.7 Physical and Computational Spaces

The functional dependencies of the coordinates in physical and computational space can be written as

$$x = x(\xi, \eta) \quad (4.48a)$$

$$y = y(\xi, \eta) \quad (4.48b)$$

After making some manipulation, the following equality and jacobian of the transformation is written:

$$\begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{bmatrix} = \frac{1}{\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi}} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial x}{\partial \eta} \\ -\frac{\partial y}{\partial \xi} & \frac{\partial x}{\partial \xi} \end{bmatrix} \quad (4.49)$$

$$J = \frac{1}{\frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta} \frac{\partial y}{\partial \xi}}$$

Here, Jacobian is the ratio of areas in the physical space to that of computational space.

Then, Equation (4.49) can be rewritten as

$$\frac{\partial \xi}{\partial x} = J \frac{\partial y}{\partial \eta} \quad (4.49a)$$

$$\frac{\partial \xi}{\partial y} = -J \frac{\partial x}{\partial \eta} \quad (4.49b)$$

$$\frac{\partial \eta}{\partial x} = -J \frac{\partial y}{\partial \xi} \quad (4.49c)$$

$$\frac{\partial \eta}{\partial y} = J \frac{\partial x}{\partial \xi} \quad (4.49d)$$

The derivatives of the computational space coordinates with respect to Cartesian coordinates are called metrics. The terms $\frac{\partial x}{\partial \xi}$, $\frac{\partial x}{\partial \eta}$, $\frac{\partial y}{\partial \xi}$ and $\frac{\partial y}{\partial \eta}$ defined above are necessary for both evaluation of Jacobian and metrics. Then, referring to Figure 4.7, the derivatives of the Cartesian coordinates with respect to computational space coordinates ($\frac{\partial x}{\partial \xi}$, $\frac{\partial x}{\partial \eta}$, $\frac{\partial y}{\partial \xi}$ and $\frac{\partial y}{\partial \eta}$) can be approximated as

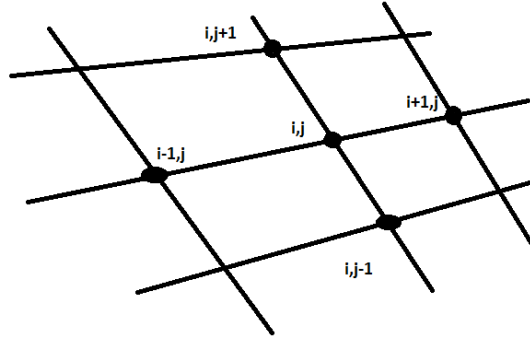


Figure 4.8 Stencil for the calculation of metrics

$$\left(\frac{\partial x}{\partial \xi}\right)_{i,j} = \frac{1}{2}(x_{i+1,j} - x_{i-1,j}) \quad (4.50a)$$

$$\left(\frac{\partial y}{\partial \xi}\right)_{i,j} = \frac{1}{2}(y_{i+1,j} - y_{i-1,j}) \quad (4.50b)$$

$$\left(\frac{\partial x}{\partial \eta}\right)_{i,j} = \frac{1}{2}(x_{i,j+1} - x_{i,j-1}) \quad (4.50c)$$

$$\left(\frac{\partial y}{\partial \eta}\right)_{i,j} = \frac{1}{2}(y_{i,j+1} - y_{i,j-1}) \quad (4.50d)$$

At the boundary nodes, this second order approximation to the derivatives cannot be obtained as defined above. That is, for $i=1$ and $j=1$ negative indices appear in Equation (4.50) but this may be overcome by the parabolic fit to the boundary nodes [45]. That is, assuming equal spacing in computational space, the derivative at the boundary can be taken as

$$\left(\frac{\partial x}{\partial \xi}\right)_{1,j} = \frac{-x_{3,j} + 4x_{2,j} - 3x_{1,j}}{2} \quad (4.51)$$

The similar approaches can be used at other boundaries using three of the interior boundary nodes.

Then the metrics can be calculated using Equation (4.49) as

$$\left(\frac{\partial \xi}{\partial x}\right)_{i,j} = J \frac{\partial y}{\partial \eta} \quad (4.52a)$$

$$\left(\frac{\partial \xi}{\partial y}\right)_{i,j} = -J \frac{\partial x}{\partial \eta} \quad (4.52b)$$

$$\left(\frac{\partial \eta}{\partial x}\right)_{i,j} = -J \frac{\partial y}{\partial \xi} \quad (4.52c)$$

$$\left(\frac{\partial \eta}{\partial y}\right)_{i,j} = J \frac{\partial x}{\partial \xi} \quad (4.52d)$$

The whole calculation for wall pressure determination will be made on the computational space. Therefore, density and velocity components must be mapped onto the computational space. The density is obtained on the wall as follows

$$R_{i,1} = 2R_{i,2} - R_{i,3} \quad (4.53)$$

where

$$R_{i,j} = \frac{\rho_{i,j}}{J_{i,j}} \quad (4.54)$$

The velocity components can be estimated in a similar manner as

$$V_{\zeta_{i,1}} = 2V_{\zeta_{i,2}} - V_{\zeta_{i,3}} \quad (4.55a)$$

$$V_{\eta_{i,1}} = 2V_{\eta_{i,2}} - V_{\eta_{i,3}} \quad (4.55b)$$

where

$$V_{\zeta_{i,j}} = \frac{|\xi_{x_{i,j}}(\rho u)_{i,j} + \xi_{y_{i,j}}(\rho v)_{i,j}|}{\rho_{i,j}}, \quad (4.56a)$$

$$V_{\eta_{i,j}} = \frac{|\eta_{x_{i,j}}(\rho v)_{i,j} + \eta_{y_{i,j}}(\rho v)_{i,j}|}{\rho_{i,j}} \quad (4.56b)$$

The V contravariant velocity on the wall ($V_{i,1}$) is equated to zero because of the tangency condition on the wall. After some manipulation, the physical velocities on the wall can be found as

$$u_{i,1} = \frac{(\eta_y)_{i,1}}{J_{i,1}} V_{\zeta_{i,1}}, \quad (4.57a)$$

$$v_{i,1} = -\frac{(\eta_x)_{i,1}}{J_{i,1}} V_{\zeta_{i,1}} \quad (4.57b)$$

Then finally, the pressure on the wall can be calculated using the procedure defined in reference [24]. The resulting tri-diagonal system can be written as

$$a_i P_{i-1,1} + b_i P_{i,1} + c_i P_{i+1,1} = d_i \quad (4.58)$$

where

$$a_i = -\frac{1}{2} \left[(\eta_x)_{i,1} \left(\frac{\xi_x}{J} \right)_{i-1,1} + (\eta_y)_{i,1} \left(\frac{\xi_y}{J} \right)_{i-1,1} \right] \quad (4.59a)$$

$$b_i = -\frac{3}{2} \left(\frac{\eta_x^2 + \eta_y^2}{J} \right)_{i-1,1} \quad (4.59b)$$

$$c_i = \frac{1}{2} \left[(\eta_x)_{i,1} \left(\frac{\xi_x}{J} \right)_{i+1,1} + (\eta_y)_{i,1} \left(\frac{\xi_y}{J} \right)_{i+1,1} \right] \quad (4.59c)$$

$$\begin{aligned} d_i = & \frac{P_{i,3}}{2} \left[(\eta_x)_{i,1} \left(\frac{\eta_x}{J} \right)_{i,3} + (\eta_y)_{i,1} \left(\frac{\eta_y}{J} \right)_{i,3} \right] \\ & - 2P_{i,2} \left[(\eta_x)_{i,1} \left(\frac{\eta_x}{J} \right)_{i,2} + (\eta_y)_{i,1} \left(\frac{\eta_y}{J} \right)_{i,2} \right] \\ & - 2 \left(\frac{\rho V_\eta}{J} \right)_{i,2} \left[\eta_{x,i,1} u_{i,2} + \eta_{y,i,1} v_{i,2} \right] \\ & + \frac{1}{2} \left(\frac{\rho V_\eta}{J} \right)_{i,3} \left[\eta_{x,i,1} u_{i,3} + \eta_{y,i,1} v_{i,3} \right] \\ & + \frac{1}{2} \left(\frac{\rho V_\zeta}{J} \right)_{i-1,1} \left[\eta_{x,i,1} u_{i-1,1} + \eta_{y,i,1} v_{i-1,1} \right] \\ & - \frac{1}{2} \left(\frac{\rho V_\zeta}{J} \right)_{i+1,1} \left[\eta_{x,i,1} u_{i+1,1} + \eta_{y,i,1} v_{i+1,1} \right] \end{aligned} \quad (4.59d)$$

The constants in Equation (4.58) is found by Equation (4.59) and it is solved by algorithm defined in references [10].

After the pressure determined, the convective flux vector on the wall, which is stated in Equation (4.47), can be calculated.

4.4.2. Inlet and Outlet Boundary Conditions

At inlet and outlet, the characteristic boundary conditions are applied. The number of physical and numerical boundary conditions at the inlet and exit are shown in Table 4.1. The physical boundary condition corresponds to free stream properties at the boundaries while numerical boundary condition corresponds to conditions coming from the interior of the domain.

Normally distance of the domain from inlet and exit must be long enough in order to prevent solution from the effects of reflection from the boundaries. When this distance is guaranteed, far field boundary condition can be applied at the inlet and outlet. However, this means additional computational cost. For domains with short distance between the body and inlet or outlet plane, the non-reflecting boundary conditions are suggested by references [11], [12], [24], [25], [26] and [27]. There are two approach to determine the inlet and outlet conditions: one dimensional method of characteristic [24], [25], [26], [27] and interpolation technique [11], [12].

Table 4.1 Inflow and outflow characteristic boundary conditions [16]

	INFLOW		OUTFLOW	
	Subsonic	Supersonic	Subsonic	Supersonic
# of Physical BCs	2	3	1	0
# of Numerical BCs	1	0	2	3

Interpolation technique will be presented for inlet and outlet boundary conditions here but both of them are implemented into the code. The all details about the one

dimensional method of characteristic are available in the references [24], [25], [26] and [27].

For supersonic inlet, all the information comes from the free stream conditions. Therefore, all primitive variables are taken from the free stream.

$$\rho_b = \rho_\infty, \quad u_b = u_\infty, \quad v_b = v_\infty, \quad P_b = P_\infty \quad (4.60)$$

But for subsonic inlet, one characteristic comes from the interior of the domain as shown in table 4.1. Then,

$$C_- = (u_d n x + v_d n y) - \frac{2c_d}{\gamma - 1} \quad (4.61)$$

where index d donates the state inside the domain. The speed of sound at the boundary can be found as

$$c_b = \frac{-R^-(\gamma - 1)}{(\gamma - 1) \cos^2 \beta + 2} \left\{ 1 + \cos \beta \sqrt{\frac{[(\gamma - 1) \cos^2 \beta + 2] c_0^2}{(\gamma - 1)(C_-)^2} - \frac{\gamma - 1}{2}} \right\} \quad (4.62)$$

Where β is the flow angle relative to the boundary and can be found as

$$\cos \beta = - \frac{(u_d n x + v_d n y)}{\sqrt{u_d^2 + v_d^2}} \quad (4.63)$$

The stagnation speed of sound in Equation (4.62) can be found as

$$c_0^2 = c_d + \frac{\gamma - 1}{2} \sqrt{u_d^2 + v_d^2} \quad (4.64)$$

Then at the boundaries, remaining primitive variables can be evaluated as

$$\begin{aligned} T_b &= T_0 \left(\frac{c_b^2}{c_0^2} \right) \\ P_b &= P_0 \left(\frac{T_b}{T_0} \right)^{\frac{\gamma}{\gamma-1}} \\ \rho_b &= \frac{P_b}{RT_b} \\ V_b &= \sqrt{2c_p(T_0 - T_b)} \end{aligned} \quad (4.65)$$

The x and y velocity components at the inlet boundary can be written as

$$u = V_b \cos(\beta), \quad v = V_b \sin(\beta) \quad (4.66)$$

For supersonic outflow, all the primitive variables are obtained from the interior of the domain because all the characteristics come from the interior point as shown in table 4.1.

$$\rho_b = \rho_d, \quad u_b = u_d, \quad v_b = v_d, \quad P_b = P_d \quad (4.67)$$

For turbomachinery applications usually the back pressure is specified at the exit plane. For subsonic outlet, the remaining variables are obtained from characteristic relations as

$$P_b = P_{back,specified} \quad (4.68a)$$

$$\rho_b = \rho_d + \frac{P_b - P_d}{c_0^2} \quad (4.68b)$$

$$u_b = u_d + \frac{n_x(P_d - P_b)}{(\rho_0 c_0)} \quad (4.68c)$$

$$v_b = v_d + \frac{n_y(P_d - P_b)}{(\rho_0 c_0)} \quad (4.68d)$$

4.4.3. Periodic Boundary Condition

The periodicity is given to the domain with the concept of dummy cell. Figure 4.8 shows the treatment of dummy cells for periodic boundaries.

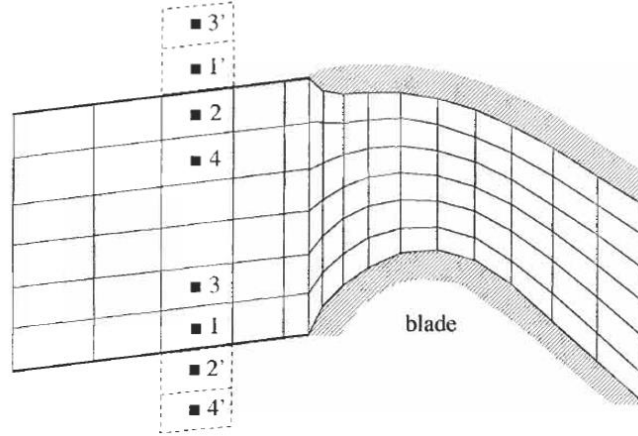


Figure 4.9 Periodic boundaries [12]

The primitive variables can be set at the dummy cells as

$$U_{1'} = U_1, \quad U_{2'} = U_2, \quad U_{3'} = U_3, \quad U_{4'} = U_4 \quad (4.69)$$

Then, in order to calculate the fluxes at the periodic boundaries, the same procedure which is used in the interior cells can be applied. First, the right and left state will be determined and then the flux at the boundary faces are calculated.

4.5. Temporal Discretization

As well as the spatial discretization, temporal discretization is also required for the time marching method. The calculated residuals are integrated in time using multistage explicit time stepping (Runge-Kutta time integration) scheme. The multistage can be viewed as sequence of update of the solution in a number of steps (stages). The most general form of the m-stage Runge-Kutta time stepping can be written as

$$\begin{aligned}
 \vec{W}_{cell}^0 &= \vec{W}_{cell}^n \\
 \vec{W}_{cell}^1 &= \vec{W}_{cell}^0 - \alpha_1 \frac{\Delta t_{cell}}{\Omega_{cell}} \vec{R}_{cell}^0 \\
 \vec{W}_{cell}^2 &= \vec{W}_{cell}^0 - \alpha_2 \frac{\Delta t_{cell}}{\Omega_{cell}} \vec{R}_{cell}^1 \\
 &\vdots \\
 &\vdots \\
 &\vdots \\
 \vec{W}_{cell}^{n+1} &= \vec{W}_{cell}^0 - \alpha_m \frac{\Delta t_{cell}}{\Omega_{cell}} \vec{R}_{cell}^{m-1}
 \end{aligned} \tag{4.70}$$

where, $\vec{W}$ is the conservative variable vector and $\vec{R}$ is the residual vector calculated from current conservative variables. The term α is the stage coefficient of the time integration and its' values for different schemes are given in table 4.2.

