Advanced Computational Techniques for
Incompressible/Compressible Fluid-Structure Interactions

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF RICE UNIVERSITY
BY

Vinod Kumar

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Richard A. Tapia, Advisor

April 2005
UMI Number: 3168096

Copyright 2005 by
Kumar, Vinod

All rights reserved.

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI

UMI Microform 3168096
Copyright 2005 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346
RICE UNIVERSITY

Advanced Computational Techniques for
Incompressible/Compressible Fluid-Structure Interactions

by

Vinod Kumar

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE
DOCTOR OF PHILOSOPHY

APPROVED, THESIS COMMITTEE:

Enrique Barrera, Chair, Professor,
Mechanical Engineering and Material
Science

Richard Tapia, Co-chair, Noah Harding
Professor, Computational and Applied
Mathematics

Yildiz Bayazitoglu, H. S. Cameron
Professor, Mechanical Engineering and
Material Science

William Symes, Noah Harding Professor,
Computational and Applied Mathematics

Chris Harris, Shell Exploration and
Production Company and Visiting
Professor, Imperial College, London

Houston, Texas
April, 2005
To the late crew of the space shuttle Columbia.
ABSTRACT

Advanced Computational Techniques for
Incompressible/Compressible Fluid-Structure Interactions

by

Vinod Kumar

Fluid-Structure Interaction (FSI) problems are of great importance to many fields of engineering and pose tremendous challenges to numerical analyst. This thesis addresses some of the hurdles faced for both 2D and 3D real life time-dependent FSI problems with particular emphasis on parachute systems. The techniques developed here would help improve the design of parachutes and are of direct relevance to several other FSI problems.

The fluid system is solved using the Deforming-Spatial-Domain/Stabilized Space-Time (DSD/SST) finite element formulation for the Navier-Stokes equations of incompressible and compressible flows. The structural dynamics solver is based on a total Lagrangian finite element formulation. Newton-Raphson method is employed to linearize the otherwise nonlinear system resulting from the fluid and structure formulations. The fluid and structural systems are solved in decoupled fashion at each nonlinear iteration. While rigorous coupling methods are desirable for FSI simulations, the decoupled solution techniques provide sufficient convergence in the time-dependent problems considered here.
In this thesis, common problems in the FSI simulations of parachutes are discussed and possible remedies for a few of them are presented. Further, the effects of the porosity model on the aerodynamic forces of round parachutes are analyzed. Techniques for solving compressible FSI problems are also discussed. Subsequently, a better stabilization technique is proposed to efficiently capture and accurately predict the shocks in supersonic flows.

The numerical examples simulated here require high performance computing. Therefore, numerical tools using distributed memory supercomputers with message passing interface (MPI) libraries were developed.
Acknowledgments

"... Scientific knowledge is a body of statements of varying degrees of certainty - some most unsure, some nearly sure, none absolutely certain..." (Richard P. Feynman)

My sincere gratitude goes to my advisor, Dr. Richard Tapia, who gave me valuable advice, educational opportunity, and shared his insight and deep experience not only on the academic world, but also on the real world.

I would like to thank Dr. Tezduyar for providing me an opportunity to work on parachute problems. I learned a lot from your experience and expertise in computational fluid dynamics and finite element modeling.

Thank you Dr. Bayazitoglu, Dr. Barrera, Dr. Symes, and Dr. Harris for willingly agreeing to serve on my oral defense committee. Thank you Dr. Glowinski for your time to evaluate my work.

I am very thankful to Dr. Stein, Dr. Accorsi, Dr. Johari, Dr. Mittal, Dr. Collis, Dr. Behr, Dr. Akin, Dr. Sorensen, and Dr. Nagaragaiah for their gracious support and care both on academic and personal fronts. I am especially grateful to Dr. Stein and Dr. Accorsi who helped me throughout my studies.

I wish to thank Richard Benney from the US Army Soldier Center, who not only supported my research but also provided many valuable comments from time to time which were important to complete my research in a timely manner. I would also like to acknowledge Richard Charles and Calvin Lee from the US Army Soldier Center for their help.

Thank you Victor, Weiqing, Mehmet, Dhruv, Rajnish, Amit, Sushant, Feby, Mili,
Tomass, Medhat, Adel, Cengizan, Yiqun, Omar, Lanese, Srinivas, Pradeep, Rajat, Patrick, Priyank, Sunil, and Masayoshi for being great friends/colleagues. Without your love and support, it would have been difficult to finish the journey. I am so proud to have come across Victor and Weiqing; two great researchers. You were not only my co-workers but also a great source of inspiration throughout my PhD life. Thank you Rajnish for being a great photo-competition partner. Your sincere attitude helped me dream big. Thank you Adria, Somarine, and Sandra for your help, whenever I needed, during my stay at Rice.

I am greatly thankful to my parents in India who missed but always motivated me in the difficult moments of my PhD life. Thank you Tuntun and Ritendra for your support in the difficult moments. Thank you Barb, my Bhaiya-Bhabhi club members, and Didi-Jijaji for your homely treatments in the alien land. I would also like to thank Maria and Dora for teaching me Spanish and for filling the gap of motherly love.

At last, but in no way it is least, I would like to thank all those whom I missed to thank.

I am thankful for the generosity of Mechanical Engineering and Material Science Department at Rice University and Army High Performance Computing Research Center at University of Minnesota for providing me the computing facilities on their supercomputers; one of the fastest in the world. I am very grateful to Natick Soldier Center, NASA Johnson Space Center, and Provost office for supporting my studies.
Contents

Abstract iii
Acknowledgments v
List of Figures xii
List of Tables xvii

1 Introduction ........................................................................... 1
1.1 Overview ....................................................................... 5

2 Governing Equations ............................................................... 8
2.1 Fluid Dynamics ................................................................. 8
2.1.1 Incompressible Flow ....................................................... 9
2.1.2 Compressible Flow ....................................................... 10
2.1.3 Turbulence Modeling ..................................................... 14
2.1.4 Types of Boundary Conditions ........................................ 15
2.2 Structural Dynamics ............................................................ 17
2.3 Conclusion ...................................................................... 20

3 Finite Element Formulation ...................................................... 21
3.1 Introduction .................................................................... 21
3.2 Fluid Dynamics ................................................................. 23
3.2.1 Deforming Spatial Domain (DSD) Slab ................................ 24
3.2.2 DSD/Stabilized Space-Time (SST) Formulation for Incompressible Flows ........................................... 25
3.2.3 DSD/SST Formulation for Compressible Flows ................. 27
3.2.4 Time Integration ................................................. 28
3.3 Structural Dynamics ................................................ 30
  3.3.1 Semi Discrete formulations .................................... 30
  3.3.2 Time Integration ............................................... 31
3.4 Conclusion .......................................................... 31

4 Simulations of Cluster of G–12 Canopies and Challenges ...... 33
  4.1 Computational Challenges .......................................... 33
    4.1.1 Fluid-Structure Coupling .................................... 34
    4.1.2 Mesh Generation and Update .................................. 37
    4.1.3 Solving Linear Systems ...................................... 38
    4.1.4 Parachute Contact ............................................ 39
  4.2 Cluster of Two G-12 Parachutes ................................. 40
    4.2.1 Construction and Properties of Single G–12 Parachute ... 41
    4.2.2 Previous Simulations and Motivation ......................... 44
    4.2.3 Problem Setup ............................................... 45
    4.2.4 Pressure Ramping ............................................ 47
    4.2.5 Incompatible Meshes .......................................... 47
    4.2.6 Pinned Payload ............................................... 49
    4.2.7 Airdrop with Payload ......................................... 51
    4.2.8 Soft Landing .................................................. 54
  4.3 Cluster of Three G-12 Parachutes ............................... 56
    4.3.1 Problem definition: ......................................... 59
    4.3.2 Results ...................................................... 61
  4.4 Conclusion ........................................................ 62
5 Contact Algorithm

5.1 Introduction ................................................. 63
5.2 Why Contact Model? ....................................... 64
5.3 Contact Models ............................................. 67
5.4 Numerical Examples ....................................... 69
  5.4.1 Cluster of two 2D parachutes ...................... 69
  5.4.2 Cluster of Three G-12 Parachutes ............... 71
5.5 Conclusion .................................................. 73

6 Effect of Porosity on Fluid Dynamics Forces ............. 76

6.1 Introduction ............................................... 76
6.2 Porosity Model ........................................... 77
6.3 Boundary Conditions for FSI Computations with Porous Boundary ........................................ 77
6.4 Approach .................................................... 78
  6.4.1 Explicit Approach ................................... 79
  6.4.2 Implicit Approach ................................... 79
  6.4.3 Fully Implicit Approach ........................... 80
6.5 Test Problem: Flow Past Cross Parachute ................ 80
  6.5.1 Problem Definition .................................. 80
  6.5.2 Implicit-Explicit Convergence .................... 80
  6.5.3 Results ............................................... 82
6.6 Simulations of a T-10 Parachute with Porous Gores ........ 86
6.7 Problem Definition ....................................... 86
  6.7.1 Definition of the Porous Section ................ 89
  6.7.2 Results ............................................... 89
A.1 Adveective and Diffusivity Coefficient Matrices  . . . . . . . . . . . 142

A.1.1 Derivation of Coefficient Matrices for 3D Compressible Flow  . 142

A.1.2 Derivation of Coefficient Matrices for 2D Compressible Flow  . 145
## List of Figures

3.1 Space time slab for deforming spatial domain. ........................................ 24

4.1 Gore layout (left) and construction schematic (right) of a G-12 parachute.  
   $D_c$ is the diameter of the flat parachute, $D_v$ is the diameter of the vent. 42

4.2 Inflated profile of a G-12 parachute (courtesy of U.S. Army). ............. 43

4.3 Cluster of two G–12 parachutes: Constructed (left) and prestressed  
   (right) structural model. ........................................................................... 46

4.4 G–12 parachute cluster with a fixed payload showing the deformed  
   shapes and canopy pressure. ..................................................................... 50

4.5 G–12 parachute cluster with a fixed payload showing pressure field (left  
   and middle) and mesh colored with pressure (right) on a cross-section  
   plane cutting through the parachutes. .................................................... 51

4.6 Airdrop of payload using cluster of two G–12 parachutes: Parachute  
   canopies colored with pressure. ................................................................. 52

4.7 Airdrop of a payload using a cluster of two G–12 parachutes: Fall  
   velocity, $V_z$ (ft/s) v/s time, t (s) for different weight of payload. .......... 53

4.8 Drag force on a cluster of two G–12 parachutes during and after soft  
   landing retraction. ...................................................................................... 55

4.9 Payload velocity during descent with a cluster of two G–12 parachutes  
   during and after soft landing retraction. ................................................... 57
4.10 G–12 parachute cluster at the start of pneumatic muscle contraction (left), at the end of contraction (middle) and after soft landing retraction showing the deformed shapes and canopy pressures. 57

4.11 Cluster of six G–12 parachute: Images captured from a video of an airdrop test (courtesy of the US Army). 58

4.12 Apollo 15 Command Module Kitty Hawk Splashdown Parachute (courtesy of the NASA JSC). 58

4.13 Fluid mesh of cluster of three G-12 parachutes. Domain box + refinement box + parachute canopies are assembled to together to generate volume mesh. 60

4.14 Cluster of three G–12 parachutes at three different instances of time. Canopies are colored with pressure. 62

5.1 FSI of cluster of three G-12 parachutes without a contact model. At the end of computations, two parachutes come in contact. 66

5.2 FSI of cluster of three G-12 parachutes without a contact model. At the end of computations, two parachutes come in contact. 66

5.3 Schematic diagram of the contact model. 67

5.4 k-chart: Variation of penalty pressure with respect to the distance for various $\kappa$. 69

5.5 Fluid mesh (left), Close-up of fluid mesh (right top), and Structure mesh (right bottom). 70
5.6 In the beginning of FSI simulations before contact (top), after contact without contact model (middle), and after contact with contact model (bottom). Left: Fluid mesh colored with pressure, Right: Structure mesh.

5.7 Mesh plane at $z = 3.3R$ colored with pressure field (left) and parachute geometry (right), at the beginning (top), before they come in contact without contact model (middle), and with contact model (bottom).

