I, Shashank Mishra, hereby submit this original work as part of the requirements for the degree of Master of Science in Mechanical Engineering.

It is entitled: Developing Novel Computational Fluid Dynamics Technique for Incompressible Flow and Flow Path Design of Novel Centrifugal Compressor

Student's name: Shashank Mishra

This work and its defense approved by:

Committee chair: Shaaban Abdallah, Ph.D.
Committee member: Clarissa Belloni, Ph.D.
Committee member: Kumar Vermaganti, Ph.D.
Novel Computational Fluid Dynamics technique for Incompressible flow and flow path design of a novel centrifugal compressor

A thesis submitted to the
Graduate School
of the University of Cincinnati
in partial fulfillment of the
requirements for the degree of

Master of Science

in the Department of Mechanical and Materials Engineering
of the College of Engineering
by

Shashank Mishra

B.Tech., Indian Institute of Technology-BHU, 2009

Date 03/17/2016

Committee Chair: Shaaban Abdallah, Ph.D.
Abstract

The present thesis work is divided into two parts. The field of computational fluid dynamics study can be broadly classified into two categories namely: Incompressible and Compressible flows. Not only the physics of these two domains vary entirely from each other, with compressible flow having subsonic, supersonic, shock waves and other complex flow structures associated with it but the mathematical and numerical formulation of incompressible and compressible flow is entirely different. The compressible flow has more complex flow physics but numerical solution of incompressible flow equations is more complicated due to some inherent issues associated with it. In the first part of my thesis a new flow solver for incompressible flow is developed and the key issues associated with numerical modeling of incompressible flow are addressed. The second part of the thesis work is on numerical solution of compressible turbulent flow equations on a modified and novel centrifugal compressor flow path.

The incompressible flow equations are function of the pressure gradients and not the pressure. The most important issue in solution of flow equations of incompressible fluid is the pressure gradient vector which is appearing as a source term in the momentum equation, but does not have any obvious equation coupling it with other dependent variables. Accurate numerical solutions are obtained for the incompressible Navier Stokes equations in primitive variables. Explicit finite difference scheme computer code is developed to solve incompressible flow equations.

In this study, consistent with the physics of incompressible flows, the velocity and pressure gradient vectors are considered as the dependent variables. In this case, that satisfies continuity equation to machine zero, the pressure gradient vector increases the number of dependent variables which requires additional equations to close the system of governing equations. Additional equations are obtained by reformulating the continuity equation and adding a time derivative term for the pressure gradient.

Upon, convergence of the numerical solution, the continuity equation will be satisfied to an arbitrary constant. To enforce that constant to be zero, the continuity equation is set to be zero on the boundary of the solution domain. It is important to note that the curl of the reformulated continuity equation automatically satisfies the curl of the pressure gradient identity. This scheme is applicable for two and three dimensions and inviscid and viscous flows.
Numerical solutions for the driven cavity problem satisfies the continuity equation with maximum dilatation of order $10^{-12}$ for Reynolds numbers up to 5,000. Numerical solutions for developing flow inside a rectangular duct for Reynolds number 1000 and flow in a backward facing step is also obtained for further validation.
Multistage axial compressor has an advantage of lower stage loading as compared to a single stage. Several stages with low pressure ratio are linked together which allows for multiplication of pressure to generate high pressure ratio in an axial compressor. Since each stage has low pressure ratio they operate at a higher efficiency and the efficiency of multi-stage axial compressor as a whole is very high.

Although, single stage centrifugal compressor has higher pressure ratio compared with an axial compressor but multistage centrifugal compressors are not as efficient because the flow has to be turned from radial at outlet to axial at inlet for each stage. The present study explores the advantages of extending the axial compressor efficient flow path that consist of rotor stator stages to the centrifugal compressor stage. In this invention, two rotating rows of blades are mounted on the same impeller disk, separated by a stator blade row attached to the casing.

A certain amount of turning can be achieved through a single stage centrifugal compressor before flow starts separating, thus dividing it into multiple stages would be advantageous as it would allow for more flow turning. Also the individual stage now operate with low pressure ratio and high efficiency resulting into an overall increase in pressure ratio and efficiency.

The baseline is derived from the NASA low speed centrifugal compressor design which is a 55 degree backward swept impeller. Flow characteristics of the novel multistage design are compared with a single stage centrifugal compressor. The flow path of the baseline and multi-stage compressor are created using 3DBGB tool and DAKOTA is used to optimize the performance of baseline as well novel design. The optimization techniques used are Genetic algorithm followed by Numerical Gradient method. The optimization resulted into improvements in incidence and geometry which significantly improved the performance over baseline compressor design. The multi-stage compressor is more efficient with a higher pressure ratio compared with the base line design for the same work input and initial conditions.
**Acknowledgments**

I would like to express my deepest gratitude to Prof. Shaaban Abdallah, my thesis advisor for his excellent ideas, guidance and encouraging me in performing research work. I would also like to thank Prof. Mark G. Turner for helping me to develop my background in turbomachines. Special thanks goes to Prof. David Thompson who helped me as my academic advisor.

My work on Centrifugal compressor would have not been possible without help from my friend Syed Moez and Kiran Siddappaji who helped me learn optimization of turbomachines. I would also like to thank Ajay Dangi for his help with latex coding and be there every time I needed help from him.

The author would like to thank the College of Engineering and Applied Sciences at University of Cincinnati for supporting this research.

Last and far from least I would like to thank my family members especially my mother Urmila Mishra who kept me motivated during my hard times and I will dedicate this work to her.
Contents

Abstract i 

Contents iii 

List of Figures v 

List of Tables viii 

Nomenclature ix 

1 Introduction and Literature Review 1 

1.1 Introduction .................................................. 1 

1.1.1 Novel Computational Fluid Dynamics Scheme ............ 1 

1.1.2 Design of novel Centrifugal Compressor flow path ....... 4 

1.2 Goals and Thesis Layout ...................................... 5 

2 Novel Computational Fluid Dynamic Technique for Incompressible flow 8 

2.1 Governing Differential Equations ................................ 8 

2.2 Primitive and Non-Primitive Formulations ...................... 10 

2.2.1 The artificial Compressibility Method ...................... 10 

2.2.2 The Pressure Poisson Equation ............................ 11 

2.2.3 The Stream-Function Vorticity and Vorticity Velocity Formulation .............. 12 

2.3 Mathematical formulation of new scheme ...................... 13 

2.3.1 Modified continuity equation method ...................... 13 

2.3.2 Finite Difference Discretization ............................ 15
3 Design of Novel Centrifugal Compressor

3.1 Modified Centrifugal compressor

3.1.1 Methodology

3.1.2 1D Design

3.1.3 Computational Fluid Dynamics

3.1.4 Aerodynamic Optimization

3.1.5 Genetic Algorithm

3.1.6 Optimization Statement

3.2 Results and Analysis

3.2.1 Optimization Results

3.2.2 Results and Discussion

3.2.3 ALTERNATE DESIGNS

4 Conclusion and future works

4.1 Modified Continuity Equation method for Incompressible flow

4.2 Novel Centrifugal Compressor flow path design

Bibliography

A Matlab Scripts for Computational Fluid Dynamics Validation Cases

A.1 Matlab script for Simulating Lid driven cavity flows

A.2 Matlab script for Simulating Flow inside a rectangular duct

A.3 Matlab script for Simulating flow in backward facing step

A.4 Matlab script for Simulating flow in 3D backward facing step

A.5 Matlab script for Generating flow path and velocity triangle of centrifugal compressor

A.6 Matlab script for Lid driven cavity flow in magnetic field

A.7 Matlab script for Multipath integration of velocity field to evaluate stream function
List of Figures

1.1 Typical centrifugal compressor flow passage ........................................ 5
1.2 30° backswept test impeller used by Krain ................................. 6

2.1 Schematic picture of flow through a rectangular duct ......................... 18
2.2 Outlet velocity profile $R_e = 1000$ .................................................. 19
2.3 Horizontal velocity contour ............................................................... 19
2.4 Convergence history of error in maximum Dilation, $\frac{\partial P}{\partial t}$ and $\frac{\partial P}{\partial t}$ ................................................... 20
2.5 Distribution of dilation inside the duct after initial iterations ............ 20
2.6 Distribution of dilation inside the duct at mid convergence ............... 21
2.7 Distribution of dilation at end of convergence .................................... 21
2.8 Backward facing step ............................................................... 22
2.9 $u$ velocity contour for flow in backward facing step ...................... 22
2.10 Distribution of dilation at convergence for flow inside backward facing step ....................... 23
2.11 Convergence history for flow inside a backward facing step ........ 23
2.12 Geometry and computational mesh for driven cavity problem .......... 25
2.13 Convergence history for Reynolds number 100, 400, 1000 and 5000 respectively ........................................ 27
2.14 Dilation distribution at different Reynolds number using Ansys Fluent ....................... 28
2.15 Maximum and Average Dilation Convergence (Fluent) .................... 29
2.16 Distribution of dilation at mid and final convergence showing decay from $1e^{-3}$ to $1e^{-12}$ ........................................ 29
2.17 Decay of dilation for Reynolds number 5000 inside the cavity at different iterations ........................................ 30
2.18 Centre line velocity comparison for different Reynolds number ........ 31
2.19 Pressure field for Reynolds number 5000 ........................................ 32
2.20 Velocity contours for different Reynolds numbers ......................... 34
2.21 Stream function for Reynolds number 100, 400, 1000 and 5000 ........................................ 35
2.22 Vorticity at Reynolds number 400, 1000 and 5000 ................................................................. 36
2.23 Curl of pressure profile at geometric centre lines at different Reynolds numbers .................. 37
2.24 Comparison of horizontal velocity with and without magnetic field for flow inside a backward facing step ................................................................. 41
2.25 Comparison of separation length: with and without magnetic field backward step flow ... 41
2.26 Convergence history for driven cavity under magnetic field at Re-100 .............................. 42
2.27 Distribution of dilation inside the cavity for Re-100, under magnetic field ......................... 42
2.28 Comparison of u velocity with and without magnetic field at Reynolds number 100 .......... 43
2.29 Effect of high magnetic field on flow field inside driven cavity at Reynolds number 100 .... 44

3.1 Novel centrifugal compressor ................................................................. 46
3.2 Flow path of new compressor .............................................................. 46
3.3 LSCC Compressor used for deriving the baseline geometry .............................................. 48
3.4 LSCC flow path ................................................................. 49
3.5 Flow path of Multistage compressor ...................................................... 50
3.6 Multistage Compressor Geometry ........................................................ 50
3.7 Single stage compressor geometry ....................................................... 51
3.8 Details of velocity triangle ................................................................. 53
3.9 Baseline compressor grid ................................................................. 55
3.10 Multistage compressor computational grid .............................................. 56
3.11 Contour of y+ at hub ................................................................. 56
3.12 y+ contour at shroud ................................................................. 57
3.13 y+ contour at blades ................................................................. 57
3.14 Optimization loop ................................................................. 58
3.15 Spanwise cubic B-spline curvature control definition at hub of baseline ...................... 60
3.16 Spanwise cubic B-spline curvature control definition at hub of 1st stage ...................... 61
3.17 Spanwise cubic B-spline curvature control definition at hub of 2nd stage ...................... 61
3.18 Spanwise cubic B-spline curvature control definition at pichline of single stage centrifugal compressor ................................................................. 62
3.19 Spanwise cubic B-spline curvature control definition at tip of single stage centrifugal compressor .................................................. 63
3.20 Spanwise cubic B-spline curvature control definition at pitchline of 1st row of novel centrifugal compressor .................................................. 64
3.21 Blade profiles of baseline and optimum design for single stage centrifugal compressor .................................................. 65
3.22 Blade profiles of baseline and optimum design for 1st rotor of multistage compressor .................................................. 66
3.23 Blade profiles of baseline and optimum design for 2nd rotor of multistage compressor .................................................. 67
3.24 Static Pressure at 1% and 5% blade to blade surface .................................................. 68
3.25 Static Pressure at 50% and 95% blade to blade surface .................................................. 69
3.26 Entropy at 1% and 5% blade to blade surface .................................................. 70
3.27 Relative Mach no at 50% and 95% blade to blade surface .................................................. 71
3.28 Relative Mach no plot at outlet .................................................. 71
3.29 Azimuthal absolute Mach no .................................................. 72
3.30 Entropy generation plot at outlet .................................................. 72
List of Tables

2.1 Numerical results for backward facing step, $\nu = 5e^{-3}$, Comparison of recirculation region length $x_r$, primary vortex strength $\psi_{min}$ and $\psi'_{min}$ location ........................................... 24

2.2 Mesh details for lid driven cavity flow ............................................................... 29

2.3 Grid dependence study for backward facing step ............................................. 44

3.1 LSCC Design Parameters .................................................................................. 47

3.2 Multistage compressor 1D calculation results. .................................................. 54

3.3 Single stage vs Multistage (CFD Results). ....................................................... 64
Nomenclature

$B$ Magnetic field vector

$b$ Induced magnetic field

$b$ width of blade

$Bo$ Externally applied Magnetic field

$D$ Dilation

$Dh$ Hydraulic diameter

$E$ Electric field vector

$Fc$ Flux tensor of non-linear convective terms
$H$ Width of the rectangular channel

$h$ Enthalpy

$L.F$ Lorentz force

$Le$ Entrance length

$P$ Pressure

$Pr$ Pressure ratio

$PT$ Pressure Ratio

$R$ Gas Constant

$R$ Universal gas constant

$r$ radius of flow path

$Re$ Reynolds number

$S$ Non-linear algebraic source term

$T$ Temperature

$Tr$ Temperature ratio

$TT$ Total Temperature

$U$ Blade tangential velocity

$u$ cartesian velocity in x direction
\( V \)  Absolute Velocity of the fluid

\( \nu \)  Cartesian velocity in y direction

\( W \)  Relative velocity of fluid

\( w \)  Cartesian velocity in z direction

**Greek Symbols**

\( \alpha \)  Absolute blade angle

\( \beta \)  Relative blade angle

\( \eta \)  Efficiency

\( \gamma \)  Adiabatic constant

\( \lambda \)  Work distribution coefficient

\( \Omega \)  Rotational Speed of the Rotor

\( \Phi \)  Flow Angle

\( \pi \)  Circumference to diameter ratio for circle

\( \rho \)  Density of the fluid

**Subscripts**

0  Total quantity

1  Inlet of stage

2  Outlet of a stage

\( \theta \)  Tangential direction
leP  Leading edge at pitchline

M    Multistage

m    Meridional direction

r    Radial direction

rise Increase in quantity

z    Axial Direction or Free stream

Abbreviations

1 − D  One Dimensional

3 − D  Three Dimensional

3DBGB Three Dimensional Blade Geometry Builder
Chapter 1

Introduction and Literature Review

1.1 Introduction

1.1.1 Novel Computational Fluid Dynamics Scheme

The fundamental differential equations governing fluid flow are Navier-Stokes equations. Navier-Stokes equations are further classified into two categories namely, incompressible flow and compressible flow. Incompressible flow refers to flow in which the density of fluid can be assumed to be constant. Incompressible flow comprises of flow which are incompressible or conditions under which constant density approximation is valid. The incompressible flow equations are function of the pressure gradients and not the pressure. The most important issue in solution of flow equations of incompressible fluid is the pressure gradient vector which is appearing as a source term in the momentum equation, but does not have any obvious equation coupling it with other dependent variables. This creates difficulties in the numerical solutions of these equations because of lack of equation for pressure or pressure gradient. There are two remedies for this problem, the first approach eliminates the pressure gradients from the governing equations using the curl of the momentum equation and introducing vorticity as a new dependent variable. This method is known as a Non-Primitive Variables formulation [1]. The stream function-vorticity, the velocity-vorticity and the vector potential-vorticity methods are examples of this approach. The second approach is known as the Primitive Variables approach with the pressure Poisson [2] and artificial compressibility [3] methods and their variations are examples of this approach. Both Primitive and Non-Primitive variable formulations suffer from several numerical issues such as stiffness, integral constraints, limitation to two-dimensional solutions [4].
and non-enforcement of Continuity equation to machine zero on collocated grids.

In non-primitive variable approach, pressure makes no appearance as a dependent variable in the governing equations and divergence-free velocity constraint is satisfied since stream function is analytical solution of continuity equation. Since, the curl of momentum equation is a third order differential equation in terms of velocity as a dependent variable, vorticity is introduced to maintain second order formulation. However, non-primitive formulation suffers from difficulty associated with specification of boundary condition. The vorticity vector and velocity vector are hard to visualize and interpret, also this approach cannot be extended to three dimensional cases easily.

In the primitive approach, a differential equation for the pressure is derived by either modifying the continuity or the momentum equations. In artificial compressibility method, a pressure-time derivative term is added to the continuity equation analogous to the continuity equation of compressible flow and the incompressible field is treated as compressible during transient calculations. Thus, mixed elliptic-parabolic type equations are transformed into a system of hyperbolic or parabolic equations in pseudo-time. This provides a mechanism in which through forward marching in pseudo-time towards divergence-free velocity field, mass and momentum conservation are achieved in pseudo steady state. While pressure Poisson method, derivies a pressure equation from the divergence of the momentum equation and enforces the continuity equation through the time derivative term in the equation. The pressure Poisson equation is second order in pressure and is consistent with the elliptical nature of pressure field. In both methods, the velocity field is calculated from the time dependent momentum equation using time marching techniques, while each method employs a different equation to compute the pressure.

An important issue in the numerical solutions of the Navier Stokes equation using primitive variable formulations is the satisfaction of the divergence-free velocity condition. Pressure Poisson and artificial compressibility method satisfy discrete continuity equation only for staggered grids to machine zero level [2] [5]. It is important to note that a discrete derivation of the Pressure Poisson equation satisfies the continuity equation of collocated grid but suffers from odd-even decoupling. The odd-even decoupling appears because of second-order finite-difference approximation of first-order continuity equation. The odd-even decoupling can be removed by the explicit addition of fourth order artificial dissipation term [6]. The same issue also exists in the artificial compressibility method and requires dissipation terms be added to the modified continuity equation. This issue in the artificial compressibility is related implicitly with the pressure derivative formulation of the modified continuity equation upon substitution of the velocity components.
from the discrete solution of the momentum equation. As a result of odd-even decoupling, the discrete continuity equation does not converge to machine zero but rather to a term proportional to the pressure fourth-order derivative [7].

Galbraith and Abdallah [8] developed a modified pressure gradient method, for solving incompressible Navier-Stokes equations. They used the curl identity of pressure gradient to obtain additional equations and close the system of governing equations. Although, curl of pressure gradient is a second order differential equation in pressure since pressure gradient was the dependent variable, the equation is of first order.

In recent years, several high order discontinuous Galerkin finite element methods for the incompressible Navier-Stokes equations based on primitive variable formulation have been presented [9] [10] [11] [12] [13] [14] [15] [16] and is still a very active topic of ongoing research. Tavelli and Dubser [17] presented an arbitrary high order accurate semi-implicit space-time discontinuous Galerkin method to solve incompressible Navier-Stokes equations in two dimensions on staggered unstructured triangular meshes.