Table 4.1 Constants of multistage schemes [12]

	First Order Schemes			Second Order Schemes		
Stages	3	4	5	3	4	5
α_1	0.1481	0.0833	0.0533	0.1918	0.1084	0.0695
α_2	0.4	0.2069	0.1263	0.4929	0.2602	0.1602
α_3	1.0	0.4265	0.2375	1.0	0.5052	0.2898
α_4		1.0	0.4414		1.0	0.5060
α_5			1.0			1.0

The time step Δt_{cell} can be given as a fixed time step or it can be calculated using local time steeping. In this study, local time steeping is used.

4.5.1. Determination of Time Step

The time step for a structured grid [12] can be found as

$$\Delta t_{cell} = CFL \frac{\Omega_{cell}}{(\tau_{cell}^I + \tau_{cell}^J)_{cell}} \quad (4.71)$$

The spectral radii can be calculated as

$$\tau_{cell}^I = (|un_x^I + vn_y^I| + c)\Delta S_{cell}^I \quad (4.72a)$$

$$\tau_{cell}^J = (|un_x^J + vn_y^J| + c)\Delta S_{cell}^J \quad (4.72b)$$

where

$$n^I = \frac{1}{2} \begin{bmatrix} n_{x1} & -n_{x2} \\ n_{y1} & -n_{y2} \end{bmatrix}, \quad \Delta S_{cell}^I = \frac{1}{2} (\Delta S_1 + \Delta S_2) \quad (4.73a,b)$$

$$n^J = \frac{1}{2} \begin{bmatrix} n_{x3} & -n_{x4} \\ n_{y3} & -n_{y4} \end{bmatrix}, \quad \Delta S_{cell}^J = \frac{1}{2} (\Delta S_3 + \Delta S_4) \quad (4.73c,d)$$

The above geometric relations can be obtained from Figure 4.9. The CFL number is given in reference [12] for this scheme as 2.0 and 0.92 for the first and second order schemes, respectively.

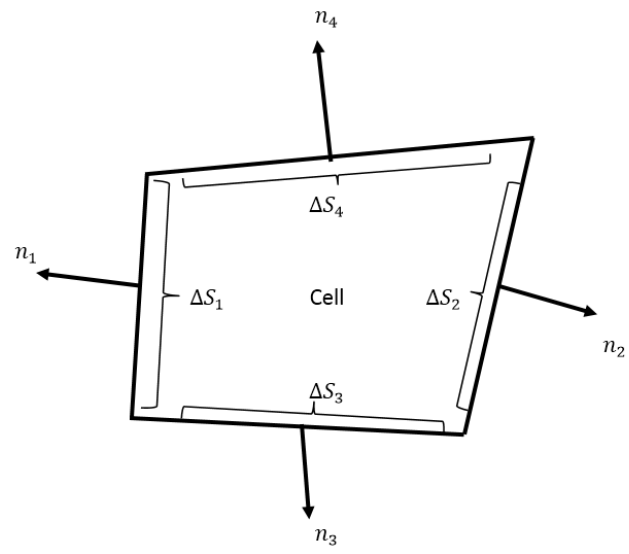


Figure 4.10 Cell Geometry for CFL calculation

CHAPTER 5

TEST CASES AND SOLVER VALIDATION

Before the validation of the solvers with various test cases, the mesh independency for both newton and time marching solver is given. The number of grid points affects the solution up to a certain level after which the accuracy of the numerical analyses does not change much.

The current Euler solvers can be used for both ducts and cascade (blade to blade) domain. For duct test case, subsonic and transonic flow solutions of Ni bump and the transonic converging-diverging nozzle will be presented. The results will be compared with references [47] and [48] in the literature.

There are five cascade test cases taken from reference [49]. The first cascade is well known Hobson turbine cascade. The second cascade is the transonic NACA 0012 cascade with the shock wave. The third one is subcritical Sanz turbine cascade. The flow is subsonic through whole domain. The fourth cascade case is the supercritical Sanz cascade. There is a supersonic flow pocket within the blade passage. The final one is the supercritical Sanz compressor cascade. The reference solutions are obtained from exact hodograph method except NACA 0012 cascade (50).

For the rotor test case, single stage R030- NACA 65/60 compressor is selected. [49], [51]. The single compressor stage test results and computational results from reference [3] will be compared with both Newton and time marching solver results.

5.1. Mesh Independency

In the numerical methods, when the number of grid points is increased, the accuracy of the results also increases. However, the effect of the number of grid point on the

accuracy of the solutions does not affect the final solution much after a certain level. In order to determine the mesh independency of the Newton and time marching Euler solvers, three mesh configurations for bump domain which are shown in Figures 5.1a, 5.1b and 5.1c are selected. The results of the mesh independency analysis for Newton and time marching solvers in subsonic and transonic flow are presented in Figures 5.2a, 5.2b, 5.3a and 5.3b.

The results has shown that the accuracy of the results increases when the number of grid points is increased. The maximum Mach number on the bump occurs at the 50% of the axial length for the subcritical flow. When the number of grid points is increased, the maximum Mach number on the bump surface is getting closer to the correct value. While the deviation between the (33 x 9) and (65 x 17) grid points is about 1.3%, it becomes about 0.43% between (65 x 17) and (129 x 33) grid points. The result of the case with the (129 x 33) grid points is sufficiently close the correct value. Therefore, using the number of grid points between (65 x 17) and (129 x 33) is enough to obtain satisfactory results.

In the Table 5.1, the E1 and E2 represent the error between mesh configurations at the location of maximum Mach number on the bump surface. The E3 gives the error between the results of the finest mesh and reference solution. Table 5.1 is presented only for the subsonic flow within the duct.

Table 5.1. Error between grid levels

Time Marching			Newton		
E1 (%)	E2 (%)	E3 (%)	E1 (%)	E2 (%)	E3 (%)
1.3	0.43	0.21	1.45	0.5	0.22

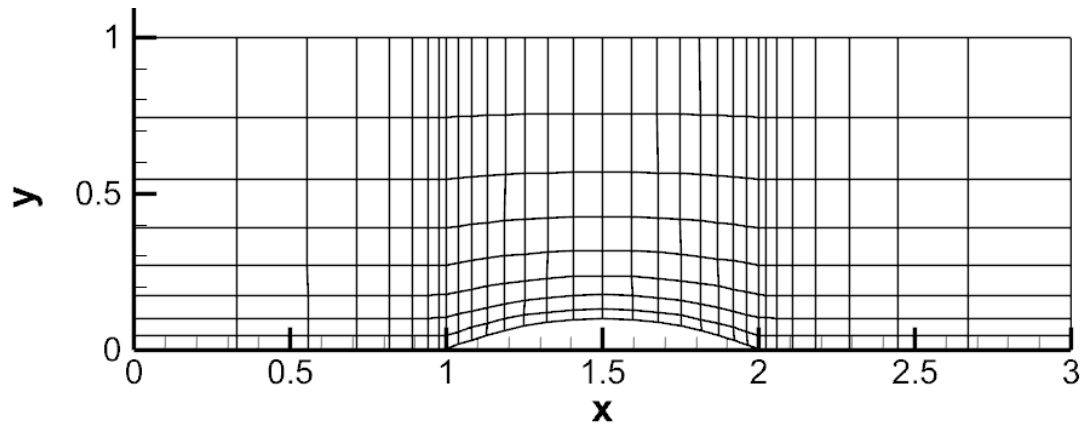


Figure 5.1a Flow domain with 33 x 9 number of grid points

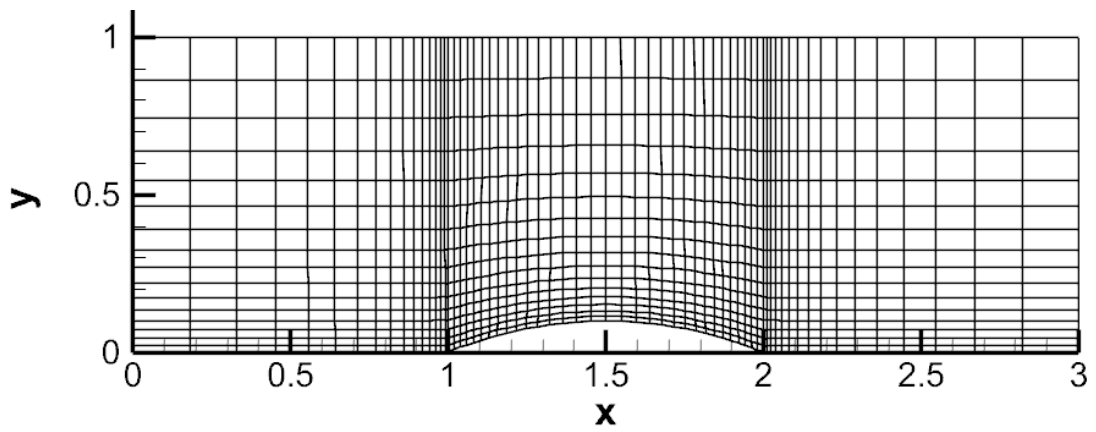


Figure 5.1b Flow domain with 65 x 17 number of grid points

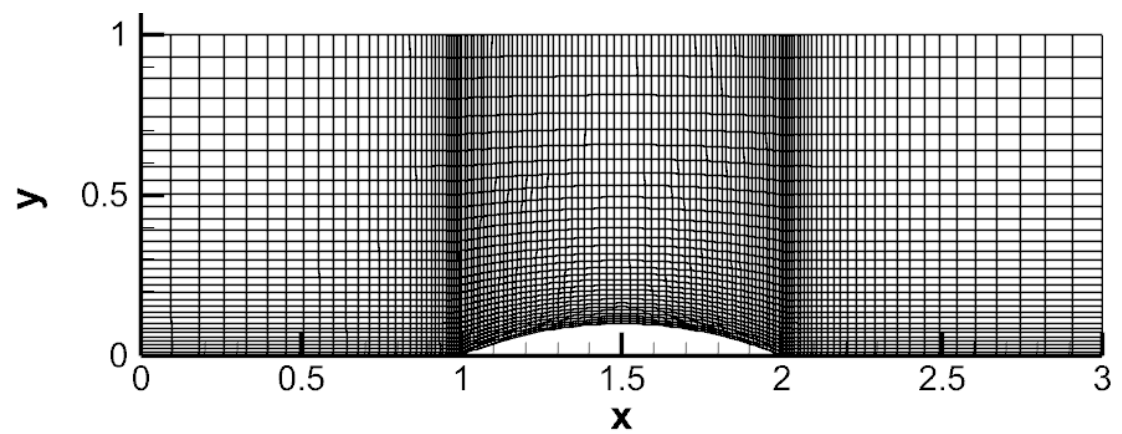


Figure 5.1c Flow domain with 129 x 33 number of grid points

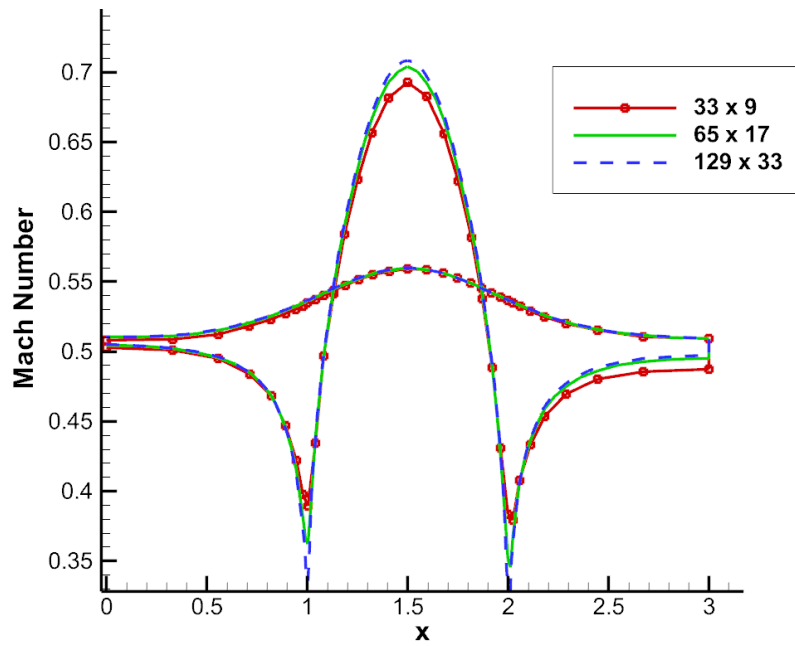


Figure 5.2a Results of time marching method for subsonic flow with the three different mesh arrangements

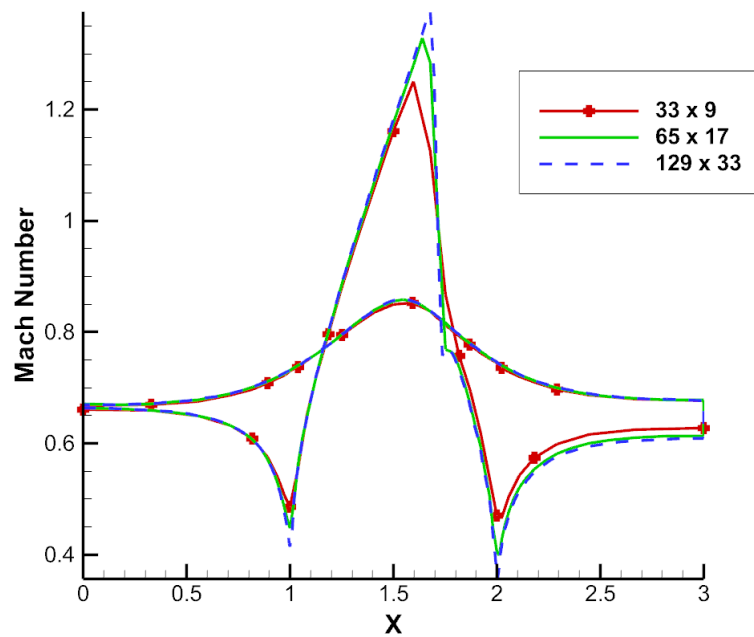


Figure 5.2b Results of time marching method for transonic flow with the three different mesh arrangements

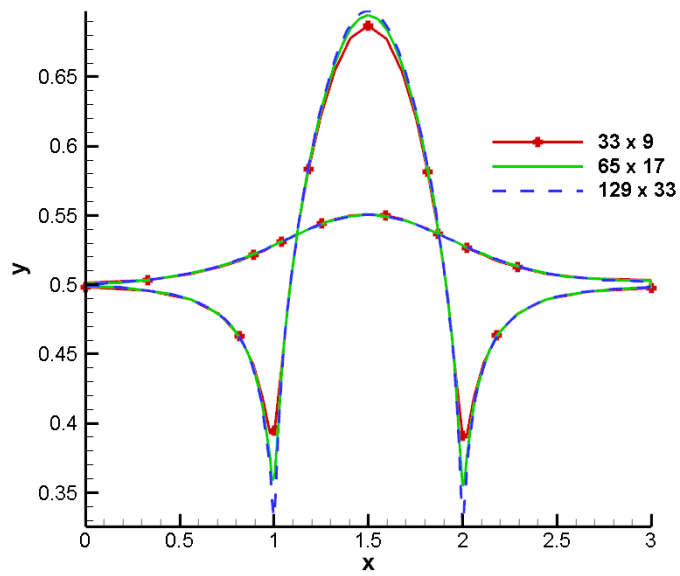


Figure 5.3a Results of Newton method for subsonic flow with the three different mesh arrangements