6.1 Domain box for cross-parachute volume mesh (left), a section of volume mesh (center), structure mesh (right).

6.2 Definition of porous boundary.

6.3 Convergence comparison for implicit and explicit algorithm with $k_{p1} = 0.0$ and $k_{p2} = k_p$.

6.4 Parachute colored with pressure (left) and velocity magnitude (right) for various values of porosity.

6.5 Force coefficient along x-direction.

6.6 Force coefficient along z-direction.

6.7 The $+x$-position of payload v/s time.

6.8 The $+x$-position of payload v/s time. Cases ($k_{xp} = 0.05^-/0.05^+$) were computed for longer duration.

6.9 T-10 personnel parachute (courtesy of the US Army).

6.10 T-10 parachute: Gore and constructed configuration.

6.11 T-10 parachute before inflation (left) and after inflation (right).

6.12 T-10 parachute geometry with 3 porous gores (left), 5 porous gores (center) and 7 porous gores (right).
6.13 Horizontal position of payload versus time t for different number of porous gores at porosity constant $K_p=0.10$. Upper: Position $x$ v/s time t. Lower: Position $y$ v/s time t. Both position and time are nondimensionalized by $D/2$ and $D/2U$, respectively. ... 92

7.1 Artistic painting showing parachute deployment sequence of the Galileo probe. With the deployment of the drogue parachute, the probe velocity decreased from 1,000mph to 430mph (courtesy of NASA Ames). ... 93

7.2 Shock-reflection problem: problem description. ... 105

7.3 Shock-reflection problem: density $v/s x$ at $y = 0.5$ for various $\beta$ (top), for various $r$ (middle), and for various $\tau$ definitions (bottom) in comparison with analytical solution. ... 107

7.4 Oblique-shock problem: problem description ... 108

7.5 Oblique-shock problem: density $v/s x$ at $y = 0.2$ for various $\beta$ (top), for various $r$ (middle), and for various $\tau$ definitions (bottom) in comparison with analytical solution. ... 110

7.6 Coupling mechanism for compressible fluid–structure interaction problems. ... 113

7.7 Rotation of body. ... 115

7.8 Mesh for wedge problem. ... 117

7.9 Density at the beginning (left) and end (right) of computations. ... 118

7.10 The fluid mesh for Jet Intake. ... 119

7.11 Supersonic flow through air intake of a jet engine: Mach number distribution at different instances in one quarter of a cycle. Only half of the geometry shown here was used in the computational domain. ... 120
7.12 Flow past a thin flexible beam: Problem definition. .................. 122
7.13 Flow past a thin flexible beam: Fluid mesh. ......................... 122
7.14 Pressure at M=0.3 (top), and displacement-time plot(bottom). . . . 123
7.15 Density at M=3.0 (top), displacement-time plot(bottom). .......... 125
7.16 Compressible FSI simulations of cross parachute at different instances.

    Time is nondimensionalized by D/2U where D is the width of the cross
    parachute in flat configuration and U is the freestream velocity. .... 127
List of Tables

2.1 Boundary condition definition for 3D compressible flows. . . . . . . . . 17
4.1 G–12 parachute system: Geometrical, material and physical properties. 43
4.2 G–12 parachute system: membrane and cable properties. . . . . . . . 44
6.1 T–10 parachute system: material properties. . . . . . . . . . . . . . . . . 88
7.1 Shock-reflection problem: exact solution. . . . . . . . . . . . . . . . . . 105
7.2 Oblique-shock problem: analytical solution. . . . . . . . . . . . . . . . . 109
7.3 Mach cone angle with moving wedge . . . . . . . . . . . . . . . . . . . . . 118
Chapter 1

Introduction

Fluids play an important role in our day-to-day life. A fluid is a substance that deforms continuously under the application of a shear stress no matter how small the shear stress may be. Knowledge and understanding of the basic principles and concepts of fluid dynamics are essential to analyze systems in which a fluid is the working medium. Some examples are flow past a parachute, flow past a spacecraft/airplane, blood flow through the arteries, flow through a channel, and water-waves. Most of these fluid dynamics problems are coupled with the dynamics of other physics e.g. structural dynamics, particle/object dynamics, magnetic field [1–4]. Fluid dynamics problems are governed by the well known Navier-Stokes equations which were derived by George S. Stokes (1819-1903) and C.H. Navier (1785-1836) independently. Navier-Stokes equations are a set of nonlinear time dependent advective-diffusive partial differential equations. In special cases, for example Stokes flow, one can simplify the equations enough to study them analytically. However, for most real life problems, it is essentially impossible to perform an analytical analysis. This leads us to the two possible alternatives – experimental and numerical analysis. Both of these approaches have their strengths and drawbacks. Experimental analysis is costly, time consuming, laborious and lacks the freedom provided by a numerical approach. A numerical analysis approach is constrained by computational resources and modeling capabilities. With the advent of faster, cheaper and more easily accessible computers in the last decade of the 20th century, numerical simulation has become a popular tool in industry and academia. The driving force for this comes from the need to develop
new products with less cost and time. Numerous research and development efforts across the globe have contributed to understanding and improving these simulation tools, but still much remains to be explored.

For many real life problems such as a cargo drop in martian atmosphere, numerical simulation is the only approach to understand the physics. It can show us what we cannot see when dealing with real products and experiments. It is flexible in changing geometries and conditions. Often, numerical simulation is a lot cheaper than the experiment. For example, the cost of one airdrop test of parachutes is about twenty thousand dollars whereas numerical analysis can be done for a fraction of this cost [5].

The reliability of the numerical analysis approach is a matter of great concern to engineers and scientists. It is limited by the computational resources, the accuracy of numerical models, proper boundary conditions etc. With the invention of affordable high performance parallel computers and advances in computational techniques in recent years, one is able to deal with and study real life problems with significantly better accuracy.

In this thesis, I concentrate on developing mathematical models to simulate the aerodynamics of parachutes. The parachute was first imagined and sketched by Leonardo Da Vinci in the late 15th century [6]. Over the centuries, it became more and more popular among hobbyists and the Army. The Army needs parachutes to airdrop cargos and to safely send paratroopers to war zones and other areas. Parachute systems are often deployed from a variety of aircraft under different altitudes, velocities, and environmental conditions. Sometimes the safety zone for landing is very narrow, so parachutes should be designed in such a way that they are robust enough to withstand the severe conditions but efficient enough to land close to the target. I
am going to mainly focus my study on T-10 and G-12 parachute systems. The T-10 parachute is a personnel parachute and the G-12 is a cargo parachute. The G-12 is often used in cluster configuration (e.g. a cluster of two G-12 parachutes, a cluster of three G-11 parachutes, a cluster of twelve G-12 parachutes) to deal with heavier cargo. Personnel parachutes (a parachute used by paratroopers), involve human life. Cargo parachutes involve important cargo (missiles, food, cash money etc. [5]) which must be delivered to the right place in the hands of the right people. Both of these scenarios are important. Failure is not acceptable. Here, I use computational techniques to better understand the dynamics of parachutes and to make suggestions on how to improve their design.

Although, much research has been done in improving the design of parachutes, many questions (e.g. the aerodynamics of a cluster of parachutes, the opening of parachute, the contact, the soft-landing, the impact of porosity) remain unanswered. A parachute is made of fabric textile materials (usually nylon). Choosing the right material for the parachute which is light as well as strong has been an issue of concern. In 1931, Appel et al. [7] analyzed the use of cotton as a substitute for silk. Many other researchers in the last century have carried out extensive experimental (wind-tunnel tests, drop tests etc.) and empirical studies [7, 1, 8, 9]. However, the larger challenge is understanding the aerodynamics of flexible canopies. The interaction between the parachute system and the surrounding flow field is dominant in all parachute operations, and thus the ability to predict the Fluid-Structure Interaction (FSI) behavior of a parachute is a recognized and important challenge within the parachute research community [10, 11], and particularly of great interest to the Army.

The dynamics of a parachute is complex and difficult to model accurately because
of fluid and structure interactions. During both the inflation process and the terminal descent stage, the dynamics are governed by a highly nonlinear coupling between the structural dynamics [12] of the parachute system and the surrounding fluid flow [13]. Semi-empirical parachute models require experimental data in order to adjust the model to represent parachute phenomena. However, these models break down if the simulated problem deviates from the problem that the empirical data supports. In order to accurately capture time-variant parachute dynamics, the structural dynamics and the fluid dynamics must be addressed as a coupled system. The modeling capabilities of a block iteratively coupled FSI solver have been successfully demonstrated for FSI problems [11, 14].

In this thesis, I will address some of the modeling issues to enhance the capabilities of an existing FSI solver. First, I will address a dynamic contact issue associated with parachute systems and then will present a mechanism to overcome this obstacle in performing FSI simulations. The dynamic contact is defined as the instantaneous contact of two surfaces. The contact of two surfaces is of usually two types – parachute-to-parachute contact and gore-to-gore contact. The parachute-to-parachute contact occurs when two or more parachutes in a cluster configuration come in close proximity to each other. The gore-to-gore contact happens when one or more gores collapse due to loss of sufficient pressure drop across the surface of the canopy. Gore-to-gore contact is often seen in soft-landing when risers are pulled in to reduce the impact velocity. It is also present in the inflation process of the parachutes. These type of contacts are dynamic in nature as they appear and disappear during the course of time. Contact is a bottleneck in the FSI simulations. Various researchers have tried to resolve the contact issues [15–18]. Xu et al. [16, 18] studied the physics of contact
for the stand alone structural dynamics case. This model, however, cannot be used to carry out the FSI simulations because it allows two surfaces to either penetrate or touch each other. I will present a contact model for a 3D parallel fluid-structure interaction solver based on a pressure-penalty formulation that deals with such dynamic contact issues.

Parachutes are made of fabric materials which are naturally porous. The natural porosity can be changed by heating the canopy or changing the orientation of fabric's structure. By changing the porosity and the area of porous section, one can create a lateral force on the parachute systems. These features can be used for providing better guidance, navigation, and control (GN&C) of round parachutes.

Often, parachutes are deployed at high speeds where the surrounding fluid is in the compressible flow regime. At this point one cannot use the incompressible fluid-structure solver to carry out numerical simulations. Thus, a compressible version of the fluid-structure interaction solver is needed to model such problems. Additionally, compressible flow in the supersonic regime suffers from the discontinuity in the solution as a result of shocks. Aliabadi [19] used a discontinuity capturing methodology based on entropy variables for the stabilized finite element formulations. It appears that entropy based discontinuity parameters generate an over-diffusive system. I will present an efficient method to compute the stabilization parameters for such problems with discontinuity.

1.1 Overview

In Chapter 2, the governing equations of the Navier-Stokes equations, written for compressible and incompressible Newtonian fluid flows, along with state of the
constitutive equations for ideal gas, are presented. The differential equations of structural dynamics for membrane type surfaces, which is accompanied by uniaxial stress equation for cables, are also presented.

In Chapter 3, following a brief introduction to the stabilized finite element method, the Deforming-Spatial Domain/Stabilized Space-Time (DSD/SST) finite element formulations are presented. The DSD/SST finite element formulations are derived from the Petrov-Galerkin version of Navier-Stokes equations [13, 20, 21]. It has a built-in capability to handle moving boundaries.

In Chapter 4, some of the challenges involved in accurate modeling of fluid dynamics problems are mentioned. The FSI results for clusters of parachutes – a cluster of two G–12 parachutes and a cluster of three G–12 parachutes, are presented. Soft-landing of a cluster of two parachutes is also discussed.

In Chapter 5, some contact models are be reviewed and a contact model based on pressure penalty formulations is discussed which deals with dynamic contact issues. This contact model is successfully applied to deal with the contact issues faced in Chapter 4 for FSI simulations of cluster of three G–12 parachutes.

In Chapter 6, a porosity model is implemented in the FSI solver and its effect, on the aerodynamic forces for cross- and round-parachute problems, is studied. Various approaches to deal with convergence issues, arising from the increase in the nonlinearity effect due to the introduction of porosity, are presented.

In Chapter 7, the capability of the existing fluid structure interaction solver is enhanced to study the aerodynamics of parachute deployments at high speed when the surrounding flow is compressible. Some issues and challenges in accurate modeling of compressible flows are discussed. In Section 7.2, the stabilization and discontinuity
parameters for compressible flows are discussed and a better shock-capturing parameter is proposed to efficiently capture and accurately predict the shocks. In Section 7.3, a methodology to solve compressible fluid-structure problems is discussed.

In Chapter 8, the present work is summarized and possible directions for future work are pointed out.
Chapter 2
Governing Equations

Fluid structure interaction of parachutes involves the physics of both fluid and structural dynamics. Fluid dynamics (FD) is governed by the Navier-Stokes equations. Structural dynamics (SD) is governed by the equilibrium equations of "a tension structure" composed of membranes, cables, and point mass. In this chapter, the governing equations of both fluid flows and structural dynamics are described. These include the Navier-Stokes equations of incompressible and compressible flows with constitutive equations of a Newtonian fluid, and conservation of linear momentum equations for structure with constitutive equations of membranes and cables.

