COUPLED ANALYSIS OF TURBOMACHINERY BLADES

by

Bülent Düz

B.S. in M.E., Yıldız Technical University, 2005

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Mechanical Engineering

Boğaziçi University

2008

ACKNOWLEDGEMENTS

I would like to give special thanks to my advisor Assist. Prof. Ali Ecder for his helpful suggestions and insightful comments. He graciously donated his time to share his invaluable ideas and perfect guidance during this study. I would like to give sincere thanks to my co-advisor Assoc. Prof. Dr. Fazıl Önder Sönmez for his help and support.

Special thanks are addressed to my examiners Assoc. Prof. Dr. Emre Aksan, Assoc. Prof. Dr. Hakan Ertürk, and Assoc. Prof. Dr. Fatih Ecevit.

I wish to acknowledge the valuable assistance and continuous support of FMS laboratory members Erhan Turan, Yalın Kaptan, Özkan Aydın and Altuğ Melik Başol.

Last but not least I owe a debt of gratitude to my family for their constant source of encouragement and support during my entire education.

This study was supported by the Turkish National Science Foundation (TÜBİTAK).

ABSTRACT

COUPLED ANALYSIS OF TURBOMACHINERY BLADES

Three-dimensional analysis of fluid flow over turbomachinery blades and the deformation of the blades under aerodynamic loads are modeled and simulated using computational techniques. Decomposition into computational sub-domains and using multi-level partitioning hierarchy characterize the procedure of solving the non-linear set of governing partial differential equations. Grid generation techniques and various formulations of Navier-Stokes equations are studied. These equations are linearized using Newton's method, and the resulting system of equations are solved using matrix-free implementations of the preconditioned Krylov techniques.

Pressure values determined as a result of fluid analysis are implemented as boundary conditions into ANSYS environment. Displacements are found after solid analysis performed by ANSYS and transferred into fluid analysis to update the geometry. The computations were carried out for different Reynolds numbers and the results of the numerical simulations are discussed and compared.

ÖZET

TURBOMAKİNA KANATLARININ BAĞLAŞIMLI ANALİZİ

Turbomakina kanatları arasındaki akışın üç boyutlu analizi ve kanatların aerodinamik yükler altındaki deformasyonu nümerik teknikler kullanılarak modellenmiş ve sunulmuştur. Farklı ağ oluşturma teknikleri ve Navier-Stokes denklemlerinin farklı formülasyonları üzerinde çalışılmıştır. Bu denklemler Newton yöntemiyle doğrusal hale getirilmiş ve sonunda açığa çıkan doğrusal sistem iyileştirilmiş Krylov çözücülerinin matrissiz uygulamalarıyla çözülmüştür. Farklı Reynolds sayılarının akış ve basıncı nasıl etkilediği gözlemlenmiş ve sonuçlar karşılaştırılmıştır.

Akış analizinden elde edilen sonuçlar turbomakina kanatları üzerindeki basınç dağılımını elde etmede kullanılmıştır. Basınç değerleri ANSYS programında kanat üzerine uygulanmış ve kanatın deformasyonu izlenerek kaydedilmiştir. Daha sonra ANSYS programında elde edilen şekil değiştirme vektörleri akış analizini yapmak için geliştirilen koda aktarılmış ve problemin çözüm alanı güncellenmiştir.

Newton yöntemi farklı Krylov çözücüleriyle birleştirilerek kullanılmıştır. Ayrıca Jacobi, simetrik Gauss-Seidel (SGS) ve tamamlanmamış matris ayrıştırma (ILU(l)) iyileştiricileri kullanılarak bunların Krylov yöntemlerininin çözüme ulaşma performansları üzerindeki etkileri incelenmişir. ILU(l) iyileştiricisinin diğerlerine göre daha etkili olduğu gözlenmiştir. Ancak bu iyileştiricinin matrissiz metodlara uygulanamamasından SGS iyileştiricili yaklaşık Newton yöntemi hem düşük hafıza kullanımıyla hem de kod yazım süresinin kısalığıyla daha uygun bir çözücü olarak gözükmektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

$A(x)$	Coefficient matrix
$A(x)^{-1}$	Inverse of Coefficient matrix
$b(x)$	Right hand side of a non-linear system
D	Diagonal of A
-E	The strict lower part of A
-F	The strict upper part of A
f(x)	Non-linear residual
I	Identity matrix
$J(x)$	Jacobian matrix
k	Iteration count
M	Preconditioner matrix
M_{left}	Left preconditioning matrix
M_{right}	Right preconditioning matrix
M^{-1}	Inverse of the preconditioner matrix
Re	Reynolds Number
x	Unknown vector
P	Pressure
d	Displacement
x_0	Initial guess for unknown vector
U_0	Inlet velocity
u	Velocity vector in the x-direction
v	Velocity vector in the y-direction
w	Velocity vector in the z-direction
μ	Dynamic viscosity
ν	Kinematic viscosity
ρ	Fluid density
Ψ	Stream function
Ω	Vorticity

Ω_x	x component of vorticity vector
Ω_y	y component of vorticity vector
Ω_z	z component of vorticity vector

DDM	Domain Decomposition Method
PDE	Partial Differential Equation
TFI	Transfinite Interpolation
FSI	Fluid-Structure Interaction
BiCGSTAB	Bi-conjugate Gradient Stabilized
GMRES	Generalized Minimal Residual
SGS	Symmetric Gauss-Seidel
ILU	Incomplete LU Decomposition

1. INTRODUCTION

Pumps and turbines (sometimes called fluid machines) are used in a wide variety of areas. In general, pumps add energy to the fluid -they do work on the fluid; turbines extract energy from the fluid -the fluid does work on them. The term pump will be used to generically refer to all pumping machines, including pumps, fans, blowers, and compressors. Fluid machines can be divided into two main categories: positive displacement machines (denoted as the static type) and turbomachines (denoted as the dynamic type).

Positive displacement machines force a fluid into or out of a chamber by changing the volume of the chamber. The pressures developed and the work done are a result of essentially static forces rather than dynamic effects. Typical examples shown in Fig. 1.1 include the common tire pump used to fill bicycle tires, the human heart, and the gear pump. In these cases the device does work on the fluid (the container wall moves against the fluid pressure force on the moving wall). The internal combustion engine in a car is a positive displacement machine in which the fluid does work on the machine, the opposite of what happens in a pump. In the car engine the piston moves in the direction of the fluid pressure force acting on the piston face during the power stroke. Turbomachines, on the other hand, involve a collection of blades, buckets, flow channels, or passages arranged around an axis of rotation to form a rotor. Rotation of the rotor produces dynamic effects that either add energy to the fluid or remove energy from the fluid. Energy is either supplied to the rotating shaft (by a motor, for example) and transferred to the fluid by the blades (a pump), or the energy is transferred from the fluid to the blades and made available at the rotating shaft as shaft power (a turbine). Briefly, A group of blades moving with or against a lift force is the essence of a turbomachine.

The fluid used can be either a gas (as with a window fan or a gas turbine engine) or a liquid (as with the water pump on a car or a turbine at a hydroelectric power plant). While the basic operating principles are the same whether the fluid is a liquid

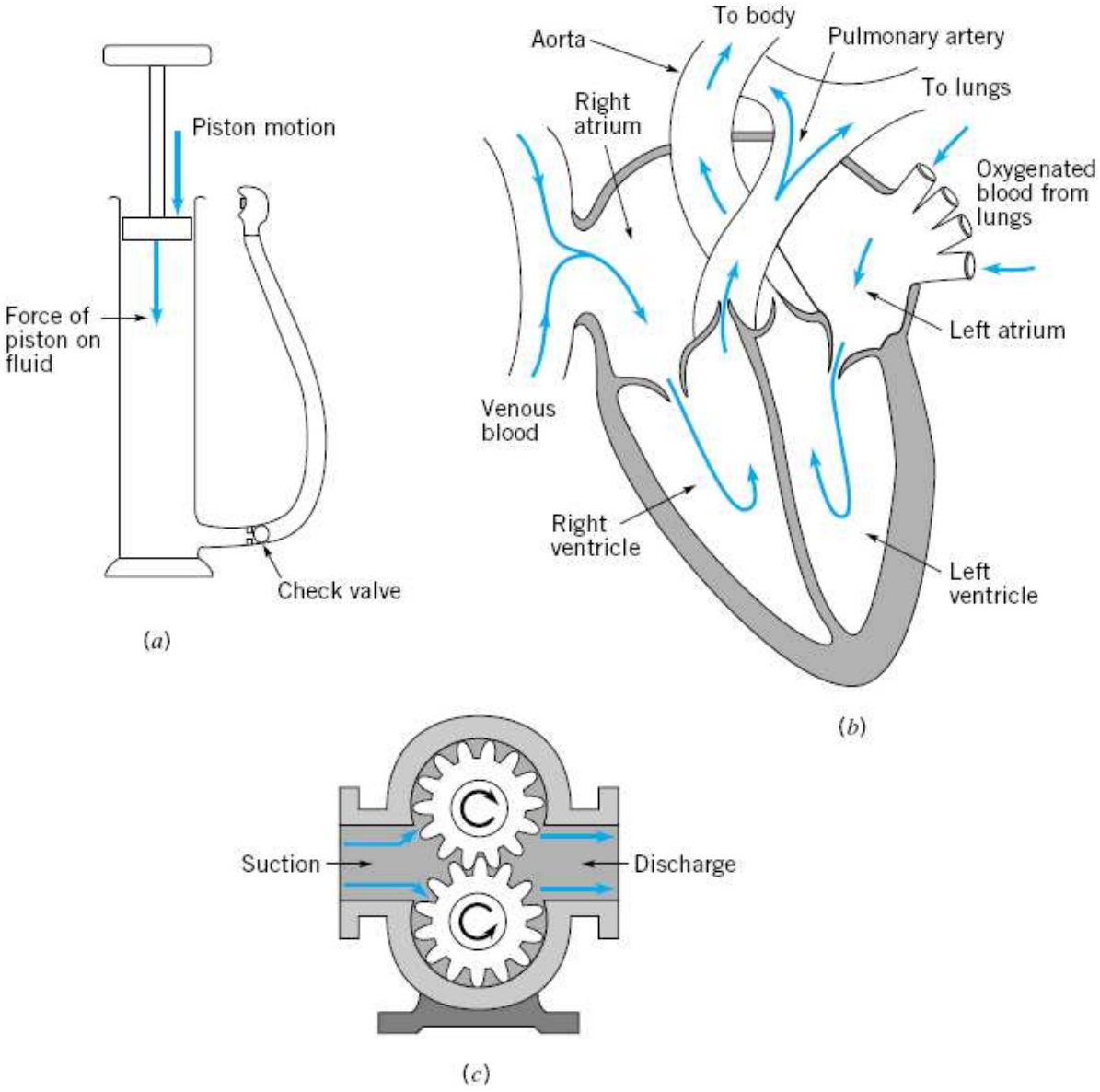


Figure 1.1. Typical Positive Displacement Pumps: (a) tire pump, (b) human heart, (c) gear pump [2]

or a gas, important differences in the fluid dynamics involved can occur. For example, cavitation may be an important design consideration when liquids are involved if the pressure at any point within the flow is reduced to the vapor pressure. Compressibility effects may be important when gases are involved if the Mach number becomes large enough.

Turbomachines are classified as axial-flow, mixed-flow, or radial-flow machines depending on the predominant direction of the fluid motion relative to the rotors axis as the fluid passes the blades (see Fig. 1.2). For an axial-flow machine the fluid maintains a significant axial-flow direction component from the inlet to outlet of the rotor. For a radial-flow machine the flow across the blades involves a substantial radial-flow component at the rotor inlet, exit, or both. In other machines, designated as mixed-flow machines, there may be significant radial- and axial-flow velocity components for the flow through the rotor row. Each type of machine has advantages and disadvantages for different applications and in terms of fluid-mechanical performance. Examples of

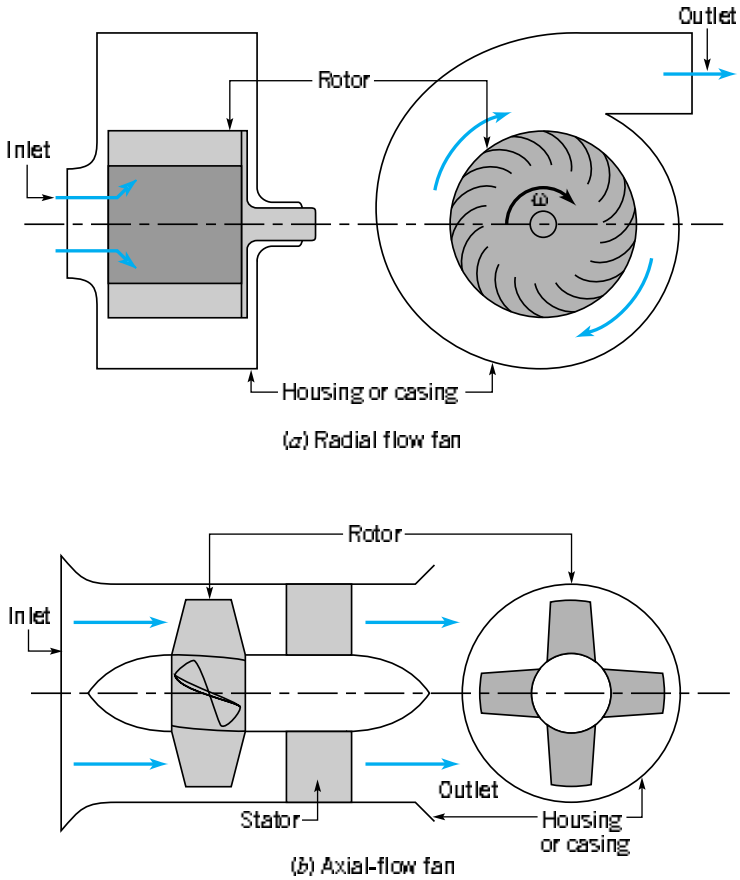


Figure 1.2. (a) A radial-flow turbomachine, (b) An Axial-flow turbomachine

turbomachine-type pumps include simple window fans, propellers on ships or airplanes, squirrel-cage fans on home furnaces, axial-flow water pumps used in deep wells, and compressors in automobile turbochargers. Examples of turbines include the turbine portion of gas turbine engines on aircraft, steam turbines used to drive generators at electrical generation stations, and the small, high-speed air turbines that power dentist drills.

Turbomachines serve in an enormous array of applications in our daily lives and thus play an important role in modern society. These machines can have a high power density (large power output per size), relatively few moving parts, and reasonable efficiency.

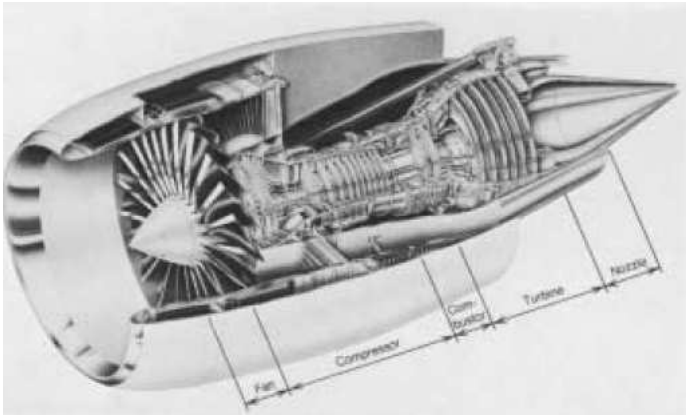


Figure 1.3. GE 90 propulsion system [2]

The conversion of total energy into shaft work or vice versa, can also be established with simple reciprocating (piston-cylinder) engines. Why should a turbomachine be applied? The answer to this question involves the limitation of power and mass flow associated with the reciprocating engines. The reciprocating engine, which works entirely on the displacement principle, is not able to transfer large amount of mass flow or mechanical energy. The largest operating Diesel engine has a power output of about 20 MW, whereas a large steam power plant may produce up to 1300 MW [2]. Unlike the reciprocating engines the working principle of a turbomachine is based on exchange of momentum between the blading and the working fluid.

Although this is by no means intended as a comprehensive review, it is nec-

essary to highlight a few important studies. A considerable amount of work has been performed on aerodynamic analysis of turbomachinery blades by using the direct and/or the inverse approach. For example, Denton [5] has numerically solved the three-dimensional Euler equations of motion and Dawes [6] and Hah et al. [14] have solved the 3D Navier-Stokes equations. Such methods are of substantial value to the designer, who can use them to investigate the flow conditions along vanes and blades. Xu and Chen [15] investigated sweep effects in a three-dimensional modern airfoil design. In addition to these studies, Carstens, Kemme and Schmitt [16] presented a technique which analyses the flutter behavior of turbomachinery bladings in the time domain. Moffatt and He [17] developed fully-coupled methods for blade forced response prediction based on the nonlinear harmonic method. Another fluid-structure interaction analysis was performed by Kamakoti and Shyy [18]. They used AGARD 445.6 wing to illustrate their problem and determined the flutter boundary for the AGARD wing geometry.

In this thesis, three-dimensional analysis of fluid flow over turbomachinery blades and the deformation of the blades under aerodynamic loads are modeled and simulated using computational techniques. Decomposition into computational sub-domains and using multi-level partitioning hierarchy characterize the procedure of solving the nonlinear set of governing partial differential equations. Grid generation techniques and various formulations of Navier-Stokes equations are studied. These equations are linearized using Newton's method, and the resulting system of equations are solved using matrix-free implementations of the preconditioned Krylov techniques. Pressure values determined as a result of fluid analysis are implemented as boundary conditions into ANSYS environment. Displacements are found after solid analysis performed by ANSYS and transferred into fluid analysis to update the geometry. The computations were carried out for different Reynolds numbers and the results of the numerical simulations are discussed and compared.

2. NUMERICAL METHODS

2.1. Components of the Numerical Solution Method

The starting point of any numerical method is the mathematical model, i.e. the set of partial differential equations and boundary conditions. After choosing an appropriate model for the target application (incompressible, inviscid, turbulent; two- or three-dimensional, etc.), a solution method is designed for the simplified set of equations. More detailed information about the mathematical modeling of this study is presented in Chapter 3.

After selecting the mathematical model, we have to choose a suitable discretization method,i.e. a method of approximating the differential equations by a system of algebraic equations for the variables at some set of discrete locations in space and time. There are many approaches, but the most important of which are: finite difference, finite volume and finite element methods. Each type of method yields the same solution if the grid is very fine. However, some methods are more suitable to some classes of problems than others. The preference is often determined by the attitude of the developer. In this study finite difference approach is used.

2.1.1. Solution Methods

Discretization yields a large system of non-linear algebraic equations. The method of solution depends on the problem. For unsteady flows, methods based on those used for initial value problems for ordinary differential equations (marching in time) are used. At each time step an elliptic problem has to be solved. Steady flow problems are usually solved by pseudo-time marching or an equivalent iteration scheme. Since the equations are non-linear, an iteration scheme is used to solve them. These methods are successive linearization of the equations and the resulting linear systems are almost always solved by iterative techniques. In this study, the governing equations are linearized using Newton's method, and the resulting system of equations are solved using matrix-free

implementations of the preconditioned Krylov techniques.

2.1.1.1. Newton's Method. The discretization of the non-linear differential equations results in nonlinear system of equations. In order to solve them using linear system solvers they have to be linearized. Newton's method is one of the most widely used methods in this manner.

Newton's algorithm is given below. Here f is a vector composed of the nonlinear equations aroused from the discretization of the governing equations with respect to the nodes in the computational domain and J is the Jacobian matrix composed of the partial derivatives of the equations with respect to all unknowns. And k indicates the number of the Newton step.

1. Guess a solution vector
2. Solve the linearized equation system below for Δx^{k+1}

$$J(x^k)\Delta x^{k+1} = -f(x^k) \tag{2.1}$$

3. $x^{k+1} = x^k + \Delta x^{k+1}$
4. Continue until norm of the f vector drops below the tolerance

$$J_{i,j} = \frac{\partial f_i^k}{\partial x_j^k} \tag{2.2}$$

To solve the linearized system of equations linear system solvers must be used. A broad range of solvers of this kind can be found in the literature. In this work only Krylov sub-space solvers are used. Brief information about them will be given in the next section. All these solvers require Jacobian matrix vector multiplications. And Newton's method can be grouped into two main categories with respect to the computation of this multiplication. In the exact Newton's method the Jacobian matrix

is written analytically in the code, stored in the memory and multiplied with the desired vector when necessary. The main drawback of the exact Newton approach is the large memory requirements for storage of the Jacobian matrix. This generally limits the use of such methods to small-scale problems unless a large memory machine is available. Moreover, multiplications with vectors are also time consuming due to the large sizes of the Jacobian matrices. However, preconditioners can be easily applied in this method due to the storage of the Jacobian matrix in the memory.

Second approach is called inexact Newton's method. In this method the matrix vector multiplications are carried out using the f vector and the directional differencing technique without computing the Jacobian matrix analytically and storing it in the memory. This approach reduces the memory requirement considerably and also speeds up the matrix vector multiplications by a great extent. However, preconditioners which require the Jacobian matrix explicitly cannot be applied in this method.

$$Jv = \frac{f(x + \epsilon v) - f(x)}{\epsilon} \tag{2.3}$$

The selection of the value of ϵ has a crucial importance. There are several formula for the computation of ϵ. In the present double-precision computations, an effective choice for ϵ was found to be

$$\epsilon = \sigma^{1/2} / \|x\| \tag{2.4}$$

when $\|x\| \neq 0$, and if $\|x\| = 0$, the result of the matrix vector product is set identically to zero. Here, σ is taken to be 10^{-14} [11].

There are also other methods for dealing the non-linear equation systems with. Although Newton's method is one of the most widely used methods it has also some drawbacks beside the advantages it has.

Advantages:

- Quadratically convergent from good starting guesses if J is non-singular.
- Exact solution in one iteration for an affine f (exact at each iteration for affine component functions of f)

Disadvantages:

- Not globally convergent for many problems.
- Requires J at each iteration.
- Each iteration requires the solution of a system of linear equations that may be singular or ill conditioned.

2.1.1.2. Krylov Sub-Space Solvers. Frequently iterative methods are preferred in CFD applications due to the surplus of the arithmetic operations in the direct methods such as Gaussian elimination, which makes them too costly by means of computation time. The term iterative method refers to a wide range of techniques that use successive approximations to obtain more accurate solutions to a linear system at each step. There are two types of iterative methods. Stationary methods such as Jacobi, Gauss-Seidel, Successive over-relaxation (SOR) are older, simpler to understand and to implement but usually not as effective due to their slow convergence behavior[12].

On the other hand, non-stationary methods are a relatively recent development, their analysis is usually harder to understand but they can be highly effective. The non-stationary methods presented here are based on the idea of sequences of orthogonal vectors. Some of the well-known methods are Conjugate Gradient (CG), Generalized Minimal Residual (GMRES), Bi-Conjugate Gradient (BiCG), Bi-Conjugate Gradient Stabilized (BiCGSTAB), Quasi Minimal Residual (QMR), Conjugate Gradient Squared (CGS) Method. These methods are called Krylov sub-space methods, because they project the original set of equations onto a so called Krylov sub-space. Each of the Krylov sub-space solvers have their own characteristics. Choosing the most appropriate solver according to the type of the problem enables faster convergence. On the other hand, in some cases choosing an inappropriate method may even lead to divergence.

For instance, CGS can be applied to non-symmetric matrices but CG can not.

In this work BiCGSTAB and GMRES methods are used. These are among the fastest and robust iterative solvers for a wide variety of applications. All of them are applicable to the non-symmetric matrices which is the case in the thesis problem. A brief summary of the properties of these methods is given below [12].

Bi-conjugate Gradient Stabilized (BiCGSTAB):

- Applicable to nonsymmetric matrices.
- Computational costs per iteration are similar to BiCG and CGS but the method doesn't require the transpose matrix.
- An alternative for CGS that avoids the irregular convergence patterns of CGS while maintaining about the same speed of convergence; as a result we often observe less loss of accuracy in the updated residual.

Generalized Minimal Residual (GMRES):

- Applicable to nonsymmetric matrices.
- GMRES leads to the smallest residual for a fixed number of iteration steps but these steps become increasingly expensive.
- In order to limit the increasing storage requirements and work per iteration step restarting is necessary. When to do so depends on A and the right-hand side; it requires skill and experience.
- GMRES requires only matrix vector products with the coefficient matrix.
- The number of inner products grows linearly with the iteration number up to the restart point. In an implementation based on a simple Gram-Schmidt process the inner products are independent so together they imply only one synchronization point. A more stable implementation based on modified Gram-Schmidt orthogonalization has one synchronization point per inner product.

2.1.2. Preconditioning

The convergence rate of iterative methods depends on the spectral properties of the coefficient matrices. Preconditioners transform the linear system into one that has the same solution but into which the linear solvers can be more efficiently applied. The need for preconditioners arouses because problems in the fluid mechanics usually give rise to matrices with undesired spectral properties especially at high Reynolds numbers.

Preconditioners increase the amount of operations per iteration. However, the additional time spent to produce the preconditioner matrix and to carry out the additional operations at each iteration is compensated by the reduction in the number of iteration steps. So they provide a decrease in the computation time on overall. Moreover, they can make a solver converge whereas a solver could diverge without the implementation of a preconditioner [12].

Accordingly, a preconditioner must be similar to the coefficient matrix as close as possible and also the preconditioner system $Mx = p$ must also be easy to solve. Taking the preconditioner matrix equal to the coefficient matrix A would enable the solver to converge in a single iteration. However, the time required for solving the system $Mx = p$ would be equal of solving the real system $Ax = b$. On the other extreme, taking the preconditioner matrix equal to the identity matrix so that the system $Mx = p$ can be easily solved would not result in any reduction in the number of iteration steps. So the ideal preconditioner must be between the two extreme cases.

The preconditioners are divided into three groups according to the application of the preconditioner matrix. These are left, right and split preconditioning.

In left-preconditioning the preconditioner matrix M is applied to the original equation ($Ax = b$) from left hand sides. The Krylov methods are then applied to the

new equation system.

$$M_{left}^{-1}Ax = M_{left}^{-1}b \tag{2.5}$$

The spectral properties of the preconditioned system ($M_{left}^{-1}A$) may be more favorable than the original one (A).

It is also possible to transform the system with right preconditioning, as given by the equation (2.6),

$$AM_{right}^{-1}y = b \quad where \quad x = M_{right}^{-1}y \tag{2.6}$$

The transformed system is first solved for y, and then, the unknown vector x is computed by the relation given in the equation (2.6).

In the case of left-preconditioning the preconditioner is applied directly to the residual vector so it may cause the algorithm stop prematurely or with delay [10]. Because of this reason right-preconditioning is preferred in this work.

The other option, split-preconditioning can be used if the preconditioner matrix is of the form

$$M = LU \tag{2.7}$$

According to this option

$$L^{-1}AU^{-1}y = L^{-1}b \quad where \quad x = U^{-1}y \tag{2.8}$$

In this work three different kinds of preconditioners are used. These are Jacobi, Symmetric Gauss-Seidel (SGS) and ILU(l). The preconditioner matrices of each of these methods are given below.

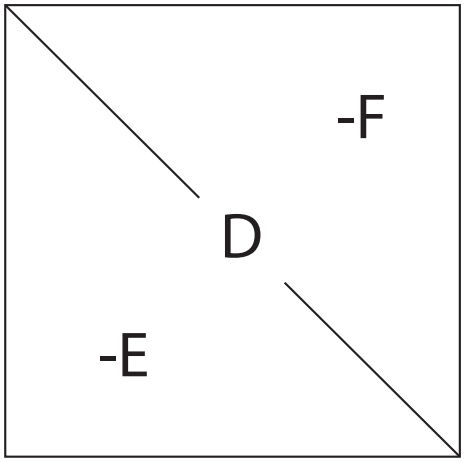


Figure 2.1. Initial partitioning of matrix A [10]

2.1.2.1. Jacobi. This method is also called diagonal scaling, since each row is scaled with respect to the entries in the diagonals.

$$M = D \tag{2.9}$$

2.1.2.2. Symmetric Gauss-Seidel(SGS). In this method the preconditioner matrix is composed of the product of a lower triangular matrix with an upper triangular matrix.

$$M = (D - E)D^{-1}(D - F) \tag{2.10}$$

In order to solve the system $Mx = p$ the factorized form of the preconditioner matrix is utilized instead of taking the inverse of the matrix. The solution procedure is given below.

- Solve $(D - E)D^{-1}y = p$ for y by forward substitution.
- Solve $(D - F)x = y$ for x by backward substitution

2.1.2.3. Incomplete LU decomposition(ILU). As it can be understood from its name the Jacobian matrix is partially decomposed into upper and lower triangular matrices

in this method. Inaccurate factorization improves the spectral properties of the matrix and provides faster convergence. The exact LU factorization would require similar amount of operation like Gaussian elimination and would be therefore an inefficient solution method. Because of this reason this factorization is carried out only at some locations in the matrix. ILU has also some versions based on the accuracy of the factorization. For ILU(0),the zero in the parenthesis indicate that the zero pattern of the decomposed matrices precisely fit the zero pattern of the Jacobian matrix and the LU factorization is carried only at the non-zero elements of the original matrix.

$$M = LU \tag{2.11}$$

In this method the system $Mx = p$ is solved using the factorized structure of the preconditioner matrix. The solution algorithm is given below.

- Solve $Ly = p$ for y by forward substitution.
- Solve $Ux = y$ for x by backward substitution

2.1.2.4. Implementation to matrix-free algorithms. As mentioned above preconditioners can have an important effect on the convergence behavior of the Krylov sub-space solvers especially when the forming matrices are ill-conditioned, which is usually the case for the flow problems at high Reynolds numbers. However, some of the preconditioners require the coefficient matrix explicitly such as ILU. Therefore these type of preconditioners cannot be adapted into a fully matrix free algorithm. On the other hand, preconditioners such as Jacobi or symmetric Gauss-Seidel (SGS) can be used in the matrix-free methods [9]. In order to apply SGS the forward and backward algorithms must be modified so that they can be carried out in a column by column manner instead of row by row. The algorithm of the matrix-free version of the backward substitution is given in Fig. 2.2 and the forward substitution algorithm in Fig. 2.3.

```
for i=1:n
    v(n+1-i,1)=1
    Av=(f(x+eps*v)-f(x))/eps
    v(n+1-i,1)=0
    x(n+1-i,1)=p(n+1-i,1)/Av(n+1-i,1)
    for j=i:n
        p(n+1-j,1)=p(n+1-j,1)-Av(n+1-j,1)*x(n+1-i,1)
    end
end
```

Figure 2.2. Backward substitution

```
for i=1:n
    v(i,1)=1;
    Av=(f(x+eps*v)-f(x))/eps
    v(i,1)=0;
    x(i,1)=p(i,1);
    for j=i:n
        p(j,1)=p(j,1)-Av(j,1)*x(i,1)/Av(i,1);
    end
end
```

Figure 2.3. Forward substitution

The application of the SGS preconditioner to the matrix free algorithms increases the operation load considerably. However, this load can be decreased combining the matrix free algorithm with the compressed column storage (CCS) scheme. In this method the Jacobian matrix is computed column by column using the directional differencing technique and stored via the CCS storage scheme. For example, using the vector with 1 only at the first entry and zeros at the remaining ones the first column of the Jacobian matrix is obtained. Continuing this process and storing every time the non-zero elements in the columns of the Jacobian matrix using CCS the matrix can be stored in a compressed manner. Using this matrix and adapting backward and forward substitutions for SGS according to the storage scheme the operation load of the algorithm is decreased considerably. This modification in the matrix free algorithm only affects the operations related with the preconditioner. The Jacobian matrix vector multiplications in the solvers are carried out matrix free. Although this modification increases the memory load it is far below than the load the exact Newton method brings.

