

Constitutive Relation Based Coupled Finite Element Formulation for Fluid-Structure Interaction problem in the Eulerian Framework

A Report Submitted
For the partial fulfillment of the M.Tech degree

by

Avinash Kumar Pandey

Roll No. 154103096



to the

**Department of Mechanical Engineering
INDIAN INSTITUTE OF TECHNOLOGY
GUWAHATI**

July 2017

Certificate

This is to certify that the work contained in the thesis titled “Constitutive relation based coupled Finite Element formulation for Fluid Structure Interaction problem on an Eulerian Framework”, by Avinash Kumar Pandey, Roll No: 154103096, in partial fulfilment of the requirements for the award of the degree of Master of Technology in Machine Design of the Indian Institute Of Technology, Guwahati, during the year 2016-2017, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

July, 2017

Dr. Annem Narayana Reddy
(Assistant Professor)
Department of Mechanical Engineering
Indian Institute of Technology, Guwahati.
Guwahati, Assam. 781039
India

Declaration

I hereby declare that the project entitled "**Constitutive Relation Based Coupled Finite Element Formulation for Fluid-Structure Interaction problem in the Eulerian Framework**" submitted to *Indian Institute of Technology, Guwahati*, is a record of original work done by me, under guidance of *Dr. Annem Narayana Reddy*, Assistant Professor, Department of Mechanical Engineering, Indian Institute of Technology, Guwahati. It represents my original ideas and where ever others' ideas or words have been used, they have been adequately cited and referenced to original sources. I also declare that I have adhered to all the principles of academic honesty and integrity, and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand the violation of any of above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which thus have not been properly cited. This project work is submitted in the partial fulfilment of the requirements for the award of the degree of Masters of Technology in Mechanical Engineering. The results embodied in this thesis have not been submitted to any other University or Institute for the ward of any degree or diploma.

July 2017
Guwahati

Avinash Kumar Pandey

Acknowledgement

It gives me immense pleasure to express my sincere gratitude towards my supervisor Dr. A. N. Reddy, Department of Mechanical Engineering, IIT Guwahati, for his continuous guidance, support and constant encouragement throughout the work. His ideas and thoughts were the only thing that kept me motivated during the work. This work would not have been possible without him. He has been more like a friend to me with whom I have never hesitated to share any trouble that I have faced during my work. I am also thankful to my senior Nishikant Maruti Mane who worked on the same project earlier and has helped me a lot taking the work forward.

I am thankful to Mr. Gaurav Kumar who has always enlightened me with his good suggestions and discussions during the project.

I am thankful to all technical and non-technical staff of Mechanical Engineering Department for their help in carrying out this project work.

I am deeply indebted to my father Dr. Arvind Kumar Pandey, my mother Meera Pandey and my siblings for the sacrifices they have borne to ensure the fulfilment of my dreams. Especially my father has always been a motivation for me who always has supported in my decisions with his guidance, suggestions and motivation.

Last but not the least all of my beloved friends specially Sanit and Sudhanshu with whom I spend most of my time have always maintained a healthy environment of discussion and fun as well, which always helps me stay more focused and stress free.

July 2017
Guwahati

Avinash Kumar Pandey

Abstract

In this thesis, a new strategy is proposed for solving fluid-structure interaction (FSI) problem where the coupling between fluid and solid is done at the level of constitutive relation. The FSI problems appear in many engineering problems such as aerospace structures, automobiles and underwater flexible robots. In addition, FSI appear in biomechanics. For example, flow of blood in veins, pumping of blood by heart, fish propulsion and bird/insect flying. Many solution strategies have been developed that can be classified into two groups: (i) partitioned approach (ii) monolithic approach. In all these algorithms they treat fluid and solid domain separately and couple them. In the proposed method, the stress is treated as a function of both strain and strain rate so that coupling takes place at constitutive level. In a domain, if the region belongs to the solid then more weight is given to strain term and negligible weight to strain rate in constitutive relation. The weights are reversed if the region belongs to fluid. Consequently, FSI problem can be solved without giving any special treatment. The proposed implementation is advantageous to carry out topology optimization on FSI problems.

As a first step, 4-node and 9-node continuum elements are implemented for plane stress problem and benchmarked with standard examples. In the second step, incompressible fluid mechanics governing equations have been implemented with pressure-velocity formulation. To assess the correctness of implementation, few benchmark problems are solved and matched the results. In the third step, the formulation for the fluid structure interaction is formulated and has been implemented.

Contents

Certificate	i
Declaration	ii
Acknowledgement	iii
Abstract	iv
Contents	v
1 Introduction	1
1.1 Fluid Structure Interaction(FSI) Problem	1
1.2 Classification of FSI systems	1
1.2.1 Weakly coupled system	1
1.2.2 Strongly coupled system	1
1.2.3 Conforming mesh methods	2
1.2.4 Non-conforming mesh	2
1.2.5 Classification according to solution procedures	2
1.3 Motivation	2
1.3.1 Example of interest:Fish propulsion problem	2
1.4 What we will do?	2
1.5 Closure	3
2 Literature Review	5
2.1 FSI Problem Formulation	5
2.2 Various well established techniques of simulating FSI problems	6
2.2.1 Monolithic approach	6
2.2.2 Partitioned approach	6
2.2.3 Single Solution Approach	6
2.3 Conforming Mesh Method	7
2.3.1 Interface data transfer	7
2.3.2 Accuracy and efficiency	8
2.4 Immersed Boundary Method	9
2.4.1 Basic formulation	10
2.4.2 Other Immersed boundary methods	12
2.4.3 Mesh size and accuracy	13
2.4.4 Stable time integration	13
2.4.5 LBB condition	13
2.5 Closure	13

3 Formulation of Elastic Structures	15
3.1 Governing equation	15
3.2 Weak form	15
3.3 Elemental equations for 4 noded element	17
3.3.1 Deflection of Cantilever Beam under Point Load	18
3.4 Closure	19
4 Formulation of Incompressible Fluids	21
4.1 Introduction	21
4.2 Governing equation	21
4.2.1 Weak form of stokes equation	22
4.2.2 Newton's Method	22
4.2.3 Formulation strategies	23
4.3 Unequal Velocity and Pressure interpolation formulation	23
4.3.1 9-Noded Velocity and 4-Noded Pressure Formulation	23
4.3.2 Flow Through sudden contraction	26
4.4 Transient Analysis	29
4.4.1 Lid Driven Cavity Problem	30
4.4.2 Contours, Vector Plot and Streamlines at Steady State	32
4.4.3 Streamlines at given time levels during time marching	32
4.5 Closure	34
5 Fluid-Structure Interaction Problem	35
5.1 Governing Equations	35
5.1.1 Momentum Equation	35
5.1.2 Conservation of mass or Continuity Equation	35
5.1.3 Velocity Displacement Relation	35
5.2 Constitutive Relations	35
5.2.1 Solids	35
5.2.2 Fluids	35
5.3 Development of Visco-Elastic Element	36
5.3.1 Weak Form	36
5.3.2 Linearised Formulation for Quadratic-Velocity Linear-Pressure and Quadratic-Displacement Formulation	38
5.4 Discussion	40
6 Volume fraction update for the elements and Solid Advection law	41
6.1 Need of update in the volume fraction for the elements	41
6.2 Method to update the Volume Fractions for the elements	42
6.3 Finding the area of intersection between an element and the solid polygon	42
6.3.1 Monte Carlo Method to find the volume fraction	42
6.4 Update in the displacements at the nodes of the solid polygon	44
6.4.1 Mapping a point from the computational domain to the master element	44
6.4.2 Flow over a Flexible Cantilever	45
6.4.3 Contours, Vector Plot and Streamlines at time t=2.0 sec	45
6.4.4 Deformation in the solid	47
6.4.5 Flow over a Flexible Rectangular Cylinder	48
6.4.6 Contours, Vector Plot and Streamlines at time t=6.3 sec	48
6.4.7 Deformation in the solid	50
6.5 Discussion	50
6.6 Closure	50
7 Conclusions and Future Scope	51
7.1 Conclusions	51
7.2 Future Scope	51

Appendix I	x
7.3 Interpolation function for 4 noded element	x
7.4 Interpolation function for 9 noded element	x
7.5 Mapping into the master element	x
7.5.1 Computation of Jacobian, Jacobian Matrix and derivatives of Shape Functions .	x
References	xi

List of Figures

1.1	Oscillating foils in soap film, Karman street left, reverse Karman street right [T. SCHNIPPER, A. ANDERSEN, and T. BOHR, 'Vortex wakes of a flapping foil', Journal of Fluid Mechanics', vol. 633, page 416, 2009][1], reproduced with permission.	3
2.1	Schematic of fluid and solid domains in a FSI problem	5
2.2	FSI methods conventional terminology	7
2.3	Examples of an (a)Immersed boundary and an (b)Immersed domain	10
3.1	Problem Statement for the Cantilever beam	18
3.2	(a)Deflections (b)Error vs. Iterations plot	19
4.1	Degrees of freedom of quadratic-velocity-linear-pressure element	25
4.2	Problem Statement for the flow through sudden contraction problem	26
4.3	Mesh used to simulate Flow through sudden Contraction problem	27
4.4	(a)u-velocity contour(b)v-velocity contour(c)Streamlines(d)Vector Plot at outlet	28
4.5	(a)Problem Statement (b)Mesh	31
4.6	(a)u-velocity contour (b)v-velocity contour (c)Vector Plot (d)Streamlines	32
4.7	Streamlines at different time levels	33
6.1	Solid advection inside the domain	41
6.2	Intersection of the element and the solid polygon	42
6.3	Monte Carlo approximation to find the percentage area of intersection between the element and the solid polygon	43
6.4	Mapping of a point in the computational domain to the master element	44
6.5	(a)Problem Statement (b)Mesh Used	45
6.6	(a)u-velocity contour (b)v-velocity contour (c)Vector Plot (d)Streamlines	46
6.7	(a)Problem Statement (b)Mesh Used	48
6.8	(a)u-velocity contour (b)v-velocity contour (c)Vector Plot (d)Streamlines	49
7.1	(a)4 noded and (b)9 noded rectangular elements	x
7.2	Natural and Cartesian coordinate system	xi

Chapter 1

Introduction

1.1 Fluid Structure Interaction(FSI) Problem

Fluid structure interaction(FSI) problem is a multiphysics problem which occurs in a system where flow of a fluid causes deformation of a solid structure which in turn changes the boundary condition of the fluid problem. FSI problems are a commonplace in engineering fields, yet presence of strong nonlinearity and multidisciplinary nature make it very difficult to formulate and analyze [2]. For most FSI problems, analytical solutions are impossible to obtain, and experiments are costly affair. Thus to investigate the fundamental physics involved in the complex interaction between fluids and solids, numerical methods may be employed. FSI applications include, but are not limited to, sedimentation (Mucha et al. 2004, Tornberg and Shelley 2004, Wang and Layton 2009), particle assembly (Liu et al. 2006), aerodynamics [3], turbulence (Kalogzin and Iaccarino 2003, Yang and Balaras 2006), complex flows in irregular domains (Fadlun et al. 2000, Udaykumar et al. 1996, 2001), electro-hydrodynamics (Hoburg and Melcher 1976), magneto-hydrodynamic flows (Grigoriadis et al. 2009), bio-fluid and bio-mechanics (such as cell aggregation and deformation, blood-heart interaction, inner ear fluid dynamics, jellyfish swimming, sperm motility, ciliary beating, etc.)[2]. There are many ways of going about this problem. That classifies the method as discussed in upcoming sections. The main problem to tackle is the different coordinate descriptions used for solid and fluid. Solids use Lagrangian approach and fluid use Eulerian approach. Second area of difficulty is the fact that fluid motion affects the solid deformation which in turn affects the fluid boundary condition. This makes us to use fully coupled algorithm

1.2 Classification of FSI systems

In general they FSI system is classified as either strongly coupled or weakly coupled as stated in multiphysics problems

1.2.1 Weakly coupled system

This is a case where a thermo-Fluid deforms a structure while the deformed structure hardly alters the flow field. E.g. problems of pressure vessels and piping in which there is small thermal deformation or low amplitude vibrations in fluid carrying pipe, turbulence induced vibrations in tube and shells.

1.2.2 Strongly coupled system

FSI are strongly coupled if alteration of flow field due to deformation of structure cannot be neglected. Another general classification of the FSI solution procedures is based upon the treatment of meshes: the conforming mesh methods and non-conforming mesh methods.

1.2.3 Conforming mesh methods

The conforming mesh methods consider the interface conditions as physical boundary conditions and require meshes that conform to the interface. It treats the interface as part of the solution. As we advance in solution meshes are redone according to changes in geometry of interface. This comes from obvious fact that at each time step or sub iteration fluid forces will deform the structure geometry and so location of interface changes. So the domains of governing equations change. Essentially we have to catch this change and so re-meshing is unavoidable.

1.2.4 Non-conforming mesh

The non-conforming mesh methods treat the boundary location and the related interface conditions as constraints imposed on the governing equations. Now the interface being part of equation, non-conforming meshes can be employed. As a result, the fluid and solid equations can be conveniently solved independently from each other with their respective grids, and re-meshing is not necessary. Conforming mesh methods have widely been studied and applied. In recent years methods like immersed methods are opening a way in for non conforming mesh method.

1.2.5 Classification according to solution procedures

FSI systems can also be studied according to complexity of the mathematics and computational procedures involved. Typically the fluid and solid components are modelled using different techniques to different levels of complexity, ranging from simple analytical solutions to 3-dimensional numerical schemes with advanced physical models. In addition, there is also the question of exchanging information, especially about boundary conditions, at the interface. To sum up all above we can classify on the basis of the level of coupling between fluid and solid, as shown [4].

1. Non-iterative over all time
2. Iterative over all time
3. Non-iterative over each time step
- Iterative over each time step

1.3 Motivation

Though there are well established numerical methods for solving the FSI problems, still these techniques use different solvers or distinguishes solid and fluid on macro level. Our motivation is to find such a method or element that can behave both like fluid or structure at the same time. This will eliminate much the complex mathematical manipulations. Although being said that it is necessary to find such an element which has open boundaries for further sophisticated development if in case such a need arises.

1.3.1 Example of interest:Fish propulsion problem

A great deal of study has been conducted on the manner and modes of swimming marine creatures from both a life sciences point of view as well as an engineering perspective. The shedding of vortices from the fish's main body and tail is believed to be the primary reason that fish swimming is as effective as it is. This idea is spurring ongoing research, with much of the scientific findings supporting the theory's credibility. Vortex shedding theory, as it relates to fish-like locomotion, states that a oscillating foil in a fluid will generate two columns of vortices behind it that, if conditions are right, will be organized in a manner consistent with a jet, and thus positive thrust will be produced. Two of the more basic structures are the Kármán street and the reverse Kármán street as shown in Fig. (1.4). Our ultimate challenge is to study this theory using our approach.

1.4 What we will do?

We will identify the key features of the methods that may be integrated to form an efficient and accurate algorithm to meet the computational challenges of FSI problems in literature review chapter.

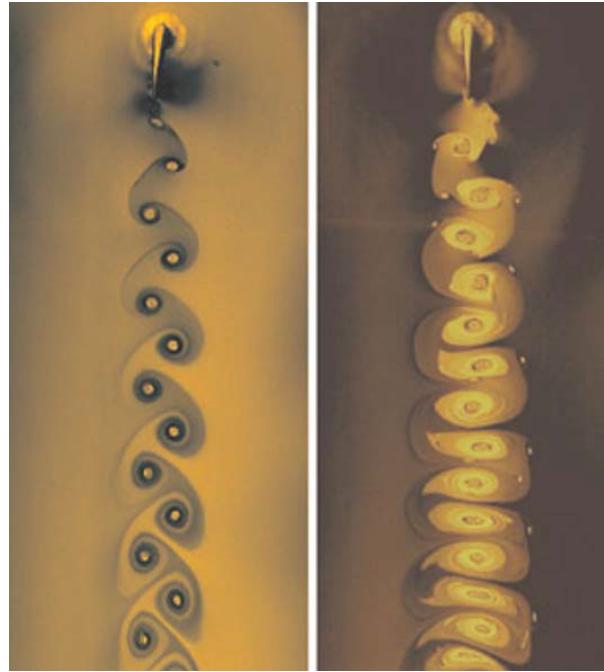


Figure 1.1: Oscillating foils in soap film, Karman street left, reverse Karman street right [T. SCHNIPPER, A. ANDERSEN, and T. BOHR, 'Vortex wakes of a flapping foil', "Journal of Fluid Mechanics", vol. 633, page 416, 2009][1], reproduced with permission.

We will first outline the basic FSI problem formulation. The partitioned approach-based conforming-mesh methods and the non-conforming mesh methods are reviewed in literature.

Then we will look into Finite Element Methods and study the conventional Elements of Structure and Fluids separately. Then in following chapters we will discuss the results.

Later on, we will try to resolve the FSI problem at constitutive level where the entire domain will be solved at one go, that is, both fluid and solid domain will be solved simultaneously.

1.5 Closure

In this chapter we introduced FSI problem. Although many times, FSI problem can be solved as decoupled problem, the presented examples of *fish propulsion* and *arterial blood flow*(flow through flexible pipes) motivates us to study the FSI problem as they are. To know better of the system and have a systematic approach we classified them into *weakly* and *strongly* coupled system, *moving* and *non-moving* mesh methods.

Chapter 2

Literature Review

In the literature review, the numerical solution methods for FSI problems in the context of incompressible flows are discussed. Particularly, we will review the solution procedures of the various methods. It is a goal of this review to identify the key features of the methods reviewed here that may be integrated to form an efficient and accurate algorithm to meet the computational challenges of FSI problems.

This review first outlines the basic FSI problem formulation. The partitioned approach-based conforming-mesh methods are reviewed in Section 2. The review of the non-conforming mesh methods is given in Section 3.

2.1 FSI Problem Formulation

We consider a computational domain, denoted by Ω , with an external boundary Γ .

Structural domain is Ω_s and the fluid domain is Ω_f

So, $\Omega = \Omega_s \cup \Omega_f$.

The fluid-structure interface is defined by $\Gamma_s = \Omega_s \cap \Omega_f$. See Fig.2.1 for illustration of the domains.

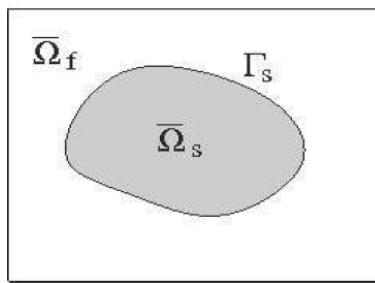


Figure 2.1: Schematic of fluid and solid domains in a FSI problem

The equations of motion for the fluid and structure may be expressed in the same index form, as a result of the D'Alembert's principle:

$$\rho \ddot{v}_i - \sigma_{ij,j} + f_i = 0 \quad (2.1)$$

where f_i is the body force, such as gravity.

2.2 Various well established techniques of simulating FSI problems

2.2.1 Monolithic approach

The monolithic approach [5][6][7][2] uses two separate sets of equations for fluid and solid and couple the fluid dynamics and structural dynamics implicitly and solve them synchronously at their common interface to form a single system equation for the entire problem. The interaction conditions are a part of governing equation. The discretized equations are solved by subiteration until convergence within one time step.

Advantages

- : 1. Better accuracy and efficiency.

These methods can be unconditionally stable and energy conservative when the modified Osher scheme is used for the fluid elements.

Disadvantages

- : 1. These methods are quite complex and computationally expensive due to the sub-iteration.

2.2.2 Partitioned approach

The partitioned approach treats the fluid and the structure as two computational fields which can be solved separately with their respective mesh discretization and numerical algorithm. The interaction conditions can obviously be applied while jumping from one field to another field.

Advantages

1. Already available (i.e., fluidic and structural) algorithms can be used and so take advantage of reliable and validated algorithms for solving many complicated fluid or structural problems.

Disadvantages

1. Very hard to coordinate the two algorithms to achieve accurate and efficient FSI solution with minimal code modification.
2. The interface location is unknown at starting and usually changes as code proceeds. Hence it requires the tracking of the new interface location and its related quantities, which can be complex and error-prone.
3. The computational overhead to run such codes is quite exorbitant as information has to pass from one code to the other in each time step, adding to the total overhead.
4. Data transfer usually requires an extra program that acts as an interface between the other two codes, thus sacrifices the modularity of the method. In the fluid structure interaction community, some researchers have focused in utilising a modular approach of the interface program for the exchange of information between two codes[8]. Such an approach is often called modular approach. An overview of the benefits and disadvantages of using these methods can be found in [9]. Partitioning leads inherently to loss of conservation of properties of the continua (fluid and structure). The energy increase in the system leads to instability which is the major drawback of this method.

2.2.3 Single Solution Approach

Some more sophisticated methods have also been proposed which can be classified as Single solution method instead of monolithic. Fig (2.2) may emphasize on difference.

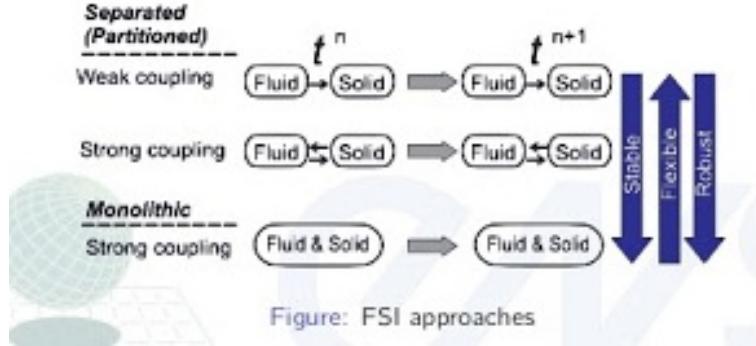


Figure 2.2: FSI methods conventional terminology

The single solution solution methods treat both fluid and solid as a continuum, thus the whole computational domain is a single entity in a single grid. Its behaviour is described by a single set of equations and is solved fully implicitly. There is no explicit exchange of information between the fluid and solid interface as it is inherently implicit[4].