It is important to note that it is the pressure gradient not pressure driving the physics of incompressible flow since velocity is function of pressure gradient. Therefore consistent with the physics of the flow, the present approach treat pressure gradient as dependent variable. Use of pressure gradient vector as a dependent variable was first conceived by Shih [18] and later independently developed by Said [19]. In the present study, I directly calculated the pressure gradients from modified continuity equations on collocated grids. The time derivatives of the pressure gradients are added to the gradient of continuity equation. The new form of the continuity equations are solved for the pressure gradients consistent with the physics of incompressible flows. The modified pressure gradient equations are not stiff and require no boundary conditions for the pressure derivatives unlike all other pressure-based methods.

In addition, the new formulation enforces the continuity equation on the solution domain boundaries as boundary conditions for the spatial pressure derivatives. This formulation is valid for two and three dimensions, viscous and inviscid flows. However, for time dependent cases, similar to artificial compressibility method, dual time step method shall be used.

Numerical solution using the present formulation for the Driven- Cavity at Reynolds number 100, 400, 1000 and 5000 are presented and the results obtained satisfies the continuity equation to $10^{-13}$. Also, popular benchmark problems of developed flow in a duct at Reynolds number 1000 and flow in a backward facing step are studied for further validation of the new approach, and max error in continuity equation was reduced to the order of $10^{-16}$. The computed results are compared with existing numerical results. Also the
new formulation is compared with existing commercial CFD solvers, Ansys Fluent [20] and CD Adapco, Star CCM+ [21].

1.1.2 Design of novel Centrifugal Compressor flow path

The classification of compressors employing rotating vanes or blades is based on the configuration of the air flow passage and are broadly divided into two categories. Centrifugal or radial compressors constituting the first category have an increasing diameter flow passage in the direction of the flow while axial compressors, the second category machines have an almost constant diameter flow passage. Figure 1.1 shows the cross-sectional view of a centrifugal compressor in which the flow enters at station 1 through the annulus region between $r_{1,h}$ and $r_{1,t}$. The cylindrical region of radius $r_2$ and blade width $b$ forms the exit of the flow at station 2. The flow is then decelerated in the diffuser, resulting into further rise of static pressure and then enters the collector scroll at station 3. The static enthalpy rise across the impeller is given by

$$h_2 - h_1 = \frac{1}{2} (U_2^2 - U_1^2) - \frac{1}{2} (W_2^2 - W_1^2) \quad (1.1)$$

where 1 is the inlet section, 2 is the outlet section, $U$ is the blade velocity, $W$ is the relative velocity of fluid and $h$ is the enthalpy. It is important to note the centrifugal action on the fluid due to change in the radius of the rotor from inlet to outlet contributes most to the static pressure and static enthalpy rise for the complete stage. This transfer of energy is loss free and is almost independent of the nature of the flow. The static enthalpy rise due to deceleration of the fluid is associated with conventional losses and the actual pressure rise is less than the rise predicted by the equation 1. Additional deceleration of the fluid takes place in the diffuser section which is a very important component since the absolute velocity leaving the rotor is usually high. The deceleration process, as mentioned before is not very efficient and therefore for this reason, the backswept configuration impeller are more efficient compared to their counterparts radial and front-swept impeller. The blade angle in backswept blades varies from tip to the hub of the blade such that it sweeps back opposite to the direction of rotation [22]. An unshrouded backswept impeller is show in Figure 1.2 [23]. The size of the diffuser, hence the losses associated with it can be reduced with backswept configuration due to high degree of reaction and reduced absolute velocity. Backswept impellers also provide a better surge margin compared to radial and forward swept blades [24]. The main issue with back-swept configuration is that the pressure ratio is lower compared to a standard radial compressor [25]. Also, the turning of the
fluid is restricted by the separation at the trailing edge. In addition to this, in spite of simpler and compact geometry, centrifugal compressors have relatively lower isentropic efficiency compared to an axial machine. This is from the fact that radial compressors are mostly single stage and multi-staging can only be done by using a series of these compressors in tandem. This issues does not exist with axial compressor and high efficiency may be realized using a series of compression-diffusion stages along the flow path. The overall pressure ratio of multistage axial compressor is product of lower pressure ratios of individual stages and since each stage have a relatively lower pressure ratio, they operate at higher efficiency.

![Typical centrifugal compressor flow passage](image)

**Figure 1.1: Typical centrifugal compressor flow passage**

### 1.2 Goals and Thesis Layout

The goal of this thesis is to discuss the new CFD technique for incompressible flow and design of novel centrifugal compressor. The thesis layout is presented below:

1. Introduction

2. Novel Computational fluid dynamic technique for Incompressible flow

   (a) Governing differential equations
Figure 1.2: 30° backswept test impeller used by Krain
(b) Primitive and Non-Primitive formulations
   i. The Artificial Compressibility method
   ii. The Pressure Poisson Equation
   iii. The Stream-function Vorticity and Velocity Vorticity formulation
(c) Mathematical formulation of new scheme
   i. Continuity equation modification
   ii. Finite difference discretization
   iii. Boundary conditions
(d) Results and Discussion
   i. Fully Developed Flow Inside a Rectangular Channel
   ii. Backward-facing Step Flow
   iii. Driven Cavity Flow
   iv. 3D Backward-facing step
(e) Conclusion and future work

3. Design of Novel Centrifugal Compressor flow path
   (a) Modified Centrifugal Compressor
   (b) Methodology
      i. Flow path generation
      ii. 1D Design
   (c) Optimization statement
      i. Objective
      ii. Genetic Algorithm
      iii. Optimization results
   (d) Results and discussions
      i. Computational fluid dynamic simulation results
      ii. Alternate designs

4. Conclusion and Future work
Chapter 2

Novel Computational Fluid Dynamic Technique for Incompressible flow

2.1 Governing Differential Equations

The non-linear partial differential equations governing the motion of incompressible Newtonian fluid are given in vector form as follows

Momentum,

\[
\frac{\partial \vec{v}}{\partial t} + \nabla \cdot F + \nabla P = \nabla \cdot (\nu \nabla \vec{v}) + S
\]  

(2.1)

Continuity,

\[
\nabla \cdot \vec{v} = 0
\]  

(2.2)

where \( \vec{v} \) is the velocity vector; \( P = \frac{P}{\rho} \) is pressure is the normalized fluid pressure; \( P \) is the physical pressure and fluid density is represented by constant \( \rho \); \( \nu \) is the kinematic viscosity of fluid; \( S \) is the nonlinear algebraic source term and \( F_c \) is the flux tensor of the nonlinear convective terms, namely:

\[
F_c = \begin{pmatrix}
u u & uv & uw \\
u u & vv & vw \\
u u & vw & ww
\end{pmatrix}
\]  

(2.3)

where \( u, v \) and \( w \) are the velocity components in \( x, y \) and \( z \) directions respectively.
In two dimension Cartesian coordinate system, the governing equations are as follows

Continuity,

\[ \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \]  \hspace{1cm} (2.4)

x-Momentum,

\[ \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -P_x + \frac{1}{Re} \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \]  \hspace{1cm} (2.5)

y-Momentum,

\[ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -P_y + \frac{1}{Re} \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \]  \hspace{1cm} (2.6)

where $Re$ is the Reynolds number. Equations (1.4), (1.5) and (1.6) with appropriate boundary conditions are solved for and in the primitive variable formulation. Pressure gradient is appearing in the momentum equation as source term and hence velocity and pressure are coupled but there is no coupling between pressure field and continuity equation which is a constraint on velocity field and must be satisfied at any instant. If a correct pressure field is already known, the velocity field from the solution of equation (1.1) and (1.2) will automatically satisfy the continuity equation (1.2), however that is not the usual case and pressure field is unknown. Therefore, continuity equation (1.2) must be mathematically modified involving both pressure and velocity and the modification has to be consistent with the physics of the flow. The primitive and non-primitive variable formulation division is shown in flowchart 2.1. A brief discussion on primitive and non-primitive variable formulation methods is presented for completion.
2.2 Primitive and Non-Primitive Formulations

2.2.1 The artificial Compressibility Method

Chorin (1967) proposed an artificial equation of state

\[ P = \beta^2 \rho \]  \hspace{1cm} (2.7)

and using a pseudo transient formulation of continuity equation (1.2), modified continuity equation is obtained by adding time derivative term for pressure

\[ \frac{\partial P}{\partial t} + \beta^2 (\nabla \cdot \vec{v}) = 0 \]  \hspace{1cm} (2.8)

where \( \beta \) is a pseudo compressibility constant which functions as a relaxation parameter and \( \vec{v} \) is the velocity vector. Artificial compressibility method for transient simulation cases is used with dual time stepping technique to drive the time dependent pressure term to machine zero and satisfy the original continuity equation (1.2) where machine zero is defined as value which the processor of the computing machine assumes as equal to zero. Machine zero value depends on the kind of processor and storage type. The modified continuity equation (1.8), together with momentum equations (1.5) and (1.6) can be written as
\[
\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} - \frac{1}{Re} L \nabla^2 Q = 0 \quad (2.9)
\]

where

\[
Q = \begin{pmatrix} P \\ u \\ v \end{pmatrix} ; E = \begin{pmatrix} \beta^2 u \\ u^2 + P \\ uv \end{pmatrix} ; F = \begin{pmatrix} \beta^2 v \\ uv \\ u^2 + P \end{pmatrix} ; L = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.10)
\]

If \( \beta^2 \) is large, the different eigenvalues of above system differ significantly from one another, and the different modes of the solution decay at significantly different rates, and the system of equation becomes very stiff. For, explicit algorithms, this stiffness manifests itself as a severe stability restriction on \( \Delta t \). This, stability issue can be avoided using implicit algorithms, and for such cases, accuracy considerations suggest that

\[
\beta^2 < \frac{1}{\Delta t} \quad (2.11)
\]

On the other hand, if \( \beta^2 \) is made too small, the continuity equation is not satisfied to accepted accuracy levels and this destabilizes the transient solution.

### 2.2.2 The Pressure Poisson Equation

The Pressure- Poisson equation, a second order elliptical equation in Pressure is derived by taking the divergence of the momentum equation:

\[
\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} = -\left(\frac{\partial \xi}{\partial x} + \frac{\partial \eta}{\partial y}\right) + \frac{\partial}{\partial t}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) \quad (2.12)
\]

where

\[
\xi = u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} - \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \quad (2.13)
\]

\[
\eta = u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} - \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \quad (2.14)
\]
Equations (1.12) does not guarantee that computed velocity field will be divergence free since continuity equation (2) is has not been used in deriving these equations.

A new variable $D$ is defined as

$$D = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$$  \hspace{1cm} (2.15)

and to enforce continuity equation (1.2) constraint on the computed velocity field, the time derivative term is approximated as

$$\frac{\partial D}{\partial t} = -\frac{1}{\triangle t} (D)^n$$  \hspace{1cm} (2.16)

Equation (1.15) is obtained by setting dilation at $(t + \triangle t)$ time level as zero in order to enforce continuity equation, while retaining the dilation at $(t)$ time level\textsuperscript{11}. By incorporating equation (1.15) into equation (1.12), one can obtain

$$\frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} = -(\frac{\partial \xi}{\partial x} + \frac{\partial \eta}{\partial y}) + \frac{D^n}{\triangle t}$$  \hspace{1cm} (2.17)

Equation (1.12) is used to resolve incompressible flow on staggered grid and non-staggered grid, and from equation (1.15), it can be seen, that dilation decays exponentially with respect to time.

### 2.2.3 The Stream-Function Vorticity and Vorticity Velocity Formulation

In stream-function method which is a non-primitive formulation, pressure gradient terms from the momentum equations (5), (6) is removed by taking curl of the equation. The identity, curl of gradient of any scalar is zero, is used to eliminate pressure terms from momentum equations (5), (6).

Therefore

$$\frac{\partial \tilde{\omega}}{\partial t} + \tilde{\nu} \nabla \tilde{\omega} = \frac{1}{\text{Re}} \nabla^2 \tilde{\omega}$$  \hspace{1cm} (2.18)

The biggest advantage of stream-function formulation is that pressure is completely removed from Navier-Stokes equation, hence any difficulty associated with its boundary conditions. Also, stream function is analytical solution of continuity equation by its definition, and now the user only need to solve two equations (1.17) and (1.18).
However, specification of boundary conditions for vorticity field is not trivial and can be easily accomplished only for simpler case. For example in a situation where an external flow is adjacent to solution domain, vorticity at the boundary can be easily set to zero, but in other situations, vorticity value at the wall is not easily set and this creates issues with convergence. The biggest disadvantage of stream-function formulation is non-existence of stream-function for three dimensions and hence this method cannot be easily applied for solution of three dimensional cases. In order to extend stream-function formulation for three dimensional cases, an additional stream function is required since the two stream-functions are basically stream surfaces.

2.3 Mathematical formulation of new scheme

2.3.1 Modified continuity equation method

From the previous discussion about various existing methods, it is obvious that they solve problems mainly associated with pressure gradient term. However, either these methods have limited scope (applicability to only 2D cases) or they fail to enforce continuity equation to machine zero level on collocated grids. These two major issues are solved in the new approach developed here. In the new scheme, time derivative of pressure gradient term is added and continuity equation is modified by introducing additional spatial derivative terms as shown in equations 2.19, 2.20 and 2.21.

The continuity equation is modified as

\[
\frac{\partial P_x}{\partial t} + \frac{\partial}{\partial x} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) = 0
\] (2.19)

\[
\frac{\partial P_y}{\partial t} + \frac{\partial}{\partial y} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) = 0
\] (2.20)

\[
\frac{\partial P_z}{\partial t} + \frac{\partial}{\partial z} \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) = 0
\] (2.21)

At steady state, time derivative of pressure gradient terms i.e. \( \frac{\partial P_x}{\partial t} \), \( \frac{\partial P_y}{\partial t} \) and \( \frac{\partial P_z}{\partial t} \) goes to zero. Therefore equation 2.19, 2.20 and 2.21 at steady state reduces to

13
A new variable dilation is defined as

\[ D = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \]  

(2.23)

Therefore, equation 2.23 can be written as

\[ \frac{\partial D}{\partial x} = 0 \]
\[ \frac{\partial D}{\partial y} = 0 \]
\[ \frac{\partial D}{\partial z} = 0 \]  

(2.24)

The only possible solution of equation 2.24 is \( D = k \) where \( k \) is some arbitrary constant. This is very important outcome this gives the user an ability to use dilation as a boundary condition. Also, dilation is nothing but the divergence free velocity constraint which is equal to zero. Hence ply setting \( k \) as zero on the boundaries will set \( D = 0 \) throughout the computational domain. Equations 2.19, 2.20 and 2.21 are then solved for \( P_x, P_y \) and \( P_z \) with boundary condition \( D = 0 \) on all the boundaries of solution domain. The algorithm of the modified continuity equation is presented in flowchart 2.3.1.
2.3.2 Finite Difference Discretization

Central finite difference discretization scheme is used to formulate explicit solver of governing equations for space derivative terms while first order forward Euler discretization scheme is used for time derivative terms. Second order accuracy is ensured throughout the domain. Matlab 2013a software package from Mathworks [26] is used to implement new method. Modified continuity equation in 2D can be discretized as follows
\[
\frac{P_{n+1}^{x(i,j)} - P_n^{x(i,j)}}{\Delta t} + \frac{D_n^{(i+1,j)} - D_n^{(i-1,j)}}{2\Delta x} = 0 \quad (2.25)
\]

and

\[
\frac{P_{n+1}^{y(i,j)} - P_n^{y(i,j)}}{\Delta t} + \frac{D_n^{(i,j+1)} - D_n^{(i,j-1)}}{2\Delta y} = 0 \quad (2.26)
\]

Momentum equations are discretized in a similar fashion

\[
\frac{u_{n+1}^{i,j} - u_n^{i,j}}{\Delta t} + u_n^{i,j} U_x + v_n^{i,j} U_y = -P_{x(i,j)} + U_{xx} + U_{yy} \quad (2.27)
\]

\[
\frac{v_{n+1}^{i,j} - v_n^{i,j}}{\Delta t} + u_n^{i,j} V_x + v_n^{i,j} V_y = -P_{y(i,j)} + V_{xx} + V_{yy} \quad (2.28)
\]

The differential terms of equation 2.9 and 2.10 are discretized as

\[
U_x = \frac{u_n^{(i+1,j)} - u_n^{(i-1,j)}}{2\Delta x}; \quad V_x = \frac{v_n^{(i+1,j)} - v_n^{(i-1,j)}}{2\Delta y}
\]

\[
U_y = \frac{u_n^{(i,j+1)} - u_n^{(i,j-1)}}{2\Delta y}; \quad V_y = \frac{v_n^{(i,j+1)} - v_n^{(i,j-1)}}{2\Delta x}
\]

\[
U_{xx} = \frac{u_n^{(i+1,j)} + u_n^{(i-1,j)} - 2u_n^{(i,j)}}{\Delta x^2}; \quad V_{xx} = \frac{v_n^{(i+1,j)} + v_n^{(i-1,j)} - 2v_n^{(i,j)}}{\Delta y^2}
\]

\[
U_{yy} = \frac{u_n^{(i,j+1)} + u_n^{(i,j-1)} - 2u_n^{(i,j)}}{\Delta y^2}; \quad V_{yy} = \frac{v_n^{(i,j+1)} + v_n^{(i,j-1)} - 2v_n^{(i,j)}}{\Delta x^2}
\]

### 2.3.3 Boundary conditions

Different cases considered for validation of new scheme are

1. Developed flow in a rectangular duct for \( Re = 1000 \).

2. Backward facing step

3. Lid driven cavity for \( Re = 100, 400 \) and \( 5000 \).

The boundary conditions used for different cases are as follows
1. Rectangular duct flow

\[
\begin{align*}
\text{Inlet : } & \quad u = 1 \\
\text{wall : } & \quad u = v = 0; \quad D = 0 \\
\text{Outflow : } & \quad \frac{\partial u}{\partial x} = 0; \quad \frac{\partial v}{\partial x} = 0
\end{align*}
\]

(2.30)

2. Lid driven cavity

\[
\begin{align*}
\text{Top wall : } & \quad u = U_{Lid} \\
\text{other walls : } & \quad u = v = D = 0
\end{align*}
\]

(2.31)

3. Backward facing step

\[
\begin{align*}
\text{Lower Plate : } & \quad u = 0; \quad v = 0; \quad on \quad y = 0.5, 0 \leq x \leq 10 \\
\text{Upper Plate : } & \quad u = 0; \quad v = 0; \quad on \quad y = -0.5, 0 \leq x \leq 10 \\
\text{Inflow : } & \quad u = 24y(0.5 - y); \quad v = 0; \quad on \quad x = 0, 0 \leq y \leq 0.5 \\
\text{Step : } & \quad u = 0; \quad v = 0; \quad on \quad x = 0, -0.5 \leq y \leq 0.5 \\
\text{Outflow : } & \quad \frac{\partial u}{\partial x} = 0 \quad \frac{\partial v}{\partial x} = 0; \quad on \quad x = 0, 0 \leq y \leq 0.5 \\
\text{Walls : } & \quad D = 0
\end{align*}
\]

(2.32)

Neumann boundary conditions on velocity gradients at outflow are discretized using one way second order accurate finite difference scheme as

\[
\begin{align*}
\frac{\partial u}{\partial x} &= 3u_{i,j} - 4u_{i-1,j} + 4u_{i-2} \\
\frac{\partial v}{\partial x} &= 3v_{i,j} - 4v_{i-1,j} + 4v_{i-2}
\end{align*}
\]

(2.33)

2.3.4 Validation of new continuity scheme

2.3.4.1 Fully developed flow inside a rectangular duct

The first classical validation case I considered is the flow inside a rectangular duct. This is one of the few problems in fluid dynamics which have an exact analytical solution from Navier-Stokes equations which makes it a perfect test case for validation. A fully developed flow problem was solved using the new technique for Reynolds no 1000. Zero dilation was imposed at inlet, top and bottom wall and outlet. The velocity profile obtained after convergence agrees excellently with the analytical solution of fully developed flow. Unit mass flow rate is imposed at the inlet of the duct. Figure 2.1 shows the schematic of internal
flow through a rectangular duct. The fluid enters from the left and boundary layer starts growing eventually filling the complete duct with maximum boundary layer thickness equal to the half of the width of channel. The entrance length denoted by $L_e$ is a function of Reynolds number, defined as

$$\frac{L_e}{D_H} = 0.06\Re_D$$

(2.34)

where $H$ is the height of the duct, $D_H$ is the hydraulic diameter and $\Re_D$ is the Reynolds number based on hydraulic diameter of the duct. The length of the channel was 50 times the height to ensure fully developed flow at the outlet. Governing equations for flow inside a rectangular duct can be solved using proper boundary conditions. The analytical solution of fully developed velocity profile is given as

$$u = \frac{H^2}{2\mu} \left( \frac{\partial P}{\partial x} \right) \left[ \left( \frac{y}{H} \right)^2 - \left( \frac{y}{H} \right) \right]$$

(2.35)

Also, the maximum velocity at the centre of the duct is given as $u_{max} = \frac{3}{2}U_\infty$, where $U_\infty$ represents the free stream velocity which is one unit in present case.