2.1.3. Compressed Storage Schemes

The Jacobian matrices arousing from the discretization of the partial differential equations have characteristic sparsity patterns. Instead of storing each element of the matrix storing only the non-zero elements of the matrix using one of the compressed storage schemes decreases the memory load of the computer considerably. Also defining the matrix vector multiplications required for the linear solvers accordingly faster convergence can be achieved. There are various kinds of compressed storage schemes. One of them is compressed column storage (CCS). According to this scheme the non-zero elements of the matrix are stored column by column in a vector. The row number of each element is written at another vector. And a third vector indicates at which element a new column starts. In this work matrix free algorithms are modified using the CCS scheme in order to apply SGS preconditioner in a fast way to the matrix free methods.

2.2. Properties of the Numerical Solution Method

The solution method should have certain properties. In most cases we analyze the components of the method rather than analyzing the complete solution method. One of the most important properties is expressed in detail below.

A numerical method is said to be convergent if the solution of the discretized equations tends to the exact solution of the differential equation as the grid spacing tends to zero. For non-linear problems which are strongly influenced by boundary conditions, the convergence of a method are difficult to demonstrate.There are different ways of deciding for convergence. In this study the two norm of the residual vector is monitored which is a frequently used way of concluding the convergence. There are two types of residual vectors in this work. These are the residual of the linear system $J\triangle x = -f$ and the non-linear residual f.

As mentioned above the discretized form of the governing equations have a non-linear character and they are linearized by the Newton's method in order to be solved by the linear solvers. So in each Newton step a linear system has to be solved. At each step the reduction of the two norm of the linear residual vector is monitored with respect to the norm of the initial residual vector.

$$\frac{\|b - Ax^k\|}{\|b - Ax^0\|} < tol \tag{2.12}$$

Moreover, a maximum number iteration steps is also set at each Newton step. If the linear system does not converge in the given number of iteration steps then a different solver is used or the same solver combined with a more efficient preconditioner is tried. After deciding for convergence at a Newton step and making the necessary corrections in the solution vector the procedure continues with the next Newton step. After each Newton step the two norm of the f vector (nonlinear residual) is checked and compared with the user specified convergence tolerance.

$$\|f\| < tol \tag{2.13}$$

3. COMPUTATIONAL MODELING

3.1. Problem Statement

There are two main approaches to the problem of aerodynamic analysis of turbomachinery blades, the direct and the inverse approach. In the direct approach the flow is computed for a given blade geometry, while in the inverse approach the required flow distribution is specified and the corresponding blade geometry is computed.
In recent years, as a result of development in computational fluid dynamics, considerable progress has been made in the numerical solution of the direct problem of turbomachinery design. For example, Denton [5] has numerically solved the three-dimensional Euler equations of motion and Dawes [6] and Hah et al. [14] have solved the 3D Navier-Stokes equations. Such methods are of substantial value to the designer, who can use them to analyze the flow conditions along vanes and blades. Ideally, it should then be possible to modify the blade shape if the flow conditions are not those required. In practice, however, there are difficulties in determining the degree and direction of any modifications, which difficulties are compounded by the fact that a change of blade shape at any location affects the flow at other parts of the blade.
This study is concerned with the development of both two- and three-dimensional direct analysis of steady, incompressible flow over turbomachinery blades. For two-dimensional case simplification of the actual case and the problem domains are shown in Fig. 3.1, 3.3 and 3.2, respectively.

For three-dimensional analysis, the blade surfaces do not vary along the z-direction which produces a symmetry as can be observed in Fig. 3.5. This approach is basically similar to the analysis of 2-D case except the fact that the length of the blades is also considered as a parameter and adapted into the problem. Grid is generated using three-dimensional transfinite interpolation and composed of $21 \times 21 \times 21$ nodes.

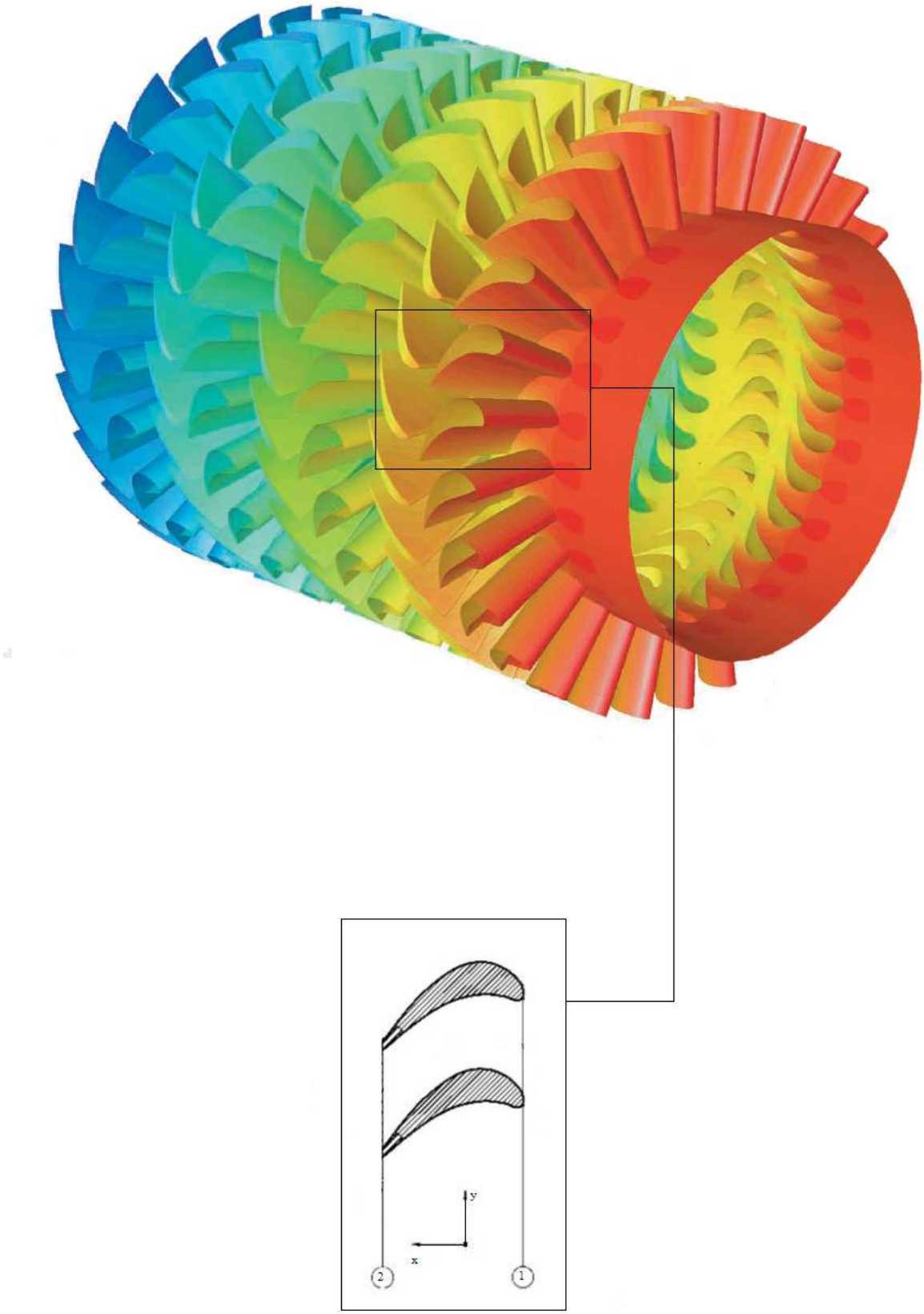


Figure 3.1. Simplification of the problem in two-dimensional analysis ($1 : Inlet$, $2 : Outlet$)

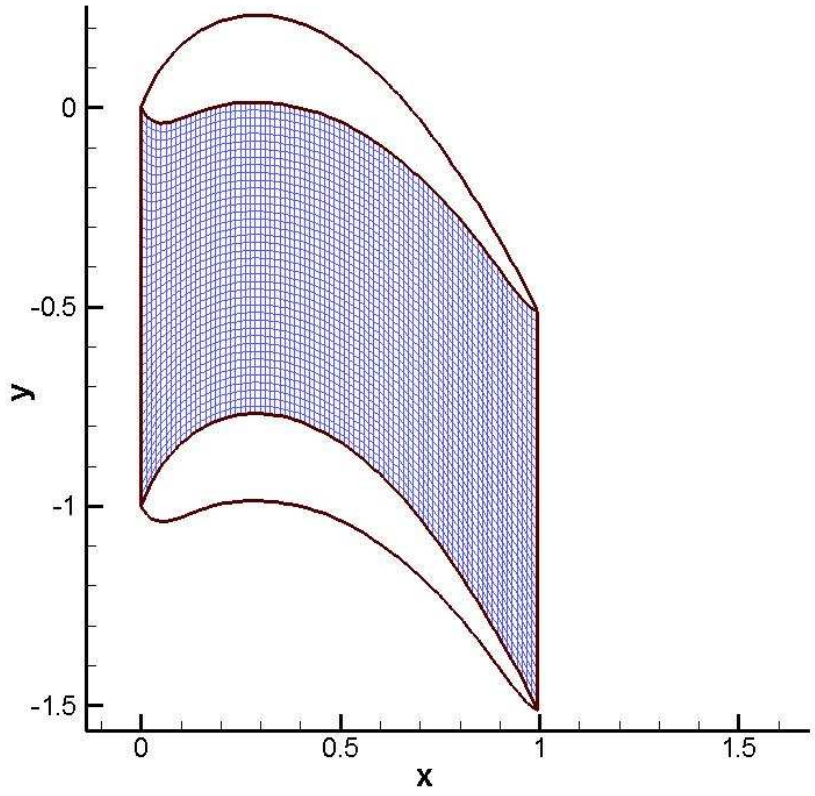


Figure 3.2. Problem domain in two-dimensional case (Algebraic grid generation)

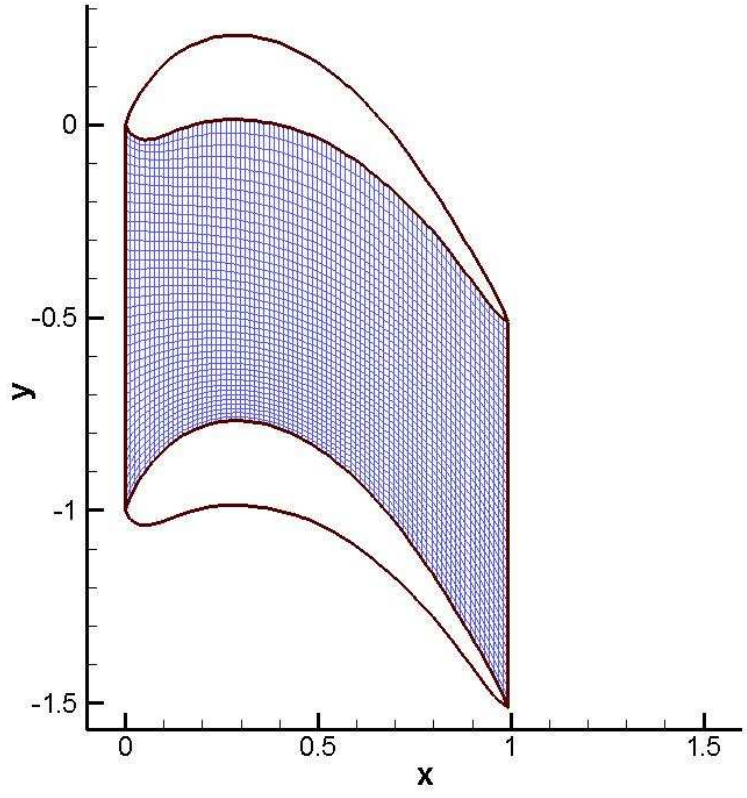


Figure 3.3. Problem domain in two-dimensional case (Elliptic grid generation)

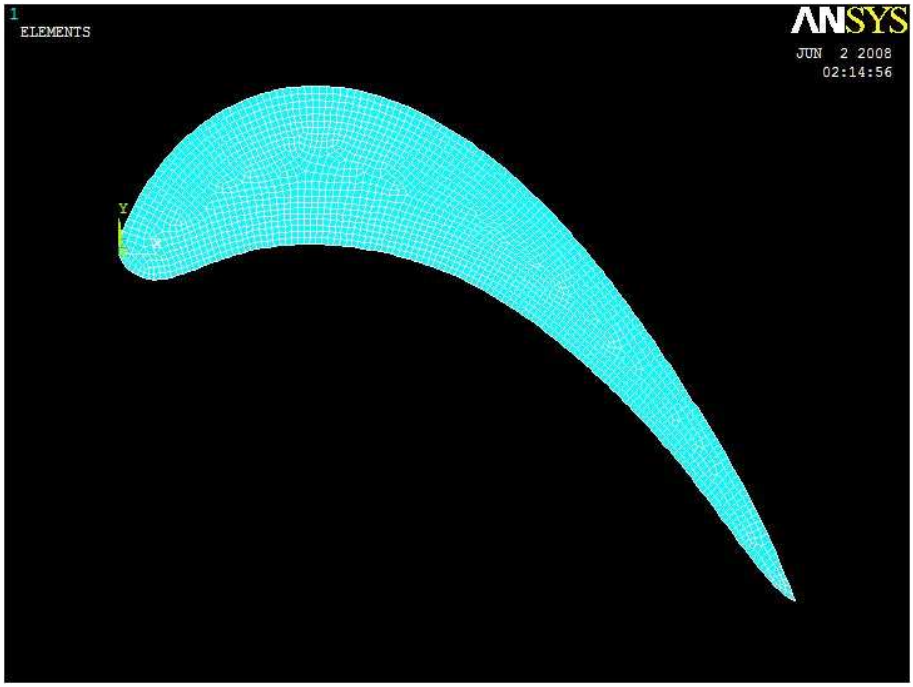


Figure 3.4. Problem domain in two-dimensional case (Solid Analysis)

3.2. Grid Generation

One of the first steps in computing a numerical solution to the equations that describe a physical problem is the construction of a grid. When the equations are expressed in terms of cartesian co-ordinates, a standard numerical method of solution is through finite differences, where a uniform rectangular grid of regularly-arranged points is constructed to cover the physical region of space (more precisely, its mathematical representation) and the partial derivatives in the equations are approximated in terms of the differences between values of the field quantities at adjacent points of the grid. A well constructed grid greatly improves the quality of the solution, and conversely, a poorly constructed grid leads to a poor result. In many applications, difficulties with numerical simulations can be traced to poor grid quality. For example, the lack of convergence to a desired level is often a result of poor grid quality. In this study, two techniques for generating grids using structured approach are applied, algebraic and differential equation methods. Algebraic and differential equation techniques can be used on complicated three-dimensional problems and of the structured methods, these have received the most use. Transfinite interpolation method for algebraic technique and elliptic grid generation for differential equation method are applied in this study.

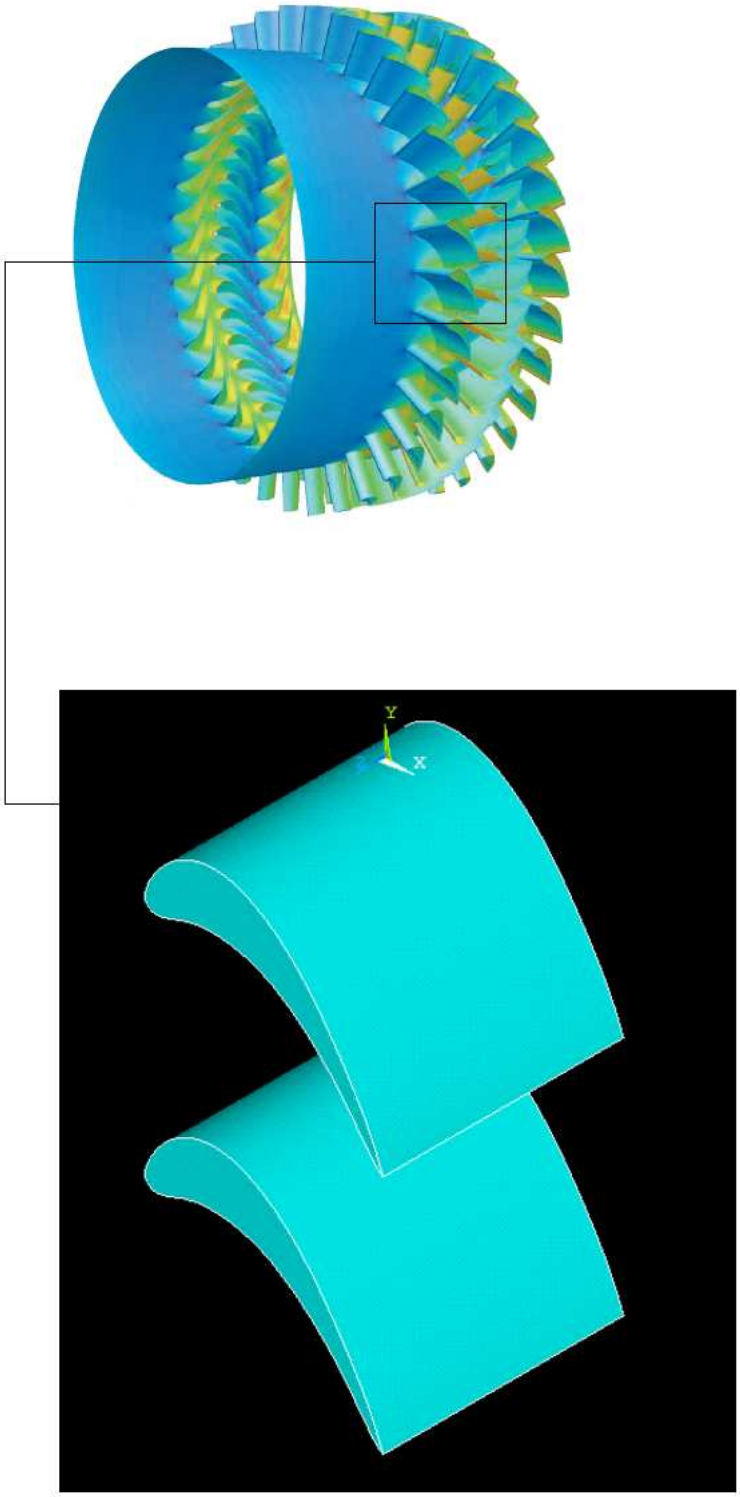


Figure 3.5. Problem domain in three-dimensional case

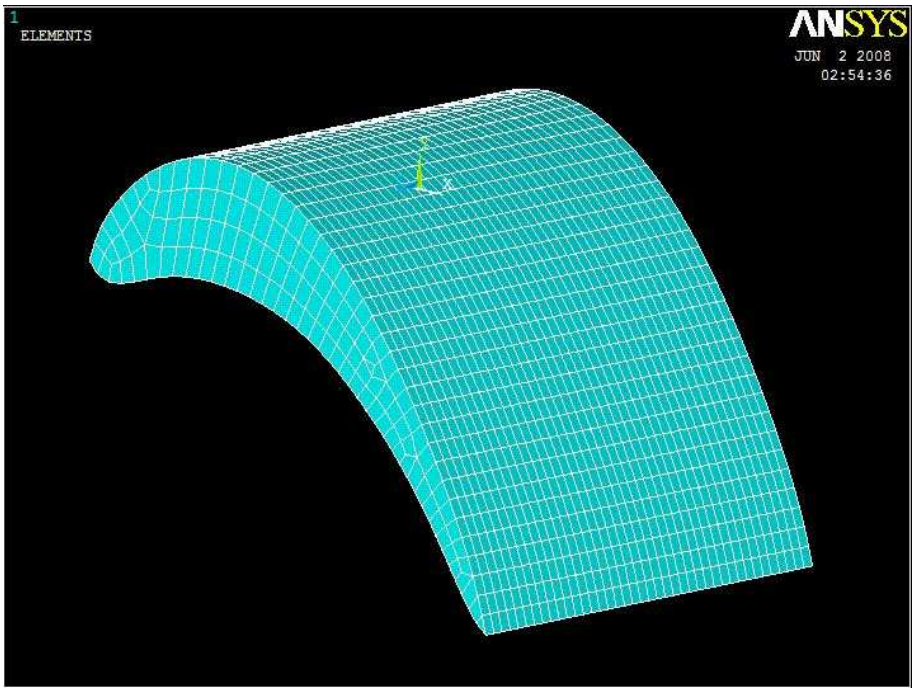


Figure 3.6. Problem domain in three-dimensional case (Solid Analysis)

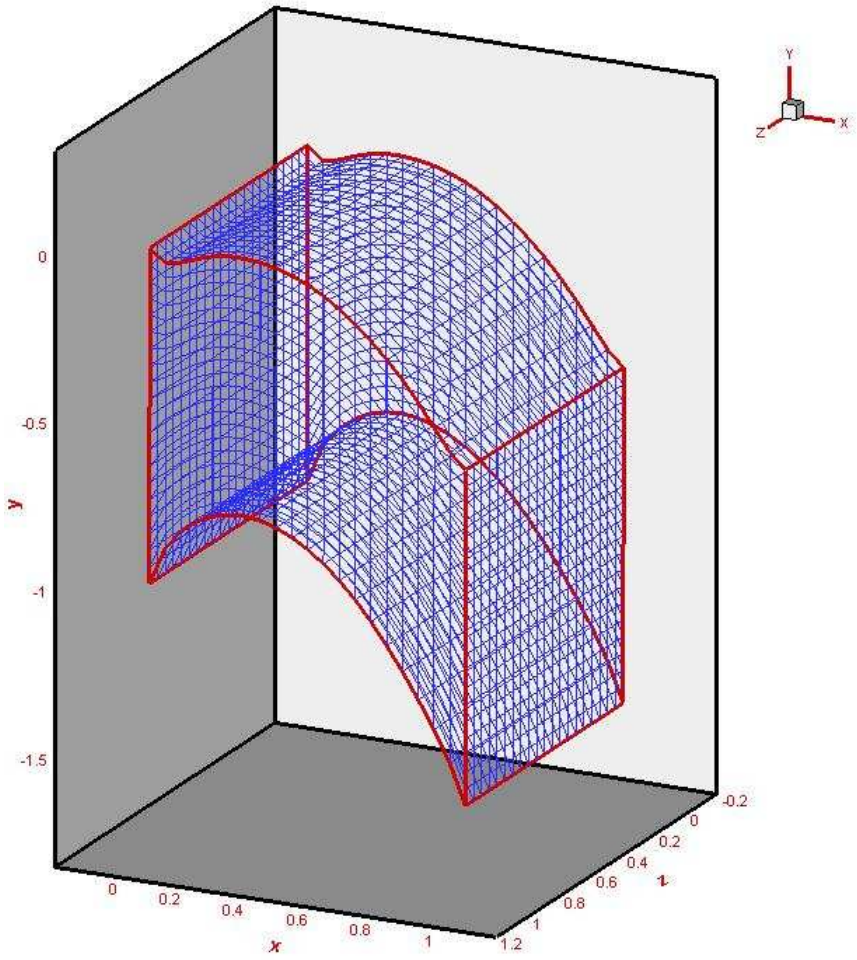


Figure 3.7. Grid generation in three-dimensional case

In Fig. 3.3 and 3.2 two different grids which are generated using algebraic and elliptic methods explained below are presented.

3.2.1. Differential Equation Method

One of the most highly developed techniques for generating acceptable grids is the differential equation method. If a differential equation is used to generate a grid, we can exploit the properties of the solution of the grid generating equation in producing the mesh. Laplace's and Poisson's equations have been extensively used for this purpose. The choice of Laplace's equation can be better understood by considering the solution of a steady heat conduction problem in two dimensions with Dirichlet boundary conditions. The solution of this problem produces isotherms which are smooth (continuous) and are nonintersecting. The number of isotherms in a given region can be increased by adding a source term. If the isotherms were used as grid lines, they would be smooth, nonintersecting and could be densely packed in any region by control of the source term. Thompson et al. [13] have worked extensively on using elliptic PDE's to generate grids. This procedure is similar to that used by Winslow [13] and transforms the physical plane into the computational plane where the mapping is controlled by a Poisson equation. This mapping is constructed by specifying the desired grid points (x, y) on the boundary of the physical domain. The distribution of points on the interior is then determined by solving

$$g_{22}\frac{\partial^2 x}{\partial \xi^2} - 2g_{12}\frac{\partial^2 x}{\partial \xi \partial \eta} + g_{11}\frac{\partial^2 x}{\partial \eta^2} = 0 \tag{3.1}$$

$$g_{22}\frac{\partial^2 y}{\partial \xi^2} - 2g_{12}\frac{\partial^2 y}{\partial \xi \partial \eta} + g_{11}\frac{\partial^2 y}{\partial \eta^2} = 0 \tag{3.2}$$

In these equations we have x and y as functions of ξ, η with g_{11}, g_{12} and g_{22} given by

$$\begin{pmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{pmatrix} = \begin{pmatrix} x_\xi^2 + y_\xi^2 & x_\xi x_\eta + y_\xi y_\eta & 0 \\ x_\xi x_\eta + y_\xi y_\eta & x_\eta^2 + y_\eta^2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.3}$$

Equations 3.1 and 3.2 are called the *Winslow equations* and used in this study to generate elliptic grid between two turbomachinery blades. The Winslow equations are in general non-linear and coupled in x and y through the coefficients g_{ij}.

3.2.1.1. Control Functions. Sometimes it may be desirable to introduce some variation of grid spacing. For example, where we expect large gradients in fluid flow variables in a boundary-layer region, we may seek a higher grid density there. If we simply use Equations 3.1 and 3.2 to generate the grid, we have no control of grid density in the interior of a physical domain R, and boundary layers cannot be properly resolved. A standard method for controlling grid density is to vary the Winslow equations by adding user-specified 'inhomogeneous' terms so that the equations become

$$g_{22}\frac{\partial^2 x}{\partial \xi^2} - 2g_{12}\frac{\partial^2 x}{\partial \xi \partial \eta} + g_{11}\frac{\partial^2 x}{\partial \eta^2} = -g\left(P\frac{\partial x}{\partial \xi} + Q\frac{\partial x}{\partial \eta}\right) \tag{3.4}$$

$$g_{22}\frac{\partial^2 y}{\partial \xi^2} - 2g_{12}\frac{\partial^2 y}{\partial \xi \partial \eta} + g_{11}\frac{\partial^2 y}{\partial \eta^2} = -g\left(P\frac{\partial y}{\partial \xi} + Q\frac{\partial y}{\partial \eta}\right) \tag{3.5}$$

where $g = g_{11}g_{22} - g_{12}g_{12}$ and $P(\xi, \eta)$, $Q(\xi, \eta)$ are suitably selected *control functions* (or *forcing functions*). It is referred as the *TTM Method* (*Thompson, Thames, and Mastin* [13]) A set of possible control functions was proposed by Thompson, Thames, and Mastin [13] as follows

$$P(\xi, \eta) = -\sum_{n=1}^{N} a_n \frac{(\xi - \xi_n)}{|\xi - \xi_n|} e^{-c_n|\xi - \xi_n|} - \sum_{i=1}^{I} b_i \frac{(\xi - \xi_i)}{|\xi - \xi_i|} e^{-d_i\left[(\xi-\xi_i)^2 + (\eta-\eta_i)^2\right]^{\frac{1}{2}}} \tag{3.6}$$

$$Q(\xi,\eta) = -\sum_{n=1}^{N} a_n \frac{(\eta-\eta_n)}{|\eta-\eta_n|} e^{-c_n|\eta-\eta_n|} - \sum_{i=1}^{I} b_i \frac{(\eta-\eta_i)}{|\eta-\eta_i|} e^{-d_i\left[(\xi-\xi_i)^2+(\eta-\eta_i)^2\right]^{\frac{1}{2}}} \tag{3.7}$$

Here N is the number of lines (co-ordinate lines $\xi = \xi_n$ and $\eta = \eta_n$) and I the number of points (with $\xi = \xi_i$, $\eta = \eta_i$, $0 \leq \xi_i,\ \eta_i \leq 1$) to which the grid is to be attracted, and a_n, c_n, b_i, d_i are positive parameters. The first term in the expression for $P(\xi,\eta)$ has the effect (with typical 'amplitude' a_n) of attracting ξ-lines (curves on which ξ is constant) towards curves $\xi = \xi_n$ in the physical domain, while the second term (with amplitude b_i) attracts ξ-lines towards points (and similarly for $Q(\xi,\eta)$). In each case the attractive effect decays with distance in computational space from the line or point in question according to the 'decay' parameters c_n, d_i.

The functions $(\xi-\xi_n)/|\xi-\xi_n|$ and $(\eta-\eta_n)/|\eta-\eta_n|$ are functions which can take only the values ± 1, and are present to ensure that the attraction takes place on both sides of ξ_n-lines and η_n-lines and in the entire neighborhood of points (ξ_i,η_i). Taking the amplitudes to be negative turns the attractive effects into repulsive ones.