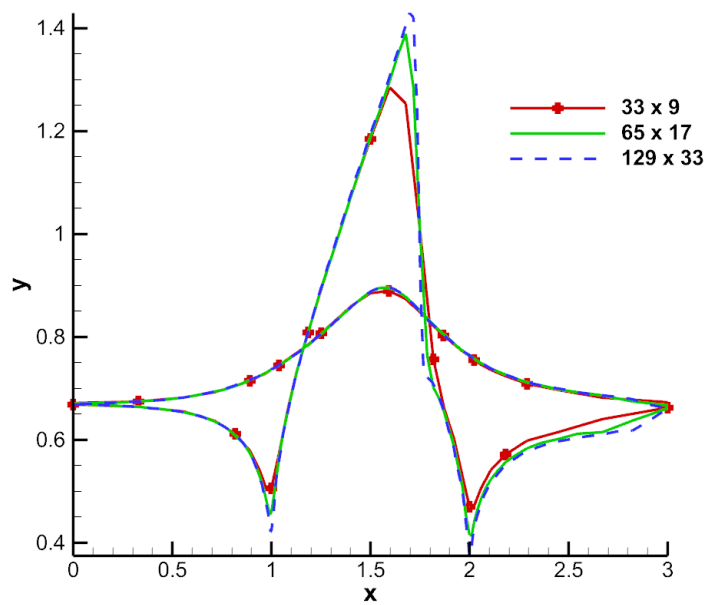


Figure 5.3b Results of Newton method for transonic flow with the three different mesh arrangements

5.2. Validation

5.2.1. Subsonic Ni Bump

This test case is well known for inviscid flow calculations in the literature including the circular bump with thickness to chord ratio of 10%. The Mach or pressure contours should be symmetric with respect to mid-plane of the whole domain for an inlet Mach number of 0.5. The flow enters the duct axially. Total pressure and temperature is given as 101325 kPa and 300 K, respectively. Since the flow is not choking, the inlet Mach number and the flow angle are specified for Newton solver as the boundary conditions. For time marching solver, back pressure, inlet total quantities and inlet flow angle are specified. The back pressure is set to inlet free stream pressure. Symmetry on the wall in terms of both static pressure and Mach number are satisfied but only the variation of the wall Mach number is presented in Figure 5.4. The results obtained from present solvers are compared with the result taken from reference [47]. In order to visualize the symmetry in the flow field, mach contours are also presented in Figures 5.5, 5.6 and 5.7.

It is clearly seen from Figure 5.4 that the results are satisfactory for both time marching and Newton solvers as compared to the computational result taken from reference [47]. The maximum deviations between the result taken from reference [47] and results of the present solvers occur at the leading and trailing edges of the bump. Since the bump has sharp leading and trailing edges, there are discontinuous changes in streamline slopes introducing vorticity at the leading and trailing edges of the bump. In reference [47], the sharp edges are removed by lifting one wall point just ahead of the corners about 0.5‰ of the chord length and corner points about 1.5‰ of the chord length. Therefore, smoother results are obtained at the corner points in the reference [47] as compared to the original results. In the present study, since there is no modification for the edges of the bump the deviation of the developed solvers from the reference solution increases at the corners of the bump.

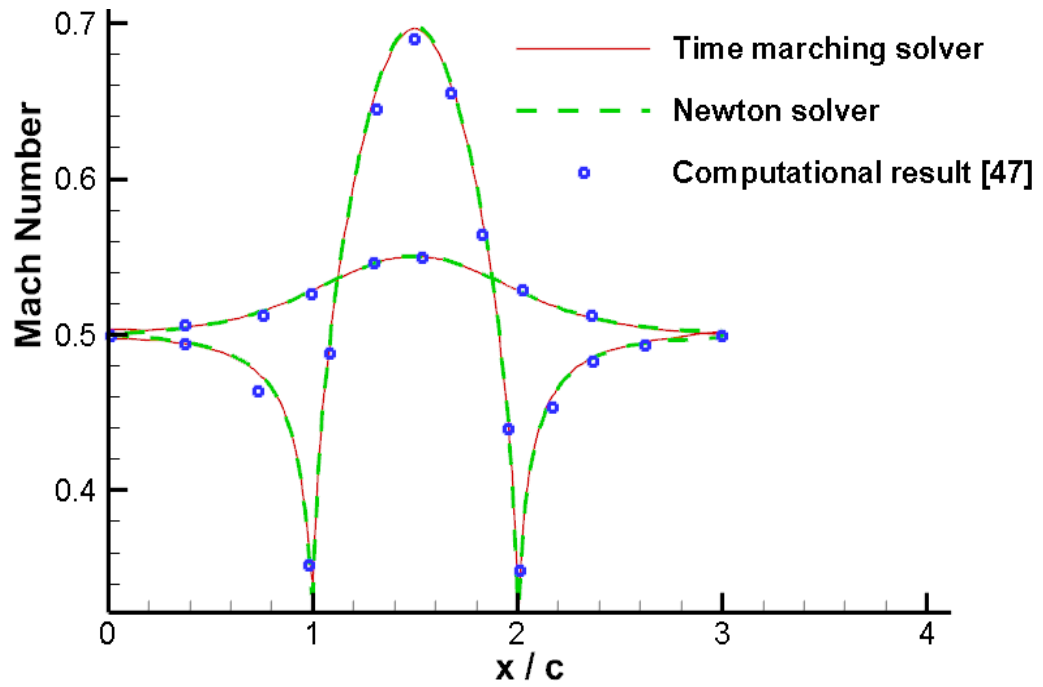


Figure 5.4 Variation of wall Mach number for subsonic flow over the bump

In reference [47], Mach number contour plot is also presented for validation of the case. In the next page, Figures 5.5a, 5.5b and 5.5c shows the isomach lines obtained from the Newton solver, time marching solver and reference [47]. Since both methods have some numerical errors, the full symmetry in flow field cannot be obtained. The solution given in the reference [47] also does not have full symmetry due to same reason though the improvements mentioned above.

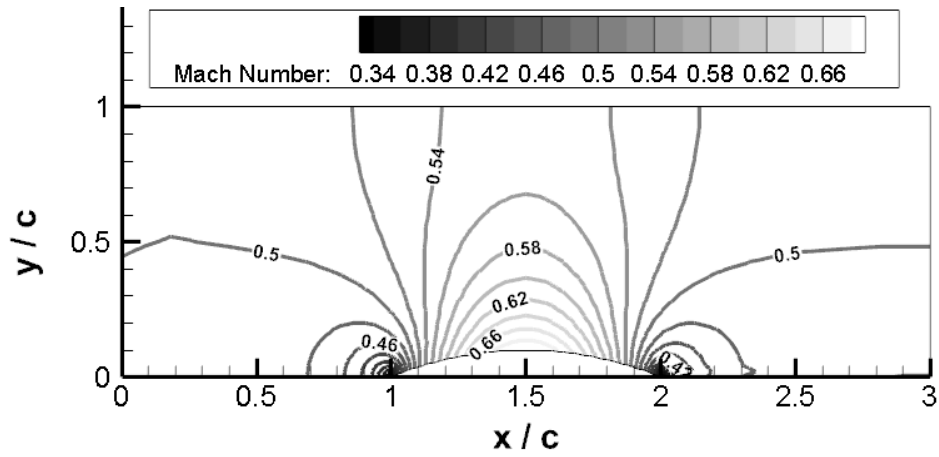


Figure 5.5a Isomach lines, time marching method

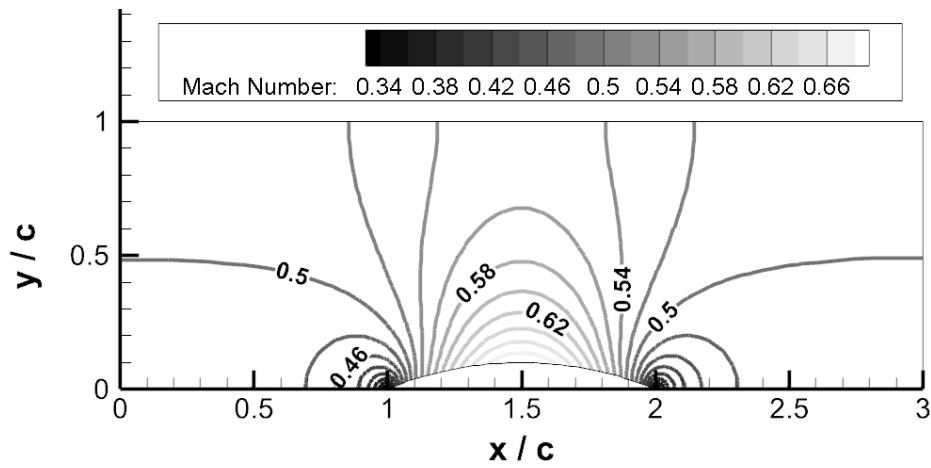


Figure 5.5b Isomach lines, Newton method

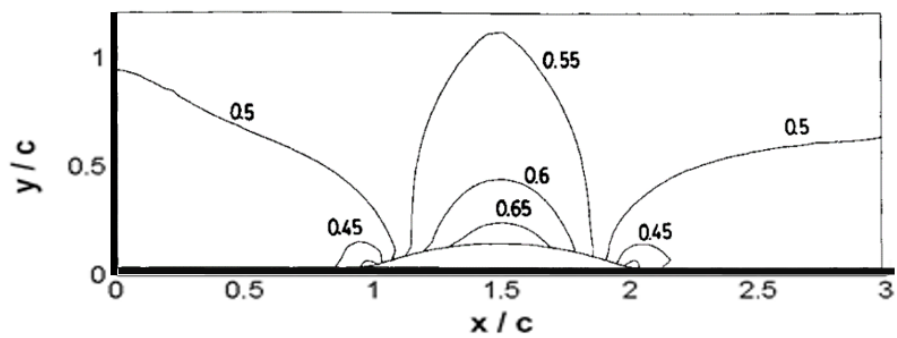


Figure 5.5c Isomach lines, computational result [47]

The numerical errors occurring in the Newton method is associated with the stagnation point. If the mass flux in each streamtube is the same, the streamtube width at the stagnation point becomes very large. In other words, the lack of mesh resolution near the stagnation point causes numerical errors in the Newton solver. The numerical error occurring in the time marching method is the direct consequence of the presence of the numerical dissipation. These errors can further be decreased by increasing the mesh resolution near the edges of the bump.

5.2.2. Transonic Ni bump

Same geometry defined in previous case is also used for transonic flow calculations with a different inlet Mach number. The inlet Mach number and flow angles are set to 0.675 and 0.0 degree, respectively. The back pressure is decreased to 0.985 fold of the inlet free stream static pressure. At this time, since the flow is choking at the throat of the duct, choking boundary conditions are used for Newton solver. Therefore, same back pressure specified for time marching solver is also given to the Newton solver as an exit boundary condition. The variation of the wall Mach number obtained from both solvers and reference [48] is given in Figure 5.6.

The computational result in reference [48] shows that shock wave is located at 72% of the chord length. Both time marching and Newton methods give satisfactory results in terms of location of shock wave. For the visualization of the flow field, Mach number contours are given in Figures 5.7a, 5.7b and 5.7c. The characteristics of the contours in reference [48] are similar with those of Newton and time marching results.