Figure 2.2 shows the fully developed parabolic velocity profile at the outlet which matches exactly with analytical solution of outlet velocity profile as per equation 2.35. The contour of horizontal velocity inside the domain is shown in the figure 2.3 and boundary layer development can also be observed from this plot.

Figure 2.4 shows the distribution of dilation inside the duct at convergence and convergence history of channel flow simulation is shown in figure 2.4. Dilation and time derivative of pressure gradient terms are reduced to machine zero level at convergence. The residual of dilation is defined as the absolute maximum value inside the domain. Residual of pressure gradient time derivative terms is defined in a similar manner.
Figure 2.2: Outlet velocity profile $R_e = 1000$

Figure 2.3: Horizontal velocity contour
The maximum value of error in Dilation and time derivative of pressure gradient terms after convergence is of the order of $10^{-14}$. The decay of dilation is shown in figures 2.5, 2.6 and 2.7.

Figure 2.4: Convergence history of error in maximum Dilation, $\frac{\partial P_x}{\partial t}$ and $\frac{\partial P_y}{\partial t}$

Figure 2.5: Distribution of dilation inside the duct after initial iterations
Figure 2.6: Distribution of dilation inside the duct at mid convergence

Figure 2.7: Distribution of dilation at end of convergence
2.3.4.2 Backward-facing step

Flow over a backward facing channel is another popular benchmark problem. The geometry of the domain is shown in figure 2.8[27]. The recirculation region length, primary vortex strength and location are compared with results from other authors and result is presented in Table 1. Convergence history for coarse (200X20) and fine grid (2000X200) is shown in figure 5, with maximum error in dilation after convergence being of the order of $10^{-14}$. The present results are in very good agreement with existing literature. Fig 2.9 represents the contours of u-component of velocity inside the step representing recirculating region. Contours of dilation inside the domain respectively for a grid size of (2000X200) is show in figure 2.10. The dilation at convergence is reduced to machine zero level and it varied from $-8e^{-14}$ to $2e^{-14}$. Results for different grid sizes are compared with work done by G.A Reis, I.V.M. Tasso, L.F. Souza and J.A. Cuminato in table 2.1 who used exact projection method for Navier-Stokes incompressible flow equations on a staggered grid with fourth-order spatial precision[27].

![Figure 2.8: Backward facing step](image1)

![Figure 2.9: $u$ velocity contour for flow in backward facing step](image2)

Convergence history is represented in figure 2.11. The maximum value of error in dilation is reduced to
Figure 2.10: Distribution of dilation at convergence for flow inside backward facing step

$2e^{-14}$, while time derivative of pressure gradient terms namely $\frac{\partial P_x}{\partial t}$ and $\frac{\partial P_y}{\partial t}$ are reduced to the order of $10^{-12}$. The residual values stabilized after $3.5e^6$ iterations.

Figure 2.11: Convergence history for flow inside a backward facing step
<table>
<thead>
<tr>
<th></th>
<th>length of recirculation, $x_r$</th>
<th>$\psi_{min}$</th>
<th>$\psi_{min}$ location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barragy [28]</td>
<td>2.67</td>
<td>-0.0331</td>
<td>(1.002,-0.203)</td>
</tr>
<tr>
<td>Chinchapatnam et al [29]</td>
<td>2.72</td>
<td>-0.0315</td>
<td>(1.333,-0.217)</td>
</tr>
<tr>
<td>Bourantas et al. [30]</td>
<td>2.64</td>
<td>-0.0331</td>
<td>(1.000,-0.200)</td>
</tr>
<tr>
<td>Loukopoulos et al. [31]</td>
<td>2.66</td>
<td>-0.0330</td>
<td>(1.001,-0.207)</td>
</tr>
<tr>
<td>G.A. Reis et al.[27] (200 X 20)</td>
<td>2.62</td>
<td>-0.0328</td>
<td>(0.993,-0.203)</td>
</tr>
<tr>
<td>G.A. Reis et al. [27] (400 X 40)[27]</td>
<td>2.64</td>
<td>-0.0330</td>
<td>(0.992,-0.201)</td>
</tr>
<tr>
<td>G.A. Reis et al. [27] (600 X 60)</td>
<td>2.65</td>
<td>-0.0331</td>
<td>(0.992,-0.201)</td>
</tr>
<tr>
<td>G.A. Reis et al. [27] (800 X 80)</td>
<td>2.65</td>
<td>-0.0331</td>
<td>(0.992,-0.199)</td>
</tr>
<tr>
<td>G.A. Reis et al.[27] (1000 X 100)</td>
<td>2.66</td>
<td>-0.0331</td>
<td>(0.992,-0.199)</td>
</tr>
<tr>
<td>G.A. Reis et al. [27] (2000 X 200)</td>
<td>2.66</td>
<td>-0.0331</td>
<td>(0.992,-0.199)</td>
</tr>
<tr>
<td>Present work (200 X 20)</td>
<td>2.625</td>
<td>-0.0327</td>
<td>(0.995,-0.203)</td>
</tr>
<tr>
<td>Present work (400 X 40)</td>
<td>2.645</td>
<td>-0.0328</td>
<td>(0.994,-0.201)</td>
</tr>
<tr>
<td>Present work (800 X 80)</td>
<td>2.6625</td>
<td>-0.0331</td>
<td>(0.992,-0.199)</td>
</tr>
<tr>
<td>Present work (1000 X 100)</td>
<td>2.6625</td>
<td>-0.0331</td>
<td>(0.992,-0.199)</td>
</tr>
<tr>
<td>Present work (2000 X 200)</td>
<td>2.6625</td>
<td>-0.0331</td>
<td>(0.992,-0.199)</td>
</tr>
</tbody>
</table>

Table 2.1: Numerical results for backward facing step, $\nu = 5e^{-3}$, Comparison of recirculation region length $x_r$, primary vortex strength $\psi_{min}$ and $\psi_{min}$ location
2.3.4.3 Driven Cavity flow

The laminar incompressible flow in a square cavity has been used extensively as a model problem for validation of numerical techniques because of its simple geometry and complex flow structure. The driven cavity problem is easy to code because of simple computational domain but the flow retains all complex physical phenomena with appearance of counter rotating vortices at the corners of the cavity.

The top wall moves with a uniform unit velocity in right direction in its own plane with no slip walls on left, right and bottom wall. Figure 2.12 shows the computational geometry and uniform mesh generated for numerical simulation. Zero dilation is imposed at the wall boundaries. Numerical solutions for Reynolds number 100, 400, 1000 and 5000 are presented and discussed. Uniform mesh is used for every Reynolds number and second order accuracy is conserved throughout the domain.

![Figure 2.12: Geometry and computational mesh for driven cavity problem](image)

Convergence history for dilation and time derivative terms of pressure gradient computed using the new method are shown in figure 2.13 for Reynolds numbers 100, 400, 1000 and 5000. For the first, in the history of computational fluid dynamics machine zero level convergence is obtained for continuity equation. There have been many studies in the past to minimize the dilation on collocated grids and most notable work in done by Sotiropoulos and Abdallah [6] where they minimized the error in dilation to $10^{-6}$ level. But as evident from the figure 2.13, for example the maximum error at Reynolds number 5000 is of the order of $10^{-14}$ and similar level of convergence is obtained for other Reynolds numbers as well. The residual in time derivative of pressure gradient terms is also reduced to machine zero level ensuring steady state.
approximation to be correct.
Figure 2.13: Convergence history for Reynolds number 100, 400, 1000 and 5000 respectively
A comparison case is setup using commercial software Ansys Fluent [20] for Reynolds number 400. Semi-implicit method for pressure linked equation SIMPLE algorithm and second order upwind discretization for momentum terms is used. The singular points at the top most corners have very high localized value of dilation even after convergence. Maximum and average value of dilation after 800000 iteration is shown in Figure 2.15. It is interesting to note that absolute maximum and average value of dilation is 38.62 and -0.01815 respectively for Reynolds number 400. The final distribution of dilation inside the cavity at Reynolds number 100, 400, 1000 and 5000 from the converged solution of Ansys Fluent is shown in figure 2.14.

![Dilation distribution at different Reynolds number using Ansys Fluent](image)

Figure 2.14: Dilation distribution at different Reynolds number using Ansys Fluent
Figure 2.15: Maximum and Average Dilation Convergence (Fluent)

<table>
<thead>
<tr>
<th>Reynolds Number</th>
<th>Mesh Size</th>
<th>Maximum error in Dilation</th>
<th>Maximum error in $\frac{\partial P}{\partial x}$</th>
<th>Maximum error in $\frac{\partial P}{\partial y}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>111X111</td>
<td>$2.38e^{-13}$</td>
<td>$5.807e^{-13}$</td>
<td>$5.567e^{-13}$</td>
</tr>
<tr>
<td>400</td>
<td>151X151</td>
<td>$3.98e^{-14}$</td>
<td>$1.315e^{-12}$</td>
<td>$1.377e^{-12}$</td>
</tr>
<tr>
<td>1000</td>
<td>251X251</td>
<td>$7.55e^{-12}$</td>
<td>$4.405e^{-11}$</td>
<td>$4.335e^{-11}$</td>
</tr>
<tr>
<td>5000</td>
<td>551X551</td>
<td>$3.71e^{-14}$</td>
<td>$1.58e^{-12}$</td>
<td>$1.578e^{-13}$</td>
</tr>
</tbody>
</table>

Table 2.2: Mesh details for lid driven cavity flow

A comparison of dilation at mid and post convergence is shown in figure 2.16 for Reynolds number 5000. Dilation is reduced drastically from $10^{-3}$ to $10^{-12}$. Also gradual decay of dilation inside the cavity is shown in figure 2.17 at different level of iterations. One important observation from the plot of dilation distribution is the new method solution does not suffer from singular points at the top most corners. This is very important especially for unsteady case since error in dilation will keep on accumulating at each time step and present method ensure machine zero level convergence. Also, the dilation is distributed in such a fashion that maximum value exists at the core of the fluid region.

Figure 2.16: Distribution of dilation at mid and final convergence showing decay from $1e^{-3}$ to $1e^{-12}$
Figure 2.17: Decay of dilation for Reynolds number 5000 inside the cavity at different iterations.
The velocity results of present scheme are compared with the ones obtained by Ghia et al. The solution obtained by Ghia et al. [32] used the stream-function vorticity formulation on a uniform mesh of $129 \times 129$. The horizontal velocity versus the ordinate at mid-length ($x=0.5$) and the vertical velocity versus abscissa at mid-height ($y=0.5$) are plotted at different Reynolds number in figure 2.18. The results presented are in excellent agreement with the reference.

Figure 2.18: Centre line velocity comparison for different Reynolds number
Although, for incompressible flow it is the differential pressure which is important but not the absolute pressure, the pressure gradient matrix obtained after convergence can be used to integrate to obtain pressure field. Using multipath integration technique, pressure field for Reynolds number 5000 is obtained and presented in figure 2.19.

Figure 2.19: Pressure field for Reynolds number 5000
The contour of absolute velocity, horizontal velocity and vertical velocity for each Reynolds number are shown in figure 2.20. As Reynolds number is increased the effect of viscous forces becomes negligible compared to the convective terms. The fluid core at Reynolds number 5000 is shifted towards the right direction that is the direction of lid velocity while for Reynolds number 100, due to dominating diffusive terms, the core is highly symmetrical with respect to the centre of the cavity.
Figure 2.20: velocity contours for different Reynolds numbers
Erturk et. al. [33] have presented an efficient numerical method and they presented steady state solutions of the cavity problem up-to Reynolds number 21,000 by implementing fine grid mesh. Also, according to them in order to obtain accurate results, a grid mesh larger than $257 \times 257$ have to be used for numerical solution. The details of mesh density and residual error for modified continuity method is given in table. The streamlines and vorticity computed from the velocity fields are compared with numerical results from Mehmet Sahin and Robert G. Owens [34]. They solved the Navier-Stokes equation in primitive variable formulation using novel-implicit cell-vertex finite volume method. The pressure term was eliminated in their study by multiplying the momentum equation with the unit normal vector of a control volume boundary and thereafter performing integration around that boundary. The streamlines computed here compare well with the results of Mehmet et al. as shown in figure 2.21. Present scheme captures the complex phenomena of counter-rotating primary and secondary vortex very efficiently. For lower Reynolds number, the vertical patterns in the bottom most corners are present and the third vortex has formed near the top left corner consistent with the results presented by Mehmet et al.

Figure 2.21: Stream function for Reynolds number 100, 400, 1000 and 5000
The vorticity results are shown in figure 2.22. As Reynolds number increases the effect of viscous forces decreases. Convective terms becomes dominant over viscous terms because of presence of $\frac{1}{Re}$ term in the momentum equation 2.1. Also the development of fluid field of different parameters is direct consequence of lid motion. The vorticity is maximum at the wall and is then diffused inside the cavity due to viscous forces. Therefore, as Reynolds number is increased , the core of the cavity starts behaving like a rigid body with almost zero angular momentum. At Reynolds number 500, the core of the fluid is having almost zero angular motion.

Figure 2.22: vorticity at Reynolds number 400, 1000 and 5000
The modified continuity equation used in present scheme satisfies the curl of gradient scalar identity. The curl of the gradient identity is shown in equation 2.36.

\[ \nabla \times (\nabla \phi) = 0 \tag{2.36} \]

The necessary and sufficient condition for equation 2.36 to hold true is that scalar field \( \phi \) has to second order differentiable. The curl of pressure gradient is

\[ \frac{\partial^2 P}{\partial x \partial y} = \frac{\partial^2 P}{\partial y \partial x} \tag{2.37} \]

The curl of pressure at geometric centre lines of the cavity for different Reynolds number is shown in figure 2.23 and the curl evaluated is of the order \( 10^{-13} \).

![Figure 2.23: Curl of Pressure profile at geometric centre lines at different Reynolds numbers](image_url)
2.3.5 Special Validation Cases (Fluid flow with Magnetic field and 3D simulations)

In order to check the stability of the new scheme, highly coupled Maxwell Navier-Stokes equations are solved for driven cavity problem at Reynolds number 100, and backward step flow. Also, a 3D case for backward facing step is solved and presented. The modified continuity scheme presented works very well in 3D case and also for solving Magnetohydrodynamics equations.

2.3.5.1 Maxwell Equations

Given, an electrically conducting fluid moving with velocity $V$ which is subjected to a magnetic field $B$ perpendicular to the direction of flow. This magnetic field will induce an electric field. Equations governing electrodynamics are given as

\[ E = V \times B \] (2.38)

\[ J = \sigma (V \times B) \] (2.39)

\[ F_B = J \times B \] (2.40)

\[ \nabla \cdot E = \frac{\rho_e}{\varepsilon_0} \] (2.41)

\[ \nabla \times B = \mu (J + \varepsilon_0 \frac{\partial E}{\partial t}) \] (2.42)

where $E$ is the induced electric field, $J$ is the current density, $F_B$ is the Lorentz force, $\sigma$ is the conductivity of the fluid and $\mu_e$ is the magnetic permeability of the fluid.

The transport equation for magnetic field is given as

\[ \frac{\partial B}{\partial t} + (V \cdot \nabla)B = \nabla \left( \frac{1}{\mu_e \sigma} \nabla B \right) + (B \cdot \nabla)V \] (2.43)

Magnetic field consist of two part $B = B_0 + b$ where $B_0$ is the externally applied field and $b$ is the induced
magnetic field due to the relative motion of the conductive fluid in magnetic field.

Numerical simulations are done for flow inside a cavity and backward facing step under the influence of electromagnetic field for fluid with low Hartmann number. Hartmann number is defined as

$$H_a = BL \sqrt{\frac{\sigma}{\nu_e}} \quad (2.44)$$

Because of low Hartmann number, the induced electric field and magnetic field can be assumed to be negligible. The induced Lorentz force is parallel but in opposite direction to $V$. Applied magnetic field is represented as

$$B_0 = B_x \hat{i} + B_y \hat{j} \quad (2.45)$$

where $B_x$ and $B_y$ are component of magnetic field in $x$ and $y$ directions respectively. The Lorentz force then can be simplified as

$$F_B = \sigma [(u \hat{i} + v \hat{j}) \times (B_x \hat{i} + B_y \hat{j})] \times (B_x \hat{i} + B_y \hat{j})$$

$$= \sigma [(uB_x + vB_y)(B_x \hat{i} + B_y \hat{j})] - [(B_x^2 + B_y^2)(u \hat{i} + v \hat{j})]$$

$$= \sigma [(uB_x^2 \hat{i} + vB_x B_y \hat{j} + vB_x B_y \hat{i} + vB_y^2 \hat{j} - uB_x^2 \hat{i} - vB_y^2 \hat{j} - vB_x^2 \hat{j} - uB_y^2 \hat{j})]$$

$$= \sigma [(uB_x B_y \hat{j} + vB_x B_y \hat{i} - uB_x^2 \hat{i} - vB_y^2 \hat{j})]$$

$$= \sigma [(vB_x B_y \hat{j} - uB_x^2) \hat{i} + (uB_x B_y - vB_y^2) \hat{j}] \quad (2.46)$$

Momentum equations in $x$ and $y$ directions can then be written as

x-Momentum,

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -P_x + \frac{1}{Re} \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + \sigma (vB_x B_y - uB_y^2)$$ \quad (2.47)
y-Momentum,

\[
\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -P_y + \frac{1}{Re} \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + \sigma (u B_x B_y - v B_x^2)
\]  

(2.48)

Two new parameters \(a\) and \(b\) are defined as

\[
a = \sqrt{\frac{\sigma}{\rho} B_x}, b = \sqrt{\frac{\sigma}{\rho} B_y}
\]  

(2.49)

Hence, momentum equations can also be written as

\[
\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -P_x + \frac{1}{Re} \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + ab v - a^2 u
\]  

(2.50)

\[
\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -P_y + \frac{1}{Re} \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + ab u - a^2 v
\]  

(2.51)

Equations 2.50 and 2.51 are solved along with modified continuity equation to get velocity field. A test problem of backward facing step for \(a = 5\) and \(b = 5\) and driven cavity problem at Reynolds number 100 for \(a = 1\) and \(b = 1\) is solved. Figure compares 2.24 the horizontal velocity with and without magnetic field.
The separation length with and without magnetic field is compared in figure 2.25. One can observe that separation region is reduced drastically due to the effect of the externally applied magnetic field. This can be explained using the effect of Lorentz force on viscous force. The reason for fluid to separate is the shear stresses induced due to viscosity and if an ideal fluid with zero viscosity is studied, the separation will be eliminated completely. The terms appearing in momentum equation due to magnetic field are source term similar to the viscous terms. Hence, Lorentz force can be used to reduce effect of viscosity and that is the region of negligible separation observed in later case. Although, additional pockets of separation are
observed near the outlet of the domain as well.