3.2.2. Algebraic Method

3.2.2.1. Two-Dimensional TFI (Projectors and bilinear mapping). Suppose there exists a transformation $r = r(\xi,\eta)$ (or $x = x(\xi,\eta)$, $y = y(\xi,\eta)$) which maps the unit square $0 < \xi < 1$, $0 < \eta < 1$ onto the interior of the region $ABDC$ in the xy (physical) plane (Fig. 3.8), such that the edges $\xi = 0, 1$ map to the boundaries AB, CD, respectively, which we can formulate as $r(0,\eta)$ and $r(1,\eta)$, the boundaries AC, BD being similarly given by $r(\xi,0)$, $r(\xi,1)$. We can write down another transformation P_ξ , called a *projector*, which maps points in computational space to points (or position vectors) in physical space, defined by

$$P_\xi(\xi,\eta) = (1-\xi)\, r(0,\eta) + \xi r(1,\eta) \tag{3.8}$$

The sides $\xi = 0, 1$ are mapped onto AB, CD respectively, and the sides $\eta = 0, 1$ are mapped onto the straight lines AC, BD. Furthermore, co-ordinate lines of constant η

are mapped into straight lines rather than co-ordinate curves in the physical plane. Similarly we can define the projector

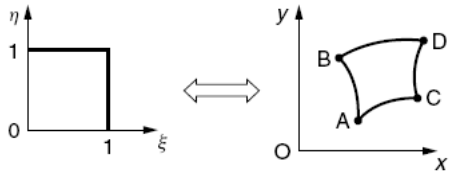


Figure 3.8. Mapping unit square onto curved four-sided figure

$$\mathrm{P}_\eta(\xi,\eta) = (1-\eta)r(\xi,0) + \eta r(\xi,1) \tag{3.9}$$

which maps the unit square onto a region which preserves the boundaries AC, BD, but replaces the boundaries AB, CD with straight lines (Fig. 3.9). We can form the

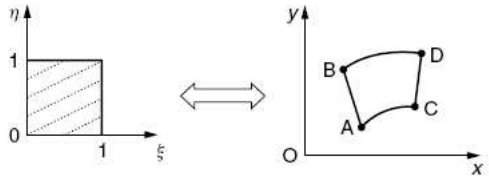


Figure 3.9. Projector P_η

composite mapping $\mathrm{P}_\xi\mathrm{P}_\eta$, such that

$$\begin{aligned}
&\mathrm{P}_\xi\left(P_\eta\left(\xi,\eta\right)\right) = P_\xi\left(\left(1-\eta\right)r\left(\xi,0\right) + \eta r\left(\xi,1\right)\right)\\
&= \left(1-\xi\right)\left[\left(1-\eta\right)r\left(0,0\right) + \eta r\left(0,1\right)\right] + \xi\left[\left(1-\eta\right)r\left(1,0\right) + \eta r\left(1,1\right)\right]\\
&= \left(1-\xi\right)\left(1-\eta\right)r\left(0,0\right)\\
&+ \left(1-\xi\right)\eta r\left(0,1\right) + \xi\left(1-\eta\right)r\left(1,0\right) + \xi\eta r\left(1,1\right)
\end{aligned} \tag{3.10}$$

This bilinear transformation has the property that the four vertices *A, B, C, D* are preserved, but the boundaries are all replaced by straight lines; that is, the unit square is mapped onto a quadrilateral $ABDC$ (Fig. 3.10). Moreover, straight lines $\xi = const.$ and $\eta = const.$ in computational space are mapped onto straight lines in physical space. It is easy to show that this composition of projectors, often referred to as the *tensor product* of P_ξ and P_η, is commutative; that is,

$$\mathrm{P}_\xi\mathrm{P}_\eta = \mathrm{P}_\eta\mathrm{P}_\xi \tag{3.11}$$

Note also that we can form the composite map $\mathrm{P}_\xi \mathrm{P}_\xi$; we obtain

$$\begin{aligned} &\mathrm{P}_\xi \left(P_\xi \left(\xi, \eta\right)\right) = \mathrm{P}_\xi \left[\left(1-\xi\right) r\left(0, \eta\right) + \xi r\left(1, \eta\right)\right] \\ &= \left(1-\xi\right) r\left(0, \eta\right) + \xi r\left(1, \eta\right) = P_\xi \left(\xi, \eta\right) \end{aligned} \tag{3.12}$$

Hence we can write

$$\mathrm{P}_\xi \mathrm{P}_\xi = \mathrm{P}_\xi \tag{3.13}$$

which is the usual defining property of *projection operators*. Let us now consider the

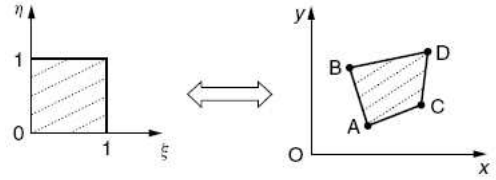


Figure 3.10. Bilinear transformation $\mathrm{P}_\xi \mathrm{P}_\eta$

various mappings of the side $\eta = 0$ of the unit square. Under P_ξ it is mapped to the straight line AC; under P_η it is mapped to the curved boundary AC; finally under $\mathrm{P}_\xi \mathrm{P}_\eta$ it is mapped to the straight line AC. Similar considerations applied to each side of the unit square show that the composite map $(\mathrm{P}_\xi + \mathrm{P}_\eta - \mathrm{P}_\xi \mathrm{P}_\eta)$ is a transformation which maps the entire boundary of the unit square onto the entire curved boundary $ABDC$. This map is called the *Boolean sum* of the transformations P_ξ and P_η, and denoted by $\mathrm{P}_\xi \oplus \mathrm{P}_\eta$. Thus

$$\mathrm{P}_\xi \oplus \mathrm{P}_\eta = \mathrm{P}_\xi + \mathrm{P}_\eta - \mathrm{P}_\xi \mathrm{P}_\eta \tag{3.14}$$

It is clear that $\mathrm{P}_\xi \oplus \mathrm{P}_\eta = \mathrm{P}_\eta \oplus \mathrm{P}_\xi$. The complete formulation is

$$\begin{aligned} &\left(\mathrm{P}_\xi \oplus \mathrm{P}_\eta\right)\left(\xi, \eta\right) = \mathrm{P}_\xi \left(\xi, \eta\right) + \mathrm{P}_\eta \left(\xi, \eta\right) - \mathrm{P}_\xi \mathrm{P}_\eta \left(\xi, \eta\right) \\ &= \left(1-\xi\right) r\left(0, \eta\right) + \xi r\left(1, \eta\right) + \left(1-\eta\right) r\left(\xi, 0\right) + \eta r\left(\xi, 1\right) \\ &- \left(1-\xi\right)\left(1-\eta\right) r\left(0, 0\right) - \left(1-\xi\right) \eta r\left(0, 1\right) \\ &- \left(1-\eta\right) \xi r\left(1, 0\right) - \xi \eta r\left(1, 1\right) \end{aligned} \tag{3.15}$$

This transformation is the basis of *transfinite interpolation (TFI)* in two dimensions. A grid will be generated by Equation 3.15 by taking discrete values ξ_i , η_j of ξ and η with

$$\begin{aligned} &0 \leq \xi_i = \tfrac{i-1}{I-1} \leq 1 \\ &0 \leq \eta_j = \tfrac{j-1}{J-1} \leq 1 \\ &i = 1, 2, ..., I, j = 1, 2, ..., J \end{aligned} \tag{3.16}$$

for some choice of I and J. Transfinite interpolation is the most common approach to algebraic grid generation. It can produce excellent grids quickly in situations where other methods would be difficult to apply, and it also allows for direct control of the location of grid nodes. Many two-dimensional regions are easy to grid accurately using TFI. However, there are some geometries, such as the airfoil, backstep, and C-grids, where TFI proves to be unsatisfactory. The main disadvantages are (1) a lack of smoothness in the generated grids, with any discontinuities in gradient in the boundary curves tending to propagate into the interior, and (2) a tendency to fold when the geometries are complex.

The method can be extended in many ways. For example, the physical region can be divided into several parts, with grids being generated in each separate part and then matched together at the interfaces. This results in discontinuities of slope at the interfaces, and Hermite polynomial interpolation may be exploited to match slopes and thus remove the discontinuities. It is also possible to use TFI with higher-order polynomials as blending functions.

3.2.2.2. Numerical Implementation Of TFI. We write Equation 3.15, with reference to Fig. 3.11, as

$$\begin{aligned} &r(\xi.\eta) = (1-\xi)\, r_l(\eta) + \xi r_r(\eta) + (1-\eta)\, r_b(\xi) + \eta r_t(\xi) \\ &-(1-\xi)(1-\eta)\, r_b(0) - (1-\xi)\,\eta r_t(0) - (1-\eta)\,\xi r_b(1) - \xi\eta r_t(1) \end{aligned} \tag{3.17}$$

where the abbreviations *l, r, b, t* stand for left, right, bottom, top. At the four vertices

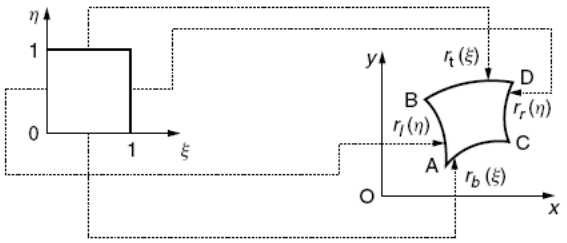


Figure 3.11. Mapping of boundary curves

of the physical domain we need the consistency conditions

$$\begin{aligned} r_b(0) &= r_l(0) \\ r_b(1) &= r_r(0) \\ r_r(1) &= r_t(1) \\ r_l(1) &= r_t(0) \end{aligned} \tag{3.18}$$

Equation 3.17 is equivalent to the two component equations

$$\begin{aligned} &x(\xi.\eta) = (1-\xi)\, x_l(\eta) + \xi x_r(\eta) + (1-\eta)\, x_b(\xi) + \eta x_t(\xi) \\ &-(1-\xi)(1-\eta)\, x_b(0) - (1-\xi)\,\eta x_t(0) - (1-\eta)\,\xi x_b(1) - \xi\eta x_t(1) \end{aligned} \tag{3.19}$$

and

$$\begin{aligned} &y(\xi.\eta) = (1-\xi)\, y_l(\eta) + \xi y_r(\eta) + (1-\eta)\, y_b(\xi) + \eta y_t(\xi) \\ &-(1-\xi)(1-\eta)\, y_b(0) - (1-\xi)\,\eta y_t(0) - (1-\eta)\,\xi y_b(1) - \xi\eta y_t(1) \end{aligned} \tag{3.20}$$

These equations can be discretized and evaluated through a nested DO loop. Suppose we choose $(m+1)$ grid nodes on the bottom and top boundaries in the computational plane, with equal increments $\Delta\xi = 1/m$ in ξ between nodes; similarly, $(n+1)$ nodes on left and right, with equal increments $\Delta\eta = 1/n$ in η. We need the boundary data for the functions r_b, r_t , r_l , r_r , i.e. the values of the (x, y) co-ordinates at the selected points corresponding to the chosen values of ξ and η on each part of the boundary. This data can be made available to the main routine through a data-file. Or, if the boundaries can be calculated according to some analytical expression, then this can be done in a subroutine. A basic program with a double loop to compute Equations 3.19 and 3.20, setting $\xi = s$, $\eta = t$, $\Delta\xi = dX = 1/m$, $\Delta\eta = dY = 1/n$, would then take the

form:

```
DO J=2:n
   t=(J-1)*dY
DO 2 I=2:m
   s=(I-1)*dX

    X(I,J) = (1.0-s)*X_l(J)+s*X_r(J)+(1.0-t)*X_b(I)+t*X_t(I)
            -(1.0-s)*(1.0-t)*X_b(1)-(1.0-s)*t*X_t(1)
            -s*(1.0-t)*X_b(m+1)-s*t*X_t(m+1)
    Y(I,J) = (1.0-s)*Y_l(J)+s*Y_r(J)+(1.0-t)*Y_b(I)+t*Y_t(I)
            -(1.0-s)*(1.0-t)*Y_b(1)-(1.0-s)*t*Y_t(1)
            -s*(1.0-t)*Y_b(m+1)-s*t*Y_t(m+1)

2 Continue
1 Continue
```

Figure 3.12. Numerical implementation of TFI in 2D

3.2.2.3. Three-Dimensional TFI. A simple approach to TFI in three dimensions is through the extension of the definition of projectors to 3D. Suppose that we have a mapping $r(\xi, \eta, \zeta)$ from the unit cube $0 \leq \xi \leq 1$, $0 \leq \eta \leq 1$, $0 \leq \zeta \leq 1$ to a six-sided volume R of physical space. The opposite planar faces of the cube given by $\xi = 0, 1$ map onto the (in general, curved) opposite faces $r(0, \eta, \zeta)$, $r(1, \eta, \zeta)$ of R. On these faces there are curvilinear co-ordinate systems with η and ζ as co-ordinates. Edges of the cube such as that given by $\eta = \zeta = 0$ (with $0 \leq \xi \leq 1$) map into edges of R such as $r(\xi, 0, 0)$, which is a ξ -co-ordinate curve. Using linear Lagrange polynomials as blending functions, the following projectors may be defined:

$$P_\xi (\xi, \eta, \zeta) = (1 - \xi) r (0, \eta, \zeta) + \xi r (1, \eta, \zeta) \tag{3.21}$$

$$P_\eta(\xi,\eta,\zeta) = (1-\xi)\, r(\xi,0,\zeta) + \eta r(\xi,1,\zeta) \tag{3.22}$$

$$P_\zeta(\xi,\eta,\zeta) = (1-\xi)\, r(\xi,\eta,0) + \zeta r(\xi,\eta,1) \tag{3.23}$$

Now the projector P_ξ still maps the opposite faces $\xi = 0, 1$ of the cube onto the opposite faces $r(0,\eta,\zeta)$, $r(1,\eta,\zeta)$ of R. It also maps all the vertices of the cube, $(0,0,0)$, $(1,0,0)$, etc., onto the vertices $r(0,0,0)$, $r(1,0,0)$, etc., of R. However, the four edges of the cube which connect opposite vertices of the faces $\xi = 0$ and $\xi = 1$ are mapped onto straight lines connecting corresponding vertices of R. For example, $(\xi,0,0) \rightarrow (1-\xi)r(0,0,0)+\xi r(1,0,0), 0 \leq \xi \leq 1$. Clearly the other projectors P_η, P_ζ have similar properties. Moreover they all satisfy the basic projection property given by Equation 3.13. If we started out with only two opposite faces of R specified and were able to construct curvilinear co-ordinate systems on these surfaces with η and ζ as coordinates Fig. 3.13, we could then have a grid on these faces corresponding to discrete values η_j , ζ_k with

$$\begin{aligned} &0 \leq \eta_j = \tfrac{j-1}{J-1} \leq 1 \\ &0 \leq \zeta_k = \tfrac{k-1}{K-1} \leq 1 \\ &j = 1,2,...,J, k = 1,2,...,K \end{aligned} \tag{3.24}$$

for some J, K. We could then use P_ξ, through Equation 3.21, to interpolate a grid between these faces, taking discrete values of ξ also, with $0 \leq \xi_i = \frac{i-1}{I-1} \leq 1, i = 1,2,...,I$. The bilinear tensor product $P_\xi P_\eta$ may be expressed in full as

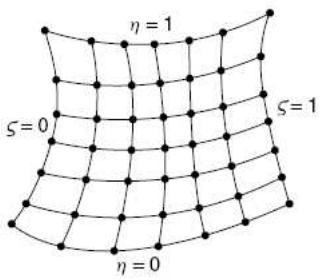


Figure 3.13. Surface grid

$$P_\xi P_\eta (\xi,\eta,\zeta) = (1-\xi)(1-\eta) r(0,0,\zeta) + (1-\xi)\eta r(0,1,\zeta) \\ +\xi(1-\eta) r(1,0,\zeta) + \xi\eta r(1,1,\zeta) \tag{3.25}$$

The effect of this transformation on the unit cube is to map all four straight edges parallel to the ζ direction onto the corresponding four curved edges $r(0,0,\zeta)$, etc., of R. Between these curved edges we then have linear interpolation in both ξ and η directions. This map could be used for linear interpolation if we started with just those four edges of R. The other bilinear products have similar properties, and are given by

$$P_\eta P_\zeta (\xi,\eta,\zeta) = (1-\eta)(1-\zeta) r(\xi,0,0) + (1-\eta)\zeta r(\xi,0,1) \\ +\eta(1-\zeta) r(\xi,1,0) + \eta\zeta r(\xi,1,1) \tag{3.26}$$

$$P_\xi P_\zeta (\xi,\eta,\zeta) = (1-\xi)(1-\zeta) r(0,\eta,0) + (1-\xi)\zeta r(0,\eta,1) \\ +\xi(1-\zeta) r(1,\eta,0) + \xi\zeta r(1,\eta,1) \tag{3.27}$$

Clearly these products all have the property of commutativity. We can also formulate the trilinear transformation $P_\xi P_\eta P_\zeta$, which may be expressed in full as

$$P_\xi P_\eta P_\zeta = (1-\xi)(1-\eta)(1-\zeta) r(0,0,0) \\ +\xi(1-\eta)(1-\zeta) r(1,0,0) + (1-\xi)\eta(1-\zeta) r(0,1,0) \\ +(1-\xi)(1-\eta)\zeta r(0,0,1) + \xi\eta(1-\zeta) r(1,1,0) \\ +\xi(1-\eta)\zeta r(1,0,1) + (1-\xi)\eta\zeta r(0,1,1) + \xi\eta\zeta r(1,1,1) \tag{3.28}$$

This *trilinear interpolant* maps the unit cube onto a region of physical space with the same vertices as R but with straight lines connecting the vertices. The Boolean sum $P_\xi \oplus P_\eta \oplus P_\zeta$ may be formulated in terms of the above mappings by successively applying the definition Equation 3.14. We have

$$P_\xi \oplus (P_\eta \oplus P_\zeta) = P_\xi \oplus (P_\eta + P_\zeta - P_\eta P_\zeta) \\ = P_\xi + P_\eta + P_\zeta - P_\eta P_\zeta - P_\xi P_\eta - P_\xi P_\zeta + P_\xi P_\eta P_\zeta \tag{3.29}$$

It is straightforward to show that the same result emerges from evaluating $(P_\xi \oplus P_\eta) \oplus P_\zeta$, which means that Boolean summation is *associative*. Putting $\xi = 0$ in the expressions Equations 3.21, 3.22, 3.23, 3.25, 3.26, 3.27 and 3.21, and combining the results according to the vector sums in Equation 3.29 shows that the face $\xi = 0$ of the unit cube maps onto the curved face $r(0, \eta, \zeta)$ of R under the Boolean sum Equation 3.29. In fact each face of the cube maps onto a face of R.

From the above discussion it is clear that, in terms of projectors, the product $P_\xi P_\eta P_\zeta$ is algebraically minimal, in that it is the weakest member of the set of projectors to generate a grid (based on TFI) in R, given that it interpolates only from the eight vertices of R. The Boolean sum $P_\xi \oplus P_\eta \oplus P_\zeta$, on the other hand, is algebraically maximal and the strongest member of the projector set. To use it we need boundary data on all six faces of R (including the twelve edges and eight vertices). Then Equation 3.29 will generate a grid within R by trilinear interpolation, taking discrete values of ξ, η, ζ. In practice, however, we may not have a complete set of boundary data. Suppose, for example, that we have only boundary data pertaining to the twelve edges of the physical region R. Since Equation 3.25 showed that the product $P_\xi P_\eta$ interpolates linearly from four edges of R, we might expect the appropriate grid generation formula to be given by the Boolean product $P_\xi P_\eta \oplus P_\eta P_\zeta \oplus P_\zeta P_\xi$. This can be easily evaluated in terms of the tensor products above with use of commutativity and the basic projection property Equation 3.13. We have

$$\begin{aligned}
&P_\xi P_\eta \oplus (P_\eta P_\zeta \oplus P_\zeta P_\xi) = P_\xi P_\eta \oplus (P_\eta P_\zeta + P_\zeta P_\xi - P_\eta P_\zeta P_\zeta P_\xi) \\
&= P_\xi P_\eta \oplus (P_\eta P_\zeta + P_\zeta P_\xi - P_\eta P_\zeta P_\xi) \\
&= P_\xi P_\eta + (P_\eta P_\zeta + P_\zeta P_\xi - P_\eta P_\zeta P_\xi) \\
&- P_\xi P_\eta \, (P_\eta P_\zeta + P_\zeta P_\xi - P_\eta P_\zeta P_\xi) \\
&= P_\xi P_\eta + P_\eta P_\zeta + P_\zeta P_\xi - P_\eta P_\zeta P_\xi \\
&- P_\xi P_\eta P_\zeta - P_\xi P_\eta P_\zeta + P_\xi P_\eta P_\zeta \\
&= P_\xi P_\eta + P_\eta P_\zeta + P_\zeta P_\xi - 2 P_\xi P_\eta P_\zeta
\end{aligned} \tag{3.30}$$

An explicit expression for this transfinite interpolation (based on twelve edges of boundary data) may be written down by combining Equations 3.25, 3.26, 3.27 and 3.28 ac-

cording to 3.30 which is used in this study to generate three-dimensional grid between turbomachinery blades.

3.3. Solution Algorithm

This study is concerned with the coupled fluid-structure analysis of turbomachinery blades. At first, geometry of the problem is generated in an aerodynamic code and blade points are transferred into ANSYS. Exact points on the blade surface are determined by using ANSYS considering the blade geometry and the number of points on the blade which are specified earlier by user. Then, new grid values on the blade are given as the boundary conditions to generate the grid between two blades and on this domain the fluid flow is analyzed so that the velocity, vorticity and pressure field are computed by using an aerodynamic code (solving discretized form of the Navier-Stokes with the assumption of steady, incompressible flow). By solving the Navier-Stokes equations fluid analysis of the first step is completed and pressure values determined as a result of this analysis are implemented as boundary conditions into ANSYS environment. Displacements are found as a result of solid analysis performed by ANSYS and transferred into fluid analysis so that the boundary conditions of the grid generators are updated. With storing displacements, the first step of FSI analysis is completely carried out and second step is started by solving the governing equations on the new physical domain which is different from the first one because of deformation. Since the geometry of the problem is changed, the velocity, vorticity and pressure fields are also changed and fluid analysis is performed again to achieve new results of fluid flow between deformed blades. It can be concluded that fluid and solid analysis are performed separately at each step of the solution procedure and the convergence is decided by monitoring the difference between the successive displacement vectors. Two norm of the resulting vector is checked to finish FSI analysis as

$$\left\| d^k - d^{k-1} \right\| < tol \tag{3.31}$$

3.4. Discretization of the Governing Equations

3.4.1. Two-Dimensional Analysis Using Velocity-Vorticity Formulation

The vorticity at a fluid point can be defined as follows

$$\overrightarrow{\Omega} = 2\overrightarrow{\omega} = \nabla \times \overrightarrow{V} \tag{3.32}$$

In this equation ω refers the angular velocity and for a two dimensional flow the equation is reduced to

$$\Omega_z = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \tag{3.33}$$

For the flow of incompressible Newtonian fluids the governing equations in Cartesian coordinates can be written as
Conservation of mass :

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{3.34}$$

Conservation of momentum :

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{3.35}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{3.36}$$

In order to derive the vorticity transport equation, the pressure is eliminated from the momentum equations by cross-differentiation. If we differentiate Equation

3.35 with respect to y we have an expression as

$$\frac{\partial^2 u}{\partial y \partial t} + \frac{\partial u}{\partial y}\frac{\partial u}{\partial x} + u\frac{\partial^2 u}{\partial x \partial y} + \frac{\partial v}{\partial y}\frac{\partial u}{\partial y} + v\frac{\partial^2 u}{\partial y^2} = -\frac{\partial^2 p}{\partial x \partial y} + \nu\left(\frac{\partial^3 u}{\partial y \partial x^2} + \frac{\partial^3 u}{\partial y^3}\right) \tag{3.37}$$

and if we differentiate Equation 3.36 with respect to x an expression will be reached as

$$\frac{\partial^2 v}{\partial x \partial t} + \frac{\partial u}{\partial x}\frac{\partial v}{\partial x} + u\frac{\partial^2 v}{\partial x^2} + \frac{\partial v}{\partial x}\frac{\partial v}{\partial y} + v\frac{\partial^2 v}{\partial x \partial y} = -\frac{\partial^2 p}{\partial x \partial y} + \nu\left(\frac{\partial^3 v}{\partial x^3} + \frac{\partial^3 v}{\partial x \partial y^2}\right) \tag{3.38}$$

Subtracting Equation 3.38 from Equation 3.37 we finally obtain

$$\begin{aligned}\frac{\partial}{\partial t}\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right) + u\frac{\partial}{\partial x}\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right) + v\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right) + \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right) \\ = \nu\left[\frac{\partial^2}{\partial x^2}\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right) + \frac{\partial^2}{\partial y^2}\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right)\right]\end{aligned} \tag{3.39}$$

By using continuity we note that the fourth term on the left-hand side will be zero. Substituting the vorticity defined by Equation 3.33 into Equation 3.39, we will obtain

$$\frac{\partial \Omega}{\partial t} + u\frac{\partial \Omega}{\partial x} + v\frac{\partial \Omega}{\partial y} = \nu\left(\frac{\partial^2 \Omega}{\partial x^2} + \frac{\partial^2 \Omega}{\partial y^2}\right) \tag{3.40}$$

Equation 3.46 is known as the vorticity transport equation and is classified as a parabolic equation with the unknown being the vorticity Ω .

In order to derive the equations for the components of velocity we need to work further with the conservation of mass and the definition of the vorticity. If we differentiate Equation 3.34 with respect to x, Equation 3.33 with respect to y and make necessary arrangements we obtain an expression as follows

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -\frac{\partial \Omega}{\partial y} \tag{3.41}$$

Similarly differentiating Equation 3.34 with respect to y, Equation 3.33 with respect

to x and making necessary arrangements will yield

$$\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = \frac{\partial \Omega}{\partial x} \tag{3.42}$$

Equations 3.41 and 3.42 will be used for the x and y component of velocity, respectively.

The vorticity transport and velocity equations can be expressed in nondimensional forms by using the nondimensional quantities as follows

$$\begin{aligned} t^* &= \frac{t u_\infty}{L} & x^* &= \frac{x}{L} & y^* &= \frac{y}{L} \\ u^* &= \frac{u}{u_\infty} & v^* &= \frac{v}{u_\infty} & \Omega^* &= \frac{\Omega L}{u_\infty} \\ \mathrm{Re} &= \frac{\rho_\infty u_\infty L}{\mu_\infty} \end{aligned} \tag{3.43}$$

where L is a characteristic length, ρ_∞ and u_∞ are the reference (freestream) density and velocity, respectively. The nondimensional forms of the velocity and the vorticity transport equations can be written by using the nondimensional quantities defined above as

$$\frac{\partial^2 u^*}{\partial x^{*2}} + \frac{\partial^2 u^*}{\partial y^{*2}} = -\frac{\partial \Omega^*}{\partial y^*} \tag{3.44}$$

$$\frac{\partial^2 v^*}{\partial x^{*2}} + \frac{\partial^2 v^*}{\partial y^{*2}} = \frac{\partial \Omega^*}{\partial x^*} \tag{3.45}$$

$$\frac{\partial \Omega^*}{\partial t^*} + u^* \frac{\partial \Omega^*}{\partial x^*} + v^* \frac{\partial \Omega^*}{\partial y^*} = \frac{1}{\mathrm{Re}} \left(\frac{\partial^2 \Omega^*}{\partial x^{*2}} + \frac{\partial^2 \Omega^*}{\partial y^{*2}} \right) \tag{3.46}$$

3.4.1.1. Transformation of the Governing Equations. We will transform the independent variables in physical space (x,y) to a new set of independent variables in transformed space (ξ,η). If we define the relations between the physical and computational

spaces, we obtain the following expressions

$$\begin{aligned} \xi &= \xi(x, y) \\ \eta &= \eta(x, y) \end{aligned} \tag{3.47}$$

Equations 3.47 represent the *transformation*. In the above relations the transformation is written in *generic* form; for an actual application the transformation must be given as some type of *specific analytical relation* or some times a *specific numerical relation*. The chain rule for partial differentiation yields the following expressions

$$\frac{\partial}{\partial x} = \frac{\partial \xi}{\partial x}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial x}\frac{\partial}{\partial \eta} \tag{3.48}$$

$$\frac{\partial}{\partial y} = \frac{\partial \xi}{\partial y}\frac{\partial}{\partial \xi} + \frac{\partial \eta}{\partial y}\frac{\partial}{\partial \eta} \tag{3.49}$$

In the following discussion the partial derivatives will be denoted using subscripts, $\xi_x = \partial\xi/\partial x$. In Equations 3.48 and 3.49 terms such as ξ_x, ξ_y, η_x and η_y appear.These transformation derivatives are defined as the metrics of transformation or simply as the metrics. The interpretation of the metrics is obvious considering the following expression

$$\xi_x = \frac{\partial \xi}{\partial x} \cong \frac{\Delta \xi}{\Delta x} \tag{3.50}$$

This expression indicates that the metrics represent the ratio of arc lengths in the computational space to that of the physical space. From Equations 3.47 the following differential expressions are obtained

$$d\xi = \xi_x dx + \xi_y dy \tag{3.51}$$

$$d\eta = \eta_x dx + \eta_y dy \tag{3.52}$$

which are written in a compact form as

$$\begin{bmatrix} d\xi \\ d\eta \end{bmatrix} = \begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} \tag{3.53}$$

Reversing the role of independent variables as

$$\begin{aligned} x &= x\left(\xi, \eta\right) \\ y &= y(\xi, \eta) \end{aligned} \tag{3.54}$$

The following can be written

$$\begin{aligned} dx &= x_\xi d\xi + x_\eta d\eta \\ dy &= y_\xi d\xi + y_\eta d\eta \end{aligned} \tag{3.55}$$

In a compact form they are written as

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix} \begin{bmatrix} d\xi \\ d\eta \end{bmatrix} \tag{3.56}$$

Comparing Equations 3.53 and 3.56, it can be concluded that

$$\begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{bmatrix}^{-1} \tag{3.57}$$

from which

$$\begin{aligned} \xi_x &= \tfrac{1}{J} y_\eta \\ \xi_y &= -\tfrac{1}{J} x_\eta \\ \eta_x &= -\tfrac{1}{J} y_\xi \\ \eta_y &= \tfrac{1}{J} x_\xi \end{aligned} \tag{3.58}$$

where

$$J = x_\xi y_\eta - y_\xi x_\eta \tag{3.59}$$

or as a matrix form

$$J = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \tag{3.60}$$

and is defined as the *Jacobian of Transformation.* For a successful transformation, the Jacobian must be *non-singular.* Namely, one should be able to invert it, which requires that determinant of J never vanishes.