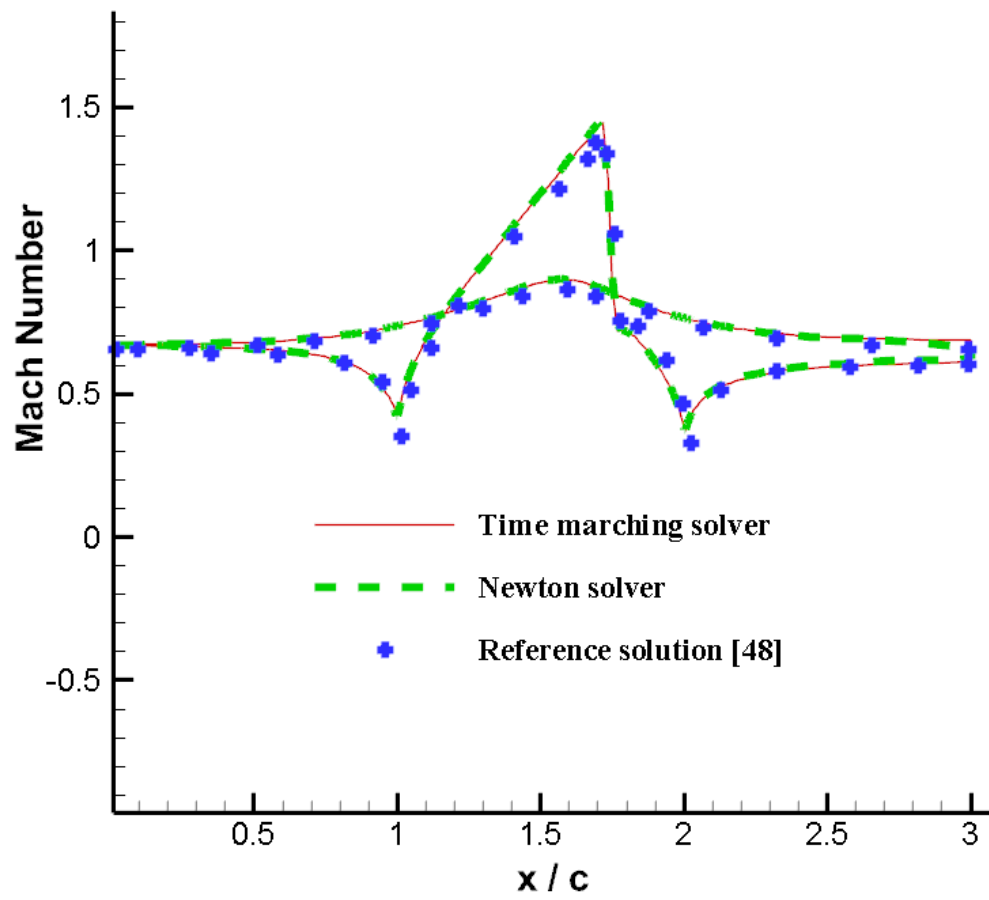


Figure 5.6 Variation of wall Mach number for transonic flow over the bump

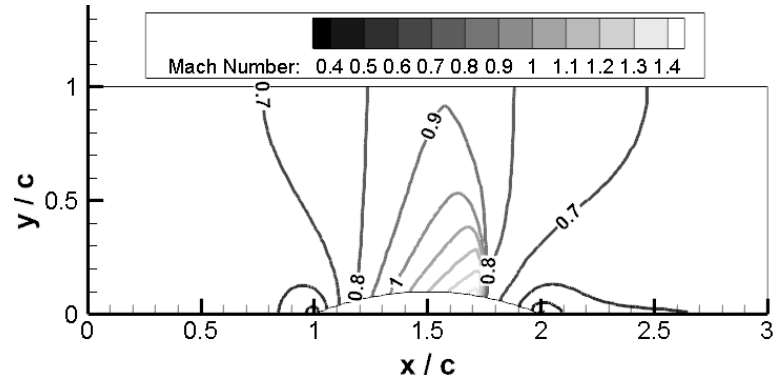


Figure 5.7a Isomach lines, Newton solver

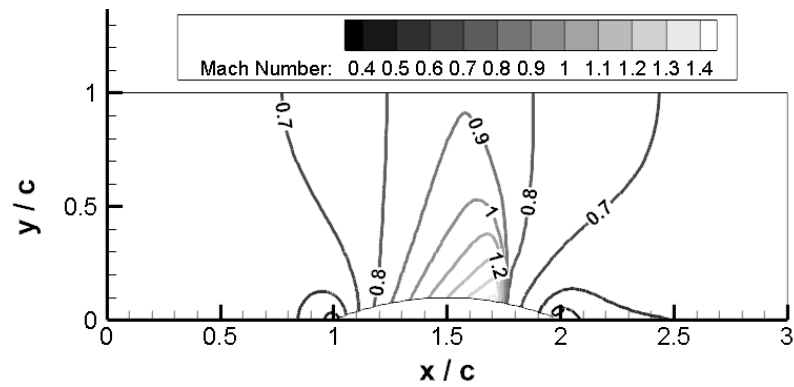


Figure 5.7b Isomach lines, Newton solver

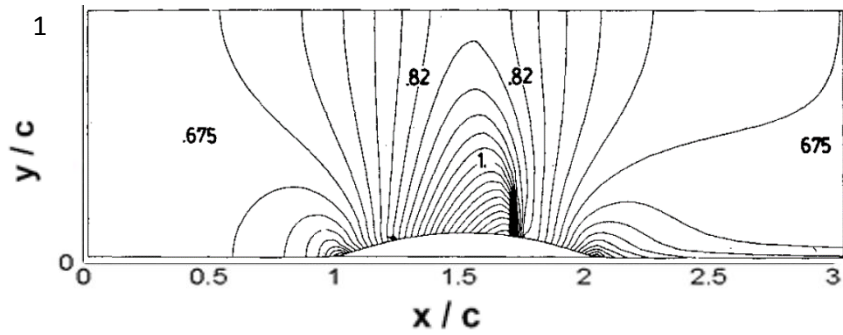


Figure 5.7c Isomach lines, computational result [48]

5.2.3. Transonic Converging Diverging Nozzle

The test case is taken from the chapter 16 of the reference [11]. For the validation, the quasi one dimensional solution of the nozzle is obtained using the gas dynamics table taken from the appendix of the reference [54]. The flow enters the nozzle axially with

an inlet Mach number of 0.24. The ratio of the throat area to inlet area is 0.4 and the flow is choking at throat of the nozzle. The ratio of the pressure at the exit to that of at the inlet is 0.8475, which will be used for the time marching solver as a boundary condition. The ratio of the total density at the exit to that of at the inlet is 0.9738, which will be used for the Newton solver as a boundary condition.

When the above boundary conditions are applied at the inlet and exit, there is a normal shock wave at the 70% of the axial distance of the nozzle. The variation of the midline Mach number of the nozzle along the axial length is given in figure 5.8.

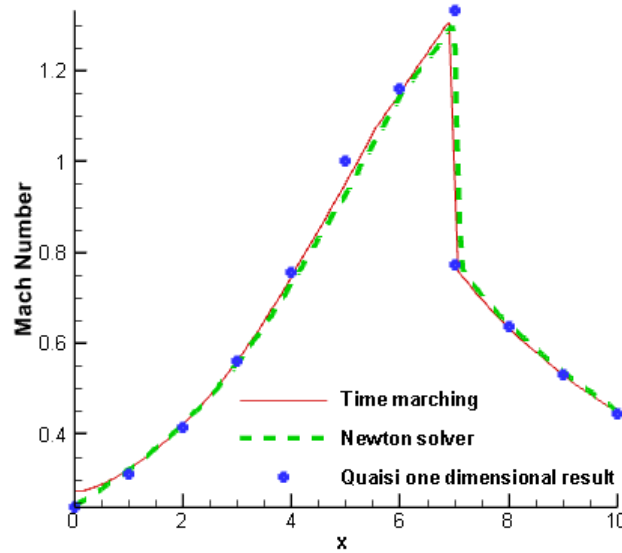


Figure 5.8 Converging-diverging nozzle, variation of the mid-line Mach number

The present codes have excellent agreement with the analytical results obtained from quasi-one dimensional analysis. The upstream and downstream Mach number of the shock wave obtained from the quasi-one dimensional flow analysis is 1.33 and 0.7712, respectively. The upstream and downstream Mach number of the shock wave obtained from the result of the time marching solver is about 1.305 and 0.76, respectively. The newton solver gives upstream and downstream Mach number of the shock wave about 1.30 and 0.7605. The deviations from the one dimensional analysis are due to the two

dimensional effect. The contour plots obtained from both solvers are shown in Figures 5.9a and 5.9b.

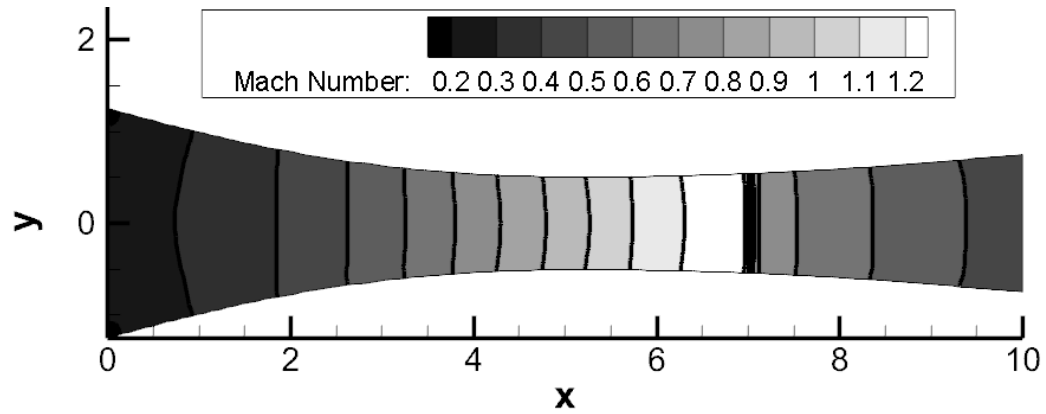


Figure 5.9a Isomach lines, time marching solver

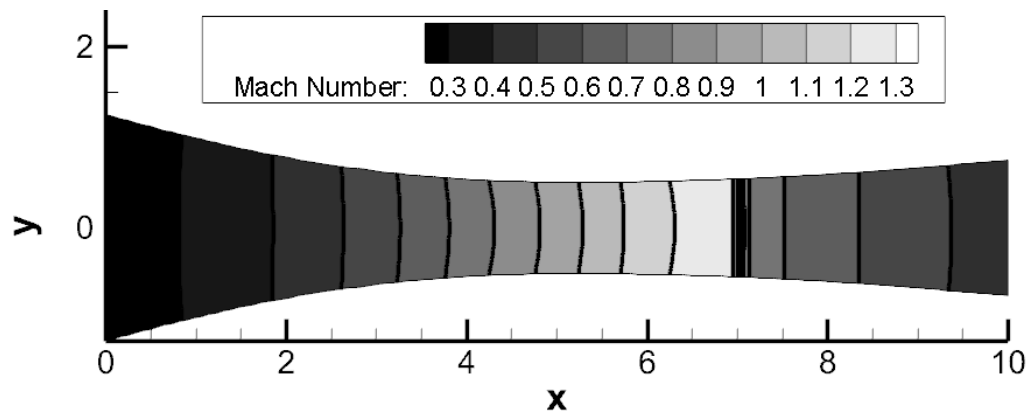


Figure 5.9 b Isomach lines, Newton solver

From the contour plots, the shock appears clearly at the 70% of the axial length of the nozzle. The flow must also be symmetric with respect to the center line of the nozzle. The symmetry in the flow field can be clearly seen in Figures 5.9a and 5.9b

5.2.4. Hobson Cascade

The Hobson cascade is the first test case for the validation of the present codes through cascade. The hodograph solution of the Hobson turbine cascade is given in reference [49]. There is a supersonic pocket on the suction side of the blade but there is no shock wave. The inlet Mach numbers and flow angle of the cascade are 0.575 and 46.123 degree, respectively. Exit boundary conditions are same with the inlet conditions. All other information about the test case can be found in reference [49]. The variations of the surface wall Mach number obtained from the Newton and time matching solvers are plotted and compared with the result in reference [49].

The variations of the wall Mach number of the blade obtained from the present codes are given in Figures 5.10 together with the result of the hodograph solution [49].

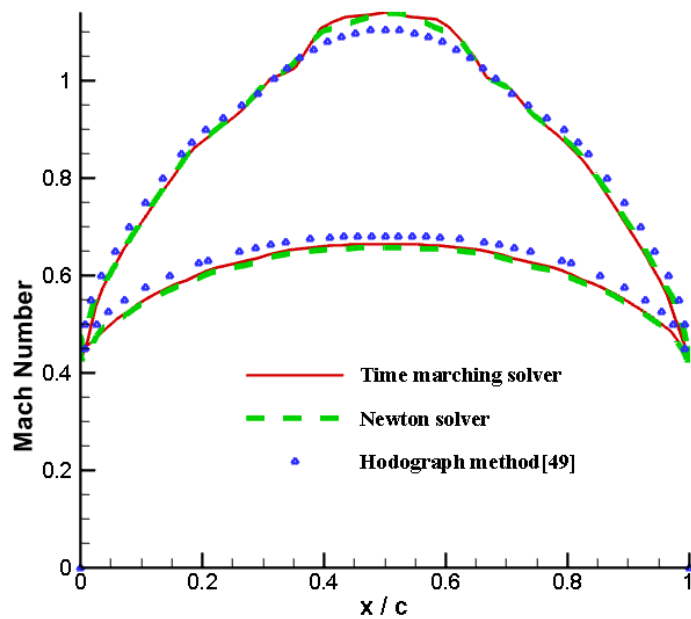


Figure 5.10 Variation of surface Mach number of Hobson turbine blade

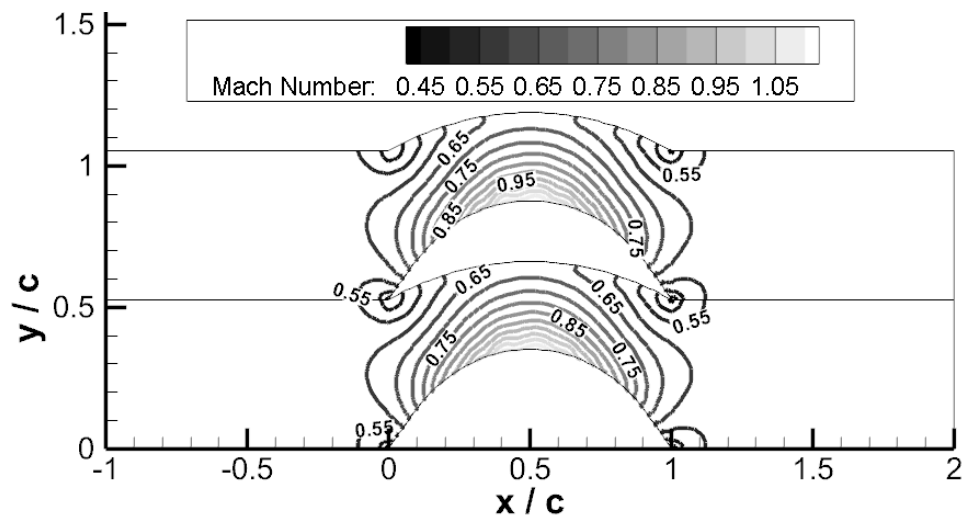


Figure 5.11a Isomach lines, time marching solver

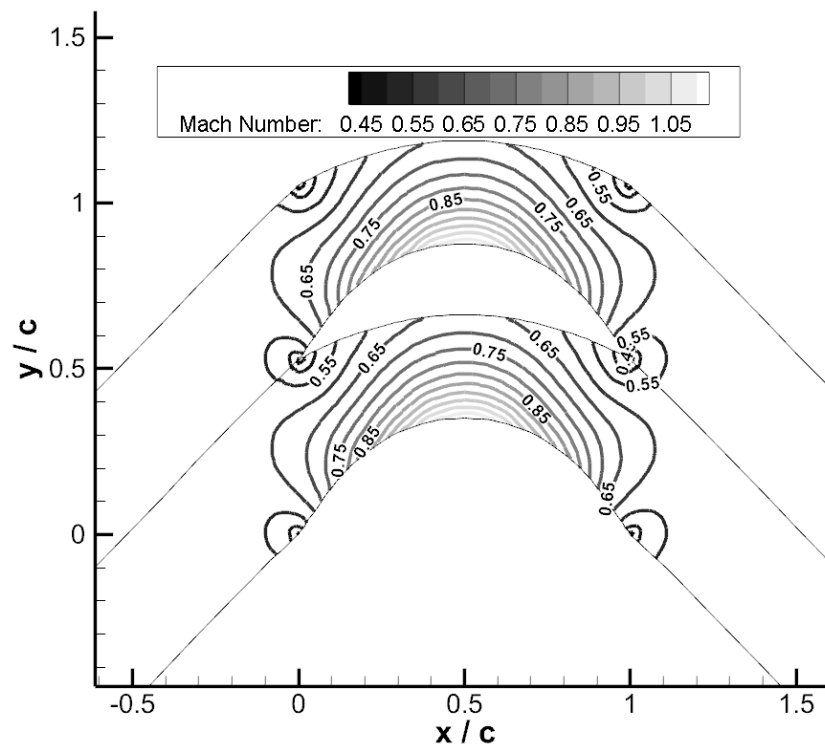


Figure 5.11b Isomach lines, Newton solver

Hobson cascade was designed with the hodograph method to give a shock-free supersonic pocket. It can be seen from Figure 5.10 that the solution is essentially shock free and in a close agreement with the original hodograph design. Since this cascade is shock-free and geometrically symmetric with respect to a vertical line passing through 50% of the axial length, the flow field must be symmetric with respect to this line. The smooth symmetric patterns shown in Fig. 5.11a and 5.11b also indicate the accuracy of the solution.

5.2.5. NACA 0012 Cascade

This test case is the flow through unstaggered NACA 0012 cascade. The solidity of the cascade is 1. The inlet Mach number is set to 0.647 and the inlet slope is set to zero. The ratio of exit pressure to the inlet stagnation pressure is 0.721.