2.1 Fluid Dynamics

The Navier-Stokes equations are a set of nonlinear time dependent advective-diffusive equations. Flows in which variations in density are negligible are termed incompressible; when density variations within a flow are not negligible, the flow is called compressible. For practical applications, when the Mach number is less than 0.3 (i.e. the maximum density variation is less than 5 percent), the flows are treated as incompressible. Beyond this Mach number, the flow is treated as compressible.

For most of airdrop applications, the Mach number is far less than 0.3. However, in some cases (e.g. the drogue parachute deployment to slow down a space vehicle before its landing), the Mach number is greater than 0.3. Both incompressible and compressible fluid flow problems are studied.
2.1.1 Incompressible Flow

Let $\Omega_t \subset \mathbb{R}^{n_{sd}}$ be the spatial domain with boundary $\Gamma_t$ at any instant of time $t \in (0, T)$. Here, $n_{sd}$ is the number of spatial dimensions and $T$ is the total time of computations. The spatial coordinates and time are denoted by $x$ and $t$, respectively. The Navier-Stokes equations of incompressible flows are

$$\rho \left( \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{f} \right) - \nabla \cdot \mathbf{\sigma} = 0 \quad \text{on } \Omega_t \quad \forall t \in (0, T), \quad (2.1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{on } \Omega_t \quad \forall t \in (0, T), \quad (2.2)$$

where $\rho$, $\mathbf{u}$ and $\mathbf{f}$ are the density, velocity and the external force, respectively. Equation (2.1) is linear momentum balance equation. Equation (2.2) is the continuity equation. The continuity equation which ensures the mass conservation satisfies the divergence free condition. The $\Omega_t$ is the computational domain at time $t$. The constitutive equation for stress tensor, $\mathbf{\sigma}$ is

$$\mathbf{\sigma}(p, \mathbf{u}) = -p \mathbf{I} + 2\mu \mathbf{\varepsilon}(\mathbf{u}), \quad (2.3)$$

where $p$, $\mathbf{I}$, and $\mu$ are the pressure, the identity tensor and the constant viscosity, respectively. The strain rate tensor, $\mathbf{\varepsilon}(\mathbf{u})$ is defined as

$$\mathbf{\varepsilon}(\mathbf{u}) = \frac{1}{2}((\nabla \mathbf{u}) + (\nabla \mathbf{u})^T). \quad (2.4)$$

Both Dirichlet- and Neumann- type boundary conditions are accounted for, and are represented as

$$\mathbf{u} = g \text{ on } (\Gamma_t)_g,$$

$$\mathbf{n} \cdot \mathbf{\sigma} = h \text{ on } (\Gamma_t)_h, \quad (2.5)$$
where \((\Gamma_t)_g\) and \((\Gamma_t)_h\) are complementary subsets of the boundary \(\Gamma_t\), \(\mathbf{n}\) is the unit normal vector at the boundary, and \(g\) and \(h\) are given functions. The initial condition is specified as

\[
\mathbf{u}(\mathbf{x}, t) = \mathbf{u}_0(\mathbf{x}), \nabla \cdot \mathbf{u}_0(\mathbf{x}) = 0 \quad \text{on} \ \Omega_0,
\]

(2.6)

where \(\mathbf{u}_0(\mathbf{x})\) is the initial value velocity. The \(\Omega_0\) is the computation domain at \(t = 0\). The initial velocity could be a function of spatial coordinate, \(\mathbf{x}\), and must satisfy the divergence free condition. I use uniform initial conditions for all the problems presented in this thesis.

### 2.1.2 Compressible Flow

Let \(\Omega_t \subset \mathbb{R}^{n_{sd}}\) be the spatial domain with boundary \(\Gamma_t\) at any instant of time \(t \in (0, T)\). Here \(n_{sd}\) is the number of spatial dimensions. The spatial coordinates and time are denoted by \(\mathbf{x}\) and \(t\), respectively. The Navier-Stokes equations of compressible flow are

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad \text{on} \ \Omega_t \ \forall t \in (0, T),
\]

(2.7)

\[
\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{uu}) + \nabla p - \nabla \cdot \mathbf{T} = 0 \quad \text{on} \ \Omega_t \ \forall t \in (0, T),
\]

(2.8)

\[
\frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho e \mathbf{u}) + \nabla \cdot (p \mathbf{u}) - \nabla \cdot \mathbf{T} \mathbf{u} + \nabla \cdot \mathbf{q} = 0 \quad \text{on} \ \Omega_t \ \forall t \in (0, T),
\]

(2.9)

where \(\rho(\mathbf{x}, t), \mathbf{u}(\mathbf{x}, t), p(\mathbf{x}, t), \mathbf{T}(\mathbf{x}, t), e(\mathbf{x}, t)\) and \(\mathbf{q}(\mathbf{x}, t)\) are the density, velocity, pressure, viscous stress tensor, total energy per unit volume, and the heat flux vector, respectively. Equations (2.7), (2.8), and (2.9) are continuity (mass conservation), linear momentum, and energy equations, respectively.

For Newtonian fluid dynamics, the constitutive equation for compressible flow are given by
\[ T = \mu(\nabla u + (\nabla u)^T) + (\lambda + \kappa)(\nabla \cdot u)I, \]  
(2.10)

where \( T \) is the stress tensor, \( \mu \) and \( \lambda \) are the viscosity coefficients. The parameter \( \kappa \) is bulk or dilatational viscosity which is zero for ideal monatomic gases. For most of the monatomic molecules, it has been shown that thermodynamic pressure is equal to mechanical pressure (Stokes’ hypothesis). As a result, these two coefficients are related by

\[ \lambda = \frac{2}{3} \mu. \]  
(2.11)

Pressure is related to density and internal energy via the equation of state. For ideal gases, the equation of state assumes the special form

\[ p = (\gamma - 1)\rho i, \]  
(2.12)

where \( \gamma \) is the ratio of specific heats and \( i \) is the internal energy per unit volume. The internal energy is related to the total energy and kinetic energy as

\[ i = e - \frac{1}{2}||u||^2. \]  
(2.13)

The heat flux vector is defined as

\[ \mathbf{q} = -\kappa \nabla \theta, \]  
(2.14)

where \( \kappa \) is the heat conductivity and \( \theta \) is the temperature. The temperature is related to internal energy through the following expression

\[ \theta = \frac{i}{C_v}, \]  
(2.15)

where \( C_v \) is the specific heat of the fluid at constant volume. For an ideal gas
\[ C_v = \frac{R}{\gamma - 1}, \quad (2.16) \]

where \( R \) is the ideal gas constant. The Prandtl number \( (P_r) \), which relates the heat conductivity of the fluid to its viscosity, is defined as

\[ P_r = \frac{\mu C_p}{\kappa}, \quad (2.17) \]

where \( C_p \) is the specific heat of the fluid at constant pressure. For an ideal gas, it simplifies to

\[ C_p = \frac{\gamma R}{\gamma - 1}. \quad (2.18) \]

Equations (2.7), (2.8), and (2.9) can be written in the terms of conservative variables as

\[
\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_i}{\partial x_i} - \frac{\partial \mathbf{E}_i}{\partial x_i} = 0 \quad \text{on } \Omega, \quad \forall t \in (0, T), \quad (2.19)
\]

where \( \mathbf{U} \) is the vector of conservative variables, \( \mathbf{F}_i \) is the Euler flux vector and \( \mathbf{E}_i \) is the viscous flux vector. They are given as

\[
\mathbf{U} = \begin{pmatrix}
\rho \\
\rho u_1 \\
\rho u_2 \\
\rho u_3 \\
\rho e
\end{pmatrix}, \quad (2.20)
\]
\[
\mathbf{F}_i = \begin{pmatrix}
    u_i \rho \\
    u_i \rho u_1 + \delta_{i1}p \\
    u_i \rho u_2 + \delta_{i2}p \\
    u_i \rho u_3 + \delta_{i3}p \\
    u_i (\rho e + p)
\end{pmatrix}, \quad (2.21)
\]

\[
\mathbf{E}_i = \begin{pmatrix}
    0 \\
    \tau_{i1} \\
    \tau_{i2} \\
    \tau_{i3} \\
    -q_i + \tau_{ik}u_k
\end{pmatrix}, \quad (2.22)
\]

where \( u_i, q_i, \) and \( \tau_{ik} \) are components of the velocity, heat flux, and viscosity stress tensor, respectively. The parameter \( \delta \) is the Kronecker delta function. Repeated indices imply summation. Equation (2.19) can be written in advection-diffusive form as

\[
\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}_i \frac{\partial \mathbf{U}}{\partial x_i} - \frac{\partial}{\partial x_i} \left( \mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial x_j} \right) = 0 \quad \text{on} \ \Omega_t \ \forall t \in (0, T), \quad (2.23)
\]

where

\[
\mathbf{A}_i = \frac{\partial \mathbf{F}_i}{\partial \mathbf{U}} \quad (2.24)
\]

is the advective Euler-Jacobian matrix, and \( \mathbf{K}_{ij} \) is the diffusivity matrix satisfying

\[
\mathbf{K}_{ij} \frac{\partial \mathbf{U}}{\partial x_j} = \mathbf{E}_i. \quad (2.25)
\]
Corresponding to the equation (2.23), Dirichlet (essential or g-type) and Neumann (natural or h-type) boundary conditions, respectively, are

\[ U = g \quad \text{on } \Gamma_g \ \forall t \in (0, T), \quad (2.26) \]

\[ n \cdot E = h \quad \text{on } \Gamma_h \ \forall t \in (0, T), \quad (2.27) \]

and the initial condition is given by

\[ U(x, 0) = U_0 \quad \text{on } \Omega_0. \quad (2.28) \]

The advective \((A_i)\) and diffusivity \((K_{ij})\) coefficient matrices are derived in Appendix A.

### 2.1.3 Turbulence Modeling

The Smagorinsky turbulence model [22] has been implemented in the fluid dynamics solver. In this model, the effective turbulent viscosity is given by

\[ \mu_t = \mu + (\kappa h_e)^2 \sqrt{2 \varepsilon(u) : \epsilon(u)}, \quad (2.29) \]

where \(\mu\) is the physical viscosity, \(h_e\) is a measure of the element length and the constant \(k\) is chosen to be equal to 0.15.

Studies by Akin et al. [23] indicate that the standard Smagorinsky model makes little difference in the results when used with the Streamline-Upwind/Petrov-Galerkin (SUPG) stabilization formulations. Their studies showed that, in most of the flow domain computations, the SUPG viscosity, in terms of its maximum magnitude which is attained in the flow direction, is much larger than the Smagorinsky viscosity. It
appears that the solution obtained with the SUPG type methods is similar to the results obtained from the large-scales of subgrid scale model. The time averaged large-scale loads and deformations on/of the chute is quite reasonable with the SUPG formulations. However, the flow characteristics in the wake where turbulence features are dominant will require a better turbulence modeling scheme in association with SUPG to efficiently capture and accurately predict the turbulence characteristics. This is a first step to model these type of problems, especially in wake regions. In my opinion, a subgrid-scale model is needed to improve the modeling capabilities. Further studies are needed to better model and understand the turbulence phenomena in the context of stabilized finite element formulations. Interested readers are referred to [24, 25].

2.1.4 Types of Boundary Conditions

To accurately represent the fluid dynamics, a proper boundary condition must be imposed on the outer boundaries of the fluid domain and on the inner surfaces representing fluid–structure interfaces. There are two types of boundary conditions - Dirichlet- and Neumann-type. Mixed type of boundary conditions are usually used.

Incompressible Flows

- **Inflow**: All components of velocity are prescribed (Dirichlet-type BCs). Freestream velocity, $U_\infty$, is specified at the inflow i.e. $U_{inflow} = U_\infty$.

- **Outflow**: This is a traction-free boundary condition. Here, the shear stresses in all the directions are set free (Neumann-type) i.e. $(\sigma \cdot n = 0)$. In all problems presented in this thesis, only the homogeneous form of Neumann-type BCs ($h = 0$) is considered at the outflow boundaries.
• **No-slip:** All components of velocity are set to zero for a fixed boundary or equal to velocity of the boundary for a moving boundary.