Also, in many applications, the transformation may be more conveniently expressed as the inverse of Equations 3.47; that is

$$\begin{aligned} x &= x(\xi, \eta) \\ y &= y(\xi, \eta) \end{aligned} \tag{3.61}$$

If we consider an arbitrary dependent variable such as , the chain rule for partial differentiation of the variable gives the following expression

$$\frac{\partial p}{\partial \xi} = \frac{\partial p}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial p}{\partial y}\frac{\partial y}{\partial \xi} \tag{3.62}$$

$$\frac{\partial p}{\partial \eta} = \frac{\partial p}{\partial x}\frac{\partial x}{\partial \eta} + \frac{\partial p}{\partial y}\frac{\partial y}{\partial \eta} \tag{3.63}$$

In order to obtain the two unknowns $\partial p/\partial x$ and $\partial p/\partial y$ we solve the system of Equations

3.62 and 3.63 using Cramer's rule

$$\frac{\partial p}{\partial x} = \frac{\begin{vmatrix} \frac{\partial p}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial p}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{vmatrix}}{\begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{vmatrix}} \tag{3.64}$$

The denominator in the above equation is defined as the Jacobian determinant such as

$$J \equiv \frac{\partial (x, y)}{\partial (\xi, \eta)} \equiv \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{vmatrix} \tag{3.65}$$

If we expand the numerator determinant we can write Equation 3.64 as follows

$$\frac{\partial p}{\partial x} = \frac{1}{J}\left[\frac{\partial p}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial p}{\partial \eta}\frac{\partial y}{\partial \xi}\right] \tag{3.66}$$

$$\frac{\partial p}{\partial y} = \frac{1}{J}\left[\frac{\partial p}{\partial \eta}\frac{\partial x}{\partial \xi} - \frac{\partial p}{\partial \xi}\frac{\partial x}{\partial \eta}\right] \tag{3.67}$$

Comparing the system of Equations 3.48, 3.49 and Equations 3.66, 3.67 we can conclude that substituting the metrics of transformation identified by Equations 3.58 into Equations 3.48 and 3.49 gives the same result as Equations 3.66 and 3.67. To complete the transformation we need to obtain the second and mixed derivatives of the arbitrary variable $p = p(x, y)$ with respect to ξ and η . The following expressions are derived using the simple rule for differentiation of a product of two terms

$$\begin{aligned} \frac{\partial^2 p}{\partial \xi^2} &= \frac{\partial}{\partial \xi}\left[\frac{\partial p}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial p}{\partial y}\frac{\partial y}{\partial \xi}\right] \\ &= \frac{\partial p}{\partial x}\frac{\partial^2 x}{\partial \xi^2} + \frac{\partial x}{\partial \xi}\frac{\partial^2 p}{\partial \xi \partial x} + \frac{\partial p}{\partial y}\frac{\partial^2 y}{\partial \xi^2} + \frac{\partial y}{\partial \xi}\frac{\partial^2 p}{\partial \xi \partial y} \end{aligned} \tag{3.68}$$

The second and fourth terms in the above equation involve differentiation with respect

to one variable in the (x, y) system and another variable in the (ξ, η) system. Since this is not appropriate for the coordinate transformation we need to work further with these terms

$$\frac{\partial^2 p}{\partial \xi \partial x} = \frac{\partial}{\partial \xi}\left(\frac{\partial p}{\partial x}\right) \tag{3.69}$$

Using the chain rule for $\partial/\partial\xi$ term, we have

$$\frac{\partial^2 p}{\partial \xi \partial x} = \left(\frac{\partial x}{\partial \xi}\frac{\partial}{\partial x} + \frac{\partial y}{\partial \xi}\frac{\partial}{\partial y}\right)\frac{\partial p}{\partial x} \tag{3.70}$$

$$\frac{\partial^2 p}{\partial \xi \partial x} = \frac{\partial^2 p}{\partial x^2}\frac{\partial x}{\partial \xi} + \frac{\partial^2 p}{\partial y \partial x}\frac{\partial y}{\partial \xi} \tag{3.71}$$

Similarly

$$\frac{\partial^2 p}{\partial \xi \partial y} = \frac{\partial^2 p}{\partial y^2}\frac{\partial y}{\partial \xi} + \frac{\partial^2 p}{\partial y \partial x}\frac{\partial x}{\partial \xi} \tag{3.72}$$

Substituting Equations 3.71 and 3.72 into 3.68 and following the same steps explained above for the second derivative with respect to y and mixed derivative, we finally obtain the system of Equations as

$$\frac{\partial^2 p}{\partial \xi^2} = \frac{\partial p}{\partial x}\frac{\partial^2 x}{\partial \xi^2} + \frac{\partial p}{\partial y}\frac{\partial^2 y}{\partial \xi^2} + \frac{\partial^2 p}{\partial x^2}\left(\frac{\partial x}{\partial \xi}\right)^2 + \frac{\partial^2 p}{\partial y^2}\left(\frac{\partial y}{\partial \xi}\right)^2 + 2\frac{\partial^2 p}{\partial x \partial y}\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \xi} \tag{3.73}$$

$$\frac{\partial^2 p}{\partial \eta^2} = \frac{\partial p}{\partial x}\frac{\partial^2 x}{\partial \eta^2} + \frac{\partial p}{\partial y}\frac{\partial^2 y}{\partial \eta^2} + \frac{\partial^2 p}{\partial x^2}\left(\frac{\partial x}{\partial \eta}\right)^2 + \frac{\partial^2 p}{\partial y^2}\left(\frac{\partial y}{\partial \eta}\right)^2 + 2\frac{\partial^2 p}{\partial x \partial y}\frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \eta} \tag{3.74}$$

$$\frac{\partial^2 x}{\partial \xi \partial \eta} + \frac{\partial p}{\partial y}\frac{\partial^2 y}{\partial \xi \partial \eta} + \frac{\partial^2 p}{\partial x^2}\frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta} + \frac{\partial^2 p}{\partial y^2}\frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta} + \frac{\partial^2 p}{\partial x \partial y}\left(\frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} + \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta}\right) \tag{3.75}$$

If we solve the system of Equations 3.73, 3.74 and 3.75 for the three unknowns $\partial^2 p/\partial x^2$,

$\partial^2 p/\partial y^2$ and $\partial^2 p/\partial x \partial y$ using Cramer's rule and replace the terms $\partial p/\partial x$ and $\partial p/\partial y$ with Equations 3.66 and 3.67 we obtain the ultimate expressions for the coordinate transformation as follows

$$\begin{aligned}\frac{\partial^2 p}{\partial x^2} = &\left(\frac{1}{J}\frac{\partial y}{\partial \eta}\right)^2 \frac{\partial^2 p}{\partial \xi^2} + \left(\frac{1}{J}\frac{\partial y}{\partial \xi}\right)^2 \frac{\partial^2 p}{\partial \eta^2} - 2\frac{1}{J^2}\frac{\partial y}{\partial \xi}\frac{\partial y}{\eta}\frac{\partial^2 p}{\partial \xi \partial \eta} \\ &+ \frac{1}{J^3}\left[\frac{\partial y}{\partial \xi}\left(\frac{\partial^2 x}{\partial \xi^2}\left(\frac{\partial y}{\partial \eta}\right)^2 + \frac{\partial^2 x}{\partial \eta^2}\left(\frac{\partial y}{\partial \xi}\right)^2 - 2\frac{\partial y}{\partial \eta}\frac{\partial y}{\partial \xi}\frac{\partial^2 x}{\partial \xi \partial \eta}\right)\right]\frac{\partial p}{\partial \eta} \\ &- \frac{1}{J^3}\left[\frac{\partial x}{\partial \xi}\left(\frac{\partial^2 y}{\partial \xi^2}\left(\frac{\partial y}{\partial \eta}\right)^2 + \frac{\partial^2 y}{\partial \eta^2}\left(\frac{\partial y}{\partial \xi}\right)^2 - 2\frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta}\frac{\partial^2 y}{\partial \xi \partial \eta}\right)\right]\frac{\partial p}{\partial \eta} \\ &+ \frac{1}{J^3}\left[\frac{\partial x}{\partial \eta}\left(\frac{\partial^2 y}{\partial \xi^2}\left(\frac{\partial y}{\partial \eta}\right)^2 + \frac{\partial^2 y}{\partial \eta^2}\left(\frac{\partial y}{\partial \xi}\right)^2 - 2\frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta}\frac{\partial^2 y}{\partial \xi \partial \eta}\right)\right]\frac{\partial p}{\partial \xi} \\ &- \frac{1}{J^3}\left[\frac{\partial y}{\partial \eta}\left(\frac{\partial^2 x}{\partial \xi^2}\left(\frac{\partial y}{\partial \eta}\right)^2 + \frac{\partial^2 x}{\partial \eta^2}\left(\frac{\partial y}{\partial \xi}\right)^2 - 2\frac{\partial y}{\partial \xi}\frac{\partial y}{\partial \eta}\frac{\partial^2 x}{\partial \xi \partial \eta}\right)\right]\frac{\partial p}{\partial \xi}\end{aligned} \tag{3.76}$$

$$\begin{aligned}\frac{\partial^2 p}{\partial y^2} = &\left(\frac{1}{J}\frac{\partial x}{\partial \eta}\right)^2 \frac{\partial^2 p}{\partial \xi^2} + \left(\frac{1}{J}\frac{\partial x}{\partial \xi}\right)^2 \frac{\partial^2 p}{\partial \eta^2} - 2\frac{1}{J^2}\frac{\partial x}{\partial \xi}\frac{\partial x}{\eta}\frac{\partial^2 p}{\partial \xi \partial \eta} \\ &+ \frac{1}{J^3}\left[\frac{\partial y}{\partial \xi}\left(\frac{\partial^2 x}{\partial \xi^2}\left(\frac{\partial x}{\partial \eta}\right)^2 + \frac{\partial^2 x}{\partial \eta^2}\left(\frac{\partial x}{\partial \xi}\right)^2 - 2\frac{\partial x}{\partial \eta}\frac{\partial x}{\partial \xi}\frac{\partial^2 x}{\partial \xi \partial \eta}\right)\right]\frac{\partial p}{\partial \eta} \\ &- \frac{1}{J^3}\left[\frac{\partial x}{\partial \xi}\left(\frac{\partial^2 y}{\partial \xi^2}\left(\frac{\partial x}{\partial \eta}\right)^2 + \frac{\partial^2 y}{\partial \eta^2}\left(\frac{\partial x}{\partial \xi}\right)^2 - 2\frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta}\frac{\partial^2 y}{\partial \xi \partial \eta}\right)\right]\frac{\partial p}{\partial \eta} \\ &+ \frac{1}{J^3}\left[\frac{\partial x}{\partial \eta}\left(\frac{\partial^2 y}{\partial \xi^2}\left(\frac{\partial x}{\partial \eta}\right)^2 + \frac{\partial^2 y}{\partial \eta^2}\left(\frac{\partial x}{\partial \xi}\right)^2 - 2\frac{\partial x}{\partial \eta}\frac{\partial x}{\partial \xi}\frac{\partial^2 y}{\partial \xi \partial \eta}\right)\right]\frac{\partial p}{\partial \xi} \\ &- \frac{1}{J^3}\left[\frac{\partial y}{\partial \eta}\left(\frac{\partial^2 x}{\partial \xi^2}\left(\frac{\partial x}{\partial \eta}\right)^2 + \frac{\partial^2 x}{\partial \eta^2}\left(\frac{\partial x}{\partial \xi}\right)^2 - 2\frac{\partial x}{\partial \xi}\frac{\partial x}{\partial \eta}\frac{\partial^2 x}{\partial \xi \partial \eta}\right)\right]\frac{\partial p}{\partial \xi}\end{aligned} \tag{3.77}$$

$$\begin{aligned}\frac{\partial^2 p}{\partial x \partial y} = &-\frac{1}{J^2}\frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \eta}\frac{\partial^2 p}{\partial \xi^2} - \frac{1}{J^2}\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \xi}\frac{\partial^2 p}{\partial \eta^2} + \frac{1}{J^2}\left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} + \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}\right)\frac{\partial^2 p}{\partial \xi \partial \eta} \\ &+ \frac{1}{J^3}\left[\frac{\partial x}{\partial \xi}\left(\frac{\partial^2 y}{\partial \xi^2}\frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \eta} + \frac{\partial^2 y}{\partial \eta^2}\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \xi} - \frac{\partial^2 y}{\partial \xi \partial \eta}\left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} + \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}\right)\right)\right]\frac{\partial p}{\partial \eta} \\ &- \frac{1}{J^3}\left[\frac{\partial y}{\partial \xi}\left(\frac{\partial^2 x}{\partial \xi^2}\frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \eta} + \frac{\partial^2 x}{\partial \eta^2}\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \xi} - \frac{\partial^2 x}{\partial \xi \partial \eta}\left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} + \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}\right)\right)\right]\frac{\partial p}{\partial \eta} \\ &+ \frac{1}{J^3}\left[\frac{\partial y}{\partial \eta}\left(\frac{\partial^2 x}{\partial \xi^2}\frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \eta} + \frac{\partial^2 x}{\partial \eta^2}\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \xi} - \frac{\partial^2 x}{\partial \xi \partial \eta}\left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} + \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}\right)\right)\right]\frac{\partial p}{\partial \xi} \\ &- \frac{1}{J^3}\left[\frac{\partial x}{\partial \eta}\left(\frac{\partial^2 y}{\partial \xi^2}\frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \eta} + \frac{\partial^2 y}{\partial \eta^2}\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \xi} - \frac{\partial^2 y}{\partial \xi \partial \eta}\left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} + \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}\right)\right)\right]\frac{\partial p}{\partial \xi}\end{aligned} \tag{3.78}$$

Now, recalling the velocity Equations 3.44, 3.45 and vorticity transport Equation 3.46, if we substitute Equations 3.66, 3.67, 3.76, 3.77 and 3.78 into Equations 3.44, 3.45, 3.46 and make necessary arrangements we obtain the transformed form of the governing equations in the new coordinate system (ξ, η) as follows

$$\begin{aligned}&\frac{\partial u}{\partial \eta}\left[\frac{\partial y}{\partial \xi}\left(\frac{\partial^2 x}{\partial \xi^2}\alpha + \frac{\partial^2 x}{\partial \eta^2}\beta - 2\frac{\partial^2 x}{\partial \xi \partial \eta}\gamma\right) - \frac{\partial x}{\partial \xi}\left(\frac{\partial^2 y}{\partial \xi^2}\alpha + \frac{\partial^2 y}{\partial \eta^2}\beta - 2\frac{\partial^2 y}{\partial \xi \partial \eta}\gamma\right)\right] \\ &+ \frac{\partial u}{\partial \xi}\left[\frac{\partial x}{\partial \eta}\left(\frac{\partial^2 y}{\partial \xi^2}\alpha + \frac{\partial^2 y}{\partial \eta^2}\beta - 2\frac{\partial^2 y}{\partial \xi \partial \eta}\gamma\right) - \frac{\partial y}{\partial \xi}\left(\frac{\partial^2 x}{\partial \xi^2}\alpha + \frac{\partial^2 x}{\partial \eta^2}\beta - 2\frac{\partial^2 x}{\partial \xi \partial \eta}\gamma\right)\right] \\ &+ \frac{\partial^2 u}{\partial \xi^2}J\alpha + \frac{\partial^2 u}{\partial \eta^2}J\beta - 2\frac{\partial^2 u}{\partial \xi \partial \eta}J\gamma = J^2\left(\frac{\partial \Omega}{\partial \xi}\frac{\partial x}{\partial \eta} - \frac{\partial \Omega}{\partial \eta}\frac{\partial x}{\partial \xi}\right)\end{aligned} \tag{3.79}$$

$$\begin{aligned}&\frac{\partial v}{\partial \eta}\left[\frac{\partial y}{\partial \xi}\left(\frac{\partial^2 x}{\partial \xi^2}\alpha + \frac{\partial^2 x}{\partial \eta^2}\beta - 2\frac{\partial^2 x}{\partial \xi \partial \eta}\gamma\right) - \frac{\partial x}{\partial \xi}\left(\frac{\partial^2 y}{\partial \xi^2}\alpha + \frac{\partial^2 y}{\partial \eta^2}\beta - 2\frac{\partial^2 y}{\partial \xi \partial \eta}\gamma\right)\right] \\ &+ \frac{\partial v}{\partial \xi}\left[\frac{\partial x}{\partial \eta}\left(\frac{\partial^2 y}{\partial \xi^2}\alpha + \frac{\partial^2 y}{\partial \eta^2}\beta - 2\frac{\partial^2 y}{\partial \xi \partial \eta}\gamma\right) - \frac{\partial y}{\partial \xi}\left(\frac{\partial^2 x}{\partial \xi^2}\alpha + \frac{\partial^2 x}{\partial \eta^2}\beta - 2\frac{\partial^2 x}{\partial \xi \partial \eta}\gamma\right)\right] \\ &+ \frac{\partial^2 v}{\partial \xi^2}J\alpha + \frac{\partial^2 v}{\partial \eta^2}J\beta - 2\frac{\partial^2 v}{\partial \xi \partial \eta}J\gamma = J^2\left(\frac{\partial \Omega}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial \Omega}{\partial \eta}\frac{\partial y}{\partial \xi}\right)\end{aligned} \tag{3.80}$$

$$u\left(\frac{\partial\Omega}{\partial\xi}\frac{\partial y}{\partial\eta}-\frac{\partial\Omega}{\partial\eta}\frac{\partial y}{\partial\xi}\right)J^2\mathrm{Re}+v\left(\frac{\partial\Omega}{\partial\eta}\frac{\partial x}{\partial\xi}-\frac{\partial\Omega}{\partial\xi}\frac{\partial x}{\partial\eta}\right)J^2\mathrm{Re}=\frac{\partial^2\Omega}{\partial\xi^2}J\alpha+\frac{\partial^2\Omega}{\partial\eta^2}J\beta-2\frac{\partial^2\Omega}{\partial\xi\partial\eta}J\gamma$$
$$+\frac{\partial\Omega}{\partial\eta}\left[\frac{\partial y}{\partial\xi}\left(\frac{\partial^2 x}{\partial\xi^2}\alpha+\frac{\partial^2 x}{\partial\eta^2}\beta-2\frac{\partial^2 x}{\partial\xi\partial\eta}\gamma\right)-\frac{\partial x}{\partial\xi}\left(\frac{\partial^2 y}{\partial\xi^2}\alpha+\frac{\partial^2 y}{\partial\eta^2}\beta-2\frac{\partial^2 y}{\partial\xi\partial\eta}\gamma\right)\right]$$
$$+\frac{\partial\Omega}{\partial\xi}\left[\frac{\partial x}{\partial\eta}\left(\frac{\partial^2 y}{\partial\xi^2}\alpha+\frac{\partial^2 y}{\partial\eta^2}\beta-2\frac{\partial^2 y}{\partial\xi\partial\eta}\gamma\right)-\frac{\partial y}{\partial\xi}\left(\frac{\partial^2 x}{\partial\xi^2}\alpha+\frac{\partial^2 x}{\partial\eta^2}\beta-2\frac{\partial^2 x}{\partial\xi\partial\eta}\gamma\right)\right] \tag{3.81}$$

where

$$\begin{aligned}\alpha&=\left(\tfrac{\partial x}{\partial\eta}\right)^2+\left(\tfrac{\partial y}{\partial\eta}\right)^2\\ \beta&=\left(\tfrac{\partial x}{\partial\xi}\right)^2+\left(\tfrac{\partial y}{\partial\xi}\right)^2\\ \gamma&=\tfrac{\partial x}{\partial\xi}\tfrac{\partial x}{\partial\eta}+\tfrac{\partial y}{\partial\xi}\tfrac{\partial y}{\partial\eta}\\ J&=\tfrac{\partial x}{\partial\xi}\tfrac{\partial y}{\partial\eta}-\tfrac{\partial y}{\partial\xi}\tfrac{\partial x}{\partial\eta}\end{aligned} \tag{3.82}$$

The system of equations derived above is the ultimate form of the transformed governing equations in (ξ,η) space and is used in this study.

3.4.1.2. Boundary Conditions. In order to solve the velocity-vorticity equations we must supply appropriate boundary conditions for the new coordinate system. In the following discussion boundary conditions are classified into three groups: inflow, solid surface and outflow.

Inflow:

At the inflow ξ component of velocity is constant, i.e.$u = constant$ and η component of velocity is zero.

$$\begin{aligned}u&=cons.\\ v&=0\end{aligned} \tag{3.83}$$

Two different vorticity boundary conditions are implemented and the effects of them on the results are investigated in this study. Pressure and deformation distributions are

calculated by using two specifications of vorticity conditions for the inflow boundaries and the outputs are presented separately. First approach is vorticity-free case at the inflow that Pao and Dogherty [3] used in their study. This application is also mentioned in [1] and specified as an alternative approach.

$$\Omega = 0 \tag{3.84}$$

Second approach is the application of the definition of the vorticity defined in Equation

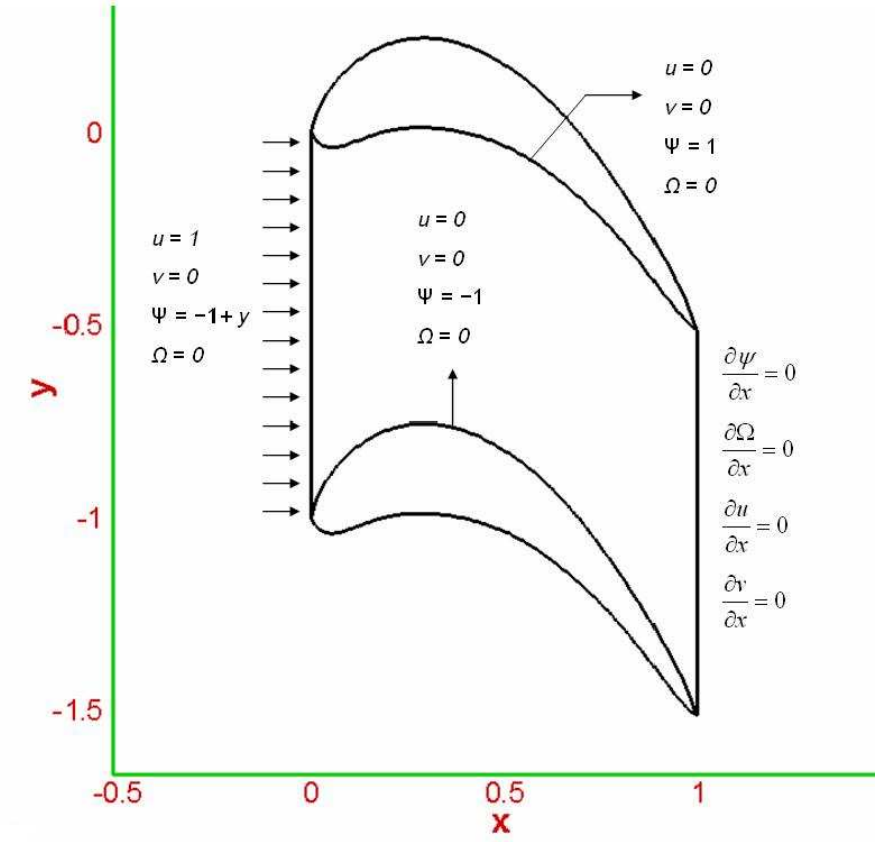


Figure 3.14. Boundary conditions in 2D

3.88. Using Equations 3.83 yields

$$\Omega = \frac{1}{J}\left(\frac{\partial v}{\partial \xi}\frac{\partial y}{\partial \eta} + \frac{\partial u}{\partial \xi}\frac{\partial x}{\partial \eta}\right) \tag{3.85}$$

Solid Surface:

For the solid surface, no-slip and no-penetration conditions yield the following expressions

$$\begin{aligned} u &= 0 \\ v &= 0 \end{aligned} \tag{3.86}$$

The vorticity is defined as the following in Equation 3.33

$$\Omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \tag{3.87}$$

If we transform this equation to the new coordinate system we have an expression as

$$\Omega = \frac{1}{J}\left(\frac{\partial v}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial v}{\partial \eta}\frac{\partial y}{\partial \xi} - \frac{\partial u}{\partial \eta}\frac{\partial x}{\partial \xi} + \frac{\partial u}{\partial \xi}\frac{\partial x}{\partial \eta}\right) \tag{3.88}$$

The velocity components vanish along the solid boundaries, therefore

$$\begin{aligned} \tfrac{\partial u}{\partial \xi} &= 0 \\ \tfrac{\partial v}{\partial \xi} &= 0 \end{aligned} \tag{3.89}$$

By using the above expressions Equation 3.88 reduces to

$$\Omega = -\frac{1}{J}\left(\frac{\partial v}{\partial \eta}\frac{\partial y}{\partial \xi} + \frac{\partial u}{\partial \eta}\frac{\partial x}{\partial \xi}\right) \tag{3.90}$$

where J is defined in Equation 3.59.

Outflow:

In this study the values of the velocity components and the vorticity at the outflow are extrapolated from the interior solution.

3.4.2. Two-Dimensional Analysis Using Stream Function-Vorticity Formulation

For a two dimensional, incompressible flow a function which relates the velocities can be defined as

$$u = \frac{\partial \psi}{\partial y} \tag{3.91}$$

$$v = -\frac{\partial \psi}{\partial x} \tag{3.92}$$

then the continuity equation is identically satisfied. Such a function is known as the stream function. From a physical point of view, the lines of constant ψ represent stream lines that are everywhere tangent to the velocities and the difference in the values of ψ between two streamlines gives the volumetric flowrate between the two. If we substitute Equations 3.91 and 3.92 into the definition of vorticity expressed in Equation 3.33, one can reach

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\Omega \tag{3.93}$$

This equation is known as the stream function equation and is classified as an elliptic PDE. The unknown is the stream function ψ, whose Ω is provided from the solution of the Vorticity Transport Equation. Once the stream function has been found, the velocity components may be determined from Equations 3.91 and 3.92. By using the nondimensional quantities defined in the previous chapter the vorticity transport and stream function equations can be expressed in a nondimensional form as

$$\frac{\partial^2 \psi^*}{\partial x^{*2}} + \frac{\partial^2 \psi^*}{\partial y^{*2}} = -\Omega^* \tag{3.94}$$

$$\frac{\partial \Omega^*}{\partial t^*} + \frac{\partial \psi^*}{\partial y^*}\frac{\partial \Omega^*}{\partial x^*} - \frac{\partial \psi^*}{\partial x^*}\frac{\partial \Omega^*}{\partial y^*} = \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 \Omega^*}{\partial x^{*2}} + \frac{\partial^2 \Omega^*}{\partial y^{*2}}\right) \tag{3.95}$$

The vorticity-stream function formulation has the following characteristics

- Since the velocity components are expressed in terms of the stream function in such a way that the continuity equation is identically satisfied, the latter does not need to be considered.
- The pressure term does not appear in the vorticity-stream function formulation which is highly practical. Therefore, the velocity components are obtained initially and if we had to solve for the pressure field, we would have to apply the Poisson equation for pressure.
- By using the vorticity-stream function approach the incompressible Navier-Stokes equations are decoupled into one elliptic equation and one parabolic equation which can be solved sequentially.
- In the vorticity-stream function formulation we employ the no-penetration and no-slip boundary conditions in a sequential instead of simultaneous way. The no-penetration condition is applied during the solution for the stream function and the no-slip condition is applied during the derivation of boundary conditions for the vorticity.
- The vorticity-stream function formulation is not practical in three dimensions because of lackness of a simple stream function in three dimensions.

3.4.2.1. Transformation of The Governing Equations. If we use the derivation of the coordinate transformation explained in the previous chapter we can write the transformed form of the governing equations using stream function-vorticity formulation as the following

Stream Function Equation

$$\frac{\partial\psi}{\partial\eta}\left[\frac{\partial y}{\partial\xi}\left(\frac{\partial^2 x}{\partial\xi^2}\alpha+\frac{\partial^2 x}{\partial\eta^2}\beta-2\frac{\partial^2 x}{\partial\xi\partial\eta}\gamma\right)-\frac{\partial x}{\partial\xi}\left(\frac{\partial^2 y}{\partial\xi^2}\alpha+\frac{\partial^2 y}{\partial\eta^2}\beta-2\frac{\partial^2 y}{\partial\xi\partial\eta}\gamma\right)\right]$$
$$+\frac{\partial\psi}{\partial\xi}\left[\frac{\partial x}{\partial\eta}\left(\frac{\partial^2 y}{\partial\xi^2}\alpha+\frac{\partial^2 y}{\partial\eta^2}\beta-2\frac{\partial^2 y}{\partial\xi\partial\eta}\gamma\right)-\frac{\partial y}{\partial\xi}\left(\frac{\partial^2 x}{\partial\xi^2}\alpha+\frac{\partial^2 x}{\partial\eta^2}\beta-2\frac{\partial^2 x}{\partial\xi\partial\eta}\gamma\right)\right]$$
$$+\frac{\partial^2\psi}{\partial\xi^2}J\alpha+\frac{\partial^2\psi}{\partial\eta^2}J\beta-2\frac{\partial^2\psi}{\partial\xi\partial\eta}J\gamma=-J^3\Omega \tag{3.96}$$

Vorticity Transport Equation

$$\mathrm{Re}\frac{\partial\psi}{\partial\eta}\frac{\partial\Omega}{\partial\xi}J\left(\frac{\partial x}{\partial\xi}\frac{\partial y}{\partial\eta}-\frac{\partial y}{\partial\xi}\frac{\partial x}{\partial\eta}\right)+\mathrm{Re}\frac{\partial\psi}{\partial\xi}\frac{\partial\Omega}{\partial\eta}J\left(\frac{\partial x}{\partial\eta}\frac{\partial y}{\partial\xi}-\frac{\partial y}{\partial\eta}\frac{\partial x}{\partial\xi}\right)=$$
$$+\frac{\partial\Omega}{\partial\eta}\left[\frac{\partial y}{\partial\xi}\left(\frac{\partial^2 x}{\partial\xi^2}\alpha+\frac{\partial^2 x}{\partial\eta^2}\beta-2\frac{\partial^2 x}{\partial\xi\partial\eta}\gamma\right)-\frac{\partial x}{\partial\xi}\left(\frac{\partial^2 y}{\partial\xi^2}\alpha+\frac{\partial^2 y}{\partial\eta^2}\beta-2\frac{\partial^2 y}{\partial\xi\partial\eta}\gamma\right)\right]$$
$$+\frac{\partial\Omega}{\partial\xi}\left[\frac{\partial x}{\partial\eta}\left(\frac{\partial^2 y}{\partial\xi^2}\alpha+\frac{\partial^2 y}{\partial\eta^2}\beta-2\frac{\partial^2 y}{\partial\xi\partial\eta}\gamma\right)-\frac{\partial y}{\partial\eta}\left(\frac{\partial^2 x}{\partial\xi^2}\alpha+\frac{\partial^2 x}{\partial\eta^2}\beta-2\frac{\partial^2 x}{\partial\xi\partial\eta}\gamma\right)\right]$$
$$+\frac{\partial^2\Omega}{\partial\xi^2}J\alpha+\frac{\partial^2\Omega}{\partial\eta^2}J\beta-2\frac{\partial^2\Omega}{\partial\xi\partial\eta}J\gamma \tag{3.97}$$

where α, β, γ and J are defined in Equations 3.82. The system of equations derived above is the ultimate form of the transformed governing equations in (ξ,η) space and is used in this study.

3.4.2.2. Boundary Conditions. In order to solve the stream function and the vorticity transport equations we must supply appropriate boundary conditions for the new coordinate system. In the following discussion boundary conditions are classified into three groups: inflow, solid surface and outflow.

