2011

Effect of high order interpolation in the stability and efficiency of the time-integration process in vorticity-velocity CFD algorithms

Pankaj A. Jagadale
Michigan Technological University

Follow this and additional works at: https://digitalcommons.mtu.edu/etds
Part of the Mechanical Engineering Commons

EFFECT OF HIGH ORDER INTERPOLATION IN THE STABILITY AND EFFICIENCY OF THE TIME-INTEGRATION PROCESS IN VORTICITY-VELOCITY CFD ALGORITHMS

By

PANKAJ A. JAGADALE

A THESIS

Submitted in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

MECHANICAL ENGINEERING

MICHIGAN TECHNOLOGICAL UNIVERSITY

2011

This thesis, "Effect of High Order Interpolation in the Stability and Efficiency of the Time-Integration Process in Vorticity-Velocity CFD Algorithms," is hereby approved in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE in Mechanical Engineering.

Department of Mechanical Engineering–Engineering Mechanics

Signatures:

Thesis Advisor ______________________________

Dr. Fernando Ponta

Department Chair ______________________________

Dr. William Predeborn

Date ______________________________

Contents

List of Figures

List of Tables

List of Definitions

ρ = Density of fluid

$\mathbf{u}$ = Velocity field

$\boldsymbol{\sigma}$ = Stress tensor

$\mathbf{g}$ = Acceleration due to gravity

p = Pressure

T = Temperature

k = Conductivity

ν = Kinematic viscosity

$\hat{U}$ = Internal energy per unit mass

$\boldsymbol{\Phi}$ = Viscous dissipation function

C_v = Specific heat at constant volume

Ω = Control volume

S = Boundary surface

$\mathbf{b}$ = Velocity at the boundary

ϕ = Velocity potential

ψ = Stream function

$\boldsymbol{\omega}$ = Vorticity

$\boldsymbol{\tau}$ = Unit vector tangential to the boundary

$\mathbf{n}$ = Unit vector normal to the boundary

$\mathbf{H}$ = Array of interpolation functions

$\mathbf{B}$ = Array of derivatives of interpolation functions

NGL = Number of nodes of the Gauss-Lobatto points, N_{GL} = $p+1$

p = Order of the interpolating polynomial

$\mathbf{U}^e$ = Elemental array of nodal velocity values

$\boldsymbol{\omega}^e$ = Elemental array of nodal vorticity values

$\mathbf{H}_\omega$ = Array of vorticity interpolation functions

$\mathbf{B}_\omega$ = Array of derivatives of vorticity interpolation functions

γ = Angle of distortion in the controlled distortion experiment

$N^* = (NGL-1)*N_{el}$

N_{el} = number of elements in the mesh along the vertical or horizontal direction for the square plate problem

C = Constant with respect to y-intercepts

C_{NGL} = Constants corresponding to respective NGL

C_1 = CFL condition constant Courant number

α = Exponent of N^* in CFL condition relation

β = Exponent of interpolation order p in CFL condition relation

Abstract

The numerical solution of the incompressible Navier-Stokes Equations offers an effective alternative to the experimental analysis of Fluid-Structure interaction i.e. dynamical coupling between a fluid and a solid which otherwise is very complex, time consuming and very expensive. To have a method which can accurately model these types of mechanical systems by numerical solutions becomes a great option, since these advantages are even more obvious when considering huge structures like bridges, high rise buildings, or even wind turbine blades with diameters as large as 200 meters. The modeling of such processes, however, involves complex multiphysics problems along with complex geometries. This thesis focuses on a novel vorticity-velocity formulation called the KLE to solve the incompressible Navier-stokes equations for such FSI problems. This scheme allows for the implementation of robust adaptive ODE time integration schemes and thus allows us to tackle the various multiphysics problems as separate modules.

The current algorithm for KLE employs a structured or unstructured mesh for spatial discretization and it allows the use of a self-adaptive or fixed time step ODE solver while dealing with unsteady problems. This research deals with the analysis of the effects of the Courant-Friedrichs-Lewy (CFL) condition for KLE when applied to unsteady Stoke's problem. The objective is to conduct a numerical analysis for stability and, hence, for convergence. Our results confirm that the time step Δt is constrained by the CFL-like condition $\Delta t \leq const.\ h^{\alpha}$, where h denotes the variable that represents spatial discretization.

Chapter 1

Introduction

Most of the things in the universe exists in three states: solid, liquid and gaseous. The term "Fluid" is used to describe any matter that can be classified as liquid or gas. Fluids have tremendous importance in our natural environment as well as in almost every technical field of any known branch of science, whether it is particle physics, aerodynamics, combustion, astronomy, oceanography, or even bio-fluids. Study of fluids at rest or in motion is vital in a number of applications like designing turbines or engines, building dams or canals, studying molecules, blood circulation across body, global climate modeling, etc. Thus, fluid flow analysis becomes one of the most important areas of research.

Mathematically, fluid flow analysis for fixed set of forces, is governed by a non-linear partial differential equation, called as Navier-Stokes equation. It is time-dependent equation for conservation of momentum and is solved with the continuity equation i.e. conservation of mass equation, as a constraint. Analytical solution to those equations has not been known yet, although, in some special cases, these equations can be simplified and may have analytical solution. Given the scope of these equations, it is important and equally challenging to find out a numerical scheme to solve them with substantial accuracy.

The numerical solution of Navier-Stoke equations can be extremely useful in modeling physics of a range of applications in academics to industrial settings such as motion of air around an aerofoil, ocean currents, analysis of pollution, prediction of behavior of subatomic particles, etc. Such a solution could replace very expensive, difficult and sometimes impossible experimental analysis of fluid flows.

1.1 Fluid Structure Interaction

One of the most important applications of the numerical solutions of these equations would be the numerical modeling of fluid-structure interaction (FSI). FSI occurs when physical structure deforms due to inside or surrounding flow changing the boundary conditions for fluid. It elucidates a highly non-linear response in numerous engineering phenomena like metal forming processes, hydroplaning due to interaction between water, ground and tires, analysis of fatigue of airplane wings, wind effect on tall buildings, cable stayed bridges, effects of shock waves induced by such explosions on the hulls of ships and submarines, etc.

FSI explains the physics of dynamic stability of a structure in a wide range of applications which present excellent opportunities for scientific discovery with a richness of technical application.

An analytical solution to FSI problems is often very complex. FSI problems have to be analyzed either by experimental analysis or by numerical simulation. The experimental approach for some of the FSI problems is not a very attractive prospect either. If we consider the case of modern wind turbine design, the economy of scales factors have driven companies to consider rotors with diameters ranging from 160 to 200 meters. Working with such enormous blades and gathering data from wind tunnels is neither economical nor simple. Another example would be the modeling of huge civil structures like bridges, tall buildings and their interaction with high winds. Such huge structures carry risk of stress-related failures, and, hence, an accurate approximation of dynamic interaction is of utmost importance owing to the enormity of the surfaces involved. On the other hand, the microscale applications with complex roto-translational motion, like Micro-Air-Vehicles produce similar difficulty in modeling of flow patterns but on the opposite scale.

A numerical approach seems like an effective alternative to experimental prototypes and would also be much more economical. But it also has its own difficulties such as the complex physics involved in slender-body aeroelastic dynamics. The aeroelastic dynamics in slender bodies depends not only on the characteristic modes of the body structure itself, but also on the frequency and amplitude of the fluctuating aerodynamic forces. These forces are strongly affected by the dynamics of the vortex-wake shed from the body, which itself depends on the body's oscillations. Vortex-induced vibration can lead to catastrophic failure of engineering systems, as was clearly illustrated by the Tacoma Narrows Bridge disaster. In essence, if the work done on the body by the surrounding fluid over a complete cycle is positive, it results in the periodic vibration. The phase of the induced side force relative to the body motion greatly affects the net energy transfer between the body and the fluid, which is also related with the timing of the vortex dynamics (1).

A significant challenge in analyzing these systems is the fact that an oscillating body can produce a vortex wake that is very different from the classic Karman vortex street, which would translate into a complex fluctuating aerodynamic force. These rototranslational mechanisms are very sensitive to the varying loads and generate sudden structural responses to small changes in fluid flow parameters. This results in the need to have a dynamic control system incorporated in those mechanisms to optimize their efficiency and increase their lifespan. The simulation of the structural response and the dynamics of the control system as well as the complexity of the geometries involved to discretize, substantially complicates the issue of finding an effective method which can solve the final multiphysics non-linear PDE system. Further complications may arise with time-marching integration of multiphysics problems. Adaptive variable timestep/variable-order ODE algorithms provide a way to improve the efficiency of time marching schemes. But finding a way to combine those adaptive algorithms with the discretization of the spatial PDE problem has proved to be difficult.

An innovative mathematical model which resolves these problems and effectively simulates fluid-multiphysics involving fluid structure interaction was introduced in a paper by Ponta (2). This computational scheme is called the Kinematic Laplacian Equation (KLE) method. The KLE is based on a generalization of the concept of the hybrid formulation of the Navier-Stokes equations in which velocity and vorticity are used as unknown variables instead of the classical formulation in terms of the variables: pressure and velocity. The KLE is a natural extension of well-established vorticity-stream function method. The emergence of vorticity-velocity methods might be considered one of the most recent innovations in the computational solution of time-dependent viscous flows. Even though the appearance of what could be regarded as the first vorticity-velocity approach may be traced as early as 1976 (3), it is only during the last decade or so that a systematic research efforts have been applied to the development of this family of methods (see (2, 4) for a complete list of references). The vorticity-velocity methods present several advantages compared with the classical formulation on primitive variables (velocity-pressure) or with their vorticity-stream-function cousins. This is discussed in greater detail in Chapter 2.

The KLE algorithm solves the vorticity transport equation as an ODE problem in time with input velocity from the solution of a modified Poisson's equation in velocity, called the Kinematic Laplacian Equation, at each spatial node. The input to solve the KLE is provided by the time integration of the vorticity at each time step. Thus, it creates an evolving scheme in which the KLE provides the input for the ODE algorithm and vice-versa. Since time is the only iteration variable present, it is now possible to couple the fluid analysis with other physical mechanisms (e.g. structural response, control-system dynamics, etc.) by adding more equations to the ODE system. The KLE also shows a substantial tolerance to the use of unstructured meshes, which allows a more suitable meshing of complex geometries than structured-mesh approaches would permit. The latter is a very convenient feature for dealing with the complex aerodynamic shape of wind-turbine blades, helicopter-rotor blades, insect wings, or other aerodynamic surfaces.

1.2 An Emerging Field of Application: The Wind Power Challenge

This research is a small yet important part of the ongoing work towards the advancement of computational mathematical models for complex multiphysics problems involving fluid-structure-control interaction present in many engineering designs, providing a fundamental tool for a better understanding of the underlying physics. One such important engineering problem is the harnessing of wind power. Given the current trend of looking towards cheaper and cleaner ways to meet the ever increasing energy demand, wind energy might just be the answer to our growing needs. It is not only getting cheaper with technological innovations to harness wind energy on a large scale, wind power is also one of the cleanest ways to produce energy. During the last three decades there has been a spontaneous ten-

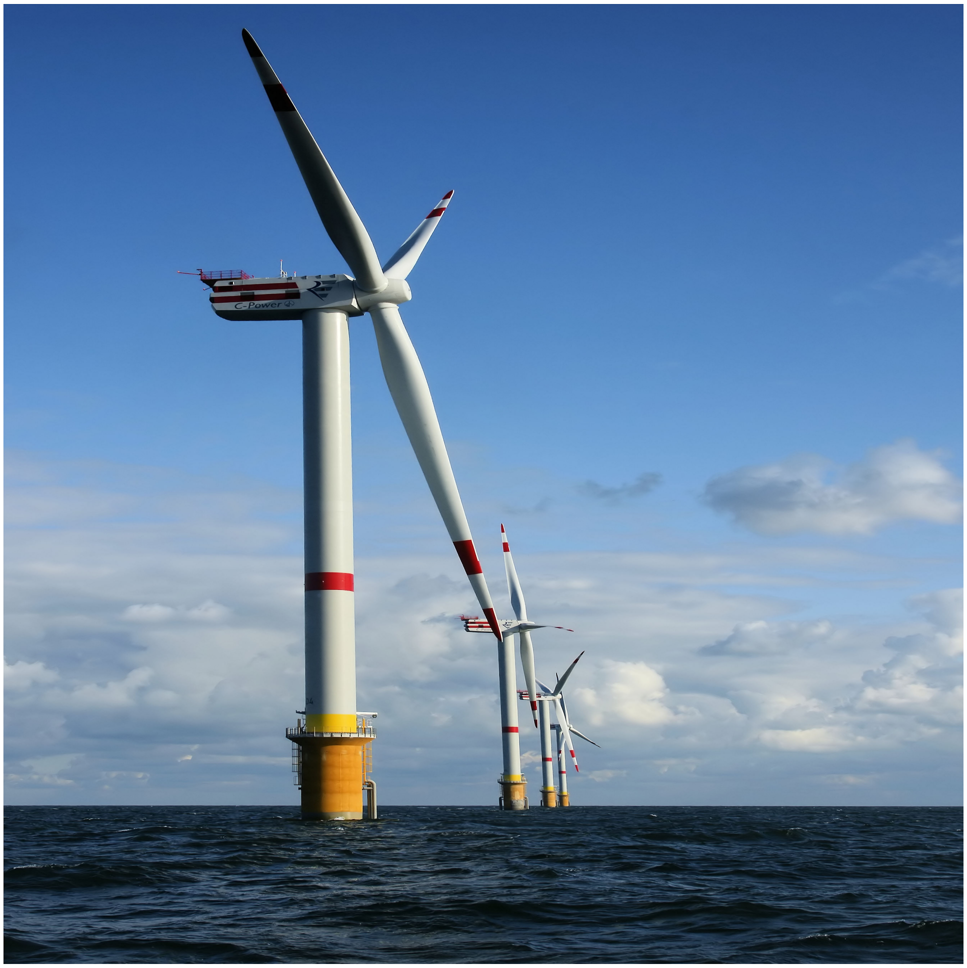


Figure 1.1: *The REpower M5 5-megawatt turbine, with a rotor diameter of approximately 126 meters (from (5)).Permission to reuse in Appendix.*

dency in the wind-turbine industry to increase the size of the state-of-the-art machine (6) and substantially reduce the cost of wind energy. Output power of these turbines range from 3 to 6 MW, rotor diameters range from 100 to 130 meters with hub height up to 143 meters making them gigantic structures (Figure 1.1). Next-generation offshore turbines with rotor diameters up to 200 meters have been suggested (7). The technological challenge in wind power nowadays is to develop the next generation of upscaled low-cost turbines that may further reduce generation costs. If this generation of superturbines is successfully developed, wind-energy costs would be reduced substantially. In fact, in favorable sites, it might be feasible to produce hydrogen as a substitute fuel in competitive terms, thereby getting rid of a significant roadblock towards developing hydrogen as an alternative fuel. Unfortunately current wind-turbine blade technology based on composite laminates is labor-intensive and requires a highly-qualified workforce, creating a critical bottleneck in terms of industrial workforce and infrastructure that hampers a rapid expansion of wind-energy. It also poses a barrier to turbine upscaling by increasing the share of the cost of the rotor as turbine size increases. The structural conception of today's blades also poses huge challenges in terms of transport logistics and crane capacity. Transportation cost increases as blades grow in length. The risk of damage during transportation, and, hence, the cost of insurance also increases with length. Moreover, while the rest of the turbine subsystems may be treated as modules assembled on site, blades are one-piece monolithic components, substantially complicating transport logistics. Limitations in crane capacity are the other critical factor to take into account during the turbine assembly phase. Thus, transport and lifting logistics may impose a premature limit for turbine upscaling, even before the actual limits in blade length for the current manufacturing technology are reached. Blades operate under a complex combination of fluctuating loads, and huge size differences complicate extrapolation of experimental data from the wind-tunnel to the prototype scale. Hence, computer models of fluid-structure interaction phenomena are particularly relevant to the design and optimization of wind-turbines. The wind-turbine industry is increasingly using computer models for blade structural design and for the optimization of its aerodynamics. But the complex interaction of physical processes that characterize the coupled aeroelastic problem still exceeds the capacities of existing commercial simulation codes. The result is an industry tendency to be cautious with the introduction of new concepts in order to ensure reliability. Innovations are likely to introduce changes in structural response and may possibly require different control strategies, which should be taken into account if the development of a new prototype blade is considered. Research efforts within the established parameters of the composite-laminate monolithic blade concept would not produce the breakthrough that is needed in wind-power evolution. A better understanding of the underlying physics is needed in order to introduce innovative concepts like modular blades and improved control strategies. This is where the KLE plays a pivotal role owing to its ability to create a common framework for modular integration of the aeroelastic model with control system dynamics.

1.3 Thesis Outline

The focus of the current thesis is to analyze the effects of Courant-Friedrichs-Lewy (CFL) condition and its dependency on the spatial discretization of an incompressible viscous flow problem using the KLE method. The experimentation is based on the canonical problem of a semi-infinite region of stationary fluid bounded by an infinite horizontal plate and given a sudden velocity in its own plane and thereafter maintained at that speed. This problem, along with its analytical solution, is discussed further in context with the KLE in Chapter

5. Since with the structured meshes it is simpler to study their impact on the solution, a structured spatial grid is considered for implementing the KLE. Given the effectiveness of self-adaptive Adam Bashforth Moulton methods with KLE, MATLAB solver ODE113 is used to run the unsteady part of the method. This solver determines its time step based on the stability and the rate of convergence, so it really suits the purpose of finding the limiting constraint on the time step.

The first part of the thesis is concerned with analyzing the nature of selected time steps by the solver on the basis of stability for the solution of the above mentioned flow problem using KLE. The results are then analyzed to find the common numerical relation between the time step and the parameters affecting the spatial grid. The ultimate objective is to come up with the CFL condition which is independent of all the factors except those parameters. The results obtained are quite promising and show the effects of CFL condition with the simple numerical correlation with the factors mentioned. All the numerical constants and courant number are computed with the analysis of the experimental data.

As mentioned before, the motivation for this research is the analysis of flow over wind turbine blades. Since this type of flow can be assumed to approximate incompressible flow, that is exactly the kind of flow which will be considered for this thesis. The next chapter deals with an introduction to the incompressible Navier-Stokes equations and gives a brief outline of some of the more popular solution methods in use. Chapter 3 is concerned with the hybrid methods based on a vorticity/velocity approach to solve the Navier-Stokes equations. The chapter then goes on to introduce the KLE along with its variational formulation. Chapter 4 deals with the numerical implementation of the KLE. Chapter 5 explains the experiments with the unsteady Stoke's problem and is concerned with the analysis of the results obtained by the KLE based on the experiment. This chapter is also concerned with the various mathematical operations and the subsequent calculations that leads to the development of CFL condition. Chapter 6 summarizes the results and ends with a brief on future prospects.

Chapter 2

Numerical methods

2.1 The Navier-Stokes Equations

Navier-Stokes Equations are non-linear partial differential equations of the second order. These consist of three basic conservation equations −

1. Conservation of mass or the continuity equation

   $$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \mathbf{u}) = 0$$

   where ρ is the density, $\mathbf{u}$ is the velocity field and t is the time.

2. Conservation of momentum

   $$\rho \frac{\mathrm{d}\,\mathbf{u}}{\mathrm{d}\,t} = \rho \mathbf{g} - \boldsymbol{\nabla} P + \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}$$

where, P is the pressure, $\boldsymbol{\sigma}$ is the stress tensor and ρg is the body force.

3. Conservation of energy

   $$\rho \frac{\mathrm{d}\,\hat{u}}{\mathrm{d}\,t} + p(\nabla \cdot \mathbf{u}) = \boldsymbol{\nabla} \cdot (k \boldsymbol{\nabla} T) + \boldsymbol{\Phi}$$

   where $\hat{U}$ is the internal energy per unit mass, k is the thermal conductivity, ν is the kinematic viscosity and the function $\boldsymbol{\Phi}$ represents energy dissipated due to viscous effects.