Driven cavity problem for Reynolds number 100 with magnetic field is solved, for different values of Hartmann number. Figure 2.26 shows the convergence history of this test problem when \( a \) and \( b \) values are one each, and it can be observed that residual error is minimized to machine zero level for this case as well. The maximum value of error in dilation is of the order of \( 1e^{-13} \) which is represented in the figure 2.27 as final distribution of dilation inside the cavity.

![Figure 2.26: Convergence history for driven cavity under magnetic field at Re-100](image)

Figure 2.27: Distribution of Dilation inside the cavity for Re-100, under magnetic field

Important observations can be made by visualizing the development of fluid velocity inside the domain. The flow inside the driven cavity is developed and maintained due the viscous effects. The fluid layer in
contact with the lid owing to the no slip boundary condition moves with the lid velocity. The fluid layers are then, moved along the direction of flow due to viscosity. The fluid near the core of the fluid is therefore driven by the lid velocity due to viscous forces. However, under the magnetic field influence, the effect of viscous forces is reduced and the penetration of lid velocity is not significant as shown in figure. The effect of lid velocity and its penetration around the core of the lid is further reduced when competitively high amplitude magnetic field is used ($a = 10, b = 10$)

![Figure 2.28: Comparison of $u$ velocity with and without magnetic field at Reynolds number 100](image)

Figure 2.28: Comparison of $u$ velocity with and without magnetic field at Reynolds number 100
2.3.6 Grid Independence Study

Grid independence study is done for backward facing step flow. Different grid size considered were $200 \times 200$, $400 \times 400$ and $800 \times 800$. Length of recirculation, $\psi_{\text{min}}$ and location of minimum stream function variables are considered for grid independence verification. One can observe as grid size is reduced, number of iterations required to achieve machine zero level convergence increases. Table shows the values of different variables for each grid size. The $400 \times 400$ and $800 \times 800$ grid size produced the same results.

<table>
<thead>
<tr>
<th>Grid Size</th>
<th>Separation length</th>
<th>$\psi_{\text{min}}$</th>
<th>Location of $\psi_{\text{min}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>200X200</td>
<td>2.625</td>
<td>-0.0327</td>
<td>(0.995,-0.203)</td>
</tr>
<tr>
<td>400X40</td>
<td>2.6625</td>
<td>-0.0331</td>
<td>(0.992,-0.199)</td>
</tr>
<tr>
<td>800X80</td>
<td>2.6625</td>
<td>-0.0331</td>
<td>(0.992,-0.199)</td>
</tr>
</tbody>
</table>

Table 2.3: Grid dependence study for backward facing step
Chapter 3

Design of Novel Centrifugal Compressor

3.1 Modified Centrifugal compressor

The novel flow path of a multistage centrifugal compressor is presented in this study. Abdallah [35] patented a fluid controller concept which utilizes the efficient axial compressor flow path in a centrifugal compressor impeller as shown in Figure 3.1. The flow passage consists of two rotating stages separated by a stator stage. Multistage axial compressors are more efficient as compared to single stage because individual stages have low pressure ratio compared to overall pressure ratio. The high pressure ratio increases loading on blade on the compressor blades. The blade loading can be expressed by the diffusion factor. Therefore, for higher pressure ratio compressors, the design diffusion factor is high. Because isentropic efficiency is inversely proportional to diffusion factor, it is necessarily low.

\[ D.F = (1 - \frac{W_2}{W_2}) + \frac{r_2W_{\theta 2} - r_1W_{\theta 1}}{2W_1 \sigma} \]  

(3.1)

where \( W \) is the relative velocity of the fluid, \( r \) is the radius, \( 1 \) is the inlet section, \( 2 \) is the outlet section and \( \sigma \) represents solidity of blades.

Also when the blade angles are too extreme, and the rotating fluid is forced to turn to an excessive degree, the fluid fails to follow the blade and separates from the main flow, resulting into further losses and increasing the turbulence. This issue is resolved using the new design since presence of a stator row allows for more turning. The orientation of the second blade row does not depend on the orientation of the first blade row because the casing blade turns the fluid to a different orientation and velocity. The appropriate
configuration of individual stages will result into significantly higher enthalpy rise compared to a single continuous blade. With higher backswept angle at the exit, the absolute velocity is lower and a smaller size diffuser is required. The present work considers one stator row and two rotating blade rows but is not limited to this configuration only and can be extended to plurality of compression-diffusion stages. Also, this configuration is not only limited for compressors but can be applied in pumps as well. This design is also suitable for multi-phase fluids to avoid cavitation.

Figure 3.1: novel centrifugal compressor.

Figure 3.2: flow path of new compressor
3.1.1 Methodology

The single stage compressor baseline is derived from Nasa low speed centrifugal compressor [36]. Figure 3.3 shows the baseline model of LSCC. The design parameters of the LSCC compressor are mentioned in Table 3.1. 3DBGB (3D Blade Geometry Builder) [37] tool is used to create the blade geometry and CFD simulations were performed using commercial software FINETURBO[38]. The stream-wise coordinates required for blade geometry generation were taken directly from the paper and the flow path of LSCC is show in Figure 3.4. It is impossible to recreate the exact LSCC compressor geometry and perform CFD simulations since many important parameters like camber line definition of airfoil section, blade thickness, flow angles are not available for LSCC model from the NASA paper. The original LSCC compressor is 92% efficient but owing to the reasons mentioned before, the baseline is designed for 82% efficiency. Also, the objective of the present study is not to recreate LSCC model but to compare the flow path of a classical centrifugal compressor with the new design. An optimization with respect to the curvature of the camber line is performed using DAKOTA [39] for the single stage as well multistage compressor. The camber line of optimum blade obtained is “S” shaped for single stage. The generated flow path of LSCC for multistage compressor design is then split into 2 rotating and one stationary blade row in such a manner that 1st blade row is similar to an axial stage while the 2nd rotor stage is having radial inlet and radial outlet. The flow path of multistage compressor is shown in Figure 3.5. The stator turning angle is 20°. The working fluid (Air) is assumed to behave as a real gas and Spalamart- Allmaras turbulence model is used for all CFD simulations. Total pressure and temperature are specified at the inlet and constant mass flow is imposed at outlet as boundary conditions. Figure 3.6 and 3.7 shows the final geometry of multistage and single stage compressor obtained from 3DBGB.

Table 3.1: LSCC Design Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rpm</td>
<td>1862</td>
</tr>
<tr>
<td>No of vanes</td>
<td>20</td>
</tr>
<tr>
<td>Backswept angle</td>
<td>55</td>
</tr>
<tr>
<td>efficiency (%)</td>
<td>92</td>
</tr>
<tr>
<td>Working fluid</td>
<td>Air</td>
</tr>
<tr>
<td>Pressure ratio</td>
<td>1.16</td>
</tr>
<tr>
<td>P0(Pascals)</td>
<td>101325</td>
</tr>
<tr>
<td>T0(Kelvin)</td>
<td>288.15</td>
</tr>
<tr>
<td>mass flow rate (kg/s)</td>
<td>30</td>
</tr>
</tbody>
</table>
Figure 3.3: LSCC Compressor used for deriving the baseline geometry.

3.1.2 1D Design

1D design calculation is done to obtain blade metal angles and input parameters for 3DBGB like blade metal angles, absolute and relative Mach numbers to create the geometries of compressors. From the design data of baseline compressor, the temperature ratio is evaluated using equation 3.2

\[ P_R = \frac{P_{02}}{P_{01}} \frac{T_{02}}{T_{01}} = (1 + \eta_{rot}(T_R - 1))^{\frac{\gamma}{\gamma - 1}} \]  

where

\[ P_R = \frac{P_{02}}{P_{01}}, T_R = \frac{T_{02}}{T_{01}} \]

The work input to the compressor is evaluated from the \( T_R \) value using equation 3.3

\[ h_{rise} = C_F T_{01} (T_R - 1) \]  

The Euler-Turbomachinery equation relates the total change in angular momentum of the fluid to the
Figure 3.4: LSCC flow path.
Figure 3.5: Flow path of Multistage compressor.

Figure 3.6: Multistage Compressor Geometry.
Figure 3.7: Single stage compressor geometry.
enthalpy rise as

\[ h_{\text{rise}} = U_2 V_{\theta 2} - U_1 V_{\theta 1} \] (3.4)

The exit tangential velocity \( V_{\theta 2} \) is calculated from equation 3.4. The tangential component of inlet velocity \( V_{\theta 1} \) is zero because of axial inlet. Using an iterative scheme, inlet density is calculated. The velocity triangle at the inlet can be fixed using equation 3.5

\[ \rho_{\text{inlet}} = \frac{P_1}{RT_1} \] (3.5)

and

\[ m = \rho_{\text{inlet}} V_m A \quad , A = 2\pi r_{le} p_{le} \]

Different velocity components and blade metal angles are evaluated using equation 3.6. The velocity triangle of a turbo-machine is shown in figure 3.8

\[ \vec{V} = \vec{U} + \vec{W} \] (3.6)

\[ \vec{V} = V_{\hat{k}} \hat{k} + V_{\hat{i}} \hat{i} + V_{\theta \hat{i}} \theta \]

\[ \vec{W} = W_{\hat{k}} \hat{k} + W_{\hat{i}} \hat{i} + W_{\theta \hat{i}} \theta \]

\[ \vec{U} = \vec{\omega} \times \vec{r}, W_{\theta} = V_{\theta} - U, \]

\[ \tan(\alpha_m) = \frac{V_{\theta}}{V_m} \]

\[ \tan(\beta_m) = \frac{W_{\theta}}{W_m} \]

The multistage compressor is designed for same work input and this work input is distributed among the rotor stages. Since, the 1st stage rotor is having an axial design and the radius is almost constant along the flow passage, the net pressure rise across it is low. Moreover, due to large change in radius, maximum amount of work done on the fluid is done by rotor 2. Therefore, the total amount of work is split in such a way that 20% is done by rotor 1 and 80% by rotor 2. The temperature ratio across each rotor stage is
calculated using equation 3.7

\[ T_{RM1} = 1 + \lambda (T_R - 1) \]  
\[ T_{RM2} = 1 + \frac{(1 - \lambda)(T_R - 1)}{T_{RM1}} \]

where

\[ \lambda = \frac{h_{rise1M}}{h_{rise}}. \]

The maximum value of pressure ratio possible from each stage can be evaluated by defining efficiency as a function of pressure and temperature ratio and constraining it to be less than 1. The maximum pressure ratio achievable from rotor stage 1 and stage 2 using efficiency constraint is 1.03747 and 1.15418 respectively. The individual stages of multistage compressor are then designed for 90% of these values. Different design parameters for multistage compressor thus evaluated are mentioned in Table 3.2.

\[ \eta_{\text{rotor}} = \frac{(P_{RM_1} - T_1 - 1)}{T_{RM_1} - 1} \]  

3.1.3 Computational Fluid Dynamics

Numeca Fine/Turbo has been used to perform Computational Fluid Dynamic (CFD) simulation and AutoGrid5 is used for structured mesh generation [38]. The Spalart-Allmaras (SA) turbulence model is used for
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Pressure ratio (stage 1)</td>
<td>1.035</td>
</tr>
<tr>
<td>Pressure ratio (stage 2)</td>
<td>1.14</td>
</tr>
<tr>
<td>Temperature ratio (stage 1)</td>
<td>1.01056</td>
</tr>
<tr>
<td>Temperature ratio (stage 2)</td>
<td>1.04717</td>
</tr>
<tr>
<td>efficiency (stage 1)</td>
<td>0.9348</td>
</tr>
<tr>
<td>efficiency (Stage 2)</td>
<td>0.912167</td>
</tr>
<tr>
<td>Total Pressure ratio</td>
<td>1.1799</td>
</tr>
<tr>
<td>Total efficiency</td>
<td>0.86</td>
</tr>
</tbody>
</table>

Table 3.2: Multistage compressor 1D calculation results.

analysis while assuming steady flow in the relative frame of reference. SA turbulence model is developed for aerospace applications. It is a one equation model and is extremely efficient in simulating flows with adverse pressure gradient [40] [41]. Total number of grid points for CFD simulation of single stage are 1019694 and 2116172 for multistage compressor. Conservative boundary conditions are imposed to simulate rotor-stator interaction for multistage design. Blade to blade computational grid is shown in figure 3.9 and figure 3.10 for single and multistage compressor respectively. The first cell wall distance has been kept small enough to efficiently capture the flow phenomena inside the boundary layer. Figure 3.11, figure 3.12 and figure 3.13 represents $y^+$ values at hub, casing and blades respectively. The SA model requires the viscous viscosity affected boundary layer to be properly resolved and hence the $y^+$ values are kept small.

The Reynolds number for centrifugal compressor is evaluated at the trailing edge of the rotor. Using the exist tip speed, inlet dynamic viscosity, inlet total density and exit tip diameter, the Reynolds number evaluated is $163.05 \times 10^5$.

3.1.4 Aerodynamic Optimization

Since the original camber line of LSCC is not available, the optimization was performed for a single stage as well multistage compressor to obtain camber line definition corresponding to maximum isentropic efficiency. The optimization is performed using genetic algorithm for a single objective function. Several control points are used in the chordwise direction to create a cubic B-spline which is defining the second derivative of airfoil section. The chordwise control points are also connected through another cubic B-spline in spanwise direction. Additional control points are used to control this cubic B-spline, thus providing for smooth spanwise variation in airfoil curvature. The mass flux and total pressure ratio is constrained within 0.5% of the design point. 18 design variables on 3 sections namely hub, pitchline and tip with 3 curvature
Figure 3.9: Baseline compressor grid.
Figure 3.10: Multistage compressor computational grid.

Figure 3.11: Contour of $y^+$ at hub
Figure 3.12: $y^+$ contour at shroud

Figure 3.13: $y^+$ contour at blades
control point values are used. The middle curvature control point have a unit chord value. The population size for genetic algorithm is 12 and maximum iterations is 100.

3.1.5 Genetic Algorithm

GA is based on Darwin’s theory of survival of fittest and it starts with random generation of design points in the parameter space. The design points which are producing good result for the objective function are only allowed to survive and reproduce till convergence is achieved. John Eddy Genetic Algorithm library of DAKOTA is used. The .trb file generated by Autogrid5 and .iec file from FineTurbo is used by optimizer to get details of required parameters like computational mesh, boundary conditions etc. The objective of the optimization loop is to improve the efficiency of compressor by varying the curvature control points. Three curvature control points are varied along hub, mid and tip section to get new camber line definition. Figure 3.14 shows the optimization loop which is controlled by DAKOTA. Using forking interface in DAKOTA, CFD simulations are run by calling an external script. CFD results are then sent to DAKOTA for post processing. The geometric variables are defined by the user in 3DBGB template file.

Figure 3.14: Optimization loop.
3.1.6 Optimization Statement

- Maximize:
  - Isentropic efficiency for a single rotor

- Constraints:
  - Constraint the mass flow rate within 0.5% of the baseline
  - Constraint the total pressure ratio within 0.5% of the baseline

- Design Variables:
  - 3 sections having 18 variables, have 3 curvature control point values, middle curvature control point have unit chord value.

- Genetic Algorithm (GA) parameters:
  - population size of 12, and maximum iterations is 100.

- Constants:
  - All other parameters used for 1D design of flow path.

3.2 Results and Analysis

3.2.1 Optimization Results

The effect of airfoil curvature on pressure gradient is well explained by [42]. Nemnen[43] used cubic B-spline to integrate curvature into blade design process. Figure 3.15, Figure 3.16 and Figure 3.17 shows the control points and the cubic spline defining the spanwise variation in curvature for hub of baseline, 1st stage and 2nd stage rotor of multistage compressor. Similarly curvature at pitchline and tip of each stage is defined and shown. Figure 3.21, figure 3.22 and figure 3.23 shows the baseline and optimum blade shape at different sections for single stage, 1st row of multistage and 2nd row of multistage compressor respectively. It is interesting to see that the optimum camber line of single stage centrifugal compressor is “S” shaped. S shape camber line has many advantages as proved in the research done by Syed Moez et al. [44].
Figure 3.15: Spanwise cubic B-spline curvature control definition at hub of baseline.
Figure 3.16: Spanwise cubic B-spline curvature control definition at hub of 1st stage.

Figure 3.17: Spanwise cubic B-spline curvature control definition at hub of 2nd stage.
The cubic B-spline used to define the second derivative of the camber line is integrated to get the camber slope and is then again integrated to get the camber line definition. The baseline shape is having a curvature definition of a conventional airfoil and the curvature remains constant in the span direction. The optimized blade shows a conventional airfoil curvature at the hub section and an inflection in curvature definition from negative to positive and back to negative at the mid and tip section, which is responsible for the S-shaped blade sections at these locations. Three curvature control points in the middle and the chord location of the center control points are varied for all three sections.

### 3.2.2 Results and Discussion

Figure 3.24 and figure 3.25 shows the variation in static pressure at different blade to blade surface location for multistage and single stage machine. Due to higher achievable pressure ratio in multistage, the overall pressure rise is much higher compared to single stage compressor. This is visible near the hub, pitchline as well at the tip section. The static temperature and total temperature plots are also shown, and multistage compressor has higher temperature due to increase in overall pressure ratio. Also, from figure 3.26, we...
can observe that entropy generation is reduced for the new flow passage, resulting into higher efficiency and low losses. The relative mach no for multistage compressor is lower compared to a single stage as shown in figure. Figure 3.27 represents variation of relative mach number for single stage and multistage compressor at different blade to blade surface locations. The relative mach number for multistage compressor at the outlet is lower compared to the single stage as shown in figure 3.28. Lower relative velocity is direct consequence of increased turning of flow due to stator row. The stator converts the kinetic energy at the exit of first rotor blade into static enthalpy. This energy transformation Also, low velocity results into lesser entropy generation and reduced losses. Variation of absolute mach number is shown in figure 3.29 and it is evident that the absolute velocity at the exit of the multistage rotor is lower compared to one in single stage because of high enthalpy rise. Therefore, the size of the diffuser required further downstream will be smaller. Table 3.3 compares the converged CFD results for multistage and single stage compressor. The percentage gain in efficiency and pressure ratio is 8.55 and 2.97 respectively.
Figure 3.20: Spanwise cubic B-spline curvature control definition at pitchline of 1st row of novel centrifugal compressor

<table>
<thead>
<tr>
<th></th>
<th>Single stage</th>
<th>Multistage</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>efficiency</td>
<td>80.41</td>
<td>87.29</td>
<td>+8.55%</td>
</tr>
<tr>
<td>no of vanes</td>
<td>20</td>
<td>20-30-40</td>
<td></td>
</tr>
<tr>
<td>Pr</td>
<td>1.110</td>
<td>1.143</td>
<td>+2.97%</td>
</tr>
<tr>
<td>mdot</td>
<td>30</td>
<td>30</td>
<td></td>
</tr>
<tr>
<td>Tr</td>
<td>1.03</td>
<td>1.03</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3: Single stage vs Multistage (CFD Results).
Figure 3.21: Blade profiles of baseline and optimum design for single stage centrifugal compressor.
Figure 3.22: Blade profiles of baseline and optimum design for 1st rotor of multistage compressor.
Figure 3.23: Blade profiles of baseline and optimum design for 2nd rotor of multistage compressor.
Figure 3.24: Static Pressure at 1% and 5% blade to blade surface.
Figure 3.25: Static Pressure at 50% and 95% blade to blade surface.
Figure 3.26: Entropy at 1% and 5% blade to blade surface.
Figure 3.27: Relative Mach no at 50% and 95% blade to blade surface.