The computational expense of the sub-iterations of the monolithic approach is expected to be avoided. The difficulty that lies with this method is the conceptual understanding of using a single set of equations to describe both fluid and solid, the choice of this single set of equations and the choice of appropriate boundary conditions. The creation this single set of equations can be done in one of two ways: use the solid as the prime model and reformulate the equations of the fluid to match the ones for the solid or the other way around. According to author the later approach is considered to be more natural for flexible vessels. In a single solution method, the distinction between the state of the continuum (fluid or solid) is associated with different coefficients in a single set of equations.

2.3 Conforming Mesh Method

The FSI methods with conforming meshes usually involve three fields that describe respectively the fluid dynamics, structural dynamics and mesh movement. The emphasis of these methods is on the coordination of data transfer and consistency between the existing fluid and structural codes. This method essentially solves two domains using separate algorithms and the effects are replicated on mesh with each iteration at a particular time step. Iterations are required to ensure that the interfacial conditions of both the displacement and the force are satisfied at the given time step. The challenges one might encounter when computing by means of an iterative coupled procedure are to maintain *proper data transfer* between the disciplines and to reach the converged solution *efficiently*.

Below we review techniques for interface data transfer and mesh movement and we discuss the accuracy, stability and efficiency of the methods.

2.3.1 Interface data transfer

The fluid dynamics mesh model usually represents the (whole) structure geometry. On the other hand, the structural analysis module concerns mainly the force bearing members. Therefore, the refined mesh will be placed around the high stress areas of structural part which most likely will not be on the fluid-structure interface. As a result, meshes of fluid and structural domain on the interface contain mismatches and even gaps. This can cause numerical difficulties in dealing with fluid dynamic load transfer and elastic deformation update(i.e. data transfer).

This can be tackled using two approaches: *the point match method* and *the artificial interface structure method*.

Point Match Method

As name suggest it basically identifies and matches a fluid mesh point to a structural mesh point on interface. Different authors have established different relations for matched points such as relation by determining the shortest distances between the points[10] or relation based upon the normal projection [2].

The displacement of the structural mesh point can be transferred to the fluid surface mesh point through a rigid element that connects the matched points [10][11][12][8]. Interpolation techniques can be used for points that are not connected but still on interface [10][8].

Another method is by conserving work. The work done by the structural load applied to the structural surface mesh is the same as that of the fluid dynamic load applied to the fluid surface mesh. But, Loads may not be conserved.i.e., the resultant fluid dynamic loads may not necessarily be equal to the resultant structural loads.

To tackle this Cebral and Lohner (1997a)[11] developed a conservative load projection method. But again it does not guarantee the consistency between the fluid and structural solutions.

Samareh (1996, 1998, 1999a,b)[13][14][15][16] developed a special connection method using non-uniform rational B-spline (NURBS). The structural displacements on the structure surface mesh points is projected onto the NURBS model. A new NURBS representing the deformed geometry is constructed with which the new fluid dynamic surface mesh may be established. The load transfer can then be accomplished in a similar manner. Although Samareh's method does not appear to be consistent or conservative, it offers a distinct advantage from the design point of view. Since the geometry of the NURBS surface is regulated by the control points, the coordinates of those points become a natural choice in shape design variables.

Artificial interface method

This method sometimes called the mortar method, introduces an artificial structure as a cover between the structural model and the fluid dynamic model. An example of the mortar method presented by Hou and Satyanarayana (2000) is given below.[2]

Let the three-dimensional unstructured computational mesh be represented by X_f^s , known as the fluid dynamic mesh. This mesh encloses the fluid domain, Ω_f . The fluid dynamic pressure on interface is converted to forces acting at the structural surface-mesh points, so that the structural module can be solved. The mesh on the fluid dynamic surface is referred to as the fluid dynamic surface mesh. The mesh on the structural surface for structural analysis is called the structural surface mesh, X_s^s . The mortar method introduces an artificial thin shell structure that covers the interface and acts as a medium for transferring the load and the displacement data between the fluid and the structural domains. Detailed discussion of such load transfer and deformation tracking are given respectively in [2]

Arbitrary Lagrangian-Eulerian(ALE) technique

One popular FSI method in this class is the Arbitrary Lagrangian-Eulerian (ALE) technique (Souli and Benson 2010) which incorporates the moving mesh explicitly into the fluid dynamics equation. It allows arbitrary motion of grid points with respect to their frame of reference by taking into account the convection of the material points. The material derivative in this case is expressed as

$$\frac{dv_f}{dt} = \frac{\partial v_f}{\partial t} + (v_f - \mathbf{U}) \cdot \nabla v_f \quad (2.2)$$

The movement of the fluid mesh, \mathbf{U} , can be set as U^a of Eq.(2.2).

2.3.2 Accuracy and efficiency

The sub-iterations between the fluid and structure solutions are important to the numerical performance of the method. In fact, Wood et al. (2010) showed that the FSI solution based on sequential computation of fluid and structural dynamics becomes unstable, if there are no sub-iteration steps

between fluid and structural computations. One additional sub-iteration can reduce two order magnitude of numerical error. And with more sub-iterations, better convergence can be achieved without a substantial increase in computational time. The particular example they studied for numerical performance is the flow-induced oscillation of a flexible cantilever. The authors used the three-step second order backward difference algorithm to approximate the first-order time derivative in the fluid solver and used the one-step Newmark predictor-corrector algorithm to solve the non-linear structural dynamic problem. The meshes range from 23,334 to 46,164 three dimensional fluid nodes and 567 to 850 two-dimensional structural nodes. The example is run with Dell PowerEdge SC1420 Severer with two Intel Xeon processors. Ten sub-iterations take about 4 minutes of wall clock time and the problem itself takes about 4 days of wall clock time for the simulation. It is interesting to note that in their study, 80% of computational time is for the fluid, 10% for the structure, and 10% for sub-iteration.

Many researchers have developed methods to improve the treatment of interface conditions, in an attempt to attain better accuracy, stability and efficiency for the three-field FSI methods. Some important techniques are:

- Estimate the interface location before starting the new FSI iteration
- Replace the standard Dirichlet and Neumann interface conditions by a general Robin Transmission [17]
- Better prediction of the interface locations(Farhat et al. (2006)[18] built a FSI method with second-order accuracy in time)
- Second-order FSI method with the computational fluid dynamics (CFD) code as a black box[3]
- Reduced-order models to improve computational efficiency[19].

These methods are described below:

2.4 Immersed Boundary Method

Most of the non-conforming mesh methods are based upon the framework of the immersed methods, which are a class of FSI methods that add force-equivalent terms to fluid equations to represent the fluid-structure interaction and to avoid mesh update in the numerical procedure. The immersed structure can be either a boundary (e.g., a curve in 2D and a surface in 3D) or a body with finite area (in 2D) or volume (in 3D), either rigid or flexible. Below we derive two classes of immersed methods, using the Lagrange multiplier approach: *the immersed boundary method* and *the immersed domain method*. Other types of immersed methods will be reviewed thereafter.

The immersed boundary method was originally developed by Peskin (1977)[20] for studying blood flow through a beating heart. This method solves the fluid equations with an additional term, the *FSI force*, which represents the effects of the immersed boundary acting on the fluid motion. The FSI force is computed explicitly from the structural configuration, which is then used to compute the fluid velocity. The no-slip condition is imposed on the immersed boundary, the location of which is updated by the structural velocity.*Essentially, the background fluid equations are solved in the entire domain with a fixed Eulerian mesh, and the moving boundary is tracked separately.* The need for mesh update is completely eliminated. ANSYS (1970-2011), one of the most popular computational mechanics and engineering software, incorporated the immersed boundary method for its FSI module in 2009. The main drwaback of this method is, it can only be applied to boundary geometry i.e. structures do not occupy volume. In essence, to approximate a volume body we have to take a set of such boundaries which may not accurately model realistic structural response.

To more accurately represent a bulk structure *the immersed domain method* was introduced. In the immersed domain method, an artificial fluid is introduced to cover the structural domain; thus, fluid domain is extended to the entire computational domain. To apply no-slip condition, the position and velocity between the artificial fluid of immersed structure and the local fluid are matched. To enforce this no-slip condition, the FSI force is imposed not only on the fluid-structure interfaces but also to every grid point in the artificial fluid domain. The fluid equation is then solved to yield the velocity field of the entire domain. Thus, the structural displacement and velocity are, at this stage, known. They can then be substituted into the suitable structural constitutive law to update the FSI

force, which in turn can be used by the fluid equation to find the new velocity of the fluid points. Representative examples of the immersed domain method include the immersed finite element method developed by Liu et al. (2006)[21], Wang and Liu (2004)[22] and Zhang et al. (2007)[3].

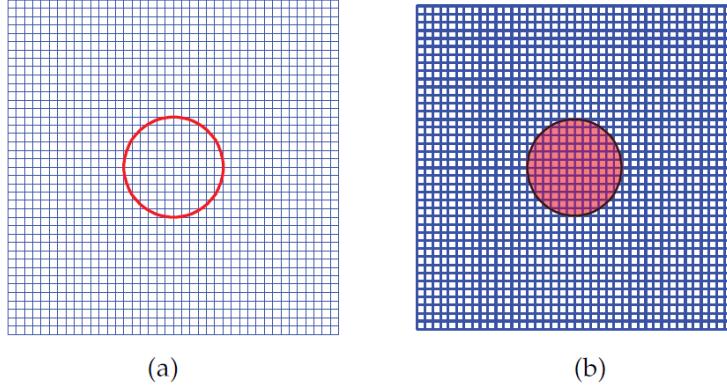


Figure 2.3: Examples of an (a)Immersed boundary and an (b)Immersed domain

2.4.1 Basic formulation

We consider the equations that describe the motion of the fluid in $\bar{\Omega}_f$ and the structure in $\bar{\Omega}_s$, given by

$$\rho^f \dot{v}_i^f - \sigma_{ij,j}^f + f_i^f = 0, \quad \text{in } \bar{\Omega}_f \quad (2.3)$$

$$\rho^s \dot{v}_i^s - \sigma_{ij,j}^s + f_i^s = 0 \quad \text{in } \bar{\Omega}_s \quad (2.4)$$

where f_i^f and f_i^s are external body forces (e.g., gravity) acting on the fluid and structure, respectively. For a FSI problem, the displacements should be the same along the interface:

$$u_i^f = u_i^s \quad \text{on } \Gamma_s \quad (2.5)$$

which may be viewed as a point-wise constraint applied to the interface, Γ_s . The no-slip condition imposed on the interface between these two domains is the result of the time differentiation of Eq. (2.19):

$$\dot{u}_i^s = \dot{u}_i^f \quad \text{on } \Gamma_s \quad (2.6)$$

$$\ddot{u}_i^s = \ddot{u}_i^f \quad \text{on } \Gamma_s \quad (2.7)$$

Or, in terms of velocities,

$$v_i^s = v_i^f \quad \text{on } \Gamma_s \quad (2.8)$$

$$\dot{v}_i^s = \dot{v}_i^f \quad \text{on } \Gamma_s \quad (2.9)$$

For simplicity, the superscript f that indicates quantity associated with the fluid field will be dropped from the notation. Based upon the principle of virtual work and the theorem of Lagrange multipliers, Eqs. (2.17)(2.19) may be combined into a single weak form as

$$0 = \int_{\bar{\Omega}_s} (\rho^s \dot{v}_i^s - \sigma_{ij,j}^s + f_i^s) \delta u_i^s dv + \int_{\bar{\Omega}_f} (\rho \dot{v}_i - \sigma_{ij,j} + f_i) \delta u_i dv + \int_{\Gamma_s} \bar{\lambda}_i (\delta u_i^s - \delta u_i) dv \quad (2.10)$$

where $\bar{\lambda}_i$ is the associated Lagrange multiplier defined over Γ_s , representing the force generated from the fluid-structure interaction. Note that the location of the interface boundary, Γ_s , is part of the unknown, and its position is determined by the interaction between the fluid and the structure.

Immersed boundary method

In the original formulation of the immersed boundary method invented by Peskin (1977,2002), the structure is represented by an immersed boundary which does not occupy a finite volume. Thus we have $\bar{\Omega}_s = \Gamma_s$ and the fluid domain becomes the entire computational domain: $\Omega = \Omega_f$. Consequently, Eq.(2.24) becomes

$$0 = \int_{\Gamma_s} \left(\rho^s \dot{v}_i^s - \sigma_{ij,j}^s + f_i^s + \bar{\lambda}_i \right) \delta u_i^s dv + \int_{\Omega} \left(\rho \dot{v}_i - \sigma_{ij,j} + f_i + \bar{\lambda}_i L(\Gamma_s) \right) \delta u_i dv \quad (2.11)$$

where the delta function, $L(\Gamma_s)$, is defined as

$$L(\Gamma_s) = \begin{cases} 1 & \text{if } x \in \Gamma_s \\ 0 & \text{if } x \notin \Gamma_s \end{cases}$$

Eq.(2.25) thus yields two independent equations

$$\rho^s \dot{v}_i^s - \sigma_{ij,j}^s + f_i^s + \bar{\lambda}_i = 0 \quad \text{on } \Gamma_s \quad (2.12)$$

$$\rho \dot{v}_i - \sigma_{ij,j} + f_i + \bar{\lambda}_i L(\Gamma_s) = 0 \quad \text{in } \Omega \quad (2.13)$$

In the immersed boundary method, the fluid-structure interaction force (i.e., the Lagrange multiplier $\bar{\lambda}_i$) is computed explicitly using Eq.(2.26). The computed force is then imposed on to Eq.(2.27), which is solved to yield fluid motion. In a numerical implementation, the discontinuous function $L(\Gamma_s)$ can be replaced by a continuous discrete delta function, which typically has compact support over a band of grid points neighbouring Γ_s . See Peskin (2002) for a detailed discussion and common choices of discrete delta functions. The use of a discrete delta function can be also regarded as an interpolation of the FSI force from the immersed boundary (the structural domain) to the fluid domain. As a result, the sharp interface is numerically represented by a thin layer of finite depth.

Once the fluid velocity is solved, the velocity of the structure is determined by applying the no-slip condition (2.22). The same discrete delta function is applied to interpolate the velocity from the fluid domain to the boundary. The location of the boundary Γ_s is then updated by using the structural velocity, and then used in the next cycle of computation.

Immersed domain method

The immersed domain method is an extension of the immersed boundary method that simulates motion of an immersed structure which occupies a finite volume. In this case, the constraint described in Eq. (2.19) is extended to the entire structural domain. Thus, Eq.(2.19) becomes

$$u_i^s = u_i \quad \text{in } \bar{\Omega}_s \quad (2.14)$$

which leads to the modification of the last term in Eq. (2.24) as

$$0 = \int_{\bar{\Omega}_s} \left(\rho^s \dot{v}_i^s - \sigma_{ij,j}^s + f_i^s \right) \delta u_i^s dv + \int_{\bar{\Omega}_f} \left(\rho \dot{v}_i - \sigma_{ij,j} + f_i \right) \delta u_i dv + \int_{\bar{\Omega}_s} \bar{\lambda}_i (\delta u_i^s - \delta u_i) dv \quad (2.15)$$

To expand the fluid domain to the entire computational domain Ω , the structural domain $\bar{\Omega}_s$ is filled with an artificial fluid. Meanwhile, the virtual work done by the expanded fluid,

$$\int_{\bar{\Omega}_s} \left(\rho \dot{v}_i - \sigma_{ij,j} + f_i \right) \delta u_i dv$$

is added to, and subtracted from, the original virtual work equation (2.29). Thus, the modified weak

form becomes

$$\begin{aligned}
0 &= \int_{\bar{\Omega}_s} \left(\rho^s v_i^s - \sigma_{ij,j}^s + f_i^s \right) \delta u_i^s dv - \int_{\bar{\Omega}_s} \left(\rho \dot{v}_i - \sigma_{ij,j} + f_i \right) \delta u_i dv \\
&\quad + \left[\int_{\bar{\Omega}_f} \left(\rho \dot{v}_i - \sigma_{ij,j} + f_i \right) \delta u_i dv + \int_{\bar{\Omega}_s} \left(\rho \dot{v}_i - \sigma_{ij,j} + f_i \right) \delta u_i dv \right] + \int_{\bar{\Omega}_s} \bar{\lambda}_i (\delta u_i^s - \delta u_i) dv \\
&= \int_{\bar{\Omega}_s} \left[\left(\rho^s v_i^s - \sigma_{ij,j}^s + f_i^s \right) \delta u_i^s - \left(\rho \dot{v}_i - \sigma_{ij,j} + f_i \right) \delta u_i \right] dv \\
&\quad + \int_{\Omega = \bar{\Omega}_f \cup \bar{\Omega}_s} \left(\rho \dot{v}_i - \sigma_{ij,j} + f_i \right) \delta u_i dv + \int_{\bar{\Omega}_s} \bar{\lambda}_i (\delta u_i^s - \delta u_i) dv \quad (2.16)
\end{aligned}$$

Note that Eq. (2.28) implies $\delta u_i^s = \delta u_i$ in $\bar{\Omega}_s$. We thus obtain

$$0 = \int_{\bar{\Omega}_s} \left[\left(\rho^s v_i^s - \sigma_{ij,j}^s - \rho \dot{v}_i + \sigma_{ij,j} + f_i^s - f_i + \bar{\lambda}_i \right) \delta u_i \right] dv + \int_{\Omega = \bar{\Omega}_f \cup \bar{\Omega}_s} \left(\rho \dot{v}_i - \sigma_{ij,j} + f_i - \bar{\lambda}_i L(\bar{\Omega}_s) \right) \delta u_i dv \quad (2.17)$$

where the function $L(\bar{\Omega}_s)$ is defined as

$$L(\bar{\Omega}_s) = \begin{cases} 1 & \text{if } x \in \bar{\Omega}_s \\ 0 & \text{if } x \notin \bar{\Omega}_s \end{cases}$$

Eq. (2.31) yields two independent equations:

$$\rho^s v_i^s - \sigma_{ij,j}^s - \rho \dot{v}_i + \sigma_{ij,j} + f_i^s - f_i + \bar{\lambda}_i = 0 \quad \text{in } \bar{\Omega}_s \quad (2.18)$$

$$\rho \dot{v}_i - \sigma_{ij,j} + f_i - \bar{\lambda}_i L(\bar{\Omega}_s) = 0 \quad \text{in } \Omega \quad (2.19)$$

Based on Eq. (2.28), the structural velocity is the same as the fluid velocity in $\bar{\Omega}_s$. Therefore, Eq. (2.32) yields

$$\bar{\lambda}_i = (\rho - \rho^s) \dot{v}_i + (\sigma_{ij,j}^s - \sigma_{ij,j}) + (f_i - f_i^s), \quad \text{in } \bar{\Omega}_s \quad (2.20)$$

Zhang and Gay (2007) derived the governing equations in their immersed finite element method which are similar to those presented here.

In the original immersed finite element method, Eq. (2.34) is first evaluated to find the FSI force, $\bar{\lambda}_i$. The known force is then imposed on Eq. (2.33) to solve the fluid motion in the entire domain. The computed fluid velocity is then interpolated back to the structural domain, based on the no-slip condition, and the structural configuration is then updated using the structural velocity.

A discrete delta function based on the Reproducing Kernel Particle Method, or RKPM (Wang and Liu 2004, Zhang and Gay 2007), is applied to interpolate the force from the structural domain to the fluid domain. To improve the accuracy at the fluid-structure interface, Lee, Chang and Choi et al. (2008) replaced the discrete delta functions by the Directly Coupled Euler-Lagrange Method (DCELM), in their simulation of rigid body motion using the immersed finite element method.[2]

We note that an important assumption in the immersed domain method is that the structure is incompressible (or nearly incompressible), since the immersed structure has to abide by the same velocity constraint as that of the surrounding incompressible fluid.

2.4.2 Other Immersed boundary methods

In addition to the immersed domain method mentioned above, notable examples include the *immersed interface method*, the *direct forcing method* [2], and the *distributed Lagrange multiplier method* [23][24][25][26][27].

2.4.3 Mesh size and accuracy

The mesh size is an important factor in determining the stability and accuracy of the immersed methods. The choice of mesh size becomes more crucial for FSI problems involving complex interface geometry and flow physics, in which smaller mesh size or better approximation of variables around the interface are required, particularly in the case of high Reynolds and Mach numbers.

Based upon a finite element error analysis, Glowinski et al. (1999)[23] indicated that the fluid mesh size, h_f , should be smaller than the structure mesh size, h_s , in order to maintain efficiency, while it is the other way around to maintain accuracy. To reach a compromise, they recommended that h_f and h_s should be on the same order. Zhao et al. (2008) studied the FSI problems associated with biological systems in which the immersed flexible body is made of neo-Hookean materials. The accuracy of the proposed FSI method achieved an accuracy between second and third order in terms of h_s , but only between first and second order in h_f . The localization of major errors near the interface was also observed in their study. Based on these observations, the authors suggested using the adaptive mesh refinement (AMR) for better accuracy.

Several studies have incorporated the local refinement techniques into the immersed methods to yield better accuracy. Kaligzin and Iaccarino (2003) combined the immersed boundary formulation with AMR to simulate 3D high Reynolds number flows. Tai et al. (2005, 2007)[28][29] introduced a densed overlapping mesh around the interface to obtain better estimate of the friction and pressure distributions on the rigid body surface. Three meshes were used: the stationary fluid mesh, the sub-domain with the overlapping mesh which is dense and wrapped around the structure, and the rigid nodes distributed within the rigid structure. A loosely coupled iterative procedure between the fluid and the structure is used in solving the FSI problem. The fluid equation with the immersed object is solved first. The fluid solution is then computed on the overlapping domain using the same Navier-Stokes solver, but with a moving grid.

2.4.4 Stable time integration

Another challenge that researchers of immersed methods frequently face is that boundary forces may impose a severe restriction on time-step size in order to maintain numerical stability (Fauci and Folgelson 1993, LeVeque and Li 1997, Peskin 2002, Stockie and Wetton 1999)[2]. The numerical stability of an immersed method can be much improved if the boundary forces are treated implicitly to advance the boundary in time. Although much effort has been invested in developing implicit and semi-implicit versions of the immersed boundary method and related methods, it remains a challenge to develop a immersed method that is computationally efficient even for stiff boundary forces.

2.4.5 LBB condition

This is a basic stability condition for either mixed principle or Lagrange multiplier variational methods for the study of wave motion and incompressible fluids in finite element methods. It ensures if the problem is well-posed. It also controls the way of discretization in finite element methods.