Inflow:

At the inflow the stream function is defined considering ξ component of velocity is constant, i.e. u=constant and η component of velocity is zero. This assumption

yields a linearly changing stream function at the inflow, therefore

$$u = \frac{\partial \psi}{\partial \eta} = const. \tag{3.98}$$

$$v = -\frac{\partial \psi}{\partial \xi} = 0 \tag{3.99}$$

Two different vorticity boundary conditions are implemented and the effects of them on the results are investigated in this study. Pressure and deformation distributions are calculated by using two specifications of vorticity conditions for the inflow boundaries and the outputs are presented separately. First approach is vorticity-free case at the inflow that Pao and Dogherty [3] used in their study. This application is also mentioned in [1] and specified as an alternative approach.

$$\Omega = 0 \tag{3.100}$$

Second approach is the application of the stream function equation defined by 3.96. By using 3.98 we can derive that

$$\frac{\partial^2 \psi}{\partial \eta^2} = 0 \tag{3.101}$$

Second derivative of stream function with respect to ξ and mixed derivative are derived as follows

$$\psi_{2,j} = \psi_{1,j} + \left.\frac{\partial \psi}{\partial \xi}\right|_{1,j} \Delta\xi + \left.\frac{\partial^2 \psi}{\partial \xi^2}\right|_{1,j} \frac{(\Delta\xi)^2}{2} + ... \tag{3.102}$$

At the inflow

$$v_{1,j} = -\left.\frac{\partial \psi}{\partial \xi}\right|_{1,j} = 0 \tag{3.103}$$

Therefore

$$\psi_{2,j} = \psi_{1,j} + \left.\frac{\partial^2 \psi}{\partial \xi^2}\right|_{1,j} \frac{(\Delta\xi)^2}{2} + \mathrm{O}\left(\Delta\xi\right)^3 \tag{3.104}$$

from which

$$\left.\frac{\partial^2 \psi}{\partial \xi^2}\right|_{1,j} = 2\frac{(\psi_{2,j} - \psi_{1,j})}{(\Delta\xi)^2} + \mathrm{O}\left(\Delta\xi\right) \tag{3.105}$$

Mixed derivative is defined as

$$\frac{\partial}{\partial \xi}\left(\frac{\partial \psi}{\partial \eta}\right) = \frac{\left(\frac{\partial \psi}{\partial \eta}\right)_{2,j} - \left(\frac{\partial \psi}{\partial \eta}\right)_{1,j}}{\Delta\xi} \tag{3.106}$$

$$\frac{\partial}{\partial \xi}\left(\frac{\partial \psi}{\partial \eta}\right) = \frac{\psi_{2,j+1} - \psi_{2,j-1} - \psi_{1,j+1} + \psi_{1,j-1}}{2\Delta\eta\Delta\xi} \tag{3.107}$$

If we substitute Equations 3.98 to 3.107 into Equation 3.96, we will obtain the ultimate expression for the vorticity boundary condition at the inflow as the following

$$\begin{aligned}\frac{\partial \psi}{\partial \eta}\left[\frac{\partial y}{\partial \xi}\left(\frac{\partial^2 x}{\partial \xi^2}\alpha + \frac{\partial^2 x}{\partial \eta^2}\beta - 2\frac{\partial^2 x}{\partial \xi \partial \eta}\gamma\right) - \frac{\partial x}{\partial \xi}\left(\frac{\partial^2 y}{\partial \xi^2}\alpha + \frac{\partial^2 y}{\partial \eta^2}\beta - 2\frac{\partial^2 y}{\partial \xi \partial \eta}\gamma\right)\right] \\ + \frac{\partial^2 \psi}{\partial \xi^2} J\alpha - 2\frac{\partial^2 \psi}{\partial \xi \partial \eta} J\gamma = -J^3\Omega\end{aligned} \tag{3.108}$$

where α, β, γ and J are defined in Equations 3.82.

Solid Surface:

Since a solid surface can be considered as a stream line and therefore the stream function is constant, its value may be assigned arbitrarily. For the solid surface, no-slip

and no-penetration conditions yield the following expressions

$$u = \frac{\partial \psi}{\partial \eta} = 0 \tag{3.109}$$

$$v = -\frac{\partial \psi}{\partial \xi} = 0 \tag{3.110}$$

$$\frac{\partial^2 \psi}{\partial \xi^2} = 0 \tag{3.111}$$

For the upper solid surface

$$\frac{\partial^2 \psi}{\partial \eta^2} = \frac{2\left(\psi_{i,n\xi-1} - \psi_{i,n\xi}\right)}{\left(\Delta\eta\right)^2} \tag{3.112}$$

For the lower solid surface

$$\frac{\partial^2 \psi}{\partial \eta^2} = \frac{2\left(\psi_{i,2} - \psi_{i,1}\right)}{\left(\Delta\eta\right)^2} \tag{3.113}$$

And mixed derivative is obtained as follows

$$\frac{\partial}{\partial \xi}\left(\frac{\partial \psi}{\partial \eta}\right) = \frac{\left(\frac{\partial \psi}{\partial \eta}\right)_{i+1} - \left(\frac{\partial \psi}{\partial \eta}\right)_{i-1}}{2\Delta\xi} = 0 \tag{3.114}$$

Substituting Equations 3.109, 3.110, 3.111, 3.112, 3.113 and 3.114 in Equation 3.96 we have

$$\frac{\partial^2 \psi}{\partial \eta^2} J\left[\left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2\right] = -J^3\Omega \tag{3.115}$$

where J is defined in Equation 3.82.

Outflow:

In this study the values of the stream function and the vorticity at the outflow are extrapolated from the interior solution.

3.4.3. Three-Dimensional Analysis Using Velocity-Vorticity Formulation

In this part we discuss only the ultimate form of the transformed governing equations and boundary conditions in 3-D. Detailed derivations and explanations are presented in appendix A.

For the flow of incompressible Newtonian fluids the governing equations for the velocity-vorticity formulation in 3-D Cartesian coordinates can be written as

x-component of velocity :

$$\nabla^2 u = \frac{\partial \Omega_y}{\partial z} - \frac{\partial \Omega_z}{\partial y} \tag{3.116}$$

y-component of velocity :

$$\nabla^2 v = \frac{\partial \Omega_z}{\partial x} - \frac{\partial \Omega_x}{\partial z} \tag{3.117}$$

z-component of velocity :

$$\nabla^2 w = \frac{\partial \Omega_x}{\partial y} - \frac{\partial \Omega_y}{\partial x} \tag{3.118}$$

vorticity transport :

$$\nabla . \left(u\Omega \right) - \left(\Omega . \nabla \right) u - \frac{1}{\mathrm{Re}} \nabla^2 \Omega = 0 \tag{3.119}$$

In expanded form the governing equations can be expressed as the following

x-component of velocity :

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = \frac{\partial \Omega_y}{\partial z} - \frac{\partial \Omega_z}{\partial y} \tag{3.120}$$

y-component of velocity :

$$\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} = \frac{\partial \Omega_z}{\partial x} - \frac{\partial \Omega_x}{\partial z} \tag{3.121}$$

z-component of velocity :

$$\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} = \frac{\partial \Omega_x}{\partial y} - \frac{\partial \Omega_y}{\partial x} \tag{3.122}$$

x-component of vorticity transport :

$$u\frac{\partial \Omega_x}{\partial x} + v\frac{\partial \Omega_x}{\partial y} + w\frac{\partial \Omega_x}{\partial z} - \Omega_x\frac{\partial u}{\partial x} - \Omega_y\frac{\partial u}{\partial y} - \Omega_z\frac{\partial u}{\partial z} - \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 \Omega_x}{\partial x^2} + \frac{\partial^2 \Omega_x}{\partial y^2} + \frac{\partial^2 \Omega_x}{\partial z^2}\right) = 0 \tag{3.123}$$

y-component of vorticity transport :

$$u\frac{\partial \Omega_y}{\partial x} + v\frac{\partial \Omega_y}{\partial y} + w\frac{\partial \Omega_y}{\partial z} - \Omega_x\frac{\partial v}{\partial x} - \Omega_y\frac{\partial v}{\partial y} - \Omega_z\frac{\partial v}{\partial z} - \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 \Omega_y}{\partial x^2} + \frac{\partial^2 \Omega_y}{\partial y^2} + \frac{\partial^2 \Omega_y}{\partial z^2}\right) = 0 \tag{3.124}$$

z-component of vorticity transport :

$$u\frac{\partial \Omega_z}{\partial x} + v\frac{\partial \Omega_z}{\partial y} + w\frac{\partial \Omega_z}{\partial z} - \Omega_x\frac{\partial w}{\partial x} - \Omega_y\frac{\partial w}{\partial y} - \Omega_z\frac{\partial w}{\partial z} - \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 \Omega_z}{\partial x^2} + \frac{\partial^2 \Omega_z}{\partial y^2} + \frac{\partial^2 \Omega_z}{\partial z^2}\right) = 0 \tag{3.125}$$

where

$$\begin{aligned} \Omega_x &= \tfrac{\partial w}{\partial y} - \tfrac{\partial v}{\partial z} \\ \Omega_y &= \tfrac{\partial u}{\partial z} - \tfrac{\partial w}{\partial x} \\ \Omega_z &= \tfrac{\partial v}{\partial x} - \tfrac{\partial u}{\partial y} \end{aligned} \tag{3.126}$$

3.4.3.1. Transformation of the Governing Equations. Transformation of the governing equations in three-dimensional velocity-vorticity formulation is derived in Appendix A.

3.4.3.2. Boundary Conditions. The numerical implementation of the boundary conditions discussed in this part is similar to that in two dimensions with some extensions to three dimensions.

Inflow:

For inflow boundaries it is appropriate to specify the velocity field. In this study the velocity field is defined in 3-D as follows

$$\begin{aligned} u &= const. \\ v &= 0 \\ w &= 0 \end{aligned} \tag{3.127}$$

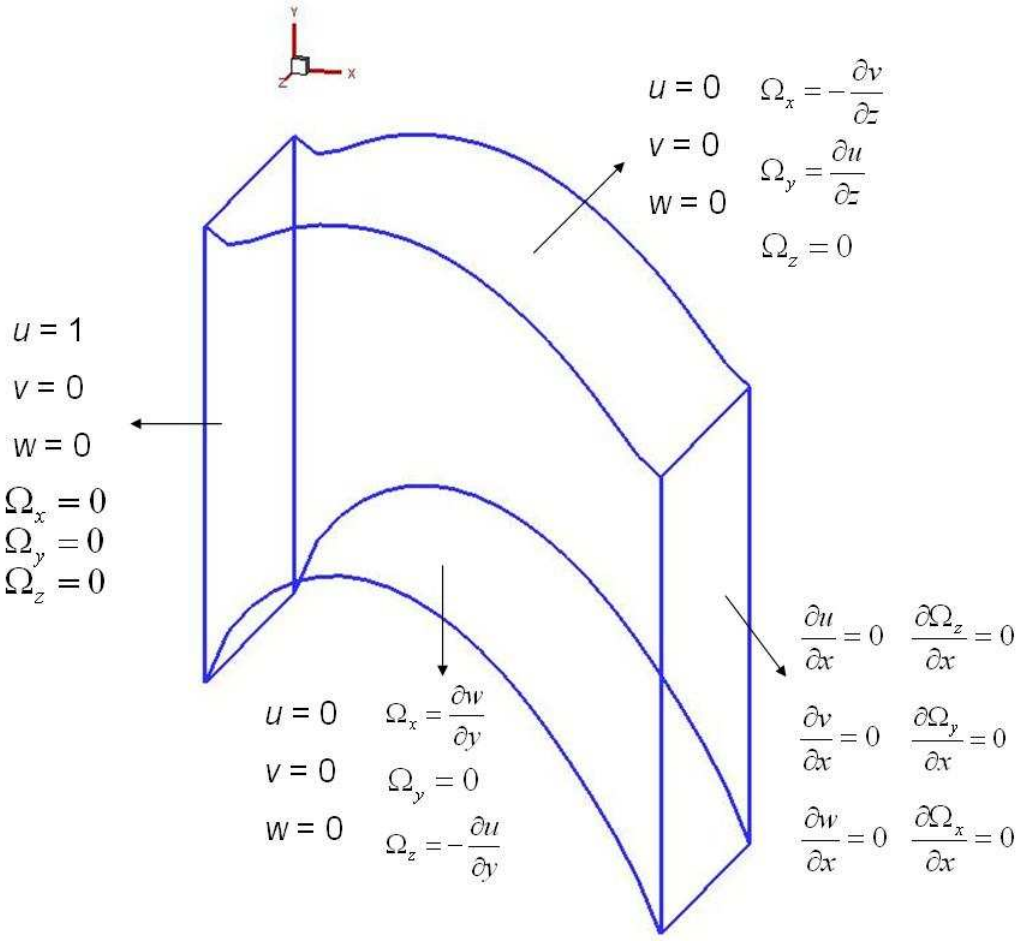


Figure 3.15. Boundary conditions in 3D

At the inflow vorticity-free boundary conditions are considered.

$$\begin{aligned} \Omega_x &= 0 \\ \Omega_y &= 0 \\ \Omega_z &= 0 \end{aligned} \tag{3.128}$$

Solid Surface:

At solid surfaces boundary conditions for the velocity field are given by no-slip and no-penetration,

$$\begin{aligned} u &= 0 \\ v &= 0 \\ w &= 0 \end{aligned} \tag{3.129}$$

and for the vorticity

i) Surface x=constant:

$$\begin{aligned} \Omega_x &= 0 \\ \Omega_y &= -\frac{\partial w}{\partial x} \\ \Omega_z &= \frac{\partial v}{\partial x} \end{aligned} \tag{3.130}$$

ii) Surface y=constant:

$$\begin{aligned} \Omega_x &= \frac{\partial w}{\partial y} \\ \Omega_y &= 0 \\ \Omega_z &= -\frac{\partial u}{\partial y} \end{aligned} \tag{3.131}$$

iii) Surface z=constant:

$$\begin{aligned} \Omega_x &= -\tfrac{\partial v}{\partial z} \\ \Omega_y &= \tfrac{\partial u}{\partial z} \\ \Omega_z &= 0 \end{aligned} \tag{3.132}$$

Outflow:

At outflow Neumann boundary conditions for the velocity components and vorticity are specified.

3.4.4. Pressure Calculation for Two-Dimensional Analysis

Since we analyze the deformation of the turbomachinery blades under aerodynamic loads, it is not necessary to solve the Poisson equation over the entire flow field. Instead, a simpler equation can be solved for the wall pressures. This equation is obtained by applying the tangential momentum equation to the fluid adjacent to the wall surface. For a wall located at $y = 0$ in a Cartesian coordinate system the steady, tangential momentum equation (x momentum equation) reduces to

$$\frac{\partial p}{\partial x}\Big)_{wall} = \frac{1}{\mathrm{Re}}\,\frac{\partial^2 u}{\partial y^2}\Big)_{wall} \tag{3.133}$$

or by taking the y derivative of the vorticity and substituting into Equation 3.133

$$\frac{\partial p}{\partial x}\Big)_{wall} = -\frac{1}{\mathrm{Re}}\,\frac{\partial^2 \Omega}{\partial y^2}\Big)_{wall} \tag{3.134}$$

which can be discretized as

$$\frac{p_{i+1,1} - p_{i-1,1}}{2\Delta x} = -\frac{1}{\mathrm{Re}}\left(\frac{-3\Omega_{i,1} + 4\Omega_{i,2} - \Omega_{i,3}}{2\Delta y}\right) \tag{3.135}$$

In order to apply Equation 3.135, the pressure must be known for at least one point on the wall surface. The pressure at the adjacent point can be determined using a first-order, one-sided difference expression for $\partial p/\partial x$ in Equation 3.134. Then, Equation 3.135 can be used to find the pressure at all other wall points.

3.5. Code Development and Validation

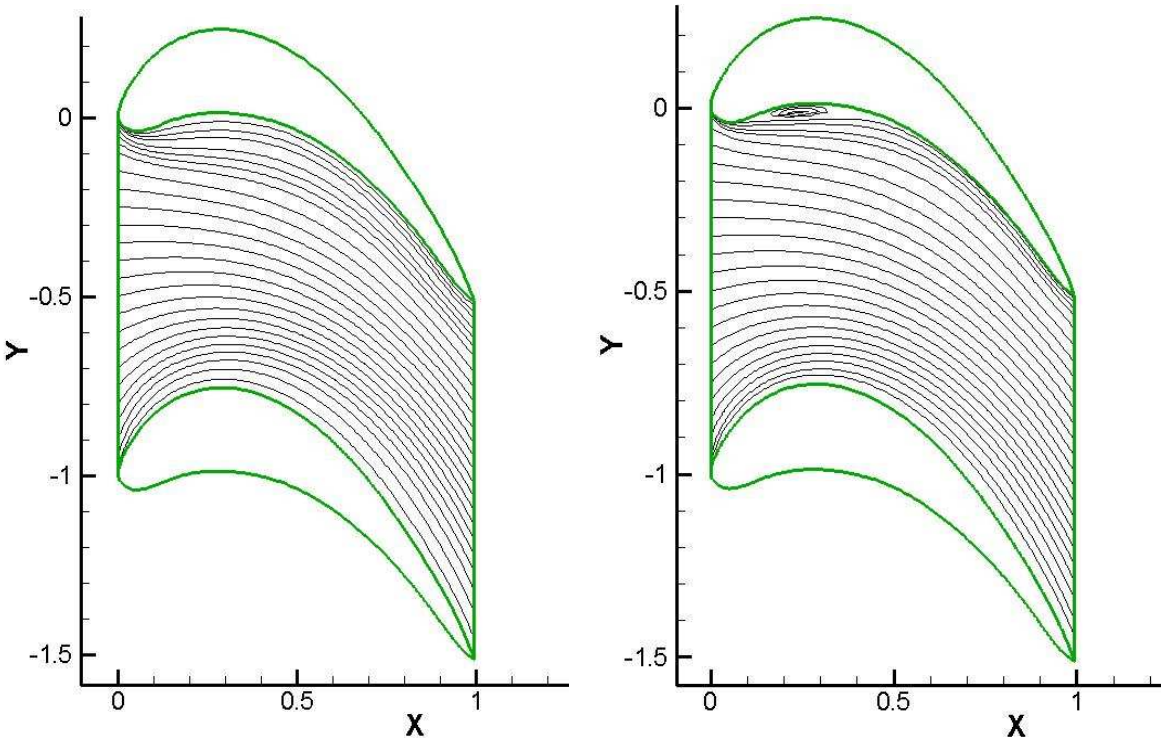
Both stream function-vorticity and velocity-vorticity formulations are applied to the channel flow and lid-driven problems to calculate pressure on the walls. At the inflow two boundary conditions are used as elaborated in 3.4.1.2 and the resulting effects are monitored. Pressure distribution on the wall is calculated by using the approach explained in 3.4.4 in detail. The solutions are achieved for two values of Reynolds number, Re=50 and Re=100 using a fine uniform grid mesh. In the test cases, different solvers and preconditioners are also applied to demonstrate and performances of numerical methods.

4. RESULTS AND DISCUSSION

4.1. Two-Dimensional Analysis

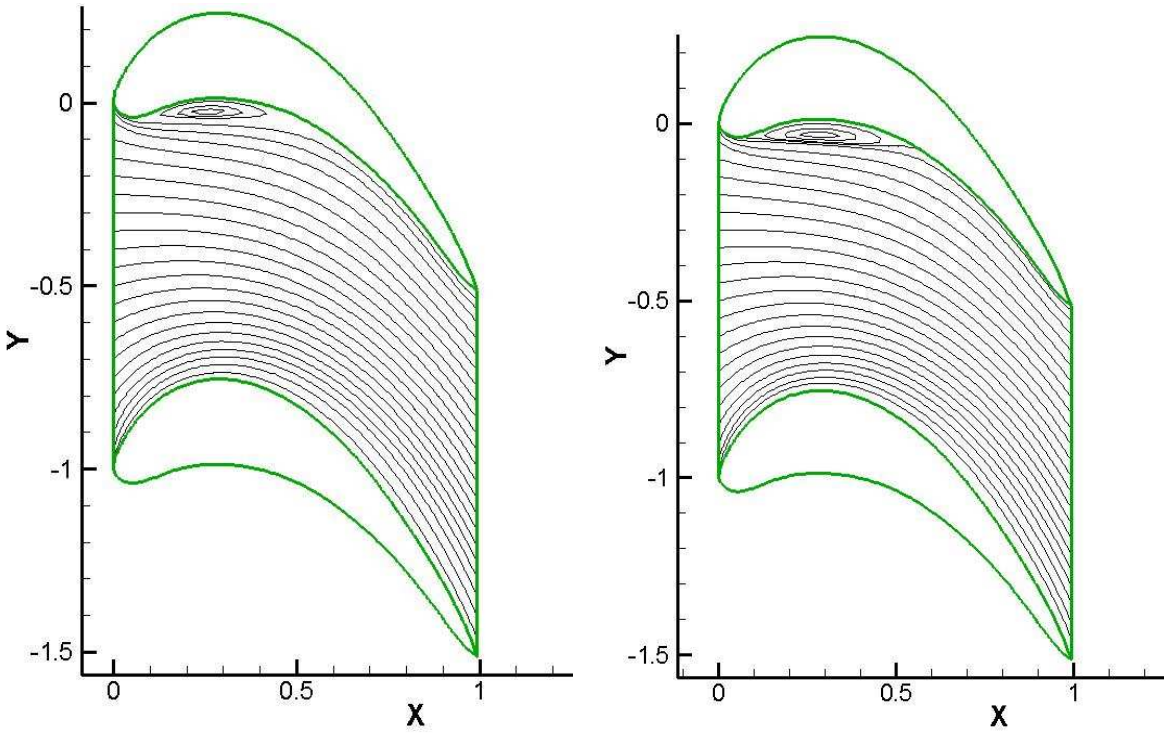
4.1.1. Stream function-vorticity formulation

4.1.1.1. Effect of the Physical Parameter: Reynolds Number. Stream function-vorticity formulation with four different values of Reynolds number (Re=100, Re=300, Re=400 and Re=500) are applied to the problem and the results are presented below. At the inflow, the definition of vorticity derived in 3.108 is used as the vorticity boundary condition. The computational domains consist of 161x121 grid nodes.



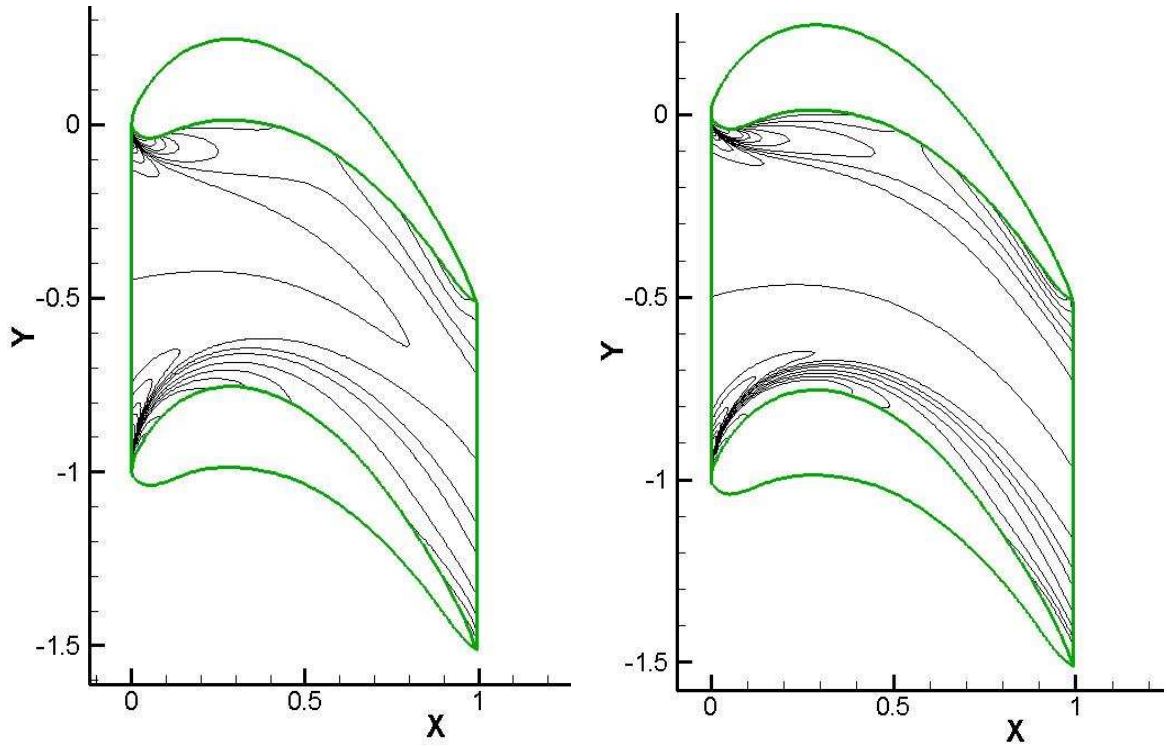
(a) Re: 100 (b) Re: 300

Figure 4.1. Stream function contours



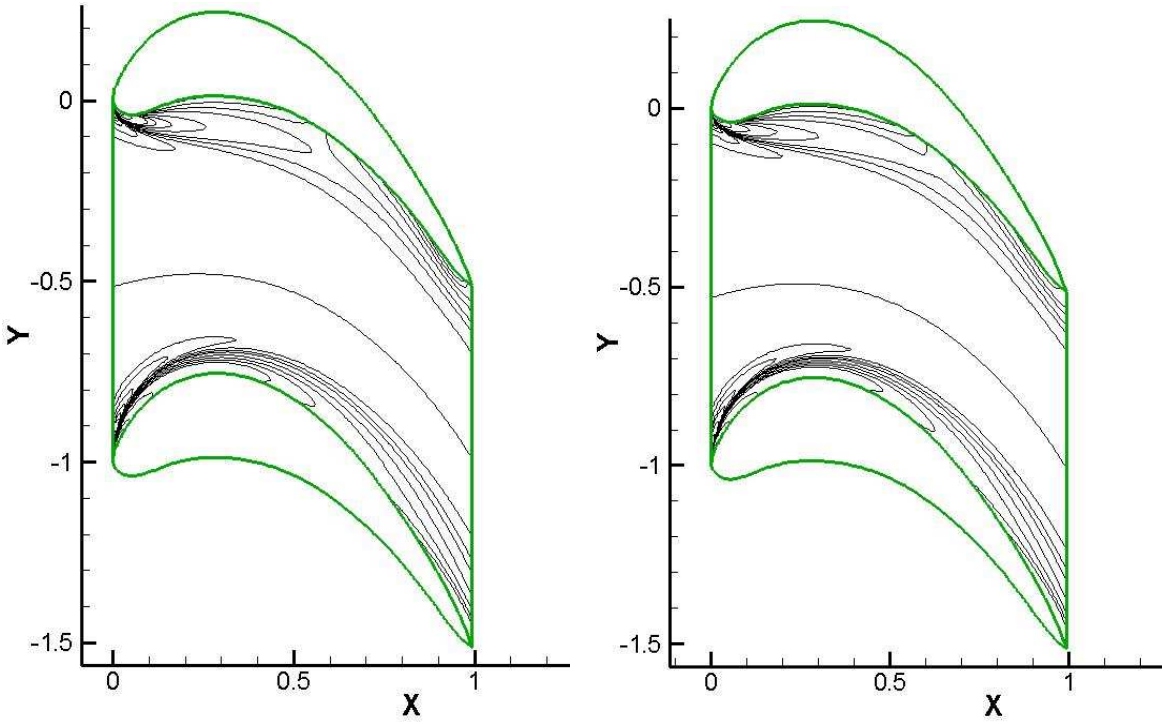
(a) Re: 400 (b) Re: 500

Figure 4.2. Stream function contours



(a) Re: 100 (b) Re: 300

Figure 4.3. Vorticity contours

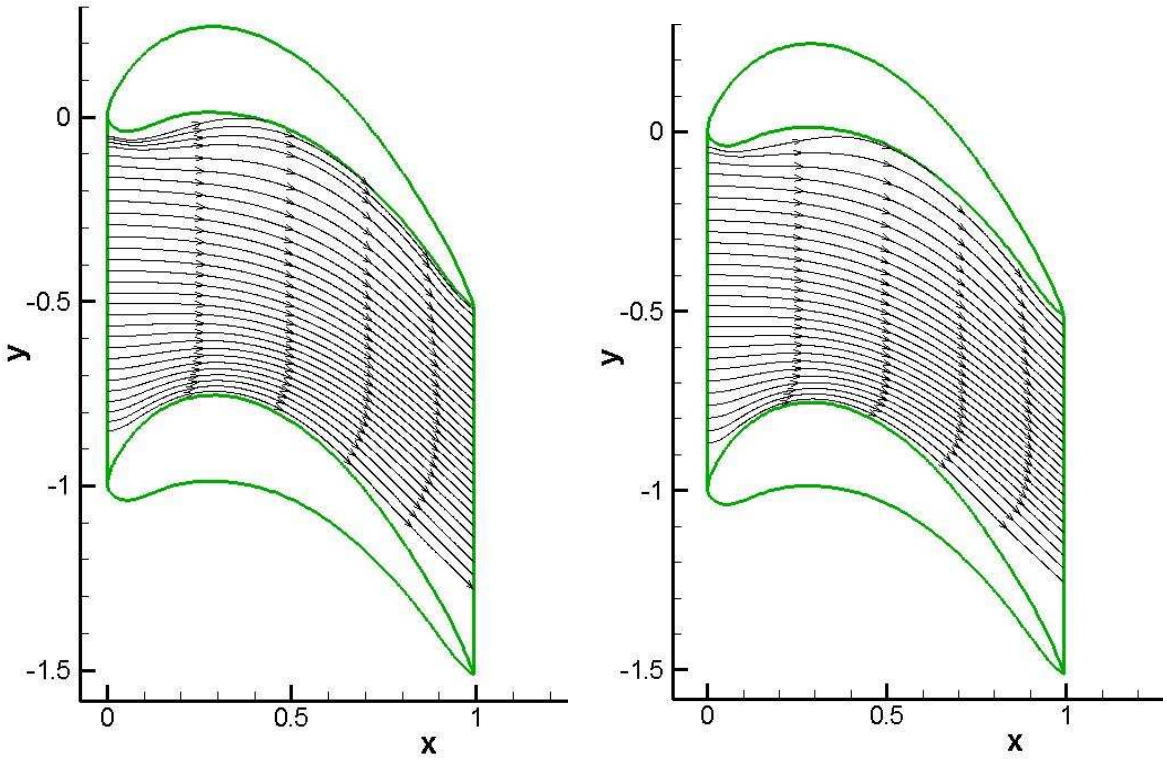


(a) Re: 400 (b) Re: 500

Figure 4.4. Vorticity contours

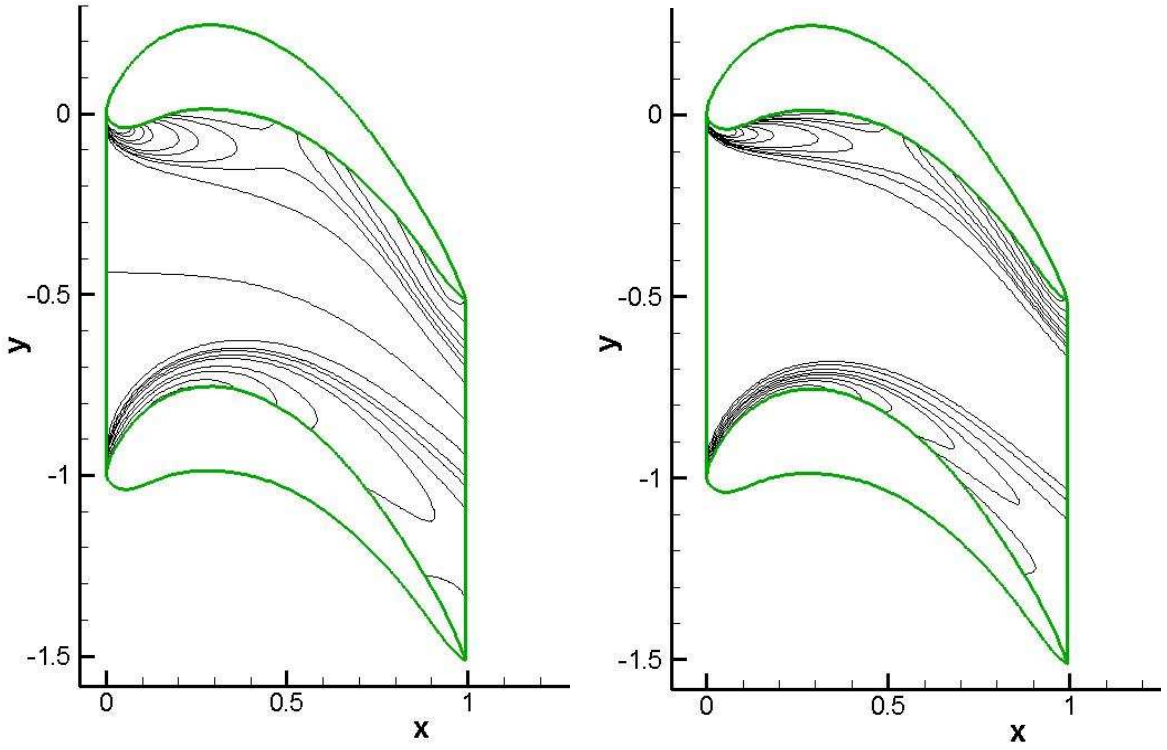
4.1.2. Velocity-vorticity formulation

Velocity-vorticity formulation with two different Reynolds numbers (Re=100, Re=300) and vorticity-free boundary condition at the inflow [3] is applied to the problem and the results are presented in the following.