These equations can describe almost every fluid flow by determining various flow parameters like velocity, density, pressure, and temperature. In practice, the simplified form of this set of equations is used by assuming an incompressible flow of a Newtonian fluid. So, the above equations are reduced to $-$

1. Conservation of mass or the continuity equation

   $$\boldsymbol{\nabla} \cdot \mathbf{u} = 0 \tag{2.1}$$

2. Conservation of momentum

   $$\rho \frac{\mathrm{d}\,\mathbf{u}}{\mathrm{d}\,t} = \rho g - \boldsymbol{\nabla} P + \mu \boldsymbol{\nabla}^2 \mathbf{u} \tag{2.2}$$

   where μ is the coefficient of dynamic viscosity.

3. Conservation of energy

   $$\rho C_v \frac{\mathrm{d}\,T}{\mathrm{d}\,t} = k \boldsymbol{\nabla}^2 T + \boldsymbol{\Phi} \tag{2.3}$$

   where C_v is the specific heat at constant volume.

2.2 Numerical Solution of the Incompressible Viscous Flow

This section aims to explain some of the popular Navier-Stoke formulations for a viscous incompressible flow of a homogeneous fluid in an inertial frame of reference. A set of Equations 2.1, 2.2, and 2.3 can be derived from basic conservative equations by keeping density constant for a homogeneous incompressible flow. These set of equations shows the decoupling of the momentum equation with the energy equation for the incompressible flow which means both of these equations can be solved independent of temperature to get required velocity and pressure field.

The most important aspect in finding a numerical scheme would be which set of variables representing the equations is chosen. Based on the selected variables, there are two main formulations of the Navier-Stokes equations –

1. Primitive variable formulation
2. Non-primitive variable formulation

Both of these formulations have their own advantages and disadvantages which will be discussed in a subsequent discussion. The proper specification of boundary conditions is the main problem and the oldest point of contention for both formulations. It is mainly a problem of not having an obvious physical representation at the boundaries for pressure for primitive variable formulation and vorticity for non-primitive formulation.

2.2.1 Primitive variables

The pressure-velocity-based formulation is the fundamental formulation of the Navier-Stokes equations. It comprised of pressure and velocity as primitive variables. The Navier-Stokes equations for incompressible viscous flow in terms of the primitive variables without considering the body forces can be written as $-$

$$\frac{\partial \mathbf{u}}{\partial t} + u \cdot \boldsymbol{\nabla} \mathbf{u} = -\boldsymbol{\nabla} p + \nu \boldsymbol{\nabla}^2 \mathbf{u} + \mathbf{g} \tag{2.4}$$

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0 \tag{2.5}$$

where, $p = P/\rho$ is defined in a spatial domain Ω with S as its boundary. Here $\mathbf{u}$ is the velocity, p is the pressure divided by the fluid density (constant), and ν is the kinematic viscosity. The Navier-stokes equations represented by the primitive variables can be called a mixed elliptic parabolic equation since they are parabolic in time, mainly because of the convective diffusive term and elliptic in space due to the interaction between pressure and the continuity equation. To define the above problem completely, the equation needs to be supplemented by a set of boundary conditions specifying it as an Initial Value Boundary Value (IVBV) problem. The most common approach would be to start with the specification of an initial value for the velocity in Ω

$$\mathbf{u}(x, t_0) = \mathbf{u}_0(x), \text{such that } \boldsymbol{\nabla} \cdot \mathbf{u}_0 = 0, \text{where } x \in \Omega \tag{2.6}$$

followed by specifying the velocity at the boundary

$$\mathbf{u}(x,t) = \mathbf{u}_S(x,t), x \in S = \mathbf{b} \tag{2.7}$$

with a global continuity condition obtained by integrating the continuity equation over the entire volume and then using the Gauss theorem

$$\oint_S \mathbf{n} \cdot \mathbf{bdS} = \mathbf{0} \tag{2.8}$$

where, $\mathbf{n}$ is a unit vector normal to the boundary surface. So with the above expression it is clear that there is no boundary condition for pressure, which causes a number of problems and makes it a topic of debate among researchers.

As described in the preceding section, the serious difficulty in obtaining the numerical solution of the Navier-Stokes equations is determining the pressure field and fulfilling the incompressibility condition simultaneously. It can be seen from Equation 2.5 that velocity has a definite constraint, unlike pressure which appears in the Equation 2.4 with the term $\boldsymbol{\nabla} p$. For pressure, no evolutionary equation exists, hence it does not have any explicit representation. Also with the assumption of incompressibility, pressure does not have its thermo-dynamical meaning but it keeps adjusting itself instantaneously in order to satisfy Equation 2.5, i.e. condition of zero divergence at all times. Theoretically, pressure imitates the behavior of sound waves through an incompressible medium and travels at infinite speed. This is the most important factor for a non-fractional, or the standard discretization, approach. So any proposed numerical scheme needs to have proper boundary conditions considering implicit coupling between velocity and pressure. In addition, the velocity and pressure variables need to be decoupled to avoid the complex simultaneous equations.

According to (8), by using just incompressibility constraint, it is still possible to solve these complex simultaneous equations without imposing any boundary conditions on pressure. It is not very common in finite difference methods due to complexity of matrices as compared to regular block diagonal matrix. Even if this method is more common in finite elements, the following factors make it complicated as stated by (9) $-$

1. Non-linearity of the equations
2. The incompressibility needs to be maintained
3. The set of PDE's have been coupled through the advection term and continuity equation and sometimes the boundary condition

The most common method is by using a Poisson equation for the computation of pressure. The basic idea to form a Poisson equation is to take divergence of the momentum equation. This method is further explained in the following sections. The two types of approaches in velocity-pressure formulation are non-fractional step and fractional step methods.

2.2.1.1 Non$-$fractional Step Methods

Since Poisson equation is involved in both of these approaches, it is important to distinguish between them. The non-fractional step methods are those in which the velocity and pressure evolve simultaneously, whereas in the fractional step methods, the pressure Poisson equation replaces the incompressibility condition, making it a significantly more attractive method. By discretizing the non-linear advection term explicitly in time and pressure and viscous linear terms implicitly in time from Equations 2.4 and 2.5 and neglecting the body forces, we get –

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^{n}}{\triangle t} + \boldsymbol{\nabla} p^{n+1} = \nu \boldsymbol{\nabla}^{2} \mathbf{u}^{n+1} - \mathbf{u}^{n} \cdot \boldsymbol{\nabla} \mathbf{u}^{n} \tag{2.9}$$

$$\boldsymbol{\nabla} \cdot \mathbf{u}^{n+1} = 0 \tag{2.10}$$

where, $\mathbf{u}$ is the velocity vector and p is the pressure term.

Equation 2.9 can be written as

$$[-\boldsymbol{\Delta} + \gamma \mathbf{I}]\mathbf{u}^{n+1} + \boldsymbol{\nabla} p^{n+1} = g(\mathbf{u}^{n}) \tag{2.11}$$

where, ν is the kinematic viscosity, $\mathbf{I}$ is an Identity matrix, $\gamma = \frac{1}{\nu \triangle t}$ and $g(\mathbf{u}^{n}) = \gamma \mathbf{u}^{n} - \nu^{-1} \mathbf{u}^{n} \cdot \boldsymbol{\nabla} \mathbf{u}^{n}$

There are other time discretization methods such as semi-implicit or fully implicit schemes or other higher order schemes which can be used similarly as the one mentioned in the above equations for explanation. One of the popular methods is to use a higher order explicit scheme like Adams-Bashforth for the advective terms and Crank-Nicolson for diffusive terms. As explained before, Poisson equation is computed by taking divergence of the momentum equation, i.e. 2.11, which gives –

$$[-\Delta + \gamma \mathbf{I}]\boldsymbol{\nabla} \cdot \mathbf{u}^{n+1} + \boldsymbol{\nabla} p^{n+1} = g(\mathbf{u}^n) \tag{2.12}$$

$$\boldsymbol{\nabla}^2 p^{n+1} = \boldsymbol{\nabla} \cdot g(\mathbf{u}^n) \tag{2.13}$$

Even after simplifying the above two Equations 2.12 and 2.13, the solution obtained does not really confirm that the incompressibility condition is satisfied. But, it simply tells us that $[\Delta + \gamma \mathbf{I}]\boldsymbol{\nabla} \cdot \mathbf{u}^{n+1} = 0$. Thus $\boldsymbol{\nabla} \cdot \mathbf{u}^{n+1}$ is harmonic, but not necessarily zero. However, it was shown by Kleiser and Schumann (10) that if the divergence at the boundary is forced to be zero, i.e.

$$\boldsymbol{\nabla} \cdot \mathbf{u}^{n+1} \mid_S = 0 \tag{2.14}$$

Then, the Equations 2.12 and 2.13 can be written as a boundary value problem consisting of two elliptic equations as follows –

$$\begin{aligned} [-\Delta + \gamma \mathbf{I}]\boldsymbol{\nabla} \cdot \mathbf{u}^{n+1} + \boldsymbol{\nabla} p^{n+1} &= g(\mathbf{u}^n) \\ \boldsymbol{\nabla}^2 p^{n+1} &= \boldsymbol{\nabla} \cdot g(\mathbf{u}^n) \\ \mathbf{u}^{n+1} \mid_S &= \mathbf{b}^{n+1} \\ \boldsymbol{\nabla} \cdot \mathbf{u}^{n+1} \mid_S = 0\mathbf{u} \mid_S &= \mathbf{b} \end{aligned} \tag{2.15}$$

with the global constraint (2.8), i.e. $\oint_S \mathbf{n} \cdot \mathbf{u_S dS} = \mathbf{0}$.

Now there are two boundary conditions for $\mathbf{u}$ and none for pressure. But these equations still need to be solved simultaneously. So, it is essential to find out some way of decoupling these two equations with an appropriate boundary condition for pressure. There is one finite element method described by Glowinksi and Pironneau in (11) based on one extra equation for scalar velocity potential. In the same paper there is an another method as proposed by Kleiser and Schumann that is based on influence matrix technique. Also Quartapelle and Napolitano (12) explained a better approach on implementing integral boundary conditions on pressure which gives a better physical interpretation because of elliptic nature of the Poisson equation. These are some of the non-fractional step methods for the time-discretized Navier-Stokes problems.

2.2.1.2 Fractional step methods

The fractional-step projection method is the most widely used for solving the primitive variable formulation of the Navier Stokes equations. This method was first introduced by Chorin and Tenman, and has been one of the first numerical schemes which are capable of solving three dimensional time dependent problems. Consider the following set of equations, neglecting body forces –

$$\frac{\partial \mathbf{u}}{\partial t} + u \cdot \boldsymbol{\nabla}\mathbf{u} = -\boldsymbol{\nabla}p + \nu\boldsymbol{\nabla}^2\mathbf{u}$$

$$\boldsymbol{\nabla}\cdot\mathbf{u} = 0 \qquad\qquad \mathbf{u}\mid_S = \mathbf{b} \tag{2.16}$$

The common procedure is to discretize the momentum equation in time with the omission of the pressure term and approximate an intermediate velocity field, which would not be satisfying the incompressibility condition, from the solution of the time-discretized equation. This velocity field is decomposed into its divergent free component for the correction of final velocity field and irrotational component to enforce incompressibility. Subsequently, a pressure field is determined for that time step. Here is the intermediate step obtained from Equation 2.16 by avoiding completely the pressure term $-$

$$\frac{\mathbf{u}^{*}-\mathbf{u}^{n}}{\triangle t}=-(\mathbf{u}^{n}\cdot\boldsymbol{\nabla})\mathbf{u}^{n}+\nu\boldsymbol{\nabla}^{2}\mathbf{u}^{n} \qquad \mathbf{u}^{*}\mid_{S}=\mathbf{b}^{n+1} \tag{2.17}$$

Obviously, as stated above, the velocity field $\mathbf{u}^{*}$ would not be divergence free, which leads to the next step:

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^{*}}{\triangle t}=-\boldsymbol{\nabla}p^{n+1} \tag{2.18}$$

$$\boldsymbol{\nabla}\cdot u^{n+1}=0 \tag{2.19}$$

$$\mathbf{n}\cdot\mathbf{u}^{\mathbf{n+1}}\mid_{\mathbf{S}}=\mathbf{n}\cdot\mathbf{b}^{\mathbf{n+1}} \tag{2.20}$$

Equation 2.18 can also be formulated as:

$$\mathbf{u}^{*}=\mathbf{u}^{n+1}+\triangle t\boldsymbol{\nabla}p^{n+1} \tag{2.21}$$

where, $\boldsymbol{\nabla}p^{n+1}$ is not the gradient of pressure but of some artificial scalar function proportional to the unknown pressure often referred to as the “Pseudo-pressure”. The quantity $\mathbf{u}^{n+1}$ which is the required velocity field, is actually the solenoidal component of $\mathbf{u}^{*}$ and not the real $\mathbf{u}^{n+1}$ as the tangential boundary condition is not necessarily met. The normal boundary value for velocity is a consequence of the above step being inviscid. So the required velocity is calculated by projecting the velocity $\mathbf{u}^*$ onto a solenoidal space. The basis for the above step is the Helmholtz-Hodge decomposition of the velocity field based on (due to Olga Ladyzhenskaya) which states that any vector field can be decomposed as

$$\mathbf{v} = \mathbf{w} + \boldsymbol{\nabla}\phi$$

where, $\mathbf{w}$ is solenoidal and $\mathbf{n}\cdot\mathbf{w} = \mathbf{0}$ and ϕ is the potential function with its gradient giving the irrotational component of $\mathbf{v}$.

The gradient term can be further decomposed into $-$

$$\boldsymbol{\nabla}\phi = \boldsymbol{\nabla}\phi_0 + \boldsymbol{\nabla}h$$

where h is a harmonic function and $\phi_0 \mid_S = 0$. This leads to the vector $\mathbf{v}$ being decomposed into $-$

$$\mathbf{v} = \mathbf{w} + \boldsymbol{\nabla}\phi_0 + \boldsymbol{\nabla}h \tag{2.22}$$

Using this decompostion for the divergent $\mathbf{u}^*$,

$$\mathbf{u}^* = \mathbf{w} + \boldsymbol{\nabla}\phi_0 + \boldsymbol{\nabla}h \tag{2.23}$$

Adding and subtracting another harmonic function h_B to the RHS of Equation 2.39

$$\mathbf{u}^* = [\mathbf{w} + \boldsymbol{\nabla}h_B] + [\boldsymbol{\nabla}(h - h_B)] + \boldsymbol{\nabla}\phi_0] \qquad \mathbf{n}\cdot\boldsymbol{\nabla}\mathbf{h_B} = \mathbf{n}\cdot\mathbf{b} \tag{2.24}$$

Clearly $[\mathbf{w} + \boldsymbol{\nabla} h_B]$ represents $\mathbf{u}^{n+1}$ for an incompressible flow not satisfying the no-slip condition for the second half step. Thus, $\mathbf{u}^{n+1}$ can be found using a projection of $\mathbf{u}^*$ on the divergence free space. In order to calculate the velocity from Equations 2.18 and 2.20 the divergence of (2.18) is substituted into (2.19) to get the Poisson equation for pressure, also called the PPE.

$$-\boldsymbol{\nabla}^2 p^{n+1} = \frac{-1}{\triangle t}\boldsymbol{\nabla} \cdot \mathbf{u}^* \tag{2.25}$$

Using Equation 2.20 along with the boundary condition $\mathbf{u}^* \mid_S = \mathbf{b}^{n+1}$ the following boundary condition for the PPE can be derived.

$$\mathbf{n} \cdot \boldsymbol{\nabla} \mathbf{p}^{\mathbf{n+1}} \mid_{\mathbf{S}} = \mathbf{0} \tag{2.26}$$

After the pressure field is calculated, Equations 2.18 and 2.20 can be used to determine the required velocity field. The basic disadvantage here, as mentioned before, is that the second half step ensuring the incompressibility condition is inviscid, thereby, able to ensure only the normal component of the velocity boundary condition. This error is slightly qualified owing to the fact that the velocity boundary condition in the first half step is the no-slip condition. There are methods that introduce the viscous component in the second half step. One such method is mentioned in (10) in the form of a Crank-Nicholson Scheme:

$$\frac{\mathbf{u}^* - \mathbf{u}^{n+1}}{\triangle t} = -(\mathbf{u}^n \cdot \boldsymbol{\nabla})\mathbf{u}^n + \frac{1}{2}\nu \boldsymbol{\nabla}^2 \mathbf{u}^n \qquad \mathbf{u}^* \mid_S = \mathbf{b}^{n+1} \tag{2.27}$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\triangle t} = -\boldsymbol{\nabla} p^{n+1} + \frac{1}{2}\nu \boldsymbol{\nabla}^2 \mathbf{u}^{n+1} \qquad \mathbf{u}^{n+1} \mid_S = \mathbf{b}^{n+1} \tag{2.28}$$

This is second order accurate for the viscous term but, like the non−fractional step schemes, some complicated pressure condition must be used to enforce the incompressibility.

2.2.1.3 Artificial Compressibility

This method was introduced by Chorin in 1967 to take advantage of the advances made in conjunction with the analysis of compressible flow. This approach relaxes the strict need of enforcing the mass conservation equation at each time iteration. The basic idea behind this method is to change the continuity equation with a slightly modified version to make it compressible and solve it as an evolution equation in pressure. In order to achieve this, the continuity equation is modified by adding a time-derivative of the pressure term which gives:

$$\frac{1}{\beta}\frac{\partial p}{\partial t} + \frac{\partial \mathbf{u}_i}{\partial \mathbf{x}_i} \tag{2.29}$$

where β is an artificial compressibility or a pseudo-compressibility parameter. This was proposed originally for steady state Navier-Stokes equations, so that the β term disappears as the steady state is achieved. Here “t” does not represent the true physical time but an artificial or “pseudo” time. This gives hyperbolic-parabolic type of time dependent system of equations and thus many implicit schemes developed for compressible flows can be used to solve them. For a steady state formulation, Chorin proposed leap-frog time differencing and Dufort-Frankel spatial differencing (13) for pressure and velocity at the same grid points. Peyret and Taylor proposed the staggered grid formation with an explicit time differencing.

Physically, this method implies that the pressure waves propagate at finite speed into the incompressible flow field. But, as mentioned before, the wave speed is infinite in the incompressible flow medium. So for pseudo-waves, the propagation speed of pressure depends upon the artificial compressibility factor β. So the value of β should be chosen carefully. Ideally β value should be as high as possible to recover incompressibility quickly but at the same time a too high value of β tends to make the equation stiff. On the other hand, if β is too small, the waves travel too slowly, which will affect other factors like the viscous boundary layer, flow separation, etc., which might prevent convergence. The governing equations are iterated in the “pseudo” time until steady state is achieved. Although, owing to the compressibility introduced in the continuity equation, this method was not preferred for unsteady flow, it has been proven to be successful for such flows as well (14). The general idea for unsteady flows would be to use an iterative procedure using an artificial compressibility method for each physical time step ensuring that incompressibility is met at each step.

2.2.2 Non-Primitive variables

The formulation of Navier-Stokes equations using non-primitive variables is primarily based on a physical property of the fluid flow called vorticity which is very important in vortex dominant flows. It is a popular alternative to primitive variable formulation and is more sensible to analyze the flow based on vorticity, which has an extensively researched and understood transport equation. The study of vortex generation at boundaries, its diffusion, and advection are important in analyzing flow seperation, drag, etc., in vortex dominated flows. The vorticity formulation computes more accurate velocity field implicitly as the vorticity $\boldsymbol{\omega}$ is one order higher than the velocity $\mathbf{u}$. Vorticity-based methods also give a better estimate of the skin friction since they are based more on the shearing process itself. For flows with a high Reynold's number, the vorticity seems to be concentrated in the wake region, greatly reducing the computational domain. But this formulation too has its own set of problems. A common problem is the absence of boundary conditions for the vorticity in the presence of no-slip boundary conditions for the velocity. Also, the kinematic problem for the vorticity-velocity formulation is overdetermined, since the both Neumann and Dirichlet boundary conditions are described.