In reference [50] it is stated that the Mach number upstream of the shock wave is about 1.32 and shock is located at about 73% of the axial chord. Both time marching and Newton solvers locate the shock wave at about 73% of the axial chord of the blade. The locations of the shock wave obtained from present codes perfectly agree with the result of reference [50]. The variations of the wall Mach number obtained from present codes and computational method in reference [50] are shown in Figure 5.12

The results in Figure 5.12 demonstrate that the present predictions agree well with the result in reference [50]. The upstream and downstream Mach numbers of the shock wave obtained from the present methods are about 1.33 and 0.78, respectively. Although there are two dimensional effects in the solution and the shock is not perfectly normal, quasi one dimensional solution of the shock wave may be used for comparison of the results. From the appendix of reference [54], the downstream Mach number corresponding to the upstream Mach number of 1.33 at shock wave is 0.7712. The present solvers gives the downstream Mach number of 0.78. The result obtained from reference [50] gives the upstream and downstream Mach number of 1.32 and 0.84 at shock wave, respectively. Therefore, the present codes seem to give more accurate result as compared to that of reference [50] in terms of shock strength.

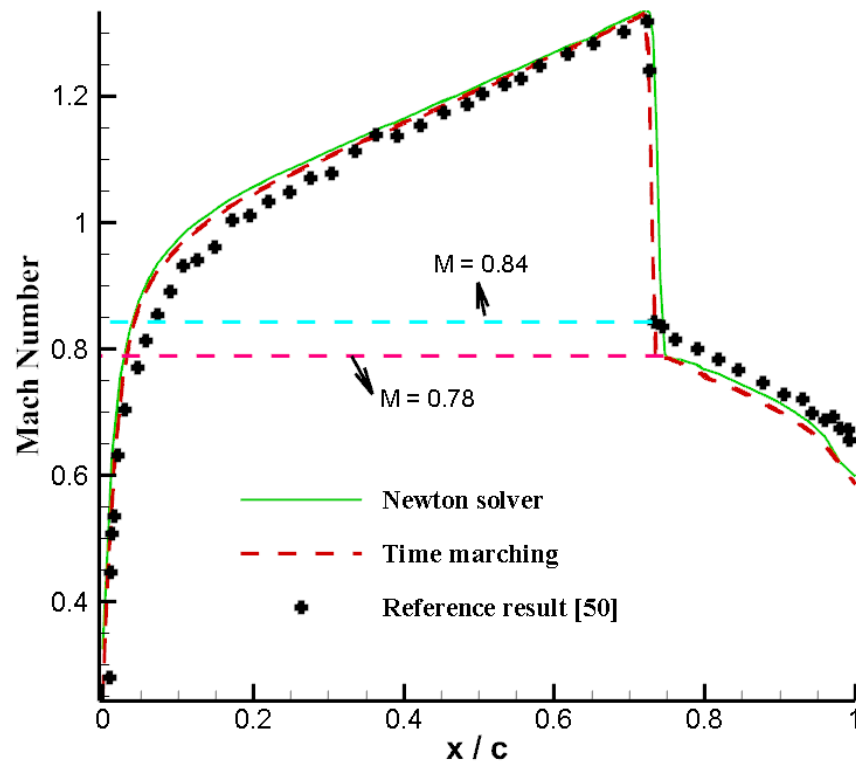


Figure 5.12 Variation of surface Mach number of NACA 0012 profile

A deviation between the results of present method and reference result is seen on the blade surface from leading edge to 35% of the axial chord. The reason for this deviation is that the mesh resolution at this location is low in the result of reference [50]. In the present analysis, the mesh resolution is better than that of used in reference [50]. Therefore, this deviation may be possible

The patterns shown in Figure 5.13a, 5.13b and 5.13c obtained from present codes are quite similar to that of computational result in reference [50].

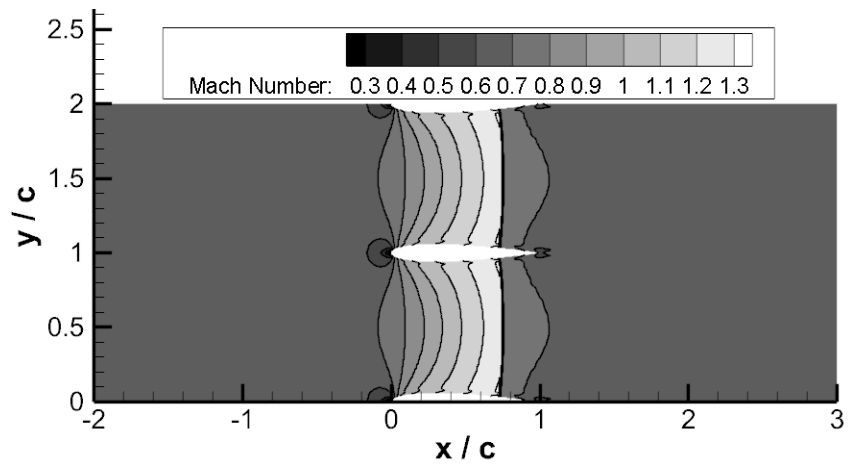


Figure 5.13a Isomach lines, Time marching solver

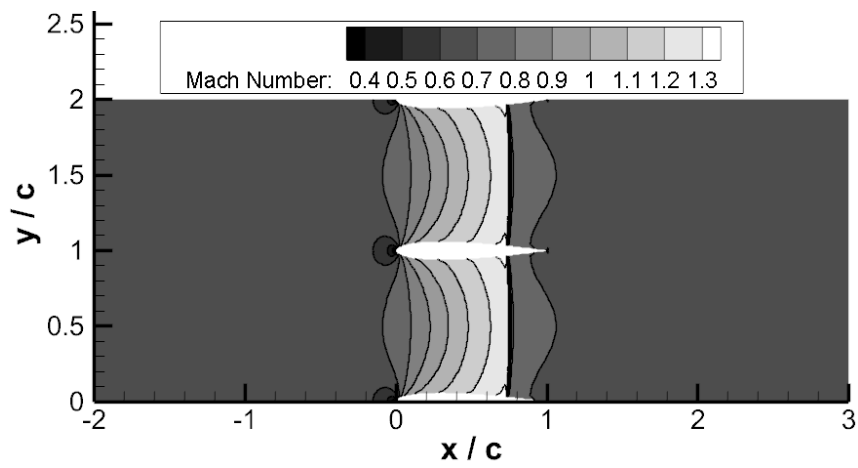


Figure 5.13b Isomach lines, Newton solver

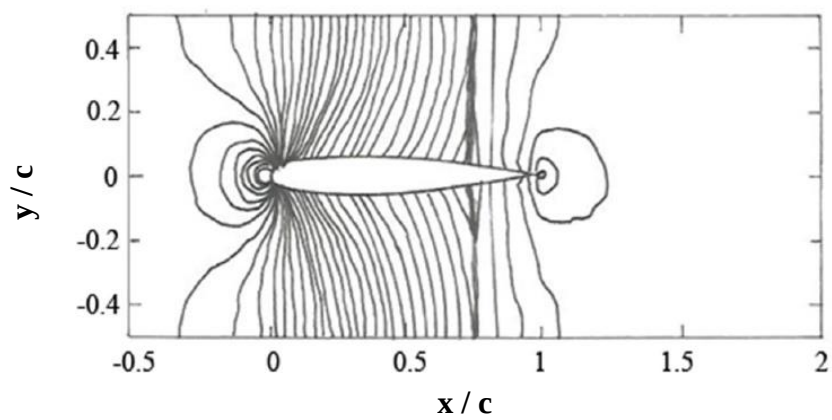


Figure 5.13c Isomach lines, computational result [50]

5.2.6. Sanz Subcritical Turbine Cascade

The Sanz subcritical turbine cascade is designed by hodograph method. The exact solution of this cascade is given in reference [49]. Flow is entirely subsonic through the passage. The cascade has an inlet Mach number of 0.343 with 36 degree inlet flow angle. The exit Mach number and flow angle are 0.765 and -57.35, respectively. The variation of the wall Mach number of the blade obtained from the present codes is given in Figure 5.14 together with result of the hodograph solution.

The results in Figure 5.14 demonstrate that both time marching and Newton solvers agree well with the hodograph method. There is a small difference in terms of static pressure between the result of the time marching solver and hodograph solution at the pressure side. This deviation is probably due to the mesh resolution at this location. The pressure ratios obtained from the time marching, Newton and hodograph method are 0.735, 0.733 and 0.736, respectively. In terms of pressure ratio, the time marching has an excellent prediction while the result of Newton solver is also quite accurate. The outlet flow angles obtained from the time marching, Newton and hodograph method are -57.42, -57.38 and -57.35 degrees, respectively. In terms of exit slope, Newton solver has closer agreement with the hodograph solution as compared to that of time marching solver.

The pressure distribution obtained from present codes are given in Figure 5.15a and 5.15b. A comment can be made for the result of time marching solver regarding to a non-uniformity at the trailing edge. Since the Kutta condition is not applied at the trailing edge in time marching solver, pressure jumps between suction and pressure sides may be seen. The Kutta condition forces the pressure of the suction and pressure side of the blade at trailing edge to be equal. Therefore, Newton solver gives smooth wave pattern at trailing edge. On the other hand, the contours obtained from time marching method at the trailing edge lose the uniformity. For the remaining parts of the domain, the contours obtained from both solvers are similar to each other. The trailing edge of the blade is not closed as shown in Figures 5.15a and 5.15b. Any attempt to close trailing edge could introduce some perturbation of the computed flow distribution.

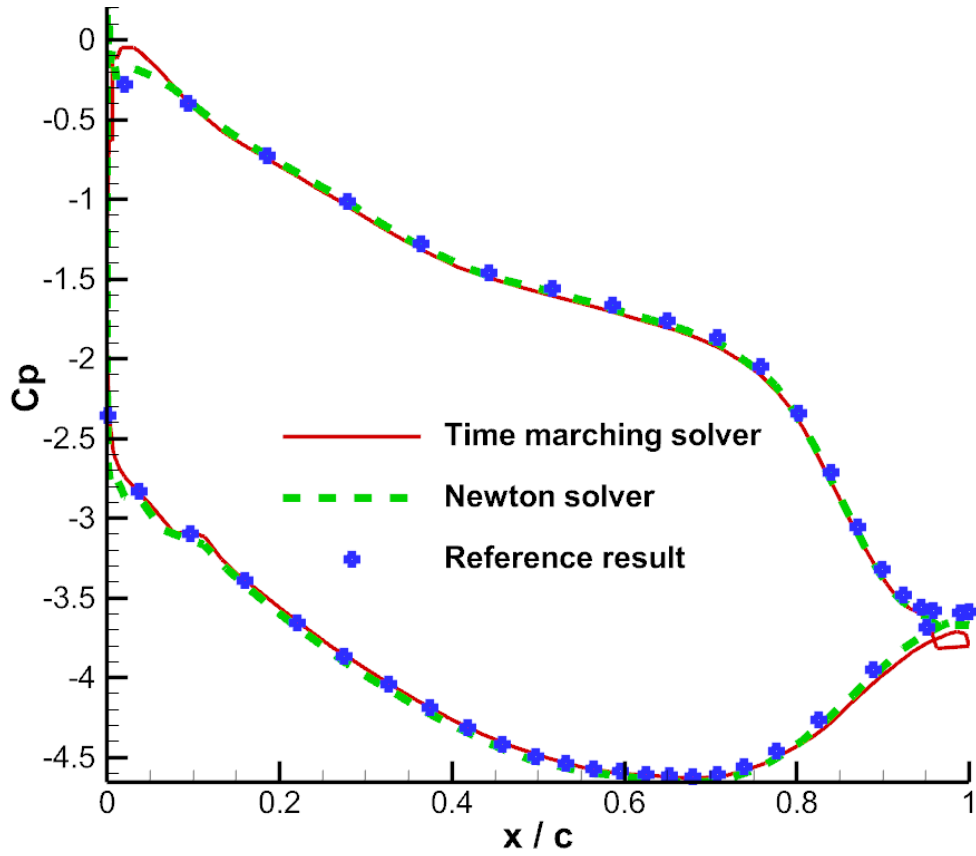


Figure 5.14 Variation of surface static pressure of Sanz subcritical turbine blade

The pressure coefficient can be defined as

$$C_p = \frac{\frac{P}{P_\infty} - 1}{\frac{1}{2}\gamma M_\infty^2} \quad (5.1)$$

where P_∞ and M_∞ is the free stream total pressure and Mach number, respectively.

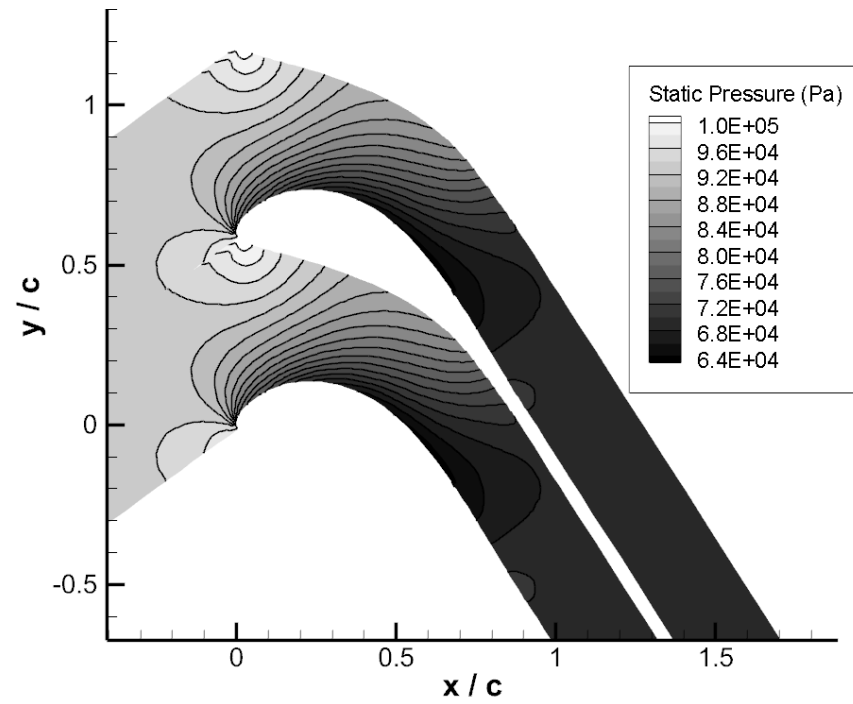


Figure 5.15a Static pressure contours, time marching solver

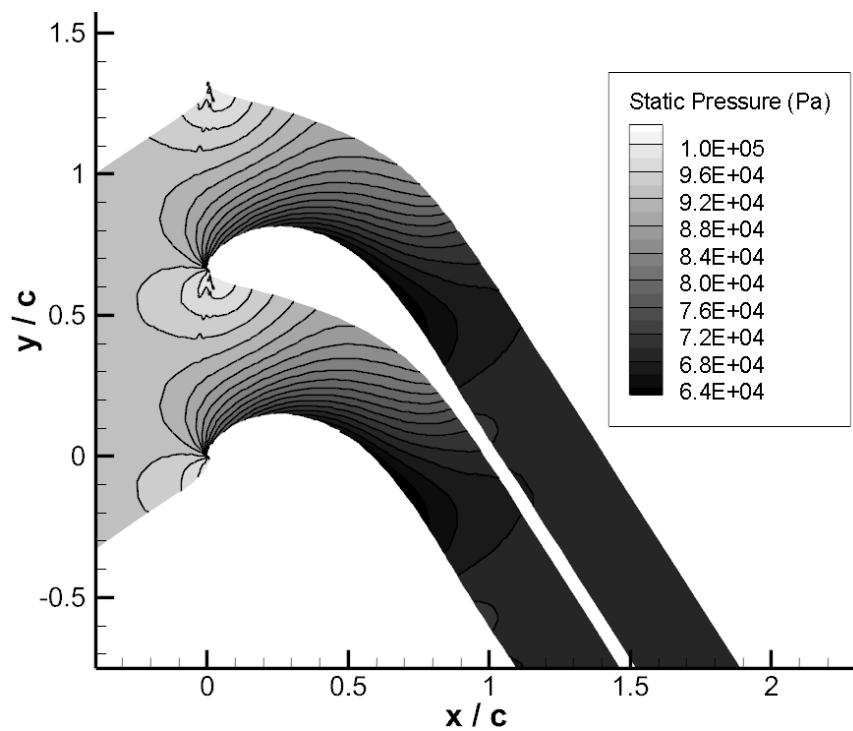


Figure 5.15b Static pressure contours, Newton solver

5.2.7. Sanz Supercritical Turbine Cascade

The Sanz supercritical turbine cascade is designed by hodograph method. The exact solution of this cascade is given in reference [49]. Flow is transonic through the passage without a shock wave. The cascade has an inlet Mach number of 0.394 with an inlet flow angle of 38.13 degree. The exit Mach number and flow angle are 0.741 and -56.93, respectively. Just like Sanz subcritical turbine, the trailing edge is not closed. Any attempt to close the trailing edge could cause some disturbance in the flow field. The wall pressure distribution of the blade obtained from the present codes is given in Figure 5.16 together with result of the hodograph solution [49].