(a) Re: 100 (b) Re: 300

Figure 4.5. Streamtrace contours



(a) Re: 100 (b) Re: 300

Figure 4.6. Vorticity contours

4.1.3. Pressure Distribution and Blade Deformation Under Aerodynamic Loads

4.1.3.1. Effect of the Physical Parameter: Reynolds Number. Pressure distribution on the blade surface is calculated by using the approach explained in 3.4.4 thoroughly.

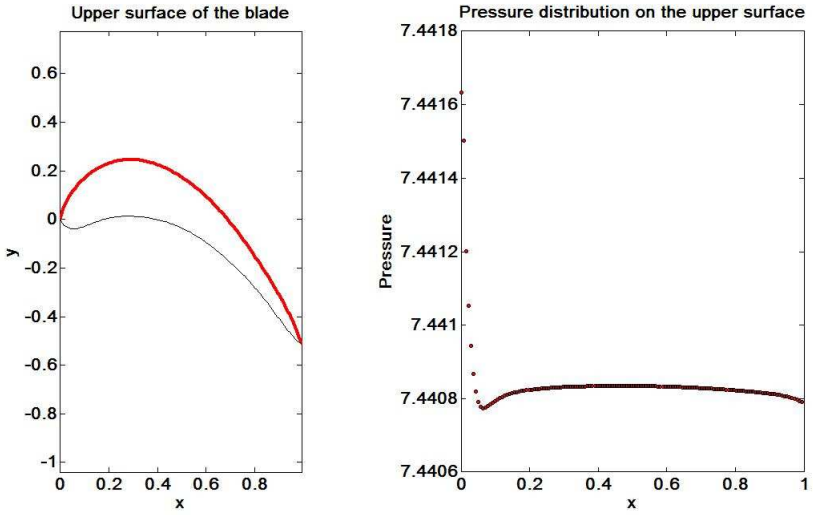


Figure 4.7. Pressure distribution with vorticity bc at the inflow: definition of vorticity (Re=100)

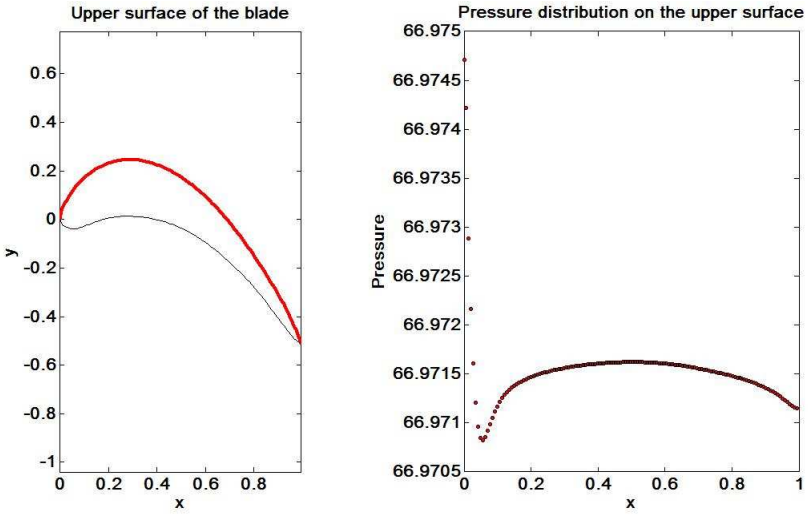


Figure 4.8. Pressure distribution with vorticity bc at the inflow: definition of vorticity (Re=300)

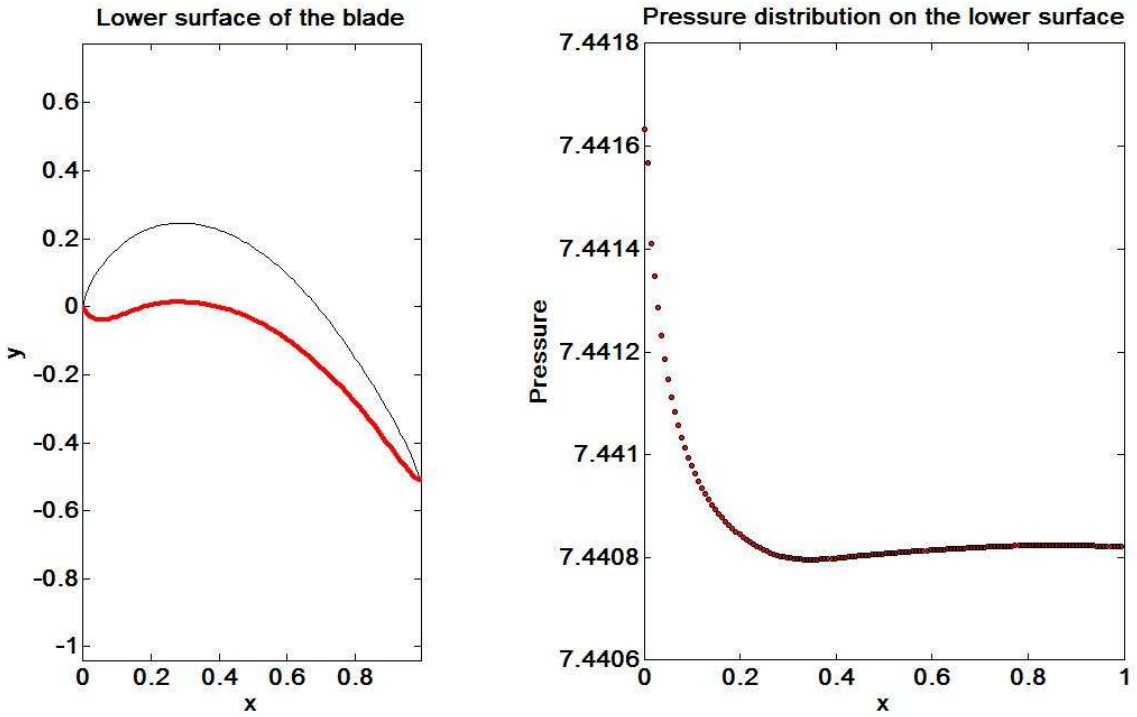


Figure 4.9. Pressure distribution with vorticity bc at the inflow: definition of vorticity (Re=100)

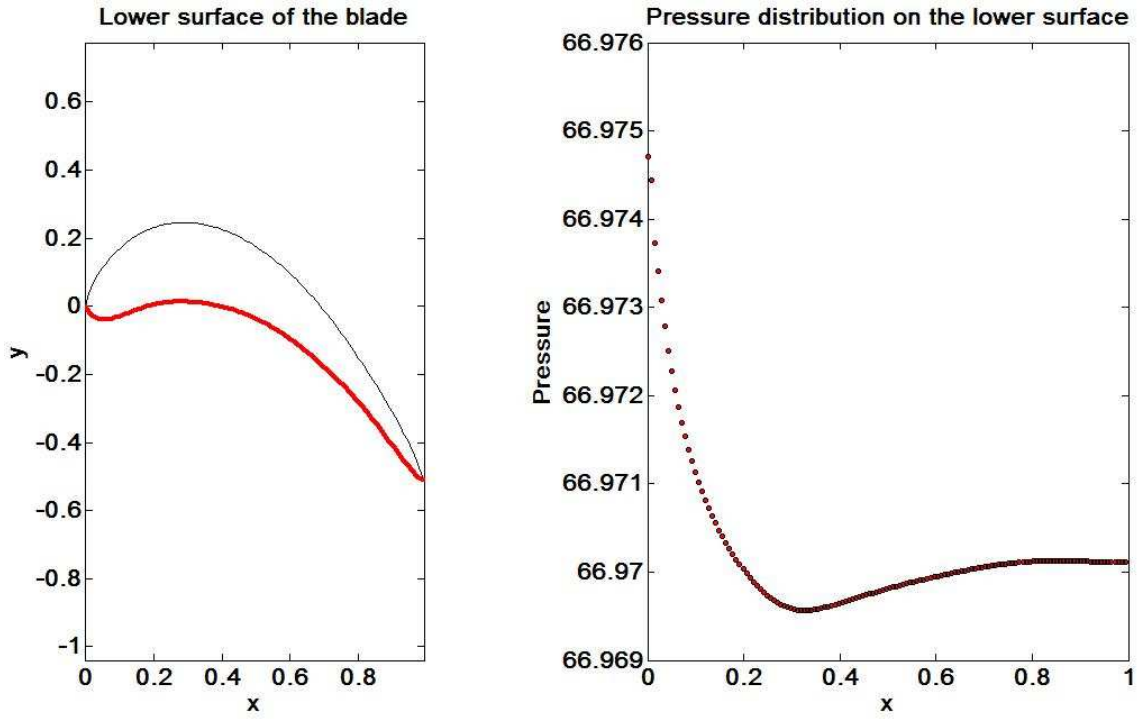


Figure 4.10. Pressure distribution with vorticity bc at the inflow: definition of vorticity (Re=300)

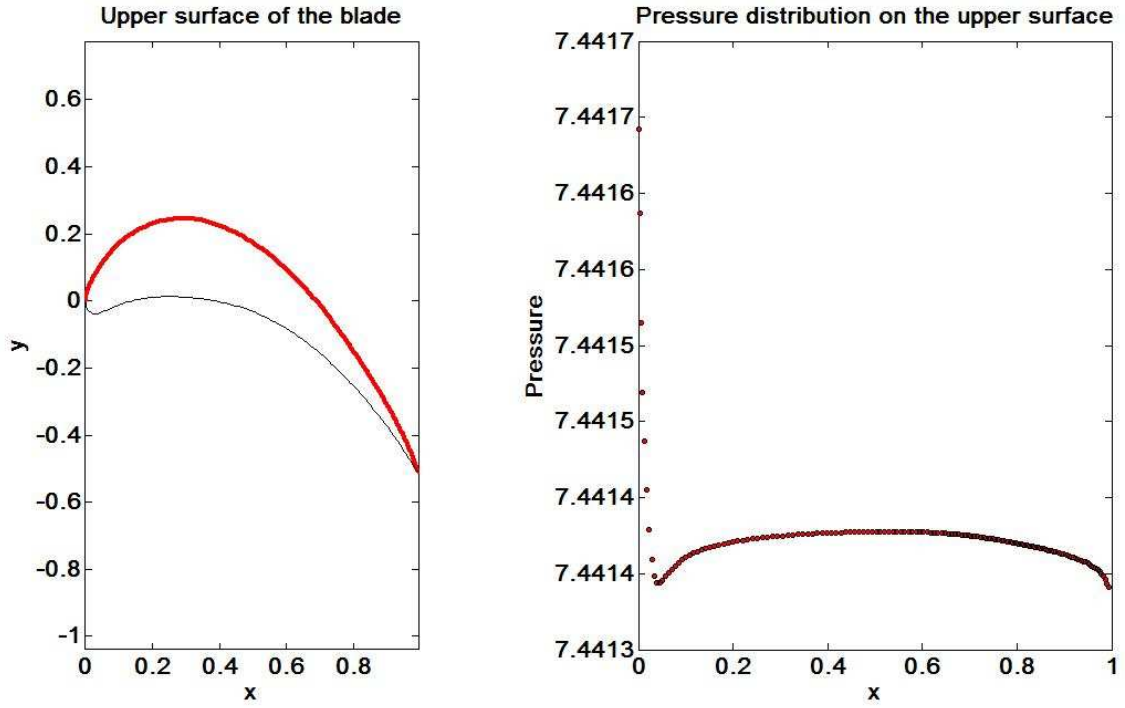


Figure 4.11. Pressure distribution with vorticity bc at the inflow: vorticity-free (Re=100)

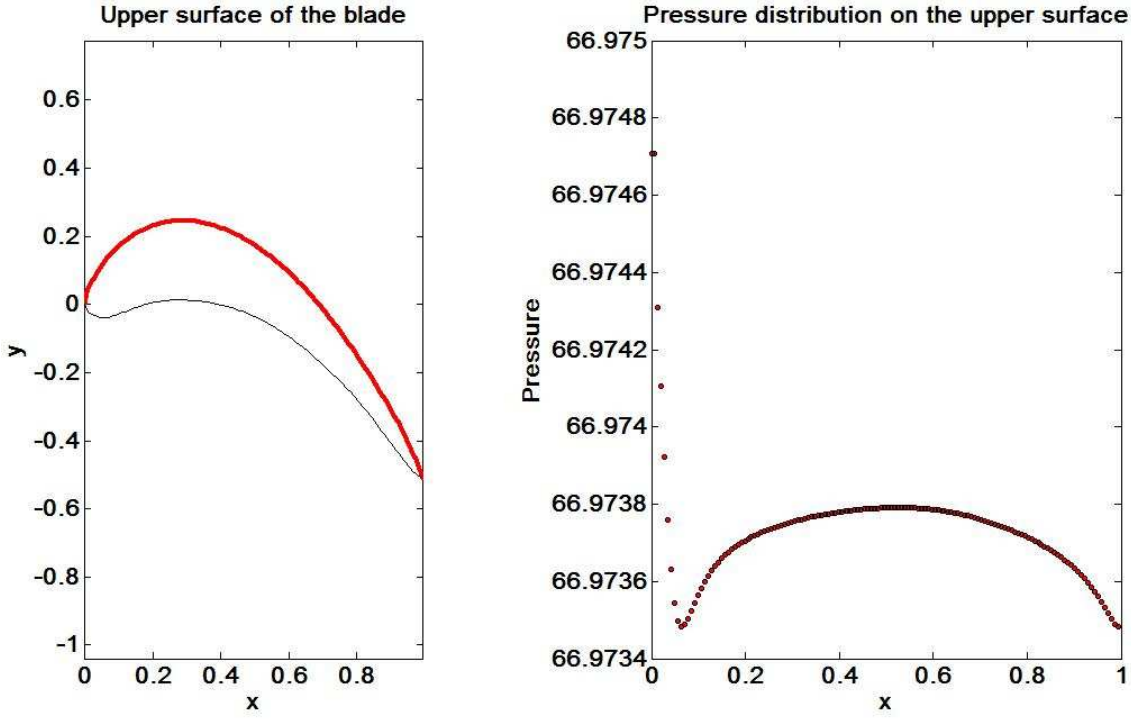


Figure 4.12. Pressure distribution with vorticity bc at the inflow: vorticity-free (Re=300)

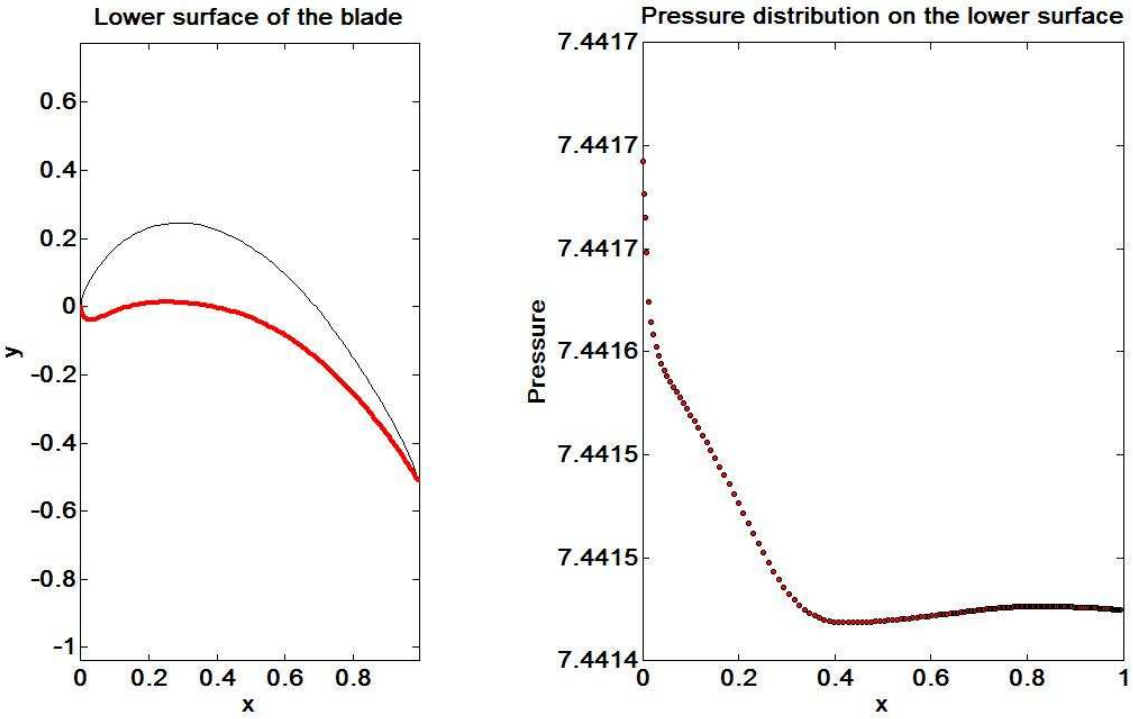


Figure 4.13. Pressure distribution with vorticity bc at the inflow: vorticity-free (Re=100)

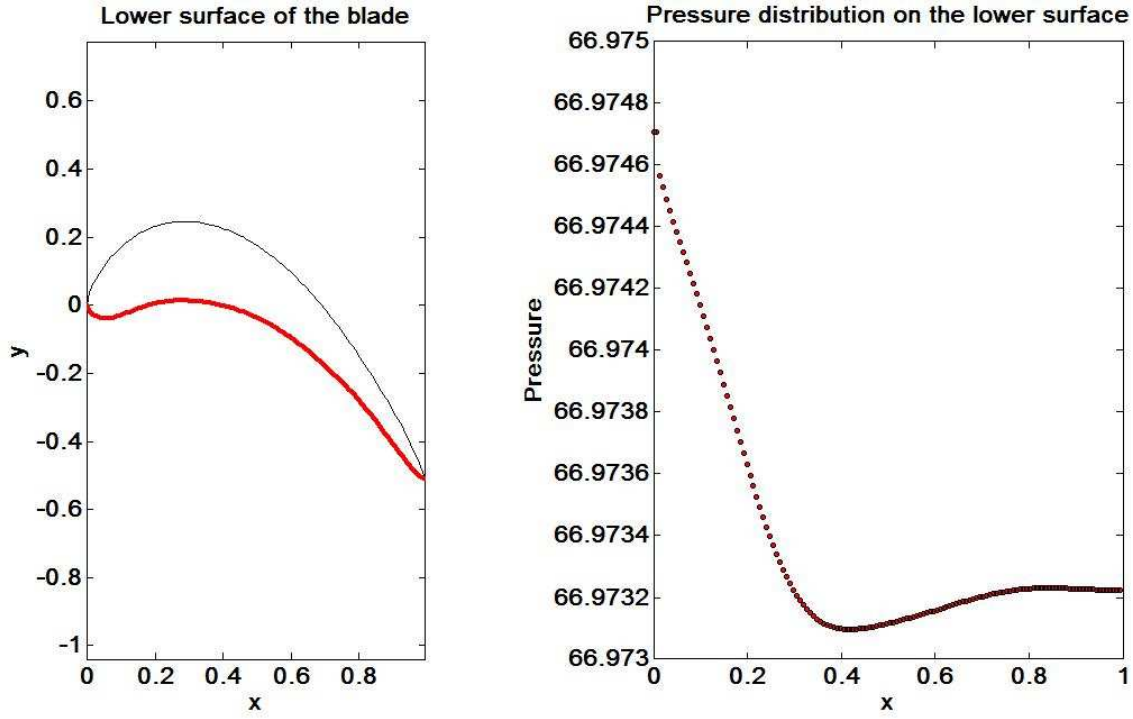


Figure 4.14. Pressure distribution with vorticity bc at the inflow: vorticity-free (Re=300)

4.2. Three-Dimensional Analysis

4.2.1. Velocity-vorticity formulation

Streamtrace and vorticity contours from three-dimensional analysis are presented below.

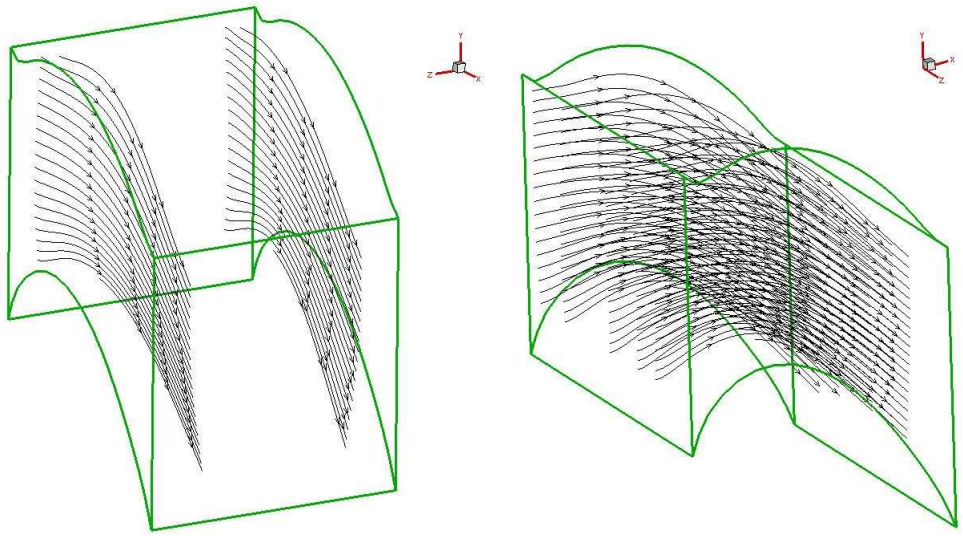


Figure 4.15. Streamtrace contours (Re=10)

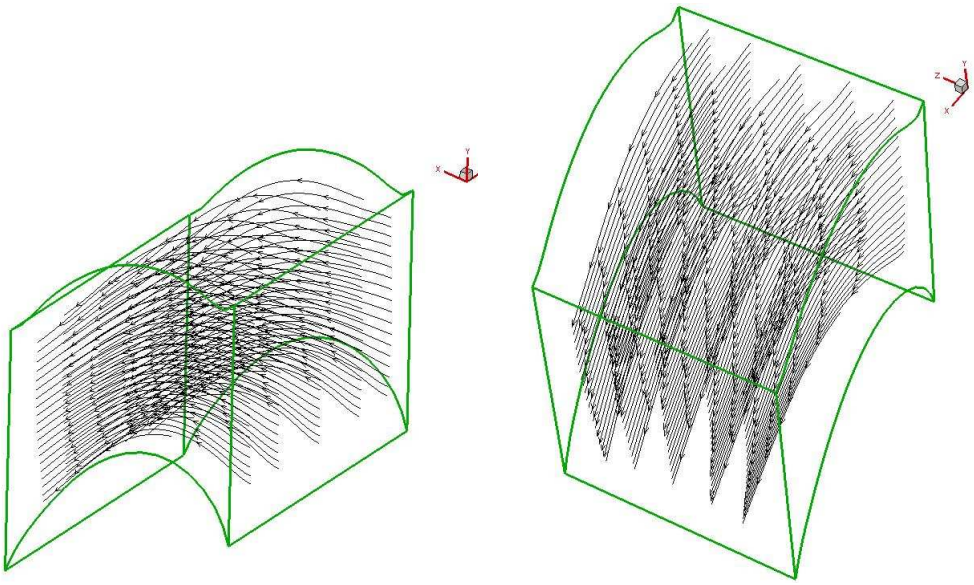


Figure 4.16. Streamtrace contours (Re=10)

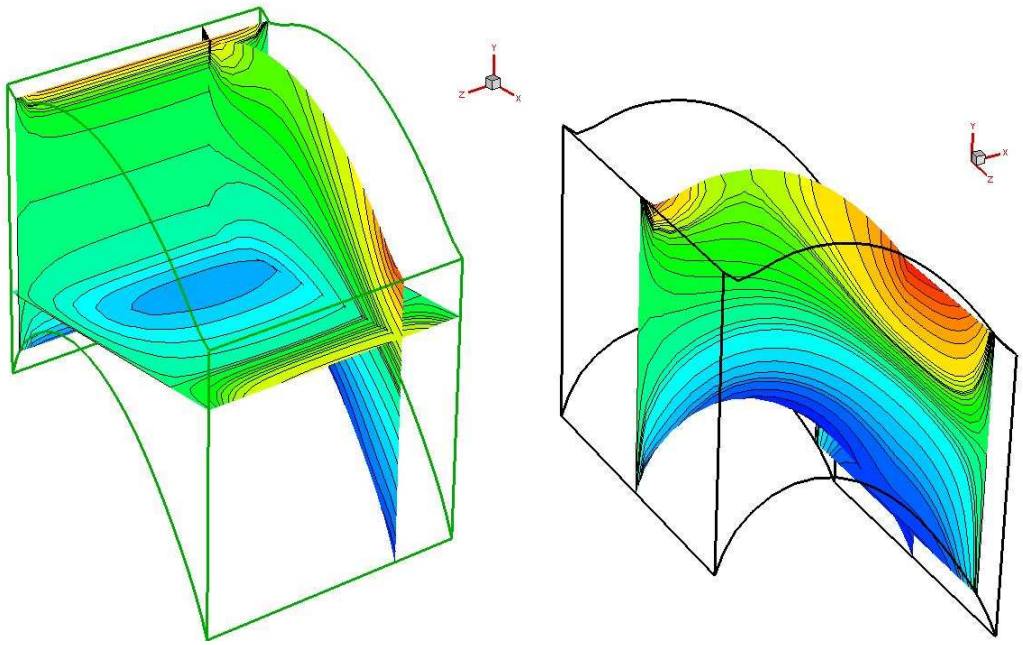


Figure 4.17. Vorticity contours (Re=10)

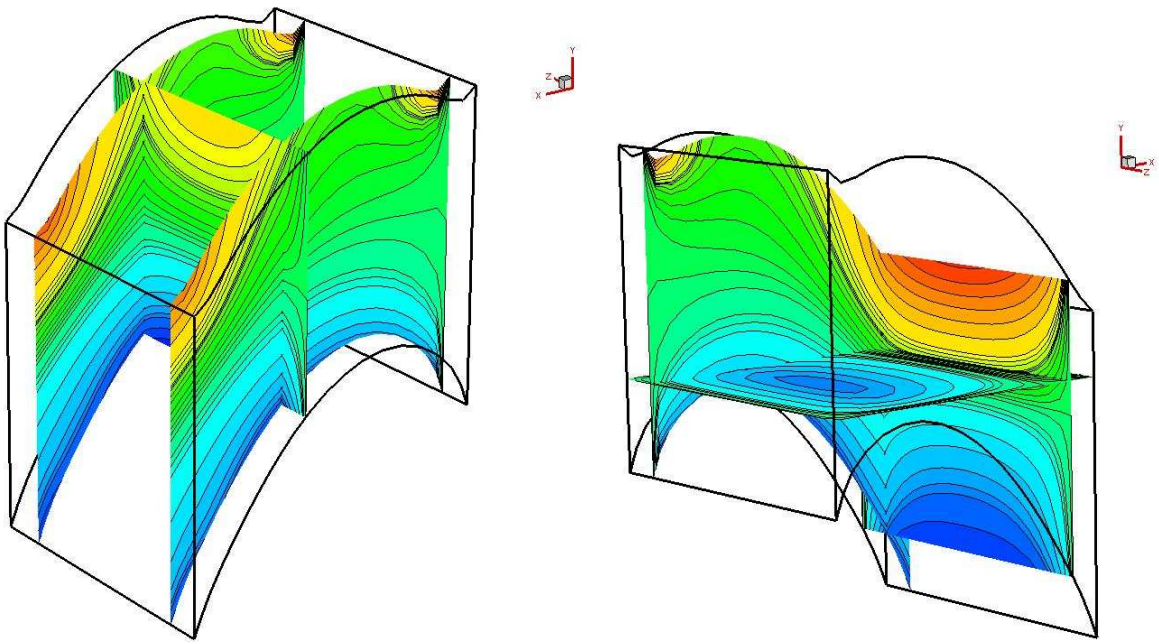


Figure 4.18. Vorticity contours (Re=10)

4.2.2. Blade Deformation Under Aerodynamic Loads

4.2.2.1. Results of the first step of FSI. In this section, deformation of the blade under aerodynamic loads is shown. As mentioned earlier after achieving the pressure field, the pressure values are applied as boundary conditions in ANSYS and resulting displacements are transferred to fluid analysis in order to update the geometry of the problem. This procedure is perpetuated until two norm of the difference of two successive displacement vectors is less than the user specified tolerance.