• **Slip:** This is an example of mixed form of Dirichlet- and Neumann-type BCs. Here, the Dirichlet- and Neumann-type BCs are used on the same boundary. In this case, the flow is allowed to slip on the boundary by letting the traction in the tangential directions be free. The normal component of velocity is prescribed to be zero (or equal to the velocity of the boundary in the case of a moving boundary problem). These type of BCs are often imposed on the side walls of the fluid domain. This condition is also appropriate for cases where the fluid domain is bounded by a frictionless wall or symmetry boundary.

• **X-Slip** (or **Y-Slip** or **Z-Slip**): This is a special case of slip BCs when the surface of the boundary is planar and its normal is aligned to $x$ (or $y$ or $z$)-direction where $x - y - z$ represent the cartesian system of coordinates. Here, the component of velocity along $x$ (or $y$ or $z$)-direction is set to zero. In the other two tangential directions, the tractions are set free.

**Compressible Flows**

Usually a mixed form of both Dirichlet- and Neumann-type BCs is used. This is achieved by defining two variables, $BC_{set}$ and $BC_{val}$. These variables are called boundary variables. The $rndset$ determines whether a boundary is defined as Dirichlet type ($BC_{set}=1$) or Neumann-type ($BC_{set}=0$) for a specified degree of freedom on a given boundary. The $BC_{val}$ specifies the corresponding values. Table 2.1 describe how to use these boundary variables to specify a boundary condition in 3D compressible flows. This generalized concept can be used for 2D flow problems as well. The concept
of boundary variables $BC_{\text{set}}$ and $BC_{\text{val}}$ can also be extended to incompressible flow. In this thesis, only the homogeneous form of Neumann-type BCs are considered.

<table>
<thead>
<tr>
<th>Deg. of freedom</th>
<th>Dirichlet-type, $BC_{\text{set}} = 1$</th>
<th>Neumann-type, $BC_{\text{set}} = 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$BC_{\text{val}} = \rho$</td>
<td>$BC_{\text{val}} = h_1$</td>
</tr>
<tr>
<td>2</td>
<td>$BC_{\text{val}} = \rho u_1$</td>
<td>$BC_{\text{val}} = h_2$</td>
</tr>
<tr>
<td>3</td>
<td>$BC_{\text{val}} = \rho u_2$</td>
<td>$BC_{\text{val}} = h_3$</td>
</tr>
<tr>
<td>4</td>
<td>$BC_{\text{val}} = \rho u_3$</td>
<td>$BC_{\text{val}} = h_4$</td>
</tr>
<tr>
<td>5</td>
<td>$BC_{\text{val}} = \rho e$</td>
<td>$BC_{\text{val}} = h_5$</td>
</tr>
</tbody>
</table>

Table 2.1 Boundary condition definition for 3D compressible flows.

2.2 Structural Dynamics

The parachute consists of primarily three components - membranes, cables, and concentrated masses. The membranes and cables are assumed to have no flexural rigidity and can experience large displacements and rotations. However, their material stiffness is assumed to be sufficiently large that the strains are small and hence follow the linear stress-strain relationship. It is assumed that the membranes are in a state of plane stress. While the membranes can experience no flexural rigidity, they can experience compressive forces. The cables are assumed to experience uniaxial stress.

Let $\Omega_s^t \subset \mathbb{R}^{n_{\text{sd}}}$ be the spatial domain of interest with boundary $\Gamma_s^t$ at any time $t \in (0, T)$. Here, the superscript "s" corresponds to the structure. The governing equations obtained from conservation of linear momentum are
\[
\rho^s \left( \frac{d^2 \mathbf{y}}{dt^2} + \eta \frac{d\mathbf{y}}{dt} - \mathbf{f}^s \right) - \nabla \cdot \mathbf{\sigma}^s = 0 \quad \text{on } \Omega_i^s \quad \forall t \in (0, T).
\]  

(2.30)

where \( \rho^s \) is the material density, \( \mathbf{y} \) is the displacement vector, \( \mathbf{f}^s \) are the external body forces acting on the structure, \( \mathbf{\sigma}^s \) is the Cauchy stress tensor, and \( \eta \) is the mass-proportional damping coefficient. The mass-proportional damping provides additional stability, but can significantly affect the dynamics of the structure. In this study, large displacements and rotations, but small strains, are assumed for nonlinear analysis. The second Piola-Kirchhoff stress (force per unit area in the original configuration) tensor, \( \mathbf{S} \), and the Green-Lagrange strain (in the original configuration) tensor, \( \mathbf{E} \), are used to write the constitutive equations using the total Lagrangian formulation. Thus, stresses are expressed in terms of the 2nd Piola-Kirchhoff stress tensor. The 2nd Piola-Kirchhoff stress tensor is related to the Cauchy stress (force per unit area in deformed configuration) tensor, \( \mathbf{\sigma}^s \), by the following kinematic transformation,

\[
\mathbf{S} = \frac{\rho_0}{\rho^s} \mathbf{F} \mathbf{\sigma}^s \mathbf{F}^T,
\]  

(2.31)

where \( \rho_0 \) is the density at original configuration. The deformation gradient tensor \( \mathbf{F} \) is defined as

\[
\mathbf{F} = \mathbf{gG}.
\]  

(2.32)

where \( \mathbf{g} \) is the covariant metric tensor in the current configuration, and \( \mathbf{G} \) is contravariant metric tensor in the original configuration. The linear stress-strain relations (Hookean materials) are assumed for both the membranes and the cables. For the membrane elements, plane stress conditions are assumed, and therefore the
constitutive equations are

\[ S^{ij} = \{\lambda_m G^{ij} G^{kl} + \mu_m (G^{ik} G^{jk} + G^{ij} G^{kl})\} E_{kl}, \]  

where,

\[ \lambda_m = \frac{2\lambda_m \mu_m}{(\lambda_m + 2\mu_m)}. \]  

(2.34)

Here, \( \lambda_m \) and \( \mu_m \) are the Lamé constants. The subscript “\( m \)” corresponds to the membrane. \( G^{ij} \) are the components of the contravariant metric tensor (the Cauchy-Green strain tensor) [26,27]. The Lamé constants are related to the Young’s modulus, \( Y_m \) and the Poisson’s ratio, \( \nu_m \) as follows

\[ \lambda_m = \frac{\nu_m Y_m}{(1 + \nu_m)(1 - 2\nu_m)}, \]  

(2.35)

\[ \mu_m = \frac{Y_m}{2(1 + \nu_m)}. \]  

(2.36)

For the cable elements, the uniaxial stress condition is assumed, and therefore the constitutive equation for cables is given by

\[ S^{11} = Y_c G^{11} G^{11} E_{11}, \]  

(2.37)

where \( Y_c \) is the cable Young’s modulus. The subscript “\( c \)” corresponds to the cable. Both Dirichlet- and Neumann-type boundary conditions are considered, and are expressed as

\[ y = g^s \text{ on } (\Gamma^s_t)_g, \]

\[ n \cdot \sigma^s = t^s \text{ on } (\Gamma^s_t)_h. \]  

(2.38)

where \( (\Gamma^s_t)_g \) and \( (\Gamma^s_t)_h \) are complementary subsets of the boundary \( \Gamma^s_t \), \( n \) is the unit normal vector at the boundary. Both the boundary displacements, \( g^s \) and the traction
forces on the structure surface, $t^s$ are known. The structure is assumed to be initially at rest, with the following initial conditions

$$y = 0, \quad \frac{dy}{dt} = 0 \text{ on } \Omega_0^s.$$ \hfill (2.39)

2.3 Conclusion

This concludes the review of governing equations for compressible and incompressible flows of Newtonian fluids along with structure and cable. In the next chapter, I will discuss the finite element formulations for these equations.
Chapter 3
Finite Element Formulation

In this chapter, I will present the finite element formulations for the governing equations, discussed in Chapter 2, of incompressible and compressible flows, and structural dynamics. I will start with a brief introduction to stabilized finite element formulations in Section 3.1 and then I will present the space-time formulation of incompressible and compressible flows in Section 3.2 respectively. Finally, I will discuss the semi-discrete formulation of structural dynamics which is based on the principle of virtual work.

3.1 Introduction

The finite element method is a numerical tool for obtaining solutions to boundary-value engineering problems governed by partial differential equations. It is especially attractive for problems that involve complex geometries where other numerical methods, such as spectral or finite difference methods, are difficult to apply.

The principle of the method is to replace an entire continuous domain by a number of sub-domains in which the unknown function is represented by simple interpolation functions with unknown coefficients. Thus, the original boundary-value problem with an infinite number of degrees of freedom is converted into a problem with a finite number of degrees of freedom or, in other words, the solution of the whole system is approximated by a finite number of unknown coefficients. Here, I construct a discretization of a weighted residual formulation in order to arrive at a linear matrix equation. The discretization can be applied in space and time. This formulation is called the space-time finite element method. Alternatively, one can carry out the
discretization in space only. Such a method is called a semi-discrete formulation. Details of these methods can be found in a number of references [28, 20, 29]

Galerkin approximation methods are used for the finite element formulations. Galerkin methods are among the most commonly used weighted residual methods. In structural analysis, where often the minimization of energy is the underlying idea, the application of Galerkin methods leads to symmetric matrices and provides "optimal" results. By "optimal" I mean that the solution possesses the best approximation property. The difference between the approximate and the exact solutions is minimized with respect to a certain norm as shown by Brooks et al. in [30].

The situation, however, is very different in the presence of advective terms. The matrix associated with the advective term is non-symmetric and the best approximation property is lost [30]. As a result Galerkin methods applied to these problems are far from optimal and show spurious node-to-node oscillations in the solutions, worsening with growing advection-domination. This not only leads to qualitatively incorrect results but also violates basic physical principles like the second law of thermodynamics [31]. The pollution of the solution with oscillations is dependent on the domination of the advection terms over other terms of the differential equation. Domination of advection is determined by dimensionless numbers such as Reynolds (ratio of inertial to viscous terms) or Peclet (ratio of advection to diffusion terms). The larger these numbers are, the more dominant is the advection term and the stronger is the numerical node-to-node oscillations in the results.

Brooks and Hughes [30] introduced the Streamline-Upwind/Petrov-Galerkin (SUPG) method and this method can be considered as the first successful stabilization technique to prevent oscillations in advection-dominated problems. The SUPG method
introduces artificial diffusion in the streamline direction. The introduction of artificial diffusion is done in a consistent way. This can be interpreted as a modification of the test function in the advection direction. Therefore, the weak (variational) form still satisfies the exact solution of the problem.

Another source of potential instabilities in standard Galerkin methods arises when velocity and pressure interpolation functions are not chosen from compatible spaces. Babuska and Brezzi [32,33] showed that compatible spaces must satisfy the \( \inf - \sup \) conditions. If they are not chosen from compatible spaces, then oscillatory behavior is observed, primarily in the pressure field. Unfortunately, many desirable function spaces are precluded due to Babuska-Brezzi conditions. Most computationally attractive combinations are the ones which employ equal order interpolations. Hughes et al. [34] proposed a consistent way to circumvent the \( \inf - \sup \) condition for the Stokes problem. As a generalization, Tezduyar et al. [35] proposed a Pressure-Stabilized/Petrov-Galerkin (PSPG) formulation for finite Reynolds number flows. The PSPG formulations reduces to the one proposed by Hughes et al. [34] formulation in the limiting case when the Reynolds number tends to zero.

### 3.2 Fluid Dynamics

In order to construct the finite element function spaces for the space-time method, the time interval \((0,T)\) is partitioned into subintervals \( I_n = (t_n, t_{n+1}) \), where \( t_n \) and \( t_{n+1} \) belong to an ordered sequence of time levels \( 0 = t_0 < t_1 < \ldots < t_N = T \). Let \( \Omega_n = \Omega_{tn} \) and \( \Gamma_n = \Gamma_{tn} \). The space-time slab \( Q_n \) is defined as the domain enclosed by the surfaces \( \Omega_n, \Omega_{n+1}, \) and \( P_n, \) where \( P_n \) is the surface described by the boundary \( \Gamma_t \) as \( t \) traverses \( I_n \). As is the case with \( \Gamma_t \), surface \( P_n \) can be decomposed into \( (P_n)_g \)
and \((P_n)_h\) with respect to the type of boundary condition (Dirichlet or Neumann) being applied for a degree of freedom of unknown vector \(d\).

### 3.2.1 Deforming Spatial Domain (DSD) Slab

A space-time slab for a 2D spatial domain is schematically shown in Fig. 3.1. The volume of the slab is the volume traversed by the two dimensional plane in time and is given by \(Q_n\). The boundary is denoted by \(P_n\). For a three spatial dimension problem, one can visualize this as a four dimensional space. The spatial domain at \(t = t_n\) is denoted by \(\Omega_n\) and at \(t = t_{n+1}\) by \(\Omega_{n+1}\). In the beginning of the computations at \(t = 0\), the spatial domain and its boundary is denoted by \(\Omega_0\) and \(\Gamma_0\) respectively. Defining spatial domain in this way allows one to handle moving and deforming bodies.