The boundary conditions are usually expressed in terms of velocities. So, the velocity boundary conditions are needed to compute the vorticity boundary conditions. The creation of vorticity can be attributed to the no slip boundary condition which results in a torque and, hence, an angular velocity being imparted on the packets of fluid. This vorticity creation at the boundary should be represented by the vorticity boundary conditions and has been the general reasoning used to tackle this issue. Lighthill (15) was one of the pioneers who proposed the basis for the most of the methods related to vorticity creation. His fractional step method was based on the part that the velocity induced by an arbitrary vorticity field does not satisfy either of the two velocity boundary conditions. Then the new velocity potential field is added so that the new velocity field will satisfy the normal velocity boundary condition, thereby getting slip velocity at the boundary. Lighthill proposed that this slip velocity is vortex sheet and it represents the creation of vorticity at the boundary. A similar approach was proposed by Chorin (15) for solving the Prandtl boundary layer equations for motionless boundaries. He basically split the Navier-Stokes equations into a viscid and an inviscid part. The Euler equation is then solved to give a slip velocity at the boundary. To get rid of this, vortex sheets are introduced and the resulting vorticity field is then used to solve the diffusion equation to get the correct vorticity field at the desired time step. This formulation does not seem to satisfy the no-slip condition as the normal boundary condition for velocity simultaneously and independent of the geometry. There are many such models for the vorticity creation at the boundary, but many believe that these models do not fully explain the vorticity interaction with solid boundaries.

This led Quartapelle and Valgriz (10) to introduce an integral constraint on the vorticity. This "nonlocal" approach couples the vorticity everywhere in the domain to the boundary velocity. A similar approach was adopted by Anderson, explained in (15) who suggested requiring the time derivative of these integral constraints be made to vanish. These methods as per Quartapelle (10) are the true representation of the vorticity diffusion and interaction with solid walls. These are of course a kind of projection methods, where an initial "wrong" vorticity, based on an arbitrary vorticty boundary value, is "corrected" by a projection onto the space of harmonic functions. This is achieved by the integral condition. This chapter deals with a particular type of formulation called the non-primitive variable formulation and a brief overview of solution strategies based on Quartapelle's book (10) is given for both two and three dimensional flows. The next chapter covers the hybrid formulation, also called the vorticity−velocity formulation.

2.2.2.1 Vorticity-Stream function formulation for two dimensional flows

One possible way to circumvent the problem of pressure boundary conditions is to eliminate the pressure term entirely. This is exactly what is achieved in the vorticity-stream function formulation of the Navier-Stokes equations. In this formulation the Navier-Stokes equations are represented in terms of the vorticity $\boldsymbol{\omega}$ and the stream function $\boldsymbol{\psi}$. So now the unknowns are $\boldsymbol{\omega}$ and $\boldsymbol{\psi}$ instead of u, v, p, reducing the number of unknowns by one. It also presents the added advantage of automatically taking care of the incompressibility condition owing to a property of the stream function. In two dimensions the above representation comprises two scalar equations obtained as follows:
In two dimensions Vorticity $\boldsymbol{\omega}$ is a scalar given by

$$\omega = (\boldsymbol{\nabla} \times \mathbf{u}) \cdot \mathbf{k} \tag{2.30}$$

while the velocity can be represented as the curl of a Stream-function ψ given by

$$\mathbf{u} = \boldsymbol{\nabla} \times \psi \tag{2.31}$$

Equation 2.31 clearly implies that $\boldsymbol{\nabla} \cdot \mathbf{u} = 0$. Substituting Equation 2.31 in 2.30 gives the Poisson's equation for the Stream-function.

$$-\boldsymbol{\nabla}^2 \psi = \omega \tag{2.32}$$

Taking the curl of the momentum equation and using Equations 2.30, 2.31 and $\boldsymbol{\nabla} \cdot \mathbf{u} = 0$ gives the vorticity transport equation.

$$\frac{\partial \omega}{\partial t} + J(\omega, \psi) = \nu \boldsymbol{\nabla}^2 \omega \tag{2.33}$$

where, $J(\omega, \psi)$ is the Jacobian matrix representing the curl of the advection term namely, $\boldsymbol{\nabla} \times [(\mathbf{u} \cdot \boldsymbol{\nabla})\mathbf{u}]$.

The Dirichlet and Neumann conditions for the above two equations are derived conditions deduced by separately tackling the normal and tangential components of boundary conditions of the velocity Equation 2.7 i.e. $\mathbf{u} \mid_S = \mathbf{b}$. They are given by, $\psi \mid_S = a$ and $\frac{\partial \psi}{\partial n} \mid_S = b$, where, $a = \int_{s_1}^{s} \mathbf{n} \cdot \mathbf{b} \mathbf{dS}$ and $b = -\boldsymbol{\tau} \cdot \mathbf{b}$ given that s_1 is any fixed point on the boundary and $\boldsymbol{\tau}$ is a unit vector tangential to the boundary. An initial condition for the vorticity can also be derived using the definition of vorticity and the initial condition for velocity Equation 2.6 giving the following initial condition.

$$\omega \mid_{t=0} = (\boldsymbol{\nabla} \times \mathbf{u}_0) \cdot \mathbf{k} \tag{2.34}$$

Therefore, the Navier-Stokes equations in the two dimensional vorticity-stream function formulation can be written as –

$$\begin{aligned} \frac{\partial \omega}{\partial t} + J(\omega, \psi) - \nu \boldsymbol{\nabla}^2 \omega = 0 \\ -\boldsymbol{\nabla}^2 \psi = \omega \\ \psi \mid_S = a \quad , \qquad \frac{\partial \psi}{\partial n} \mid_S = b \\ \omega \mid_{t=0} = (\boldsymbol{\nabla} \times \mathbf{u}_0) \cdot \mathbf{k} \end{aligned} \tag{2.35}$$

provided that

$$\begin{aligned} \boldsymbol{\nabla} \cdot \mathbf{u}_0 = 0 \\ \frac{\partial a(S,0)}{\partial s} = \mathbf{n} \cdot \mathbf{u_0} \end{aligned} \tag{2.36}$$

One of the problems with this kind of formulation is the nonlinear advection term which also couples the vorticity and stream function variables. The other important issue is the overspecification of ψ owing to both Dirichlet and Neumann conditions present as opposed to the underdetermined problem for ω with no boundary condition specified for it. The nonlinear terms can be dealt with using the standard linearizing techniques for nonlinear equations. Some of the ways in which the problem of the boundary conditions are tackled are discussed below.

2.2.2.2 Biharmonic formulation

One way to avoid the boundary value problem for vorticity is to eliminate the vorticity term itself from the transport equation. This is achieved by substituting the Poisson equation for the stream function into the voticity transport equation resulting in the following equation:

$$\frac{\partial^2 \boldsymbol{\nabla}^2 \psi}{\partial t} - \nu \boldsymbol{\nabla}^4 \psi + J(\boldsymbol{\nabla}^2 \psi, \psi) = 0$$

$$\psi \mid_S = a \quad , \qquad \frac{\partial \psi}{\partial n} \mid_S = b \tag{2.37}$$

where, ψ_0 is the solution of the Dirichlet problem

$$-\boldsymbol{\nabla}^2 \psi_0 = (\boldsymbol{\nabla} \times \mathbf{u}_0) \cdot \mathbf{k} \quad , \qquad \psi_0 \mid_S = a(S,t) \tag{2.38}$$

where $\mathbf{u}_0$ and a satisfy the solenoidal and the compatibility condition and $\mathbf{k}$ is the unit vector in the Z direction. Since the equation is fourth order elliptic the specification of both Dirichlet and Neumann conditions no longer make it overspecified.

2.2.2.3 Coupled formulation in vorticity and stream function

Another method to eliminate the problems associated with the overdetermined problem is to solve Equation 2.35 as a coupled equation in vorticity and stream function even in the absence of the nonlinear term. This is achieved by a unique coupling through the boundary conditions by associating one boundary condition with the transport equation and the other with the Poisson's equation. This can be written as $-$

$$\frac{\partial^2 \boldsymbol{\nabla}^2 \psi}{\partial t} - \nu \boldsymbol{\nabla}^4 \psi + J(\boldsymbol{\nabla}^2 \psi, \psi) = 0 \qquad \text{such that} \quad \psi \mid_S = a$$

$$-\boldsymbol{\nabla}^2 \psi = \omega \qquad \text{such that} \quad \frac{\partial \psi}{\partial n} \mid_S = b \tag{2.39}$$

The spatial discretization for both these methods can be done by any of the three methods, namely finite elements, finite differences, or spectral methods. In spite of no direct implementation of the integral constraint, the couple formulation still satisfies the nonlocal character of vorticity, hence, further fortifying the idea of an integral condition on vorticity.

2.2.2.4 Uncoupled formulation using vorticity integral conditions

In order to split the two terms in the vorticity stream function formulation, it becomes necessary to determine some supplementary conditions for the vorticity to account for its lack of boundary conditions. Quartapelle and Valz-Cris (10) came up with an integral constraint on vorticity.

$$\int \omega\eta d\Omega = \oint (a\frac{\partial\eta}{\partial n} - b\eta)dS \tag{2.40}$$

giving the following set of linearized equations

$$(-\Delta + \gamma)\omega = f, \qquad \int \omega\eta d\Omega = \oint (a\frac{\partial\eta}{\partial n} - b\eta)dS \tag{2.41}$$

$$-\nabla^2\psi = \omega, \qquad \psi \mid_S = a$$

where η is any harmonic function defined in the domain Ω. This is a semi-implicit discretization in time with $\omega \equiv \omega_{n+1}$ and $\psi \equiv \psi^{n+1}$ at the new time level t^{n+1}. One of the ways to implement the integral conditions and solve the uncoupled equations is based on utilizing the linearity of the above formulation. It consists of decomposing the vorticity transport equation using the principle of superposition. The split formulation can be represented in the following way which is similar to that of for the primitive variable formulation:

$$\omega(\mathbf{x}) = \omega^0(\mathbf{x}) + \oint \omega'(\mathbf{x};\zeta')\lambda(\zeta')dS(\zeta') \tag{2.42}$$

where, ω^0 and, ω' are the solutions to

$$(-\Delta + \gamma)\omega^0 = f \qquad\qquad \omega^0 \mid_S = 0 \tag{2.43}$$

$$(-\Delta + \gamma)\omega' = 0 \qquad\qquad \omega'(\mathbf{x};\zeta') = \delta(s - \zeta') \tag{2.44}$$

for any $\zeta' \in S$ and δ is the dirac delta function over the boundary

The value of the boundary function can then be evaluated by imposing on ω the integral constraint (2.40) with respect to all harmonic functions on the boundary which are the solution to the following problem –

$$-\boldsymbol{\nabla}^2\eta = 0 \qquad \text{such that} \quad \eta(\mathbf{x};\zeta) = \delta(s-\zeta) \qquad \text{for any } \zeta \in S \tag{2.45}$$

And finally leads to a linear equation of the type $\mathbf{A}\boldsymbol{\lambda} = \boldsymbol{\beta}$

where, the value of the matrix $\mathbf{A}$ is of the form $\int \omega(\mathbf{x};\zeta')\eta(x;\zeta)d\Omega$ and can be calculated and stored in the beginning.

The stream function in the above form exists only for flow in two dimensions, so the vorticity-stream function formulation is rather difficult to implement for a three dimensional flow. Also, the vorticity is now a vector with two tangential components on the boundary. The solenoidal property of the vorticity is no longer implied by its definition but needs to be enforced. It will be shown in the next chapter that the divergence of the vorticity vector in three dimensions can be enforced to be equal to zero by the following two boundary conditions –

$$\boldsymbol{\nabla} \cdot \boldsymbol{\omega} \mid_S = 0 \tag{2.46}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{\omega} \mid_{t=0} = \boldsymbol{\nabla} \cdot (\boldsymbol{\nabla} \times u_0) \tag{2.47}$$

But the real problem starts with the boundary conditions for the three dimensional "equivalent" for the stream function for which different schemes involving different vector potentials have been developed, each having its own set of boundary conditions and its own set of elliptic equations to solve. But each method has to come up with a set of boundary conditions for the vector function to ensure its unique solution. Apart from having to solve such complex equations with often debatable boundary conditions, these methods are also not well suited for a variational approach which often turns out to be computationally very expensive. On account of these issues with the three dimensional approach, another method has begun to garner a lot of interest. This new approach, called the hybrid methods, uses a vorticity velocity formulation of the Navier–Stokes equations and seems to be quite well suited to two-dimensional as well as three-dimensional flows. This method will form the basis of this research and is explained in the next chapter.

Chapter 3

Hybrid methods

3.1 Introduction

These methods are based on hybrid formulations in terms of the primitive and nonprimitive variables velocity and vorticity. They are well suited for both two and three-dimensional flows. Some of the advantages of vorticity-velocity ($\boldsymbol{\omega}$, $\boldsymbol{v}$) formulations compared to the classical formulation with primitive variables or with the nonprimitive vorticity-stream function methods (2) are:

1. Vorticity is a relevant physical variable which has been extensively studied and its distribution is of immense importance. The velocity is perhaps the most important physical variable which completely defines the kinematical problem at hand. And

the fact that they are related by a simple elliptic equation makes this approach all the more advantageous.

2. The velocity can be supplemented by a unique set of boundary conditions as opposed to a vast number of boundary conditions necessary for a unique solution of the stream vectors or the velocity potentials.

3. The non-inertial terms caused by an accelerating reference frame enter into the flow solution through the initial and boundary conditions, without having to do anything extra to evaluate those non-inertial terms.

4. Relative ease of implementing vorticity conditions at infinity as compared to that for pressure.

But hybrid formulation also has some disadvantages too. As already mentioned, the issue with this method is the over-determined kinematic problem and the underdetermined dynamic problem. The unsteady problem in three dimensions has six unknowns compared to the four in primitive variable methods.The general formulation can be written as $-$

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{\nabla} \times (\boldsymbol{\omega} \times \mathbf{u}) = \nu \nabla^2 \boldsymbol{\omega} \tag{3.1}$$

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0 \tag{3.2}$$

$$\boldsymbol{\omega} = \boldsymbol{\nabla} \times \mathbf{u} \tag{3.3}$$

Many methods involve a Poisson's equation obtained from the curl of Equation 3.3 and utilizing (3.2) to give

$$\boldsymbol{\nabla}^2 \mathbf{u} = -\boldsymbol{\nabla} \times \boldsymbol{\omega} \tag{3.4}$$

The major problem here is ensuring the divergence condition as well as the curl of velocity. As mentioned before, the zero divergence of velocity can be ensured throughout the domain by enforcing it on the boundary. But now the solenoidal property for the vorticity also has to be ensured, which can be done in the following way.

Taking the divergence of the transport equation gives the following diffusion

$$\frac{\partial(\boldsymbol{\nabla} \cdot \boldsymbol{\omega})}{\partial t} = \nu \boldsymbol{\nabla}^2 (\boldsymbol{\nabla} \cdot \boldsymbol{\omega}) \tag{3.5}$$

Imposing the boundary condition $\boldsymbol{\nabla} \cdot \boldsymbol{\omega} \mid_S = 0$ on the divergence of vorticity, along with the obvious initial condition $\boldsymbol{\nabla} \cdot \boldsymbol{\omega} \mid_{t=0} = \boldsymbol{\nabla} \cdot (\boldsymbol{\nabla} \times \mathbf{u}_0)$, should give a unique solution to the diffusion equation for $\boldsymbol{\nabla} \cdot \boldsymbol{\omega}$, i.e. $\boldsymbol{\nabla} \cdot \boldsymbol{\omega} = 0$ therefore, the solenoidal property is confirmed.

Gatski (16) has classified the solution strategies into method A and method B. Method A "utilizes" the continuity and curl Equations 3.2 and 3.3 as the kinematic equations to solve and Equation 3.1 as the dynamic transport equation. Method B comprises of solving Equations 3.1 and 3.4.

Fasel (3) was among the first to publish numerical results of this method. He used the normal component of the Poisson equation for velocity and the tangential derivative of the continuity equation along with the vorticity transport equation. The boundary condition, in addition to the usual velocity condition, was simply the above-mentioned Poisson's equation on the boundary. Like this, many formulations based on a derived vorticity condition were formulated. Since none of these conditions are genuine constraints, many authors have solved the governing equations without using any vorticity boundary conditions whatsoever. This has led to the use of an integral constraint on the vorticity rather than a local boundary condition.

Since most of the methods do not ensure a solenoidal vorticity field by virtue of the boundary conditions (17), many authors have resorted to using a projection method to ensure the solenoidal property of vorticity. Of course, as shown by Wu, et al. (18), the non-solenoidal vorticity can be used to solve for a solenoidal velocity field. However, to find a vorticity field that is solenoidal, the vorticity is decomposed by the Helmholtz theorem to get the Poisson equation,

$$\boldsymbol{\nabla}^2\phi = \boldsymbol{\nabla}\cdot\boldsymbol{\omega}^0 \qquad (3.6)$$

where, $\boldsymbol{\nabla}\phi$ is the solenoidal part of the computed vorticity. Once $\boldsymbol{\nabla}\phi$ is solved for from the above equation, the non-solenoidal vorticity can be projected onto the solenoidal field using the relation $-$

$$\boldsymbol{\omega} = \boldsymbol{\omega}^0 - \boldsymbol{\nabla}\phi$$

The next section deals with some of the formulations of the vorticity$-$velocity methods as presented in (10)

3.1.1 Equations in three dimensions

Using the above-mentioned boundary and initial conditions for vorticity along with the definition $\boldsymbol{\nabla} \times \mathbf{u} = \boldsymbol{\omega}$, the vorticity$-$velocity formulation in three dimensions can be represented as $-$

$$\begin{aligned} &\frac{\partial(\boldsymbol{\nabla}\cdot\boldsymbol{\omega})}{\partial t} - \nu\boldsymbol{\nabla}^2(\boldsymbol{\nabla}\cdot\boldsymbol{\omega}) = 0 \qquad \boldsymbol{\omega}\mid_{t=0} = \boldsymbol{\nabla}\times\mathbf{u}_0 \\ &\boldsymbol{\nabla}^2\mathbf{u} = -\boldsymbol{\nabla}\times\boldsymbol{\omega} \qquad \mathbf{u}_S = \boldsymbol{b} \qquad \boldsymbol{\nabla}\cdot\mathbf{u}_S = 0 \\ &\boldsymbol{n}\cdot\boldsymbol{\omega}\mid_S = \boldsymbol{n}\cdot\boldsymbol{\nabla}\mid_S \times\boldsymbol{b} \qquad \boldsymbol{\nabla}\cdot\boldsymbol{\omega}\mid_S = 0 \end{aligned} \tag{3.7}$$

Given the compatibilty conditions $\oint \boldsymbol{n}\cdot\boldsymbol{b}dS = 0$, $\boldsymbol{\nabla}\cdot\mathbf{u}_0 = 0$, $\boldsymbol{n}\cdot\boldsymbol{b}_{t=0} = \boldsymbol{n}\cdot\mathbf{u}_0\mid_S$

The above formulation can be solved numerically by a semi-implicit discretization in time and a spectral method for spatial discretization. The lack of boundary values for vorticity can be taken care of by the influence matrix technique as introduced by Daube (19). This will be briefly introduced in the next section for two dimensional flows along with a temporal discretization scheme to linearize the formulation before resolving it in space.

An uncoupled formulation using an integral constraint can be written in the following way,

$$\begin{aligned} &\frac{\partial(\boldsymbol{\nabla}\cdot\boldsymbol{\omega})}{\partial t} - \nu\boldsymbol{\nabla}^2(\boldsymbol{\nabla}\cdot\boldsymbol{\omega}) = 0 \qquad \boldsymbol{\omega}\mid_{t=0} = \boldsymbol{\nabla}\times\mathbf{u}_0 \\ &\int \boldsymbol{\nabla}\times\boldsymbol{\omega}\cdot\boldsymbol{\eta}d\Omega = \oint(\boldsymbol{n}\times\boldsymbol{b}\cdot\boldsymbol{\nabla}\times\boldsymbol{\eta} + \boldsymbol{n}\cdot\boldsymbol{b}\boldsymbol{\nabla}\cdot\boldsymbol{\eta})dS \\ &\boldsymbol{n}\cdot\boldsymbol{\omega}\mid_S = \boldsymbol{n}\cdot\boldsymbol{\nabla}\mid_S \times\boldsymbol{b} \qquad \boldsymbol{\nabla}\cdot\boldsymbol{\omega}\mid_S = 0 \\ &\boldsymbol{\nabla}^2\mathbf{u} = -\boldsymbol{\nabla}\times\boldsymbol{\omega} \qquad \mathbf{u}_S = \boldsymbol{b} \end{aligned} \tag{3.8}$$

Where, η is a harmonic vector field defined by the following problem,

$$-\boldsymbol{\nabla}^2\eta = 0, \qquad \boldsymbol{n} \times \boldsymbol{\eta} \mid_S = 0 \tag{3.9}$$

The problem here is the implementation of the the integral constraint in a variational form. The culprits are the boundary conditions $\boldsymbol{n} \cdot \boldsymbol{\omega} \mid_S = \boldsymbol{n} \cdot \boldsymbol{\nabla}_S \times \boldsymbol{b}$ and $\boldsymbol{\nabla} \cdot \boldsymbol{\omega} \mid_S = 0$ which cannot be used together in a variational formulation.