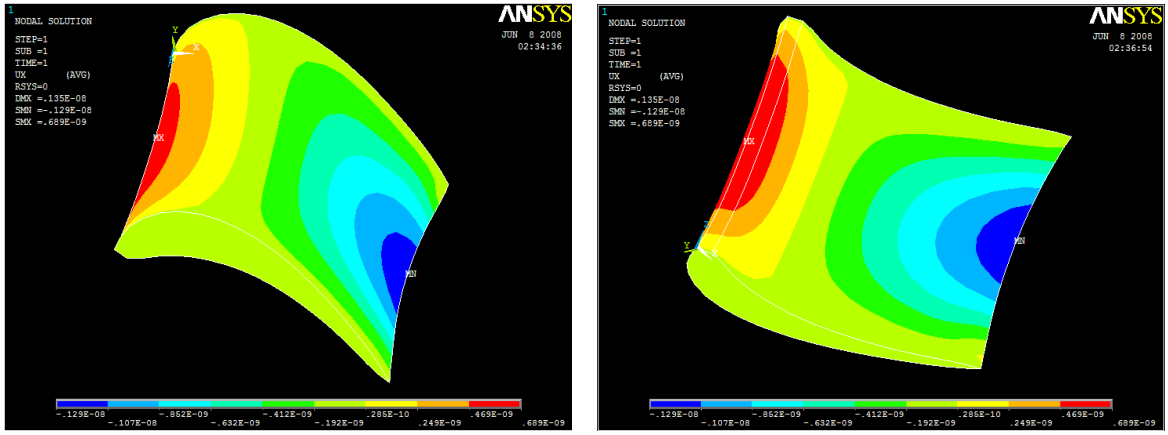


Figure 4.19. X-Component of displacement on the blade

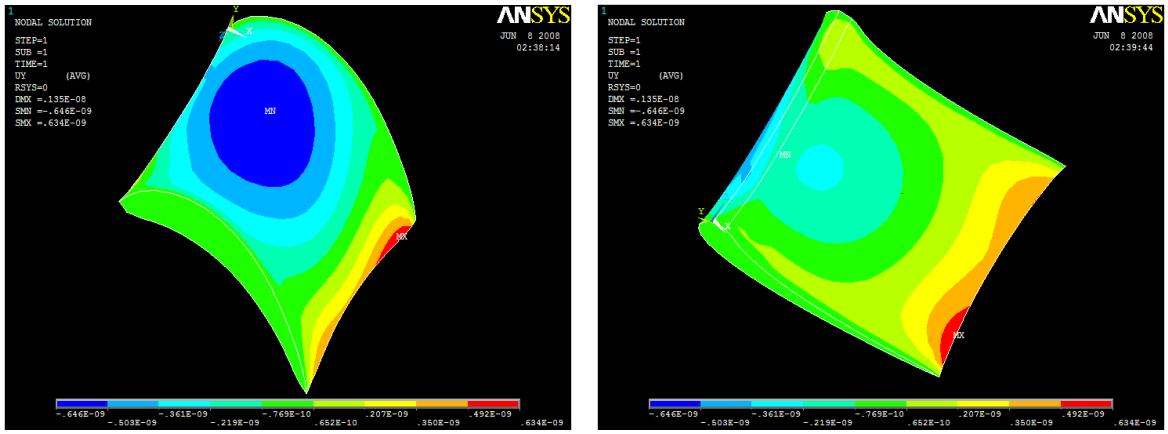


Figure 4.20. Y-Component of displacement on the blade

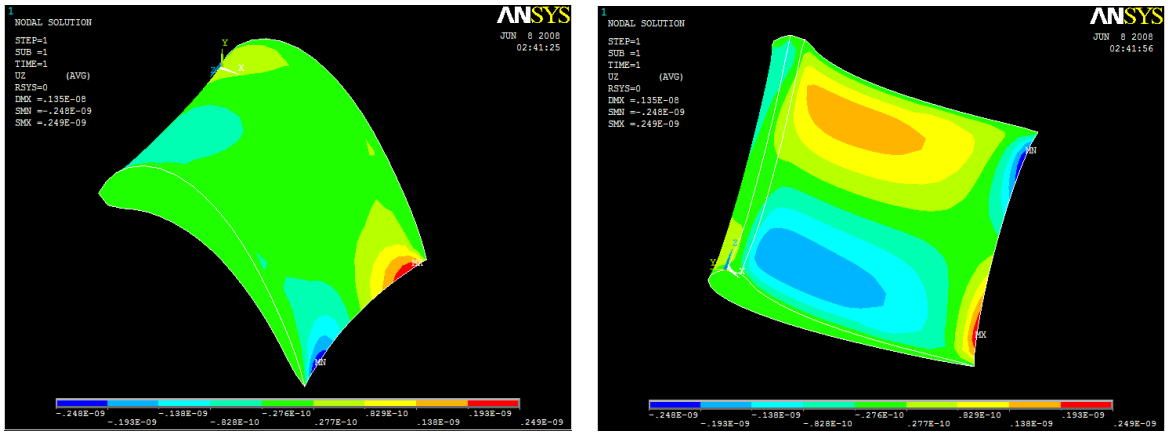


Figure 4.21. Z-Component of displacement on the blade

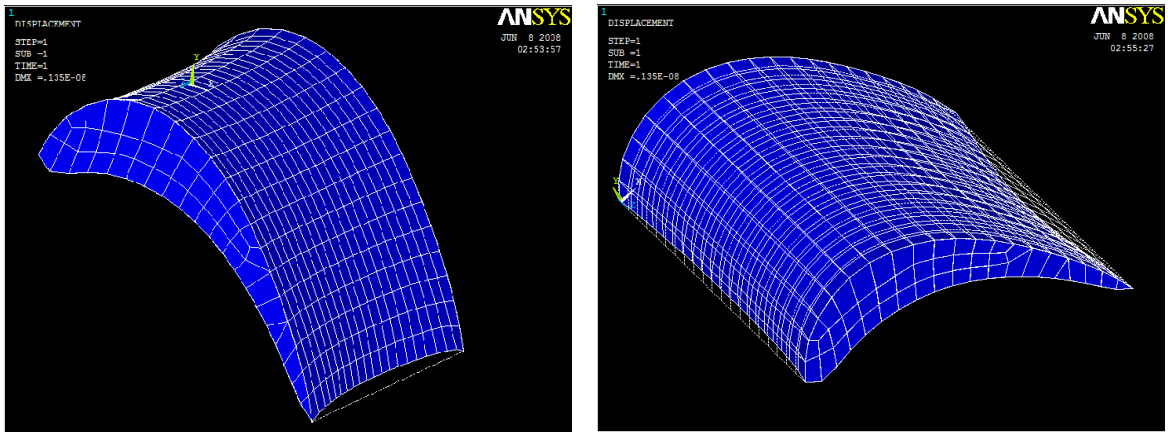


Figure 4.22. Deformed shape with undeformed edge

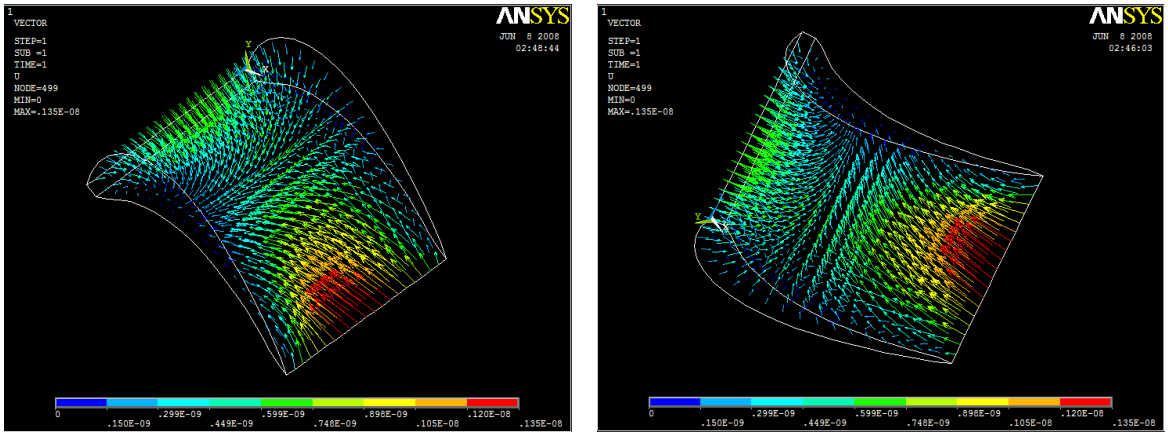


Figure 4.23. Vector plot of translation

4.2.2.2. Results after the convergence of displacement vectors. After last step of FSI, the results are demonstrated as the final shape of the blade.

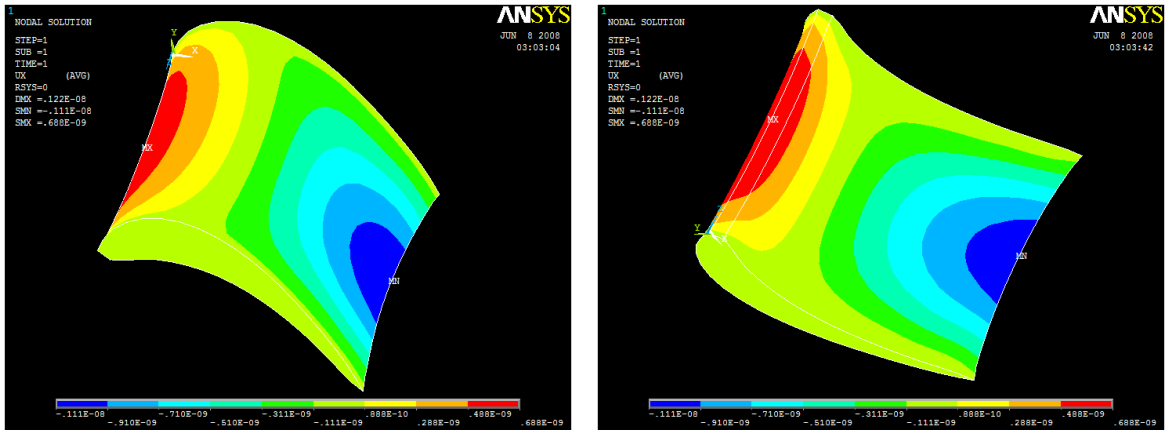


Figure 4.24. X-Component of displacement on the blade

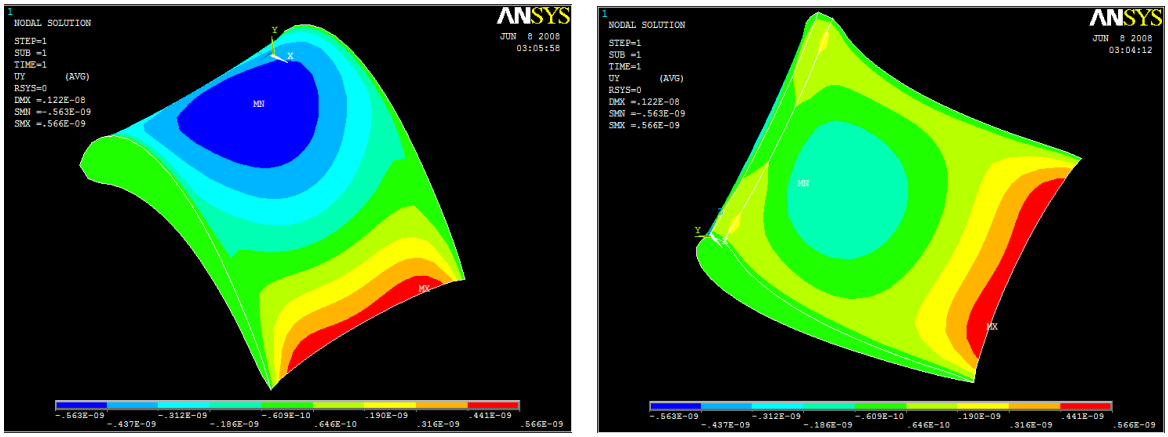


Figure 4.25. Y-Component of displacement on the blade

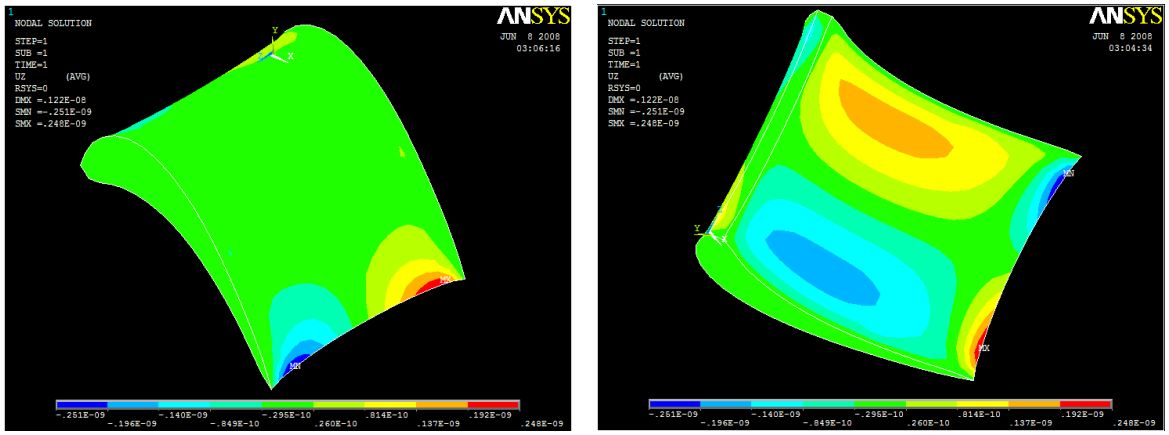


Figure 4.26. Z-Component of displacement on the blade

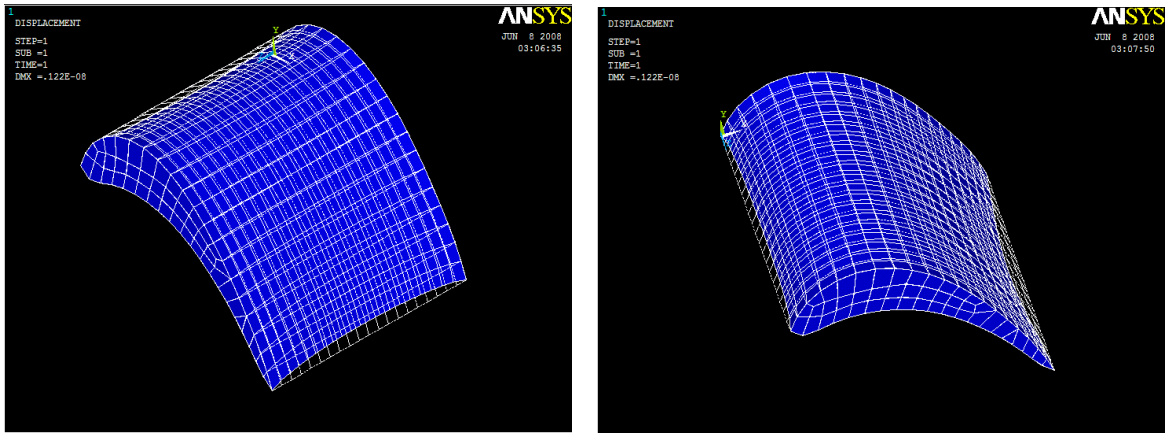


Figure 4.27. Deformed shape with undeformed edge

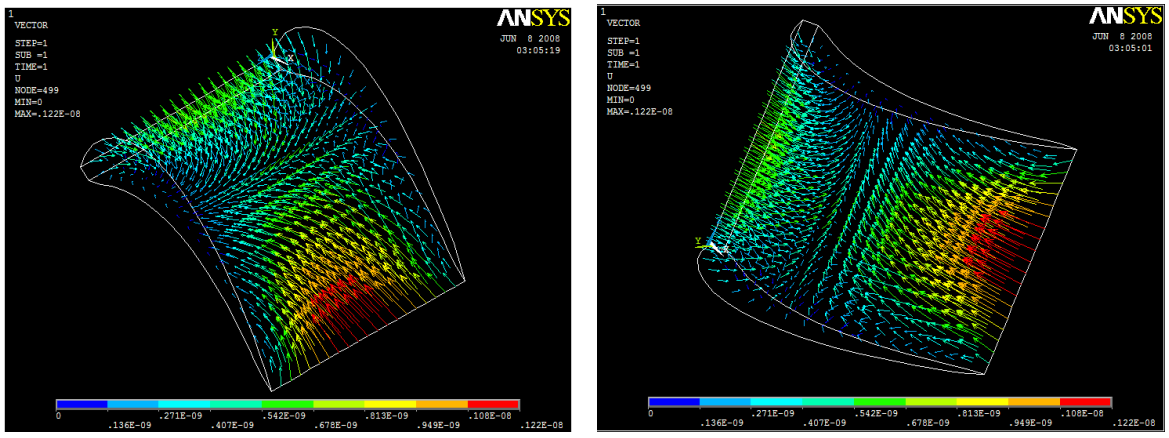


Figure 4.28. Vector plot of translation

4.3. Comparison of The Computational Parameters

The variation of the non-linear residual with respect to the number of Newton steps is given in the Fig.4.29. Zero vector is used as the initial guess both for two- and three-dimensional cases. Different from the linear residual absolute non-linear residual is monitored and the tolerance is set to 1×10^{-6}. In the Fig.4.30 the convergence behav-

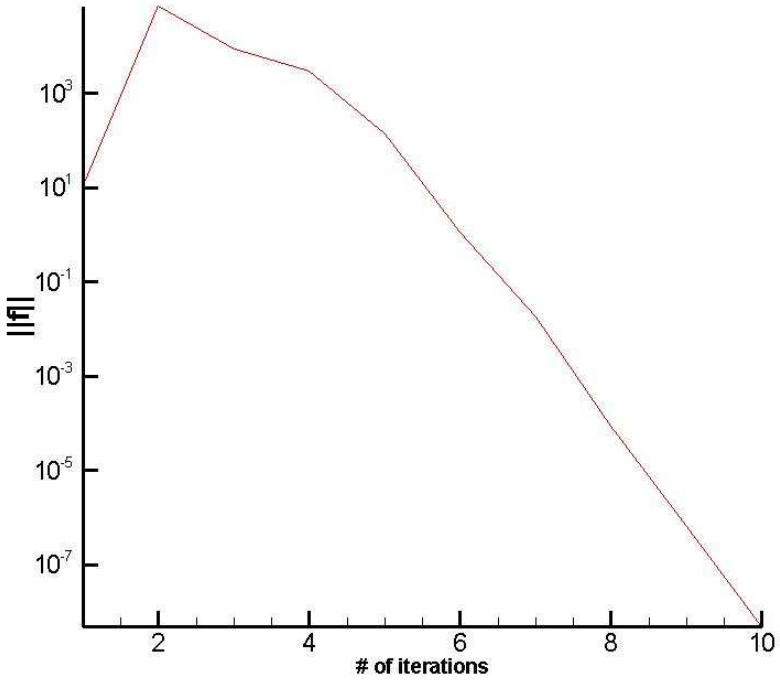


Figure 4.29. 2-norm of the non-linear residual vs. Newton steps

ior of the solvers and the effect of the preconditioners on this behavior are visualized. Three different preconditioners namely Jacobi, SGS and ILU(l) are combined with two different Krylov sub-space methods such as BiCGSTAB and GMRES.Relative residual is checked to decide for convergence and is taken as 1×10^{-5}. Since ILU needs Jacobian explicitly, it can not be applied to the matrix-free algorithms. In order to compare its effect with the effects of Jacobi and SGS, exact Newton algorithm is used and this algorithm is adapted to the compressed column storage(CCS) scheme. Otherwise the grid density in the problem would lead to high computational costs. The residual of the linear system is evaluated relative to the initial residual of the system. That's why the residuals equal to unity at the beginning of the iterations. GMRES is the one with the most regular convergence pattern. Iterations in GMRES are restarted after 100

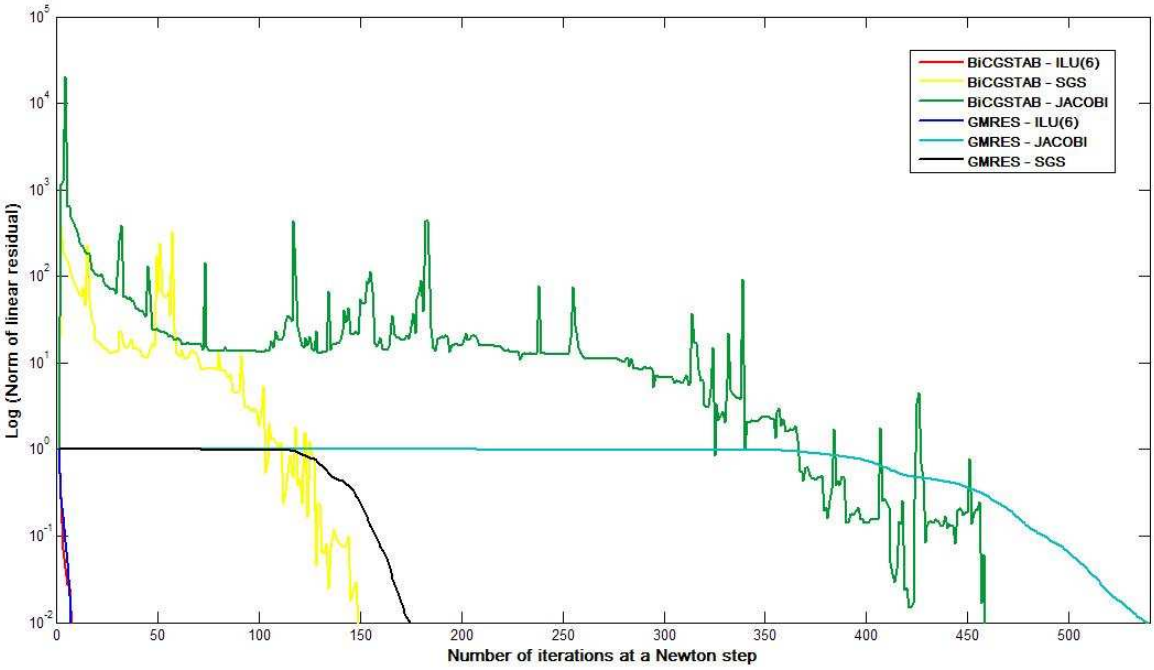


Figure 4.30. Comparison of the preconditioners

Table 4.1. Comparison of the solvers and preconditioners by means of iteration number and computation time

Method	# of iterations per Newton Step (average)	Time (sec.)
Jacobi-BiCGSTAB	490.5	25.7
SGS-BiCGSTAB	108.8	23.1
ILU(6)-BiCGSTAB	16.2	21.2
Jacobi-GMRES(100)	313.6	26.1
SGS-GMRES(100)	97.7	22.7
ILU(6)-GMRES(100)	12.5	20.4

iterations. The accumulated data are cleared and the intermediate results are used as the initial data for the next 100 iterations. In GMRES the residual of the linear system always decreases as iteration continues which is not the case at BiCGSTAB. However, it requires more iteration steps in the sample problem than BiCGSTAB requires to converge. On the other hand, the robust and stable character of GMRES can be used at the ill-conditioned matrices where the other solvers may diverge.

Preconditioners have a significant effect on the convergence of the solvers such that Jacobian matrix solvers without preconditioners could not make the solvers converge in the prescribed iteration limit. Some methods even diverge when used without a preconditioner.

Jacobi is one of the most easily applicable preconditioners however it does not play an important role on the convergence behavior of the solvers. On the other hand, SGS and ILU(l) methods greatly enhance the convergence behavior of all two methods not only in the number of iteration steps but also in computation time. Looking to the graphs it is understood that ILU(l) preconditioned methods have a superior convergence character to the SGS preconditioned ones.

5. CONCLUSIONS

In this study, both two- and three-dimensional fluid-solid interaction analysis of turbomachinery blades are investigated. For the fluid analysis part of the solution procedure, discretized form of the governing equations are solved and for solid analysis ANSYS is applied. As mentioned earlier, both exact and inexact Newton's method are used to linearize the non-linear governing equations and matrix-free implementations of the preconditioned Krylov techniques are implemented to solve the resulting linear equations. Two different solvers, BiCGSTAB and GMRES, and three preconditioners, SGS, JACOBI and, are chosen as solution methods.

While stream function-vorticity and velocity-vorticity formulations are applied in two-dimensional analysis, only velocity-vorticity approach is used in three-dimensional analysis.
After achieving the velocity and vorticity field, pressure distribution is calculated on the blade surface. For solid analysis ANSYS is implemented to compute the deformation of the blades under the acquired conditions. Different Reynolds numbers are chosen to study their effects on the results of FSI analysis and two boundary conditions for the inflow are defined to monitor the changes on the flow field and solid deformation. Both algebraic and elliptic grid generation methods are performed to obtain a desirable grid for the physical domain to achieve more accurate numerical results.

Transformation of governing equations is preferred to be the fundamental idea behind the fluid analysis in both two- and three-dimensional cases since the geometry of turbomachinery blades is highly complex. Due to the high level of complexity of the transformed form of the equations and computational memory for data storage especially in three-dimensional case, computation time increases considerably. This fact causes coarser grid density in three dimensional case which produces less accurate results. In order to avoid this difficulty a more powerful and expensive computational facility is required since it is beyond the capability of numerical methods. Putting aside this disadvantage, it can be concluded that the FSI analysis enables to demonstrate

the complex interaction between the fluid flow and blade structure accurately.

As mentioned earlier, both two- and three-dimensional analysis can be modified by using numerical implementations since especially the fluid analysis requires the major part of time spent to complete one cycle of FSI. Some of these applications are summarized in the following part as domain decomposition, nonlinear preconditioning, parallel computing, multigrid/multilevel techniques.

Domain decomposition methods are based on a partitioning of the domain of the physical problem. These methods typically involve independent system solution on the subdomains, and some way of combining data from the subdomains on the separator part of the domain. Since the subdomains can be handled separately, such methods are very attractive for coarse-grain parallel computers. On the other hand, it should be stressed that they can be very effective even on sequential computers. Generally, there are two kinds of approaches depending on whether the subdomains overlap with one another (Schwarz methods) or are separated from one another by interfaces (Schur Complement methods, iterative substructuring). Both of the approaches can be very practical in the manner of FSI analysis of turbomachinery blades since the geometry is highly complex and the fluid flow between the blades is three-dimensional with some particular difficulties.

Another way which may also be linked with domain decomposition phenomena is parallel computing. Since the iterative methods share most of their computational kernels, inner products, vector updates, matrix-vector products and preconditioner solves can be easily parallelized. On distributed-memory machines these basic time-consuming schemes then have to be sent to other processors to be combined for their global procedures.

Multigrid and multilevel methods are also extremely efficient for especially simple iterative methods (such as the Jacobi method) which tend to damp out high frequency components of the error fastest. Multigrid techniques are based on the following heuristic: 1- Perform some steps of a basic method in order to smooth out the error. 2-

Restrict the current state of the problem to a subset of the grid points, the so-called "coarse grid", and solve the resulting projected problem. 3- Interpolate the coarse grid solution to the original grid, and perform a number of steps of the basic method again. Steps 1 and 3 are called "pre-smoothing" and "post-smoothing" respectively; by applying this method recursively to step 2 it becomes a true "multigrid" method.