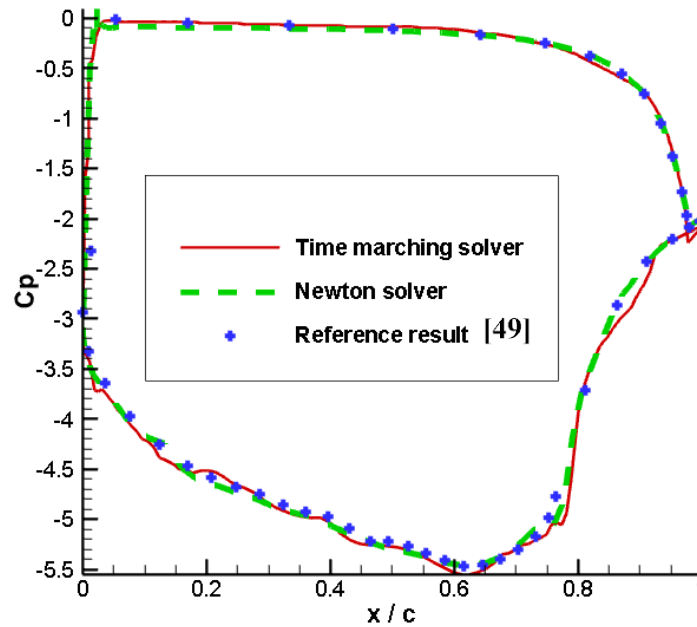


Figure 5.16 Variation of surface static pressure of Sanz supercritical turbine blade

The Mach number contours obtained from the present methods are given in Figures 5.17a and 5.17b.

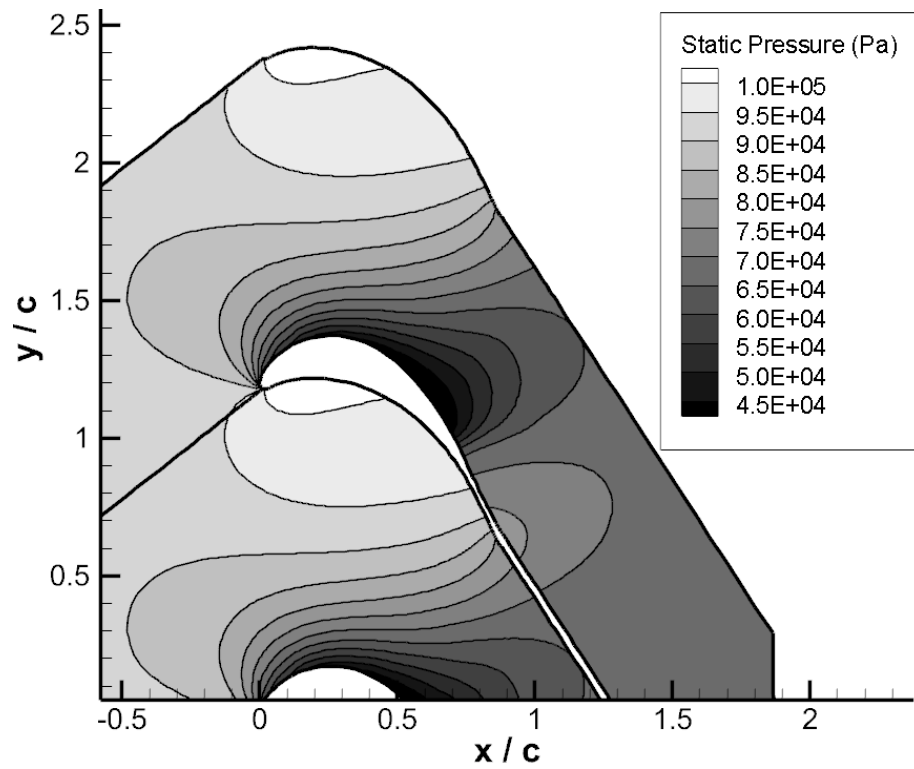


Figure 5.17a Isopressure lines, time marching solver

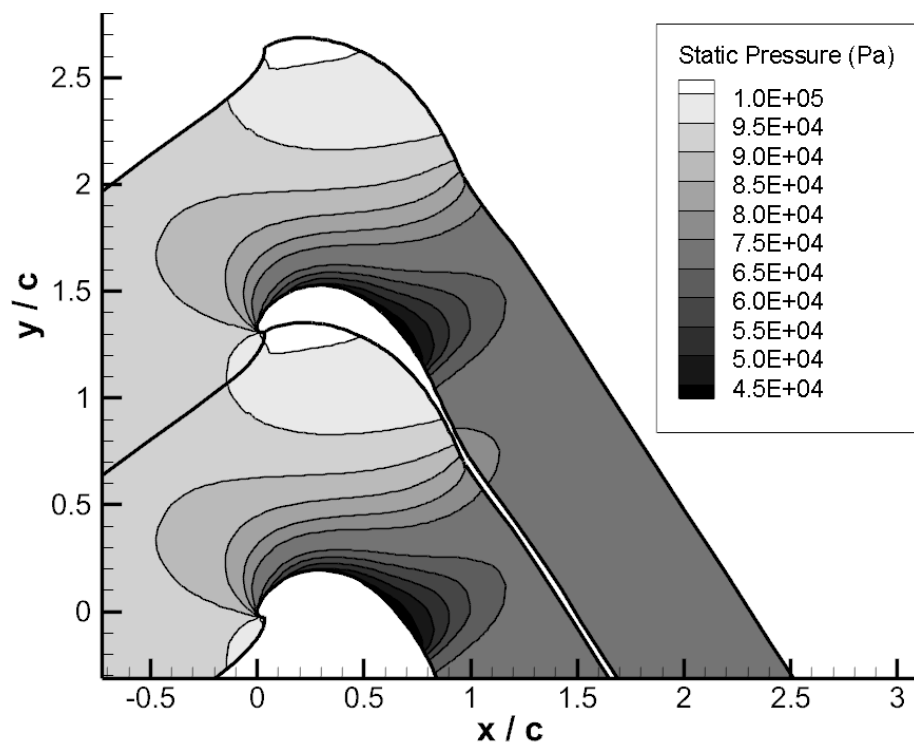


Figure 5.17b Isopressure lines, Newton solver

The results of the present methods which are shown in Figure 5.16 are matched well with the results of hodograph solution. There are two different points in the solution of the time marching method, which can easily be seen in Figure 5.16. The result of the time marching method deviates from the results of the hodograph solution between 73% and 76% of the axial chord. This is probably due to the quality of the mesh at this interval. The pressure ratios obtained from the time marching, Newton and hodograph method are 0.749, 0.752 and 0.748, respectively. In terms of pressure ratio, the time marching has better prediction as compared to that of Newton solver. The outlet flow angles obtained from the time marching, Newton and hodograph method are -56.97, -56.95 and -56.93 degrees, respectively. In terms of exit slope, Newton solver has better prediction as compared to that of time marching solver. The problem of trailing edge pressure jump mentioned in Section 5.2.7 is also seen in Figure 5.16 for this test case.

5.2.8. Sanz Supercritical Compressor Cascade

The Sanz compressor cascade is the last test case related to the flow through the cascade. The flow is entirely transonic within the blade passage. The trailing edge is almost closed. Again it should not be closed in order to carry out a valid comparison with the hodograph solution. The axial distance is nondimensionalized by the axial chord of the blade for the C_p versus axial distance plots. The cascade has an inlet Mach number of 0.711 together with an inlet flow angle of 30.81 degree. The outlet flow angle and outlet Mach number are -0.35 degree and 0.544, respectively. The pressure distribution over the blade surface obtained from the present solvers together with the result of hodograph solution is given in Figure 5.19.

The results of time marching and Newton solver agree well with the hodograph solution. The small differences in terms of surface static pressure between the results of three methods are seen around the 30% of the axial chord of the blade. These deviations are again due to the quality of the mesh on this region. If the mesh is refined around the 30% of the axial chord on the suction side, the present solution could be enhanced. The remaining parts of the pressure distribution obtained from the present solvers has excellent an agreement with the result of the hodograph method.

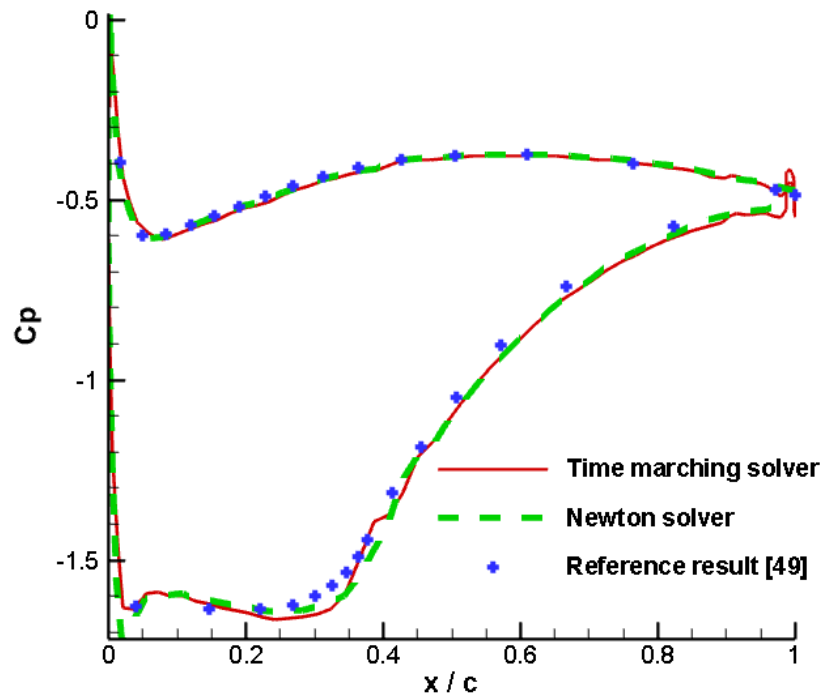


Figure 5.18 Variation of surface Mach number of Sanz supercritical compressor blade

The pressure ratios obtained from the time marching, Newton and hodograph method are 1.144, 1.144 and 1.145, respectively. In terms of pressure ratio, the time marching and Newton solvers have excellent prediction. The outlet flow angles obtained from the time marching, Newton and hodograph method are -0.21° , -0.20° and -0.35° degrees, respectively. The exit slope predictions are also quite accurate.

The Mach number contours obtained from time marching and Newton solvers are given in Figures 5.19a and 5.19b, respectively. The similar wave patterns in the flow fields are obtained from the present solvers. The problem regarding to the trailing edge nonuniformity for time marching method also appears for this test case.

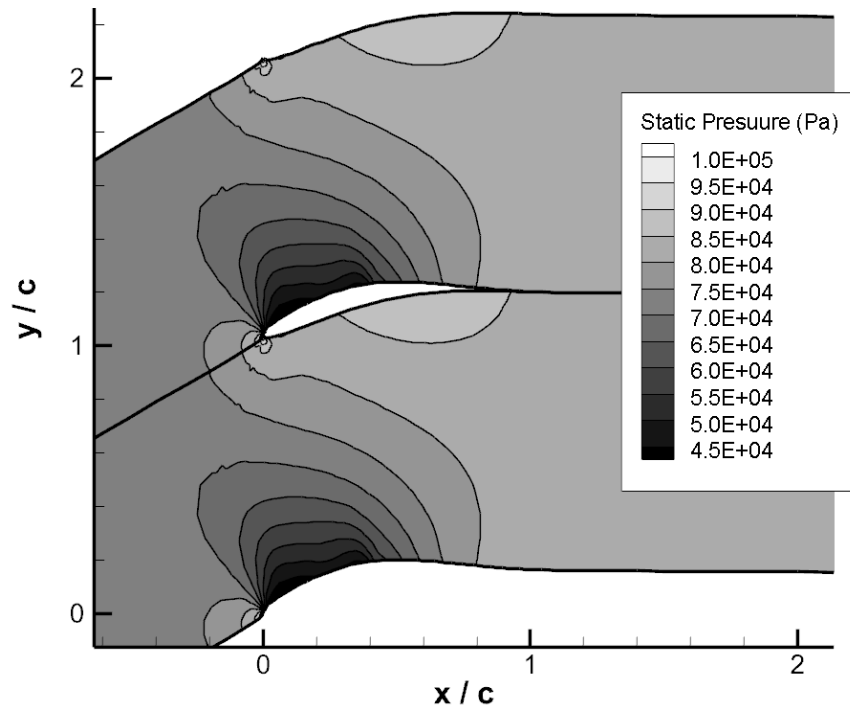


Figure 5.19a Isopressure lines, time marching solver

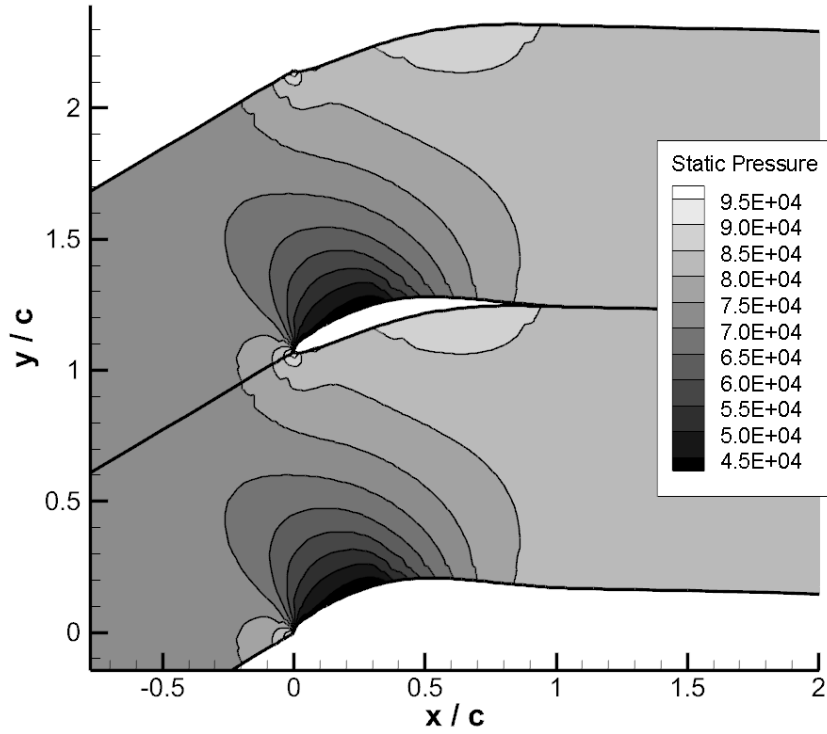


Figure 5.19b Isopressure lines, Newton solver

5.2.9. R030 Rotor

For the simulation of the flow within a rotor blade passage, single stage axial compressor available in reference [49] is selected. There are 8 streamline passing through the blade from hub to tip as shown in Figure 5.20.