![Space time slab for deforming spatial domain.](image-url)
3.2.2 DSD/Stabilized Space-Time (SST) Formulation for Incompressible Flows

For each space-time slab, we define the following finite element interpolation function spaces for the conservation variables

\[(S_u^h)_n = \{ u^h \mid u^h \in [H^{1h}(Q_n)]^{n_{sd}}, u^h \doteq g_d^h \text{ on } (P_n)_g, d = 1...n_{sd} \}, \quad (3.1)\]

\[(V_w^h)_n = \{ w^h \mid w^h \in [H^{1h}(Q_n)]^{n_{sd}}, w^h \doteq 0 \text{ on } (P_n)_g, d = 1...n_{sd} \}, \quad (3.2)\]

\[(S_p^h)_n = (V_p^h)_n = \{ q^h \mid q^h \in H^{1h}(Q_n), w^h \doteq 0 \text{ on } (P_n)_g \}, \quad (3.3)\]

where \(H^{1h}(Q_n)\) is the finite-dimensional function space over the space-time slab \(Q_n\). Over the element domain, this space is formed by using first-order polynomials in both space and time. The interpolation functions are continuous in space but discontinuous in time.

The stabilized space-time (SST) formulation of momentum balance (equation (2.1)) and continuity (equation (2.2)) equations for deforming spatial domains (DSD) can be written as follows:

Given \((u^h)^-\), find \(u^h \in (S_u^h)_n\) and \(p^h \in (S_p^h)_n\) such that \(\forall w^h \in (V_w^h)_n\) and \(q^h \in (V_p^h)_n\)

\[
\begin{align*}
\int_{Q_n} w^h \cdot \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h + h^h \right) dQ &+ \int_{Q_n} \varepsilon(w^h) : \sigma(p^h, u^h) dQ \\
+ \int_{\Omega_n} q^h \nabla \cdot u^h dQ &+ \sum_{c=1}^{(n_{el})_n} \int_{\Omega_n^c} \frac{\tau}{\rho} \left[ \rho \left( \frac{\partial w^h}{\partial t} + u^h \cdot \nabla w^h + h^h \right) - \nabla \cdot \sigma(q^h, w^h) \right] \\
\left[ \rho \left( \frac{\partial u^h}{\partial t} + u^h \cdot \nabla u^h + h^h \right) - \nabla \cdot \sigma(p^h, u^h) \right] dQ &+ \sum_{c=1}^{(n_{el})_n} \int_{\Omega_n^c} \delta \nabla \cdot w^h \rho \nabla \cdot u^h dQ \\
+ \int_{\Omega_n} (w^h)^+ \cdot \rho \left( (u^h)^+ - (u^h)^- \right) d\Omega &\quad = \int_{(P_n)_h} w^h \cdot h^h dP. \quad (3.4)
\end{align*}
\]
This process is applied sequentially to all the space-time slabs $Q_1, Q_2, \ldots, Q_{N-1}$. The computations start with $(u^h)^+ = u_0$.

The first three terms, the sixth term, and the right-hand-side comprise the Galerkin formulation of the problem. The first element-level integrals (terms under the summation $\sum_{e=1}^{n_e} n_e$) in equation (3.4) are least-squares terms based on the momentum equation. The second element-level integrals are added to the formulation for numerical stability at high Reynolds numbers. These are least-squares terms based on the continuity equation. The stabilization coefficients, $\tau$ and $\delta$, are defined at the element level. Both stabilization terms are weighted residuals, and therefore maintain the consistency of the formulation. Since the interpolation functions are discontinuous in time, the sixth term weakly enforces continuity of the velocity field across the space-time slabs.

The stabilization coefficients, $\tau$ and $\delta$, are given by

$$\tau = \left[ \left( \frac{2}{\Delta t} \right)^2 + \left( \frac{2||u^h||}{h_e} \right)^2 + \left( \frac{4\nu}{h_e^2} \right)^2 \right]^{-\frac{1}{2}},$$

$$\delta = \frac{h_e}{2} ||u^h||z,$$

where $\nu$ is the kinematic viscosity, $\Delta t$ is the time step, and $h_e$ is a suitable measure of element length. One obvious choice for element length is the maximum edge length. This, however, gives rise to excessive numerical diffusion when the flow direction is not aligned along the maximum edge. A directional element length based on advection direction works better in this case.

The parameter $z$ is defined as

$$z = \begin{cases} \left( \frac{Re_u}{3} \right) & \text{if } Re_u \leq 3, \\ 1 & \text{if } Re_u \geq 3, \end{cases}$$

(3.7)
where $Re_u = \frac{\|h\|}{2\nu}$ is the element Reynolds number. A detailed discussion of these stabilization terms and their origin can be found in [36].

### 3.2.3 DSD/SST Formulation for Compressible Flows

For each space-time slab, we define the following finite element interpolation function spaces for the conservation variables

\[(S_U^h)_{n} = \{ U^h | U^h \in [H^1(Q_n)]^{n_{dof}}, U^h \cdot e_d = g_d^h \quad \text{on} \quad (P_n)_{g}, d = 1 : n_{dof} \}\]  
\[(V_U^h)_{n} = \{ W^h | W^h \in [H^1(Q_n)]^{n_{dof}}, W^h \cdot e_d = 0 \quad \text{on} \quad (P_n)_{g}, d = 1 : n_{dof} \}\]

Over the element domain, the interpolation is constructed by using first-order polynomials in space and, either zeroth- or first-order polynomials in time. Globally, the interpolation functions are continuous in space but discontinuous in time.

The stabilized least-squares space-time formulation of partial differential equation (2.23), discussed in the previous chapter, can be written as

Given $(U^h)_n^-$, find $U^h \in (S_U^h)_n$ such that $\forall W^h \in (V_U^h)_n$

\[
\int_{Q_n} W^h \cdot \left( \frac{\partial U^h}{\partial t} + A^h \cdot \frac{\partial U^h}{\partial x_i} \right) dQ + \int_{Q_n} \left( \frac{\partial W^h}{\partial t} + \frac{\partial W^h}{\partial x_i} \right) \cdot \left( K_{ij}^h \frac{\partial U^h}{\partial x_j} \right) dQ \\
+ \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_{\text{SUPG}} \left[ \frac{\partial W^h}{\partial t} + (A^h_k)^2 \frac{\partial W^h}{\partial x_k} - \frac{\partial}{\partial x_k} \left( K_{kl}^h \frac{\partial W^h}{\partial x_l} \right) \right] dQ \\
+ \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_{\text{SHOC}} \left( \frac{\partial W^h}{\partial x_i} \right) \cdot \left( \frac{\partial U^h}{\partial x_i} \right) dQ = \int_{(P_n)_h} W^h \cdot h^h dP. \tag{3.10}
\]
The stabilization coefficients, $\tau_{\text{SUPG}}$ and $\tau_{\text{SHOC}}$ will be discussed in Section 7.2. The notational conventions used in equation (3.10) are shown below

$$\left(U^h_n\right)_{\pm} = \lim_{\varepsilon \to 0} U(t_n \pm \varepsilon)$$  \hspace{1cm} (3.11)

$$\int_{Q_n} ... dQ = \int_{I_n} \int_{\Omega^h} ... d\Omega dt$$  \hspace{1cm} (3.12)

$$\int_{P_n} ... dp = \int_{I_n} \int_{\Gamma^h} ... d\Gamma dt$$  \hspace{1cm} (3.13)

The solution to equation 3.10 is obtained sequentially for all space-time slabs $Q_1, Q_2, ..., Q_{N-1}$. The computations start with the initial condition

$$\left(U^h\right)_{0}^{+} = U_0.$$  \hspace{1cm} (3.14)

3.2.4 Time Integration

After carrying out the spatial discretization of time dependent fluid dynamics equations, one arrives at the following set of coupled non-linear ordinary differential equations

$$Ma + Cv + N(v) = F,$$  \hspace{1cm} (3.15)

where $v$ is the vector of nodal values of unknowns, $a$ is its time derivative, $M$ is the mass matrix, $C$ is the damping matrix, $N$ is the non-linear vector function of $v$ representing the rest of the terms from the steady state equations, and $F$ is the force vector. To solve this system of equations, a predictor/multi-corrector transient algorithm is used [30].

Given $v_n$ and $a_n$, the goal is to find $v_{n+1}$ and $a_{n+1}$. The subscript $n$ implies that the variable is defined at time $t = t_n$ where $n$ is the time step counter. This algorithm can be summarized in the following steps.
1. Predictor phase

- Set nonlinear iteration counter, \( i = 0 \)
- \( \mathbf{v}_{n+1}^i = \mathbf{v}_n + \Delta t (1 - \alpha) \mathbf{a}_n \)
- \( \mathbf{a}_{n+1}^i = 0 \)

2. Corrector phase

- Compute \( \mathbf{R}_{n+1}^i = \mathbf{F}_{n+1}^i - \left[ \mathbf{M} \mathbf{a}_{n+1}^i + \mathbf{C} \mathbf{v}_{n+1}^i + \mathbf{N}(\mathbf{v}_{n+1}^i) \right] \)
- Solve \( \mathbf{M}^* \Delta \mathbf{a}_{n+1}^i = \mathbf{R}_{n+1}^i; \mathbf{M}^* = \mathbf{M} + \alpha \Delta t \mathbf{C} \)
- Update: \( \mathbf{v}_{n+1}^{i+1} = \mathbf{v}_{n+1}^i + \alpha \Delta t \Delta \mathbf{a}_{n+1}^i; \mathbf{a}_{n+1}^{i+1} = \mathbf{a}_{n+1}^i + \Delta \mathbf{a}_{n+1}^i \)
- Advance, \( i = i + 1 \); and repeat predictor phase until converged

Here, \( \mathbf{M}^* \) is a free parameter and can be replaced or approximated with anything we want, provided that the iterations converge. The idea is to approximate \( \mathbf{M}^* \) with something sufficiently cheap to converge within a reasonable number of iterations. Some choices for \( \mathbf{M}^* \) are

1. \( \mathbf{M}^* = \mathbf{M}_L + \alpha \Delta t \ \text{diag}(\mathbf{C}) \)
2. \( \mathbf{M}^* = \text{diag}(\mathbf{M}) + \alpha \Delta t \ \text{diag}(\mathbf{C}) \)
3. \( \mathbf{M}^* = \text{blockdiag}(\mathbf{M}) + \alpha \Delta t \ \text{blockdiag}(\mathbf{C}) \)

where subscript "L" stands for lumped matrix, \( \text{diag}(\cdot) \) stands for the diagonal terms of a matrix, and \( \text{blockdiag}(\cdot) \) implies that the approximated matrix is formed by taking small diagonal blocks from the full matrix. The size of each block is usually taken as \( n_{dof} \times n_{dof} \).
The parameter $\alpha \in [0, 1]$ governs the stability and accuracy of the method. To obtain the steady state solution, $\alpha$ is set to one (backward difference type). To carry out the time-accurate computations, the parameter $\alpha$ is set to 0.5 (second order accurate in time). The parameter $\alpha = 0$ gives a fully explicit type algorithm (see [37] for further details).

3.3 Structural Dynamics

3.3.1 Semi Discrete formulations

A semi-discrete finite element formulation for the Structural Dynamics (SD) equations of motion is obtained from the principle of virtual work. Finite displacements of the structure are taken into account by using a total Lagrangian description of the problem.

For the structural domain, $\Omega^s$, the corresponding finite element function spaces $S^h$ and $V^h$ are defined as follows

\begin{align}
S^h &= \{ y^h | y^h \in [H^{1h}(\Omega^s)]^n_{sd}, y^h = (g^s)^h \text{ on } \Gamma^s_{g} \}, \\
V^h &= \{ w^h | w^h \in [H^{1h}(\Omega^s)]^n_{sd}, w^h = (g^s)^h \text{ on } \Gamma^s_{g} \}.
\end{align}

Here, $H^{1h}(\Omega^s)$ is a finite-dimensional function space over $\Omega^s$. The total Lagrangian formulation is then written as follows:

Find $y^h \in S^h$ such that $\forall w^h \in V^h$

\begin{align}
\int_{\Omega^s} \rho^s \frac{d^2 y^h}{dt^2} \cdot w^h d\Omega^s + \int_{\Omega^s} \nu \rho^s \frac{dy^h}{dt} \cdot w^h d\Omega^s + \int_{\Omega^s} S^h : \delta E d\Omega^s d\Omega^s \\
= \int_{\Gamma^s_i} t \cdot w^h d\Gamma^s + \int_{\Gamma^s_i} \rho^s \phi^s \cdot w^h d\Gamma^s.
\end{align}
Here, the weighting function $w^h$ is also the virtual displacement. The pressure contribution is included in the traction force $t$. The pressure term is a "follower force" (since it follows the deforming structural geometry) and thus increases the overall nonlinearity of the formulation. The left-hand-side terms of equation (3.18) are referred to in the original configuration of structure and the right-hand-side terms for the deformed configuration of structure at time $t$.