2.5 Closure

The last few decades have seen a tremendous number of numerical methods developed for the simulation of FSI. The primary driving force for these developments is the demand from a wide range of scientific and engineering disciplines, where FSI problems are playing increasingly important roles. Meanwhile, the fast improvement of computational powers has made large-scale FSI simulations possible and has facilitated many realistic applications of these numerical techniques. Owing to the multidisciplinary nature of FSI problems, in this review, we have emphasized the numerical procedures used by various methods to treat the interface conditions between fluids and structures.

The first class of methods we reviewed is based on the partitioned approach which requires conforming mesh. The partitioned approach allows the fluid dynamics and structural mechanics that are involved in the FSI problems to be solved separately by their respective algorithms and codes. Since it provides flexibility in spatial meshing, the methods using the partitioned approach can conveniently

catch the detailed physics along the fluid-structure interface. However, the difficulty in data handling along the fluid-structure interface and instability discourages the use of the partitioned approach for broad FSI applications.

The second class of methods we focused on are the immersed methods which use non-conforming mesh. The immersed methods, in recent years, have become increasingly popular in FSI simulations owing to their simplicity, efficiency and flexibility, as well as their capability to handle complex flows and large structural deformations, compared to the partitioned methods with conforming meshes. *A major disadvantage of these methods, however, is their lack of resolution near the interface* (an exception is the immersed interface method). Typically, the immersed methods smear out sharp interfaces to a thickness in the order of mesh width. To some extent, *local grid refinement* blurs the distinction between the immersed methods and those with moving meshes. *The immersed interface method is one of the few immersed methods that can achieve second-order spatial accuracy and preserve sharp fluid-structure interface. The development of second or higher-order immersed method to accurately compute FSI problems with embedded bulk structures remains an open question.*

Furthermore, on additional note, we are studying the numerical methods and applications that consider the interaction between immersed structures and one fluid (i.e., single-phase flow). Still big challenges remain in the simulation of fluid-structure interaction in multiphase flows. Notable examples of related applications include high-speed boats cruising on water, wind turbines floating in oceans, and energy buoys interacting with waves. Such FSI problems challenges the current numerical methods, as both the fluid-fluid interface and fluid-solid interface have to be accurately computed to reliably represent the physics involved.

Chapter 3

Formulation of Elastic Structures

3.1 Governing equation

For most finite element methods in solid mechanics, the equilibrium equation

$$\sigma_{ij,j} + f_i = 0, \quad \text{in } \Omega \quad (3.1)$$

is often chosen as the departure point, where σ_{ij} is the Cauchy stress, f_i is the body force, and Ω is the problem domain enclosed by Γ boundary.

3.2 Weak form

For an approximated solution residual of above equation can be written as,

$$R_i = \sigma_{ij,j} + f_i$$

Using a weight function W_i weighted integral statement is made,

$$\int_{\Omega} W_i R_i d\Omega = 0$$

or

$$\int_{\Omega} W_i (\sigma_{ij,j} + f_i) d\Omega = 0 \quad (3.2)$$

simplifying this following forms are obtained,

$$\int_{\Omega} (W_i \sigma_{ij})_{,j} d\Omega - \int_{\Omega} W_{i,j} \sigma_{ij} d\Omega + \int_{\Omega} W_i f_i d\Omega = 0$$

using divergence theorem,

$$\int_{\Omega} W_{i,j} \sigma_{ij} d\Omega - \int_{\Gamma} W_i (\sigma_{ij} n_j) d\Gamma - \int_{\Omega} W_i f_i d\Omega = 0$$

if \mathbf{t} is to be traction vector then $\sigma_{ij} n_j = t_i$. Incorporating this into above equation gives basic weak form as follows,

$$\int_{\Omega} W_{i,j} \sigma_{ij} d\Omega = \int_{\Gamma} W_i t_i d\Gamma + \int_{\Omega} W_i f_i d\Omega \quad (3.3)$$

Considering the constitutive relation, stress can be put in terms of displacements.

$$\sigma_{ij} = C_{ijkl} \epsilon_{kl}$$

where ϵ_{kl} is the strain. It can be stated in terms of displacement as(considering small displacements),

$$\epsilon_{kl} = \frac{1}{2} (u_{k,l} + u_{l,k})$$

now taking into consideration the first term in the weak form, It actually denotes inner product. If $W_{i,j}$ is split into symmetric and antisymmetric part as $\frac{1}{2}(W_{i,j} + W_{j,i}) + \frac{1}{2}(W_{i,j} - W_{j,i})$ and then multiplied with σ_{ij} , the following elation is obtained,

$$\int_{\Omega} W_{i,j} \sigma_{ij} d\Omega = \int_{\Omega} \frac{1}{2}(W_{i,j} + W_{j,i}) \sigma_{ij} d\Omega$$

because σ is symmetric and $\frac{1}{2}(W_{i,j} - W_{j,i})$ is antisymmetric.

The quantity $\frac{1}{2}(W_{i,j} + W_{j,i})$ can be looked at as virtual displacement. To complete the formulation using Galerkin method we can use same interpolation function to W as that of u . the more sophisticated form of above weak form can be written as,

$$\int_{\Omega} \frac{1}{2}(W_{i,j} + W_{j,i}) C_{ijkl} \frac{1}{2}(u_{k,l} + u_{l,k}) d\Omega = \int_{\Gamma} W_i t_i d\Gamma + \int_{\Omega} W_i f_i d\Omega \quad (3.4)$$

approximating the u and W as,

$$u_k = N_{kn} \hat{u}_n \quad W_i = N_{im} \delta \hat{u}_m \quad (3.5)$$

Where N_{ij} is the j^{th} interpolation function(corresponding to j^{th} dof) for i^{th} component field variable u . Now putting this back in equation,

$$\left[\int_{\Omega} \frac{1}{4} (N_{im,j} + N_{jm,i}) C_{ijkl} (N_{kn,l} + N_{ln,k}) d\Omega \right] \hat{u}_n = \int_{\Gamma} N_{mi} t_i d\Gamma + \int_{\Omega} N_{mi} f_i d\Omega \quad (3.6)$$

which is equivalent to

$$K_{mn} \hat{u}_n = f_m$$

The equation involves a fourth order tensor C_{ijkl} . Hence we drag down things from fourth order to second order and second order to first order and then formulate. This can be expressed more easily in vector form as follows

Governing equation is $\nabla \sigma + \mathbf{f} = 0$ in equivalent notations as above we get

$$\int_{\Omega} \sigma : \nabla W d\Omega = \int_{\Gamma} t W d\Gamma + \int_{\Omega} f W d\Omega$$

using constitutive relation, strain displacement relation and splitting ∇W

$$\begin{aligned} \sigma &= C \epsilon \\ \epsilon &= \frac{1}{2} (\nabla u^T + \nabla u) \\ \nabla W &= \frac{1}{2} (\nabla W^T + \nabla W) + \frac{1}{2} (\nabla W^T - \nabla W) \end{aligned}$$

following result is obtained,

$$\int_{\Omega} [D_c W]^T C [D_c u] d\Omega = \int_{\Gamma} W^T t d\Gamma + \int_{\Omega} W^T f d\Omega$$

Where D_c is differential operator that depicts the operation as in earlier expression. With the interpolation function as below,

$$u = N \hat{u} \quad W = N \delta \hat{u} \quad (3.7)$$

for galerkin approach following final form of equation is obtained

$$\left[\int_{\Omega} [D_c N]^T [C] [D_c N] d\Omega \right] \{\hat{u}\} = \int_{\Gamma} [N]^T \{t\} d\Gamma + \int_{\Omega} [N]^T \{f\} d\Omega$$

if $D_c N = B$ and representing $[c]$ as $[D]$,

$$\int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega \hat{\mathbf{u}} = \int_{\Gamma} \mathbf{N}^T \mathbf{t} d\Gamma + \int_{\Omega} \mathbf{N}^T \mathbf{f} d\Omega$$

Therefore it boils down to following relation of stiffness matrix and force matrix as,

$$[K]\{\hat{u}\} = \{f\} \quad (3.8)$$

where

$$\mathbf{K}_{2n \times 2n} = \int_{\Omega} \mathbf{B}_{2n \times 3}^T \mathbf{D}_{3 \times 3} \mathbf{B}_{3 \times 2n} d\Omega \quad (3.9)$$

$$\mathbf{f}_{2n \times 1} = \int_{\Gamma} \mathbf{N}_{2n \times 2}^T \mathbf{t}_{2 \times 1} d\Gamma + \int_{\Omega} \mathbf{N}_{2n \times 2}^T \mathbf{f}_{2 \times 1} d\Omega \quad (3.10)$$

n = order of element

3.3 Elemental equations for 4 noded element

For 4 noded element:

$$\mathbf{N} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \quad (3.11)$$

$$\hat{\mathbf{u}} = \begin{Bmatrix} u_{x_1} \\ u_{y_1} \\ u_{x_2} \\ u_{y_2} \\ u_{x_3} \\ u_{y_3} \\ u_{x_4} \\ u_{y_4} \end{Bmatrix} \quad (3.12)$$

where $u_{i,j}$ represents displacement of j^{th} node in i^{th} direction.

$[\mathbf{B}]\{\hat{\mathbf{u}}\}$ essentially represents strain $\{\epsilon\}$

$$\{\epsilon\} = \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} u_{x,x} \\ u_{y,y} \\ u_{x,y} + u_{y,x} \end{Bmatrix} = \begin{bmatrix} N_{1,x} & 0 & N_{2,x} & 0 & N_{3,x} & 0 & N_{4,x} & 0 \\ 0 & N_{1,y} & 0 & N_{2,y} & 0 & N_{3,y} & 0 & N_{4,y} \\ N_{1,y} & N_{1,x} & N_{2,y} & N_{2,x} & N_{3,y} & N_{3,x} & N_{4,y} & N_{4,x} \end{bmatrix} \{\hat{\mathbf{u}}\}$$

therefore \mathbf{B} can be formulated as

$$\mathbf{B} = \begin{bmatrix} N_{1,x} & 0 & N_{2,x} & 0 & N_{3,x} & 0 & N_{4,x} & 0 \\ 0 & N_{1,y} & 0 & N_{2,y} & 0 & N_{3,y} & 0 & N_{4,y} \\ N_{1,y} & N_{1,x} & N_{2,y} & N_{2,x} & N_{3,y} & N_{3,x} & N_{4,y} & N_{4,x} \end{bmatrix} \quad (3.13)$$

The Matrix \mathbf{D} from constitutive relation \mathbf{D} can be given as

For plain strain:

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \quad (3.14)$$

For plain stress:

$$\mathbf{D} = \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad (3.15)$$

Where E is Elastic modulus and ν is poisson's ratio.

3.3.1 Deflection of Cantilever Beam under Point Load

The deflection behaviour of a Cantilever Beam has been studies under a load of 100 KN. For implementation 4 noded elements were used.

Problem Statement

The problem statement with dimensions and boundary conditions has been shown in fig. 3.1

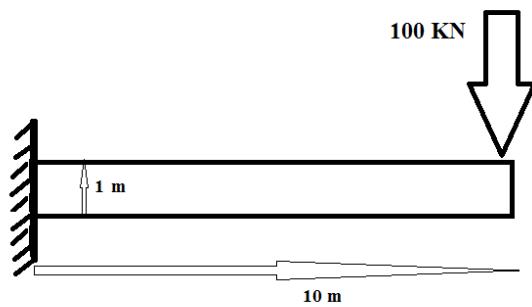


Figure 3.1: Problem Statement for the Cantilever beam

Results

The fig. 3.2(a) shows the deflection of the beam at the end for different mesh configurations while the fig. 3.2(b) shows the how the error value changes at with the number iterations.

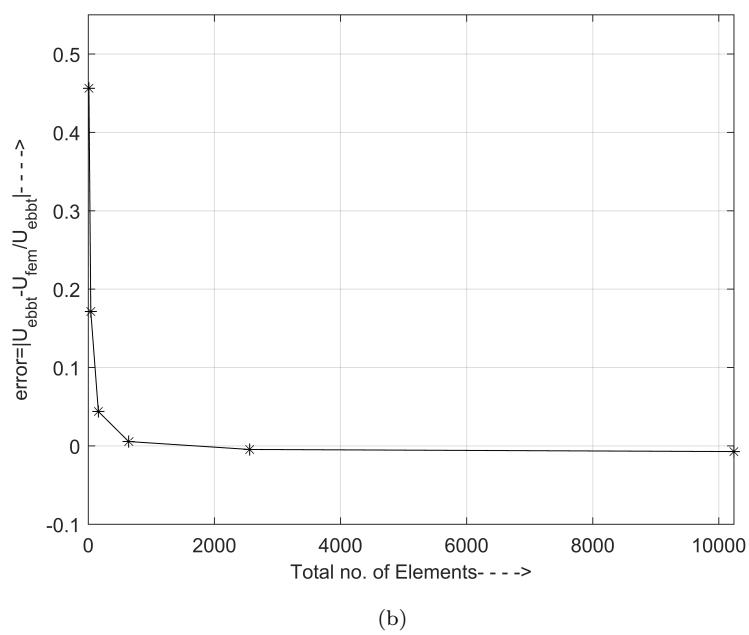
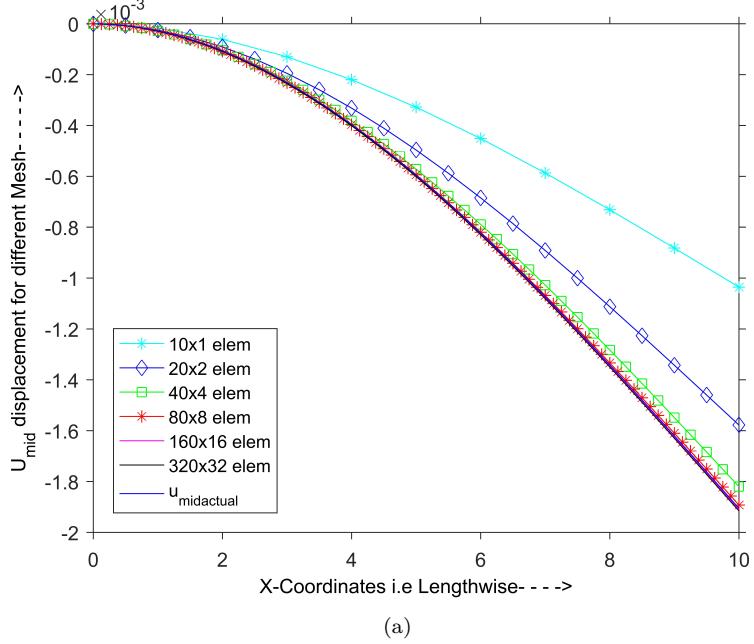


Figure 3.2: (a)Deflections (b)Error vs. Iterations plot

3.4 Closure

In this chapter Galerkin method was implemented to solids and only one case was tested as it's not the main emphasis of the project, only a step to understand the nature of the solids.

Chapter 4

Formulation of Incompressible Fluids

4.1 Introduction

There are two methods that can be used to formulate a fluid element.

- Continuous-pressure formulation(velocity-pressure formulation)
- Discontinuous-pressure formulation(Penalty formulation)

Both the formulations yield solutions of comparable accuracy with penalty finite element approach being more computationally efficient due to implicit treatment of pressure field. But continuous pressure formulation seems more robust and it obtains nodal values of pressure without the need for any extrapolation and averaging(Jog and Kumar,2009). Looking at this background, continuous-pressure formulation is used to formulate the incompressible fluids.

4.2 Governing equation

Let Ω denote the domain with boundary Γ . continuity equation can be given as,

$$\nabla \cdot \mathbf{v} = 0 \quad (4.1)$$

Momentum conservation equation:

$$\rho \left[\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right] = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b} \quad (4.2)$$

Constitutive relation for viscous fluid

$$\boldsymbol{\sigma} = -p \mathbf{I} + \boldsymbol{\tau} \quad (4.3)$$

with

$$\boldsymbol{\tau} = 2\mu \dot{\epsilon} \quad (4.4)$$

where $\dot{\epsilon}$ stands for shear strain rate

$$\dot{\epsilon} = \frac{1}{2} [(\nabla \mathbf{v}) + (\nabla \mathbf{v})^T] \quad (4.5)$$

$\nabla \mathbf{v}$ denotes gradient of velocity., p is pressure, μ is dynamic viscosity coefficient.

4.2.1 Weak form of stokes equation

Considering only stokes approximate equation for formulation which is obtained by neglecting transient and convective terms in momentum equation. hence the governing equations are now,

$$\nabla \cdot \sigma + \rho \mathbf{b} = 0$$

$$\nabla \cdot \mathbf{v} = 0$$

weighted residual integral equation can be written as,

$$\int_{\Omega} \mathbf{W} [\nabla \cdot \sigma + \rho \mathbf{b}] d\Omega = 0$$

using, $\nabla(\sigma^T \mathbf{W}) = \mathbf{W}(\nabla \sigma) + \mathbf{W} : \sigma$ following relation is obtained,

$$\int_{\Omega} [\nabla \cdot (\mathbf{W}^T \sigma) - \mathbf{W}^T : \sigma + \rho \mathbf{W}^T \mathbf{b}] d\Omega = 0$$

finally weak form is obtained as follows,

$$\int_{\Omega} \mathbf{W}^T : \sigma d\Omega = \int_{\Gamma} \mathbf{W}^T \mathbf{t} d\Gamma + \int_{\Omega} \rho \mathbf{W}^T \mathbf{b} d\Omega \quad (4.6)$$

$$\int_{\Omega} \mathbf{W}_p \nabla \mathbf{v} d\Omega = 0 \quad (4.7)$$

4.2.2 Newton's Method

Although Picards Method of converging is hugely accepted and gives considerably accurate results, the rate of convergence is very low. So more sophisticated method can be implemented. One such method is Newtons iterative method also known as Newton Raphson iterative method. This method is peculiar as it linearises complete nonlinear term using the Gradient of that term. The matrix form of gradient term is called *Tagent Matrix* or *Jacobian Matrix*.

The procedure follows following analogy.

Say $G(U) = 0$ is a non-linear function in U . If one is to find numerical solution u then it is first assumed that at some iteration $k+1$ solution is converged or satisfactory such that,

$$G(u_{k+1}) = 0$$

Expanding using taylor's expansion,

$$G(u_k + \Delta u) = 0$$

$$G(u_k) + \left(\frac{\partial G}{\partial u} \right)_{u_k} \Delta u + \dots = 0$$

neglecting the higher order terms,

$$u_{k+1} - u_k = \Delta u = - \left(\frac{\partial G}{\partial u} \right)_{u_k}^{-1} G(u_k)$$

hence the new solution is,

$$u_{k+1} = u_k - \left(\frac{\partial G}{\partial u} \right)_{u_k}^{-1} G(u_k)$$

If \mathbf{J} is to denote *Jacobian* of \mathbf{G} i.e. $(\frac{\partial G}{\partial u})$ then final equation can be written as

$$u_{k+1} = u_k - [\mathbf{J}(u_k)]^{-1} \cdot \{\mathbf{G}(u_k)\} \quad (4.8)$$

In the given context of 4 noded velocity and 4 noded pressure formulation \mathbf{G} can be interpreted as

$$\{\mathbf{G}_k\} = \int_{\Omega} \mathbf{B}^T \mathbf{DB} \hat{v}_c^k d\Omega + \int_{\Omega} \rho \mathbf{N}_c^T \mathbf{R}^{(k)} \mathbf{B}_{NL} \hat{v}_c^k d\Omega - \int_{\Gamma} \mathbf{N}_c^T \mathbf{t} d\Gamma - \int_{\Omega} \rho \mathbf{N}_c^T \mathbf{b} d\Omega$$

$$[\mathbf{J}_k] = \int_{\Omega} \mathbf{B}^T \mathbf{DB} d\Omega + \int_{\Omega} \rho \mathbf{N}^T \mathbf{R}^{(k)} \mathbf{B}_{NL} d\Omega + \begin{bmatrix} \int_{\Omega} \rho \mathbf{N}^T (\nabla \mathbf{v}_k) \mathbf{N} d\Omega & 0 \\ 0 & 0 \end{bmatrix}$$

where $(\nabla \mathbf{v}_k)$ can be given as,

$$\nabla \mathbf{v}_k = \begin{bmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} \end{bmatrix} = \begin{bmatrix} \hat{v}_{x_i} \frac{\partial N_i}{\partial x} & \hat{v}_{x_i} \frac{\partial N_i}{\partial y} \\ \hat{v}_{y_i} \frac{\partial N_i}{\partial x} & \hat{v}_{y_i} \frac{\partial N_i}{\partial y} \end{bmatrix}$$

Here \hat{v}_{x_i} denotes nodal values of velocities along x-direction while \hat{v}_{y_i} denotes nodal values of velocities along y-direction.

4.2.3 Formulation strategies

The Continuity Equation has only velocity as unknown variable, while Momentum Equation has both pressure and velocity as unknown variables. These variables can either be interpolated separately or using different order interpolation functions. Hence we develop multiple strategies for elementization such as,

- 4 Noded Velocity and 4 Noded Pressure formulation
- 9 Noded Velocity and 4 Noded Pressure formulation
- 4 Noded Velocity and Constant Pressure formulation

There are various advantages and disadvantages of them. However, in the present work only 9 Noded Velocity and 4 Noded Pressure formulation has been implemented.

4.3 Unequal Velocity and Pressure interpolation formulation

4.3.1 9-Noded Velocity and 4-Noded Pressure Formulation

From the last section as pressure showed mathematical imbalance in formulation while using equal degree interpolation functions for both velocity and pressure, it was suggested to use mixed formulation using different interpolation functions for velocity and pressure. In this section quadratic functions are used to interpolate velocity while linear functions are used to interpolate pressure.