3.1.2 Equations in two dimensions

As shown in the vorticity$-$stream function formulation for two dimensions, the vorticity is now a scalar variable given by $\omega = \boldsymbol{\nabla} \times \mathbf{u} \cdot \boldsymbol{k}$ giving the following set of governing equations $-$

$$(\frac{\partial \omega}{\partial t} - \nu \boldsymbol{\nabla}^2 \omega)\boldsymbol{k} + \boldsymbol{\nabla} \times (\omega \boldsymbol{k} \times \mathbf{u}) = 0, \qquad \omega_{t=0} = \boldsymbol{\nabla} \times \mathbf{u}_0 \cdot \boldsymbol{k} \tag{3.10}$$

$$\int \omega d\Omega = \oint \boldsymbol{\tau} \cdot \boldsymbol{b} dS \tag{3.11}$$

$$-\boldsymbol{\nabla}^2 \mathbf{u} = \boldsymbol{\nabla}\omega \times \boldsymbol{k} \qquad \boldsymbol{\nabla} \cdot \mathbf{u} \mid_S = 0 \qquad \mathbf{u} \mid_S = \boldsymbol{b} \tag{3.12}$$

Provided the compatibility conditions $\oint \boldsymbol{n} \cdot \boldsymbol{b} dS = 0$, $\boldsymbol{\nabla} \cdot \mathbf{u}_0 = 0$, $\boldsymbol{n} \cdot \boldsymbol{b}_{t=0} = \boldsymbol{n} \cdot \mathbf{u}_0 \mid_S$ are met. Here $\boldsymbol{\tau}$ is a unit vector tangential to the boundary.

As before, a semi-implicit dicretization can be performed by first discretizing the advection term explicitly, followed by an implicit scheme like the Crank-Nicolson scheme for the diffusion terms. This gives the following linearized time dicretized formulation,

$$
\begin{aligned}
(-\boldsymbol{\nabla}^2+\gamma)\omega^{n+1} &= f, \qquad \text{in } \Omega \\
-\boldsymbol{\nabla}^2\mathbf{u}^{n+1} = \boldsymbol{\nabla}\omega \times \boldsymbol{k}^{n+1} \qquad\qquad \mathbf{u} &\mid_S = \boldsymbol{b} \\
\boldsymbol{\nabla} \times \mathbf{u}^{n+1} &= \omega \boldsymbol{k}^{n+1} \\
\oint_S \boldsymbol{b}^{n+1} \cdot \boldsymbol{n} dS &= 0
\end{aligned}
\tag{3.13}
$$

Just as for the primitive variable formulation, an influence matrix technique devised by Kleiser and Schumann can be used for the vorticity−velocity formulation as well.

Solving these methods numerically, as with the three dimensional case, has not been free of problems for both the finite element and the finite difference methods. The integral constraint does offer a better representation of vorticity diffusion and its interaction with solid boundaries, there is an inclination towards using such constraints. Several innovative techniques like the staggered discretization of vorticity by Napolitano and Pascazio (10) have resulted in avoiding certain problems related to a doubly singular influence matrix in the above linear equation. The following sections describes how these problems can be solved by a new method belonging to the hybrid family.

3.2 The Kinematic Laplacian equation method

The Kinematic Laplacian equation method was first introduced by Ponta in a paper (20). It is a vorticity−velocity method which decouples the evolution of vorticity from the spatial solution of a velocity field. Vorticity is advanced in time by integrating a vorticity transport equation for which an initial velocity field is obtained from the solution of the weak form of a PDE called the Kinematic Laplacian equation (the KLE). The KLE in turn is solved using the vorticity field obtained by the time integration of vorticity from the previous time step. The no-slip, no-normal flow Boundary conditions for velocity required for solving the KLE are solved over a sequence of two steps. This basically involves two integral projection in each time step ensuring compatibility of the two fields at each step. This is explained in greater detail in the following sections which are taken from a paper on KLE by F.L. Ponta (21).

3.2.1 The constant-curl Laplacian equation

As stated in (22), the idea behind using a Laplacian was to come up with a simple linear PDE along the lines of a potential flow equation, but, which could also account for rotational effects as seen in turbines. This lead to a Kinematic equation for solving time dependant flows over slender bodies with no flow separation under the assumption of incompressible flow and a constant curl.The following vector relation can be used to get the

Laplacian of the velocity field –

$$\nabla^2\mathbf{u} = \boldsymbol{\nabla} \cdot \boldsymbol{\nabla}\mathbf{u} = \boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \mathbf{u}) - \boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times \mathbf{u}). \tag{3.14}$$

The first and second terms can be ignored on account of the incompressibility and constant curl condition.Thus, the Laplacian $\nabla^2\mathbf{u} = 0$ can be solved numerically under the incompressibility condition and the constant curl constraint. i.e. $\boldsymbol{\nabla} \cdot \mathbf{u} = 0$ and $\boldsymbol{\nabla} \times \mathbf{u} = c$, where c is a constant.

This earlier version of KLE called the *constant curl Laplacian equation* (CCLE) (22), had a narrow field of application owing to the constraint of no flow separation. Nevertheless, CCLE was quite successfully used in the study of wind turbine blades (21).

3.2.2 A generalized Laplacian (ω, u) method: The KLE

As mentioned previously the KLE can be solved to get the spatial distribution of vorticity and velocity. It's a more general PDE expression than the CCLE, not limited to non-separated flows. Consider a vorticity velocity formulation for a three dimensional Navier–Stokes equation for incompressible viscous flow. Consider a domain with a solid boundary S and a far field external boundary Ω. Therefore in a non-inertial reference frame,

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = -\mathbf{u} \cdot \boldsymbol{\nabla}\boldsymbol{\omega} + \nu\nabla^2\boldsymbol{\omega} + \boldsymbol{\omega} \cdot \boldsymbol{\nabla}\mathbf{u} \tag{3.15}$$

If the velocity field is known at a particular time step then the above equation can be written as –

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = -\mathbf{u} \cdot \boldsymbol{\nabla}(\boldsymbol{\nabla} \times \mathbf{u}) + \nu \nabla^2 (\boldsymbol{\nabla} \times \mathbf{u}) + (\boldsymbol{\nabla} \times \mathbf{u}) \cdot \boldsymbol{\nabla} \mathbf{u} \tag{3.16}$$

This can be integrated in time to solve for ω at each node using an ODE solver using the vorticity and velocity field from the previous time step. However, the vorticity field so calculated is not compatible with the instantaneous boundary conditions, and so to solve for the correct vorticity and velocity fields in the spatial domain, the following Laplacian equation is used –

$$\nabla^2 \mathbf{u} = \boldsymbol{\nabla} \mathcal{D} - \boldsymbol{\nabla} \times \boldsymbol{\omega} \tag{3.17}$$

$$\boldsymbol{\nabla} \cdot \mathbf{u} = \mathcal{D} \tag{3.18}$$

$$\boldsymbol{\nabla} \times \mathbf{u} = \boldsymbol{\omega} \tag{3.19}$$

As explained in (20) the KLE is basically a solution of the weak form of (3.17) under the simultaneous imposition of the expansion rate and the curl of the velocity i.e. the vorticity field. These constraints are given by (3.18) and (3.19).

Sections 2.4 to 2.7 of (23) gives a good explanation of the physical significance of the two constraints. Most hybrid methods simultaneously solve equations (3.17) and (3.16) under the $\boldsymbol{\nabla} \cdot \mathbf{u} = 0$ constraint i.e. incompressibility. The KLE, however, as mentioned earlier solves (3.17) independent of the vorticity transport equation. Therefore the vorticity distribution given by (3.19) can be used as a second constraint along with the rate of expansion given by (3.18) to solve for the velocity field in space. For a brief validation, consider the decomposition of the velocity field into three orthogonal components: the irrotational component $\mathbf{u}_{\mathcal{D}}$ with zero divergence, the solenoidal component $\mathbf{u}_\omega$ with no vorticity and the harmonic component $\mathbf{u}_h$. Given the no-normal flow at the boundary along with the vorticity distribution the above mentioned decomposition i.e. $\mathbf{u} = \mathbf{u}_{\mathcal{D}} + \mathbf{u}_\omega + \mathbf{u}_h$ has a unique solution (23). (3.18) and (3.19) can be used to solve for $\mathbf{u}_{\mathcal{D}}$ and $\mathbf{u}_\omega$ as $-$

$$\boldsymbol{\nabla} \cdot \mathbf{u} = \boldsymbol{\nabla} \cdot \mathbf{u}_{\mathcal{D}} = \mathcal{D} \tag{3.20}$$

$$\boldsymbol{\nabla} \times \mathbf{u} = \boldsymbol{\nabla} \times \mathbf{u}_\omega = \boldsymbol{\omega}. \tag{3.21}$$

For $\mathbf{u}_h$ substitute the above mentioned decomposition in to (3.16),

$$\begin{aligned} \nabla^2(\mathbf{u}_h + \mathbf{u}_{\mathcal{D}} + \mathbf{u}_\omega) &= \nabla^2\mathbf{u}_h + \boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \mathbf{u}_{\mathcal{D}}) - \boldsymbol{\nabla} \times (\boldsymbol{\nabla} \times \mathbf{u}_\omega) \\ &= \boldsymbol{\nabla}\mathcal{D} - \boldsymbol{\nabla} \times \boldsymbol{\omega} \end{aligned} \tag{3.22}$$

Substituting (3.20) and (3.21) in (3.22) gives,

$$\nabla^2\mathbf{u}_h = 0 \tag{3.23}$$

This Laplacian equation gives the solution for $\mathbf{u}_h$. Therefore the KLE ensures a complete and unique solution of the velocity field.

To impose the no-normal flow and no-slip velocity boundary conditions on S together with the correspondingly compatible boundary conditions on the vorticity, a solution method based on two consecutive solutions of the KLE is used: the first under free-slip and the second under no-slip boundary conditions on the solid surface. The algorithmic sequence explained below (20, 21) is iteratively performed at each time step within an iterative time integration performed by an adaptive variable-stepsize ODE solver for incompressible flow.

1. The vorticity is advanced in time by integrating (3.16) in time at each node in space to get an initial vorticity field $\tilde{\boldsymbol{\omega}}^n$ field. Since velocity from the $(n-1)^{st}$ time step is used to get vorticity for step n, this field is not compatible with the velocity boundary conditions.

2. Enforce homogeneous conditions $\tilde{\boldsymbol{\omega}}^n$ at the boundary surface to get $\tilde{\boldsymbol{\omega}}_0^n$. This is done by imposing a zero boundary value for vorticity at each node on the boundary.

3. Applying the no normal flow velocity boundary conditions and setting $\frac{\partial u_x}{\partial \boldsymbol{n}} = 0$ on the solid boundary, equation (3.17) i.e. the KLE is solved for $\vec{\boldsymbol{u}}^n$ under the 2 constraints given by (3.18) and (3.19) using $\tilde{\boldsymbol{\omega}}_0^n$ as the vorticity field. Here u_x is the tangential component of $\boldsymbol{u}$.

4. Using this $\vec{\boldsymbol{u}}^n$ the vorticity field is again calculated as $\boldsymbol{\omega}^n = \boldsymbol{\nabla} \times \vec{\boldsymbol{u}}^n$, only this time both boundary conditions i.e. no-normal flow $\boldsymbol{u}.\boldsymbol{n} = 0$ and the no slip condition $\boldsymbol{u}.\boldsymbol{\tau} = 0$ are applied on S. This $\boldsymbol{\omega}^n$ can be seen as a vorticity field produced as an effect of the slip induced in the previous step, somewhat like the vorticity creation methods (24, 25, 26).

5. Using the above calculated vorticity field $\boldsymbol{\omega}^n$ a fixed velocity field $\boldsymbol{u}^n$ is computed as a solution to the KLE using both constraints and the two boundary conditions i.e. the no normal flow and the no slip condition.

For the velocity boundary condition on the far field external boundary S_∞, the corresponding Dirichlet conditions are applied.

The above algorithm clearly shows the vorticity in time and velocity in space approach of KLE. The momentum equation is solved in step 1 itself. Step 2-5 consist of solving the KLE for each time step to get the spatial distribution of velocity, compatible with both the vorticity distribution as well as the velocity boundary conditions. Setting vorticity equal to zero at boundary is consistent with the free slip boundary condition for velocity in step 2 and finally as in vorticity creation methods, the no slip condition of step 4 gives the final vorticity field in response to the slip induced in step 3. It can be seen that the two solutions of KLE , each with a different set of boundary conditions take care of the vorticity boundary conditions also. These two integral projections on the velocity field ensure a vorticity compatible with the velocity boundary conditions in each time step.

This decoupling between the vorticity evolution and the solution to get velocity distribution along with the compatible vorticity distribution, makes it possible to solve problems with different constitutive relations using this method since the physics involved in any such relation is independent of the spatial solution of KLE. It also becomes much simpler to implement the variational formulation since the PDE system now does not depend either on time or the constitutive relations, but are simply a set of kinematic equations. Since this method is integral and not limited to just the boundary data for calculating boundary vorticity, it does seem to have a somewhat better physical interpretation then the other vorticity generation methods (10).

3.2.3 Variational formulation of KLE

A variational form of (3.17) can be written using the Galerkin method (21) as follows,

$$\int_\Omega (\boldsymbol{\nabla} \cdot \boldsymbol{\nabla} \mathbf{u}) \cdot \boldsymbol{\delta u} \, \mathrm{d}\Omega = - \int_\Omega (\boldsymbol{\nabla} \times \boldsymbol{\omega}) \cdot \boldsymbol{\delta u} \, \mathrm{d}\Omega, \tag{3.24}$$

where $\boldsymbol{\delta u}$ is a virtual, arbitrary velocity field on Ω that is set to zero where ever Dirichlet conditions are applied. The next step would be to integrate the left hand side of (3.24) by parts and using the divergence theorem to get, $\boldsymbol{\delta u}$ vanishes on S_∞,

$$\int_\Omega \boldsymbol{\nabla} \mathbf{u} : \boldsymbol{\nabla} \boldsymbol{\delta u} \, \mathrm{d}\Omega - \int_S \boldsymbol{n} \cdot \boldsymbol{\nabla} \mathbf{u} \cdot \boldsymbol{\delta u} \, \mathrm{d}S = \int_\Omega (\boldsymbol{\nabla} \times \boldsymbol{\omega}) \cdot \boldsymbol{\delta u} \, \mathrm{d}\Omega. \tag{3.25}$$

The no slip (as well as the free slip) and no normal flow boundary conditions ensure that $\boldsymbol{\delta u} = 0$, thereby reducing (3.25) to:

$$\int_\Omega \boldsymbol{\nabla} \mathbf{u} : \boldsymbol{\nabla} \boldsymbol{\delta u} \, \mathrm{d}\Omega = \int_\Omega (\boldsymbol{\nabla} \times \boldsymbol{\omega}) \cdot \boldsymbol{\delta u} \, \mathrm{d}\Omega. \tag{3.26}$$

The Laplacian operator also has an equivalent minimization formulation which gives for the variational form of KLE the following functional,

$$\Pi = \int_\Omega \frac{1}{2} \boldsymbol{\nabla} \mathbf{u} : \boldsymbol{\nabla} \mathbf{u} \, \mathrm{d}\Omega - \int_\Omega (\boldsymbol{\nabla} \times \boldsymbol{\omega}) \cdot \mathbf{u} \, \mathrm{d}\Omega. \tag{3.27}$$

To impose the constraints (3.18) and (3.19) the penalty method was used over other possible schemes. A brief on why it is preferred over other more rigorous alternatives can be found in (2). The penalty terms according to the two constraints given by (3.18) and(3.19) are added to (3.27) giving the modified functional $\tilde{\Pi}$ as,

$$\tilde{\Pi} = \Pi + \int_\Omega \frac{\alpha_{\mathcal{D}}}{2} (\boldsymbol{\nabla} \cdot \mathbf{u})^2 + \frac{\alpha_\omega}{2} (\boldsymbol{\nabla} \times \mathbf{u} - \boldsymbol{\omega}) \cdot (\boldsymbol{\nabla} \times \mathbf{u} - \boldsymbol{\omega}) \, \mathrm{d}\Omega \tag{3.28}$$

Where the penalty constants are given by α_ω and $\alpha_\mathcal{D}$ The stationary of $\tilde{\Pi}$ with respect to $\mathbf{u}$ can be written as,

$$\begin{aligned} \delta\tilde{\Pi} \;\; &= \int_\Omega \boldsymbol{\nabla}\mathbf{u} : \boldsymbol{\nabla}\boldsymbol{\delta u} - (\boldsymbol{\nabla}\times\boldsymbol{\omega})\cdot\boldsymbol{\delta u} + \alpha_\mathcal{D}(\boldsymbol{\nabla}\cdot\mathbf{u})(\boldsymbol{\nabla}\cdot\boldsymbol{\delta u}) \\ &\quad + \alpha_\omega(\boldsymbol{\nabla}\times\mathbf{u}-\boldsymbol{\omega})\cdot(\boldsymbol{\nabla}\times\boldsymbol{\delta u})\,\mathrm{d}\Omega = 0. \end{aligned} \tag{3.29}$$

Reordering the above equation gives,

$$\begin{aligned} &\int_\Omega \boldsymbol{\nabla}\mathbf{u} : \boldsymbol{\nabla}\boldsymbol{\delta u} + \alpha_\mathcal{D}(\boldsymbol{\nabla}\cdot\mathbf{u})(\boldsymbol{\nabla}\cdot\boldsymbol{\delta u}) + \alpha_\omega(\boldsymbol{\nabla}\times\mathbf{u})\cdot(\boldsymbol{\nabla}\times\boldsymbol{\delta u})\,\mathrm{d}\Omega = \\ &\int_\Omega (\boldsymbol{\nabla}\times\boldsymbol{\omega})\cdot\boldsymbol{\delta u} + \alpha_\omega\boldsymbol{\omega}\cdot(\boldsymbol{\nabla}\times\boldsymbol{\delta u})\,\mathrm{d}\Omega, \end{aligned} \tag{3.30}$$

(3.30) gives the variational formulation for KLE for incompressible flow, with (3.18) and (3.19) as the constraints. As mentioned before, this variational form can be solved by a spatial discretization using finite elements or spectral methods. This thesis deals with a spectral element approach which will be introduced in the next chapter.

Even though in previous paragraphs the KLE was referred to as a "vorticity-in-time/velocity-in-space *split* approach", this is more a general description of its time-space/vorticity−velocity uncoupled nature than a strict definition of its algorithmic structure. Strictly speaking, time-marching *splitting* or *fractional-step* methods replace simultaneous processes by sequential steps as a means to increase efficiency (27). Split may be by dimensions (e.g. a three-dimensional process split into three one-dimensional substeps), or by physics (e.g. advection on one fractional step, pressure adjustment on another, and diffusion on a third). For the hydrodynamic equations, the advantage of splitting-by-process is that the nonlinear advection process can be treated by a different algorithm than pressure adjustment, which in turn can be different from diffusion, the latter two involve a linear solution each. The advective step is usually advanced explicitly and the adjustment of fields, is integrated implicitly. A typical example of this technique is the very successful AB3CN (third-order Adams-Bashforth/Crank-Nicholson) three-step scheme (see (28, 29), among others). Besides its advantages, splitting also has some drawbacks, mostly related with consistency and the treatment of boundary conditions (see (27), Sec. 13.1–13.4). The choice of appropriate boundary conditions is quite important in minimizing the splitting error, as shown by Karniadakis *et al.* (30), where high-order pressure boundary conditions are found to be the key to the time accuracy of the splitting scheme.