3.3.2 Time Integration

The Hilber-Hughes-Taylor (HHT) method [38] is used to carry out the time integration. The HHT method (also called the $\alpha$-method) is an extension to the Newmark method [39]. With the HHT method it is possible to introduce numerical dissipation (numerical damping) without degrading the order of accuracy, and this numerical damping can be continuously controlled. Decreasing $\alpha$ means increasing the numerical damping. This damping is low for low-frequency modes and high for high-frequency modes. It is second-order accurate and unconditionally stable if $-\frac{1}{3} \leq \alpha \leq 0$, $\gamma = \frac{(1-2\alpha)}{2}$, and $\beta = \frac{(1-\alpha)^2}{4}$ where $\alpha$, $\beta$, and $\gamma$ are HHT parameters as defined in [38]. The HHT parameters determine the accuracy and stability of the time integration method.

3.4 Conclusion

In this chapter, the DSD/SSD formulations of fluid flows in incompressible and compressible flow regime were discussed. The finite element formulations for structural dynamics were also presented. Appropriate time integration methods for both fluid and the structure were discussed.

In the next chapter, I will review some computational challenges in FSI simulations
which will be followed by the results from the FSI simulations of clusters of G–12 parachutes under various circumstances.
Chapter 4
Simulations of Cluster of G–12 Canopies and Challenges

My objective is to study fluid-structure interaction behavior of three dimensional complex geometries with particular emphasis on parachute systems. I use computational fluid dynamics and structural dynamics tools for this.

The parachute canopy is made of very flexible fabric materials. It responds quickly to any small changes in fluid behavior. Consequently, there is a strong interaction between the fluid and structure. The bigger the canopy, the stronger the interactions. Stronger interactions make the problem very nonlinear in nature and hence increase the difficulty of numerical modeling. In this chapter, I will first mention some of the numerical challenges encountered in the FSI simulations. Then, I will present results from the FSI simulations of clusters of two and three G-12 parachutes.

4.1 Computational Challenges

There are many computational challenges that one faces in performing the FSI simulations accurately and efficiently. They are turbulence modeling, fluid-structure interaction coupling, the convergence of a nonlinear iteration loop, capturing the physical discontinuity in the flow field, efficiently and accurately solving a large set of linear equations on parallel computers, improving the parallel performance etc. I am going to mention some of these challenges here and will address a few in the following chapters.
4.1.1 Fluid-Structure Coupling

The FSI solver consists of three modules: a fluid dynamics (FD) solver, a structural dynamics (SD) solver, and a linear elasticity solver for mesh motion. The information exchange between FD/mesh motion and SD is carried by a coupling mechanism. There are two evident coupling mechanisms – direct coupling and iterative coupling.

In direct coupling, all structural, mesh and fluid variables are fully coupled and updated simultaneously by solving a single set of equations. Let \( N_F, N_M \) and \( N_S \) be the nonlinear operators corresponding to the governing equations for the fluid, mesh and structure respectively. Here, “F”, “M”, and “S” stands for “Fluid”, “Mesh”, and “Structure” respectively. After employing the discretization technique on the finite element formulations discussed in the previous chapter, one can simplify and write them in the form of nonlinear ordinary scalar equations as follows:

\[
N_S(d_S, d_M, d_F) = 0, \tag{4.1}
\]
\[
N_M(d_S, d_M, d_F) = 0, \tag{4.2}
\]
\[
N_F(d_S, d_M, d_F) = 0, \tag{4.3}
\]

where \( d_S, d_M \) and \( d_F \) are the vectors of unknown variables for structure, mesh and fluid respectively. By carrying out the Taylor expansion on the nonlinear scalar functions and keeping only the first order terms (the Newton-Raphson’s method), one can then linearize the systems and derive the following set of linear equations

\[
Ax = \begin{bmatrix}
    A_{SS} & A_{SM} & A_{SF} \\
    A_{MS} & A_{MM} & A_{MF} \\
    A_{FS} & A_{FM} & A_{FF}
\end{bmatrix}
\begin{bmatrix}
    x_S \\
    x_M \\
    x_F
\end{bmatrix} = \begin{bmatrix}
    b_S \\
    b_M \\
    b_F
\end{bmatrix}, \tag{4.4}
\]
where $A_{ij} = \frac{\partial N_i}{\partial d_j}$, $x_i = \Delta d_i$ and $b_i = -N_i$. Here, the indices $i$ and $j$ take "S", "M" or "F". The matrix $A_{ij}$ is called the jacobian matrix and determines the nonlinearity of the problem. The cross term of matrix $A$ determines the strength of coupling between fluid and structure. The matrix $A$ is not symmetric.

The resulting matrix from the fully coupled systems is very large and sometimes unmanageable. It will require more computational power to solve each linear solve-iteration and more memory to store the data. Additionally, it is extremely cumbersome or nearly impossible to find the coupling terms (e.g. $A_{SF}$) for most practical life applications. As an alternative to a fully coupled system, I use an iterative coupling approach. The use of a fully coupled system, however, would give a better convergence of the nonlinear iterations for the coupled system.

In iterative coupling, systems of equations for structure, mesh and fluid are solved one-by-one within the nonlinear iteration loop. Coupling is achieved through the transfer of matching information on fluid-structure interface between structure, mesh or fluid. The transferred information from one is treated as the boundary conditions for the other. First, I compute the surface displacements and velocities by solving the structural dynamics (SD) equations. Surface displacements from the SD solution are used to update the fluid mesh. The fluid mesh is modeled as a "pseudo-solid", and surface displacements are treated as Dirichlet type boundary conditions in the linear elastic mesh moving scheme. After solving the mesh equations, updated mesh coordinates for the fluid mesh are obtained. An updated mesh is used to solve the fluid dynamics (FD) equations. Surface velocities from the SD solution are treated as Dirichlet boundary conditions in the FD solver. In return, parachute nodal surface tractions are transferred from the fluid for treatment in the SD solver.
In matrix form, these equations can be written as

\[
\begin{bmatrix}
A_{SS} & 0 & 0 \\
A_{MS} & A_{MM} & 0 \\
A_{FS} & A_{FM} & A_{FF}
\end{bmatrix}
\begin{bmatrix}
x_S \\
x_M \\
x_F
\end{bmatrix}
= 
\begin{bmatrix}
b_S \\
b_M \\
b_F
\end{bmatrix} \quad (4.5)
\]

This approach is known as the block-iterative coupling technique (See equation (4.5)). This technique can be viewed as an approximate Newton-Raphson method, where the fluid-structure coupling matrices are ignored while solving the linear equation system at every nonlinear iteration. Difficulties will be encountered with this coupling approach for parachute FSI simulations when the structure is very light and the structural response is very sensitive to small changes in fluid dynamics. In these kinds of interactions, achieving acceptable levels of convergence in the nonlinear equation system can be difficult. In certain circumstances the block-iterative coupling approach may not provide adequate convergence or may fail to converge. Therefore, modifications or alternative approaches must be considered. Different approaches can be taken to improve convergence of solutions for these problems. One such approach is to artificially augment the inertia of the structure by increasing the first term on the left-hand-side of the SD equations. This results in softer coupling by adding mass in the inertia terms, but not in the gravitational body terms. This approach has been used in some simulations where it was difficult to get required convergence. In another approach, a simple modification is made to the block-iterative coupling. Here, difficulties due to the light weight of the parachute canopy are alleviated by artificially increasing the mass matrix that arises from the discretization to reduce the over-response of the structure during the coupling iterations. In another words, one has reduced the nonlinearity of the coupling effects by doing so. However, the
right-hand-side vectors are not altered, which allows the coupled system to converge to the correct structural mass solutions.

4.1.2 Mesh Generation and Update

Proper selection of the appropriate mesh update approach depends on several factors. These include the complexity of the moving boundary (or interface) as well as the overall geometry, the unsteadiness of the moving boundary (or interface), and the qualities of the starting mesh. In general, the mesh update should have two components: moving the mesh for as long as it is possible, and full or partial remeshing (i.e. generating a new set of elements, and sometimes also a new set of nodes) when element distortion becomes too high.

In mesh moving strategies, our only requirement for the mesh motion is that at the moving boundary (or interface) the normal velocity of the mesh has to match the normal velocity of the fluid. Beyond that, the mesh can be moved in any way desired, with the main objective being to reduce the frequency of remeshing. In 3D simulations that rely on an automatic mesh generator, the cost of automatic mesh generation becomes a major reason for trying to reduce the frequency of remeshing. Additionally, mesh generation is often done on a single CPU, losing the parallel performance. Furthermore, when I remesh I need to project the solution from the old mesh onto the new one. This step introduces projection errors along with an additional computational cost that is not trivial for 3D computations involving complex geometries. Projection errors can destroy the divergence free condition in the domain giving rise to pressure oscillations [4]. All of these factors provide a strong motivation for utilizing mesh update strategies which minimize the frequency of remeshing.
My parachute simulations involve complex geometries and arbitrary motions for which it is difficult to design special-purpose mesh moving techniques. For parachute clusters, the behavior of parachute interactions becomes even more erratic. For these problems I use an automatic mesh moving scheme [40,41,4] to move the nodal points, as governed by the equation of linear elasticity, and where the smaller elements enjoy more protection from mesh deformation. The motion of the internal nodes is determined by solving these additional equations. The boundary conditions for the mesh motion are specified in such a way that they ensure the matching of the normal velocity of the fluid at the interface. In all the fluid-structure interactions simulations presented in this thesis I use this automatic mesh moving technique.

4.1.3 Solving Linear Systems

After carrying out finite element discretization and linearizing the nonlinear system, one ends up with a large set of linear equations given by $Ax = b$. For most of my problems, the size of matrix $A$ is on the order of millions-by-millions. So, it would require a large amount of memory to store the entire matrix $A$. The matrix $A$ is usually non-symmetric and sparse. Using the sparsity patterns of the matrix, only non-zeros of matrix $A$ are stored in sparse-storage format. I use a preconditioned GMRES [42] iterative method on a parallel computer to arrive at the solutions of my linear system. Choosing the right preconditioner has been a subject of debate. One of the simplest preconditioning matrix is $diag(A)$ (i.e. the diagonal entries of $A$ as the preconditioning matrix for the GMRES solver) which is also very easy to implement on parallel computers and gives good parallel performance. For most of the results presented in this thesis I used the diagonal preconditioner.
In compressible flow (discussed in details in Chapter 7), I found that the blockdiag-preconditioner (i.e. the blocks along the diagonal entries of A) gives a better convergence than the diag-preconditioner. The diag-preconditioner required smaller time step to get the acceptable convergence of linear solver whereas the blockdiag-preconditioner allowed one to choose a bigger time step. With a smaller time step it takes essentially a very long time get steady state or fully developed solutions. The size of each diagonal block used is $n_{dof} \times n_{dof}$ (5×5 in 3D problems). Similar kinds of convergence issues also arise in FSI simulations. The stronger the coupling between the fluid and the structure, the worse the convergence of the FD/SD solver.

The condition number of matrix A goes up with larger time steps or strong fluid-structure interactions. Hence, a better choice of a preconditioner matrix will let one choose a larger time step to have acceptable convergence of the linear solver. More studies are needed to find a better preconditioning matrix (LU/ILU, CG/BiCG, full block matrix in high gradient areas etc.) for the GMRES solver to efficiently and inexpensively solve these huge linear systems.

4.1.4 Parachute Contact

Dynamic contact phenomena can be experienced in parachutes and parachute clusters. It presents an important challenge and is a current barrier in my parachute modeling efforts. Dynamic contact can occur between gores (gore-to-gore contact) in a single canopy when one or more gores collapses. It can also occur between different canopies (parachute-to-parachute contact) in a parachute cluster. Gore-to-gore contact is also present in the opening stages of a parachute. In each of these situations, contact algorithms are necessary to prevent the penetration or the contact
of two surfaces in the FSI simulations. Without a contact model capability, the FSI simulations of parachute are impossible at the onset of contact because the FD solver cannot handle negative or zero volume elements. In Chapter 5, I present a method to overcome the parachute-to-parachute contact issues.