Starting from governing equation and neglecting time dependent term for now.

$$\rho [(\nabla \mathbf{v}) \cdot \mathbf{v}] = \nabla \cdot \sigma + \rho \mathbf{b} \quad (4.9)$$

If \mathbf{W} is the weight function then,

$$\int_{\Omega} \mathbf{W}^T [\rho (\nabla \mathbf{v}) \cdot \mathbf{v} - \nabla \cdot \sigma - \rho \mathbf{b}] d\Omega = 0$$

Carrying out the same steps as in last section following form is obtained,

$$\int_{\Omega} \rho \mathbf{W}^T (\nabla \mathbf{v}) \cdot \mathbf{v} d\Omega + \int_{\Omega} \nabla \mathbf{W}^T : \sigma d\Omega = \int_{\Gamma} \mathbf{W}^T \cdot \mathbf{t} d\Gamma + \int_{\Omega} \rho \mathbf{W}^T \mathbf{b} d\Omega$$

After resolving $\nabla \mathbf{W}$ into symmetric and antisymmetric part and substituting constitutive relations 4.3, 4.4 and 5.7 we arrive at following expression,

$$\int_{\Omega} \rho \mathbf{W}^T (\nabla \mathbf{v}) \cdot \mathbf{v} d\Omega + \int_{\Omega} \left[\frac{1}{2} ((\nabla \mathbf{W}) + (\nabla \mathbf{W})^T) \right] \mathbf{D} \left[\frac{1}{2} ((\nabla \mathbf{v}) + (\nabla \mathbf{v})^T) \right] d\Omega + \int_{\Omega} \frac{1}{2} dia[(\nabla \mathbf{W}) + (\nabla \mathbf{W})^T] p d\Omega = \int_{\Gamma} \mathbf{W}^T \cdot \mathbf{t} d\Gamma + \int_{\Omega} \rho \mathbf{W}^T \mathbf{b} d\Omega \quad (4.10)$$

Where \mathbf{v} is velocity vector and p is pressure. They are interpolated using following functions.

$$\begin{aligned} v &= N^T \hat{v} \\ p &= H^T \hat{p} \\ W &= N^T \hat{u}_{\delta} \end{aligned} \quad (4.11)$$

Using these equations in equation 4.10 and again same analogy as in previous section, where $\mathbf{B} \cdot \hat{v}$ represents $\dot{\epsilon}$ or $\frac{1}{2}((\nabla \mathbf{v}) + (\nabla \mathbf{v})^T)$, following form is obtained,

$$\begin{aligned} \int_{\Omega} \hat{\mathbf{v}}_{\delta}^T \rho \mathbf{N} (\nabla \mathbf{v}) \mathbf{N}^T \hat{\mathbf{v}} d\Omega + \int_{\Omega} \hat{\mathbf{v}}_{\delta}^T \mathbf{B}^T \mathbf{D} \mathbf{B} \hat{\mathbf{v}} d\Omega + \int_{\Omega} \hat{\mathbf{v}}_{\delta}^T \mathbf{B}^T \mathbf{D}_p \mathbf{H}^T d\Omega &= \int_{\Gamma} \hat{\mathbf{v}}_{\delta}^T \mathbf{N}^T \cdot \mathbf{t} d\Gamma + \int_{\Omega} \hat{\mathbf{v}}_{\delta}^T \rho \mathbf{N}^T \mathbf{b} d\Omega \\ \text{or} \\ \int_{\Omega} \rho \mathbf{N} (\nabla \mathbf{v}) \mathbf{N}^T \hat{\mathbf{v}} d\Omega + \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} \hat{\mathbf{v}} d\Omega + \int_{\Omega} \mathbf{B}^T \mathbf{D}_p \mathbf{H}^T \hat{p} d\Omega &= \int_{\Gamma} \mathbf{N}^T \cdot \mathbf{t} d\Gamma + \int_{\Omega} \rho \mathbf{N}^T \mathbf{b} d\Omega \end{aligned} \quad (4.12)$$

Here \mathbf{D}_p makes sure that pressure terms are added only in diagonal terms. Similarly if γ is used as the weight function for continuity equation 4.1 then weak form of continuity equation can be written as,

$$\int_{\Omega} \gamma^T \nabla \cdot \mathbf{v} d\Omega = 0$$

If the structure of $\nabla \cdot \mathbf{v}$ is observed then it can be replaced by $tr(\nabla \mathbf{v})$. which can in turn be $-\mathbf{D}_p^T \mathbf{B} \hat{\mathbf{v}}$. Hence if following weight function is chosen for continuity equation and multiply throughout by '1' then formulation becomes consistent.

$$\gamma = \mathbf{H}^T \beta_{\delta}$$

The weak form of continuity equation becomes,

$$\begin{aligned} \int_{\Omega} \beta_{\delta}^T \mathbf{H} \mathbf{D}_p^T \mathbf{B} \hat{\mathbf{v}} d\Omega &= 0 \\ \text{or} \\ \int_{\Omega} \mathbf{H} \mathbf{D}_p^T \mathbf{B} \hat{\mathbf{v}} d\Omega &= 0 \end{aligned} \quad (4.13)$$

If we write equations (4.12) and (4.13) together in matrix form we get.

$$\begin{bmatrix} \int_{\Omega} \rho \mathbf{N} (\nabla \mathbf{v}) \mathbf{N}^T d\Omega + \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega & \int_{\Omega} \mathbf{B}^T \mathbf{D}_p \mathbf{H}^T d\Omega \\ \int_{\Omega} \mathbf{H} \mathbf{D}_p^T \mathbf{B} d\Omega & 0 \end{bmatrix} \begin{Bmatrix} \hat{\mathbf{v}} \\ \hat{p} \end{Bmatrix} = \begin{Bmatrix} \int_{\Gamma} \mathbf{N}^T \cdot \mathbf{t} d\Gamma + \int_{\Omega} \rho \mathbf{N}^T \mathbf{b} d\Omega \\ 0 \end{Bmatrix} \quad (4.14)$$

This form is very helpful in visualizing the problem and get solution using Picards iterative method. By dropping out the term $\int_{\Omega} \rho \mathbf{N} (\nabla \mathbf{v}) \mathbf{N}^T d\Omega$ one can simulated stokes flow as well. But to implement Newton's iterative method Jacobian matrix and Force residue matrices must be found out. Using earlier analogy of $(k+1)$ as unknown iteration and k as known iteration, Jacobian matrix can be obtained.

$$\mathbf{J} = \begin{bmatrix} \mathbf{K}_v + \mathbf{C}_v + \mathbf{J}_v & \mathbf{K}_p \\ \mathbf{K}_p^T & 0 \end{bmatrix} \quad (4.15)$$

where,

$$\mathbf{K}_v = \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} d\Omega$$

$$\mathbf{C}_v = \int_{\Omega} \rho \mathbf{N}(\nabla \mathbf{v}^k) \mathbf{N}^T d\Omega$$

$$\mathbf{J}_v = \int_{\Omega} \rho \mathbf{N} \begin{bmatrix} \mathbf{v}_x^k \mathbf{N}_{,x}^T + \mathbf{v}_y^k \mathbf{N}_{,y}^T & 0 \\ 0 & \mathbf{v}_x^k \mathbf{N}_{,x}^T + \mathbf{v}_y^k \mathbf{N}_{,y}^T \end{bmatrix} d\Omega$$

$$\mathbf{K}_p = \int_{\Omega} \mathbf{B}^T \mathbf{D}_p \mathbf{H}^T d\Omega$$

Residue matrix is nothing but $\mathbf{G}(u)$ of the equation (4.8) i.e. the non-linear function evaluated at approximated solution itself and in this case it is given as,

$$\mathbf{F} = \begin{Bmatrix} \mathbf{F}_v \\ \mathbf{F}_p \end{Bmatrix} \quad (4.16)$$

where,

$$\mathbf{F}_v = \int_{\Omega} \rho \mathbf{N}(\nabla \mathbf{v}^k) \mathbf{N}^T \hat{\mathbf{v}}^k d\Omega + \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} \hat{\mathbf{v}}^k d\Omega + \int_{\Omega} \mathbf{B}^T \mathbf{D}_p \mathbf{H}^T \hat{p}^k d\Omega - \int_{\Gamma} \mathbf{N}^T \cdot \mathbf{t} d\Gamma - \int_{\Omega} \rho \mathbf{N}^T \mathbf{b} d\Omega$$

$$\mathbf{F}_p = \int_{\Omega} \mathbf{H} \mathbf{D}_p^T \mathbf{B} \hat{\mathbf{v}}^k d\Omega$$

Regarding the 9 noded velocity and 4 noded pressure element, a particular way of identifying and denoting the degrees of freedom was used. The figure (4.1) shows he dof's of Quadratic-Linear element.

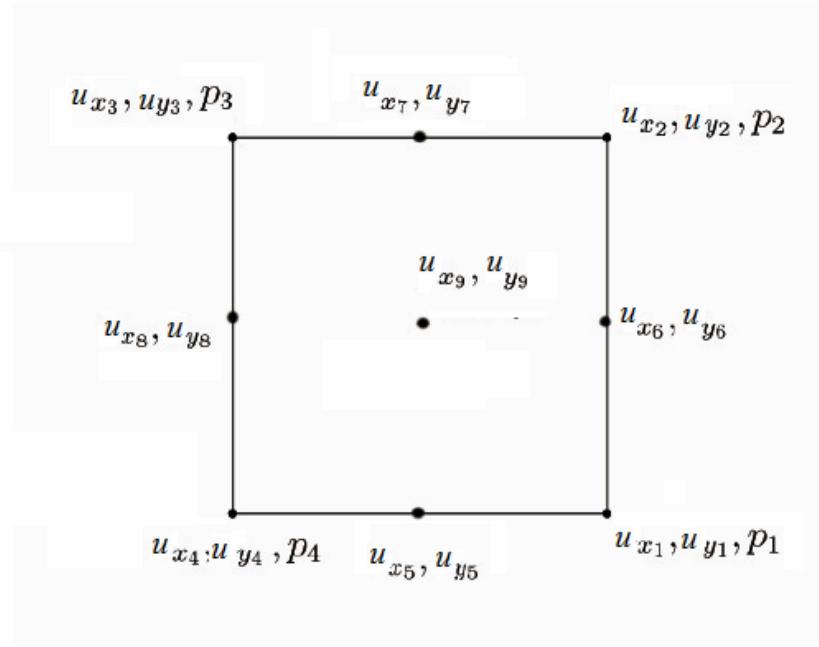


Figure 4.1: Degrees of freedom of quadratic-velocity-linear-pressure element

The different Matrices used in above formulation for quadratic velocity and linear pressure may be given as,

$$\mathbf{v} = \begin{Bmatrix} v_x \\ v_y \end{Bmatrix} \quad \mathbf{H} = \begin{Bmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \end{Bmatrix} \quad \mathbf{N} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & \dots & N_9 & 0 \\ 0 & N_1 & 0 & N_2 & \dots & 0 & N_9 \end{bmatrix} \quad \hat{\mathbf{v}} = \begin{Bmatrix} v_{x1} \\ v_{y1} \\ v_{x2} \\ v_{y2} \\ \dots \\ v_{x9} \\ v_{y9} \end{Bmatrix} \quad \hat{\mathbf{p}} = \begin{Bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{Bmatrix} \quad (4.17)$$

$$\nabla \mathbf{v} = \begin{bmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{N} \mathbf{n}^T}{\partial x} \cdot \hat{\mathbf{v}}_x & \frac{\partial \mathbf{N} \mathbf{n}^T}{\partial y} \cdot \hat{\mathbf{v}}_x \\ \frac{\partial \mathbf{N} \mathbf{n}^T}{\partial x} \cdot \hat{\mathbf{v}}_y & \frac{\partial \mathbf{N} \mathbf{n}^T}{\partial y} \cdot \hat{\mathbf{v}}_y \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} N_{1,x} & 0 & N_{2,x} & 0 \dots & N_{9,x} & 0 \\ 0 & N_{1,y} & 0 & N_{2,y} \dots & 0 & N_{9,y} \\ N_{1,y} & N_{1,x} & N_{2,y} & N_{2,x} \dots & N_{9,y} & N_{9,x} \end{bmatrix} \quad (4.18)$$

$$\mathbf{D} = \begin{bmatrix} 2\mu & 0 & 0 \\ 0 & 2\mu & 0 \\ 0 & 0 & \mu \end{bmatrix}$$

$$\mathbf{D}_p = \begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix}$$

4.3.2 Flow Through sudden contraction

A flow through a channel in which at the outlet the channel contracts suddenly has been simulated using the fluid formulation for steady state. The problem statement for the simulation has been shown in fig. 4.2 It has been simulated over a 9-node velocity 4-node pressure element.

Problem Statement

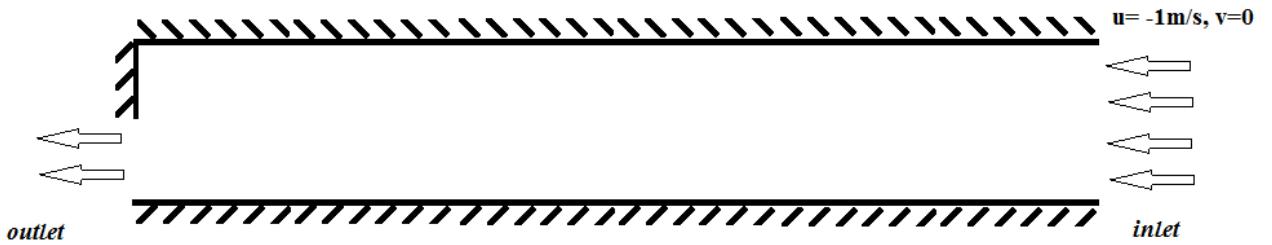


Figure 4.2: Problem Statement for the flow through sudden contraction problem

Mesh Used

A 40x15 structured and orthogonal mesh has been used for the simulation. The mesh used to simulate the problem has been shown in the fig. 4.3

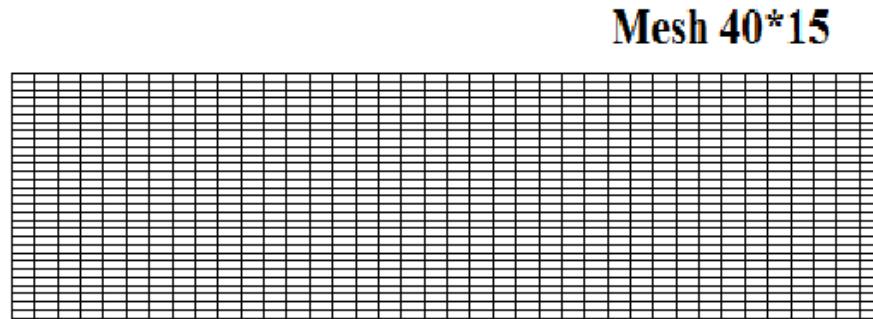


Figure 4.3: Mesh used to simulate Flow through sudden Contraction problem

Results

The results of the simulation have been shown in the following figures. The fig. 4.4(a) and 4.4(b) show the contours of u- and v- velocities over the domain. The fig. 4.4(c) shows the streamlines in the domain and the fig. 4.4(d) shows the vectors in the computational domain.

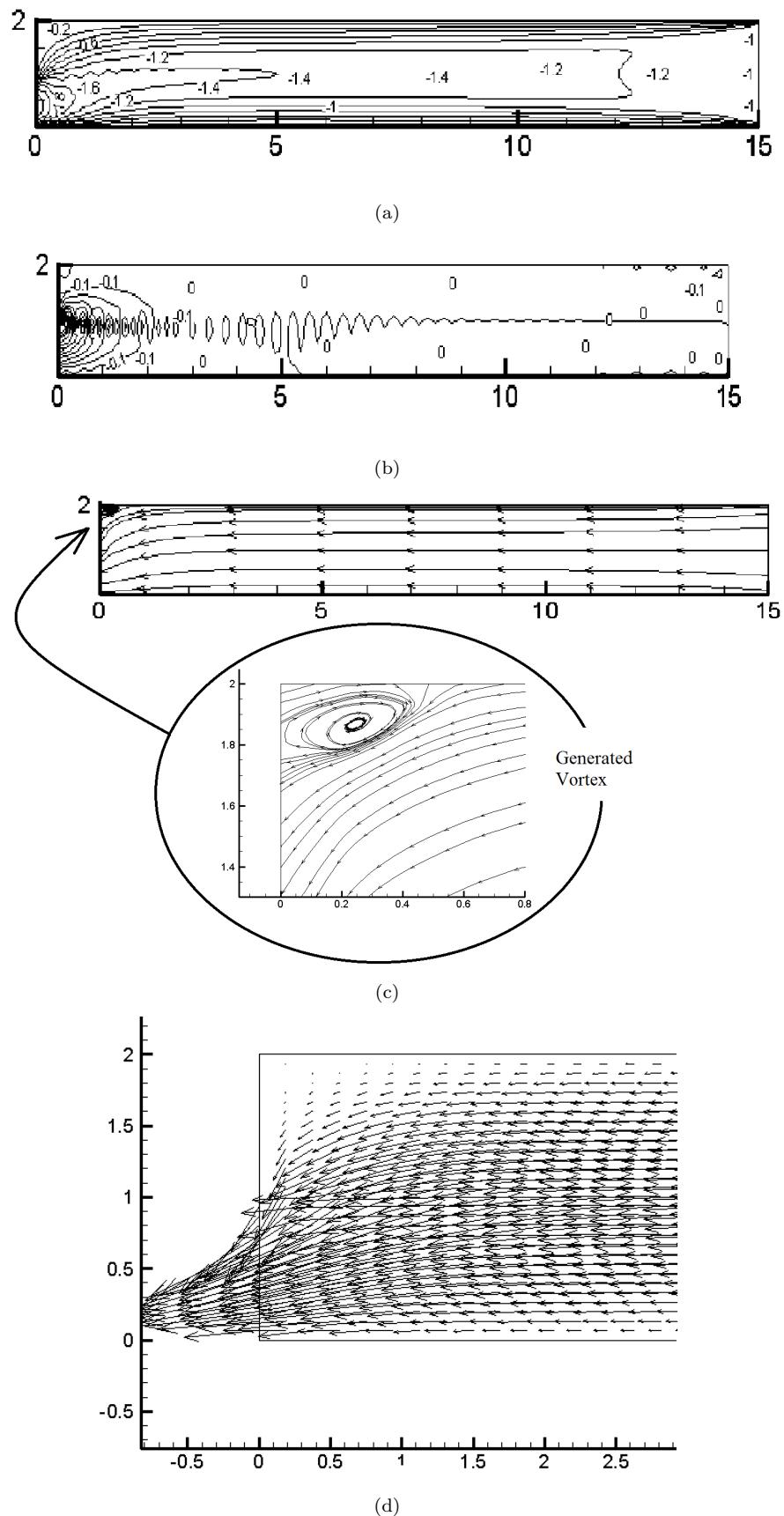


Figure 4.4: (a)u-velocity contour(b)v-velocity contour(c)Streamlines(d)Vector Plot at outlet

4.4 Transient Analysis

The most general form of governing shown in equation(4.2)contains time derivative of primary variable. Hence in all, momentum equation is a non-linear time dependent ordinary differential equation.

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} \right) + \rho \left((\nabla \mathbf{v}) \cdot \mathbf{v} \right) = \nabla \cdot \sigma + \rho \mathbf{b}$$

Now doing all the common steps as that in previous section i.e. multiplying by weight function and then integrating by parts we get following equation,

$$\int_{\Omega} \rho \mathbf{N} \mathbf{N}^T \dot{\mathbf{v}} d\Omega + \int_{\Omega} \rho \mathbf{N} (\nabla \mathbf{v}) \mathbf{N}^T \dot{\mathbf{v}} d\Omega + \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} \dot{\mathbf{v}} d\Omega + \int_{\Omega} \mathbf{B}^T \mathbf{D}_p \mathbf{H}^T \dot{p} d\Omega = \int_{\Gamma} \mathbf{N}^T \cdot \mathbf{t} d\Gamma + \int_{\Omega} \rho \mathbf{N}^T \mathbf{b} d\Omega \quad (4.19)$$

In this equation $\dot{\mathbf{v}}$ is time continuous space discretized velocity vector i.e.

$$\dot{\mathbf{v}} = [\dot{v}_{x_1} \quad \dot{v}_{x_2} \quad \dots \quad \dot{v}_{x_9} \quad \dot{v}_{y_1} \quad \dot{v}_{y_2} \quad \dots \quad \dot{v}_{y_9}]$$

but it is not primary variable directly discretizable, hence it is necessary to employ discrete approximation for time derivatives. Conventionally, it is possible to discretize time with finite elements but they yield same end results as finite difference time integration scheme so we have applied finite difference time discretization scheme in our formulation. First of all the given time interval is discretized in smaller segments called *timesteps*.

$$\Delta t = t^{n+1} - t^n$$

where superscript n and $n + 1$ denotes the time steps viz. previous and current respectively. There are various implicit semi-implicit and explicit schemes available for time discretization. The most commonly used time integration methods are part of a one-parameter family, called α -family approximations.

$$v^{n+1} = v^n + \Delta t[(1 - \alpha)\dot{v}^n + \alpha\dot{v}^{n+1}] \quad (4.20)$$

One can generalize time discretization process as follows. Consider the following form of time continuous space discretized form of equation,

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{K}\mathbf{v} = \mathbf{f}$$

This equation can be converted into a set of algebraic equations using formula in Eq(4.20):

$$\hat{\mathbf{K}}^{(n+1)} v^{(n+1)} = \hat{\mathbf{F}}^{(n,n+1)} \quad (4.21)$$

where

$$\begin{aligned} \hat{\mathbf{K}}^{(n+1)} &= \mathbf{M} + \alpha \Delta t \mathbf{K}^{(n+1)} \\ \hat{\mathbf{F}}^{(n,n+1)} &= \Delta t [\alpha \mathbf{F}^{(n+1)} + (1 - \alpha) \mathbf{F}^n] + [\mathbf{M} - ((1 - \alpha) \Delta t) \mathbf{K}^n] v^n \end{aligned}$$

Starting from initial solution, after assemble and applying boundary conditions this equation can be solved at each time step. for different values of α we obtain different time approximation schemes:

$$\alpha = 0 \quad \textit{Forward difference scheme (conditionally stable)} \quad (4.22)$$

$$\alpha = 0.5 \quad \textit{Crank – Nicholson scheme (unconditionally stable)} \quad (4.23)$$

$$\alpha = 2/3 \quad \textit{Galerkin scheme (unconditionally stable)} \quad (4.24)$$

$$\alpha = 1 \quad \textit{Backward difference scheme (unconditionally stable)} \quad (4.25)$$

This methods can be combined with predictor-corrector method. In such method to march rapidly in non-linearity, a predictor step is used which gives an approximate solution near to exact solution calculated from previous time steps' solution. This saves us from effort of calculating the non-linear

solution from zero initial solution and also save computational time by converging in lesser number of steps.