Contrarily, there is no splitting whatsoever in the KLE method. All terms in the physical problem are solved simultaneously during time integration of the vorticity field, and all the spatial components of the velocity are solved together by the KLE. Since the KLE is an entirely Kinematic equation with the entire physics concerned with any of the nonlinearities and complex constitutive relations limited to the time integration schemes, it favors modeling complex flow problems like non-Newtonian flows or turbulent flows etc. Since it is also a universal vector equation, basically any field represented by this relation can be solved for as long as the divergence and curl of that field has a solvable transport equation. Also, since the vector relation is independent of the time iteration process, the KLE can be coupled with other processes like heat transfer or chemical processes by simply adding the required relation to the existing ODE system. So basically just the source term to the KLE is changed to solve different physical problems.

Chapter 4

Numerical implementation of the KLE

4.1 Introduction

This chapter deals with the numerical implementation of the KLE using a spectral element method to discretize Equation 3.30 in space along with a predictor-corrector time integration scheme. The spatial discretization scheme used here is a two dimensional isoparametric spectral element with a high order Lagrangian polynomial to interpolate solutions within the element. An isoparametric element was chosen because of the complex differential equations involved as also the complexity of the intended surfaces to be modeled. The main advantage of this type of an element is that the integration has to be performed over the “parent” element which represents a normalized domain in terms of a local co-ordinate

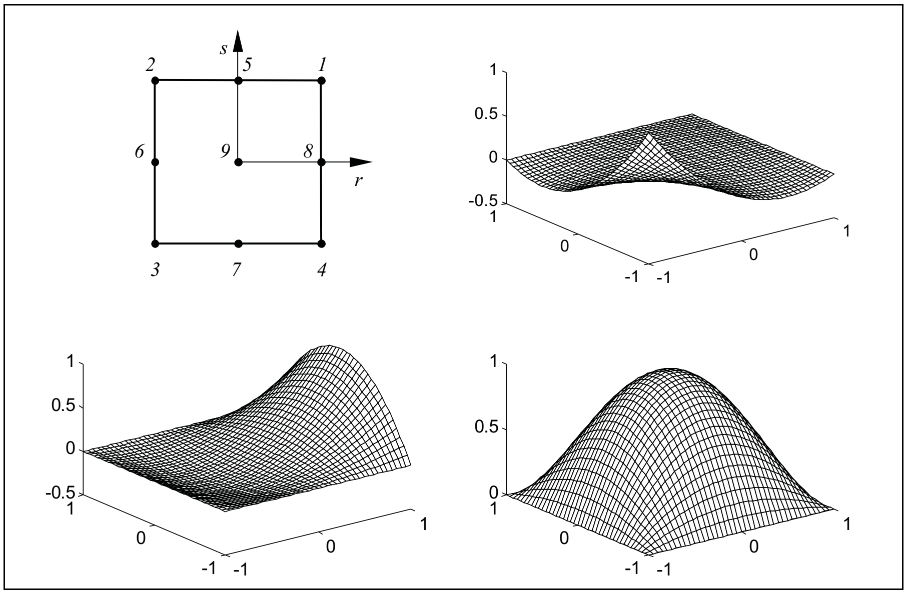


Figure 4.1: *A two-dimensional nine node isoparametric element in its natural coordinate system along with a graphical representation of three of its nine interpolation functions i.e nodes 3, 8 and 9.*

system varying between +1 and -1. This makes it easier to implement any numerical technique. An isoparametric element uses the same Lagarange polynomial (shape functions) to interpolate the unknown variable within the elements as the ones used to map the global to local coordinates. Figure 4.1 shows an example of biquadratic interpolation functions of a nine-node isoparametric quadrilateral element on its natural system of coordinates, i.e. (r,s). A quadrilateral element was chosen because of its high convergence rate and its ability to reduce the skin error on curvilinear bondaries when compared to linear elememts. Nevertheless, other discretization techniques may be applied to the implementation of the KLE method. For our experimentation, the spectral element method is used for the KLE which will be discussed in the following section.

4.2 The Spectral-element method for KLE

The general trend in finite element methods had been to use Lagrange polynomials of a particular order as shape functions. To improve the accuracy, the number of elements was increased. This is called an h-type finite element method. For sufficiently smooth problems these methods converge at an algebraic rate with the error being proportional to $\frac{1}{N^{p+1}}$ where N is the number of degrees of freedom and p is the order of the Lagrange polynomial (31). Another approach would be to follow the h-type discretization with an increase in the order of the interpolating polynomial within each element to improve accuracy. Again, for sufficiently smooth solutions this would give an exponential convergence rate (31). These are called the p-type methods. The spectral method is a particular implementation of the p-version of an hp finite element method.

The spectral-element method was introduced some twenty years back (32, 33). It's main purpose was to tackle complicated domains which the spectral methods were not able to handle. As shown in (34) this $h - p$ type of method was capable of local refinements, and where thus good for complex geometries and at the same time preserved the high convergene rates seen in spectral methods. Owing to the $h - p$ discretization, a high accuracy can be achieved for less number of nodes, amking it a highly *memory-minimizing* method (27).

Mostly the Legendre or Chebyshev polynomials are used by the spectral element methods in order to come up with suitable basis functions. The same points are used for the interpolation functions as are for the numerical integration within the elements. These collocation points are called the Gauss-Legendre-Lobatto (GLL) quadrature points. This leads to diagonal mass matrices making the system more efficient.

As mentioned at the beginning of this chapter, in this particular analysis an isoparametric element is used with the Lagrangian polynomials as interpolating functions for the solution. The variational formulation for the KLE using the Galerkin method, shown in the previous chapter, is used to solve for the velocity field at the nodal points. The nodes are at the GLL points.For higher order elements using the GLL points in place of the regular equispaced points is more economical (35). Giraldo (36), through experiments has shown that for higher order interpolating polynomials (in excess of 4) the solution results for Gauss Legendre and GLL quadrature are comparable.

As shown in (21) the finite-element discretization of the velocity field and its gradient can be represented as,

$$\mathbf{u} = \begin{bmatrix} u_x \\ u_y \end{bmatrix} = \mathbf{H} \cdot \mathbf{U}^e, \qquad \boldsymbol{\nabla}\mathbf{u} = \begin{bmatrix} \frac{\partial u_x}{\partial x} \\ \frac{\partial u_x}{\partial y} \\ \frac{\partial u_y}{\partial x} \\ \frac{\partial u_y}{\partial y} \end{bmatrix} = \mathbf{B} \cdot \mathbf{U}^e, \tag{4.1}$$

where $\mathbf{H}$ is the interpolation-function, $\mathbf{B}$ it's derivative and $\mathbf{U}^e$ is the array of discretized velocity at nodes of each element,

$$\mathbf{U}^e = \begin{bmatrix} u_x^1 \\ u_y^1 \\ u_x^2 \\ \vdots \\ u_x^{NGL^2} \\ u_y^{NGL^2} \end{bmatrix}, \qquad \mathbf{H} = \begin{bmatrix} h^1 & 0 & h^2 & \cdots & h^{NGL^2} & 0 \\ 0 & h^1 & 0 & \cdots & 0 & h^{NGL^2} \end{bmatrix}, \tag{4.2}$$

$$\mathbf{B} = \begin{bmatrix} \frac{\partial h^1}{\partial x} & 0 & \frac{\partial h^2}{\partial x} & \cdots & \frac{\partial h^{NGL^2}}{\partial x} & 0 \\ \frac{\partial h^1}{\partial y} & 0 & \frac{\partial h^2}{\partial y} & \cdots & \frac{\partial h^{NGL^2}}{\partial y} & 0 \\ 0 & \frac{\partial h^1}{\partial x} & 0 & \cdots & 0 & \frac{\partial h^{NGL^2}}{\partial x} \\ 0 & \frac{\partial h^1}{\partial y} & 0 & \cdots & 0 & \frac{\partial h^{NGL^2}}{\partial y} \end{bmatrix}, \tag{4.3}$$

where $NGL = p + 1$ is the number of nodes of the Gauss-Lobatto interpolation.

The elements of (4.3) are given by,

$$\begin{bmatrix} \frac{\partial h^k}{\partial x} \\ \frac{\partial h^k}{\partial y} \end{bmatrix} = \mathbf{J}^{-1} \cdot \begin{bmatrix} \frac{\partial h^k}{\partial r} \\ \frac{\partial h^k}{\partial s} \end{bmatrix}, \qquad k = 1, \ldots, NGL^2, \tag{4.4}$$

where **J** is the Jacobian operator which relates the natural to the local coordinate derivates,

$$\mathbf{J} = \begin{bmatrix} \sum_{k=1}^{NGL^2} \frac{\partial h^k}{\partial r} x^k & \sum_{k=1}^{NGL^2} \frac{\partial h^k}{\partial r} y^k \\ \sum_{k=1}^{NGL^2} \frac{\partial h^k}{\partial s} x^k & \sum_{k=1}^{NGL^2} \frac{\partial h^k}{\partial s} y^k \end{bmatrix}, \tag{4.5}$$

and (x^k, y^k) the local coordinates of the nodes. The divergence of the velocity field is given by –

$$\nabla \cdot \mathbf{u} = \mathbf{m} \cdot \mathbf{B} \cdot \mathbf{U}^e, \qquad \mathbf{m} = \begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}, \tag{4.6}$$

and the curl of the velocity ω_z (the only component of the curl not equal to zero), is obtained as,

$$\nabla \times \mathbf{u} = \mathbf{r} \cdot \mathbf{B} \cdot \mathbf{U}^e, \qquad \mathbf{r} = \begin{bmatrix} 0 & -1 & 1 & 0 \end{bmatrix}. \tag{4.7}$$

Similarly for vorticity,

$$\boldsymbol{\omega} = \mathbf{H}_\omega \cdot \boldsymbol{\omega^e}, \qquad \nabla \times \boldsymbol{\omega} = \begin{bmatrix} \frac{\partial \omega}{\partial y} \\ -\frac{\partial \omega}{\partial x} \end{bmatrix} = \mathbf{B}_\omega \cdot \boldsymbol{\omega^e}, \tag{4.8}$$

where $\boldsymbol{\omega^e}$ gives the vorticity values at nodes of each element calculated by integrating in time the vorticity transport equation, and as shown for velocity $\mathbf{H}_\omega$ and $\mathbf{B}_\omega$ are the vorticity interpolation-functions snd their derivatives respectively,

$$\boldsymbol{\omega^e} = \begin{bmatrix} \omega^1 \\ \omega^2 \\ \vdots \\ \omega^{NGL^2} \end{bmatrix}, \qquad \mathbf{H}_\omega = \begin{bmatrix} h^1 & h^2 & \dots & h^{NGL^2} \end{bmatrix}, \tag{4.9}$$

$$\mathbf{B}_\omega = \begin{bmatrix} \frac{\partial h^1}{\partial y} & \frac{\partial h^2}{\partial y} & \dots & \frac{\partial h^{NGL^2}}{\partial y} \\ -\frac{\partial h^1}{\partial x} & -\frac{\partial h^2}{\partial x} & \dots & -\frac{\partial h^{NGL^2}}{\partial x} \end{bmatrix}. \tag{4.10}$$

For the finite element analysis, first each element can be thought of as a discretized subdomain (Ω^e). Therefore if equation (3.30) is considered at each Ω^e and the corresponding discretized values of the velocity and vorticity fields are substituted for, the following equation is obtained,

$$\boldsymbol{\delta}\mathbf{U}^{eT} \cdot \underbrace{(\mathbf{K}_L^e + \mathbf{K}_{\mathcal{D}}^e + \mathbf{K}_\omega^e)}_{\mathbf{K}^e} \cdot \mathbf{U}^e = \boldsymbol{\delta}\mathbf{U}^{eT} \cdot \underbrace{(\mathbf{R}_L^e + \mathbf{R}_\omega^e)}_{\mathbf{R}^e} \cdot \boldsymbol{\omega^e}, \tag{4.11}$$

where

$$\begin{aligned}
\mathbf{K}_L^e &= \int_{\Omega^e} \mathbf{B}^T \cdot \mathbf{B} \, \mathrm{d}\Omega = \int_{-1}^{1}\int_{-1}^{1} \mathbf{B}^T \cdot \mathbf{B} \, |\mathbf{J}| \, \mathrm{d}r\mathrm{d}s, \\
\mathbf{K}_{\mathcal{D}}^e &= \int_{-1}^{1}\int_{-1}^{1} \alpha_{\mathcal{D}} \, \mathbf{B}^T \cdot \mathbf{m}^T \cdot \mathbf{m} \cdot \mathbf{B} \, |\mathbf{J}| \, \mathrm{d}r\mathrm{d}s, \\
\mathbf{K}_\omega^e &= \int_{-1}^{1}\int_{-1}^{1} \alpha_\omega \, \mathbf{B}^T \cdot \mathbf{r}^T \cdot \mathbf{r} \cdot \mathbf{B} \, |\mathbf{J}| \, \mathrm{d}r\mathrm{d}s, \\
\mathbf{R}_L^e &= \int_{-1}^{1}\int_{-1}^{1} \mathbf{H}^T \cdot \mathbf{B}_\omega \, |\mathbf{J}| \, \mathrm{d}r\mathrm{d}s, \\
\mathbf{R}_\omega^e &= \int_{-1}^{1}\int_{-1}^{1} \alpha_\omega \, \mathbf{B}^T \cdot \mathbf{r}^T \cdot \mathbf{H}_\omega \, |\mathbf{J}| \, \mathrm{d}r\mathrm{d}s,
\end{aligned}$$

$\boldsymbol{\delta U^e}$ gives the array of values at nodes of each element for the arbitrary $\boldsymbol{\delta u}$.

The arrays and the matrices of (4.11) are assembled for each element to give the following global system,

$$\mathbf{K} \cdot \mathbf{U}^e = \mathbf{R} \cdot \boldsymbol{\omega}. \tag{4.12}$$

As mentioned earlier a quadrilateral element has high convergence rate and reduces the skin error on circular boundaries. At the same time triangular elements find it easier to change mesh density in a more smooth and gradual manner and are also more suitable for unstructured meshing (21, 22). Thus the domain was first dicretized using triangular elements which was subsequently converted to a quadrilateral mesh, by dividing each tri-

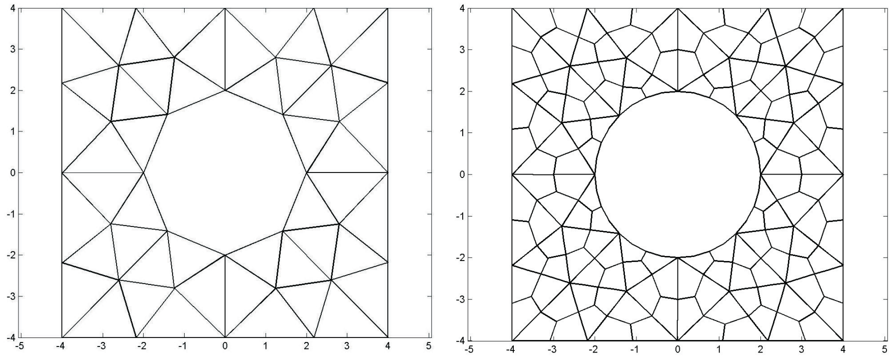


Figure 4.2: *A tri-quadrilateral finite element mesh derived from an unstructured triangular mesh.*

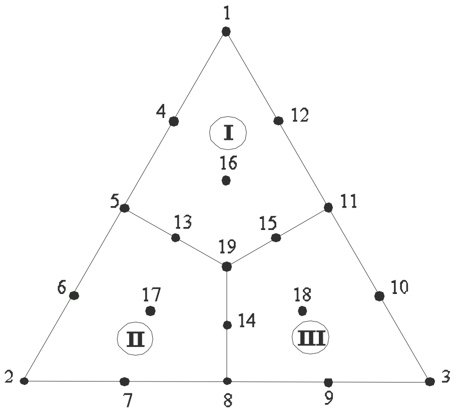


Figure 4.3: *The internal topology of a tri-quadrilateral element. Quadrilateral elements (I)–(III) are the nine-node isoparametric elements. 1–19 is the in-triangle global numbering of the nodes.*

angle into three quadrilaterals. This can be seen in figure (4.2). An important advantage of this "tri-quadrilateralization" is a process called static condensation of internal nodes. These nodes lying inside the triangle though used for elemental integration are not used while assembling the final global structure matrices. These are later recovered from the values obtained by the solution of the external nodes. To establish an equation for this condensation, as shown in (37), the system $\mathbf{K}^e \cdot \mathbf{U}^e = \mathbf{R}^e \cdot \boldsymbol{\omega}^{\boldsymbol{e}}$ are partitioned as,

$$\begin{bmatrix} \mathbf{K}^e_{aa} & \mathbf{K}^e_{ab} \\ \\ \mathbf{K}^e_{ba} & \mathbf{K}^e_{bb} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{U}^e_a \\ \\ \mathbf{U}^e_b \end{bmatrix} = \begin{bmatrix} \mathbf{R}^e_a \\ \\ \mathbf{R}^e_b \end{bmatrix} \cdot \boldsymbol{\omega}^{\boldsymbol{e}} \tag{4.13}$$

where a is for the degrees of freedom 1–24 of the velocity at nodes 1–12 and b is for the degrees of freedom 25–38 of the velocity at nodes 13–19. The second row of the above equation gives,

$$\mathbf{U}^e_b = \underbrace{(\mathbf{K}^e_{bb})^{-1} \cdot \mathbf{R}^e_b}_{\mathbf{R}^e_b} \cdot \boldsymbol{\omega}^{\boldsymbol{e}} - \underbrace{(\mathbf{K}^e_{bb})^{-1} \cdot \mathbf{K}^e_{ba}}_{\mathbf{K}^e_{ba}} \cdot \mathbf{U}^e_a, \tag{4.14}$$

substituting this into the first row of (4.13) and reordering,

$$\begin{aligned} &\underbrace{\left(\mathbf{K}^e_{aa} - \mathbf{K}^e_{ab} \cdot (\mathbf{K}^e_{bb})^{-1} \cdot \mathbf{K}^e_{ba}\right)}_{\mathbf{K}^e} \cdot \mathbf{U}^e_a = \\ &\underbrace{\left(\mathbf{R}^e_a - \mathbf{K}^e_{ab} \cdot (\mathbf{K}^e_{bb})^{-1} \cdot \mathbf{R}^e_b\right)}_{\mathbf{R}^e} \cdot \boldsymbol{\omega}^{\boldsymbol{e}}, \end{aligned} \tag{4.15}$$

This is the condensed form. Assembling the arrays and matrices of (4.14) and (4.15) gives the following global condensed system,

$$\mathbf{K} \cdot \mathbf{U}_a = \mathbf{R} \cdot \boldsymbol{\omega}, \tag{4.16}$$

$$\mathbf{U}_b = \mathbf{R}_b \cdot \boldsymbol{\omega} - \mathbf{K}_{ba} \cdot \mathbf{U}_a, \tag{4.17}$$

The static condensation process leads to almost a 40% reduction in the size of the global system to be solved and also leads to a better condition number of the global structure matrices. This is in accordance with the Schur complement method, where the condensed matrix forms the Schur complement for the in-triangle nodes of the original system. As mentioned earlier none of the structure matrices depend on the physics of the problem, in this case the vorticity or the time, they can be calculated and stored and used over and over again as required. Since $\mathbf{KK}$ is positive definite and symmetric it is factorized by the Cholesky decomposition method and the factor (triangular) so obtained can be used to solve for $\mathbf{U}_a$. One problem with the spectral element methods is the loss of the exponential convergence and also the higher accuracy in case of singularities like shock in compressible flow (38). This is often seen while interpolating non-smooth functions (abrupt changes in boundaries and forces etc.) using high order polynomials.

Evaluating the right-hand side of the vorticity transport equation

As shown in (21) for the two-dimensional implementation of the time-integration procedure, the vorticity transport Equation (3.16) can be written in a more convenient way as follows,

$$\frac{\partial \boldsymbol{\omega}}{\partial t} = \boldsymbol{F}\left(\boldsymbol{\omega}, t\right) = \boldsymbol{\nabla} \times \left(\nu\, \boldsymbol{\nabla} \cdot \boldsymbol{\nabla}\mathbf{u} - \mathbf{u} \cdot \boldsymbol{\nabla}\mathbf{u}\right). \tag{4.18}$$

The RHS of (4.18)is solved for by carrying out the respective curl, divergence and gradient operations on the discretized counterpart of u as found by the KLE algorithm previously explained. Since, for the spectral-element case, the Gauss-Lobatto points are the same as the nodes,therefore, for those lying on the inter-element boundaries, an average of the values from elements sharing those boundaries can be used.