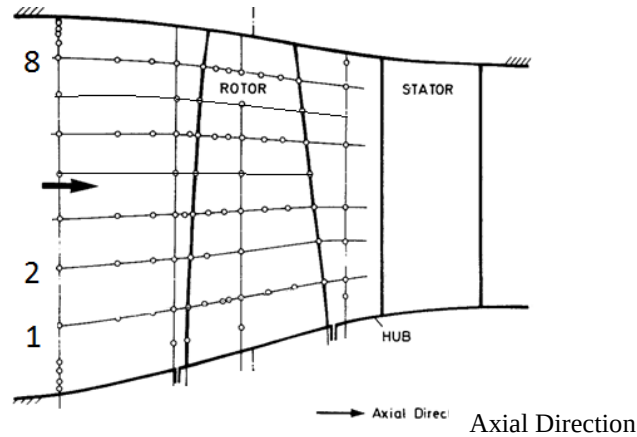


Figure 5.20 Throughflow calculation stations [49]

The second streamline which is at 18% of the blade height is selected for blade to blade calculation. The rotor has an inlet Mach number of 0.9215 at the specified location above. The flow angle and streamline curvature angle are 54.60 and 11.86 degrees, respectively. The outlet flow angle and Mach number are -8.0 degree and 0.62, respectively. The inlet flow regimes for the streamlines other than the first one are supersonic. Since the Newton solver is not capable of handling supersonic inlet, the 18% height is selected for blade to blade calculations. The flow is transonic and there is no shockwave within the passage. The rotational speed of the rotor is 20,260 rpm. The geometric parameters of the first streamline are available in reference [49]. The variations of surface static pressure are presented for the comparison of the results. The axial distance and surface static pressure are nondimensionalized by axial the chord of the blade and the inlet total pressure, respectively.

First of all, experimental results will be used for the comparison of the results. The pressure distributions obtained from the present codes are compared with the experimental result taken from reference [51] in Figure 5.21.

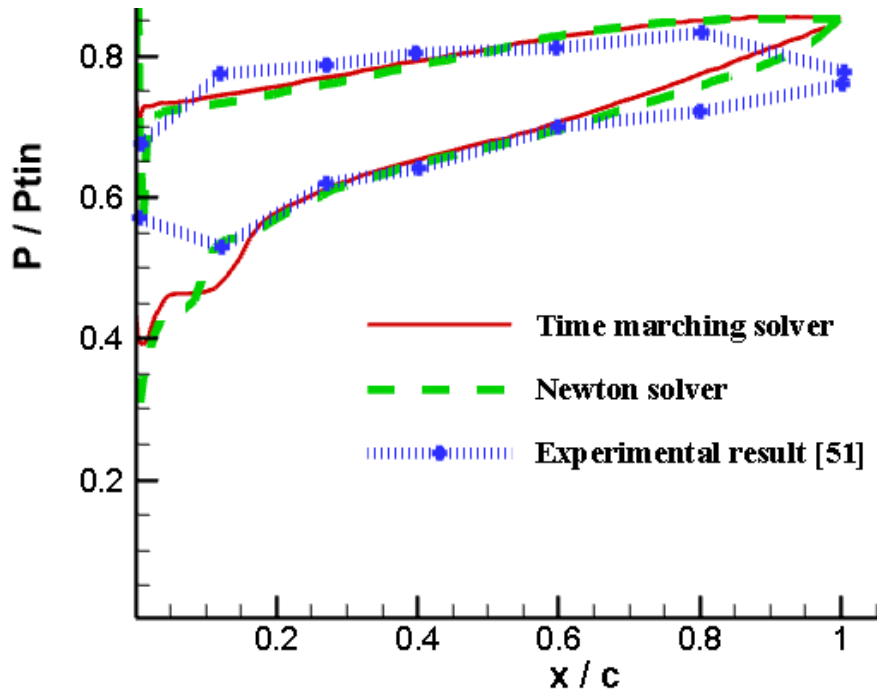


Figure 5.21 Rotor R030, comparison of the computational results with experimental results

The computational results of both newton solver and time marching solver deviate from the experimental results [51] as shown in Figures 5.21. This is expected because experimental results are viscous while both solvers are inviscid. In order to check whether the inviscid solvers developed in this thesis are accurate or not for the rotor solution, the results are also compared with inviscid computational result in reference [3].

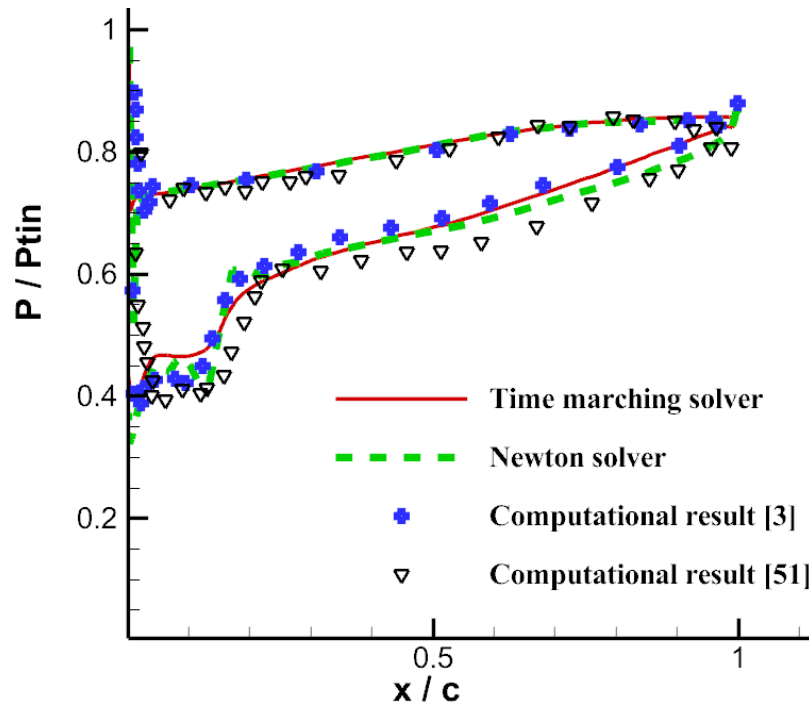


Figure 5.22 Rotor R030, comparison of the computational results with results in reference [3]

The results have shown that both of the solvers match well with the result of the computational results in reference [3] and [51]. Although the trend of the pressure distributions obtained from the present solvers are similar with that of the computational results in reference [3] and [51], there are deviations in some regions. At leading edge, the sudden increases in the static pressure are seen on the suction surface. In reference [51], it is stated that the single value from the data may be insufficient for comparison. The pressure behind the leading edge on suction surface may have sudden decrease. This may be possible because four computational results insist on sudden pressure decrease at leading edge. The pressure ratios obtained from the time marching, Newton and experiment are 1.443, 1.445 and 1.43, respectively. Both solvers give slightly higher pressure ratios as compared to experimental result. This is also expected since experiment result has the viscous effects. Due to viscous losses, the pressure rise through the compressor are always less than the pressure rise calculated by the inviscid solvers.

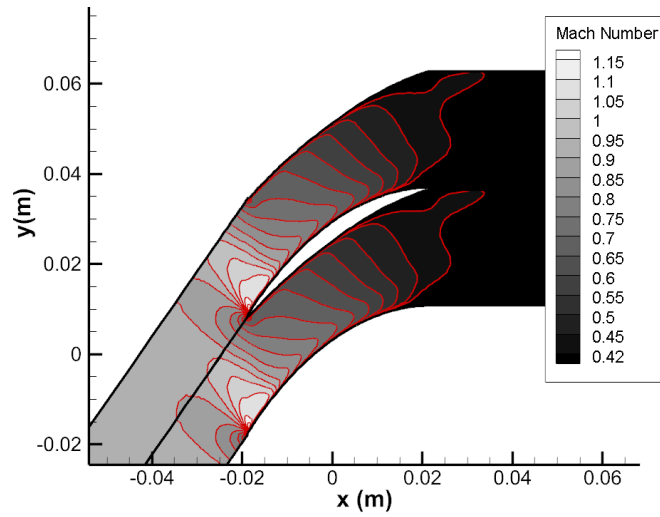


Figure 5.23a Mach number contours, time marching solver

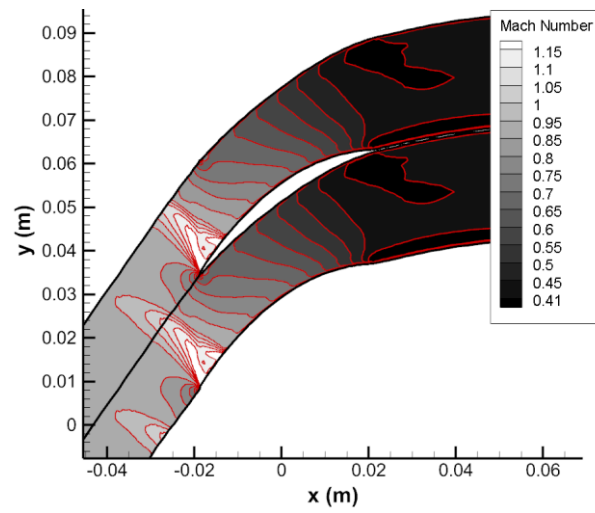


Figure 5.23b Mach number contours, Newton solver

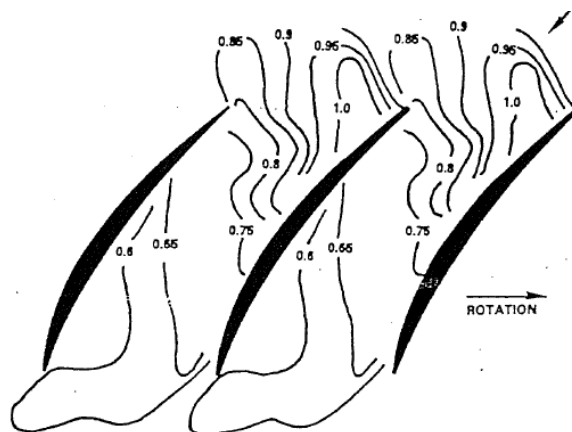


Figure 5.23c Mach number contours, experimental result

CHAPTER 6

CONCLUSION AND DISCUSSION

6.1. Conclusion

In order to develop appropriate design tools for blade to blade calculation, two different approaches are used in this thesis. In the literature, while the linear Euler solver is very popular for blade to blade analysis and design, the time marching finite volume approach is the most common among the solution methods for the Euler equations. Therefore, in order to see which one is better as compared to other, both method is studied in this thesis.

The Newton solver was first developed by the Giles [6] and Drela [34] in order to solve blade to blade flow. The method applies integral form of the Euler equations to each of the quadrilateral cells finite volume cells. Since the flow equation is of elliptic type, the physical boundary conditions are applied at the inlet and exit boundaries. The wall boundary condition is only related to the movement of the grid points. No pressure extrapolation from interior of the domain is required. The two sides of the cell are the streamline of the flow while the other two sides are the inlet and outlet of the computational cell. This formulation makes the method different than the common methods in the literature because the positions of the grids are determined during the calculation procedure. Actually, this approach was first applied by the streamline curvature method. Since the streamline curvature method solves differential form of the flow equation, it does not have conservative characteristic leading to the incorrect results in transonic flow problem with shock waves. Then, since the integral form of the Euler equations ensures the correct Rankine-Hugoniot shock jump relations, Giles [6] applied the integral form of the Euler equations on the intrinsic streamline grid and he obtained good results in transonic flow with shock waves.

The method actually developed for the steady subsonic flow. The transonic behavior is implemented with the concept of artificial compressibility. In order to verify this approach, Giles [6] conducted numerical experiments for a quasi-one dimensional flow. As a result of the series of experiments, even the sharp shocks could be captured. In addition to the current method, a second order correction was implemented in order to obtain second order accuracy [6]. The main advantage of the Newton solver is the convergence speed. It is so fast that the convergence is obtained in a second. 10^{-6} reduction in the residual is obtained in about 5-10 iterations for stator and duct case. For rotor cases, the relaxation factor is decreased two-folds of its' calculated value because the addition of the source terms can cause huge changes during the calculation. The speed of the method is mainly due to three reasons: the use of intrinsic streamline grid coordinates, reduction of the flow variables from four to two.

The time marching finite volume method is the most advanced and popular method for the Euler equations nowadays. The integral form of the continuity, momentum and energy equations are solved with various discretization schemes. The characteristic of the unsteady form of the Euler equations is always hyperbolic leading to marching of flow variables in time. Therefore, transonic flow can automatically is covered without any additional treatment like artificial compressibility. At the inlet and exit, characteristic boundary conditions are applied. On the wall, flow tangency and slip conditions are enforced. The variables are integrated in time using four stage Runge-Kutta time integration. In order to accelerate the solver, local time stepping is applied.

Basically, there are two type of finite volume schemes: cell centered and cell vertex. While the cell centered schemes store the flow variables at the centroid of the cells, cell vertex scheme stores at the grid points. The cell centered scheme has advantageous over the cell vertex scheme [6] and it has more straightforward implementation in a flow solver, so the cell centered approach is used in this thesis. There are various methods in order to calculate the fluxes at the cell faces. These methods can be grouped into two: central schemes with artificial viscosity (dissipation) and upwind schemes (characteristic based schemes). In terms of computational cost, the central schemes have advantageous over the upwind schemes but the upwind schemes can capture discontinuities much more accurately than central schemes. Since the accuracy of the

solver is more important as compared to the computational cost, upwind schemes are used in this thesis.

6.2. Comparison of Two Solvers

The results have clearly shown that both solvers have no superiority over each other in terms of accuracy. Both solver gives quite accurate results as compared to the experimental results and other validated computational results. The Newton solver enforces the Kutta condition at the trailing edge of the blade so there is no pressure jump at the trailing edge. On the other hand, in some solutions of the time marching solver, the pressure jump between the suction side and pressure side of the trailing edge may be seen. This is common problem of the time marching solver in the literature because of the absence of the trailing edge Kutta condition. However, this non-uniformity does not affect the final solution if fine mesh is used at the trailing edge. One method can seem to be more accurate compared to another for a specific problem but in general there is no distinct difference between the solvers in terms of accuracy.