In given context of equation (4.19) we can say that it takes following form:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{K}(\hat{\mathbf{v}})\hat{\mathbf{v}} = \mathbf{f} \quad (4.26)$$

We will use first order integration method that is unconditionally stable. One such simplest method is backward Euler method. When this method is applied to above equation it yields following result

$$\left[\frac{1}{\Delta t} \mathbf{M} + \mathbf{K}(\hat{\mathbf{v}}^{n+1}) \right] \hat{\mathbf{v}}^{n+1} = \frac{1}{\Delta t} \mathbf{M}\hat{\mathbf{v}}^n + \mathbf{f}(\hat{\mathbf{v}}^{n+1}) \quad (4.27)$$

where

$$\mathbf{M} = \begin{bmatrix} \int_{\Omega} \rho \mathbf{N} \mathbf{N}^T d\Omega & 0 \\ 0 & 0 \end{bmatrix} \quad (4.28)$$

$$\mathbf{K} = \begin{bmatrix} \int_{\Omega} (\rho \mathbf{N}(\nabla \mathbf{v}) \mathbf{N}^T + \mathbf{B}^T \mathbf{D} \mathbf{B}) d\Omega & \int_{\Omega} \mathbf{B}^T \mathbf{D}_p \mathbf{H}^T d\Omega \\ \int_{\Omega} \mathbf{H} \mathbf{D}_p^T \mathbf{B} d\Omega & 0 \end{bmatrix} \quad (4.29)$$

$$\mathbf{f} = \begin{bmatrix} \int_{\Gamma} \mathbf{N}^T \cdot \mathbf{t} d\Gamma + \int_{\Omega} \rho \mathbf{N}^T \mathbf{b} d\Omega \\ 0 \end{bmatrix} \quad (4.30)$$

This form is useful while solving nonlinear equation using Picard iterative method. But to solve using Newton's iterative method it takes following form:

$$v_{k+1}^{(n+1)} = v_k^{(n+1)} - [\mathbf{J}(v_k^{(n+1)})]^{-1} \cdot \{\mathbf{F}(v_k^{(n+1)}, v_{final}^{(n)})\} \quad (4.31)$$

Here subscript k marches in nonlinearity while superscript n marches in time. Hence it can be seen as every time step has defined number of iterations to solve the non-linearity.

In the above equation (4.31) the Jacobian \mathbf{J} and \mathbf{F} stands for following expressions:

$$\mathbf{J}(v_k^{(n+1)}) = \begin{bmatrix} \frac{1}{\Delta t} \mathbf{M} + \mathbf{K}_v + \mathbf{C}_v(v_k^{(n+1)}) + \mathbf{J}_v(v_k^{(n+1)}) & \mathbf{K}_p \\ \mathbf{K}_p^T & 0 \end{bmatrix} \quad (4.32)$$

where \mathbf{K}_v , \mathbf{C}_v , \mathbf{J}_v and \mathbf{K}_p are the same as that in equation (4.15) while \mathbf{M} can be Eq(4.28) and

$$\mathbf{F}(v_k^{(n+1)}, v_{final}^{(n)}) = \begin{Bmatrix} F_v \\ F_p \end{Bmatrix} \quad (4.33)$$

$$\begin{aligned} \mathbf{F}_v &= \int_{\Omega} \rho \mathbf{N} \mathbf{N}^T \left(\frac{\hat{\mathbf{v}}_k^{n+1} - \hat{\mathbf{v}}^n}{\Delta t_n} \right) d\Omega + \int_{\Omega} \rho \mathbf{N}(\nabla \mathbf{v}_k^{n+1}) \mathbf{N}^T \hat{\mathbf{v}}_k^{n+1} d\Omega + \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} \hat{\mathbf{v}}_k^{n+1} d\Omega \\ &\quad + \int_{\Omega} \mathbf{B}^T \mathbf{D}_p \mathbf{H}^T \hat{\mathbf{p}}_k^{n+1} d\Omega - \int_{\Gamma} \mathbf{N}^T \cdot \mathbf{t} d\Gamma - \int_{\Omega} \rho \mathbf{N}^T \mathbf{b} d\Omega \\ \mathbf{F}_p &= \int_{\Omega} \mathbf{H} \mathbf{D}_p^T \mathbf{B} \hat{\mathbf{v}}_k^{n+1} d\Omega \end{aligned}$$

In this way whole formulation for fluid element was completed. Then we studied two cases using this formulation to examine the validity of formulation. Here are the two case studies:

4.4.1 Lid Driven Cavity Problem

Problem Statement and Mesh used

The well known benchmark problem Lid Driven Cavity has been simulated using above formulation. The problem statement has been shown in the figure (4.5)(a), where a 2D cavity was filled with a fluid with given properties and three edges of the cavity was fixed with the free top edge at which a plate was being moved with a constant given velocity.

The problem has been simulated at Reynolds No. 400 and by using an orthogonal mesh of size 16*16 shown in the figure (4.5)(b).

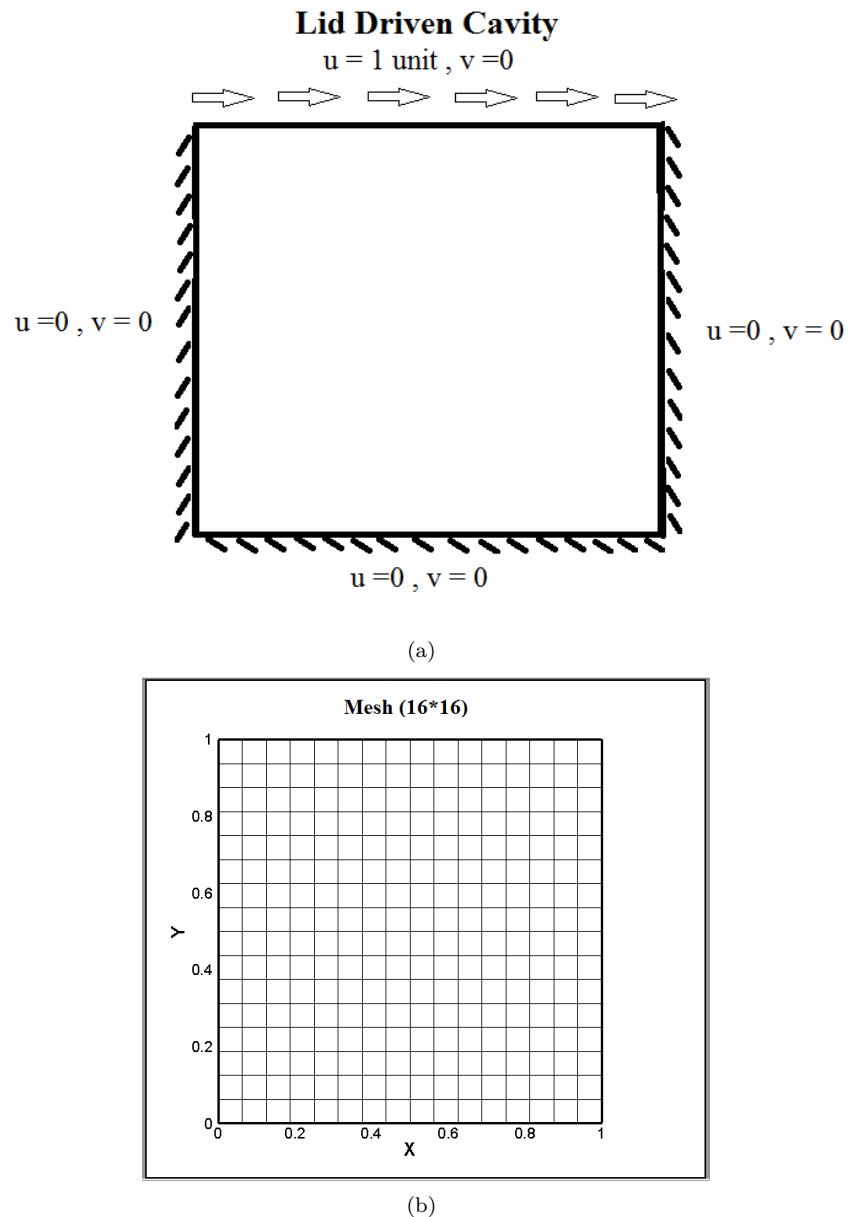


Figure 4.5: (a)Problem Statement (b)Mesh

4.4.2 Contours, Vector Plot and Streamlines at Steady State

Figure (4.6)(a) and Figure (4.6)(b) show the u- and v- velocity contours after the steady state has been achieved while figure (4.6)(c) shows the vector plot of the velocities. Also figure (4.6)(d) shows the streamlines after the steady state. The time step used was 0.1 sec and for time marching Backward Euler method has been used.

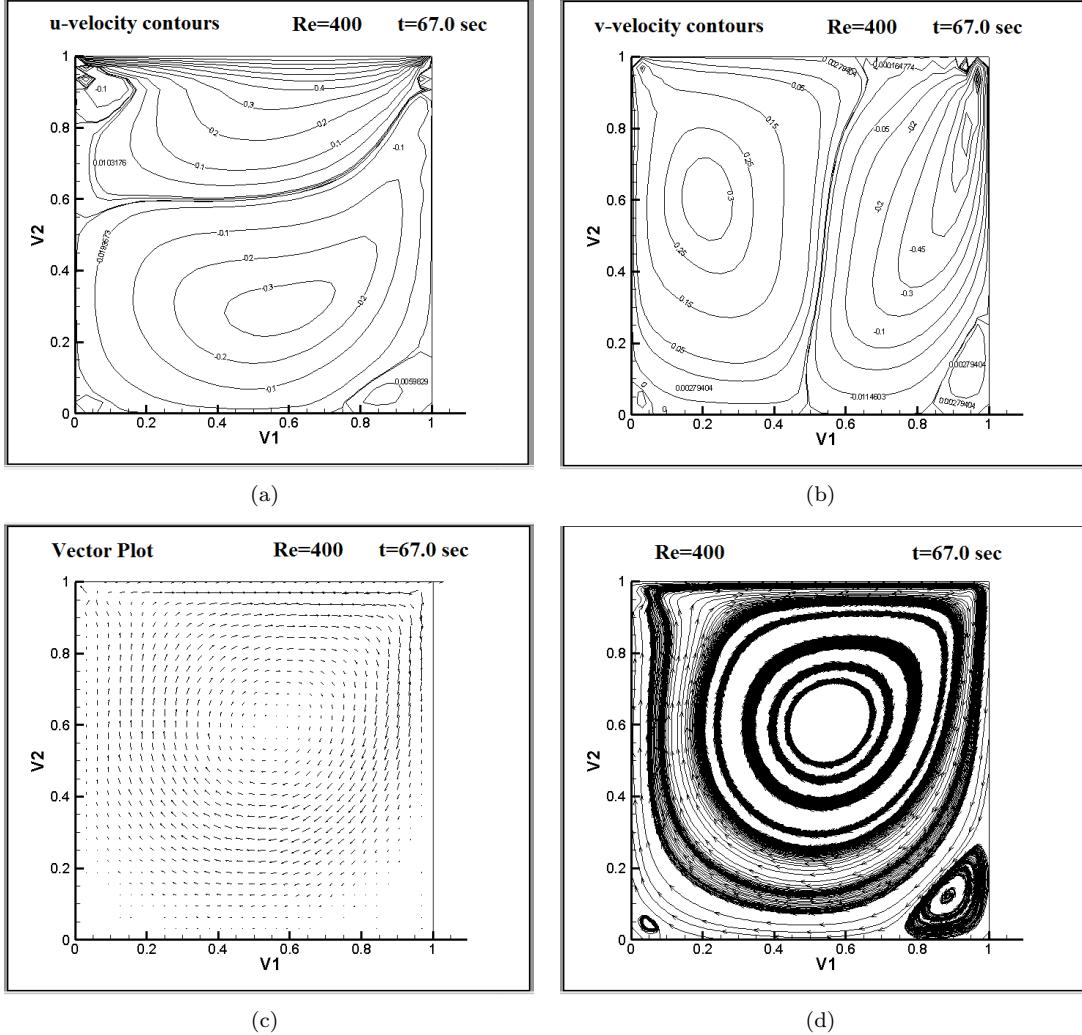


Figure 4.6: (a)u-velocity contour (b)v-velocity contour (c)Vector Plot (d)Streamlines

4.4.3 Streamlines at given time levels during time marching

To understand the development of the flow the streamlines have been plotted at different time levels. The streamlines are shown by the fig. 4.7(a) at time 0.3 sec, fig. 4.7(b) at time 1.0 sec, fig. 4.7(c) at time 3.0 sec, fig. 4.7(d) at time 6.0 sec, fig. 4.7(e) at time 20.0 sec and fig. 4.7(f) at time 67.0 sec which is eventually the steady state of the flow.

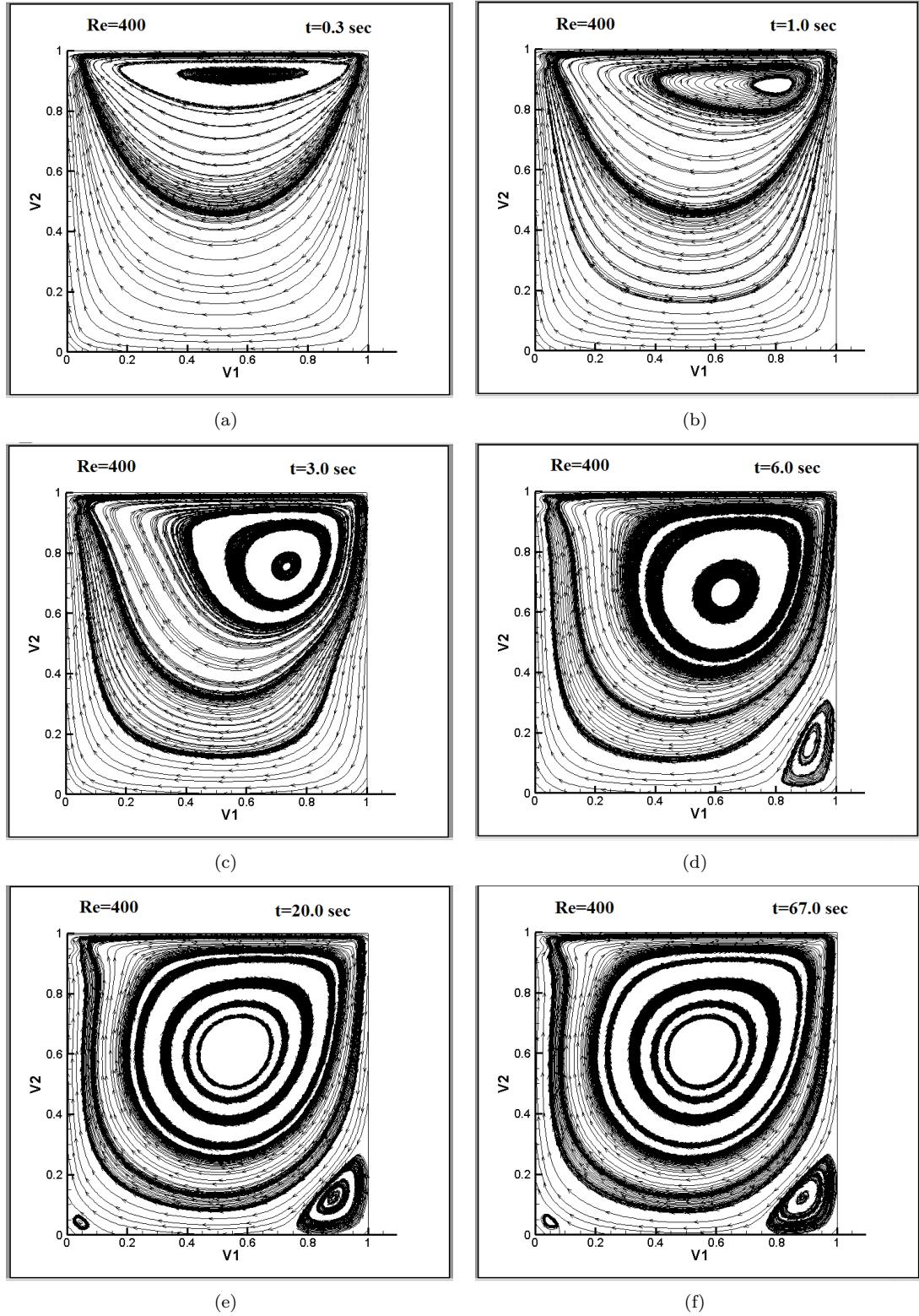


Figure 4.7: Streamlines at different time levels

4.5 Closure

In this chapter we carried the same procedure, as in solids in last chapter, for fluid dynamic equations. Here we used continuous pressure method to formulate the velocity and pressure field. We tackled problem of non linearity in the equation using Newton Raphson method. But equal interpolation function created mathematical imbalance which was tacked using unequal interpolation functions. Further, while doing various simulations we came across the fact of stability of flow with mesh and simulated proper flows for the computationally affordable mesh. Close observation has also inferred that Quadratic-Linear formulation is more efficient than Linear-constant as a mesh of 12x12 grid of Quadratic-Linear formulation gave same results as results of 30 x 30 mesh of Linear-Constant formulation. Furthermore, We applied our code to Lid Driven cavity problem and also to flow past cylinder and simulated the phenomenon of vortex shedding.

Chapter 5

Fluid-Structure Interaction Problem

5.1 Governing Equations

5.1.1 Momentum Equation

Considering a computational domain in an Eulerian framework the necessary governing equations are

$$\rho_f \left[\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right] = \nabla \cdot \boldsymbol{\sigma} + \rho_f \mathbf{b} \quad \text{in } \bar{\Omega} \quad (5.1)$$

5.1.2 Conservation of mass or Continuity Equation

$$\nabla \cdot \mathbf{v} = 0 \quad (5.2)$$

5.1.3 Velocity Displacement Relation

To compute the displacement field the following relation has been used by which the displacement field for the entire computational domain will be computed

$$\mathbf{v} = \frac{D\mathbf{u}}{Dt} = \frac{\partial \mathbf{u}}{\partial t} + (\nabla \mathbf{u}) \cdot \mathbf{v} \quad (5.3)$$

5.2 Constitutive Relations

The constitutive relations may be given by,

5.2.1 Solids

$$\boldsymbol{\sigma} = \mathbf{D} \cdot \boldsymbol{\epsilon} \quad (5.4)$$

with the strain displacement relation as

$$\boldsymbol{\epsilon} = \frac{1}{2} [(\nabla \mathbf{u}) + (\nabla \mathbf{u})^T] \quad (5.5)$$

5.2.2 Fluids

$$\boldsymbol{\sigma} = -p \mathbf{I} + 2\mu \dot{\boldsymbol{\epsilon}} \quad (5.6)$$

with strain rate velocity relation as

$$\dot{\boldsymbol{\epsilon}} = \frac{1}{2} [(\nabla \mathbf{v}) + (\nabla \mathbf{v})^T] \quad (5.7)$$

5.3 Development of Visco-Elastic Element

The momentum equation is valid for both fluid and solid. The used governing equation consists of only velocities.

$$\rho \left[\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right] = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{b} \quad \text{in } \bar{\Omega} \quad (5.8)$$

Coupling between the fluid and solid domain can be done at constitutive level. If in a given element α is the volume fraction of fluid and β is the volume fraction of solid then the net stress for the element can be given as

$$\boldsymbol{\sigma} = \alpha \boldsymbol{\sigma}_f + \beta \boldsymbol{\sigma}_s \boldsymbol{\sigma} = -p \mathbf{I} + \alpha \mathbf{D}_f \dot{\epsilon} + \beta \mathbf{D}_s \epsilon \quad (5.9)$$

as α and β are the volume fractions for fluid and solid respectively in a given element hence

$$\alpha + \beta = 1 \quad (5.10)$$

always follows. Also both fluid and solid are incompressible hence $-p \mathbf{I}$ terms is included in both stresses and a pressure field will be computed for the entire domain, both fluid and solid.

for a typical 2-D problem the above equation will take form as:

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{Bmatrix} = \begin{Bmatrix} -1 \\ -1 \\ 0 \end{Bmatrix} p + \alpha \begin{bmatrix} 2\mu & 0 & 0 \\ 0 & 2\mu & 0 \\ 0 & 0 & \mu \end{bmatrix} \begin{Bmatrix} \dot{\epsilon}_{xx} \\ \dot{\epsilon}_{yy} \\ \dot{\epsilon}_{xy} \end{Bmatrix} + \beta \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{Bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{xy} \end{Bmatrix} \quad (5.11)$$

In the overall formulation velocity displacement relation and continuity equation are also valid for both fluid and solid domains.

$$\mathbf{v} = \frac{D\mathbf{u}}{Dt} = \frac{\partial \mathbf{u}}{\partial t} + (\nabla \mathbf{u}) \cdot \mathbf{v} \quad (5.12)$$

$$\nabla \cdot \mathbf{v} = 0 \quad (5.13)$$

This way our model is ready to be analysed further.

5.3.1 Weak Form

Now we will multiply the equations by the corresponding volume fractions and integrate over the domain.

$$\int_{\Omega} \mathbf{W}^T \left[\rho \left(\frac{\partial \mathbf{v}}{\partial t} \right) + \rho ((\nabla \mathbf{v}) \cdot \mathbf{v}) - \nabla \cdot \boldsymbol{\sigma} - \rho \mathbf{b} \right] d\Omega = 0$$

Further without using the constitutive relations we will first integrate the stress term $\nabla \cdot \boldsymbol{\sigma}$ by parts with weight function to make it radical free. The detailed derivation is exactly same that of a transient fluid equation formulation. Here is the end weak form obtained:

$$\int_{\Omega} \mathbf{W}^T \rho \left(\frac{\partial \mathbf{v}}{\partial t} \right) d\Omega + \int_{\Omega} \rho \mathbf{W}^T (\nabla \mathbf{v}) \cdot \mathbf{v} d\Omega + \int_{\Omega} \nabla \mathbf{W}^T : \boldsymbol{\sigma} d\Omega = \int_{\Gamma} \mathbf{W}^T \cdot \mathbf{t} d\Gamma + \int_{\Omega} \rho \mathbf{W}^T \mathbf{b} d\Omega \quad (5.14)$$

Let $\boldsymbol{\sigma}$ be taken care afterwards for now let us discretize the weight function \mathbf{W} and \mathbf{v} as:

$$\mathbf{W} = \mathbf{N}^T \hat{\mathbf{v}}_{\delta} \quad \mathbf{v} = \mathbf{N}^t \hat{\mathbf{v}} \quad (5.15)$$

Again converting ∇W into symmetric and anti symmetric parts (i.e. $\nabla W = \frac{1}{2}(\nabla W^T + \nabla W) + \frac{1}{2}(\nabla W^T - \nabla W)$) we can eliminate anti-symmetric part as $\boldsymbol{\sigma}$ is symmetric.