The weight of each Gaussian point depends on the mesh geometry and is calculated during assembly. So the arrays for the differential operators are assembled at the same time as the Finite element matrices. Those arrays perform the differential operations on any vector or tensor field, as a dot product with the corresponding discrete solution of that field. For instance, the discrete form of the curl of the velocity field $\boldsymbol{\nabla} \times \mathbf{u}$ is given by the dot product $\boldsymbol{C_{url}} \cdot \mathbf{U}$. Thus, the discrete form of (4.18) is written as,

$$\mathbf{F}(\boldsymbol{\omega}, t) = \mathbf{C}_{url} \cdot (\nu\, \mathbf{D}_{iv} - \mathbf{U}_{adv}) \cdot \mathbf{G}_{rad} \cdot \mathbf{U}, \tag{4.19}$$

where $\mathbf{G}_{rad}$ gives the gradient, $\mathbf{D}_{iv}$ the divergence of $\mathbf{G}_{rad}$, $\mathbf{C}_{url}$ the curl vector and $\mathbf{U}_{adv}$ is obtained by reordering $\mathbf{U}$ to perfrom the dot product $\mathbf{u} \cdot \boldsymbol{\nabla}\mathbf{u}$ in the advective term.

Since none of them depend on the vorticity field or time, they can be, as with the structure matrices, calculated and stored for further use. For the time integration Adams-Bashforth-Moulton predictor-corrector (ABM-PECE) solver with multivariable order and adaptive stepsize is used and the results show that it is efficient enough to pursue further research (2) as discussed in the following sections.

4.3 The Adaptive time step solver for KLE

With the evolution of the various numerical schemes for complex transport problems or multiphysics problems with combined flows, there is increasing demand for improved algorithms like self adaptive time step solvers and other advancements like self adaptive grid refinement and coarsening. As the name suggests, both the adaptive time step solver and grid adaptation techniques receive the response from their calculated parameters and then the initial or intermediate values modify themselves to the next refined value. The adaptive stepsize techniques are employed to control the accuracy of the simulations and to enhance their efficiency to provide stable steady state or transient solutions. The adaptive time-step algorithms usually use the values of approximate local truncation error or some of them are even based on some kind of "Thumb Rules". In some of the standard algorithms, user is required to specify the accuracy requirement on local truncation error which is compared with computed values of the same which should be within accepted accuracy or tolerance range (39).

As mentioned in previous section, Adams-Bashforth-Moulton predictor-corrector (ABM-PECE) solver with adaptive stepsize control has been used with KLE successfully. The solver ODE113 in MATLAB which is based on ABM-PECE, is used for the experiments for the determination of CFL condition.

4.3.1 ODE113

The MATLAB code ODE113 was derived from the well-known code STEP (40). For more explanation about code STEP, original research paper by L.F. Shampine and M.K. Gordon can be referred. ODE113 is the modification to the code STEP and it is a variable stepsize, variable order and compiled using ABM PECE method with local extrapolation. The modified divided difference form of the interpolating polynomials is used in this method. It means when step size is constant, they are just normal backward differences. The acceptance or rejection of the time step is largely dependent upon the local error estimate which is nothing but the difference between the two corrector formulae of two consecutive orders. This local error estimate is calculated and compared using the inbuilt checking criteria, before the final evaluation in the $\mathrm{P}_k\mathrm{EC}_{k+1}\mathrm{E}$ method. Thus, a rejected time step only requires only one function evaluation. In this code, the order from 1 to 12 can be altered (41).

The rationale behind using ODE113 for the experimentation lies in its effectiveness with KLE. A few other reasons are stated as follows:

1. Only two function evaluations of right hand side (RHS) are required per accepted time step as compared to ODE45 which requires six function evaluations per integration step.
2. Rejected time step only requires one function evaluation of right hand side (RHS).
3. It is good for stringent error tolerances.
4. It is also convenient for computationally intensive problems like KLE.

Chapter 5

Experiment and analysis

5.1 Introduction

This chapter deals with an experiment performed with the aim of determining Courant-Friedrichs-Lewy condition (CFL condition) which defines the time-step restraint for the given numerical scheme. In mathematical terms, it only provides a necessary condition for convergence while solving the series of differential equations. But, it is not a sufficient condition; thus even after CFL condition is satisfied, the numerical solution still can be unstable. This condition first introduced in 1928 by Richard Courant, Kurt Friedrichs, and Hans Lewy for the case of the difference approximations in terms of the concept the domain of dependence. According to an early conceptualization (42) by the above mentioned trio, the convergence is attainable only if the ratio of mesh widths in different dimensions fulfills certain inequalities which then themselves depend on the position of the characteristics relative to the mesh. In simpler words, the general idea behind the CFL condition is the solution of numerical scheme should not be independent of any of the data that determines the solution of the given differential equation, unless the data that is being omitted has a negligible effect. Along with the advancement of numerical methods, the CFL condition has also evolved greatly in terms of determining stability and providing the idea of the critical characteristics required for stability. An example of the computation of CFL condition for a very basic numerical scheme is explained in the following section.

5.2 CFL condition

For most of the numerical methods (43), in order to optimize the CFL condition, it is possible to define all the parameters of the method while defining it. Before starting any experiments with KLE to establish its CFL condition, it is important to understand how method defining factors for any numerical scheme contribute to the CFL condition. Consider this simple initial value problem,

$$\frac{\mathrm{d}\,u}{\mathrm{d}\,t} + c\frac{\mathrm{d}\,u}{\mathrm{d}\,x} = 0 \tag{5.1}$$

with $u(x,0) = f(x)$

where, $c > 0$ is a constant

For the numerical advances in space and time in the (x, t) plane small values of Δx and Δt are used to lay out the grid.

After using backward difference scheme

$$\frac{\mathrm{d}\,u}{\mathrm{d}\,x}(x,t) = \frac{u_{(x,t)} - u_{(x-\Delta x,t)}}{\Delta x} + O(\Delta x)$$

and by using forward difference for Δt

$$\frac{\mathrm{d}\,u}{\mathrm{d}\,t}(x,t) = \frac{u_{(x,t+\Delta t)} - u_{(x,t)}}{\Delta t} + O(\Delta t)$$

Then, truncation error, $\sigma = O(\Delta t) + O(\Delta x)$. After neglecting truncation error, comparing with equation (5.1)

$$\frac{u_{(x,t+\Delta t)} - u_{(x,t)}}{\Delta t} + c\frac{u_{(x,t)} - u_{(x-\Delta x,t)}}{\Delta x} = 0$$

Sloving

$$u_{(x,t+\Delta t)} = \left(1 - \frac{c}{r}\right) u_{(x,t)} + \left(\frac{c}{r}\right) u_{(x-\Delta x,t)}$$

where, $r = \dfrac{\Delta x}{\Delta t}$ and $\dfrac{c}{r} \leq 1$ for the solution to exist.

$$c\Delta t \leq \Delta x$$

This condition is known as CFL condition. The ratio $\dfrac{c}{r}$ is called as the CFL number or Courant number. Here, it can be easily seen that for space discretization and time marching for the above method, Δx and Δt are need to be employed. Thus all the method defining factors are present in the inequality that depends upon the relation between those factors and the Courant number.

Although it is possible in some of the simpler methods to determine the CFL condition analytically; in the case of KLE, several factors limit the possibility of computing it analytically. The KLE algorithm is so complicated, and it uses the spectral elements for spatial discretization which adds to the complexity in determining the relation between the time step and the characteristics of the h-p type elements. All the terms in the physical problem are solved simultaneously during time integration of the vorticity field, and all the spatial components of the velocity are solved together by the KLE. With several numerical operations being involved in KLE, it is viable to look for some other way to achieve the given objective. Thus, the experiments have to be designed to find out CFL condition numerically. There are a number of factors which can be crucial in making the analysis of the results much easier such as the selection of the test problem, the type of the mesh, and the total time span. As mentioned in the previous chapter, this thesis is concerned with the spectral element implementation of the KLE; therefore both the number of elements h and the order of interpolation p refinement test results are equally important. The description of the experiment and the dependence of the time-step Δt on various factors are dealt with in subsequent sections.

5.3 The canonical test problem

For the experimentation, a well-defined test problem was required, so Stoke's problem of a semi-infinite region of stationary fluid bounded by an infinite horizontal flat plate was selected. This problem has become canonical for the vorticity creation in the vortex dominant flows and hence, was ideal for our tests. In that problem, a flat plate which is at $y = 0$, which is suddenly given a velocity U in the horizontal plane and then continued at the same speed. The exact analytical solution to this problem is known (see (23), Sec.4.3, among others) which is useful in order to determine the induced error in the computational solution. The velocity field described in a frame of reference fixed to a plate moving in the negetive x direction is

$$u(y,t) = U \operatorname{erf}\left(\frac{y}{\sqrt{4\,\nu\,t}}\right), \tag{5.2}$$

Here, erf is the error function and y is the vertical coordinate. If the velocity u and y is normalized as u/U and y/Y (5.2), also the term $\tau = \sqrt{4\,\nu\,t}/Y$

$$\frac{u}{U} = \operatorname{erf}\left(\frac{y/Y}{\tau}\right), \tag{5.3}$$

where Y is the height of the test mesh i.e. total domain size and quantitatively Y is taken as unity for the experimentation. The normalized vorticity distribution for this incompressible flow is given by the Gaussian function as−

$$\frac{\omega}{U/Y} = \frac{2}{\tau\,\sqrt{\pi}}\, e^{-\left(\frac{y/Y}{\tau}\right)^2}. \tag{5.4}$$

As mentioned above, this problem resembles to the fundamental process of ($\boldsymbol{\omega}$, $\boldsymbol{v}$) methods, i.e. the vorticity generation at a solid surface due to the induced slip, and its further propagation to the body of the fluid. The Gaussian and the error function of the spatial coordinates give the analytical solution for the velocity and vorticity field, respectively, for a specified time. For our experimentation, the time-dependent analysis of Stoke's problem was performed, but, the spatial analytical solution can be used to test several parameters for the spatial discretization of the KLE. Also, for the spatial discretization, the symmetric quadrilateral mesh elements are used to further simplify the process in the experimentation.

5.4 The experiment

This section deals with the methodology for the test experiment as well as the reasoning behind it. It then goes on to discuss the analysis of the results in order to come up with the limiting condition on the time-step for KLE. The first step was to decide on a variable which will give some measure of the spatial grid size. In h-p type spectral element, the nodes are not always equidistant. So a new variable in the form of the number of intervals in each dimension $N^{\star}$ is introduced, which is equal to the number of nodes minus one and represents the inverse of the average intermodal distance. As mentioned in the previous chapter, the solver ODE113 itself computes the best possible time-step for the KLE by analyzing the stability criteria. Thus in order to establish any relation between the spatial grid size and the chosen time-step, it is the most logical to use $N^{\star}$; as it accounts for both

h and p for our method. The next step was to solve KLE for many different values of h and p and compare the adapted time-steps with each other to foresee any possible connection with $N^{\star}$.

Here it is worthwhile to note that, when used inside the time-marching process of the vorticity-velocity scheme, the source term for the KLE solution at a given time is provided by a computation made by the ODE integrator from an approximation in weak form of the velocity field at the previous time step. This has the tendency to smooth out the shock introduced at the initial stage. Hence, forcing the theoretical vorticity distribution given by expression (5.4) at the initial stages as a source term for the KLE poses a very strict trial on the KLE solution. This sharp forcing is actually more challenging than KLE's normal operational requirements as the spatial counterpart in a vorticity-velocity scheme. So for the experimentation, the values of both h and p were tried over a range of values keeping one constant while varying the other. For the simplification of the input values and for better physical understanding of the spatial discretization, instead of interpolation order p, number of Gauss-Lobatto-Legendre node points i.e. NGL is used for the experimentation which is equal to $p+1$ in our type of mesh. The NGL values from 3 to 11 are used for conducting the experiment. As explained in the previous section, the normalized parameter τ is considered in the interpretation of the results instead of the actual time span t in seconds. The value of τ ranges from 0 to 1 and gives the idea of the flow progression phase for the given time period.

It is important to note here that as already mentioned in chapter 4, for polynomial orders $p \geq 4$ non-exact GLL integration is used i.e. for $NGL \geq 5$. For $NGL = 3$, the direct interpolation function can be used. It meant that only for $NGL4$, classical Gauss-Legendre integration is required for interpolation. Hence, it is decided that all the results associated with $NGL4$ will be omitted further on from all the experimental analysis.

5.5 Experimental results and Analysis

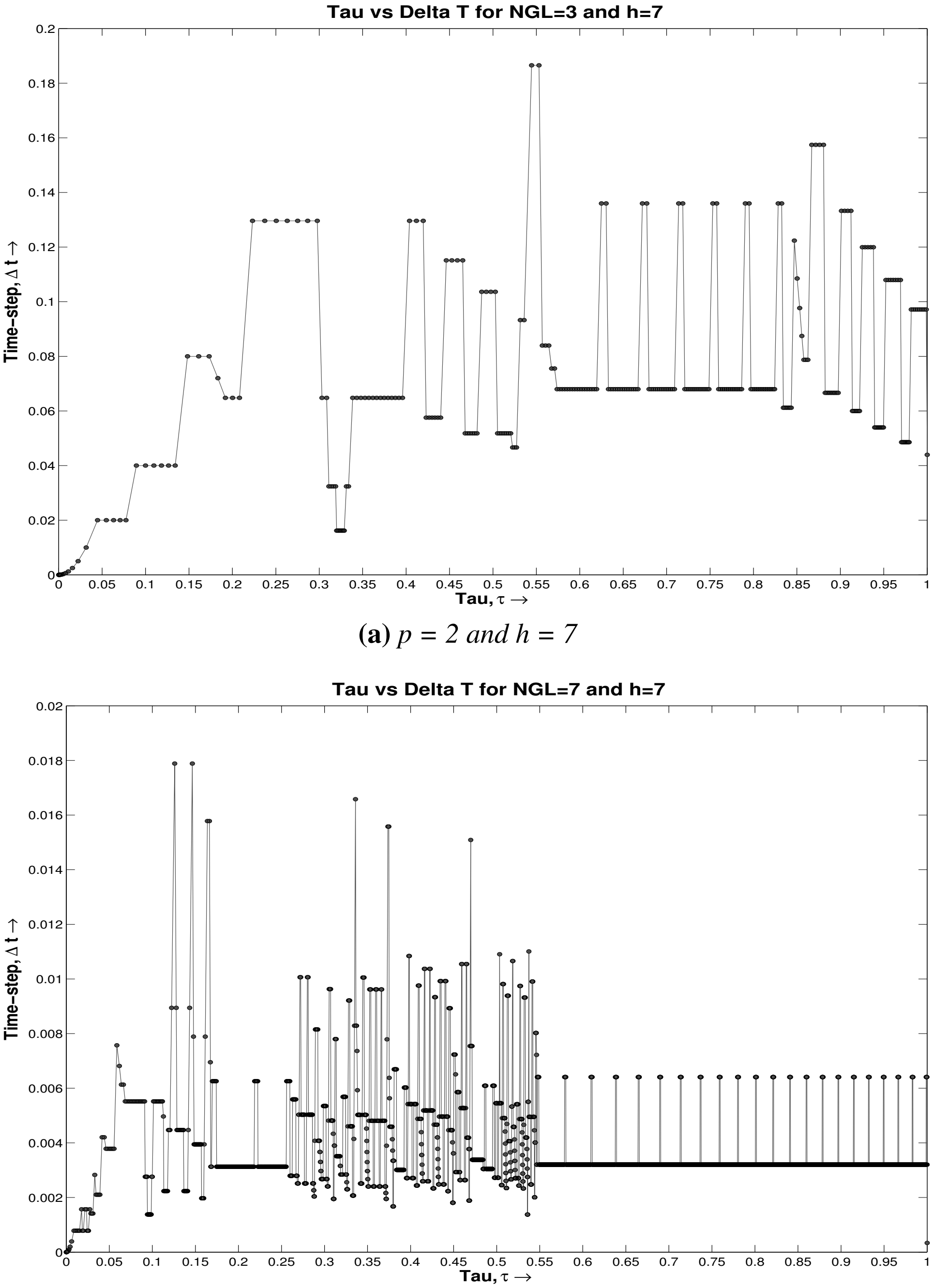
The analysis of the experiments to find out CFL condition consisted of a number of stages. In the first stage, all the adapted times steps by ODE113 that stored throughout the process of flow progression are compared with the normalized time τ. Then in the next stages, time-steps are compared with the spatial grid size parameters and the generalized common numerical relation is deduced.

5.5.1 Average time step

The purpose of this stage is to try to identify the nature of the adapted time steps by ODE113 for each and every value of NGL and h. The primary step is to plot the time steps against τ. Since the experimental results are taken for both an h-refinement as well as a p-refinement, varying one while keeping the other constant; it is important to identify if all the cases have similar nature for the time steps.

Figures 5.1 and 5.2 show plots for the chosen three random cases with NGL =3,7 and 11 with $h = 7$. These plots explain the general behavior of the ODE113 for the KLE. As it can be seen that the solver takes some time to warm up and reach to certain lower values of the Δt. Then as explained in previous chapter, it try to attain the higher values of Δt till it reaches the given accuracy limit and again the solver try to limit Δt to the corresponding stability requirement. It also keeps the adapted value of Δt for certain steps and then starts varying again with respect to the perturbations in the numerical solution. From the Figure 5.1(b) it can be seen that for some cases ODE113 smoothens with the development of the flow and stabilize the time-step size between two limits. It is important to understand that the curves in Figures 5.1(a) and 5.1(b), do not represent the error, but a natural process of selecting and checking an ideal time-step, which after initial shock of impulse remains around approximately constant value. The error systematically kept within the specified tolerance limits.