APPENDIX A: Transformation of the Governing Equations in 3-D

In this chapter coordinate transformation of the governing equations for the velocity-vorticity formulation expressed in the Cartesian coordinate system (x, y, z) from physical space to computational space (ξ, η, ζ) is investigated. Time derivatives in the equations are neglected since only the steady state solution of the problem is considered. The governing equations are transformed from the physical space (x, y, z) to the computational space (ξ, η, ζ) by the following relations

$$\begin{aligned} \xi &= \xi(x, y, z) \\ \eta &= \eta(x, y, z) \\ \zeta &= \zeta(x, y, z) \end{aligned} \tag{A.1}$$

In the following discussion the partial derivatives will be denoted using subscripts, $\xi_x = \partial\xi/\partial x$. The chain rule of partial differentiation provides the following expressions for the Cartesian derivatives

$$\begin{aligned} \frac{\partial}{\partial x} &= \xi_x \frac{\partial}{\partial \xi} + \eta_x \frac{\partial}{\partial \eta} + \zeta_x \frac{\partial}{\partial \zeta} \\ \frac{\partial}{\partial y} &= \xi_y \frac{\partial}{\partial \xi} + \eta_y \frac{\partial}{\partial \eta} + \zeta_y \frac{\partial}{\partial \zeta} \\ \frac{\partial}{\partial z} &= \xi_z \frac{\partial}{\partial \xi} + \eta_z \frac{\partial}{\partial \eta} + \zeta_z \frac{\partial}{\partial \zeta} \end{aligned} \tag{A.2}$$

From Equations A.2, it is obvious that the value of the metrics ξ_x, η_x, ζ_x, ξ_y, η_y, ζ_y, ξ_z, η_z and ζ_z must be provided in the same fashion. In most cases the analytical determination of the metrics is not possible and, therefore, they must be computed numerically. Since the stepsizes in the computational domain are equally spaced, ξ_x, ξ_y, ξ_z, etc., can be computed by various finite difference approximations. Thus, if the metrics appearing in Equations A.2 can be expressed in terms of these derivatives, the numerical computation of metrics is completed. To obtain such relations, the following

differential expressions are considered :

$$\begin{aligned} dx &= x_\xi d\xi + x_\eta d\eta + x_\zeta d\zeta \\ dy &= y_\xi d\xi + y_\eta d\eta + y_\zeta d\zeta \\ dz &= z_\xi d\xi + z_\eta d\eta + z_\zeta d\zeta \end{aligned} \tag{A.3}$$

Equations A.3 are expressed in a matrix form as

$$\begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{bmatrix} \begin{bmatrix} d\xi \\ d\eta \\ d\zeta \end{bmatrix} \tag{A.4}$$

Reversing the role of the independent variables,

$$\begin{aligned} d\xi &= \xi_x dx + \xi_y dy + \xi_z dz \\ d\eta &= \eta_x dx + \eta_y dy + \eta_z dz \\ d\zeta &= \zeta_x dx + \zeta_y dy + \zeta_z dz \end{aligned} \tag{A.5}$$

which are expressed as

$$\begin{bmatrix} d\xi \\ d\eta \\ d\zeta \end{bmatrix} = \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} \tag{A.6}$$

Comparing Equations A.4 and A.6, one concludes that

$$\begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix} = \begin{bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{bmatrix}^{-1} \tag{A.7}$$

From which,

$$\begin{aligned}
\xi_x &= \tfrac{1}{J}\left(y_\eta z_\zeta - y_\zeta z_\eta\right) \\
\xi_y &= \tfrac{1}{J}\left(x_\zeta z_\eta - x_\eta z_\zeta\right) \\
\xi_z &= \tfrac{1}{J}\left(x_\eta y_\zeta - x_\zeta y_\eta\right) \\
\eta_x &= \tfrac{1}{J}\left(y_\zeta z_\xi - y_\xi z_\zeta\right) \\
\eta_y &= \tfrac{1}{J}\left(x_\xi z_\zeta - x_\zeta z_\xi\right) \\
\eta_z &= \tfrac{1}{J}\left(x_\zeta y_\xi - x_\xi y_\zeta\right) \\
\zeta_x &= \tfrac{1}{J}\left(y_\xi z_\eta - y_\eta z_\xi\right) \\
\zeta_y &= \tfrac{1}{J}\left(x_\eta z_\xi - x_\xi z_\eta\right) \\
\zeta_z &= \tfrac{1}{J}\left(x_\xi y_\eta - x_\eta y_\xi\right)
\end{aligned} \tag{A.8}$$

where J is the Jacobian of transformation defined by

$$J = \frac{\partial\left(\xi,\eta,\zeta\right)}{\partial\left(x,y,z\right)} = x_\xi\left(y_\eta z_\zeta - y_\zeta z_\eta\right) - x_\eta\left(y_\xi z_\zeta - y_\zeta z_\xi\right) + x_\zeta\left(y_\xi z_\eta - y_\eta z_\xi\right) \tag{A.9}$$

To complete the transformation we need to obtain the second and mixed derivatives of any arbitrary variable with respect to ξ, η, ζ. Let x-component of velocity u be the dependent variable such as $u = u(x,y,z)$, where $x = x(\xi,\eta,\zeta)$, $y = y(\xi,\eta,\zeta)$ and $z = z(\xi,\eta,\zeta)$. The following expressions are derived using the simple rule for differentiation of a product of two terms

$$\begin{aligned}
\frac{\partial^2 u}{\partial x^2} &= \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x}\right) = \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial u}{\partial \eta}\frac{\partial \eta}{\partial x} + \frac{\partial u}{\partial \zeta}\frac{\partial \zeta}{\partial x}\right) \\
&= \frac{\partial^2 u}{\partial \xi \partial x}\frac{\partial \xi}{\partial x} + \frac{\partial u}{\partial \xi}\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial^2 u}{\partial \eta \partial x}\frac{\partial \eta}{\partial x} + \frac{\partial u}{\partial \eta}\frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 u}{\partial \zeta \partial x}\frac{\partial \zeta}{\partial x} + \frac{\partial u}{\partial \zeta}\frac{\partial^2 \zeta}{\partial x^2}
\end{aligned} \tag{A.10}$$

The first, third and fifth terms in the above equation involve differentiation with respect to one variable in the (x,y,z) system and another variable in the (ξ,η,ζ) system. Since this is not appropriate for the coordinate transformation we need to work further with

these terms

$$\frac{\partial^2 u}{\partial \xi \partial x} = \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial \xi}\right) = \left(\frac{\partial}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial}{\partial \eta}\frac{\partial \eta}{\partial x} + \frac{\partial}{\partial \zeta}\frac{\partial \zeta}{\partial x}\right)\frac{\partial u}{\partial \xi} \tag{A.11}$$

Expanding the above yields

$$\frac{\partial^2 u}{\partial \xi \partial x} = \frac{\partial^2 u}{\partial \xi^2}\frac{\partial \xi}{\partial x} + \frac{\partial^2 u}{\partial \xi \partial \eta}\frac{\partial \eta}{\partial x} + \frac{\partial^2 u}{\partial \xi \partial \zeta}\frac{\partial \zeta}{\partial x} \tag{A.12}$$

Similarly

$$\frac{\partial^2 u}{\partial \eta \partial x} = \frac{\partial^2 u}{\partial \xi \partial \eta}\frac{\partial \xi}{\partial x} + \frac{\partial^2 u}{\partial \eta^2}\frac{\partial \eta}{\partial x} + \frac{\partial^2 u}{\partial \zeta \partial \eta}\frac{\partial \zeta}{\partial x} \tag{A.13}$$

$$\frac{\partial^2 u}{\partial \zeta \partial x} = \frac{\partial^2 u}{\partial \xi \partial \zeta}\frac{\partial \xi}{\partial x} + \frac{\partial^2 u}{\partial \zeta \partial \eta}\frac{\partial \eta}{\partial x} + \frac{\partial^2 u}{\partial \zeta^2}\frac{\partial \zeta}{\partial x} \tag{A.14}$$

If we rewrite Equation A.10 by substituting Equations A.12, A.13 and A.14

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial \xi}\frac{\partial^2 \xi}{\partial x^2} + \frac{\partial u}{\partial \eta}\frac{\partial^2 \eta}{\partial x^2} + \frac{\partial u}{\partial \zeta}\frac{\partial^2 \zeta}{\partial x^2} + \frac{\partial^2 u}{\partial \xi^2}\left(\frac{\partial \xi}{\partial x}\right)^2 + \frac{\partial^2 u}{\partial \eta^2}\left(\frac{\partial \eta}{\partial x}\right)^2 + \frac{\partial^2 u}{\partial \zeta^2}\left(\frac{\partial \zeta}{\partial x}\right)^2 + 2\frac{\partial^2 u}{\partial \xi \partial \eta}\frac{\partial \xi}{\partial x}\frac{\partial \eta}{\partial x} + 2\frac{\partial^2 u}{\partial \xi \partial \zeta}\frac{\partial \xi}{\partial x}\frac{\partial \zeta}{\partial x} + 2\frac{\partial^2 u}{\partial \zeta \partial \eta}\frac{\partial \zeta}{\partial x}\frac{\partial \eta}{\partial x} \tag{A.15}$$

By using the same approach explained above for the partial derivatites $\partial^2 u/\partial y^2$, $\partial^2 u/\partial z^2$, $\partial^2 u/\partial x \partial y$, $\partial^2 u/\partial x \partial z$ and $\partial^2 u/\partial y \partial z$ we have the following expressions

$$\frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial \xi}\frac{\partial^2 \xi}{\partial y^2} + \frac{\partial u}{\partial \eta}\frac{\partial^2 \eta}{\partial y^2} + \frac{\partial u}{\partial \zeta}\frac{\partial^2 \zeta}{\partial y^2} + \frac{\partial^2 u}{\partial \xi^2}\left(\frac{\partial \xi}{\partial y}\right)^2 + \frac{\partial^2 u}{\partial \eta^2}\left(\frac{\partial \eta}{\partial y}\right)^2 + \frac{\partial^2 u}{\partial \zeta^2}\left(\frac{\partial \zeta}{\partial y}\right)^2 + 2\frac{\partial^2 u}{\partial \xi \partial \eta}\frac{\partial \xi}{\partial y}\frac{\partial \eta}{\partial y} + 2\frac{\partial^2 u}{\partial \xi \partial \zeta}\frac{\partial \xi}{\partial y}\frac{\partial \zeta}{\partial y} + 2\frac{\partial^2 u}{\partial \zeta \partial \eta}\frac{\partial \zeta}{\partial y}\frac{\partial \eta}{\partial y} \tag{A.16}$$

$$\frac{\partial^2 u}{\partial z^2} = \frac{\partial u}{\partial \xi}\frac{\partial^2 \xi}{\partial z^2} + \frac{\partial u}{\partial \eta}\frac{\partial^2 \eta}{\partial z^2} + \frac{\partial u}{\partial \zeta}\frac{\partial^2 \zeta}{\partial z^2} + \frac{\partial^2 u}{\partial \xi^2}\left(\frac{\partial \xi}{\partial z}\right)^2 + \frac{\partial^2 u}{\partial \eta^2}\left(\frac{\partial \eta}{\partial z}\right)^2 + \frac{\partial^2 u}{\partial \zeta^2}\left(\frac{\partial \zeta}{\partial z}\right)^2 + 2\frac{\partial^2 u}{\partial \xi \partial \eta}\frac{\partial \xi}{\partial z}\frac{\partial \eta}{\partial z} + 2\frac{\partial^2 u}{\partial \xi \partial \zeta}\frac{\partial \xi}{\partial z}\frac{\partial \zeta}{\partial z} + 2\frac{\partial^2 u}{\partial \zeta \partial \eta}\frac{\partial \zeta}{\partial z}\frac{\partial \eta}{\partial z} \tag{A.17}$$

$$\frac{\partial^2 u}{\partial x \partial y} = \frac{\partial u}{\partial \xi}\frac{\partial^2 \xi}{\partial x \partial y} + \frac{\partial u}{\partial \eta}\frac{\partial^2 \eta}{\partial x \partial y} + \frac{\partial u}{\partial \zeta}\frac{\partial^2 \zeta}{\partial x \partial y} + \frac{\partial^2 u}{\partial \xi^2}\frac{\partial \xi}{\partial x}\frac{\partial \xi}{\partial y} + \frac{\partial^2 u}{\partial \eta^2}\frac{\partial \eta}{\partial x}\frac{\partial \eta}{\partial y} + \frac{\partial^2 u}{\partial \zeta^2}\frac{\partial \zeta}{\partial x}\frac{\partial \zeta}{\partial y} + \frac{\partial^2 u}{\partial \xi \partial \eta}\left(\frac{\partial \xi}{\partial y}\frac{\partial \eta}{\partial x} + \frac{\partial \xi}{\partial x}\frac{\partial \eta}{\partial y}\right) + \frac{\partial^2 u}{\partial \xi \partial \zeta}\left(\frac{\partial \xi}{\partial y}\frac{\partial \zeta}{\partial x} + \frac{\partial \xi}{\partial x}\frac{\partial \zeta}{\partial y}\right) + \frac{\partial^2 u}{\partial \zeta \partial \eta}\left(\frac{\partial \eta}{\partial y}\frac{\partial \zeta}{\partial x} + \frac{\partial \eta}{\partial x}\frac{\partial \zeta}{\partial y}\right) \tag{A.18}$$

$$\frac{\partial^2 u}{\partial x \partial z} = \frac{\partial u}{\partial \xi}\frac{\partial^2 \xi}{\partial x \partial z} + \frac{\partial u}{\partial \eta}\frac{\partial^2 \eta}{\partial x \partial z} + \frac{\partial u}{\partial \zeta}\frac{\partial^2 \zeta}{\partial x \partial z} + \frac{\partial^2 u}{\partial \xi^2}\frac{\partial \xi}{\partial x}\frac{\partial \xi}{\partial z} + \frac{\partial^2 u}{\partial \eta^2}\frac{\partial \eta}{\partial x}\frac{\partial \eta}{\partial z} + \frac{\partial^2 u}{\partial \zeta^2}\frac{\partial \zeta}{\partial x}\frac{\partial \zeta}{\partial z} + \frac{\partial^2 u}{\partial \xi \partial \eta}\left(\frac{\partial \xi}{\partial z}\frac{\partial \eta}{\partial x} + \frac{\partial \xi}{\partial x}\frac{\partial \eta}{\partial z}\right) + \frac{\partial^2 u}{\partial \xi \partial \zeta}\left(\frac{\partial \xi}{\partial z}\frac{\partial \zeta}{\partial x} + \frac{\partial \xi}{\partial x}\frac{\partial \zeta}{\partial z}\right) + \frac{\partial^2 u}{\partial \zeta \partial \eta}\left(\frac{\partial \eta}{\partial z}\frac{\partial \zeta}{\partial x} + \frac{\partial \eta}{\partial x}\frac{\partial \zeta}{\partial z}\right) \tag{A.19}$$

$$\frac{\partial^2 u}{\partial y \partial z} = \frac{\partial u}{\partial \xi}\frac{\partial^2 \xi}{\partial y \partial z} + \frac{\partial u}{\partial \eta}\frac{\partial^2 \eta}{\partial y \partial z} + \frac{\partial u}{\partial \zeta}\frac{\partial^2 \zeta}{\partial y \partial z} + \frac{\partial^2 u}{\partial \xi^2}\frac{\partial \xi}{\partial y}\frac{\partial \xi}{\partial z} + \frac{\partial^2 u}{\partial \eta^2}\frac{\partial \eta}{\partial y}\frac{\partial \eta}{\partial z} + \frac{\partial^2 u}{\partial \zeta^2}\frac{\partial \zeta}{\partial y}\frac{\partial \zeta}{\partial z}$$
$$+ \frac{\partial^2 u}{\partial \xi \partial \eta}\left(\frac{\partial \xi}{\partial z}\frac{\partial \eta}{\partial y} + \frac{\partial \xi}{\partial y}\frac{\partial \eta}{\partial z}\right) + \frac{\partial^2 u}{\partial \xi \partial \zeta}\left(\frac{\partial \xi}{\partial z}\frac{\partial \zeta}{\partial y} + \frac{\partial \xi}{\partial y}\frac{\partial \zeta}{\partial z}\right) + \frac{\partial^2 u}{\partial \zeta \partial \eta}\left(\frac{\partial \eta}{\partial z}\frac{\partial \zeta}{\partial y} + \frac{\partial \eta}{\partial y}\frac{\partial \zeta}{\partial z}\right) \tag{A.20}$$

As mentioned in the beginning of this chapter if the transformation, Equations A.1, is given analytically, then it is possible to obtain analytic values for the metric terms. However, sometimes, the transformation is given numerically, hence the metric terms are calculated as finite differences.

Also, the transformation may be expressed as the inverse of Equations A.1, then we may have available the inverse transformation

$$\begin{aligned} x &= x(\xi, \eta, \zeta) \\ y &= y(\xi, \eta, \zeta) \\ z &= z(\xi, \eta, \zeta) \end{aligned} \tag{A.21}$$

In Equations A.21, ξ, η and ζ are the *independent* variables. However, in the derivative transformations given by Equations A.2 to A.20, the metric terms are partial derivatives in terms of x, y and z as the independent variables. Therefore, in order to calculate the metric terms in these equations from the inverse transformation in Equations A.21, we need to relate $\partial\xi/\partial x$, $\partial\eta/\partial y$, etc., to the inverse forms $\partial x/\partial\xi$, $\partial y/\partial\eta$, etc. Let us proceed to find such relations

$$\frac{\partial^2 \xi}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial \xi}{\partial x}\right) = \left(\frac{\partial}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial}{\partial \eta}\frac{\partial \eta}{\partial x} + \frac{\partial}{\partial \zeta}\frac{\partial \zeta}{\partial x}\right)\left[\frac{1}{J}\left(\frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \eta}\right)\right] \tag{A.22}$$

$$\frac{\partial^2 \xi}{\partial x^2} = \frac{\partial \xi}{\partial x}\frac{\partial}{\partial \xi}\left[\frac{1}{J}\left(\frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \eta}\right)\right] + \frac{\partial \eta}{\partial x}\frac{\partial}{\partial \eta}\left[\frac{1}{J}\left(\frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \eta}\right)\right] + \frac{\partial \zeta}{\partial x}\frac{\partial}{\partial \zeta}\left[\frac{1}{J}\left(\frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \zeta} - \frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \eta}\right)\right] \tag{A.23}$$

$$\begin{aligned}\xi_{xx} = \xi_x \frac{1}{J^2}\left[\left(y_{\eta\xi} z_\zeta + y_\eta z_{\xi\zeta} - y_{\zeta\xi} z_\eta - z_{\xi\eta} y_\zeta\right) J - \left(y_\eta z_\zeta - y_\zeta z_\eta\right) J_\xi\right] \\ + \eta_x \frac{1}{J^2}\left[\left(y_{\eta\eta} z_\zeta + y_\eta z_{\eta\zeta} - y_{\zeta\eta} z_\eta - z_{\eta\eta} y_\zeta\right) J - \left(y_\eta z_\zeta - y_\zeta z_\eta\right) J_\eta\right] \\ + \zeta_x \frac{1}{J^2}\left[\left(y_{\eta\zeta} z_\zeta + y_\eta z_{\zeta\zeta} - y_{\zeta\zeta} z_\eta - z_{\zeta\eta} y_\zeta\right) J - \left(y_\eta z_\zeta - y_\zeta z_\eta\right) J_\zeta\right]\end{aligned} \tag{A.24}$$

We can derive other relations by using the same fashion expressed above as the following

$$\begin{aligned}\xi_{yy} = \xi_y \frac{1}{J^2}\left[\left(x_{\zeta\xi} z_\eta + x_\zeta z_{\xi\eta} - x_{\eta\xi} z_\zeta - z_{\xi\zeta} x_\eta\right) J - \left(x_\zeta z_\eta - x_\eta z_\zeta\right) J_\xi\right] \\ + \eta_y \frac{1}{J^2}\left[\left(x_{\zeta\eta} z_\eta + x_\zeta z_{\eta\eta} - x_{\eta\eta} z_\zeta - z_{\eta\zeta} x_\eta\right) J - \left(x_\zeta z_\eta - x_\eta z_\zeta\right) J_\eta\right] \\ + \zeta_y \frac{1}{J^2}\left[\left(x_{\zeta\zeta} z_\eta + x_\zeta z_{\zeta\eta} - x_{\eta\zeta} z_\zeta - z_{\zeta\zeta} x_\eta\right) J - \left(x_\zeta z_\eta - x_\eta z_\zeta\right) J_\zeta\right]\end{aligned} \tag{A.25}$$

$$\begin{aligned}\xi_{zz} = \xi_z \frac{1}{J^2}\left[\left(x_{\eta\xi} y_\zeta + x_\eta y_{\xi\zeta} - x_{\zeta\xi} y_\eta - y_{\xi\eta} x_\zeta\right) J - \left(x_\eta y_\zeta - x_\zeta y_\eta\right) J_\xi\right] \\ + \eta_z \frac{1}{J^2}\left[\left(x_{\eta\eta} y_\zeta + x_\eta y_{\eta\zeta} - x_{\zeta\eta} y_\eta - y_{\eta\eta} x_\zeta\right) J - \left(x_\eta y_\zeta - x_\zeta y_\eta\right) J_\eta\right] \\ + \zeta_z \frac{1}{J^2}\left[\left(x_{\eta\zeta} y_\zeta + x_\eta y_{\zeta\zeta} - x_{\zeta\zeta} y_\eta - y_{\zeta\eta} x_\zeta\right) J - \left(x_\eta y_\zeta - x_\zeta y_\eta\right) J_\zeta\right]\end{aligned} \tag{A.26}$$

$$\begin{aligned}\eta_{xx} = \xi_x \frac{1}{J^2}\left[\left(y_{\zeta\xi}z_\xi + y_\zeta z_{\xi\xi} - y_{\xi\xi}z_\zeta - z_{\xi\zeta}y_\xi\right)J - \left(y_\zeta z_\xi - y_\xi z_\zeta\right)J_\xi\right] \\ + \eta_x \frac{1}{J^2}\left[\left(y_{\zeta\eta}z_\xi + y_\zeta z_{\eta\xi} - y_{\xi\eta}z_\zeta - z_{\eta\zeta}y_\xi\right)J - \left(y_\zeta z_\xi - y_\xi z_\zeta\right)J_\eta\right] \\ + \zeta_x \frac{1}{J^2}\left[\left(y_{\zeta\zeta}z_\xi + y_\zeta z_{\zeta\xi} - y_{\xi\zeta}z_\zeta - z_{\zeta\zeta}y_\xi\right)J - \left(y_\zeta z_\xi - y_\xi z_\zeta\right)J_\zeta\right]\end{aligned} \tag{A.27}$$

$$\begin{aligned}\eta_{yy} = \xi_y \frac{1}{J^2}\left[\left(x_{\xi\xi}z_\zeta + x_\xi z_{\xi\zeta} - x_{\zeta\xi}z_\xi - z_{\xi\xi}x_\zeta\right)J - \left(x_\xi z_\zeta - x_\zeta z_\xi\right)J_\xi\right] \\ + \eta_y \frac{1}{J^2}\left[\left(x_{\xi\eta}z_\zeta + x_\xi z_{\eta\zeta} - x_{\zeta\eta}z_\xi - z_{\eta\xi}x_\zeta\right)J - \left(x_\xi z_\zeta - x_\zeta z_\xi\right)J_\eta\right] \\ + \zeta_y \frac{1}{J^2}\left[\left(x_{\xi\zeta}z_\zeta + x_\xi z_{\zeta\zeta} - x_{\zeta\zeta}z_\xi - z_{\zeta\xi}x_\zeta\right)J - \left(x_\xi z_\zeta - x_\zeta z_\xi\right)J_\zeta\right]\end{aligned} \tag{A.28}$$

$$\begin{aligned}\eta_{zz} = \xi_z \frac{1}{J^2}\left[\left(x_{\zeta\xi}y_\xi + x_\zeta y_{\xi\xi} - x_{\xi\xi}y_\zeta - y_{\xi\zeta}x_\xi\right)J - \left(x_\zeta y_\xi - x_\xi y_\zeta\right)J_\xi\right] \\ + \eta_z \frac{1}{J^2}\left[\left(x_{\zeta\eta}y_\xi + x_\zeta y_{\xi\eta} - x_{\xi\eta}y_\zeta - y_{\eta\zeta}x_\xi\right)J - \left(x_\zeta y_\xi - x_\xi y_\zeta\right)J_\eta\right] \\ + \zeta_z \frac{1}{J^2}\left[\left(x_{\zeta\zeta}y_\xi + x_\zeta y_{\xi\zeta} - x_{\xi\zeta}y_\zeta - y_{\zeta\zeta}x_\xi\right)J - \left(x_\zeta y_\xi - x_\xi y_\zeta\right)J_\zeta\right]\end{aligned} \tag{A.29}$$

$$\begin{aligned}\zeta_{xx} = \xi_x \frac{1}{J^2}\left[\left(y_{\xi\xi}z_\eta + y_\xi z_{\xi\eta} - y_{\eta\xi}z_\xi - z_{\xi\xi}y_\eta\right)J - \left(y_\xi z_\eta - y_\eta z_\xi\right)J_\xi\right] \\ + \eta_x \frac{1}{J^2}\left[\left(y_{\xi\eta}z_\eta + y_\xi z_{\eta\eta} - y_{\eta\eta}z_\xi - z_{\eta\xi}y_\eta\right)J - \left(y_\xi z_\eta - y_\eta z_\xi\right)J_\eta\right] \\ + \zeta_x \frac{1}{J^2}\left[\left(y_{\xi\zeta}z_\eta + y_\xi z_{\zeta\eta} - y_{\eta\zeta}z_\xi - z_{\zeta\xi}y_\eta\right)J - \left(y_\xi z_\eta - y_\eta z_\xi\right)J_\zeta\right]\end{aligned} \tag{A.30}$$

$$\begin{aligned}\zeta_{yy} = \xi_y \frac{1}{J^2}\left[\left(x_{\eta\xi} z_\xi + x_\eta z_{\xi\xi} - x_{\xi\xi} z_\eta - z_{\xi\eta} x_\xi\right) J - \left(x_\eta z_\xi - x_\xi z_\eta\right) J_\xi\right] \\ + \eta_y \frac{1}{J^2}\left[\left(x_{\eta\eta} z_\xi + x_\eta z_{\eta\xi} - x_{\xi\eta} z_\eta - z_{\eta\eta} x_\xi\right) J - \left(x_\eta z_\xi - x_\xi z_\eta\right) J_\eta\right] \\ + \zeta_y \frac{1}{J^2}\left[\left(x_{\eta\zeta} z_\xi + x_\eta z_{\zeta\xi} - x_{\xi\zeta} z_\eta - z_{\zeta\eta} x_\xi\right) J - \left(x_\eta z_\xi - x_\xi z_\eta\right) J_\zeta\right]\end{aligned} \tag{A.31}$$

$$\begin{aligned}\zeta_{zz} = \xi_z \frac{1}{J^2}\left[\left(x_{\xi\xi} y_\eta + x_\xi y_{\eta\xi} - x_{\eta\xi} y_\xi - y_{\xi\xi} x_\eta\right) J - \left(x_\xi y_\eta - x_\eta y_\xi\right) J_\xi\right] \\ + \eta_z \frac{1}{J^2}\left[\left(x_{\xi\eta} y_\eta + x_\xi y_{\eta\eta} - x_{\eta\eta} y_\xi - y_{\eta\xi} x_\eta\right) J - \left(x_\xi y_\eta - x_\eta y_\xi\right) J_\eta\right] \\ + \zeta_z \frac{1}{J^2}\left[\left(x_{\xi\zeta} y_\eta + x_\xi y_{\eta\zeta} - x_{\eta\zeta} y_\xi - y_{\zeta\xi} x_\eta\right) J - \left(x_\xi y_\eta - x_\eta y_\xi\right) J_\zeta\right]\end{aligned} \tag{A.32}$$

where

$$\begin{aligned}J_\xi = x_{\xi\xi}\left(y_\eta z_\zeta - y_\zeta z_\eta\right) + x_\xi\left(y_{\eta\xi} z_\zeta + z_{\zeta\xi} y_\eta - y_{\zeta\xi} z_\eta - z_{\eta\xi} y_\zeta\right) \\ - x_{\xi\eta}\left(y_\xi z_\zeta - y_\zeta z_\xi\right) - x_\eta\left(y_{\xi\xi} z_\zeta + z_{\zeta\xi} y_\xi - y_{\zeta\xi} z_\xi - z_{\xi\xi} y_\zeta\right) \\ + x_{\xi\zeta}\left(y_\xi z_\eta - y_\eta z_\xi\right) + x_\zeta\left(y_{\xi\xi} z_\eta + z_{\eta\xi} y_\xi - y_{\eta\xi} z_\xi - z_{\xi\xi} y_\eta\right)\end{aligned} \tag{A.33}$$

$$\begin{aligned}J_\eta = x_{\eta\xi}\left(y_\eta z_\zeta - y_\zeta z_\eta\right) + x_\xi\left(y_{\eta\eta} z_\zeta + z_{\zeta\eta} y_\eta - y_{\zeta\eta} z_\eta - z_{\eta\eta} y_\zeta\right) \\ - x_{\eta\eta}\left(y_\xi z_\zeta - y_\zeta z_\xi\right) - x_\eta\left(y_{\xi\eta} z_\zeta + z_{\zeta\eta} y_\xi - y_{\zeta\eta} z_\xi - z_{\xi\eta} y_\zeta\right) \\ + x_{\eta\zeta}\left(y_\xi z_\eta - y_\eta z_\xi\right) + x_\zeta\left(y_{\xi\eta} z_\eta + z_{\eta\eta} y_\xi - y_{\eta\eta} z_\xi - z_{\xi\eta} y_\eta\right)\end{aligned} \tag{A.34}$$

$$
\begin{aligned}
J_\zeta = {} & x_{\zeta\xi}\left(y_\eta z_\zeta - y_\zeta z_\eta\right) + x_\xi\left(y_{\eta\zeta} z_\zeta + z_{\zeta\zeta} y_\eta - y_{\zeta\zeta} z_\eta - z_{\eta\zeta} y_\zeta\right) \\
& - x_{\zeta\eta}\left(y_\xi z_\zeta - y_\zeta z_\xi\right) - x_\eta\left(y_{\xi\zeta} z_\zeta + z_{\zeta\zeta} y_\xi - y_{\zeta\zeta} z_\xi - z_{\xi\zeta} y_\zeta\right) \\
& + x_{\zeta\zeta}\left(y_\xi z_\eta - y_\eta z_\xi\right) + x_\zeta\left(y_{\xi\zeta} z_\eta + z_{\eta\zeta} y_\xi - y_{\eta\zeta} z_\xi - z_{\xi\zeta} y_\eta\right)
\end{aligned}
\tag{A.35}
$$

and ξ_x, ξ_y, ξ_z, η_x, η_y, η_z, ζ_x, ζ_y, ζ_z and J are defined in Equations A.8 and A.9.

REFERENCES

1. Roache, P.J., *Fundamentals of Computational Fluid Dynamics*, Hermosa Pub., USA, 1998

2. Munson, B.R., D.F. Young and T.H. Okiishi, *Fundamentals of Fluid Mechanics*, John Wiley and Sons, Inc., USA, 2006

3. Pao, Y.H. and R.J. Daugherty, *Time-dependent viscous incompressible flow past a finite flat plate*, Boeing Scientific Research Laboratories, D1-82-0822, January, 1969

4. Schobeiri, M., *Turbomachinery Flow Physics and Dynamic Performance*, Springer, 2005

5. Denton, J.D., "An improved time marching method for turbomachinery flow calculations", *Trans. ASME, J . Eng. Power*, 105, 514, 1983

6. Dawes, W.N., *Application of full Navier-Stokes solvers to turbomachinery flow problems*, VKI Lecture Series 2: Numerical Techniques for Viscous Flow Calculations in Turbomachinery Blading, 1986

7. Hah, C., A.C. Bryans, Z. Moussa and M.E. Tomsho, "Application of viscous flow computations for the aerodynamic performance of a backswept impeller at various operating conditions", *ASME J. Turbomach.*, 110, 303-311, 1988

8. Hoffman, K.A., *Computational Fluid Dynamics for Enginners - Volume I*, Engineering Education System, Kansas, 1997.

9. Blazek, J., *Computational Fluid Dynamics:Principles and Applications*, Elsevier Science Ltd., UK, 2004

10. Saad, Y., *Iterative Methods for Sparse Linear Systems*, PWS Publishing Company, Boston, 1996.

11. Ning Qin, David K. Ludlow and Scott T. Shaw, "A matrix-free preconditioned Newton:GMRES method for unsteady Navier–Stokes solutions", *International Journal for Numerical Methods in Fluids*, Vol. 33, pp. 223248, 2000

12. Barrett, R., M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine and H. Van der Vorst, "Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods, 2nd Edition", *SIAM*, Philadelphia, PA, 1994.

13. Farrashkhalvat, M., Miles, J.P., *Basic Structured Grid Generation*, Butterworth-Heinemann, GB, 2003

14. Hah, C., A.C. Bryans, Z. Moussa and M.E. Tomsho, "Application of viscous flow computations for the aerodynamic performance of a backswept impeller at various operating conditions", *ASME J. Turbomach.*, 110, 303-311, 1988

15. Xu, C. and W. J. Chen, "Computational analysis on a compressor blade", *Int. Conf. On Jets, Wakes and Separated Flows*, 2005

16. Cartens, V., R. Kemme and S. Schmitt, "Coupled simulation of flow-structure interaction in turbomachinery", *Aerospace Science and Technology*, Elsevier, 2003

17. Moffatt, S. and L. He, "On decoupled and fully-coupled methods for blade forced response prediction", *Journal of Fluids and Structures*, Elsevier, 2005

18. Kamakoti, R. and W. Shyy, "Fluidstructure interaction for aeroelastic applications", *Progress in Aerospace Sciences*, Elsevier, 2004