From each terms nodal virtual velocity vector $\hat{\mathbf{v}}_{\delta}$ can be taken out. Arguing again that it is completely

arbitrary given it belongs to set of admissible velocities, we can equate remaining part to zero. Finally we arrive at following weak equation:

$$\int_{\Omega} \rho \mathbf{N} \mathbf{N}^T \hat{\mathbf{v}} d\Omega + \int_{\Omega} \rho \mathbf{N} (\nabla \mathbf{v}) \mathbf{N}^T \hat{\mathbf{v}} d\Omega + \int_{\Omega} \frac{1}{2} (\nabla \mathbf{N}^T + \nabla \mathbf{N}) : \boldsymbol{\sigma} d\Omega = \int_{\Gamma} \mathbf{N}^T \cdot \mathbf{t} d\Gamma + \int_{\Omega} \rho \mathbf{N}^T \mathbf{b} d\Omega \quad (5.16)$$

Consider the term $\int_{\Omega} \frac{1}{2} (\nabla \mathbf{N}^T + \nabla \mathbf{N}) : \boldsymbol{\sigma} d\Omega$. Now we substitute constitutive relations in it. Here we use same interpolation function for velocity as well as displacement but not for pressure.

$$\begin{aligned} \mathbf{u} &= \mathbf{N}^T \hat{\mathbf{u}} \\ p &= \mathbf{H}^T \hat{p} \end{aligned}$$

Using these interpolation functions we get

$$\begin{aligned} &\int_{\Omega} \frac{1}{2} (\nabla \mathbf{N}^T + \nabla \mathbf{N}) : \boldsymbol{\sigma} d\Omega \\ &= \int_{\Omega} \frac{1}{2} (\nabla \mathbf{N}^T + \nabla \mathbf{N}) : \left(-p \mathbf{I} + 2\mu \frac{1}{2} (\nabla \mathbf{v}^T + \nabla \mathbf{v}) + \mathbf{D}_s \frac{1}{2} (\nabla \mathbf{u}^T + \nabla \mathbf{u}) \right) d\Omega \\ &= \int_{\Omega} \mathbf{B}^T \mathbf{D}_p \mathbf{H}^T \hat{p} d\Omega + \int_{\Omega} \mathbf{B}^T \mathbf{D}_f \mathbf{B} \hat{\mathbf{v}} d\Omega + \int_{\Omega} \mathbf{B}^T \mathbf{D}_s \mathbf{B} \hat{\mathbf{u}} d\Omega \end{aligned}$$

putting this back in equation 5.16 we get,

$$\mathbf{M}_v \frac{\partial \hat{\mathbf{v}}}{\partial t} + \mathbf{C}_v \hat{\mathbf{v}} + \mathbf{K}_v \hat{\mathbf{v}} + \mathbf{K}_p \hat{p} + \mathbf{K}_u \hat{\mathbf{u}} = \mathbf{f} \quad (5.17)$$

Similarly, skipping all the details of the derivation, the continuity equation can directly be written as:

$$\mathbf{K}_p^T \hat{\mathbf{v}} = 0 \quad (5.18)$$

Now the newer equation is the velocity-displacement equation. If \mathbf{W}_z is considered as the weight function then our equation will look as:

$$\int \mathbf{W}_z^T \left[\frac{\partial \mathbf{u}}{\partial t} + (-\mathbf{v} + (\nabla \mathbf{u}) \cdot \mathbf{v}) \right] d\Omega = 0 \quad (5.19)$$

If $\mathbf{W}_z^T = \zeta^T \beta_\delta$ then eliminating β , it being completely arbitrary in admissible set, we get

$$\int \zeta \left[\frac{\partial \mathbf{u}}{\partial t} + (-\mathbf{I} + \nabla \mathbf{u}) \cdot \mathbf{v} \right] d\Omega = 0 \quad (5.20)$$

substituting the interpolation function and writting equation in nodal unknowns' form:

$$\int \zeta \mathbf{N}^T \frac{\partial \hat{\mathbf{u}}}{\partial t} d\Omega - \int \zeta \mathbf{N}^T \hat{\mathbf{v}} d\Omega + \int \zeta (\nabla \mathbf{u}) \cdot \mathbf{N}^T \hat{\mathbf{v}} d\Omega = 0 \quad (5.21)$$

If we observe this relation closely we can say that ζ is similar to interpolation function of displacement \mathbf{u} i.e. in this case \mathbf{N} . This is also necessary to maintain the symmetry and squareness of final set of algebraic equations.Hence above equation takes following form

$$\mathbf{M}_z \frac{\partial \hat{\mathbf{u}}}{\partial t} - \mathbf{M}_z \hat{\mathbf{v}} + \mathbf{C}_u \hat{\mathbf{v}} = 0 \quad (5.22)$$

Now if we combine equations (5.17),(5.18) and (5.22) we get following set of equations:

$$\begin{bmatrix} \mathbf{M}_v & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \mathbf{M}_z \end{bmatrix} \begin{Bmatrix} \hat{\mathbf{v}} \\ \hat{p} \\ \hat{\mathbf{u}} \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_v + \mathbf{C}_v & \mathbf{K}_p & \mathbf{K}_u \\ \mathbf{K}_p^T & 0 & 0 \\ -\mathbf{M}_z + \mathbf{C}_u & 0 & 0 \end{bmatrix} \begin{Bmatrix} \hat{\mathbf{v}} \\ \hat{p} \\ \hat{\mathbf{u}} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f} \\ 0 \\ 0 \end{Bmatrix} \quad (5.23)$$

This is a set of transient non-linear equations of form:

$$[\mathbf{M}]\{\dot{\mathbf{v}}\} + [\mathbf{K}(\mathbf{v})]\{\mathbf{v}\} = \{\mathbf{f}\}$$

5.3.2 Linearised Formulation for Quadratic-Velocity Linear-Pressure and Quadratic-Displacement Formulation

To discretize in time we will use backward Euler Scheme for both $\dot{\mathbf{u}}$ and $\dot{\mathbf{v}}$ and we will implement Newton's iterative scheme to linearise the equations.

$$\frac{\partial v}{\partial t} = \frac{\hat{v}^{n+1} - \hat{v}^n}{\Delta t_n} , \quad \frac{\partial u}{\partial t} = \frac{\hat{u}^{n+1} - \hat{u}^n}{\Delta t_n} \quad (5.24)$$

This will convert Eq.(5.23) into following form:

$$\begin{bmatrix} \frac{1}{\Delta t} \mathbf{M} + \mathbf{K}_v + \mathbf{C}_v & \mathbf{K}_p & \mathbf{K}_u \\ \mathbf{K}_p^T & 0 & 0 \\ -\mathbf{M}_z + \mathbf{C}_u & 0 & \frac{1}{\Delta t} \mathbf{M}_z \end{bmatrix} \begin{Bmatrix} \hat{\mathbf{v}}^{(n+1)} \\ \hat{p}^{(n+1)} \\ \hat{\mathbf{u}}^{(n+1)} \end{Bmatrix} - \begin{bmatrix} \frac{1}{\Delta t} \mathbf{M} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\Delta t} \mathbf{M}_z \end{bmatrix} \begin{Bmatrix} \hat{\mathbf{v}}^{(n)} \\ \hat{p}^{(n)} \\ \hat{\mathbf{u}}^{(n)} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}^{n+1} \\ 0 \\ 0 \end{Bmatrix} \quad (5.25)$$

Using Newton's iterative method to solve a problem it is put in more useful form as:

$$vpu_{k+1}^{(n+1)} = vpu_k^{(n+1)} - [\mathbf{J}(vpu_k^{(n+1)})]^{-1} \cdot \{\mathbf{F}(vpu_k^{(n+1)}, vpu_{final}^{(n)})\} \quad (5.26)$$

Here vpu stands for,

$$\begin{Bmatrix} \hat{\mathbf{v}} \\ \hat{p} \\ \hat{\mathbf{u}} \end{Bmatrix}$$

While Jacobain \mathbf{J} and Residue \mathbf{F} can be given as:

$$\mathbf{J}(v_k^{(n+1)}) = \begin{bmatrix} \frac{1}{\Delta t} \mathbf{M} + \mathbf{K}_v + \mathbf{C}_v(v_k^{(n+1)}) + \mathbf{J}_v(v_k^{(n+1)}) & \mathbf{K}_p & \mathbf{K}_u \\ \mathbf{K}_p^T & 0 & 0 \\ -\mathbf{M}_z + \mathbf{C}_u(u_k^{(n+1)}) & 0 & \frac{1}{\Delta t} \mathbf{M}_z + \mathbf{J}_u(u_k^{(n+1)}) \end{bmatrix} \quad (5.27)$$

Where,

$$\mathbf{M} = \int_{\Omega} \rho \mathbf{N} \mathbf{N}^T d\Omega \quad (5.28)$$

$$\mathbf{K}_v = \int_{\Omega} \mathbf{B}^T \mathbf{D}_f \mathbf{B} d\Omega \quad (5.29)$$

$$\mathbf{C}_v = \int_{\Omega} \rho \mathbf{N} (\nabla \mathbf{v}^k) \mathbf{N}^T d\Omega \quad (5.30)$$

$$\mathbf{J}_v = \int_{\Omega} \rho \mathbf{N} \begin{bmatrix} \mathbf{v}_x^k \mathbf{N} \mathbf{n}_{,x}^T + \mathbf{v}_y^k \mathbf{N} \mathbf{n}_{,y}^T & 0 \\ 0 & \mathbf{v}_x^k \mathbf{N} \mathbf{n}_{,x}^T + \mathbf{v}_y^k \mathbf{N} \mathbf{n}_{,y}^T \end{bmatrix} d\Omega \quad (5.31)$$

$$\mathbf{J}_u = \int_{\Omega} \rho N X N^T v d\Omega \quad (5.32)$$

where,

$$X = \begin{bmatrix} \frac{\partial N_i}{\partial x} & \frac{\partial N_i}{\partial y} \\ 0 & 0 \end{bmatrix}; i = 1 - 9 \quad (5.33)$$

$$X = \begin{bmatrix} 0 & 0 \\ \frac{\partial N_i}{\partial x} & \frac{\partial N_i}{\partial y} \end{bmatrix}; i = 10 - 18 \quad (5.34)$$

$$\mathbf{K}_p = \int_{\Omega} \mathbf{B}^T \mathbf{D}_p \mathbf{H}^T d\Omega \quad (5.35)$$

$$\mathbf{K}_u = \int_{\Omega} \mathbf{B}^T \mathbf{D}_s \mathbf{B} d\Omega \quad (5.36)$$

$$\mathbf{M}_z = \int_{\Omega} \mathbf{N} \mathbf{N}^T d\Omega \quad (5.37)$$

$$\mathbf{C}_u = \int_{\Omega} \rho \mathbf{N} (\nabla \mathbf{u}^k) \mathbf{N}^T d\Omega \quad (5.38)$$

$$\nabla \mathbf{v} = \begin{bmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{N} \mathbf{n}^T}{\partial x} \cdot \hat{\mathbf{v}}_x & \frac{\partial \mathbf{N} \mathbf{n}^T}{\partial y} \cdot \hat{\mathbf{v}}_x \\ \frac{\partial \mathbf{N} \mathbf{n}^T}{\partial x} \cdot \hat{\mathbf{v}}_y & \frac{\partial \mathbf{N} \mathbf{n}^T}{\partial y} \cdot \hat{\mathbf{v}}_y \end{bmatrix} \quad (5.39)$$

$$\nabla \mathbf{u} = \begin{bmatrix} \frac{\partial u_x}{\partial x} & \frac{\partial u_x}{\partial y} \\ \frac{\partial u_y}{\partial x} & \frac{\partial u_y}{\partial y} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{N} \mathbf{n}^T}{\partial x} \cdot \hat{\mathbf{u}}_x & \frac{\partial \mathbf{N} \mathbf{n}^T}{\partial y} \cdot \hat{\mathbf{u}}_x \\ \frac{\partial \mathbf{N} \mathbf{n}^T}{\partial x} \cdot \hat{\mathbf{u}}_y & \frac{\partial \mathbf{N} \mathbf{n}^T}{\partial y} \cdot \hat{\mathbf{u}}_y \end{bmatrix} \quad (5.40)$$

$$\mathbf{D}_f = \begin{bmatrix} 2\mu & 0 & 0 \\ 0 & 2\mu & 0 \\ 0 & 0 & \mu \end{bmatrix} \quad (5.41)$$

$$\mathbf{D}_s = \frac{E}{(1 - \nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad (5.42)$$

$$\mathbf{D}_p = \begin{bmatrix} -1 \\ -1 \\ 0 \end{bmatrix} \quad (5.43)$$

and

$$\mathbf{F}(v_k^{(n+1)}, v_{final}^{(n)}) = \left\{ \begin{array}{l} \mathbf{F}_v \\ \mathbf{F}_p \\ \mathbf{F}_u \end{array} \right\} \quad (5.44)$$

where

$$\begin{aligned} \mathbf{F}_v &= \int_{\Omega} \rho \mathbf{N} \mathbf{N}^T \left(\frac{\hat{\mathbf{v}}_k^{n+1} - \hat{\mathbf{v}}^n}{\Delta t_n} \right) d\Omega + \int_{\Omega} \rho \mathbf{N} (\nabla \mathbf{v}_k^{n+1}) \mathbf{N}^T \hat{\mathbf{v}}_k^{n+1} d\Omega + \int_{\Omega} \mathbf{B}^T \mathbf{D}_f \mathbf{B} \hat{\mathbf{v}}_k^{n+1} d\Omega \\ &\quad + \int_{\Omega} \mathbf{B}^T \mathbf{D}_p \mathbf{H}^T \hat{p}_k^{n+1} d\Omega + \int_{\Omega} \mathbf{B}^T \mathbf{D}_s \mathbf{B} \hat{\mathbf{u}}_k^{n+1} d\Omega - \int_{\Gamma} \mathbf{N}^T \cdot \mathbf{t} d\Gamma - \int_{\Omega} \rho \mathbf{N}^T \mathbf{b} d\Omega \end{aligned} \quad (5.45)$$

$$\mathbf{F}_p = \int_{\Omega} \mathbf{H} \mathbf{D}_p^T \mathbf{B} \hat{\mathbf{v}}_k^{n+1} d\Omega \quad (5.46)$$

$$\mathbf{F}_u = \int_{\Omega} \mathbf{N} \mathbf{N}^T \left(\frac{\hat{\mathbf{u}}_k^{n+1} - \hat{\mathbf{u}}^n}{\Delta t_n} \right) d\Omega - \int_{\Omega} \mathbf{N} \mathbf{N}^T \hat{v}_k^{n+1} + \int_{\Omega} \mathbf{N} (\nabla \mathbf{u}_k^{n+1}) \mathbf{N}^T \hat{\mathbf{v}}_k^{n+1} d\Omega \quad (5.47)$$

For Quadratic-Velocity Linear-Pressure and Quadratic-Displacement Formulation different terms involved in above formulation can be given from Eq(4.17) and Eq(4.18) with $\hat{\mathbf{u}}$ given as:

$$\hat{\mathbf{u}} = [\hat{u}_{x_1} \quad \hat{u}_{x_2} \quad \dots \quad \hat{u}_{x_9} \quad \hat{u}_{y_1} \quad \hat{u}_{y_2} \quad \dots \quad \hat{u}_{y_9}]^T \quad (5.48)$$

As elements may share both fluid and solid. Hence the volume fractions α for fluid and β for solid are defined as following:

$$\alpha = \frac{v_f}{v} \quad (5.49)$$

$$\beta = \frac{v_s}{v} \quad (5.50)$$

where,

$$\begin{aligned} v_f &= \text{volume of fluid in a given element} \\ v_s &= \text{volume of solid in a given element} \\ v &= \text{volume the given element} \end{aligned}$$

Hence the values of these volume fractions vary as following:

for fluid only:

$$\alpha = 1, \beta = 0 \quad (5.51)$$

for solid only:

$$\alpha = 0, \beta = 1 \quad (5.52)$$

for an element consisting of both fluid and solid:

$$0 < \alpha, \beta < 1 \quad (5.53)$$

Taking into account the volume fractions, revised Jacobian and Residue can be given as:

$$\mathbf{J}(v_k^{(n+1)}) = \begin{bmatrix} \frac{1}{\Delta t} \mathbf{M} + \alpha \mathbf{K}_v + \mathbf{C}_v(v_k^{(n+1)}) + \mathbf{J}_v(v_k^{(n+1)}) & \mathbf{K}_p & \beta \mathbf{K}_u \\ \mathbf{K}_p^T & 0 & 0 \\ -\mathbf{M}_z + \beta \mathbf{C}_u(u_k^{(n+1)}) & 0 & \frac{1}{\Delta t} \mathbf{M}_z + \beta \mathbf{J}_u(v_k^{(n+1)}) \end{bmatrix} \quad (5.54)$$

and

$$\mathbf{F}(v_k^{(n+1)}, v_{final}^{(n)}) = \left\{ \begin{array}{l} \mathbf{F}_v \\ \mathbf{F}_p \\ \mathbf{F}_u \end{array} \right\} \quad (5.55)$$

where

$$\begin{aligned} \mathbf{F}_v &= \int_{\Omega} \rho \mathbf{N} \mathbf{N}^T \left(\frac{\hat{\mathbf{v}}_k^{n+1} - \hat{\mathbf{v}}^n}{\Delta t_n} \right) d\Omega + \int_{\Omega} \rho \mathbf{N} (\nabla \mathbf{v}_k^{n+1}) \mathbf{N}^T \hat{\mathbf{v}}_k^{n+1} d\Omega + \alpha \int_{\Omega} \mathbf{B}^T \mathbf{D}_f \mathbf{B} \hat{\mathbf{v}}_k^{n+1} d\Omega \\ &\quad + \int_{\Omega} \mathbf{B}^T \mathbf{D}_p \mathbf{H}^T \hat{p}_k^{n+1} d\Omega + \beta \int_{\Omega} \mathbf{B}^T \mathbf{D}_s \mathbf{B} \hat{\mathbf{u}}_k^{n+1} d\Omega - \int_{\Gamma} \mathbf{N}^T \cdot \mathbf{t} d\Gamma - \int_{\Omega} \rho \mathbf{N}^T \mathbf{b} d\Omega \end{aligned} \quad (5.56)$$

$$\mathbf{F}_p = \int_{\Omega} \mathbf{H} \mathbf{D}_p^T \mathbf{B} \hat{\mathbf{v}}_k^{n+1} d\Omega \quad (5.57)$$

$$\mathbf{F}_u = \int_{\Omega} \mathbf{N} \mathbf{N}^T \left(\frac{\hat{\mathbf{u}}_k^{n+1} - \hat{\mathbf{u}}^n}{\Delta t_n} \right) d\Omega - \int_{\Omega} \mathbf{N} \mathbf{N}^T \hat{v}_k^{n+1} + \int_{\Omega} \mathbf{N} (\nabla \mathbf{u}_k^{n+1}) \mathbf{N}^T \hat{\mathbf{v}}_k^{n+1} d\Omega \quad (5.58)$$

5.4 Discussion

The formulation is ready and the derivatives have been tested by using Finite Difference Method to cross examine the Jacobian matrix.

Chapter 6

Volume fraction update for the elements and Solid Advection law

6.1 Need of update in the volume fraction for the elements

Due to the fluid flow, forces act over the solid domain and the solid gets deformed. As the solid deforms it moves away from its current location due to which some area which was consisting of solid previously, falls under fluid domain and some part of the fluid domain gets occupied by the solid domain. It can better be understood by fig. :

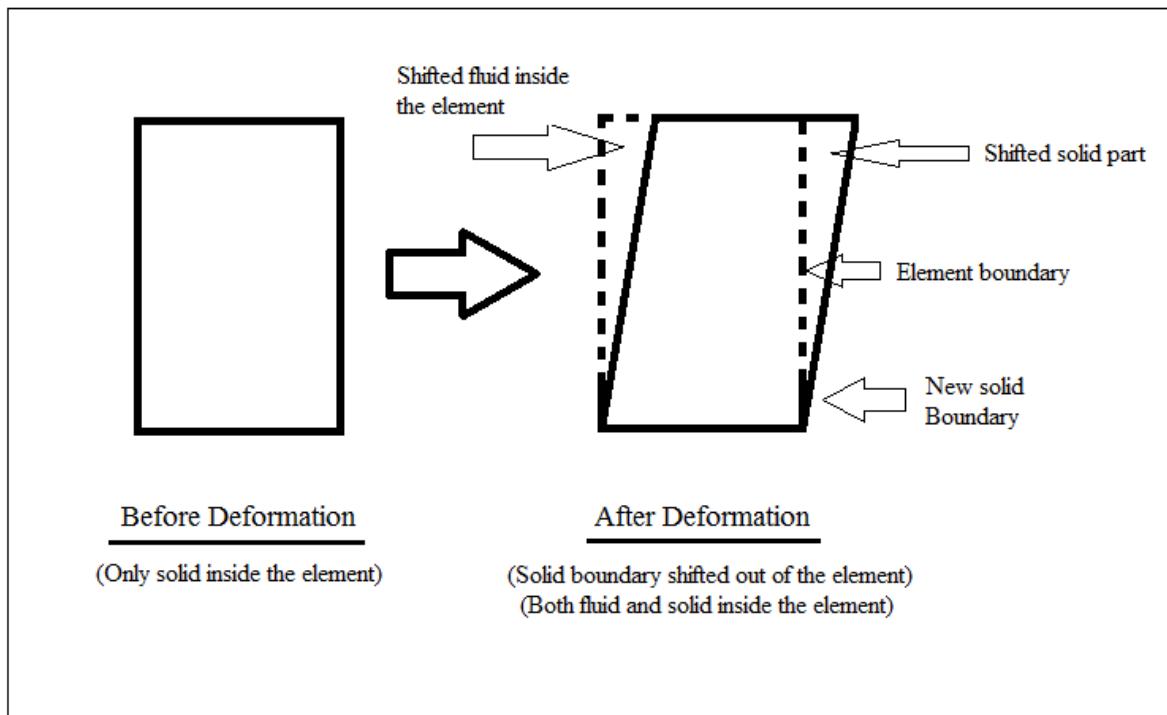


Figure 6.1: Solid advection inside the domain

As Eulerian framework is being used, the mesh will be fixed. As a result the volume fractions of the fluid and solid α and β need to be updated to capture the new location of the solid and affect the flow accordingly.