After observing all the cases, it is not farfetched to conclude that the solver fluctuates around a certain average value of the time step. The solver has also defined the upper and lower limit of the time steps for the given set of input values. In order to avoid the instability and to keep the required accuracy limit, the time step has to be in between those limits. Hence, it is safer to say if the mean of all the time steps is computed, that value will give the best possible value of the limiting constraint for Δt. But it is important to eliminate



(a) *p = 2 and h = 7*

(b) *p = 6 and h = 7*

Figure 5.1: *Plots of the non-dimensionalized time τ vs ODE113 selected time-steps Δt for some selective cases*

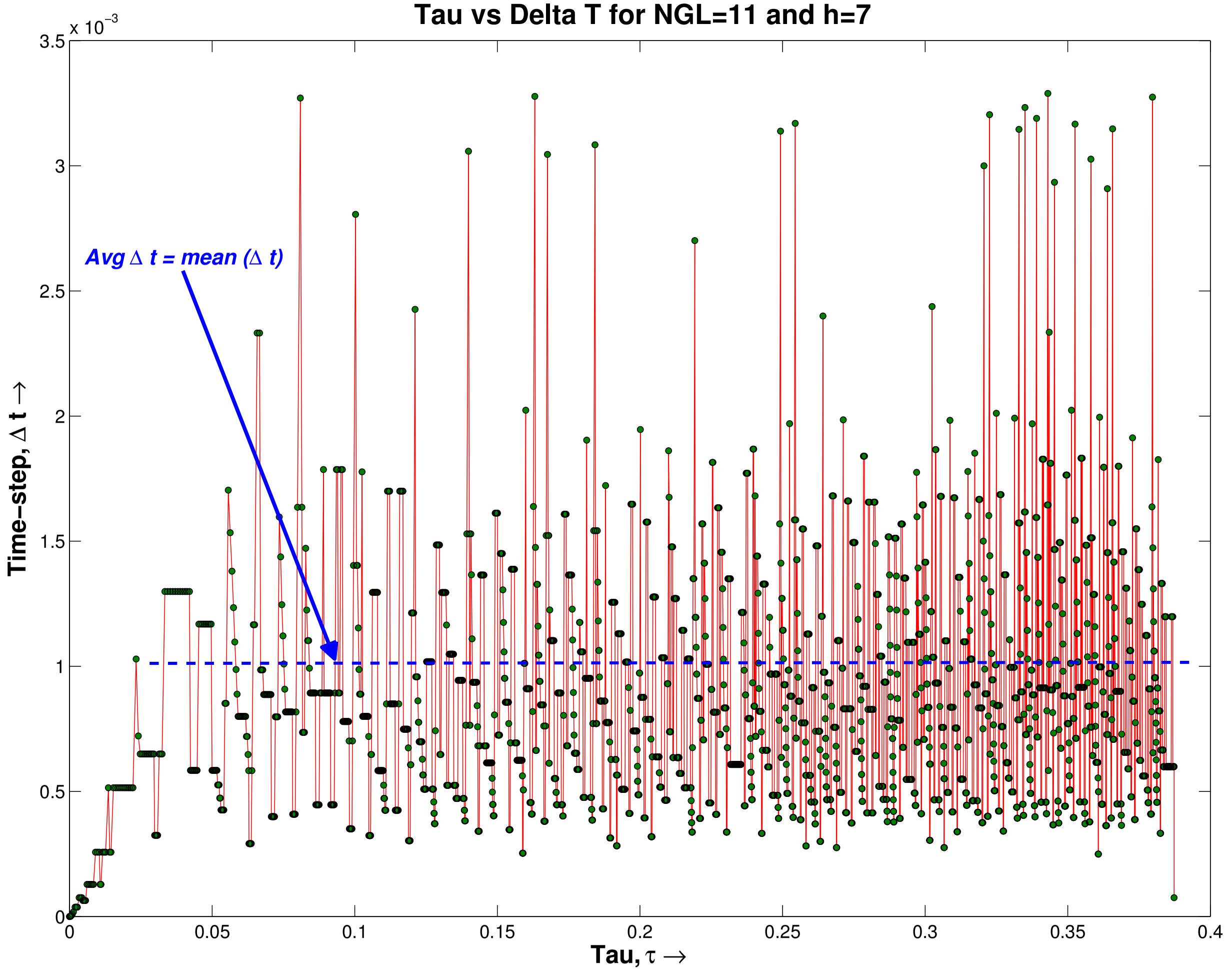


Figure 5.2: *The non-dimensionalized time τ vs ODE113 selected time-steps Δt for p = 10 and h = 7, showing computation of $Avg\Delta t$*

the initial values of Δt before the solver stabilizes at certain point of the time marching. It must be noted that the solver gets stabilized at different values of τ for different cases. Higher the node density or higher the value of $N^{\star}$, lower the value of τ at which the Δt starts getting stabilized. Hence, the average value of Δt is found out for all the cases by taking mean of all the values of time-steps from the stabilized region of τ.

5.5.2 The evaluation of CFL condition

The three plots shown here are the part of a series of plots for an interpolation order p ranging from 2 till order 10, but the number of elements are limited by the computational power available. So, instead of trying to vary h, the experiments are done by keeping limit on the value of $N^{\star}$ to ensure that they did not go beyond 120. So in addition to the reason already mentioned for using $N^{\star}$, it also gives the data for same node density for different p, to make it very convenient to compare.

After the calculation of average time steps for all the cases, the next step was to try and correlate the computed Δt values with the factors affecting the discretization. Since the node density or value of $N^{\star}$ is comprised of both h and p, there are two ways to compare the $Avg\Delta t$ i.e. h- refinement and p-refinement. So, the $Avg\Delta t$ values are plotted against $N^{\star}$ with both h- refinement and p-refinement. But it is very clear from the nature of the curve for h- refinement that there is some kind of exponential power law between these plotted variables. In order to verify it, both the values are again plotted on log scale for h-refinement which can be seen in Figure 5.3. All the curves for $NGL = 3$ to 11 are linear and represent the straight lines. Not only they are linear but also they are parallel to each other. But still it is imperative to find out the slopes of all the lines so that the nature of the lines can be compared numerically. Although it looks like there are irregularities in some of the lines on the plot, the complexity of the KLE plays the huge part in making them slightly deviated from the mean value of $Avg\Delta t$. This further demands the need of numerical confirmation of the slope values. Since all the lines on the plot should follow the standard equations of a line in co-ordinate geometry, it is very logical to start with the equation of the line in slope-y intercept format as follows $-$

$$y = mx + c$$

Where, x and y are co-ordinate axes, m is the slope and c is the y intercept.

So, from N^* v/s Avg(Δt) plot in the logarithmic scale we get,

$$log\Delta t = \alpha(logN^*) + u \tag{5.5}$$

Where, α is the slope and u is the y intercept.

These variables are chosen to maintain the consistency in further representation.

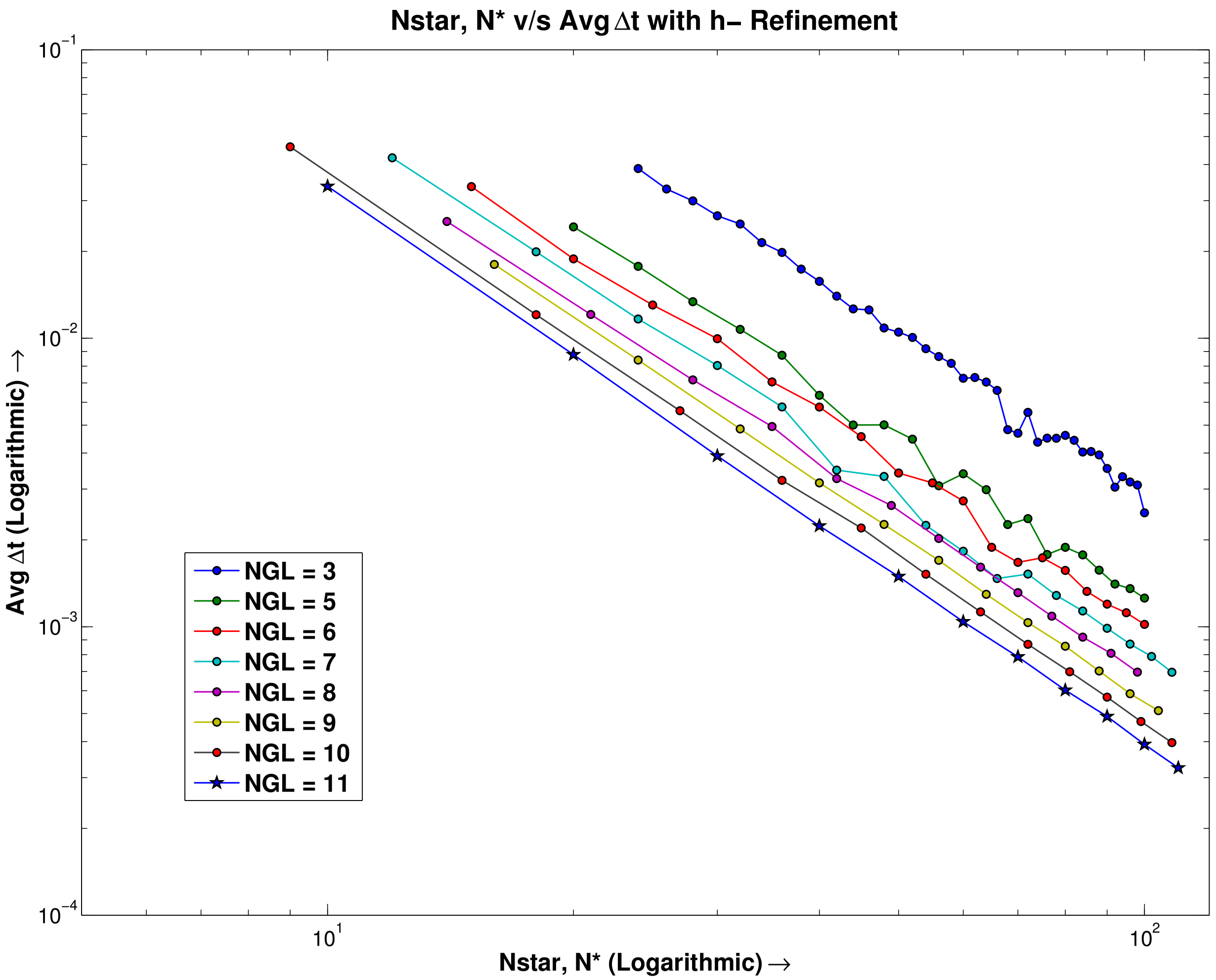


Figure 5.3: *A grid density $N^{\star}$ vs $Avg\Delta t$ for h-refinement showing its linear nature for all the curves*

Table 5.1
The values of slope α and Interpolation order p specific constants C_{NGL} after Least square approximation

NGL	α	C
3	-1.854095	14.6011915733936
5	-1.8782592	6.88230097450442
6	-1.9009012	5.97733575041749
7	-1.879249	4.48849971626125
8	-1.8831505	3.91524418051608
9	-1.9059706	3.58583549178151
10	-1.9066073	3.02491829136806
11	-1.9284963	2.81428209317862

If the anti-log of Equation 5.5 is taken, then (5.5) becomes $-$

$$\Delta t = e^{u}(N^{*})^{\alpha} \tag{5.6}$$

$$\text{if} \quad e^{u} = C$$

$$\text{then} \quad \Delta t = C(N^{*})^{\alpha} \tag{5.7}$$

The plot represents the set of points for each curve which should be identified by the common value of α, as far as the values of C are concerned, they are different for every NGL. The curve fitting is need to be done for each line and the method of "Least Square approximation" can be used to compute the required values of α and C. Table 5.1 shows the values of α and C for corresponding values of NGL. Now it can easily be concluded that all the NGL share the common value of α and have the same slope.

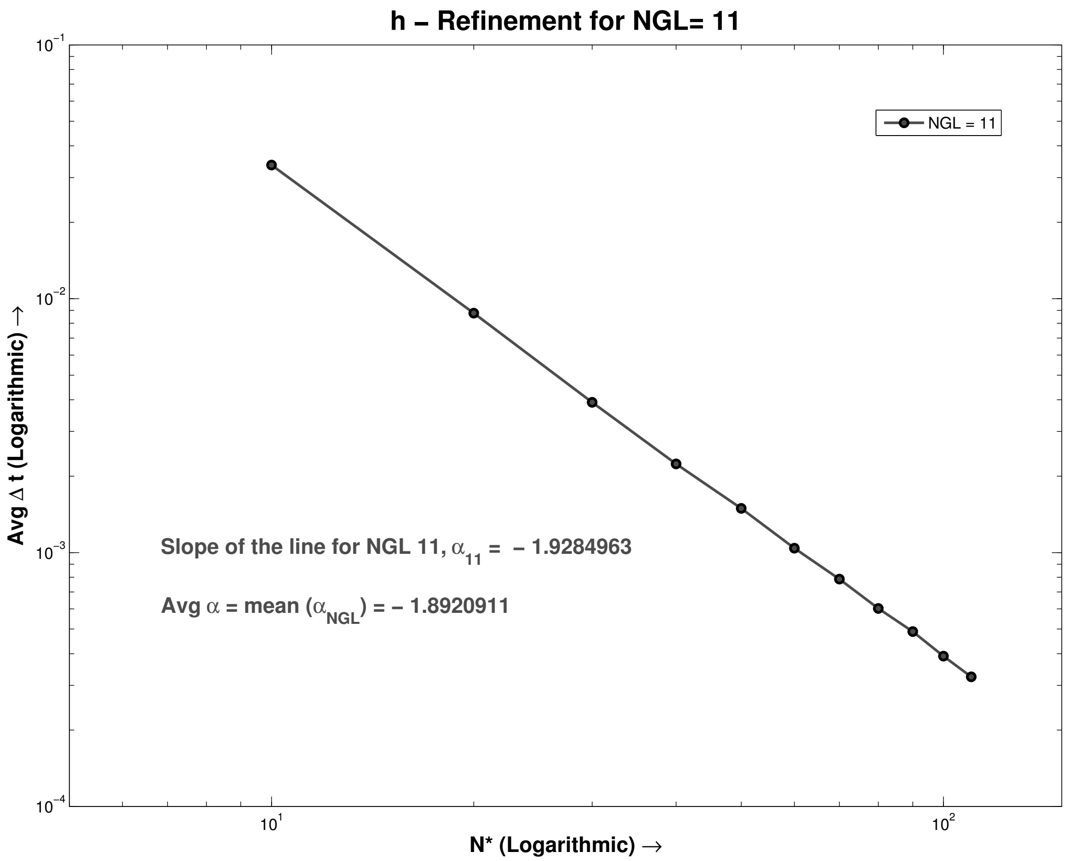


Figure 5.4: *The log scale plot of $N^\star$ vs $Avg\ \Delta t$ with average value of slope α*

If the generalized equation for the CFL condition is compared with Equation 5.7, the values of C could be identified as "constants" for the respective NGL. So, the above equation becomes –

$$\Delta t = C_{NGL}(N^*)^{-1.8921}$$

The above relation explains the limiting constraint on Δt with respect to the node density $N^\star$ for the spatial discretization. Although it gives the near perfect representation of the CFL condition, it has its own disadvantage in the form of "constant C_{NGL}". Here the constant C_{NGL} is dependent on the interpolation order or NGL. Since the above relation does not give the common value of "constant" for KLE, it is very inconvenient while deciding the time-step for adapting this numerical scheme to other physical problems. Also it should be remembered that the experiments have been performed only for a limited number of interpolation orders, so it shunts the possibility of using other values of NGL unless more experimentation is done with that specific value of NGL. But that clearly defies the generalized concept of CFL condition. Thus, the above relation is not satisfactory as far as the objective of this research is concerned.

As explained already, it becomes necessary to delve further with the values of NGL specific "constants". From the Table 5.1, it can be seen that the values of C are inversely proportional to the interpolation orders or NGL. It means C are decreasing with the increase in the order of interpolation. Each constant C is defined by the lines on the $N^{\star}$ v/s Δt plot for different NGL. The next step was to find out the relation between C_{NGL} and NGL, if any. It can begin the same way as was done with the earlier data. The values of C can be plotted with NGL on different types of scales to identify any numerical factor. When both these values are plotted on logarithmic scale, it again shows the existence of some kind of power law. Henceforth, to relate that power law with above established equation, instead of NGL, the term interpolation order p is used because p gives more identification to the spectral method discretization in general practice. So, if the values of p and C are plotted on the logarithmic scale; they explains the power law that combines them. Figure 5.5 shows the set of points that most certainly represents a linear behavior and can be simplified to a common equation as was done for the previous plot as follows –

From the equation of line in slope y intercept format

$$log(C) = \beta(log\, p) + u_1$$

Where, β is slope and u_1 is y intercept.

After taking anti-log

$$C = e^{u_1} p^{\beta} \tag{5.8}$$

$$\text{let,} \quad C_1 = e^{u_1}$$

$$\text{then} \quad C = C_1\, p^{\beta} \tag{5.9}$$

Comparing this equation with Equation 5.7,

$$\boxed{\Delta t = C_1\, p^{\beta} (N^*)^{\alpha}} \tag{5.10}$$

From least square approximation the values that we got for the constants in above equation are:

$$C_1 = 29.554032$$

$$\alpha = -1.8920911$$

$$\beta = -1.0286275$$

So the final CFL condition obtained is:

$$\Delta t = 29.554032\, p^{-1.0286275} (N^*)^{-1.8920911} \tag{5.11}$$

Where, $p =$ Interpolation order, N^* represents node density and $C_1 =$ Courant number for

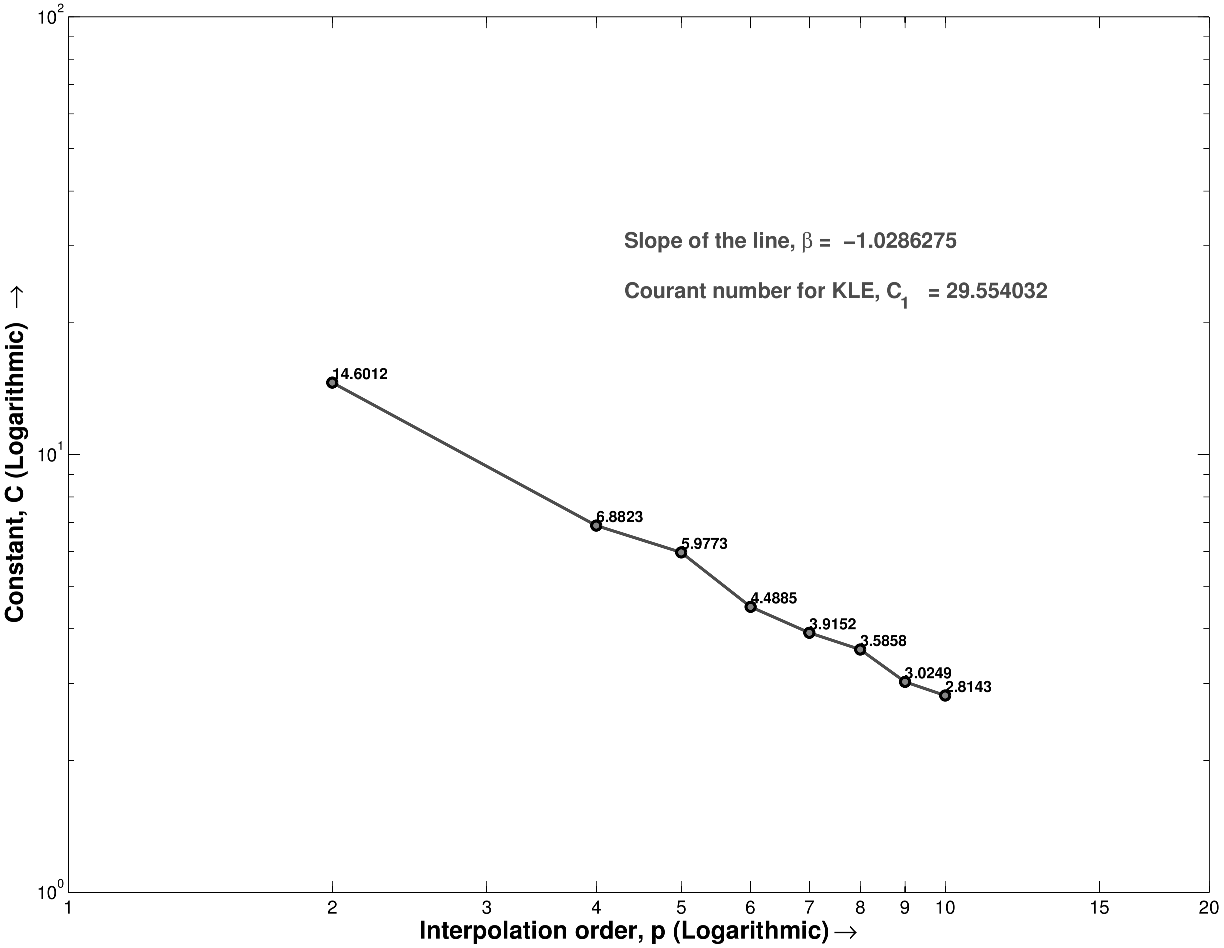


Figure 5.5: *The log scale plot of interpolation order p vs constants C_{NGL}*

KLE. The Equation 5.10 gives the ultimate relation for CFL condition and the Equation 5.11 gives the values of constants present in that equation. The value of Δt is restricted by the polynomial equation with the variables in the form of interpolation order p and the node density $N^{\star}$. Unlike the Equation 5.6, Courant number is not dependent on anything but a actual constant irrespective of the interpolation order p or any other input variables. With the developed CFL condition, the ideal time step Δt can be easily predicted with the values defining the spatial grid. The p and $N^{\star}$ are the parameters who set up the mesh on the given geometry using the spectral elements. As these are the only variables on which the time step is dependent, the basic definition for the CFL condition has been fulfilled.

As already mentioned, ABM PECE solver with adaptive step-size control is used for the experimentation. It has been used with KLE successfully and hence, MATLAB ode113 solver is employed here and the CFL condition on stability is evaluated. But, in order to validate the obtained relation on stability, variable step Runge Kutta 5th order method i.e. MATLAB's ode45 solver is used. With ODE45, after performing all the experiments for some selected interpolation orders, the final CFL condition on stability is given as,

$$\Delta t = C_1 \, p^{\beta} (N^{*})^{\alpha} \tag{5.12}$$

where, $p =$ Interpolation order, N^{*} represents node density and $C_1 =$ Courant number for KLE for ODE45.