The great distinction between two methods appears in terms of computational time. For moderate meshes (for example 86×17), the computational time for the Newton solver is about 1-2 seconds while that of the time marching solver is about one minutes. This comparison between the two solvers in terms of computational time is made for the same number of grid points. However, sometimes the number of grid points which is sufficient for the Newton solver may not be sufficient for the time marching solver. Therefore, the computational time of time marching solver for fine meshes may increase.

At this point, mesh dependency plays an important role for the comparison of the computational time of the solvers. For any case, the Newton solver can give quite accurate results with moderate (86×17) number of grid points. However, this sometimes does not work in time marching solver because it highly depends on the grid points. Especially for the transonic flow problem with shock wave, mesh should be fine enough to capture the shock accurately. Another example to the mesh

dependency of time marching solver can be given for the high turning adverse pressure gradient turbomachines (i.e. compressors). Since the Euler equation is actually inviscid, the separation cannot be seen on the surface of the blade. But in adverse pressure gradient flows with high turning angles the Euler gives some artificial results because of the direct consequence of the presence of the numerical dissipation. This is best explained in reference [11] for the case of flow over the cylinder. In order to overcome the problem, one should always attempt to refine the grid up to a level where numerical dissipation becomes sufficiently low resulting in minimization of the numerical entropy generation. When large number of grid point is required for time marching solver, the computational time required for steady state solution also increases. For example, in the case of Sanz subcritical compressor cascade and R030 rotor, the number of grid points used in the Newton solver is not sufficient for the time marching solver. More refined mesh is used for time marching solver as compared to Newton solver in order to obtain required accuracy. As a result of the mesh refinement, the computational time of the time marching solver is increased to about 3-4 minutes for the test cases mentioned above.

6.3. Future Work

Both the Newton solver and time marching solver requires additional studies. The main purpose of the blade to blade solver in a quasi-three dimensional design process is providing pressure distribution and loss curve as outputs. The losses include stagnation pressure losses due to shock waves and viscous losses due to friction on the body. The current solvers cannot handle viscous losses since both of them are inviscid solvers. For a full blade to blade solver, viscous solution capability must be implemented to both Newton and time marching solvers. For the Newton solver, the viscous treatment can only be given by the compressible integral boundary layer concept because the solver uses intrinsic streamline grid. Since the streamline cannot be defined for separated flow, the full Navier-Stokes cannot be solved by the intrinsic streamline approach. Integral boundary layer formulation interacts with the inviscid flow via displacement thickness. Edge velocity distribution over blade surfaces coming from the Euler solution is used in order to calculate the displacement thickness.

After displacement thickness is calculated the flow domain is updated by adding the displacement thickness on the physical boundary. Same approach can also be used for time marching solver. However, time marching method can also handle full Navier-Stokes equation with the implementation of the viscous fluxes with various turbulence modeling. This flexibility makes the time marching solver more attractive than the Newton solver. In the literature, the validity of the both approaches is made for compressible turbulent flow. However, an optimization between the computational time and accuracy should be considered. The accuracy of the Navier-Stokes can be more accurate than the accuracy of the Euler/Boundary layer solver but computational time of the Navier-Stokes solver will be probably much larger than the computational time of Euler/boundary layer solver.

The Newton solver cannot handle flow with supersonic inlet so there is one more additional improvement for the Newton-Solver other than implementation of the integral compressible boundary layer coupling. At the inlet with supersonic flow, Prandtl-Meyer flow angle is related to the inlet flow angle. The treatment of the supersonic inlet changes depending on the shock position at the inlet (attached or detached). On the other hand, since the time marching solver developed in this study has characteristic boundary conditions, it can handle supersonic inlet.

The only acceleration technique which is used in the time marching solver is the local time stepping in this thesis. However, the computational time still long as compared to the 2-D Euler solvers available in the literature. Therefore, more acceleration techniques need to be implemented to current time marching solver. The first improvement can be made in the temporal discretization by changing the solver from explicit to implicit. The stability criteria in the explicit time stepping restrict the time step, so the number of iterations for convergence increases. Implicit time integration gives the possibility of using larger time step resulting in rapid convergence rate as compared to the explicit scheme. The most important acceleration technique which is widely used in the literature is the multigrid method. The calculation is carried out on a fine grid and then transferred to coarser grids. After the coarsest grid is reached, the solution corrections are interpolated to the initial fine grid. The number of iterations with multigrid acceleration is decreased approximately 20 folds as compared to that

of the solver without multigrid acceleration. The other acceleration techniques widely used in the literature are the enthalpy damping, residual smoothing and preconditioning. All the acceleration techniques except the residual smoothing can be used for both explicit and implicit time stepping since residual smoothing was developed for the explicit schemes.

The most important advantage of the Newton's method appears in inverse design. Giles et.al [6] has stated that for the transonic flow, the inverse problem solution becomes extremely difficult. However, this method facilitates the process since only the specification of the pressure distribution on the blade surface is sufficient to determine the blade shape. Since the wall boundary condition in the Newton solver is given in terms of nodal displacement only, the nodes on the wall can be moved just like interior nodes. Therefore by changing the boundary conditions from hard wall (stationary wall grid points) to free wall (moving wall grid points), the blade shape satisfying the pressure ratio can be easily determined. Also, the solution of the viscous problem can easily be handled by this method. The integral momentum equation for compressible turbulent flow is developed and implemented to this method by Drela [34]. The application of the inverse problem and boundary layer coupling are not included in this thesis because they are the future work of this study.

In conclusion, acceleration techniques should be implemented to the time marching solver. If both codes are to be used for blade to blade analysis, the viscous treatment must be implemented to both of the solvers. Additionally, some of the compressors have supersonic flow at the inlet. The Newton solver cannot handle the blade to blade flow with supersonic inlet. If the turbomachinery with supersonic inlet needs to be solved, the supersonic inlet boundary condition must be implemented to the Newton solver.

REFERENCES

- [1] AGARD Lecture Series No.167 “Blading Design for Axial Turbomachines”, AGARD LS 167 (1989).
- [2] C.H. Wu, “A General Theory of Three Dimensional Flow in Subsonic and Supersonic Turbomachines of Axial, Radial and Mixed Flow Types”, Transactions of ASME, 1363 – 1378, (November 1952).
- [3] Uzol O., “Quasi-Three-Dimensional Numerical Design of Turbomachinery Blades”, M.Sc. Thesis, Department of Aeronautical Engineering, Middle East Technical University, July 1995.
- [4] J.D. Denton, W.N. Dawes., Computational Fluid Dynamics for Turbomachinery design. Proceedings of the Institution of Mechanical Engineers, 213,1999.
- [5] Murman, E.M., Cole, J.D., “Calculation of Plane Steady Transonic Flow”, AIAA Journal, Vol.9, pp.114-121, January 1971.
- [6] Giles M.B., “Newton Solution of Steady Two-Dimensional Transonic Flow”, Ph.D. Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, June 1985.
- [7] Childs, R.E., Pulliam, T.H., “A Newton Multigrid Method for the Euler Equations”, AIAA84-0430, Presented at AIAA 22nd Aerospace Sciences Meeting, January 9-12, 1984.
- [8] Mulder, W.A., Van Leer, B., “Implicit Upwind Methods for the Euler Equations”, AIAA-83-1930, Presented at the AIAA 6th Computational Fluid Dynamics Conference, at Danvers, Massachusetts, July 13-15, 1983.
- [9] Novak, R.A., “Streamline Curvature Computing Procedures for Fluid Flow Problems,” ASME Paper no.66-WA/GT-3, Presented at the Winter Annual Meeting of the American Society of Mechanical Engineers, November 1966.

- [10] Thomas, L.H. (1949)., Elliptic Problems in Linear Differential Equations over a Network, Watson Sci. Comput. Lab Report, Columbia University, New York.
- [11] Hirsh, C., “Numerical Computation of Internal and External Flows- Volume 1&2”, John Wiley & Sons, 1988.
- [12] Blazek, J., “Computational Fluid Dynamics: Principles and Applications”, 1st edition, Elsevier, 2001.
- [13] Steger J.L., Warming, R.F.: Flux Vector Splitting of the Inviscid Gas Dynamic Equations with Application to Finite Difference Methods. J. Computational Physics, 40 (1981), pp. 263-293.
- [14] Liou, M.-S., Steffen, C.J. Jr.: A New Flux Splitting Scheme. NASA TM-104404, 1991; also J. Computational Physics, 107 (1993), pp. 23-39.
- [15] Jameson, A.: Positive Schemes and Shock Modelling for Compressible Flow. Int. J. Numerical Methods in Fluids, 20 (1995), pp. 743-776.
- [16] Godunov, S.K., A Difference Scheme for Numerical Computation Discontinuous Solution of Hydrodynamic Equations. Math. Sbornik (in Russian), 47 (1959), pp. 271-306; translated US Joint Publ. Res. Service, JPRS 7226, 1969.
- [17] Roe, P.L., Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. J. Computational Physics, 43 (1981), pp. 357-372.
- [18] Osher, S.; Solomon, F.: Upwind Difference Schemes for Hyperbolic Systems of Conservation Laws. Math. Comp., 38 (1982), pp. 339-374.
- [19] Harten, A., High Resolution Schemes for Hyperbolic Conservation Laws. J. Computational Physics, 49 (1983), pp. 357-393.
- [20] Struijs, R., Roe, P.L.; Deconinck, H.: Fluctuation Splitting Schemes for the 2D Euler Equations. VKI Lecture Series 1991-01, 1991.

- [21] Lax, P. D., and Wendroff, B., “Systems of Conservation Laws”, Comm. 132 Pure and Applied Mathematics, vol. 13, pp. 217-37, 1960
- [22] MacCormack, R. W., “The effect of viscosity in hypervelocity impact cratering”, AIAA-66-354, 1966.
- [23] Ni, R. H., “A Multiple-Grid Scheme for Solving the Euler Equations”, AIAA 137 Journal, Vol. 20, pp. 1565-1571, 1982.
- [24] Aksel, M. H., “ME 485 Computational Fluid Dynamics Using Finite Volume Method Lecture notes”, Mechanical Engineering Department, METU, 2013
- [25] Levent K., “Development of an Axisymmetric Euler Solver Using Finite Volume Method for Internal and External Flows”, M.Sc. Thesis, Department of Mechanical Engineering, Middle East Technical University, July 1999.
- [26] Sabanca M., “Development of a two-dimensional euler solver using finite volume method for external flows”, M.Sc. Thesis, Department of Mechanical Engineering, Middle East Technical University, July 1997.
- [27] Telçeker., N, Mutaş., Ö., “One Dimensional Flow Analysis by Using the Euler Equations ”, TÜBİTAK-SAGE, Ankara 1992.
- [28] Versteeg, H.K., Malalasekera W., “An Introduction to Computational Fluid Dynamics - The Finite Volume Method”, Longman, 1995
- [29] Braembussche R.A.V Von Karman Lecture Notes No.172 “Blade to Blade Flow in Turbomachines”, (2005).
- [30] Ekici K., “Quasi Three Dimensional Analysis and Design of Turbomachinery Blades with Boundary Layer Coupling”, M.Sc. Thesis, Department of Aeronautical Engineering, Middle East Technical University, June 1997.
- [31] Drela, M., “A User’s Guide to MISES 2.63” February 2008

- [32] Öksüz Ö., “Aerodynamic Optimization of Turbine Cascades Using an Euler/Boundary Layer Solver Coupled Genetic Algorithm”, M.Sc. Thesis, Department of Aeronautical Engineering, Middle East Technical University, June 2002.
- [33] Bercan S., “A Two Dimensional Euler Flow Solver on Adaptive Cartesian Grids”, M.Sc. Thesis, Department of Aeronautical Engineering, Middle East Technical University, June 2008.
- [34] Drela M., “Two Dimensional Transonic Aerodynamics Design and Analysis Using the Euler Equations”, Ph.D. Thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, December 1985.
- [35] Çetinkaya T.A., “2-D and Axisymmetric Euler Solver for Aerodynamic Analysis and Design”, Ph.D. Thesis, Department of Mechanical Engineering, Middle East Technical University, June 1990.
- [36] Çetinkaya, T. A., “ME 517 Advanced Fluid Dynamics”, Mechanical Engineering Department, METU, 2014
- [37] Yiqing, S., Huerta, M.A., “Rotated Hybrid Low Diffusion ECUSP-HLL scheme and Its Applications to Hypersonic Flows”, AIAA Journal, 27-30 June 2011
- [38] Nishikawa, H., Kitamura, K., “Very simple, carbuncle-free, boundary-layer-resolving, rotated-hybrid Riemann solvers”, Journal of Computational Physics 227 (2008) 2560–2581
- [39] Pandolfi, m., Ambrossio, D., “Numerical Instabilities in Upwind Methods: Analysis and Cures for the “Carbuncle” Phenomenon”, Journal of Computational Physics 166, 271–301 (2001)
- [40] Volker, E., “The Carbuncle Phenomenon Is Incurable”, Acta Mathematica Scientia 2009,29B(6):1647–1656
- [41] Masatsuka, K. Favorite Codes and Subroutines. Retrieved Feb 4, 2015, from [http:// www.cfdbooks.com/](http://www.cfdbooks.com/)

- [42] Kırkköprü K., Uygun M., “Numerical Solution of the Euler Equations by Finite Volume Methods: Central Versus Upwind Schemes”, Journal of Aeronautics and Space Technologies January 2005 Volume 2 Number 1 (47-55)
- [43] Venkatakrishnan, V., “On the Accuracy of Limiters and Convergence to Steady State solutions”, Presented at AIAA 22nd Aerospace Sciences Meeting, January 11-14, 1993.
- [44] Thompson, K.W., “Time Dependent Boundary Conditions for Hyperbolic Systems”, J. Computational Physics”, 68 (1987), pp. 1-24.
- [45] Chapra, C. S., Canale, R. P., “Numerical Methods for Engineers”, Engineering Education System, McGraw-Hill, 2009
- [46] Giles, M.B., Non-Reflecting Boundary Conditions for Euler Equation Calculations.
AIAA Paper 89-1942, 1989.
- [47] Eidelman, S., Colela, P., Shreeve, R.P., “Aplication of the Godunov Method and its Second order Extension to Cascade Flow Modelling”, AIAA Journal, Vol 22 1984
- [48] Erol Ö., “Development of an Euler Solver for compressible fluids using adaptive grids”, M.Sc. Thesis, Department of Mechanical Engineering, Middle East Technical University, January 2002.
- [49] AGARD, Working Group 18, “Test Cases for Computation of Internal Flows in Aero Engine Components” AGARD AR 275, (1990)
- [50] Jameson, A., Liu, F., “Multigrid Calculations for Cascades”, Princeton University, Princeton, NJ 08544
- [51] R.J. Dunker, “Test Case E/CO-4 Single Transonic Compressor Stage”, AGARD AR-275,245-285, 1990
- [52] Dixon, S. L., “Fluid Mechanics, Thermodynamics of Turbomachinery”, Pergamon Press, 1982

- [53] Hoffmann, K. A., Chiang, S. T., “Computational Fluid Dynamics”, Engineering Education System, 2000
- [54] Aksel, M. H., Eralp, O. C., “Gas Dynamics”, Prentice Hall, 1997