6.2 Method to update the Volume Fractions for the elements

A solid, after meshing consists of some given number of nodes. The boundary of the solid also consists of some given number of nodes and the solid boundary can easily be represented by a polygon of 'n' points where 'n' is the number of solid nodes falling at the solid boundary. The resulting polygon can be called as the solid polygon may intersect an element as shown in fig. 6.2. Due to this partial involvement of the solid inside an element there is a need to update the volume fraction of both solid and fluid to find the effective density and net stress over a given element.

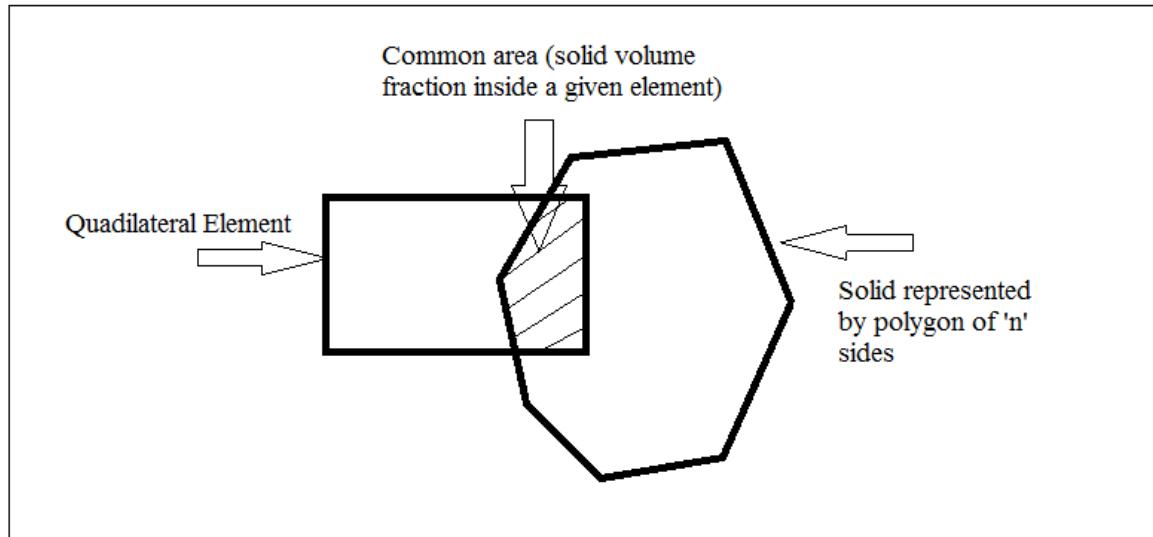


Figure 6.2: Intersection of the element and the solid polygon

To capture the advection of the solid domain, the displacements at the polygon nodes need to be updated after every time step. After an update in the displacements the new location of the solid can be found. Once the new locations of the solid are found the next task is to update the volume fractions of each element. For this purpose, the corresponding element can also be represented by a polygon and by finding the area of intersection of the two polygons, the volume fraction for the corresponding element can be updated.

6.3 Finding the area of intersection between an element and the solid polygon

To find out the volume fraction of the element that falls inside the solid polygon the area of intersection of the element to the solid polygon needs to be calculated. For 2D problems thickness can be taken as unit and hence:

$$\text{volume fraction} = \frac{\text{percentage area of intersection between element and the solid polygon}}{\text{area of element}}$$

As the two (the element and the solid polygon) have been modeled as two polygons, Monte-Carlo method can be used to find out the area of intersection.

6.3.1 Monte Carlo Method to find the volume fraction

Inside a given element 'n' number of random data points can be taken and every point can be checked if it falls inside the solid polygon. Hence the percentage of the element area that falls inside the solid

polygon can be given by the equations 6.1 and 6.2:

$$\beta = \frac{N_{sharing}}{N_{total}} \quad (6.1)$$

$$\alpha = 1 - \beta \quad (6.2)$$

where,

$N_{sharing}$ = number of random points falling inside the solid polygon

N_{total} = total number of random data points taken inside the given element

β = volume fraction of the solid inside the element

α = volume fraction of the fluid inside the element

The implementation of this method can be understood by the fig. 6.3

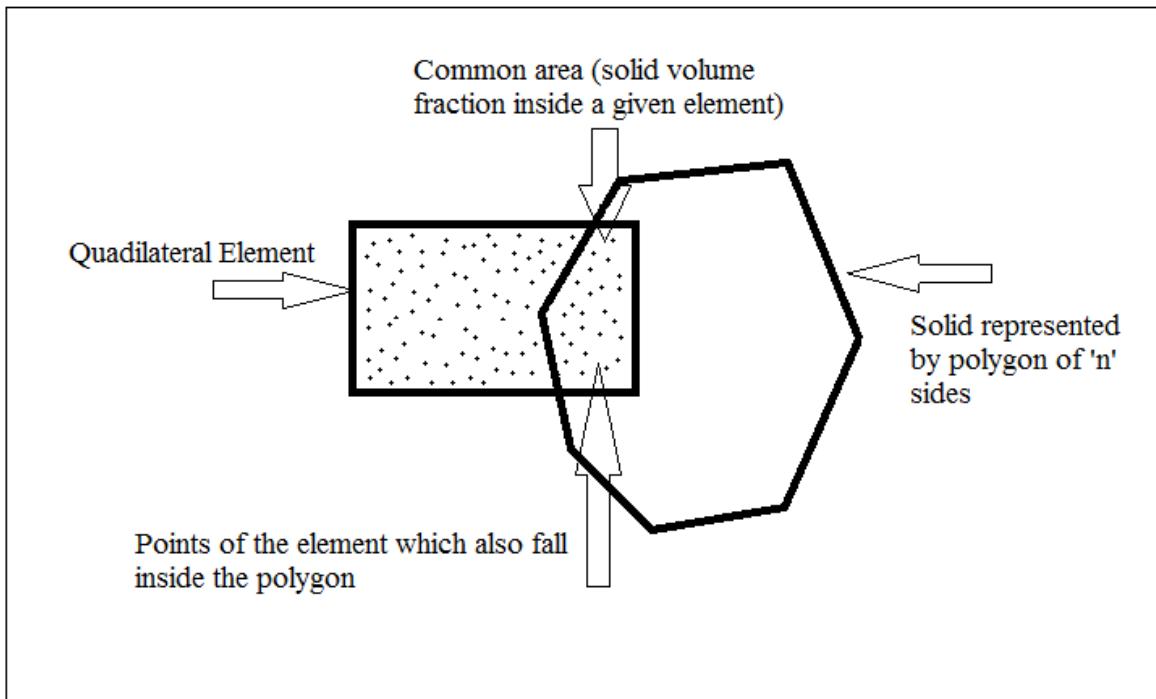


Figure 6.3: Monte Carlo approximation to find the percentage area of intersection between the element and the solid polygon

Even Odd Rule

To find if a given point of the element falls inside the solid polygon Even Odd rule is used. A ray in a certain direction can be drawn passing through the polygon and the number of sides that are crossed by the ray can be counted and if:

number of crossed sides by the ray = even, point falls outside the polygon
 number of crossed sides by the ray = odd, point falls inside the polygon

By using the above methodology the area of intersection between a given element and the solid polygon can be computed and the volume fractions can be updated for all the elements one by one.

6.4 Update in the displacements at the nodes of the solid polygon

As the solid moves, the nodes of the solid polygon may or may not match with the nodes of the mesh under consideration. If the a node at the solid polygon doesn't match with predefined mesh nodes, the displacement needs to be interpolated at that point. The interpolation can be done by using the formula shown in 6.4:

$$u = \sum_{j=1}^9 u_j^e \hat{N}_j^e \quad v = \sum_{j=1}^9 v_j^e \hat{N}_j^e \quad (6.3)$$

To interpolate the displacements at a node of the solid polygon, the Even Odd rule can be used to find out that the node falls inside which element at present. After finding the current element eq. can be used to interpolate the displacement over the element and the displacements at the solid polygon node can be updated.

6.4.1 Mapping a point from the computational domain to the master element

As the shape functions have been defined over the master element, to interpolate the displacements at a point in the computational domain, first it needs to be mapped over the master element to find the corresponding point and compute the values of the shape functions.

In fig. 6.4 a point (X_a, Y_a) has been taken in the physical domain and it is mapped over the corresponding point (x_a, y_a) inside the computational domain over the corresponding master element.

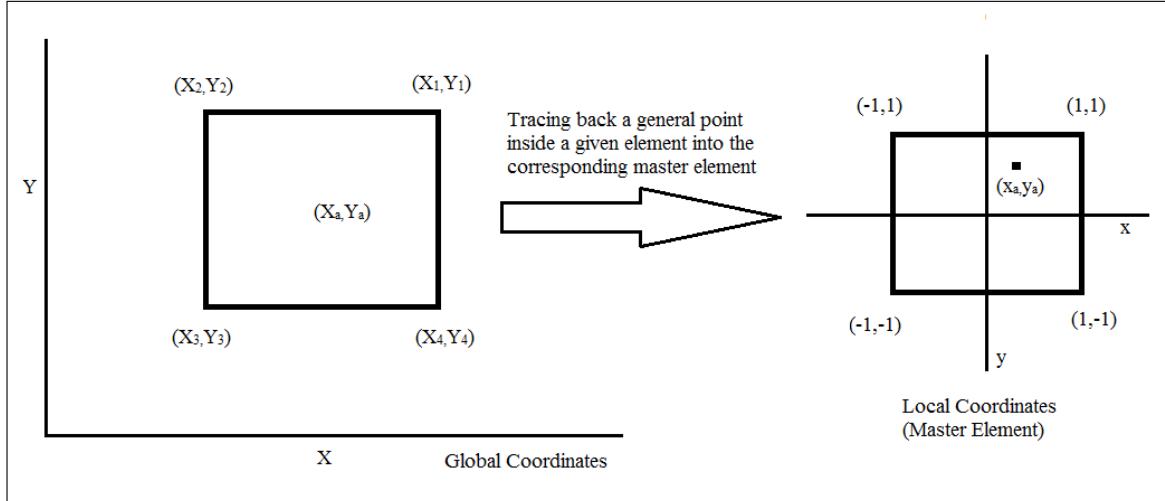


Figure 6.4: Mapping of a point in the computational domain to the master element

While the relation between these two points is given by equation 6.4

$$x = \sum_{j=1}^m x_j^e \hat{N}_j^e \quad y = \sum_{j=1}^n y_j^e \hat{N}_j^e \quad (6.4)$$

After finding the corresponding point, the displacements at the nodes of the solid polygon can be interpolated and the solid boundary can be updated for the computations of the next time step, hence accommodating a proper solid advection law.

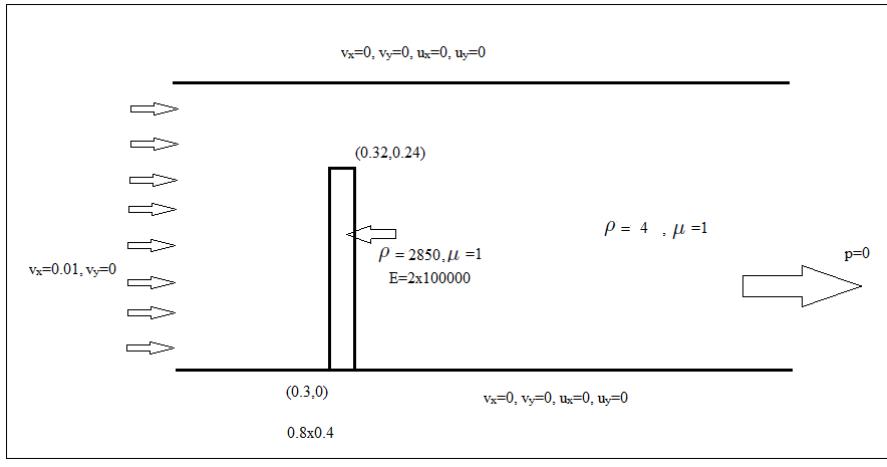
6.4.2 Flow over a Flexible Cantilever

Problem Statement and Mesh used

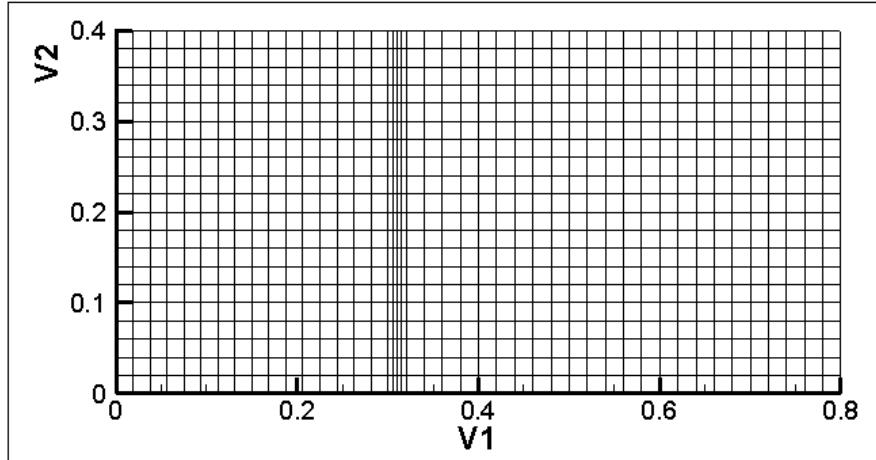
To understand the behaviour of the formulation flow over a flexible cantilever has been attempted to be solved. Fig. 6.5(a) shows the problem statement with the dimensions. The fluid and solid both are considered incompressible. The properties taken are:

$$\rho^s = 7850, \rho^f = 100, \mu^s = 1, \mu^f = 1, E = 200000, \nu^s = 0.5, \nu^f = 0.5$$

Fig.6.5(b) shows the mesh that has been used for the simulation. The mesh used is structured and orthogonal.



(a)

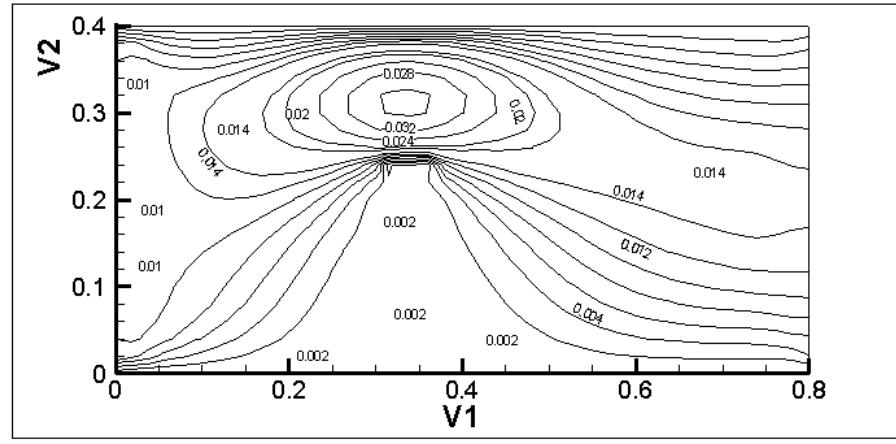


(b)

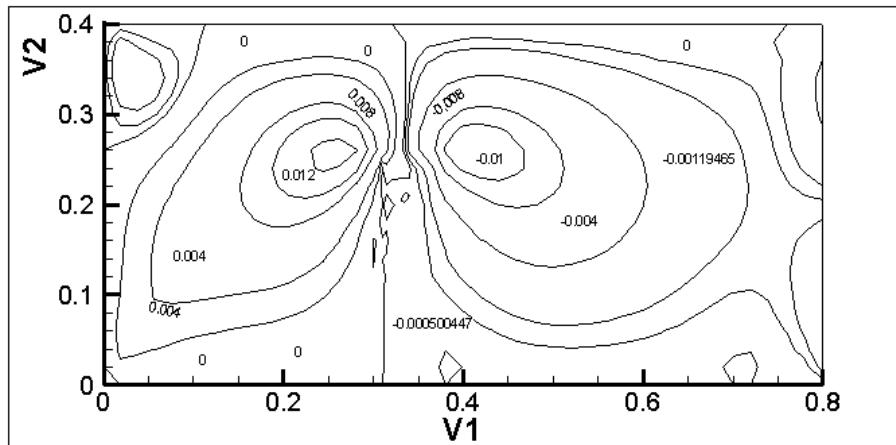
Figure 6.5: (a)Problem Statement (b)Mesh Used

6.4.3 Contours, Vector Plot and Streamlines at time t=2.0 sec

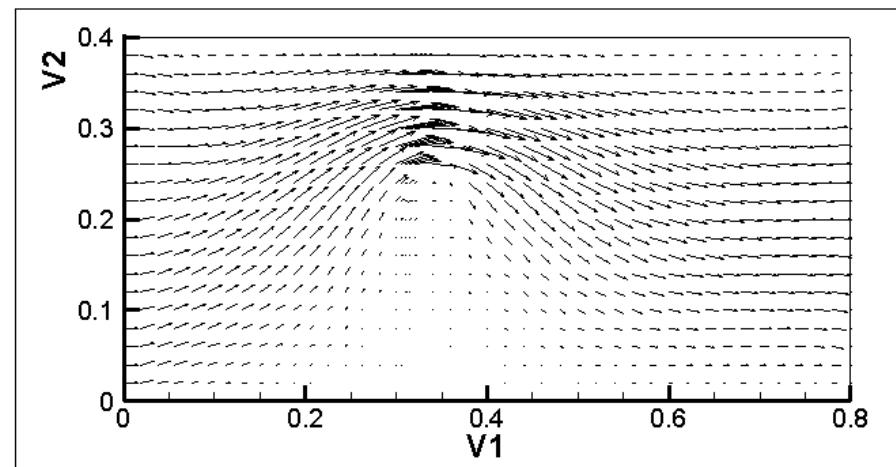
Fig. 6.6(a) and Figure 6.6(b) show the u- and v- velocity contours at time level t=2.0 sec. Fig. 6.6(c) shows the vector plot of the velocities. Also fig. 6.6(d) shows the streamlines at the given time level. The time step used was 0.1 sec and for time marching Backward Euler method has been implemented.



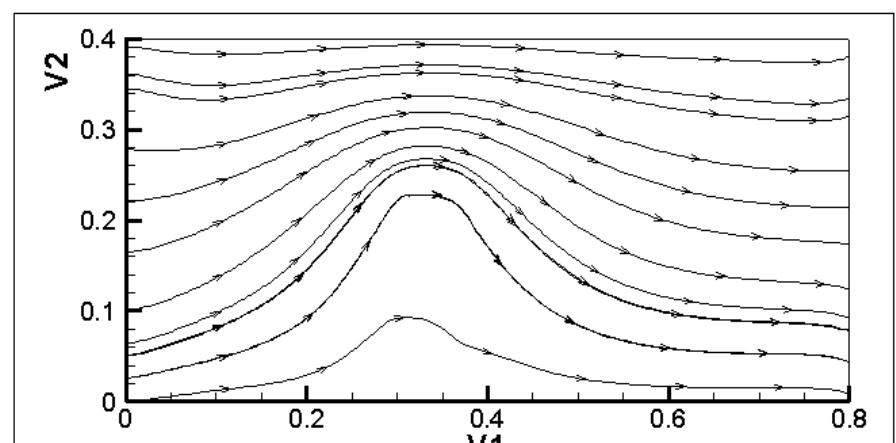
(a)



(b)

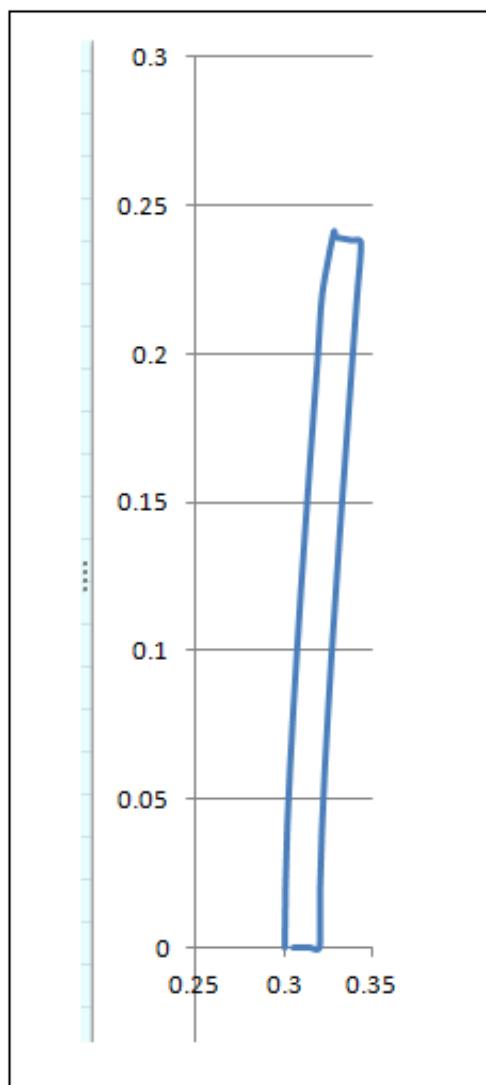


(c)



6.4.4 Deformation in the solid

Fig. 6.4.4 shows the deformation in the solid at time level t=2.0.