Figure 3.28: Relative Mach no plot at outlet.
Figure 3.29: Azimuthal absolute Mach no.

Figure 3.30: Entropy generation plot at outlet.
3.2.3 ALTERNATE DESIGNS

Different combinations of rotor stator stages can be utilized to create several alternate configurations to maximize the efficiency and pressure ratio of the centrifugal compressor. This exploration expands the design space to multistage configurations where the process of compression followed by diffusion is repeated successively.
Chapter 4

Conclusion and future works

4.1 Modified Continuity Equation method for Incompressible flow

A novel finite difference method approach based on primitive variable formulation is programmed and developed in Matlab and presented for the simple and efficient solution of incompressible Navier-Stokes equations. The method involves modification of continuity equation by introducing pressure gradient terms and solving momentum equation for primitive variables. Thus any difficulty associated with unknown pressure field or vorticity boundary conditions are eliminated. The most remarkable feature of the scheme method is that it enables us to satisfy continuity equation to machine zero and gives us freedom to use dilatation as a boundary condition. Popular validation problems, such as lid-driven cavity, flow inside a rectangular duct and flow in a backward step channel showed the robustness of the new scheme. The method is applied to lid-driven cavity flow problem for steady state solution at Reynolds number up to 5000, developed flow inside a duct at Reynolds number 1000 and flow in a backward facing step. Special validation cases are solved where effect of magnetic field is simulated for driven cavity flow problem and backward step flow. The solutions are non-oscillatory and are in perfect match with benchmark results in the literature. In particular, the curl identity of pressure gradient is verified for lid-driven case at each Reynolds number.

The extension of the present scheme to three dimensional situations is straightforward. Also, the present approach can be used for numerical solution of unsteady cases by introduction of dual time step method similar to that in artificial compressibility approach. Three dimensional analysis is achieved by solving the momentum equation in the Z direction and introducing another modified continuity equation to solve for additional pressure gradient term. The incompressible Navier-Stokes equation is solved explicitly, therefore
each derivative is calculated separately. It will be interesting to see the unsteady formulation using this algorithm since it imposes dilatation to be zero throughout the control volume and does not suffer from singularity at the corner points. Also, future work includes application of present scheme to non-curvilinear grid and adding multi-grid capability to the algorithm.

4.2 Novel Centrifugal Compressor flow path design

A novel multistage centrifugal compressor design is presented. The new design utilizes more efficient flow path of an axial compressor in a centrifugal impeller which incorporates the advantage of successive compression diffusion stages. The overall turning imparted to the fluid is increased due to the presence of a stator row between the rotating stages. This increased the overall efficiency and pressure ratio. The new concept also reduces the size of the diffuser required for further deceleration of the fluid since the absolute velocity of fluid exiting the impeller is low. The small size diffuser further reduces the losses. It is observed that the individual stages have very high efficiency because the pressure ratio of rotating stages is substantially less than the overall pressure ratio of the compressor. A low pressure ratio centrifugal compressor is selected as a baseline for comparison. The low pressure ratio is to match a large pressure rig. It is observed that multistage concept compressor have higher efficiency and pressure ratio than the single stage compressor mainly because it can take advantage of the exit swirl coming from the first rotor and extract additional static enthalpy rise in the stator row. The optimization of both single row and multistage configurations showed that there are optimum designs with efficiency very close to the 1D design analysis. The optimization was performed to obtain optimum camber line of blade rows using curvature control point because the camber line definition of LSCC compressor blade is not available. The blade designs obtained through this process can be used as a starting point for design and analysis of flow path consisting of different numbers of compression diffusion stages. Incorporating multistage in centrifugal compressor eliminates the requirement of splitter blades. Also an improved solidity can be achieved with this concept since number of blades used in the second rotor stage can be increased which is not possible in case of single stage.

Future work will include multistage compressor optimization for a range of angle of stator turning and corresponding efficiency and pressure ratio, 3-D CFD on the obtained flow path. Since the objective of the present study is to compare the new design with classical compressor, CFD simulations are only carried out at design point. It will be interesting to see the effect of multistaging concept on the operating range.
of the compressor. Multistaging is extensively used in axial compressor configuration and this study opens the door for similar study in centrifugal compressor. The present study serves as a starting point in this direction. A speed line analysis will be done for single stage as well multistage device to explore the effect of new concept on the operating range. One more important parameter to be considered in future studies is the clearance between the stator and rotor rows. The effect of this concept on the surge limit of compressor will also be studied. Also structural analysis for the multistage compressor will be done since it is important to consider the stresses in designing the new blades. A different design case where a high pressure ratio baseline is selected for comparison will also be investigated. The radial spacing between the different stages is crucial to the performance and is another area to be explored and a study about its effect on overall performance of compressor needs to be performed. Currently, the multistage design have separation and losses near the tip of the last rotor which needs to be improved.
Bibliography


[40] Allmaras, S. R., Johnson, F. T., and Spalart, P. R. “Modification and clarifications for the implementation of the spalart-allmaras turbulence model”.

[41] Spalart, P. R., and Allmaras, S. R. “A one-equation turbulence model for aerodynamic flows”.


Appendix A

Matlab Scripts for Computational Fluid Dynamics Validation Cases

A.1 Matlab script for Simulating Lid driven cavity flows

```matlab
1 clear all; clc; close all;
2 Lid Driven Cavity Solution for Non Uniform Grid Using Continuity Equation
3 tic
4 s1 = '! mkdir Lid_Driven_Cavity_RE_5000';
5 nx=441;ny=441;
6 len_x= 1; len_y= 1;
7 h= len_x/(nx-1);
8 del_t= 1e-7 ;
9 Uwall= 1;
10 f=0.1;
11 k=0;
12 beta=0.25;alpha=0.1;
13 if (alpha<0.15 && beta<0.5)
```
for Re=1000:100:1000
beta=0.16; alpha=0.001;
h = (2*beta)/(alpha*Re);
del_t = beta*2*h;

nx=1/(h)+1;
xn=floor(nx);
if rem(nx,2)== 0
nx= nx+1;
end
if nx<251
nx=101;
end
nx=101;
y=x

h = 1/(nx-1);
del_t = (Re*h*h)*alpha; beta = del_t/(2*h);
s2 = strcat(s1, '\Re', num2str(Re));

for c= 1:0.1:1
qx=zeros(nx,nx);
qy=zeros(nx,nx);
x=0:1/(nx-1):1;
vor = zeros(nx,ny);
u = zeros(nx,ny);
v = zeros(nx,ny);
D = zeros(nx,ny); Px = zeros(nx,ny); Py = zeros(nx,ny);
Px = Px; Py = Py;

del_Px = ones(nx,nx); del_Py = ones(nx,ny);
del_Px(1,:) = 0; del_Py(1,:) = 0;
end
u(1,:) = 1;
u_star = u; v_star = v;
s3 = strcat(s2, '\c', num2str(c*100));
k = 0;
clear DD PX PY xx
eval(s3);
while (k<1000 || max(max(D))>1e-13) ||
    max(max(del_Px)>1e-3)) ||
    max(max(del_Py)>1e-3)) ||
    min(min(del_Px))<-1e-3 ||
    min(min(del_Py))<-1e-3)
k=k+1;
if rem(k, 50000) == 0
    figure(10), contourf(D, 50, 'LineStyle', 'none');colorbar;
title(strcat('Distribution of continuity at Re = ', num2str(Re), 'Iteration = ', num2str(k)));
h10 = figure(10);
s4 = strcat(s3(9:end), '\k', num2str(k), '.jpg');
s5 = strcat(s3(9:end), '\k', num2str(k), '.fig');
saveas(h10, s4);
saveas(h10, s5);
end
u(1,:) = 1 - exp(-(k/10)*del_t);
u_star(1,:) = 1*(1 - exp(-(k/10)*del_t));
u(nx,:) = -1*(1 - exp(-(k/1.5)*del_t));
u_star(nx,:) = -1*(1 - exp(-(k/1.5)*del_t));
for j = 2:ny-1
    u_star(nx-1:-1:2,j) = u(nx-1:-1:2,j) + alpha*(u(nx-1:-1:2,j+1)+u(nx-1:-1:2,j-1)+u(nx-1:-1:3,j)+u(nx-2:-1:1,j)-4*u(nx-1:-1:2,j))...
    -del_t*Pz(nx-1:-1:2,j)/1 -u(nx-1:-1:2,j)*beta.*u(nx-1:-1:2,j+1)-u(nx-1:-1:2,j-1)-v(nx-1:-1:2,j)*beta.*u(nx-2:-1:1,j)-u(nx-1:-3,j))  
    v_star(nx-1:-1:2,j) = v(nx-1:-1:2,j) + alpha*(v(nx-1:-1:2,j+1)+v(nx-1:-1:2,j-1)+v(nx-1:-1:3,j)+v(nx-2:-1:1,j)-4*v(nx-1:-1:2,j))...
-del_t*Py(nx-1:-1:2,j)/1 -u(nx-1:-1:2,j)*beta.*(v(nx-1:-1:2,j+1)-v(nx-1:-1:2,j-1)) - v(nx-1:-1:2,j)*beta.*(v(nx-2:-1:1,j)-v(nx:-1:3,j));
end
j= 2:ny-1;
u_star(nx-1:-1:2,j)= u(nx-1:-1:2,j) + alpha*(u(nx-1:-1:2,j+1)+u(nx-1:-1:2,j-1)+u(nx-1:-1:3,j)+u(nx-2:-1:1,j) -4*u(nx-1:-1:2,j))
-del_t*Px(nx-1:-1:2,j)/1 -u(nx-1:-1:2,j)*beta.*(u(nx-1:-1:2,j+1)-u(nx-1:-1:2,j-1)) - v(nx-1:-1:2,j)*beta.*(u(nx-2:-1:1,j)-u(nx:-1:3,j));
v_star(nx-1:-1:2,j)= v(nx-1:-1:2,j) + alpha*(v(nx-1:-1:2,j+1)+v(nx-1:-1:2,j-1)+v(nx-1:-1:3,j)+v(nx-2:-1:1,j) -4*v(nx-1:-1:2,j));
-del_t*Py(nx-1:-1:2,j)/1 -u(nx-1:-1:2,j)*beta.*(v(nx-1:-1:2,j+1)-v(nx-1:-1:2,j-1)) - v(nx-1:-1:2,j)*beta.*(v(nx-2:-1:1,j)-v(nx:-1:3,j));
....Post processing........ %
u= u_star;
v= v_star;
77 Calculation of continuity Matrix
D(nx-1:-1:2,2:ny-1)= (u(nx-1:-1:2,3:ny)-u(nx-1:-1:2,1:ny-2)+v(nx-2:-1:1,2:ny-1)-v(nx:-1:3,2:ny-1))/2*h;
79 Calculation of Pressure Gradient Matrix
Px_star(nx-1:-1:2,2:ny-1)= Px(nx-1:-1:2,2:ny-1) - c*beta*(D(nx-1:-1:2,3:ny)-D(nx-1:-1:2,1:ny-2));
Py_star(nx-1:-1:2,2:ny-1)= Py(nx-1:-1:2,2:ny-1) - c*beta*(D(nx-2:-1:1,2:ny-1)-D(nx:-1:3,2:ny-1));
82 Calculation of Dpx/Dt & Dpy/Dt
del_Px= (Px_star- Px)/ del_t;
del_Py= (Py_star- Py)/ del_t;
Px=(1-f)*Px_star+ (f)*Px;
Py= (1-f)*Py_star+ (f)*Py;
....Post processing........ %
88  \texttt{xx(k)=k;}
89  \texttt{DD(k) = max(max(D));}
90  \texttt{PX(k)= max(max(del_Px));}
91  \texttt{PY(k)= max(max(del_Py));}
92  \texttt{fprintf(' The value of error in Continuity is \%f\n', max(max(D)))}
93  \texttt{end}
94  \texttt{figure(1), plot(xx,log10(DD),xx,log10(PX),xx,log10(PY));}
95  \texttt{legend('Continuity','Horizontal - Pressure - Gradient','Vertical - Pressure - Gradient')}
96  \texttt{fprintf(' The value of error in Continuity is \%f\n', max(max(D)))}
97  \texttt{title(strcat('Covergence at Re = ', num2str(Re), ' and c = ', num2str(c)))}.
98  \texttt{ylabel('Logarithmic Scale');}
99  \texttt{xlabel('Iteration Count');}
100 \texttt{h1 = figure(1);}
101 \texttt{vel= zeros(nx,ny);}
102 \texttt{vel(1,:)=Uwall;}
103 \texttt{%%%%% Calculation of Absolute Velocity %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%}
104 \texttt{vel( nx-1:-1:2, 2:nx-1)= sqrt(u(nx-1:-1:2,2:nx-1).*u(nx-1:-1:2,2:nx-1)-1)+v(nx-1:-1:2,2:nx-1).*v(nx-1:-1:2,2:nx-1));}
105 \texttt{%%%%}
106 \texttt{vor(nx,:)= - (u(nx-1,1:nx)-u(nx,1:nx))/(2*h); \% Bottom wall \%
107 \texttt{vor(nx-1:-1:1,nx)= - (v(nx-1:-1:1,nx)-v(nx-1:-1:1,nx-1))/(2*h); \%
108 \texttt{Right \ wall \%
109 \texttt{vor(1,nx-1:-1:1)= (u(1,nx-1:-1:1)-u(2,nx-1:-1:1))/(2*h); \% Top \ wall \%
109 \texttt{vor(2:nx-1,1)= (v(2:nx-1,2)-u(2:nx-1,1))/(2*h); \% Left \ wall \%}
109
109
FOR INTERNAL NODES

\[ \text{vor}(\text{nx-1}:-1:2,2:\text{nx-1}) = \left( (v(\text{nx-2}:-1:1,2:\text{nx-1})-v(\text{nx}:1,2:\text{nx-1}))-u(\text{nx-1}:-1:2,3:ny)-u(\text{nx-1}:-1:2,1:\text{nx-2})) / (2*h) \right); \]

```matlab
qx = u(end:-1:1,:);
qy = v(end:-1:1,:);
figure(2), plot([0:1/(nx-1):1],qx(:,(nx/2)+0.5),[0:1/(nx-1):1],qy((nx/2)+0.5,:),[0:1/(nx-1):1],zeros(1,nx));
ylabel('Velocity');
xlabel('Length');
legend('Horizontal Centre line Velocity','Vertical Centre line Velocity');
title(strcat('Centre Line Velocities at Re =', num2str(Re), ' and c = ', num2str(c)));
h2 = figure(2);
figure(3), contourf(D, 50, 'LineStyle', 'none');
title('Contour of Continuity');colorbar;
h3 = figure(3);
h4 = figure(4);
figure(4), contourf(vel(end:-1:1,:), 50, 'LineStyle', 'none');
colorbar;
title(strcat('Absolute velocity at Re =', num2str(Re), ' and c = ', num2str(c)));
h5 = figure(5);
figure(5), contourf(qx, 50, 'LineStyle', 'none');colorbar;
title(strcat('Horizontal Velocity at Re =', num2str(Re), ' and c = ', num2str(c)));```

86
h6 = figure(6);

figure(6), contourf(qy, 50, 'LineStyle', 'none');colorbar;
title(strcat('Vertical Velocity at Re =', num2str(Re), ' and c = ',
    num2str(c)));

figure(7), plot([0:1/(nx-1):1],qx(:,(nx/2)+0.5));
title(strcat('X-Velocity-Centre-Line at Re =', num2str(Re), ' and c = ',
    num2str(c)));

h7 = figure(7);

figure(8), plot([0:1/(nx-1):1],qy((nx/2)+0.5,:));
title(strcat('Y-Velocity-Centre-Line at Re =', num2str(Re), ' and c = ',
    num2str(c)));

h8 = figure(8);

figure(9), contourf(vor, 50, 'LineStyle', 'none');colorbar;
title(strcat('Vorticity Contour at Re =', num2str(Re), ' and c = ',
    num2str(c)));

h9 = figure(9);

for ff = 1:9
    s4 = strcat(s3(9:end), '\fig_c', num2str(c*10), '_', num2str(ff), '.
        jpg');
    s5 = strcat(s3(9:end), '\fig_c', num2str(c*10), '_', num2str(ff), '.
        fig');
    fs1 = strcat('saveas(h', num2str(ff), ', s4);');
    fs2 = strcat('saveas(h', num2str(ff), ', s5);');
    eval(fs1);
    eval(fs2);
end

fid1 = fopen(strcat(s3(9:end), '\Re', num2str(Re), '_c', num2str(c
    *100), '.dat'), 'w+');

fprintf(fid1, '\nThe maximum value of Continuity at any Node of the

87
Domain is = %f', max(max(D)));  
152 fprintf(fid1, '\nThe total value of Continuity residual throughout
the Domain is = %f', sum(sum(D)));  
153 fprintf(fid1, '\nNo of Iterations required for convergence for Re= %f
for C= %.1f is = %d', Re,c,k);  
154 fprintf(fid1, '\nThe maximum value of Horizontal Pressure gradient w.
r.t Time at any Node of the Domain is = %f', max(max(del_Px)));  
155 fprintf(fid1, '\nThe total value of Horizontal Pressure Gradient
residual w.r.t Time throughout the Domain is = %f', sum(sum(del_Px)));  
156 fprintf(fid1, '\nThe maximum value of Vertical Pressure gradient w.r.
time at any Node of the Domain is = %f', max(max(del_Py)));  
157 fprintf(fid1, '\nThe total value of Vertical Pressure Gradient w.r.t
Time residual throughout the Domain is = %f', sum(sum(del_Py)));  
158 fprintf(fid1, '\nGrid Size for convergence at Re=%d, and C=%0.1f is =
%d X %d', Re,c,nx,nx);  
159 fprintf(fid1, '\nDel_t for convergence at Re=%d, and C=%0.1f is = %f'
, Re,c,del_t);  
160 fprintf(fid1, '\n Value of Stability criterion parameters Alpha and
Beta for this simulation run is %.3f & %.3f respectively',alpha,beta);  
161 fclose(fid1);  
162 dlmwrite(strcat(s3(9:end), '\Re', num2str(Re), '_c', num2str(c*100),
'u.dat'), u, ',');  
163 dlmwrite(strcat(s3(9:end), '\Re', num2str(Re), '_c', num2str(c*100),
'v.dat'), v, ',');  
164 dlmwrite(strcat(s3(9:end), '\Re', num2str(Re), '_c', num2str(c*100),
'vor.dat'), vor, ',');  
165 dlmwrite(strcat(s3(9:end), '\Re', num2str(Re), '_c', num2str(c*100),
'vort.dat'), vor, ',');
A.2 Matlab script for Simulating Flow inside a rectangular duct