The comparison of the values of the coefficients and Courant numbers in the obtained CFL condition for ODE113 and ODE45 is shown in Table 5.2.

Table 5.2

A comparison of values of Courant number and the coefficients for CFL condition

	Runge Kutta 5^{th} order method (MATLAB's ode45.m)	ABM PECE self adaptive method (MATLAB's ode113.m)
C_1	30.405897	29.554032
α	-1.9189584	-1.8920911
β	-1.2248473	-1.0286275

From the Figures 5.6 and 5.7, it can be seen that both ODE113 and ODE45 solvers give the same nature for the selected time steps and their relation with interpolation order p and $N^{\star}$. The $Avg\Delta t$ values show identical linear and parallel curves for the selected $NGLs$, exactly like ABM PECE solver when plotted against $N^{\star}$ values. Even the values of Courant number and coefficient α are similar. The value of β is different, but it is understandable since we are using two different ODE solvers which act as diagnostic algorithm to identify the optimum time step which is get restricted by stability condition. So the selection of time steps would not be exactly the same.

The evaluation of CFL condition with ODE45 certainly validates the experimental results and the stability condition that are obtained with ODE113. Hence, the CFL condition is not the characteristic of the specific solver, but, it explains the effect of the interpolation order p and mesh density on the time step for stability of the KLE for spectral element implementation.

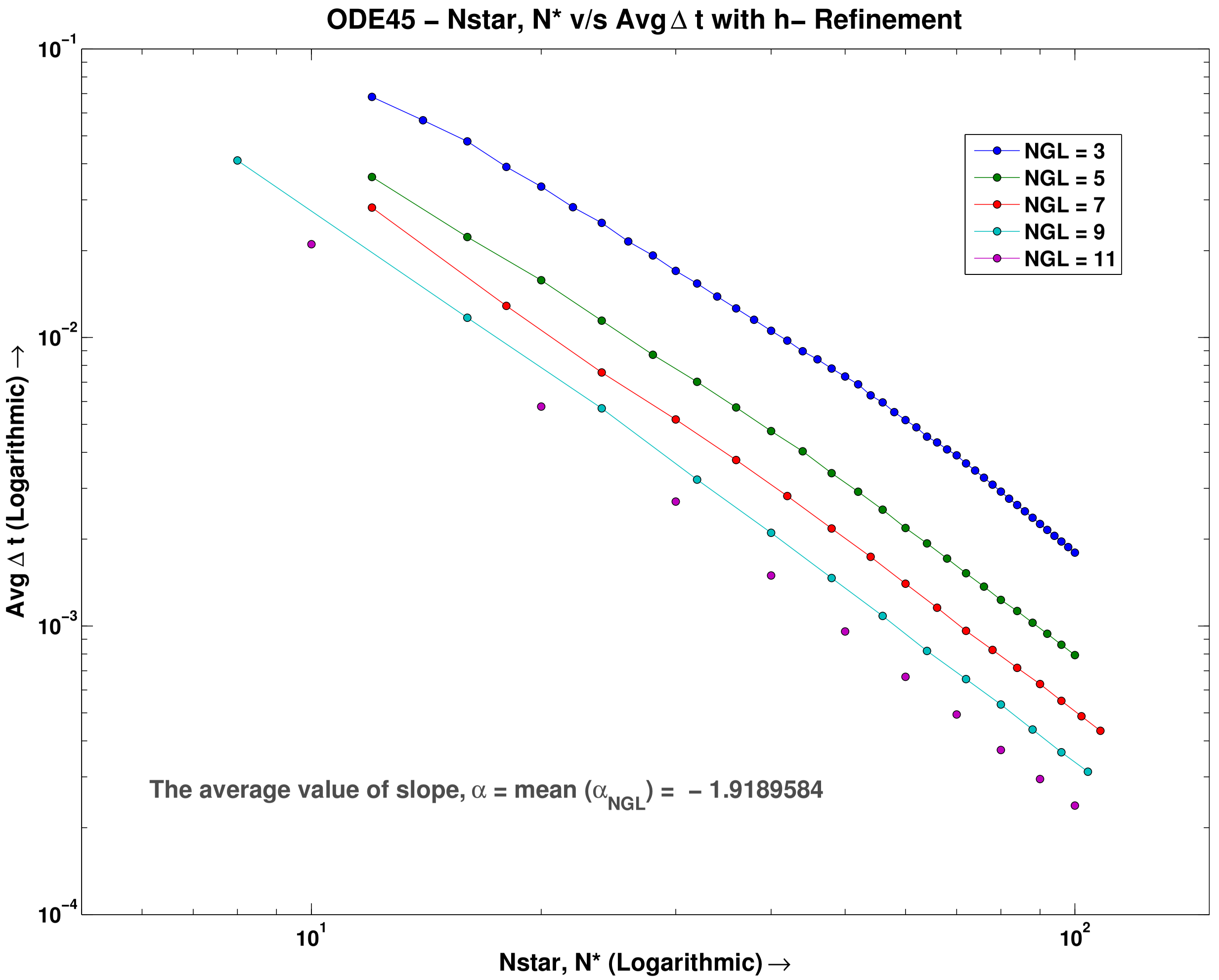


Figure 5.6: *ODE45: The log scale plot of $N^{\star}$ vs $Avg\ \Delta t$ with average value of slope α*

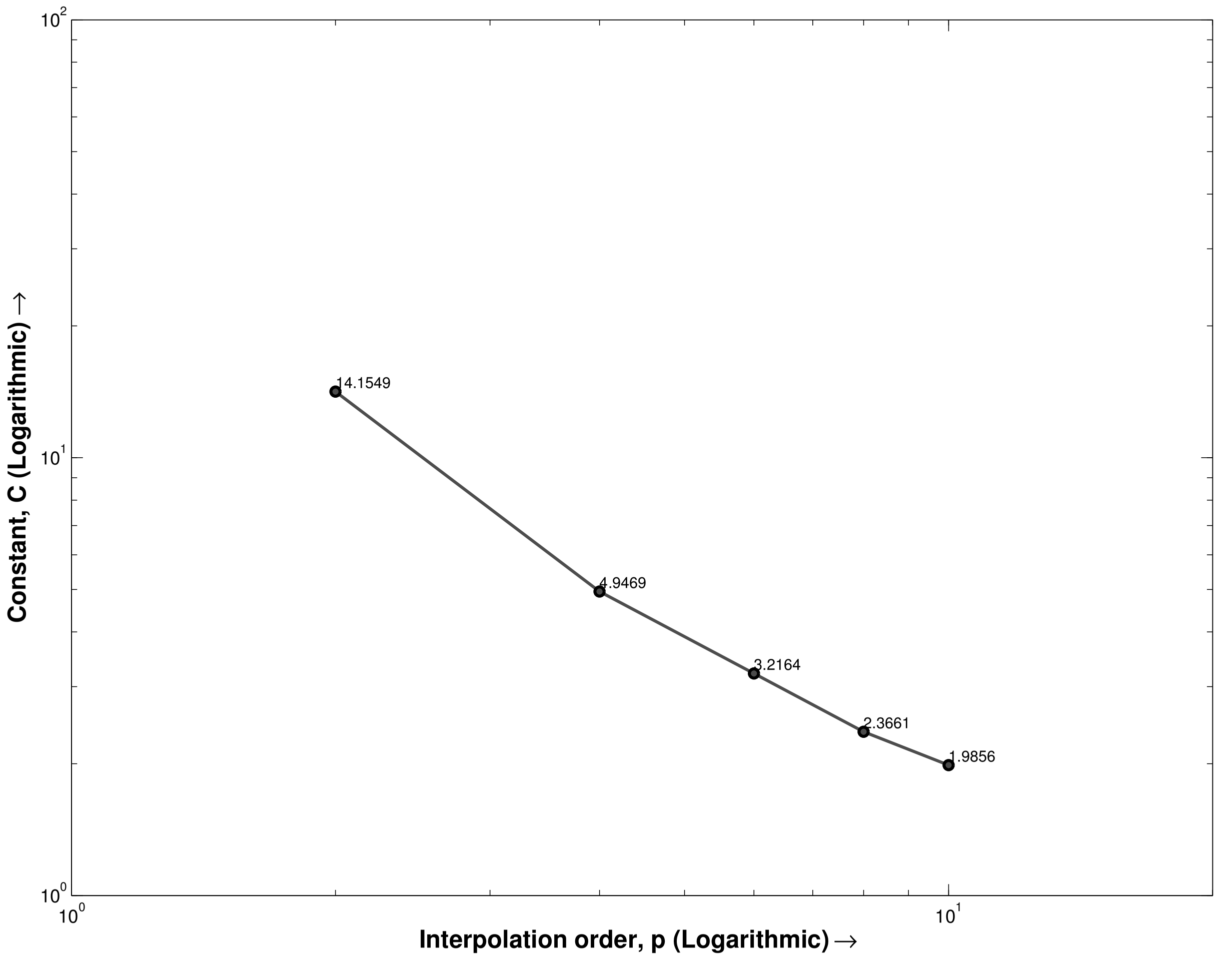


Figure 5.7: *ODE*45*: The log scale plot of interpolation order* p *vs constants* C_{NGL}

Chapter 6

Concluding Remarks and outlook for further work

The experiments have been performed to compute the CFL condition to maintain the stability of the KLE vorticity-velocity, spectral element discretization method. A canonical problem of boundary layer development over a flat plate with the structured quadrilateral mesh was used for the experimentation. For adaptive time marching ODE solvers, it has been well established that adaptivity based upon local error control can produce required stability properties for linear and non linear equations whether the underlying method is explicit or implicit and the stability region is very small. Due to this robustness, the solver ODE 113 has been employed for solving the unsteady Stokes's problem. The whole process can be broken down into the following major steps:

1. The range of values of h and p have been selected for the experimentation so that the common range of $N^{\star}$ is available for the comparison.

2. The values of Δt are plotted against the non-dimensionalized time τ for h and p refinement to observe the nature of the resultant curves.

3. The average time step for each and every case is calculated by taking mean of the time steps after the flow is fully developed which is believed to be representing the most ideal value of Δt between the values maxima and minima for the stable solution.

4. The $Avg\Delta t$ is then plotted against the node density $N^{\star}$ to establish the type of mathematical relation between them.

5. The interpolation order p dependent Courant number with time step restraint along with the exponential constant of $N^{\star}$ being the slope of the linear curve, is determined.

6. Since the dependency of Courant number on p doesn't give the required flexibility to the CFL condition, the values of C_{NGL} are then placed together with the interpolation order p to develop the possible correlation.

7. The final step in the process was similar to the computation of earlier relation due to the exactness of the linear nature of the p v/s C_{NGL} curve. Thus, the final equation gives an equality relation between the time step and the spatial grid size parameters.

The derived CFL condition gives the exact value of Δt; but if the time step above the stability limit is selected, any explicit or implicit algorithms will produce the errors in the propagation in the time marching, or it will not converge which means the numerical scheme will not be stable. On the other hand, Δt smaller than the stability limit actually produces very stable solution. So the CFL condition for the KLE can be written as $-$

$$\Delta t \leq C_1 \, p^{\beta}(N^{*})^{\alpha}$$

The above relation will guarantee the consistent stability for KLE application for unsteady incompressible flow. For h-p type finite element method, there are two ways of increasing accuracy: by increasing value of h or by increasing interpolation order p. If higher p is selected, there will be less nodes for the same accuracy i.e. less cost per time step. But with higher p, there will be more number of time steps. So, the obtained condition gives the effect of the change of p on the total computational time. $N^{\star}$ represents h, so it also expains the impact of the change of h on the stability. Hence, we arrive at complete characterization of the effect of CFL condition on stability for both p and h refinement. Whatever the order of the polynomial approximation, the solution will still show the convergence making it conditionally stable. Although this condition is an indirect way of evaluating CFL condition, the reason lies in the fact that no analytical stability criteria could be found out and it would be almost impossible to locate the region of Absolute Stability. Having an approximate estimate of Δt will give many advantages such as $-$

1. Maintaining an acceptable rate of convergence

2. Getting the desired accuracy when numerical solution is not sufficiently accurate.

An outlook for further work in this research line could be divided into two parts. The computation of CFL condition paved the way for improvement of the blended methods. The blending of the solvers like ODE113 and ODE $23s$ can be very effective for solving stiff problems. The mixing of explicit and implicit solvers or a combination of numerical schemes which are termed as methods with memory in typical MATLAB jargon can be used in splitting the multiphysics problems and used successfully as separate time stepping algorithms. The determination of CFL condition gives the perfect platform in identifying the time step restraint for these blended methods.

Another aspect of future work would be analyzing the stability using eigenvalues of Jacobian for this complex non-linear algorithm. It is well known that the eigenvalues play pivotal role in determining stability of the system of differential equations. For KLE to establish any analytical relation for stability, using eigenvalues is very complicated, and since the stability condition is already achieved, it would be interesting to start with numerical eigenvalues of jacobian and try to find their significance towards the stability.

Appendix

Permission for reusing the image in figure 1.1

This file is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license. You are free:

1. **to share** to copy, distribute and transmit the work
2. **to remix** to adapt the work

Under the following conditions:

1. **attribution** You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
2. **share alike** If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

ATTRIBUTION

Hans Hillewaert / CC-BY-SA-3.0

References

[1] Williamson CHK, Govardhan R. Vortex-induced vibrations. Annual Review of Fluid Mechanics . 2004;36:413–455.

[2] Ponta FL. The kinematic Laplacian equation method. Journal of Computational Physics . 2005;207:405–426.

[3] Fasel H. Investigation of the stability of boundary layers by a finite difference model of the Navier–Stokes equations. Journal of Fluid Mechanics . 1976;78:355–383.

[4] Clercx HJH. A spectral solver for the Navier–Stokes equations in the velocity-vorticity formulation for flows with two nonperiodic directions. Journal of Computational Physics . 1997;137:186–211.

[5] Hillewaert H. Windmills D1-D4 (Thornton Bank). `http://en.wikipedia.org/wiki/File:Windmills_D1-D4_(Thornton_Bank).jpg`. 2009. See Appendix for documentation of permission to republish this material.

[6] NREL. Wind Power Today. Report DOE/GO-102005-2115, U.S. Department of Energy. 2005.

[7] de Vries E. Thinking bigger: Are there limits to turbine size? Renewable Energy World . 2005;8(3).

[8] Rempfer D. On Boundary Conditions for Incompressible Navier-Stokes Problems. Applied Mechanics Reviews . 2006;59:107–125.

[9] Glowinski R. Finite Element Methods for Navier-Stokes equations. Annual Review of Fluid Mechanics . 1992;24:107–125.

[10] Quartapelle L. Numerical solution of the incompressible Navier-Stokes equations. Basel, Switzerland: Birkäuser. 1993.

[11] Glowinksi R, Pironneau O. Numerical methods for the ŕrst biharmonic equation and for the two-dimensional Stokes problem. SIAM Review . 1979;12:167–212.

[12] Quartapelle L, Napolitano M. Integral Conditions for the pressure in the computation of Incompressible viscous flows. Journal of Computational Physics . 1986;62:340–348.

[13] Fletcher C. Computational Techniques for Fluid Dynamics Vol.2. 2nd ed. Springer. 1988.

[14] Kwak D, Kiris C, Kim CS. Computational challenges of viscous incompressible flows. Computers and Fluids . 2005;34:283–299.

[15] Kempka S, Strickland J, Glass M, Peery J, Ingber PM. Velocity Boundary Conditions for Vorticity Formulations of the Incompressible Navier-Stokes Equations. In: Forum

on the Application of Vortex Methods to Engineering Problems. The U.S. Dept. of Energy.

[16] Gatski TB. Review of incompressible fluid flow computations using the vorticity-velocity formulation. Applied Numerical Mathematics . 1991;7:227–239.

[17] Davies C, Carpenter PW. A Novel Velocity−Vorticity Formulation of the Navier−Stokes Equations with Applications to Boundary Layer Disturbance Evolution. Journal of Computational Physics . 2001;172:119–165.

[18] X H Wu JZW, Wu JM. Effective vorticity-velocity formulations for 3D incompressible viscous flows. Journal of Computaional Physics . 1995;122:68–82.

[19] Karniadakis GE, Sherwin S. Spectral/hp Element Methods for CFD. 2nd ed. Oxford University Press. 2005.

[20] Ponta FL. Kinematic Laplacian Equation Method: A Velocity−Vorticity Formulation for the Navier-Stokes Equations. Journal of Applied Mechanics . 2006;73:1031–1038.

[21] Ponta FL, Jacovkis PM. A vortex model for Darrieus turbine using finite element techniques. Renewable Energy . 2001;24:1–18.

[22] Ponta FL, Jacovkis PM. Constant-curl Laplacian equation: a new approach for the analysis of flows around bodies. Computers and Fluids . 2003;32:975–994.

[23] Batchelor GK. An introduction to fluid dynamics. Cambridge, UK: Cambridge University Press. 2000.

[24] Anderson CR. Observations on vorticity creation boundary conditions. In: Caflisch RE, editor, Mathematical Aspects of Vortex Dynamics. SIAM. 1988;pp. 144–159.

[25] Chorin AJ. Numerical study of slightly viscous flow. Journal of Fluid Mechanics . 1973;57:785–796.

[26] Chorin AJ. Vortex sheet approximation of boundary layers. Journal of Computational Physics . 1978;27:428–442.

[27] Boyd JP. Chebyshev and Fourier spectral methods. Mineola, New York, USA: Dover. 2000.

[28] Sherwin SJ, Karniadakis GE. Tetrahedral hp Finite Elements: Algorithms and Flow Simulations. Journal of Computational Physics . 1996;124:14–45.

[29] Thompson MC, Hourigan K, Sheridan J. Three–dimensional instabilities in the wake of a circular cylinder. Experimental Thermal and Fluid Science . 1996;12:190–196.

[30] Karniadakis GE, Israeli M, Orszag SA. High-Order Splitting Methods for the Incompressible Navier-Stokes Equations. Journal of Computational Physics . 1991; 97:414–443.

[31] Sprague MA, Geers TL. Legendre spectral finite elements for structural dynamics analysis. Communications in Numerical Methods in Engineering . 2007;00:1–13.

[32] Patera AT. A spectral element method for fluid dynamics: laminar flow in a channel expansion. Journal of Computational Physics . 1984;54:468–488.

[33] Karniadakis GE, Bullister ET, Patera AT. A spectral element method for solution of two- and three-dimensional time-dependent incompressible Navier-Stokes equations. In: Finite Element Methods for Nonlinear Problems. New York/Berlin: Springer-Verlag, p. 803.

[34] Henderson RD, Karniadakis GE. Unstructured spectral element methods for simulation of turbulent flows. Journal of Computational Physics . 1995;122:191–217.

[35] Hourigan K, Thompson MC, Tan BT. Self–sustained oscilations in flows around long blunt plates. Journal of Fluids and Structures . 2001;15:387–398.

[36] Giraldo FX. The Lagrange–Galerkin spectral element method on unstructured quadrilateral grids. Journal of Fluid Mechanics . 1998;147:114–146.

[37] Bathe KJ. Finite element procedures. Englewood Cliffs, New Jersey, USA: Prentice Hall. 1996.

[38] Pathria D, Karniadakis GE. Spectral element method for elliptic problems in nonsmooth domains. Journal of Computational Physics . 1994;122:83–95.

[39] A M P Valli GFC, Coutinho ALGA. Control strategies for timestep selection in finite element simulation of incompressible flows and coupled reactionŰconvectionŰdiffusion processes. International Journal for numerical methods in fluids . 2005;47:201–231.

[40] Shampine LF, Gordon MK. Computer solution of ordinary differential equations: the initial value problem. San Francisco, California, USA: W. H. Freeman. 1975.

[41] George Hall AU. Modified order and stepsize strategies in Adams codes. Journal of Computational and Applied Mathematics . 1999;111:113–122.

[42] R Courant KF, Lewy H. On the partial difference equations of mathematical physics. Mathematische annalen . 1928;100:32–74.

[43] Kamga JBA, Després B. CFL condition and boundary conditions for DGM approximation of convection-diffusion. SIAM Journal on numerical analysis . 2006;44:2245–2269.