4.2 Cluster of Two G-12 Parachutes

A single G–12 parachute is used to drop cargo weighing up to 2,200 lb. There is a current interest in the Army for parachute systems that can drop cargos exceeding 2,200 lb. In order to achieve this objective, multiple parachutes in cluster configurations are often used.

The opening stage of a parachute to a fully inflated shape is a critical issue. Usually more than one stage is used to get the fully inflated shape of parachutes in a cluster configuration. One example of multi-stage opening is the use of drogue parachutes. First, a drogue parachute opens and then it pulls the other two G–12 parachutes. After all the parachutes are fully opened, the challenge is to safely drop the cargo into the hands of the right people. The strong parachute-to-parachute aerodynamic interaction can potentially destroy the efficiency of the system if it is not carefully designed. Various researchers have analyzed the cluster of parachute systems using both drop tests and computer simulations. Butler [8] presented the qualitative results from the airdrop tests using a drogue parachute to deploy the G–12 canopies. From the snapshots of a cluster of three G–12 canopy airdrop tests, one clearly observes strong interactions between fluid and canopies. Sahu et al. [43] carried out the numerical simulations using a quasi-static approach by imposing a symmetry condition for the three canopies in an attempt to predict the equilibrium configuration of a
cluster of three half-scaled C-9 parachutes. Stein et al. [44] presented results from the semi-discrete simulations for the aerodynamic interactions between the canopies of parachute clusters of varying numbers and arrangements. In these computations, the canopies were assumed to be rigid. In real life, however, the canopy of a parachute deforms in accordance with aerodynamic forces. The placement of parachutes in cluster configuration does not remain stationary. Therefore, FSI simulations are needed to replicate the real life scenario and to understand the dynamics of cluster of parachutes.

Here, I present results from the FSI simulations of a cluster of two G–12 parachutes. The results from a cluster of three G–12 parachutes will be discussed in the next section. The DSD/SST formulations as discussed in Chapter 2, along with appropriate mesh-update strategies, allowed me to study the interaction of canopies in a cluster. I will also present results from the soft-landing simulations of a cluster of two G-12 parachutes.

4.2.1 Construction and Properties of Single G–12 Parachute

The G–12 is a huge parachute and is used to drop heavy cargo. It has a flat circular geometry when it is not inflated. The canopy is made of a regular polygon of N sides, constructed as a flat surface with a central vent. This design is the basis for most circular parachutes. Flat circular parachutes are simple and economical to construct, handle and inspect. They are often used in clusters. The G–12 parachutes are very reliable. A schematic of flat constructed parachute and gore layout is given in Fig. 4.1. Here, the parameters $h_s$ and $e_s$ are calculated using the total projected surface area $S_o$ of a parachute in flat configuration and number of gores $N$ as follows

$$h_s = sqrt[S_o/N \tan(180^\circ/N)],$$

(4.6)
\[ e_s = 2h_s \tan(180^\circ/N). \] (4.7)

An inflated profile of a single G-12 parachute is shown in Fig. 4.2. Table 4.1

![Construction Schematic]

**Figure 4.1** Gore layout (left) and construction schematic (right) of a G-12 parachute. \( D_c \) is the diameter of the flat parachute, \( D_v \) is the diameter of the vent.

contains the geometrical, material and physical properties of G-12 parachutes.

Using the material properties from table 4.1, I calculate the physical properties of membrane and cable elements. They are the input variables for numerical modeling and are given in the table 4.2. The riser cables are eight times stiffer than the suspension lines because there are eight suspension lines per riser. I used radial
Figure 4.2  Inflated profile of a G-12 parachute (courtesy of U.S. Army).

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Diameter (in flat configuration)</td>
<td>64 ft</td>
</tr>
<tr>
<td>Number of gores, $N$</td>
<td>64</td>
</tr>
<tr>
<td>Number of risers</td>
<td>8</td>
</tr>
<tr>
<td>Terminal descent velocity, $V_o$</td>
<td>28 ft/s</td>
</tr>
<tr>
<td>Max. deployment velocity</td>
<td>200 knots = 341 ft/s</td>
</tr>
<tr>
<td>Max. Payload</td>
<td>2,200 lbs</td>
</tr>
<tr>
<td>Total parachute weight</td>
<td>130 lbs</td>
</tr>
<tr>
<td>Canopy (nylon) strength</td>
<td>2.25 oz/yd$^2$</td>
</tr>
<tr>
<td>Suspension line (nylon) strength</td>
<td>1000 lbs</td>
</tr>
</tbody>
</table>

Table 4.1  G–12 parachute system: Geometrical, material and physical properties.
<table>
<thead>
<tr>
<th></th>
<th>Membranes</th>
<th>Cables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Young’s modulus (lb/ft²)</td>
<td>$2.0 \times 10^6$</td>
<td>$12.28 \times 10^6$</td>
</tr>
<tr>
<td>Density (slugs/ft³)</td>
<td>12.27</td>
<td>12.27</td>
</tr>
<tr>
<td>Thickness (ft)</td>
<td>$10^{-4}$</td>
<td>-</td>
</tr>
<tr>
<td>Area (ft²)</td>
<td>-</td>
<td>$10^{-4}$</td>
</tr>
</tbody>
</table>

**Table 4.2** G–12 parachute system: membrane and cable properties.

reinforcement cables which have the same material properties as that of suspension lines. Radial reinforcement cables stiffen the skirt of the parachute and reduce the extent of nonlinear coupling between fluid and structure. This helps to get a better convergence for nonlinear iterations.

### 4.2.2 Previous Simulations and Motivation

In [45, 2, 44], Stein et al. presented results from a series of simulations for the aerodynamic interactions between the canopies for three to six canopies in a cluster. For these simulations, the parachute model was represented by a set of identical C–9 canopies that are positioned and oriented relative to a prescribed confluence point. Note that C–9 is a personnel parachute which is rarely used in cluster configuration for airdropping cargo. In all of the simulations, parachute canopies were assumed to be rigid. Deformation, translation and/or rotation were not allowed. Results from these simulations motivated me to carry out fluid–structure interaction simulations to better understand the interactions between multiple canopies. To carry out the FSI study, I used more than one G–12 parachutes in a cluster configuration. Unlike C–9 parachutes, the G–12 is a heavy duty cargo parachute and is often used in cluster
configuration. Both deformation of parachute canopies and rotation of parachutes with respect to each other are allowed. As the number of parachutes in the cluster configuration increases, the level of difficulty in modeling also increases.

4.2.3 Problem Setup

The G–12 is a 64 ft diameter cargo parachute designed to deliver a payload of 2200 lb at a descent speed of approximately 28 ft/s. Clusters of G–12 parachutes are commonly used to deliver larger payloads. The G–12 is constructed with 64 suspension lines which extend to risers. For a single G–12 parachute, the confluence point of the risers is connected to a retraction cable which supports the payload. For a cluster of two G–12 parachutes, the retraction cable is connected to two cables, which connect to the confluence point of the risers for the two G–12 parachutes.

The structural model is composed of membranes, cables, and concentrated masses. The canopy is modeled with triangular membrane elements. Linear cable elements are used to model the suspension lines, radial reinforcements along the canopy, risers, and payload support cables. In each example, the payload is modeled with a single concentrated mass. Material properties are selected to be representative of the G–12. A model for the cluster of G–12 parachutes in constructed (unstressed) and inflated (prestressed) configurations is shown in Fig. 4.3.

Several preparations are required for each fluid–structure interaction simulation. First, a stand alone structural deformation simulation is carried out to determine the inflated (i.e. prestressed) shape of the G–12. The initial inflation pressure is assumed to be equal to stagnation pressure. From my experience, I have observed that this pressure gives a better approximation for the initial shape of the parachute
canopies. The prestressed configuration for the G–12 cluster is shown in Fig. 4.3 (right). The unstressed configuration is shown in Fig. 4.3(left). Using the parachute canopy from the prestressed configuration, a fluid mesh is generated and a stand alone fluid simulation is carried out to obtain a developed flow. Fluid simulations are expensive and time consuming. Therefore, to arrive at developed flow quickly, first semi-discrete [41,36] formulation is used to compute the flow field. Semi-discrete formulations are first order in time. Using this solution as the initial condition, space-time [13] computations are carried out. I start the simulations with a first-order-time accurate integration scheme. This helps to clear the start-up vortices quickly. Start-up vortices are generated due to large differences in the initial conditions and the exact solutions. After the start-up vortices are cleared from the domain, the second order accurate time-integration scheme is applied.

Now, the stage is set to start fluid-structure interaction simulations. Using the results from space-time simulations, a fluid-structure interaction computation is carried out. To remove the mismatch in the initial guessed prestressed configuration, I
use the pressure ramping to soften the exchange of information.

4.2.4 Pressure Ramping

For some cases, the ramping of pressure was absolutely necessary to obtain converged solutions. It was desirable especially in those cases where the guess for the initial shape of the parachute canopies was not close to the real life parachute geometry. The convergence of nonlinear interactions at the start of the FSI simulations was very erratic in the absence of a ramping technique. Pressure ramping was successfully used in these cases.

Here, the information exchange between structure and fluid is linearly ramped. The ramping is carried out such that the acting pressure on the canopy is equal to the guess pressure (used for inflating the parachute) at the start of the FSI simulations and equal to the fluid pressure at the end of the ramping. This is achieved by

\[ \Delta P_S \leftarrow r(t) \Delta P_G + (1 - r(t)) \Delta P_F, \]  

where \( \Delta P_S \) is the effective pressure applied on the structure, \( \Delta P_G \) is the guess pressure used to get the inflated shape of the parachute, and \( \Delta P_F \) is the pressure drop across the canopies coming from fluid simulations. The ramping factor \( r(t) \) is assumed to be linear in time and varies from 0 to 1 in time \( t_{ramp} \) where \( t_{ramp} \) is the ramping time.

4.2.5 Incompatible Meshes

Incompatible meshes [40] for structure and interface surface are used. This gives us freedom to choose different types (e.g. higher order bi-quadratic elements for parachute canopy and triangular mesh for fluid-structure interface) and different refinements. In this problem I choose the same type of triangular elements for both
canopy and fluid-structure interface meshes. However, a canopy mesh is very refined (to properly capture and model the gores of the canopy) as compared to a fluid-structure interface mesh. The same refinement on a fluid-structure interface mesh (in case compatible case) results in an unbelievably large number of elements in the fluid mesh. This is not manageable with the given computational resources. Additionally, I believe that one does not need such refined elements on the interface to capture the fluid dynamics. Incompatible meshes allows one to choose different refinements to model the fluid interface and the structure canopy.

Whether one uses different types of elements or different levels of refinements, a projection mechanism is needed to transfer the information from fluid to structure and vice-versa. Errors (usually called projection errors) come along with the projection. Some of the projection techniques are the linear projection and the least-square projection. I used the least-square projection technique to achieve this goal. The least-square projection technique minimizes the sum of residuals in the entire domain. The least-square projection is achieved by solving the following minimization problem

$$min \int_{\Gamma^f} ||d_S - d_F||^2 d\Gamma^f,$$  \hspace{1cm} (4.9)  

where $d_F$ (the fluid variables e.g. pressure drop across the canopy) is projected to $d_S$ (the structure variables e.g. pressure applied on the canopies). Integration is carried over the interface domain $\Gamma^f$. Similar techniques are used to project the velocity and displacement from structure to fluid-interface.
4.2.6 Pinned Payload

The pinned payload case corresponds to the wind–tunnel testing where parachutes are fixed to a confluence point. The payload is pinned and is not allowed to move in any direction. The objective here is to understand the aerodynamics of two G–12 flexible canopies.