```matlab
A.2 Matlab script for Simulating Flow inside a rectangular duct

1 clear all; clc ; close all;
2 Flow in a Duct Using Pressure Differential Equation
3 s1 = '! mkdir Flow_in_Rectangular_Duct';
4 nx=101; ny=101;
5 len_x=1; len_y= 1;
6 hx= len_x/(ny-1);
7 hy= len_y/(nx-1);
8 del_t= 1e-3 ;
9 Uwall= 0;
```

```matlab
166 dlmwrite(strcat(s3(9:end), '\Re', num2str(Re), '_c', num2str(c*100), '_del_Px.dat'), del_Px, ',');
167 dlmwrite(strcat(s3(9:end), '\Re', num2str(Re), '_c', num2str(c*100), '_Py.dat'), del_Py, ',');
168 dlmwrite(strcat(s3(9:end), '\Re', num2str(Re), '_c', num2str(c*100), '_Py.dat'), del_Py, ',');
169 dlmwrite(strcat(s3(9:end), '\Re', num2str(Re), '_c', num2str(c*100), '_D.dat'), del_Py, ',');
170 close all;
171 end
172 end
173 else
174 fprintf(' The convergence cannot be obtained for this grid and del_t value')
175 end
176 toc
```
10    f=1;
11    k=0;
12    Re=500;
13    c=1;
14    beta=0.25; alphax=0.1; alphay=0.1
15    betax= del_t/(2*hx);
16    betay=del_t/(2*hy);
17    alphax= del_t/(Re*hx^2);
18    alphay= del_t/(Re*hy^2);
19    u= zeros(nx,ny);
20    u(:,1)=0; u(1,:)=0; u(:,end)=0; u(end,:)=0;
21    x= 0:hy:len_y;
22    u(:,1)=4*x.*(1-x);
23    u(:,1)=1;
24    u(2:end-1,1)=ny/(ny-1);
25    v= zeros(nx,ny);
26    D= zeros(nx,ny);
27    Px= zeros(nx,ny); Py = zeros(nx,ny);
28    Px(end,:)=-1e5;
29    Px(2:end-1,end) = -5;
30    u(1,:) = 1;
31    D(:,end)=0; D(:,1)=0;
32    D(1,:)= 0; D(end,:)=0;
33    Px_star = Px; Py_star = Py;
34    del_Px= zeros(nx,ny); del_Py= zeros(nx,ny);
35    del_Px= ones(nx,ny); del_Py= ones(nx,ny);
36    del_Px(1,:) = 0; del_Py(1,:) = 0;
37    u_star = u; v_star = v;
38    k=0;
while (k<1e5 || (max(max(D))>1e-5) || min(min(D))<-1e-4 || max(max(del_Px)>1e-1)) || max(max(del_Py)>1e-1)) || min(min(del_Px))<-1e-1 || min(min(del_Py))<-1e-1))

k=k+1;

u_star(nx-1:-1:2,2:ny-1)= u(nx-1:-1:2,2:ny-1) + alphax*(u(nx-1:-1:2,3:ny)+u(nx-1:-1:2,1:ny-2)-2*u(nx-1:-1:2,2:ny-1)) + alphay*(u(nx:-1:3,2:ny-1)+u(nx-2:-1:1,2:ny-1)-2*u(nx-1:-1:2,2:ny-1))... -del_t*P_x(nx-1:-1:2,2:ny-1)/1 -u(nx-1:-1:2,2:ny-1)*betax.*(u(nx-1:-1:2,3:ny)-u(nx-1:-1:2,1:ny-2))- v(nx-1:-1:2,2:ny-1)*betay.*(u(nx-2:-1:1,2:ny-1)-u(nx:-1:3,2:ny-1));

v_star(nx-1:-1:2,2:ny-1)= v(nx-1:-1:2,2:ny-1) + alphax*(v(nx-1:-1:2,3:ny)+v(nx-1:-1:2,1:ny-2)-2*v(nx-1:-1:2,2:ny-1)) + alphay*(v(nx:-1:3,2:ny-1)+v(nx-2:-1:1,2:ny-1)-2*v(nx-1:-1:2,2:ny-1))... -del_t*P_y(nx-1:-1:2,2:ny-1)/1 -u(nx-1:-1:2,2:ny-1)*betax.*(v(nx-1:-1:2,3:ny)-v(nx-1:-1:2,1:ny-2))- v(nx-1:-1:2,2:ny-1)*betay.*(v(nx-2:-1:1,2:ny-1)-v(nx:-1:3,2:ny-1));

v_star(:,end-1)= v_star(:,end);
u_star(:,end)= u_star(:,end-1);
v_star(:,end)= v_star(:,end-1);

Boundary condition impose on Wall for fully developed flow

u_star(:,end)=(4*u_star(:,end-1)-u_star(:,end-2))/3; % Boundary condition specifying ux =0 at outlet
v_star(:,end)=(4*v_star(:,end-1)-v_star(:,end-2))/3; % Boundary condition specifying ux =0 at outlet

u= u_star;
v= v_star;

Calculation of continuity Matrix
D(nx-1:-1:2,2:ny-1)=(u(nx-1:-1:2,3:ny)-u(nx-1:-1:2,1:ny-2))/(2*hx 
)+(v(nx-2:-1:1,2:ny-1)-v(nx:-1:3,2:ny-1))/(2*hy);

Calculation of Pressure Gradient Matrix
Px_star(nx-1:-1:2,2:ny-1)=Px(nx-1:-1:2,2:ny-1)-c*betax*(D(nx 
-1:-1:2,3:ny)-D(nx-1:-1:2,1:ny-2));
Py_star(nx-1:-1:2,2:ny-1)=Py(nx-1:-1:2,2:ny-1)-c*betay*(D(nx 
-2:-1:1,2:ny-1)-D(nx:-1:3,2:ny-1));

Px_star(:,end)=Px_star(:,end-1);
Py_star(end,:)=Py_star(end-1,:);

Px_star(end-1,:)=Px_star(end,:);
Calculation of Dpx/Dt & Dpy/Dt
del_Px= (Px_star- Px)/ del_t;
del_Py= (Py_star- Py)/ del_t;
Px=f*Px_star+ (1-f)*Px;
Py= f*Py_star+ (1-f)*Py;
Px(:,end)=Px(:,end-1);
Px(:,1)=Px(:,2);

if max(max(D))>abs(min(min(D)))
    error=max(max(D));
    fprintf(' The value of error in Continuity is %f\n',max(max(D )))
else
error=abs(min(min(D)));  
fprintf(' The value of error in Continuity is %f \n', abs(min(min(D))));
end
error=max(max(D));
fprintf(' The value of error in Continuity is %f \n', max(max(D)));
xx(k)=k;
DD(k)=error;
PX(k)=max(max(del_Px));
PY(k)=max(max(del_Py));
mdif(k)=(sum(u(:,1))-sum(u(:,end)))*hy;
figure(1), plot(xx,log10(DD),xx,log10(PX),xx,log10(PY),xx,mdif);
end
figure(1), plot(xx,log10(DD),xx,log10(PX),xx,log10(PY),xx,mdif);
legend('Continuity','Horizontal-Pressure-Gradient','Vertical-Pressure-Gradient','del_mdot');
fprintf(' The value of error in Continuity is %f \n', max(max(D)));
title(strcat('Covergence at Re = ', num2str(Re), ' and c = ', num2str(c)));
ylabel('Logarithmic Scale');
xlabel('Iteration Count');

if (alpha<0.5 && beta<0.5)  
  for Re=100:100:1000
    beta=0.0128; alpha=0.002;
    h=(2*beta)/(alpha*Re);
\begin{verbatim}
104    del_t= beta*2*h;
105    nx=1/(h)+1;
106    nx=floor(nx);
107    if rem(nx,2)== 0
108        nx= nx+1;
109    end
110    if nx<51
111        nx=51;
112    end
113    ny=nx;
114    h= 1/(nx-1);
115    del_t= (Re*h*h)*alpha;  beta= del_t/(2*h);
116    s2 = strcat(s1, '\Re', num2str(Re));
117    for c= 1:0.2:0.5/beta
118        qx=zeros(nx,nx);
119        qy=zeros(nx,nx);
120        x=[0:1/(nx-1):1];
121        u= zeros(nx,ny);
122        v= zeros(nx,ny);
123        D= zeros(nx,ny);  Px= zeros(nx,ny); Py= zeros(nx,ny);
124        Px_star= Px; Py_star= Py;
125        del_Px= ones(nx,nx); del_Py= ones(nx,ny);
126        del_Px(1,:)=0; del_Py(1,:)=0;
127        u(1,:)= Uwall;
128        u_star= u; v_star= v;
129        s3 = strcat(s2, '\c', num2str(c*100));
130        k = 0;
131        clear DD PX PY xx
132        h1=figure(1);
\end{verbatim}
%   hold on;
while((max(max(D)>1e-2)) || max(max(del_Px>1e-1)) || max(max(del_Py>1e-1)) || min(min(del_Px)<-1e-1) || min(min(del_Py)<-1e-1))
k=k+1;
u_star(nx-1:-1:2,2:ny-1)= u(nx-1:-1:2,2:ny-1) + alpha*(u(nx-1:-1:2,3:ny)+u(nx-1:-1:2,1:ny-2)+u(nx:-1:3,2:ny-1)+u(nx-2:-1:1,2:ny-1)-4*u(nx-1:-1:2,2:ny-1))
-v(nx-1:-1:2,2:ny-1)*beta*(u(nx-2:-1:1,2:ny-1)-u(nx:-1:3,2:ny-1));
v_star(nx-1:-1:2,2:ny-1)= v(nx-1:-1:2,2:ny-1) + alpha*(v(nx-1:-1:2,3:ny)+v(nx-1:-1:2,1:ny-2)+v(nx:-1:3,2:ny-1)+v(nx-2:-1:1,2:ny-1)-4*v(nx-1:-1:2,2:ny-1))
-v(nx-1:-1:2,2:ny-1)*beta*(v(nx-2:-1:1,2:ny-1)-v(nx:-1:3,2:ny-1));
temp_u=u; temp_v=v;

%....Post processing .........%
error_u=(tempu((nx-1),(nx-1)) - u_star((nx-1),(nx-1)))^2;
error_v=(tempv((nx-1),(nx-1)) - v_star((nx-1),(nx-1)))^2;
u= u_star;
v = v_star;

% Calculation of continuity Matrix
D(nx-1:-1:2,2:ny-1) = (u(nx-1:-1:2,3:ny)-u(nx-1:-1:2,1:ny-2)+ v(nx-2:-1:1,2:ny-1)-v(nx-1:3,2:ny-1))/(2*h)
;

% Calculation of Pressure Gradient Matrix
Px_star (nx-1:-1:2,2:ny-1) = Px(nx-1:-1:2,2:ny-1)-c*beta*(D(nx-1:-1:2,3:ny)-D(nx-1:-1:2,1:ny-2));
Py_star (nx-1:-1:2,2:ny-1) = Py(nx-1:-1:2,2:ny-1)-c*beta*(D(nx-2:-1:1,2:ny-1)-D(nx-1:3,2:ny-1));

% Calculation of Dpx/Dt & Dpy/Dt
del_Px = (Px_star - Px)/del_t;
del_Py = (Py_star - Py)/del_t;
Px = f*Px_star + (1-f)*Px;
Py = Py_star + (1-f)*Py;

%....Post processing...........

xx(k)=k;
uerr(k)= sqrt(error_u);
verr(k)= sqrt(error_v);
DD(k) = max(max(D));
PX(k) = max(max(del_Px));
PY(k) = max(max(del_Py));
figure(1), plot(xx,log10(D),xx,log10(PX),xx,log10(PY));
fprintf(' The value of error in Continuity is %f\n', DD(k))
end

legend('Continuity','Horizontal-Pressure-Gradient','Vertical-Pressure-Gradient')
title(strcat('Covergence at Re = ', num2str(Re), ' and c = ', num2str(c)));
ylabel('Logarithmic Scale');
xlabel('Iteration Count');
vel= zeros(nx,ny);
vel(1,:)=Uwall;
for i = nx-1:-1:2
    for j= 2:nx-1
        vel(i,j)= sqrt(u(i,j)*u(i,j)+v(i,j)*v(i,j));
    end
end
qx= u(end:-1:1,:);
qy= v(end:-1:1,:);
figure(2), plot([0:1/(nx-1):1],qx(:,(nx/2)+0.5),[0:1/(nx -1):1],qy((nx/2)+0.5,:),[0:1/(nx-1):1],zeros(1,nx));
ylabel('Velocity');
xlabel('Length');
legend('Horizontal Centre line Velocity','Vertical Centre line Velocity');
title(strcat('Centre Line Velocities at Re = ', num2str(Re ), ' and c = ', num2str(c)));
h2 = figure(2);
figure(3), contourf(D, 50, 'LineStyle', 'none');
title('Contour of Continuity');
h3 = figure(3);
h4 = figure(4);
```matlab
figure(4), contourf(vel(end:-1:1,:), 50, 'LineStyle', 'none');
title(strcat('Absolute velocity at Re = ', num2str(Re), ' and c = ', num2str(c)));
h5 = figure(5);
figure(5), contourf(qx, 50, 'LineStyle', 'none');
title(strcat('Horizontal Velocity at Re = ', num2str(Re), ' and c = ', num2str(c)));
h6 = figure(6);
figure(6), contourf(qy, 50, 'LineStyle', 'none');
title(strcat('Vertical Velocity at Re = ', num2str(Re), ' and c = ', num2str(c)));
figure(7), plot([0:1/(nx-1):1],qx(:,(nx/2)+0.5));
title(strcat('X-Velocity-Centre-Line at Re = ', num2str(Re), ' and c = ', num2str(c)));
h7 = figure(7);
figure(8), plot([0:1/(nx-1):1],qy((nx/2)+0.5,:));
title(strcat('Y-Velocity-Centre-Line at Re = ', num2str(Re), ' and c = ', num2str(c)));
h8 = figure(8);
eval(s3);
for ff = 1:8
    s4 = strcat(s3(9:end), '\fig_c', num2str(c*10), '\_', num2str(ff), '.jpg');
    s5 = strcat(s3(9:end), '\fig_c', num2str(c*10), '\_', num2str(ff), '.fig');
    fs1 = strcat('saveas(h', num2str(ff), ', s4);');
    fs2 = strcat('saveas(h', num2str(ff), ', s5);');
eval(fs1);
```

98
eval(fs2);

end

fid1 = fopen(strcat(s3(9:end), '\
Re', num2str(Re), '_c',
num2str(c*100), '.dat'), 'w+');

fprintf(fid1, '\nThe maximum value of Continuity at any
Node of the Domain is = %f', max(max(D)));

fprintf(fid1, '\nThe total value of Continuity residual
throughout the Domain is = %f', sum(sum(D)));

fprintf(fid1, '\nNo of Iterations required for
convergence for Re= %f, for C= %f is = %f',Re,c,k);

fprintf(fid1, '\nThe maximum value of Horizontal Pressure
gradient w.r.t Time at any Node of the Domain is = %f'
', max(max(del_Px)));

fprintf(fid1, '\nThe total value of Horizontal Pressure
Gradient residual w.r.t Time throughout the Domain is = %f', sum(sum(del_Px)));

fprintf(fid1, '\nThe maximum value of Vertical Pressure
gradient w.r.t Time at any Node of the Domain is = %f'
', max(max(del_Py)));

fprintf(fid1, '\nThe total value of Vertical Pressure
Gradient w.r.t Time residual throughout the Domain is
= %f', sum(sum(del_Py)));

fprintf(fid1, '\nGrid Size for convergence ar Re=\%f, and
C=\%f is = \%f X \%f', Re,c,nx,nx);

close(fid1);
end

end
else
fprintf(' The convergence cannot be obtained for this grid and
del_t value')
end

A.3 Matlab script for Simulating flow in backward facing step

1
2  % BACKWARD FACING STEP %%%%%%%
3  % Boundary condition
4  clear all
5  close all
6
7  nx=801; ny=81; mu=5*1e-3;
8  len_x=10; len_y=1;
9  hx=len_x/(nx-1);
10  hy=len_y/(ny-1);
11  u=zeros(ny,nx);
12  v=zeros(ny,nx);
13  y_top= -0:hy:0.5;
14  y_total= -0.5:hy:0.5;
15  x=0:hx:10;
16  u(1:(ny+1)/2,1)= 24*y_top.*(0.5-y_top); % Step Velocity Profile
   Inflow boundary condition
17  u(:,end)= 3*(0.5-y_total).*(y_total+0.5); % Outlet velocity profile
   Outflow boundary condition
18  D= zeros(ny,nx); % Initial Guess of Dilation
19  Px= zeros(ny,nx); Py= zeros(ny,nx); % Initial guess of Pressure
   gradient terms
20  Px_star= Px; Py_star=Py;
21 del_Px=zeros(ny,nx);
22 del_Py= zeros(ny,nx);
23 del_t=5e-3;
24 betax= del_t/(2*hx);
25 betay= del_t/(2*hy);
26 alphax= del_t*mu/(hx^2);
27 alphpay= del_t*mu/(hy^2);
28 u_star= u; v_star= v;
29 % Relaxation coeffiecient and pressure speed constat
30 f=1; c=1;
31 k=1;
32 while(k<1000|| max(max(D))>1e-14)
33 % for j=ny-1:-1:2
34 % for i= 2:nx-1
35 % u_star(j,i)=u(j,i)-betax*(u(j,i+1)-u(j,i-1))*u(j,i)-
36 % betay*(u(j-1,i)-u(j+1,i))*v(j,i)...
37 % -del_t*Px(j,i) + alphax*(u(j+1,i)+u(j-1,i)+u(j,i+1)
38 % +u(j,i-1)-4*u(j,i));
39 %
40 %
41 % end
42 % end
43 u_star(ny-1:-1:2,2:nx-1)=u(ny-1:-1:2,2:nx-1)-betax*(u(ny-1:-1:2,3:nx)
44 % -u(ny-1:-1:2,1:nx-2)).*u(ny-1:-1:2,2:nx-1)...
45 % -betay*(-u(ny-1:3,2:nx-1)+u(ny-2:-1:1,2:nx
46 % -1)).*v(ny-1:-1:2,2:nx-1)...

101
\[ -\frac{\Delta t}{P} x(ny-1:1:2,2:nx-1) + \alpha x(u(ny-1:1:3,2:nx-1) + u(ny-2:1:1,2:nx-1) - u(ny-1:1:2,3:nx) + u(ny-1:1:2,1:nx-2) - 4u(ny-1:1:2,2:nx-1)); \]

\[ v_{\text{star}}(ny-1:1:2,2:nx-1) = v(ny-1:1:2,2:nx-1) - \beta x(v(ny-1:1:2,3:nx) - v(ny-1:1:2,1:nx-2)).u(ny-1:1:2,2:nx-1) \]

\[ -\frac{\beta y}{P} y(ny-1:1:2,2:nx-1) + \alpha x(v(ny-1:1:3,2:nx-1) + v(ny-2:1:1,2:nx-1) + v(ny-1:1:2,3:nx) + v(ny-1:1:2,1:nx-2) - 4v(ny-1:1:2,2:nx-1)); \]

\% for \( j = \text{ny-1:1:2} \)
\% for \( i = 2:nx-1 \)
\% \begin{align*}
D(j,i) &= \frac{(u(j,i+1)-u(j,i-1))/(2hx) + (u(j+1,i)-v(j-1,i))}{2\times hy};
\end{align*}
\% end
% end

D(ny-1:1:2,2:nx-1)= (u(ny-1:1:2,3:nx)-u(ny-1:1:2,1:nx-2))/(2*hx)+
(-v(ny:-1:3,2:nx-1)+v(ny-2:-1:1,2:nx-1))/(2*hy);

% Calculation of Pressure Gradient Matrix
Px_star(ny-1:1:2,2:nx-1)= Px(ny-1:1:2,2:nx-1)-betax*(D(ny
-1:-1:2,3:nx)-D(ny-1:-1:2,1:nx-2));
Py_star(ny-1:1:2,2:nx-1)= Py(ny-1:1:2,2:nx-1)-betay*(-D(ny
-1:1:3,2:nx-1)+D(ny-2:-1:1,2:nx-1));

% Calculation of Dpx/Dt & Dpy/Dt
del_Px= (Px_star- Px)/ del_t;
del_Py= (Py_star- Py)/ del_t;
Px=f*Px_star+ (1-f)*Px;
Py= f*Py_star+ (1-f)*Py;
k=k+1;

%....Post processing............% 
xx(k)=k;
DD(k) = max(max(D));
PX(k)= max(max(del_Px));
PY(k)= max(max(del_Py));

fprintf(' The value of error in Continuity is %f
', max(max(D)))
end
A.4 Matlab script for Simulating flow in 3D backward facing step

1 ˚ Boundary condition
2 clear all
3 close all
4
5 nx=201;ny=21;nz=21;mu=5*1e-3;
6 len_x=10:len_y=1:len_z=1;
7 hx=len_x/(nx-1);
8 hy=len_y/(ny-1);
9 hz=len_z/(nz-1);
10 u=zeros(ny,nz,nx);
11 v=zeros(ny,nz,nx);
12 w=zeros(ny,nz,nx);
13 y_top= -0:hy:0.5;
14 qq=size(y_top,2);
15 y_top=y_top'*ones(1,qq);
16 y_total= -0.5:hy:0.5;
17 clear qq
qq=size(y_total,2);
y_total=y_total'*ones(1,qq);
x=0:hx:10;
u(1:(ny+1)/2,1:(nz+1)/2,1)= 24*y_top.*(0.5-y_top); % Step Velocity Profile Inflow boundary condition
u(:,:,end)= 3.*(0.5-y_total).*y_total+0.5; % Outlet velocity profile Outflow boundary condition
D= zeros(ny,nz,nx); % Initial Guess of Dilation
Px= zeros(ny,nz,nx); Py= zeros(ny,nz,nx); Pz= zeros(ny,nz,nx); % Initial guess of Pressure gradient terms
Px_star= Px; Py_star=Py; Pz_star=Pz;
del Px=zeros(ny,nz,nx);
del Py= zeros(ny,nz,nx);
del Pz=zeros(ny,nz,nx);
del_t=5e-3;
betax= del_t/(2*hx);
betay= del_t/(2*hy);
betaz= del_t/(2*hz);
alphax= del_t*mu/(hx^2);
alphay= del_t*mu/(hy^2);
alphaz= del_t*mu/(hz^2);
u_star= u; v_star=v; w_star=w;
% Relaxation coefficient and pressure speed constant
f=1; c=1;
k=1;
while (k<1000|| max(max(max(D)))>1e-14)
  % for j=ny-1:-1:2
  % for i= 2:nx-1
  u_star(j,i)=u(j,i)-betax*(u(j,i+1)-u(j,i-1))*u(j,i)-
betay*(u(j-1,i)-u(j+1,i))*v(j,i)...
\% -del_t*Pz(j,i) + alphax*(u(j+1,i)+u(j-1,i)+u(j,i+1)
+u(j,i-1)-4*u(j,i));
\%
\% v_star(j,i)=v(j,i)-betax*(v(j,i+1)-v(j,i-1))*u(j,i)-
betay*(v(j-1,i)-u(j+1,i))*v(j,i)...
\% -del_t*Py(j,i) + alphax*(v(j+1,i)+v(j-1,i)+v(j,i+1)
+v(j,i-1)-4*v(j,i));
\%
end
\%

u_star(ny-1:-1:2,nz-1:-1:2,2:nx-1)=u(ny-1:-1:2,nz-1:-1:2,2:nx-1)-
betax*(u(ny-1:-1:2,nz-1:-1:2,3:nx)-u(ny-1:-1:2,nz-1:-1:2,1:nx-2))
.*u(ny-1:-1:2,nz-1:-1:2,2:nx-1)...

-betay*(-u(ny-1:-1:3,nz-1:-1:2,2:nx-1)+u(ny
-2:-1:1,nz-1:-1:2,2:nx-1)).*v(ny-1:-1:2,
    nz-1:-1:2,2:nx-1)...

-betaz*(-u(ny-1:-1:2,nz:-1:3,3:nx)+u(ny
-1:-1:2,nz-2:-1:1,1:nx-2)).*w(ny-1:-1:2,
    nz-1:-1:2,2:nx-1)...

-del_t*Pz(ny-1:-1:2,nz-1:-1:2,2:nx-1) +
alphax*(u(ny-1:-1:3,nz-1:-1:2,2:nx-1)+u(ny
-2:-1:1,nz-1:-1:2,2:nx-1)+u(ny-1:-1:2,nz
-1:-1:2,3:nx)+u(ny-1:-1:2,nz-1:-1:2,1:nx
-2)...
+u(ny-1:-1:2,nz:-1:3,3:nx)+u(ny-1:-1:2,nz
-2:-1:1,1:nx-2)-6*u(ny-1:-1:2,nz
-1:-1:2,2:nx-1));


\[ v_{\text{star}}(ny-1:-1:2,nz-1:-1:2,2:nx-1) = v(ny-1:-1:2,nz-1:-1:2,2:nx-1) - \beta_{\text{ax}}(v(ny-1:-1:2,nz-1:-1:2,3:nx)-v(ny-1:-1:2,nz-1:-1:2,1:nx-2)) \times u(ny-1:-1:2,nz-1:-1:2,2:nx-1) \]

\[ \beta_{\text{ay}}(-v(ny-1:-1:3,nz-1:-1:2,2:nx-1)+v(ny-2:-1:1,nz-1:-1:2,2:nx-1)) \times v(ny-1:-1:2,nz-1:-1:2,2:nx-1) \]

\[ \beta_{\text{az}}(-v(ny-1:-1:2,nz-1:-1:3,3:nx)+v(ny-1:-1:2,nz-2:-1:1,1:nx-2)) \times w(ny-1:-1:2,nz-1:-1:2,2:nx-1) \]

\[ \text{del}_t \times Py(ny-1:-1:2,nz-1:-1:2,2:nx-1) + \alpha_{\text{ax}}(v(ny-1:-1:3,nz-1:-1:2,2:nx-1)+v(ny-2:-1:1,nz-1:-1:2,2:nx-1)+v(ny-1:-1:2,nz-1:-1:2,3:nx)+v(ny-1:-1:2,nz-1:-1:2,1:nx-2)) \times v(ny-1:-1:2,nz-2:-1:1,1:nx-2) - 6 \times v(ny-1:-1:2,nz-1:-1:2,2:nx-1) \]

\[ w_{\text{star}}(ny-1:-1:2,nz-1:-1:2,2:nx-1) = w(ny-1:-1:2,nz-1:-1:2,2:nx-1) - \beta_{\text{ax}}(w(ny-1:-1:2,nz-1:-1:2,3:nx)-w(ny-1:-1:2,nz-1:-1:2,1:nx-2)) \times u(ny-1:-1:2,nz-1:-1:2,2:nx-1) \]

\[ \beta_{\text{ay}}(-w(ny-1:-1:3,nz-1:-1:2,2:nx-1)+w(ny-2:-1:1,nz-1:-1:2,2:nx-1)) \times v(ny-1:-1:2,nz-1:-1:2,2:nx-1) \]

\[ \beta_{\text{az}}(-w(ny-1:-1:2,nz-1:-1:3,3:nx)+w(ny-1:-1:2,nz-2:-1:1,1:nx-2)) \times w(ny-1:-1:2,nz-1:-1:2,2:nx-1) \]

\[ \text{del}_t \times Pz(ny-1:-1:2,nz-1:-1:2,2:nx-1) + \alpha_{\text{ax}}(w(ny-1:-1:3,nz-1:-1:2,2:nx-1)+w(ny-2:-1:1,1:nx-2)) \times w(ny-1:-1:2,nz-1:-1:2,2:nx-1) \]
-2:-1:1, nz-1:-1:2, 2:nx-1)+w(ny-1:-1:2, nz
-1:-1:2, 3:nx)+w(ny-1:-1:2, nz-1:-1:2, 1:nx
-2)... +w(ny-1:-1:2, nz:-1:3, 3:nx)+w(ny-1:-1:2, nz
-2:-1:1, 1:nx-2)-6*w(ny-1:-1:2, nz
-1:-1:2, 2:nx-1));

70 \% v_star(ny-1:-1:2, 2:nx-1)=v(ny-1:-1:2, 2:nx-1)-betax*(v(ny
-1:-1:2, 3:nx)-v(ny-1:-1:2, 1:nx-2)).*u(ny-1:-1:2, 2:nx-1)...  
71 \% -betay*(-v(ny:-1:3, 2:nx-1)+v(ny-2:-1:1, 2:nx-1))
.*v(ny-1:-1:2, 2:nx-1)...  
72 \% -del_t*Py(ny-1:-1:2, 2:nx-1) + alphax*(v(ny
:-1:3, 2:nx-1)+v(ny-2:-1:1, 2:nx-1)+v(ny-1:-1:2, 1:
x-2)-4*v(ny-1:-1:2, 2:nx-1));

74 \% u_star(:, end)=u_star(:, end-2); \% Boundary condition specifying ux
    =0 at outlet
75 \% v_star(end-2,:)=v_star(end,:); \% Boundary condition specifying vy
    =0 at outlet
76 \% ...Post processing...........
77 u = u_star;
78 v = v_star;
79 w = w_star;
80 \% Calculation of continuity Matrix
81
82
83 \% for j=ny-1:-1:2
84 \% for i= 2:nx-1
\[
D(j,i) = \frac{(u(j,i+1) - u(j,i-1))/(2*hx) + (u(j+1,i) - v(j-1,i))/(2*hy)}{(2*hy)};
\]

\[
end
\]

\[
end
\]

\[
D(ny-1:1:2,nz-1:-1:2,2:nx-1)= \frac{(u(ny-1:1:2,nz-1:-1:2,3:nx)-u(ny
-1:-1:2,nz-1:-1:2,1:nx-2))/(2*hx) + \ldots
\]

\[
(-v(ny-1:1:3,nz-1:-1:2,2:nx-1)+v(ny
-2:-1:1,nz-1:-1:2,2:nx-1))/(2*hy)
\]

\[
+\ldots
\]

\[
(-w(ny-1:-1:2,nz-1:3,2:nx-1)+w(ny
-1:-1:2,nz-2:-1:1,2:nx-1))/(2*hz);
\]

%Calculation of Pressure Gradient Matrix

\[
Px_{\text{star}}(ny-1:1:2,nz-1:-1:2,2:nx-1) = Px(ny-1:-1:2,nz-1:-1:2,2:nx
-1)-betax*(D(ny-1:1:2,nz-1:-1:2,3:nx)-D(ny-1:-1:2,nz
-1:-1:2,1:nx-2));
\]

\[
Py_{\text{star}}(ny-1:-1:2,nz-1:-1:2,2:nx-1) = Py(ny-1:-1:2,nz-1:-1:2,2:nx
-1)-betay*(-D(ny-1:1:3,nz-1:-1:2,2:nx-1)+D(ny-2:-1:1,nz
-1:-1:2,2:nx-1));
\]

\[
Pz_{\text{star}}(ny-1:-1:2,nz-1:-1:2,2:nx-1) = Pz(ny-1:-1:2,nz-1:-1:2,2:nx
-1)-betaz*(-D(ny-1:-1:2,nz-1:3,2:nx-1)+D(ny-1:-1:2,nz
-2:-1:1,2:nx-1));
\]

%Calculation of Dpx/Dt & Dpy/Dt

\[
del_Px= (Px_{\text{star}}- Px)/ del_t;
\]

\[
del_Py= (Py_{\text{star}}- Py)/ del_t;
\]
A.5 Matlab script for Generating flow path and velocity triangle of centrifugal compressor