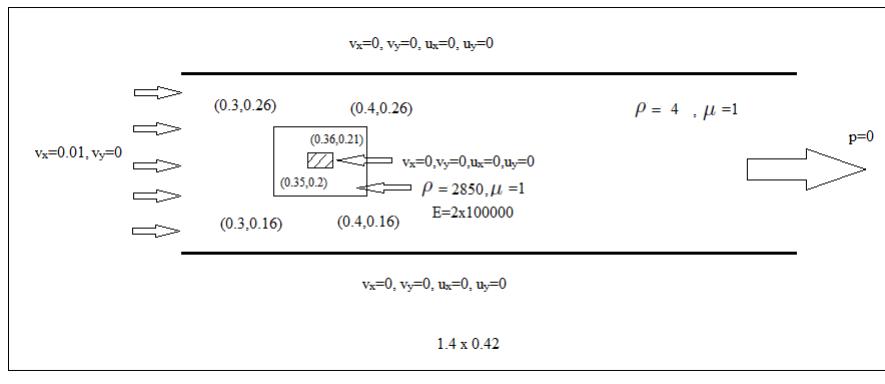
6.4.5 Flow over a Flexible Rectangular Cylinder

Problem Statement and Mesh used

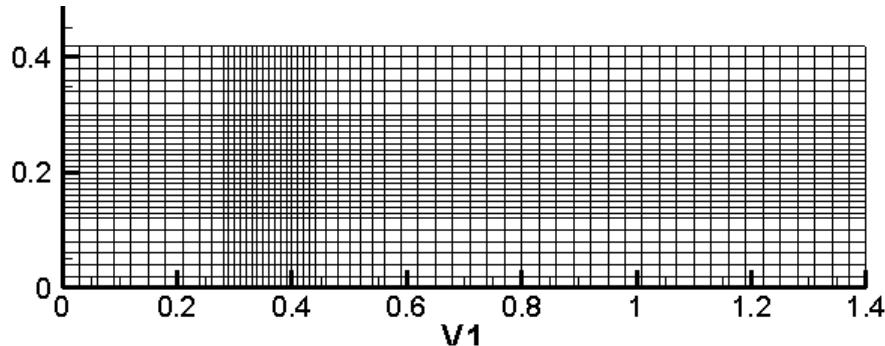
One more problem, flow over a flexible cylinder has been attempted to be solved. Fig. 6.7(a) shows the problem statement with the dimensions. The fluid and solid both are considered incompressible. The properties taken are:

$$\rho^s = 7850, \rho^f = 4, \mu^s = 1, \mu^f = 1, E = 200000, \nu^s = 0.5, \nu^f = 0.5$$

Fig.6.7(b) shows the mesh that has been used for the simulation. The mesh used is a structured and orthogonal.



(a)



(b)

Figure 6.7: (a)Problem Statement (b)Mesh Used

6.4.6 Contours, Vector Plot and Streamlines at time t=6.3 sec

Fig. 6.8(a) and Figure 6.8(b) show the u- and v- velocity contours at time level t=6.3 sec. Fig. 6.8(c) shows the vector plot of the velocities. Also fig. 6.8(d) shows the streamlines at the given time level. The time step used was 0.1 sec and for time marching Backward Euler method has been implemented.

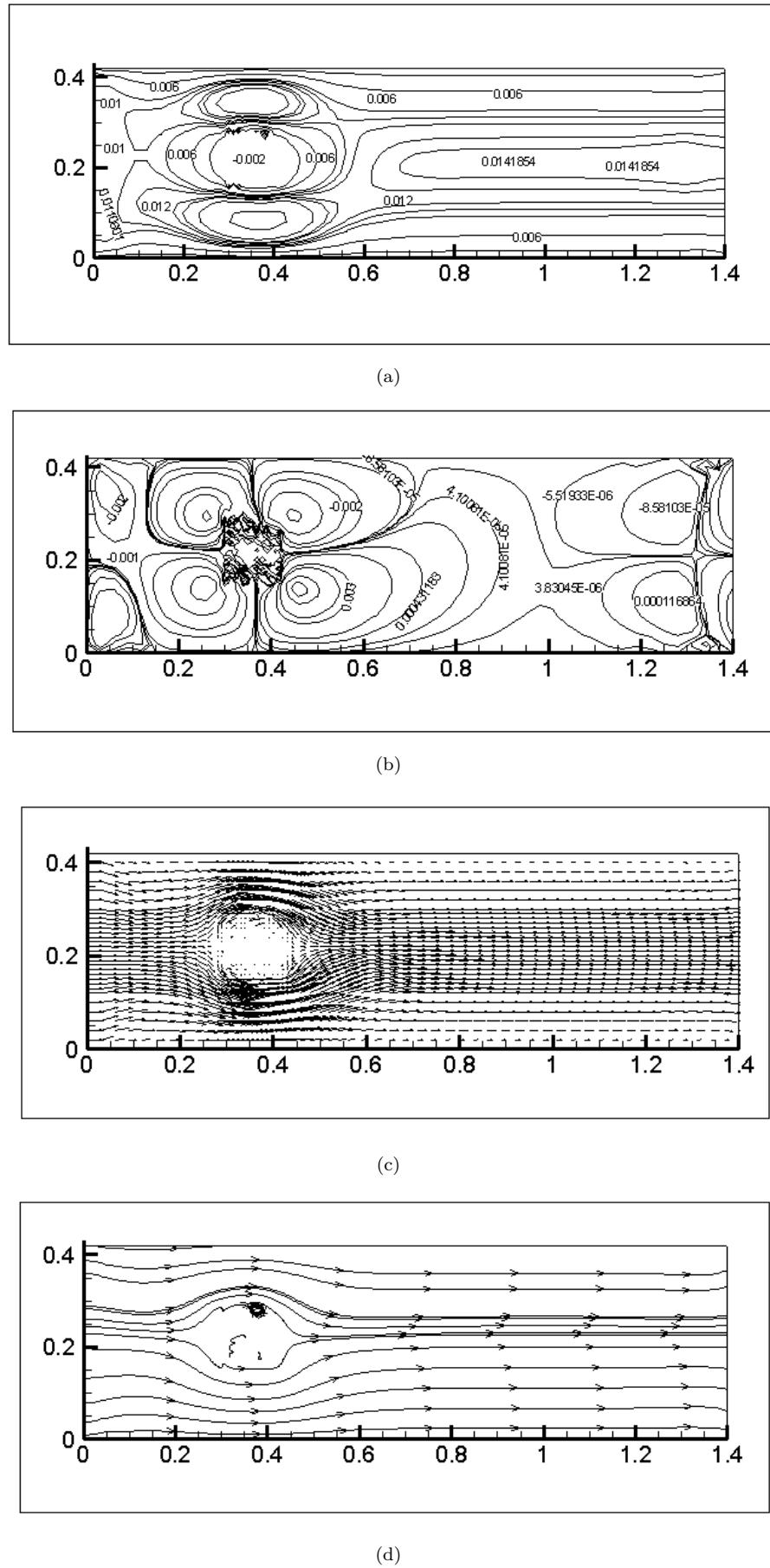
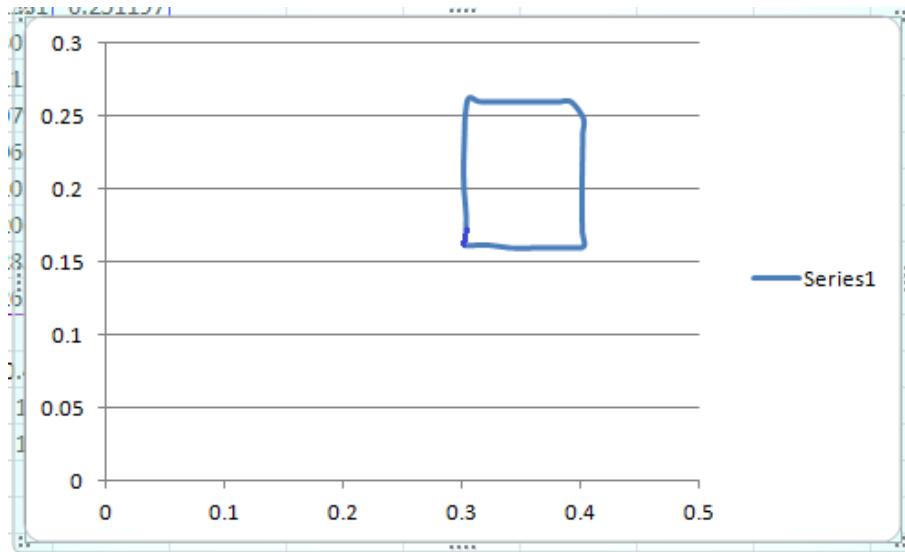


Figure 6.8: (a)u-velocity contour (b)v-velocity contour (c)Vector Plot (d)Streamlines

6.4.7 Deformation in the solid

Fig. 6.4.7 shows the deformation in the solid at time level $t=6.3$.



6.5 Discussion

Until some time the deformation of the solid and the velocities look fine but after that one point of the solid suddenly shoots to a higher deformation about 10 times the adjacent points and the simulation diverges after sometime. The probable cause of such phenomenon happening may be the interpolations of the displacement over the elements. Due to the interpolation some error gets introduced which keeps on accumulating with time and after a certain time limit it shoots to higher value which does not satisfy the physics. Resolving this problem may make the simulation more accurate.

6.6 Closure

Although the problem simulates fine for some time, but there seems to be some serious issue. The major suspicion is the error introduced due to the interpolations of the displacements. If this gets corrected in terms of increased accuracy of the interpolations, the simulation may work fine.

Chapter 7

Conclusions and Future Scope

7.1 Conclusions

- Finite element method has been implemented for solid mechanics problems and benchmarked with standard examples .
- The conventional velocity-pressure formulation for fluid problems are implemented and benchmarked with the available solution in literature.
- The proposed strategy of velocity-pressure-displacement formulation for fluid-structure interaction problem has been implemented but it works only for a given duration of time. After that the simulation becomes unstable and diverges.
- The interpolation of the displacements over the visco elastic elements introduces error at each time which shoots to a higher value after some time. The error due to interpolation needs to be reduced.

7.2 Future Scope

- The velocity-pressure-displacement formulation for fluid structure can be revisited and it can be implemented once again for some test problems as it gives fine results for some time.
- After the formulation works properly for fluid structure systems, also some Topology Optimization problems in fluid structure interaction will be solved.
- The simulation is taking time more than expected so some optimization techniques will be used in order to speed up the simulations. Also different solvers and solver libraries will be tested to achieve an enhanced simulation speed and more accurate results.

Appendix I

7.3 Interpolation function for 4 noded element

$$N_1 = \frac{1}{4}(1-\xi)(1-\eta) \quad N_2 = \frac{1}{4}(1+\xi)(1-\eta) \quad N_3 = \frac{1}{4}(1+\xi)(1+\eta) \quad N_4 = \frac{1}{4}(1-\xi)(1+\eta)$$

7.4 Interpolation function for 9 noded element

$$\begin{bmatrix} N_1 & N_2 & N_3 \\ N_4 & N_5 & N_6 \\ N_7 & N_8 & N_9 \end{bmatrix} = \begin{Bmatrix} \frac{1}{2}\xi(\xi-1) \\ -(\xi+1)(\xi-1) \\ \frac{1}{2}\xi(\xi+1) \end{Bmatrix} \begin{bmatrix} \frac{1}{2}\eta(\eta-1) & -(\eta+1)(\eta-1) & \frac{1}{2}\eta(\eta+1) \end{bmatrix} \quad (7.1)$$

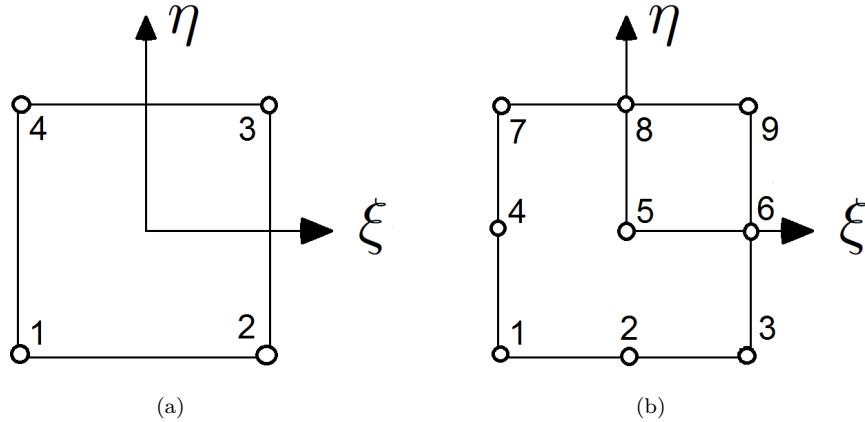


Figure 7.1: (a)4 noded and (b)9 noded rectangular elements

7.5 Mapping into the master element

The Transformation between Ω_e and $\hat{\Omega}$ is accomplished by a coordinate transformation of form

$$x = \sum_{j=1}^m x_j^e \hat{N}_j^e \quad y = \sum_{j=1}^n y_j^e \hat{N}_j^e \quad (7.2)$$

Depending on relative degree of approximations used for geometry, (x, y) and dependent variable, (u) the finite element formulations are categorised as:

1. *Superparametric*($m > n$)
 2. *Isoparametric*($m = n$)
 3. *Subparametric*($m < n$)

In our formulation we are using iso-parametric mapping.

7.5.1 Computation of Jacobian, Jacobian Matrix and derivatives of Shape Functions

$$\begin{Bmatrix} \frac{\partial N_i^e}{\partial \xi} \\ \frac{\partial N_i^e}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \frac{\partial N_i^e}{\partial x} \\ \frac{\partial N_i^e}{\partial y} \end{Bmatrix} \equiv [J] \begin{Bmatrix} \frac{\partial N_i^e}{\partial x} \\ \frac{\partial N_i^e}{\partial y} \end{Bmatrix} \quad (7.3)$$

The matrix $[J]$ is Jacobian matrix. Inversing above relation gives us,

$$\left\{ \frac{\partial N_i^e}{\partial x_i^e} \right\} = [J]^{-1} \left\{ \frac{\partial N_i^e}{\partial \xi} \right\} \quad (7.4)$$

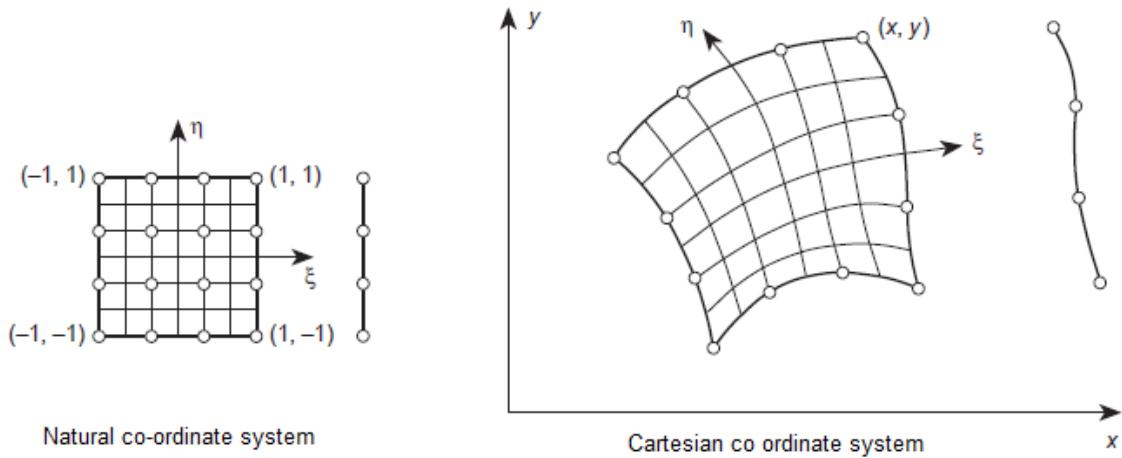


Figure 7.2: Natural and Cartesian coordinate system

Jacobian is now calculated from Eq(??). Here is the final expression in matrix form:

$$[J] = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \dots & \frac{\partial N_m}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \dots & \frac{\partial N_m}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \dots \\ x_m & y_m \end{bmatrix} \quad (7.5)$$

similarly element area $dA = dx dy$ in element Ω_e is transformed to

$$dA \equiv dx dy = \mathbf{J} d\xi d\eta \quad (7.6)$$

where \mathbf{J} is Jacobian i.e. determinant of Jacobian Matrix.

References

- [1] T. SCHNIPPER, A. ANDERSEN, and T. BOHR, “Vortex wakes of a flapping foil,” *Journal of Fluid Mechanics*, vol. 633, pp. 411– 423, 2009.
- [2] G. Hou, J. Wang, and A. Layton, “Numerical methods for fluid-structure interaction,” *Commun. Comput. Phys*, vol. 12, no. 2, pp. 337–377, 2012.
- [3] L. Zhang and M. Gay, “Immersed finite element method for fluid-structure interactions,” *Journal of Fluids and Structures*, vol. 23, no. 6, pp. 839–857, 2007.
- [4] C. G. Giannopapa, *Fluid structure interaction in flexible vessels*. PhD thesis, King’s College London, University of London, 2004.
- [5] B. Hübner, E. Walhorn, and D. Dinkler, “A monolithic approach to fluid–structure interaction using space–time finite elements,” *Computer methods in applied mechanics and engineering*, vol. 193, no. 23, pp. 2087–2104, 2004.
- [6] C. Michler, S. Hulshoff, E. Van Brummelen, and R. De Borst, “A monolithic approach to fluid–structure interaction,” *Computers & fluids*, vol. 33, no. 5, pp. 839–848, 2004.
- [7] P. Ryzhakov, R. Rossi, S. Idelsohn, and E. Oñate, “A monolithic lagrangian approach for fluid–structure interaction problems,” *Computational mechanics*, vol. 46, no. 6, pp. 883–899, 2010.
- [8] C. Farhat, M. Lesoinne, and P. Le Tallec, “Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity,” *Computer methods in applied mechanics and engineering*, vol. 157, no. 1, pp. 95–114, 1998.
- [9] K. Park, C. Felippa, and R. Ohayon, “Partitioned formulation of internal fluid structure interaction problems via localized lagrange multipliers,” *Comp. Meth. Appl. Mech. Engrg*, vol. 85, no. 190, p. 2989–3007, 2001.
- [10] S. Brown, “Displacement extrapolation for cfd+ csm aeroelastic analysis,” *AIAA paper*, vol. 1090, p. 1997, 1997.
- [11] J. R. Cebral and R. Lohner, “Conservative load projection and tracking for fluid-structure problems,” *AIAA journal*, vol. 35, no. 4, pp. 687–692, 1997.
- [12] J. R. Cebral and R. Loehner, “Fluid-structure coupling: Extensions and improvements,” *AIAA Paper*, vol. 858, 1997.
- [13] J. A. Samareh, “Use of cad geometry in mdo,” in *The 6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, vol. 8898, 1996.
- [14] J. A. Samareh, “Aeroelastic deflection of nurbs geometry,” in *6th international conference on numerical grid generation in computational field simulation. London (UK): University of Greenwich, Avery Hill Campus*, Citeseer, 1998.
- [15] J. A. Samareh, *A novel shape parameterization approach*. National Aeronautics and Space Administration, Langley Research Center, 1999.

- [16] J. A. Samareh, “A survey of shape parameterization techniques,” in *NASA Conference Publication*, pp. 333–344, Citeseer, 1999.
- [17] S. Badia, F. Nobile, and C. Vergara, “Fluid–structure partitioned procedures based on robin transmission conditions,” *Journal of Computational Physics*, vol. 227, no. 14, pp. 7027–7051, 2008.
- [18] C. Farhat, K. G. Van der Zee, and P. Geuzaine, “Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity,” *Computer methods in applied mechanics and engineering*, vol. 195, no. 17, pp. 1973–2001, 2006.
- [19] J. Vierendeels, K. Dumont, and P. Verdonck, “A partitioned strongly coupled fluid-structure interaction method to model heart valve dynamics,” *Journal of Computational and Applied Mathematics*, vol. 215, no. 2, pp. 602–609, 2008.
- [20] C. S. Peskin, “Numerical analysis of blood flow in the heart,” *Journal of computational physics*, vol. 25, no. 3, pp. 220–252, 1977.
- [21] W. K. Liu, S. Tang, *et al.*, “Mathematical foundations of the immersed finite element method,” *Computational Mechanics*, vol. 39, no. 3, pp. 211–222, 2007.
- [22] X. Wang and W. K. Liu, “Extended immersed boundary method using fem and rkpm,” *Computer Methods in Applied Mechanics and Engineering*, vol. 193, no. 12, pp. 1305–1321, 2004.
- [23] R. Glowinski, T.-W. Pan, T. I. Hesla, and D. D. Joseph, “A distributed lagrange multiplier/fictitious domain method for particulate flows,” *International Journal of Multiphase Flow*, vol. 25, no. 5, pp. 755–794, 1999.
- [24] R. Glowinski, T.-W. Pan, T. I. Hesla, D. D. Joseph, and J. Periaux, “A distributed lagrange multiplier/fictitious domain method for the simulation of flow around moving rigid bodies: application to particulate flow,” *Computer methods in applied mechanics and engineering*, vol. 184, no. 2, pp. 241–267, 2000.
- [25] R. Glowinski, T. Pan, T. Hesla, D. Joseph, and J. Periaux, “A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow,” *Journal of Computational Physics*, vol. 169, no. 2, pp. 363–426, 2001.
- [26] Z. Yu and X. Shao, “A direct-forcing fictitious domain method for particulate flows,” *Journal of computational physics*, vol. 227, no. 1, pp. 292–314, 2007.
- [27] Z. Yu, “A dlm/fd method for fluid/flexible-body interactions,” *Journal of computational physics*, vol. 207, no. 1, pp. 1–27, 2005.
- [28] C. H. Tai, Y. Zhao, and K. Liew, “Parallel computation of unsteady incompressible viscous flows around moving rigid bodies using an immersed object method with overlapping grids,” *Journal of Computational Physics*, vol. 207, no. 1, pp. 151–172, 2005.
- [29] C. Tai, K. Liew, and Y. Zhao, “Numerical simulation of 3d fluid–structure interaction flow using an immersed object method with overlapping grids,” *Computers & structures*, vol. 85, no. 11, pp. 749–762, 2007.