```matlab
103    del_Pz= (Pz_star- Pz)/ del_t;
104    Px=f*Px_star+ (1-f)*Px;
105    Py= f*Py_star+ (1-f)*Py;
106    Pz= f*Pz_star+ (1-f)*Pz;
107    k=k+1;
108    
109    \%...Post processing.......\%
110    xx(k)=k;
111    DD(k) = max(max(max(D)));
112    PX(k)= max(max(max(del_Px)));
113    PY(k)= max(max(max(del_Py)));
114    PZ(k)= max(max(max(del_Pz)));
115
116    fprintf(' The value of error in Continuity is %f\n', max(max(max(D))));
117
118    end
119
120    plot(xx,log10(DD),xx,log10(PX),xx,log10(PY),xx,log10(PZ));
121    legend('Dilation','dPz/dt','dPy/dt','dPy/dt')
122    fprintf(' The value of error in Continuity is %f\n', max(max(D)))
123    title(strcat('Covergence History at grid size ', num2str(ny), ' X ', num2str(nz), ' X ', num2str(nx )));
124    ylabel('Logarithmic Scale');
125    xlabel('Iteration Count');
126    axis square
```
All units are in SI system

clear all;
close all;

%HUB DEFINITION
x1= -5:0.5:0;
n1=size(x1,2);
y1=ones(1,n1)*34.5416;
x2= 0:0.1:120.5;
xh= [x1 x2];
y2= -sqrt(186^2-(x2+64.5).*(x2+64.5))+209;
yh=[y1 y2];

%plot(xh,yh);

%Casing Definition
xc1= 0:0.1:105;
yc1= 209-sqrt((1-(xc1/105).*(xc1/105))*86.5^2);
xc=[x1 xc1];
yc=[ones(1,n1)*122.5 yc1];

%%%%%%%%%%%%%%%%

%LEading Edge definition
xle=-5*ones(1,n1);
yle= yh(1,1):(yc(1,1)-yh(1,1))/(n1-1):yc(1,1);

plot(xh,yh,xc,yc,xle,yle,'LineWidth',2)
legend('Hub','Casing','Leading Edge')
30 title('Single Stage Flow Path')

33 \%Inlet and outlet conditions
34 N = 22345; \%Hub speed
35 Ns = 80; \%Specific speed
36 Zr1 = 24; \%Number of blades for 1st stage rotor
37 Zs1 = 24; \%Number of blades for 1st stage stator
38 Zr2 = 24; \%Number of blades for 2nd stage rotor
39 rs = 0.0444323; \%Shaft radius
40 m = 4; \%mass flow rate
41 P01 = 101325; \%Inlet Total Pressure
42 T01 = 288.15; \%Inlet Total Temperature
43 PR = 4; \%Design pressure ratio

45 \%Thermodynamic properties of air
46 g = 1.4; \%Gamma
47 R = 287; \%Universal gas constant of air
48 Cp = 1004.5; \%Pressure constant

50 \%assumptions
51 e = 0.88; \%Isentropic Overall Efficiency
52 e1 = 0.92; \%Isentropic 1st stage efficiency

54 \%Input parameters
55 Vt1 = 0; \%No prewhirl at inlet
56 \alpha1 = 0; \%Absolute flow angle

58 \% GeometricProperties for each section
%Inlet
r1h = 0.044542;  %Inlet radius
r1t = 0.1225;
z1h = 0;
z1t = 0;
b1 = z1h-z1t;  %blade width at inlet
A1 = pi*(r1t^2-r1h^2);
R1 = linspace(r1h,r1t,5);
%1st stage
r2h = 0.075;  %1st stage radius at hub
r2t = 0.13;
r2_1 = 0.5*(r2t+r2h);
z2h = 0.053228501;
z2t = 0.042766;
A2 = pi*(r2h+r2t)*sqrt((r2h-r2t)^2+(z2h-z2t)^2);
%2nd stage
r3h = 0.1375;
r3t = 0.16425;
r3_1 = 0.5*(r3t+r3h);
z3h = 0.10269375;
z3t = 0.0898568;
A3 = pi*(r3h+r3t)*sqrt((r3h-r3t)^2+(z3h-z3t)^2);
%Outlet
r4 = 0.2;  %Outlet radius at hub and tip
z4h = 0.120527;
z4t = 0.10443;
b4 = 0.0155;  %blade width at outlet
A4 = pi*r4*(z4h-z4t);
%Initial calculations for thermodynamic properties

w = 2*pi*N/60;               %Angular velocity
TR = (PR^((2/7)-1)/e+1;       %Overall Temperature Ratio
T04 = T01*TR;                %Outlet Temperature
dH = Cp*(T04-T01);           %Overall work done
dH1 = 0.3*dH;                %Work done in 1st stage
T02 = T01+dH1/Cp;            %Total Temperature at R1 outlet
TR1 = T02/T01;
PR1 = (s1*(TR1-1)+1)^((7/2));
PR2 = PR/PR1;

%Thermodynamic Properties
P02 = PR1*P01;
P04 = PR*P01;                %Outlet Total Pressure
rho01 = P01/(R*T01);
h01 = Cp*T01;
h02 = Cp*T02;
h04 = Cp*T04;
s1 = 1650.55;
h03 = h02;
T03 = T02;

% Calculating properties at R1 inlet
U1 = w*R1;
rho1 = rho01;
error = 5;
while abs(error)>0.001
    Vz1 = m/(rho1*A1);
    V1 = Vz1;
T1 = T01-V1.^2/(2*Cp);
P1 = P01/(T01/T1)^-(3.5);
rho1p = P1/(R*T1);
error = 100*(1-rho1p/rho1);
rho1 = rho1p;
end
Vz1 = m/(rho1*A1);
V1 = Vz1;
T1 = T01-V1.^2/(2*Cp);
P1 = P01*(T1/T01)^-(3.5);
W1 = -U1;
beta1 = atand(W1/Vz1);
W1 = Wt1./sind(beta1);
M1 = V1/(sqrt(g*R*T1));
Wz1 = Vz1;
M1rel = W1/(sqrt(g*R*T1));

%Calculating properties at R1 outlet
U2 = w*r2_1;
Vt2 = (h02-h01+U1*Vt1)./U2;
rho02 = P02/(R*T02);
rho2 = rho02;
error = 5;
while abs(error)>0.001
    Vz2 = m/(rho2*A2);
    V2 = sqrt(Vz2.^2+Vt2.^2);
    T2 = T02-V2.^2/(2*Cp);
    P2 = P02./(T02./T2).^(3.5);
    rho2p = P2/(R*T2);
 error = 100*(1-rho2p/rho2);
 rho2 = rho2p;
 end
 Vz2 = m/(rho2*A2);
 V2 = sqrt(Vz2.^2+Vt2.^2);
 T2 = T02-V2.^2/(2*Cp);
 P2 = P02.*(T2./T02).^3.5);
 Wt2 = Vt2-U2;
 beta2 = atand(Wt2./Vz2);
 W2 = Wt2./sind(beta2);
 M2 = V2/(sqrt(g*R*T2));
 Wz2 = Vz2;
 alpha2 = atand(Vt2/Vz2);
 M2rel = W2/(sqrt(g*R*T2));

Calculating properties at R outlet
 U4 = w*r4;
 Vt4 = (h04-h01+U1*Vt1)./U4;
 Wt4 = Vt4-U4;
 beta4 = -30;
 Vz4 = Wt4./tand(beta4);
 V4 = sqrt(Vz4.^2+Vt4.^2);
 T4 = T04-V4.^2/(2*Cp);
 P4 = P04./(T04./T4).^3.5);
 rho4 = P4/(R*T4);
 W4 = Wt4./sind(beta4);
 alpha4 = atand(Vt4/Vz4);
 M4 = V4/(sqrt(g*R*T4));
 Wz4 = Vz4;
175  M4rel = W4/(sqrt(g*R*T4));

176

177  \%Calculating properties at R2 intet
178  U3 = w*r3_1;
179  lc = 0.03;
180  P03 = P02-lc*(P02-P2(1));
181  Vt3 = r2_1*Vt2/r3_1;
182  \hat{W}t3 = Vt3-U3;
183  \% alpha3 = alpha2-15;
184  \% Vz3 = Vt3./tand(alpha3);
185  \% beta3 = atand(Wt3./Vz3);
186  \% W3 = Wt3./sind(beta3);
187  \% V3 = sqrt(Vz3.^2+Vt3.^2);
188  \% T3 = T03-V3.^2./Cp;
189  \% M3 = V3./(sqrt(g*R*T3));
190  \% Wz3 = Vz3;
191  \% M3rel = W3/(sqrt(g*R*T3));
192  \% rho3 = m./(Vz3*A3);
193  \% P3 = P03.*((T3./T03).^3.5));
194
195  rho03 = P03/(R*T03);
196  rho3 = rho03;
197  \texttt{error} = 5;
198  \texttt{while abs(error)}\texttt{>0.001}
199    \texttt{Vz3 = m.}/(rho3*A3);
200    \texttt{V3 = sqrt(Vz3.^2+Vt3.^2)};
201    \texttt{T3 = T03-V3.^2/(2*Cp)};
202    \texttt{P3 = P03-0.5*rho3*V3(3).^2};
203    \texttt{rho3p = P3./(R*T3)};
error = 100*(1-rho3p/rho3);
rho3 = rho3p;
end
Vz3 = m. /(rho3*A3);
V3 = sqrt(Vz3.^2+Vt3.^2);
T3 = T03-V3.^2/(2*Cp);
P3 = P03-0.5*rho3*V3(3).^2;

beta3 = atand(Wt3./Vz3);
W3 = Wt3./sind(beta3);
M3 = V3/(sqrt(g*R*T3));
Wz3 = Vz3;

alpha3 = atand(Vt3/Vz3);
M3rel = W3/(sqrt(g*R*T3));

ywall1 = 6*((Vz1/0.0000157)^(-7/8))*((0.2)^(1/8));
ywall2 = 6*((Vz3(3)/0.0000157)^(-7/8))*((0.2)^(1/8));

%Velocity triangles at R1 inlet
figure();
quiver(0,0,V1,0);
hold on;
quiver(0,0,Wz1,Wt1(3));
hold on;
quiver(V1,0,0,-U1(3));
hold off;
title 'R1 inlet';

%Velocity triangles at R1 outlet
figure();
233  quiver(0,0,Vz2,0);
234  hold on;
235  quiver(0,0,Vz2,Wt2(3));
236  hold on;
237  quiver(0,0,Vz2,Vt2(3));
238  hold on;
239  quiver(Vz2,Vt2(3),0,-U2);
240  hold off;
241  title 'R1 outlet';
242
243
244  \%Velocity triangles at R2 inlet
245  figure();
246  quiver(0,0,Vz3(3),0);
247  hold on;
248  quiver(0,0,Vz3(3),Wt3(3));
249  hold on;
250  quiver(0,0,Vz3(3),Vt3(3));
251  hold on;
252  quiver(Vz3(3),Vt3(3),0,-U3);
253  hold off;
254  title 'R2 inlet';
255
256  \%Velocity triangles at R2 outlet
257  figure();
258  quiver(0,0,Vz4(3),0);
259  hold on;
260  quiver(0,0,Vz4(3),Wt4(3));
261  hold on;
262    \texttt{quiver(0,0,Vz4(3),Vt4(3));}
263  \texttt{hold \ on;}
264    \texttt{quiver(Vz4(3),Vt4(3),0,-U4);}  \texttt{hold\ off;}
265  \texttt{title 'R2\ outlet';}
266
267    \texttt{T = table(V1,Vz1,Vt1,W1(3),Wz1,Wt1(3),alpha1,beta1(3),U1(3),M1,M1rel}}
268    \texttt{(3));}
269  \texttt{T(:,:);}  
270    \texttt{filename = 'Streamlines.xlsx';}
271  \texttt{writetable(T,\texttt{filename,'Sheet'},2,'Range','B6')}  
272
273    \texttt{T = table(V2(3),Vz2,Vt2(3),W2(3),Wz2,Wt2(3),alpha2(3),beta2(3),U2,M2,}
274    \texttt{M2rel1);}  \texttt{T(:,:);}  
275    \texttt{filename = 'Streamlines.xlsx';}
276  \texttt{writetable(T,\texttt{filename,'Sheet'},2,'Range','B8')}  
277
278    \texttt{T = table(V3(3),Vz3(3),Vt3(3),W3(3),Wz3(3),Wt3(3),alpha3,beta3(3),U3,}
279    \texttt{M3,M3rel1);}  \texttt{T(:,:);}  
280    \texttt{filename = 'Streamlines.xlsx';}
281  \texttt{writetable(T,\texttt{filename,'Sheet'},2,'Range','B10')}  
282
284    \texttt{M4rel1);}  \texttt{T(:,:);}  
285    \texttt{filename = 'Streamlines.xlsx';}
286  \texttt{writetable(T,\texttt{filename,'Sheet'},2,'Range','B12')}  

120
A.6 Matlab script for Lid driven cavity flow in magnetic field

```matlab
clear all; clc; close all;

% Lid Driven Cavity Solution in magnetic field
s1 = '! mkdir Lid_Driven_Cavity_RE_100_Magnetic_Field';
nx=101;ny=101;
len_x = 1; len_y = 1;
h = len_x/(nx-1);
del_t = 5e-1;
Uwall = 1;
f = 1;
k = 0;

beta = 0.25; alpha = 0.1;
a = 1; b = 1;

if (alpha < 0.15 && beta < 0.5)
    for Re = 100:100:100
        beta = 0.3; alpha = 0.1;
h = (2*beta)/(alpha*Re);

        % del_t = beta*2*h;
        nx = 1/(h)+1;
        nx = floor(nx);
        if rem(nx,2)== 0
            nx = nx+1;
        end

    end
    if nx < 151
        nx = 151;
    end
    ny = nx;
```
h = 1/(nx-1);

del_t = (Re*h*h)*alpha; beta = del_t/(2*h);

s2 = strcat(s1, '\Re', num2str(Re));

for c = 1:0.1:1
    qx=zeros(nx,nx);
    qy=zeros(nx,nx);
    x=0:1/(nx-1):1;
    vor= zeros(nx,ny);
    u= 0.5*ones(nx,ny);
    u(1,:)=0;u(end,:)=0;u(:,end)=0;u(:,1)=0;
    v= zeros(nx,ny);
    D= zeros(nx,ny); Px= zeros(nx,ny); Py= zeros(nx,ny);
    Px_star= Px; Py_star= Py;
    del_Px= ones(nx,nx); del_Py= ones(nx,ny);
    del_Px(1,:)=0; del_Py(1,:)=0;
    u(1,:)= 1;
    u_star= u; v_star= v;
    s3 = strcat(s2, '\c', num2str(c*100));

k = 0;

clear DD PX PY xx

eval(s3);

% D(200,200)=0.5;

while (max(max(D)>1e-12) || k<1) || max(max(del_Px>1e-13)) ||
    max(max(del_Py>1e-3)) || min(min(del_Px)<-1e-3) || min(
    min(del_Py))<-1e-3)
    k=k+1;

% if rem(k, 100000) == 0

% figure(10), contourf(D, 50, 'LineStyle', 'none ');
colorbar;


```matlab
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
```
-1:-1:2,j)*beta.*(v(nx-1:-1:2,j+1)-v(nx-1:-1:2,j-1)) - v(nx-1:-1:2,j)*beta.*(v(nx-2:-1:1,j)-v(nx-1:-1:2,j));

end

j= 2:ny-1;

u_star(nx-1:-1:2,j) = u(nx-1:-1:2,j) + alpha*(u(nx-1:-1:2,j+1)+u(nx-1:-1:2,j-1)+u(nx-1:-1:2,j)) ;

-v(nx-1:-1:2,j)*beta.*(u(nx-1:-1:2,j+1)+u(nx-1:-1:2,j-1)+u(nx-1:-1:2,j)) ;

v_star(nx-1:-1:2,j) = v(nx-1:-1:2,j) + alpha*(v(nx-1:-1:2,j+1)+v(nx-1:-1:2,j-1)+v(nx-1:-1:2,j)) ;

-v(nx-1:-1:2,j)*beta.*(v(nx-1:-1:2,j+1)+v(nx-1:-1:2,j-1)+v(nx-1:-1:2,j));

%....Post processing.........%

u= u_star;

v= v_star;

% Calculation of continuity Matrix

D(nx-1:-1:2,2:ny-1) = (u(nx-1:-1:2,3:ny)-u(nx-1:-1:2,1:ny-2)+v(nx-2:-1:1,2:ny-1)-v(nx-1:-1:2,ny-1)-v(nx-1:-1:2,1:ny-2))
\[
\frac{(x_n - 1 - 1; 2; 2: ny - 1)}{(2*\h)};
\]

%Calculation of Pressure Gradient Matrix

\[
\begin{align*}
\text{Px\_star}(nx-1;1;2,2: ny-1) &= \text{Px}(nx-1;1;2,2: ny-1) - c* \\
&\quad \text{beta}*(D(nx-1;1;2,3: ny)-D(nx-1;1;2,1: ny-2)) \\
P_{\text{y\_star}}(nx-1;1;2,2: ny-1) &= \text{Py}(nx-1;1;2,2: ny-1) - c* \\
&\quad \text{beta}*(D(nx-2;1;1,2: ny-1)-D(nx-1;3,2: ny-1));
\end{align*}
\]

%Calculation of Dpx/Dt & Dpy/Dt

def\_px = (Px\_star - Px) / del\_t;

def\_py = (Py\_star - Py) / del\_t;

Px\_uf = f*Px\_star + (1-f)*Px;

Py\_uf = f*Py\_star + (1-f)*Py;

%....Post processing .........%

xx(k) = k;

DD(k) = \max(\max(D));

PX(k) = \max(\max(del\_px));

PY(k) = \max(\max(del\_py));

fprintf(' The value of error in Continuity is \%f', max(\max(D)))

end

figure(1), plot(xx, log10(DD), xx, log10(PX), xx, log10(PY));

legend('Continuity', 'Horizontal-Pressure-Gradient', 'Vertical-Pressure-Gradient')

fprintf(' The value of error in Continuity is \%f', max(}

125
max(D))

title(strcat('Convergence at Re = ', num2str(Re), ' and c = 
', num2str(c)));
ylabel('Logarithmic Scale');
xlabel('Iteration Count');

h1 = figure(1);
vel = zeros(nx, ny);
vel(1,:) = Uwall;

%% Calculation of Absolute Velocity

vel(nx-1:-1:2, 2:nx-1) = sqrt(u(nx-1:-1:2,2:nx-1).*u(nx
-1:-1:2,2:nx-1)+v(nx-1:-1:2,2:nx-1).*v(nx-1:-1:2,2:nx
-1));

vor(nx,:) = - (u(nx-1,1:nx)-u(nx,1:nx))/(2*h); % Bottom wall

vor(nx-1:-1:1,nx) = - (v(nx-1:-1:1,nx)-v(nx-1:-1:1,nx-1))
/(2*h); % Right wall

vor(1,nx-1:-1:1) = (u(1,nx-1:-1:1)-u(2,nx-1:-1:1))/(2*h);
% Top wall

vor(2:nx-1,1) = (v(2:nx-1,2)-u(2:nx-1,1))/(2*h); % Left wall
wall

%%% % FOR INTERNAL NODES

\begin{verbatim}
  vor(nx-1:-1:2,nx-1)= ((v(nx-2:-1:1,nx-1)-v(nx:-1:3,2:
    nx-1))- (u(nx-1:-1:2,3:ny)-u(nx-1:-1:2,1:nx-2)))/(2*h)
  
  qx= u(end:-1:1,:);
  qy= v(end:-1:1,:);
  figure(2), plot([0:1/(nx-1):1],qx(:,(nx/2)+0.5),[0:1/(nx
    -1):1],qy((nx/2)+0.5,:),[0:1/(nx-1):1],zeros(1,nx));
  ylabel('Velocity');
  xlabel('Length');
  legend('Horizontal Centre line Velocity','Vertical Centre
    line Velocity');
  title(strcat('Centre Line Velocities at Re = ', num2str(Re
    ), ' and c = ', num2str(c)));

  h2 = figure(2);
  figure(3), contourf(D, 50, 'LineStyle', 'none');
  title('Contour of Continuity');colorbar;

  h3 = figure(3);
  h4 = figure(4);
  figure(4), contourf(vel(end:-1:1,:), 50, 'LineStyle', 'none' );colorbar;
  title(strcat('Absolute velocity at Re = ', num2str(Re), ' 

\end{verbatim}

127
and c = ', num2str(c)));

h5 = figure(5);
figure(5), contourf(qx, 50, 'LineStyle', 'none');colorbar;
title(strcat('Horizontal Velocity at Re =', num2str(Re),
    ' and c = ', num2str(c)))

h6 = figure(6);
figure(6), contourf(qy, 50, 'LineStyle', 'none');colorbar;
title(strcat('Vertical Velocity at Re =', num2str(Re), ' and c = ', num2str(c)))

figure(7), plot([0:1/(nx-1):1],qx(:,(nx/2)+0.5));
title(strcat('X-Velocity-Centre-Line at Re =', num2str(Re), ' and c = ', num2str(c)))

h7 = figure(7);
figure(8), plot([0:1/(nx-1):1],qy((nx/2)+0.5,:));
title(strcat('Y-Velocity-Centre-Line at Re =', num2str(Re), ' and c = ', num2str(c)))

h8 = figure(8);
figure(9), contourf(vor, 50, 'LineStyle', 'none');colorbar;
title(strcat('Vorticity Contour at Re =', num2str(Re), ' and c = ', num2str(c)))

h9 = figure(9);

for ff = 1:9
    s4 = strcat(s3(9:end), '\fig_c', num2str(c*10), '_', num2str(ff), '.jpg');

    s5 = strcat(s3(9:end), '\fig_c', num2str(c*10), '_', num2str(ff), '.fig');
fs1 = strcat('saveas(h', num2str(ff), ', s4);');
fs2 = strcat('saveas(h', num2str(ff), ', s5);');
eval(fs1);
eval(fs2);
end

fid1 = fopen(strcat(s3(9:end), '\Re', num2str(Re), '_c',
num2str(c*100), '.dat'), 'w+');

fprintf(fid1, '\nThe maximum value of Continuity at any Node of the Domain is = %f, max(max(D)));

fprintf(fid1, '\nThe total value of Continuity residual throughout the Domain is = %f', sum(sum(D)));

fprintf(fid1, '\nNo of Iterations required for convergence for Re= %f, for C= %.1f is = %d', Re, c, k);

fprintf(fid1, '\nThe maximum value of Horizontal Pressure gradient w.r.t Time at any Node of the Domain is = %f', max(max(del_Px)));

fprintf(fid1, '\nThe total value of Horizontal Pressure Gradient residual w.r.t Time throughout the Domain is = %f', sum(sum(del_Px)));

fprintf(fid1, '\nThe maximum value of Vertical Pressure gradient w.r.t Time at any Node of the Domain is = %f', max(max(del_Py)));

fprintf(fid1, '\nThe total value of Vertical Pressure Gradient w.r.t Time residual throughout the Domain is = %f', sum(sum(del_Py)));

fprintf(fid1, '\nGrid Size for convergence at Re=%d, and C=%.1f is = %d X %d', Re, c, nx, nx);

fprintf(fid1, '\nDel_t for convergence at Re=%d, and C =%.1f is = %f', Re, c, del_t);
fprintf(fid1, \n Value of Stability criterion parameters
Aplha and Beta for this simulation run is %.3f & %.3f respectively', alpha, beta);

fclose(fid1);

dlmwrite(strcat(s3(9:end), '\Re', num2str(Re), '_c',
    num2str(c*100), '_u.dat'), u, ',');

dlmwrite(strcat(s3(9:end), '\Re', num2str(Re), '_c',
    num2str(c*100), '_v.dat'), v, ',');

dlmwrite(strcat(s3(9:end), '\Re', num2str(Re), '_c',
    num2str(c*100), '_vor.dat'), vor, ',');

dlmwrite(strcat(s3(9:end), '\Re', num2str(Re), '_c',
    num2str(c*100), '_del_Px.dat'),del_Px, ',');

dlmwrite(strcat(s3(9:end), '\Re', num2str(Re), '_c',
    num2str(c*100), '_del_Py.dat'), del_Py, ',');

dlmwrite(strcat(s3(9:end), '\Re', num2str(Re), '_c',
    num2str(c*100), '_Px.dat'),Px, ',');

dlmwrite(strcat(s3(9:end), '\Re', num2str(Re), '_c',
    num2str(c*100), '_Py.dat'),Py, ',');

dlmwrite(strcat(s3(9:end), '\Re', num2str(Re), '_c',
    num2str(c*100), '_D.dat'),D, ',');

close all;

end

end

else
printf(' The convergence cannot be obtained for this grid and
del_t value')

end
A.7 Matlab script for Multipath integration of velocity field to evaluate stream function

```matlab
1 % Evaluating stream function using Multi-path integration method
2 u=importdata('Re400_c100_u.dat');
3 v=importdata('Re400_c100_v.dat');
4 nx=size(u,1);
5 dx=1/(nx-1);
6 dy=dx;
7 psi=zeros(nx,nx);
8 for j=nx-1:-1:2
9    psi(j,1) = ((u(j,1)+u(j+1,1))/2*dy)+psi(j+1,1);
10   end
11 % Now integrate across each row left to right to get P at each point
12 for j=2:nx-1  % for all rows
13    for i=2:nx-1  % for columns 2-m
14        psi(j,i) = ((v(j,i)+v(j,i-1))/2*dx)+psi(j,i-1);
15    end
16  end
17 psi=psi(end:-1:1,:);
```
I, Shashank Mishra, hereby submit this original work as part of the requirements for the degree of Master of Science in Mechanical Engineering.

It is entitled:
Developing Novel Computational Fluid Dynamics Technique for Incompressible Flow and Flow Path Design of Novel Centrifugal Compressor

Student’s name: Shashank Mishra

This work and its defense approved by:

Committee chair: Shaaban Abdallah, Ph.D.
Committee member: Carissa Belloni, Ph.D.
Committee member: Kamar Venaganti, Ph.D.