Fig. 4.4 presents the parachute cluster showing the deformed shapes and canopy pressure at different instances of time. Color coding ranging from blue to red represents low to high magnitudes of pressure, respectively. At the time $t = 00.27s$, two parachute canopies are close to each other. They move away from each other as time progresses. This may be because the initial configuration that I assumed to start the FSI simulations is not in equilibrium. The canopies become flatter ($t = 03.36s$) as a result of change in pressure distribution. One notices a strong interaction between the fluid and the canopies. Pressure distribution keeps changing with time. Parachute canopies are made of very flexible fabric materials and modeled as a membrane structure in the FSI simulations. As a result, the canopies quickly respond to any change in pressure distribution by adjusting their shape as observed in Fig. 4.4. As a result of dynamic behavior, two canopies come closer to each other and then move farther away. This is a time dependent phenomenon. At time $t = 03.36s$, these two canopies start going in conical motion in counterclockwise direction about their vertical axis. This motion can be clearly seen in Figs. 4.5 (left and right). They rotate by about 45° in 10.71s. Interestingly, this conical motion has also been observed in real life scenarios. I am not sure how the direction of rotation (clockwise or counter-clockwise) is chosen. I believe that a slight asymmetry in the mesh generated by the automatic mesh generator can give rise to counter-clockwise as being the preferred direction. In
real life, the preferred direction of rotation could be initiated from slight asymmetry in flow conditions introduced by a wind profile or a wake field of an airplane.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure4.4.png}
\caption{G-12 parachute cluster with a fixed payload showing the deformed shapes and canopy pressure.}
\end{figure}

Fig. 4.5 shows the pressure distribution on a plane cutting through the volume mesh and passing through the parachute canopy surfaces. As expected, there is a high pressure region inside of the canopy and lower pressure outside. This pressure gradient keeps the canopy inflated. In Fig. 4.5(middle), a pair of vortices (the blue color spot signifies lower pressure at the core of a vortex) that are shed by the canopy can be seen in the downstream. These pairs of vortices shed from the left and right canopies are of different strengths implying that the aerodynamic responses of parachutes are not symmetric. Fig. 4.5(right) shows a close view of the mesh viewed from the
top and colored with pressure at $t = 10.74s$. Mesh is very fine close to the canopy surface. Whole simulations required 10 remeshes. Each remesh resulted in about 3 million elements and 0.5 million nodes. This implies that about 4 million unknown fluid variables were computed at each nonlinear iteration in each time step. Total computational time required was about 200 hours using 32 processors on 16 nodes of a Linux cluster (2GB RAM/node, 1.7GHz P4 Xeon with 1.2GB/s Myrinet Switch) at Rice University.

![Images of parachute simulations]

$t = 00.03$ s, Top view $t = 05.94$ s, Side view $t = 10.74$ s, Top view

**Figure 4.5** G-12 parachute cluster with a fixed payload showing pressure field (left and middle) and mesh colored with pressure (right) on a cross-section plane cutting through the parachutes.

### 4.2.7 Airdrop with Payload

The pin was removed from the payload and the parachutes were released to fall with the payload weight. The total weight of the payload was 4,400 lb which is twice as heavy as that for a single parachute case. The flow results from a pinned case at $t = 06.10s$, when the FSI results were fully developed and free from start up conditions, were used as the initial conditions. Fig. 4.6 shows the shape of the parachutes and their fall in the vertical direction at three instants of time. In this
figure, the fall position is determined with respect to an object falling with an assumed freestream velocity of 28 \( ft/s \). The canopies are colored with pressure. With time they fall downward as expected implying that the terminal velocity in this case is higher than the assumed freestream velocity. Here, the cluster of parachutes with a

\[
\begin{align*}
t &= 06.45 \, s \\
t &= 08.50 \, s \\
t &= 09.87 \, s
\end{align*}
\]

**Figure 4.6** Airdrop of payload using cluster of two G–12 parachutes: Parachute canopies colored with pressure.

payload of 4,400 \( lb \) is cruising at a terminal velocity of around 31 \( ft/s \) instead of 28 \( ft/s \). Terminal velocity for a single parachute case is 28 \( ft/s \). As expected, I lost the efficiency in cluster configuration.

This implies that the impact velocity for the cluster at the time of landing would be higher than for a single parachute. This is not advisable for the safety of the payload. So, to find the terminal velocity of 28 \( ft/s \) in this case, I simulated a few more cases with lower payloads as shown in Fig. 4.7. Following the trend of fall velocity for various payload weights from this figure, one notices that the lighter the payload, the lower the terminal velocity. One observes that the ideal weight for a payload would be about 3,600 \( lb \). This weight will give a terminal velocity of 28 \( ft/s \).
Figure 4.7  Airdrop of a payload using a cluster of two G–12 parachutes: Fall velocity, $V_z$ (ft/s) v/s time, $t$ (s) for different weight of payload.
4.2.8 Soft Landing

I carried out simulations of two G–12 cargo parachutes to study the soft landing behavior of such systems. "Soft landing" implies making the landing of paratroopers or cargos softer. Various techniques are used to achieve this. Pneumatic Muscle Actuator (PMA) is one technology to achieve this objective. In PMA, the risers are made of pneumatic muscles. When the cargo is very close to the ground the muscle is pressurized. The length of PMA starts contracting and in the process the canopies get pulled downward and the payload gets pulled upward reducing the impact velocity.

Previous soft landing simulations focused on the soft landing for single T–10 personnel parachute and on comparisons with drop test data [46]. These simulations provide a level of confidence for simulations of soft landings with our computational methods. A follow-on simulation was carried out for a single G–12 cargo parachute with a 2,200 lb payload and a G–12 parachute weight of 130 lb [47]. In this example, soft landing is modelled by reducing the natural length of the retraction cable from 12.80 ft to 2.88 ft in 1 second.

While soft landing systems have been shown to be effective for single parachutes, little is known about the retraction process for a cluster of parachutes. In the second simulation, the soft landing of a 4,400 lb cargo with a cluster of two G–12 parachutes is modeled to study the behavior during and after the retraction process.

The soft landing computation is carried out after the parachute has reached a state of terminal descent. In this computation, a 26 ft retraction device is modeled with a cable that connects the confluence point of the two parachutes to the payload. Parachute–payload retraction is modelled by reducing the natural length of the retraction cable by 7.68 ft in 0.27 seconds. Finally, a series of computations are
carried out with the retraction cable held at its reduced length after the retraction is completed to study the post retraction behavior of the G–12 cluster. In addition to the soft landing simulation, a second computation is carried out without soft landing retraction. This no-retraction case is a shorter computation and serves as a baseline for comparison with the soft landing case in Fig. 4.8–4.9.

The aerodynamic drag force on the G–12 cluster during and after soft landing retraction is shown in Fig. 4.8. The effect of the soft landing is apparent from the sharp increase in the drag force during retraction. It is important to note the dramatic drop in drag shortly after retraction ends. This drop is accompanied by canopy collapse and suggests that “harder” landings can be experienced if ground impact is delayed too long after retraction. The payload velocity during descent of the G–12 cluster with and without soft landing retraction is shown in Fig. 4.9. Decreased descent speed resulting from soft landing retraction is very evident, with the payload

![Graph showing drag force over time](image)

**Figure 4.8** Drag force on a cluster of two G–12 parachutes during and after soft landing retraction.

descent rate decreasing from 31.0 ft/s to 12.0 ft/s. A sequence of snapshots of the
two G–12 parachutes during and after the soft landing, colored with the corresponding differential pressures on the canopy is shown in Fig. 4.10. The first two snapshots (left and middle) correspond to times at the start of retraction and at the end of retraction. The final snapshot (right) corresponds to a time well after retraction has finished, with the canopies showing more severe parachute deformations. At the end, before the simulations is stopped, few gores collapsed resulting in gore-to-gore contact. Further FSI simulations are not possible without a contact model.

4.3 Cluster of Three G-12 Parachutes

Varying numbers of parachutes have been used in cluster configurations by the Army to test the efficiency of each configuration. The most commonly used configurations are clusters of two and three parachutes. However, up to twelve parachutes have been successfully tested. Two snapshots from an airdrop test of six G–12 parachutes are shown in the Fig. 4.11. The figure on the right is captured 4s after the left one. It is evident from these two snapshots that there are strong dynamic interactions among the parachutes.

A cluster of three parachutes is one of the simplest clusters of parachute configurations. One benefit of this configuration is that this configuration does not have the natural tendency of going in conical motion as observed for a two-parachute cluster [5].

A snapshot from the landing of the Apollo 15 command module is shown in Fig. 4.12. This module used a cluster of three parachutes to land. The Apollo 15 Command Module "Kitty Hawk", with Astronauts David R. Scott, Alfred M. Worden and James B. Irwin aboard, safely managed to land in the mid-Pacific Ocean to end their lunar landing mission on August 7, 1971 (see Fig. 4.12). Although causing no
Figure 4.9  Payload velocity during descent with a cluster of two G–12 parachutes during and after soft landing retraction.

Figure 4.10  G–12 parachute cluster at the start of pneumatic muscle contraction (left), at the end of contraction (middle) and after soft landing retraction showing the deformed shapes and canopy pressures.
Figure 4.11  Cluster of six G–12 parachute: Images captured from a video of an airdrop test (courtesy of the US Army).

Figure 4.12  Apollo 15 Command Module Kitty Hawk Splashdown Parachute (courtesy of the NASA JSC).
harm to the crewmen, one of the three main parachutes failed to function properly. Little is known about the cause of failure, fluid–structure interaction behavior of this cluster of parachute system, what went wrong, and what can be done to improve the design.

Stein et al. [45, 2, 44] presented the results from the stand-alone fluid dynamics simulations of a cluster of three rigid parachutes. In Section 4.2, I discussed the FSI results of a cluster of two parachutes. Next, I am going to present the results from the FSI simulations of a cluster of three parachutes. The three parachute case presented more challenges in starting the FSI simulations than the two parachute case. Pressure ramping along with smaller time step was used to overcome the start up problems. The FSI simulations for a cluster of more than three parachutes are beyond the scope of this thesis.

4.3.1 Problem definition:

A fluid mesh is generated by joining three pieces (domain box, refinement box and parachutes) as shown in Fig. 4.13 together. Several preparation stages similar to the one discussed for a cluster of the two parachute case in Section 4.2.3 are required. Geometry modeler software package called GAMBIT [48], was used to model the computational domain and parachute geometry and to generate the surface meshes. The volume mesh is generated by an unstructured automatic mesh generator [41, 49, 50]. A refinement boundary is used to get a satisfactory level of mesh refinements near the canopy’s boundary to capture the flow physics accurately. The automatic mesh generator creates about 6 million tetrahedral elements and 1 million nodes of fluid mesh, resulting in approximately 8 million unknowns each nonlinear iteration
Figure 4.13  Fluid mesh of cluster of three G-12 parachutes. Domain box + refinement box + parachute canopies are assembled together to generate volume mesh.
at each time step with DSD/SST formulations. The size of the structure mesh is negligible compared to the fluid mesh. The Reynolds number, based on radius \(D/2\) of a flat G–12 parachute, is 5.5 million. I used the parallel FSI solver on 128 Cray T3E processors to arrive at the results. Computed results, shown in Fig. 4.14, are carried out with fixed payloads and uniform “inflow” boundary conditions. The freestream velocity is assumed to be 20 ft/s which corresponds to a terminal velocity of 20 ft/s.

4.3.2 Results

The computed results from the FSI simulations are shown in Fig. 4.14. From Fig. 4.14(left), one observes that the pressure distributions on each parachute canopy looks similar. This symmetric distribution is lost in Fig. 4.14(center) as a result of the dynamic behavior of parachutes in a cluster configuration. The changes in pressure distribution on the canopies surfaces result in stronger parachute-to-parachute aerodynamic interactions. As a results, the symmetric configuration of parachute arrangements in a cluster configuration is lost.

From Fig. 4.14(center), the parachutes start losing their symmetry. It is very evident from Fig. 4.14(right). In fact, two of the parachutes came very close to each other as observed in (Fig. 4.14 (right)). The closer they got, the higher the frequency of remeshes were and the more difficult it became to perform FSI simulations. At the onset of contact, the simulations have to be terminated in the absence of a contact model. A contact model is discussed in next chapter and is successfully applied to deal with these dynamic contact issues.
Figure 4.14  Cluster of three G–12 parachutes at three different instances of time. Canopies are colored with pressure.

4.4  Conclusion

In this chapter, some computational challenges in FSI simulations were reviewed and results from the FSI simulations of a cluster of two G–12 parachutes under three conditions – fixed payload, airdrop, and soft-landing, were presented. The G–12 is a very large cargo parachute and its analysis presents convergence difficulties at the start of FSI simulations. It was found that a pressure ramping technique to smoothly transfer the information between the fluid and the structure solves the convergence problem encountered at the start of FSI simulations. It was shown that one can ideally achieve an impact velocity as small as 12 ft/s using a soft-landing technique. The results from the FSI simulations of a cluster of three G–12 parachutes were also presented. The FSI simulations of both clusters of two and three parachute systems were found to suffer from contact issues.

In the next chapter, I will review some contact models and present a contact model based on pressure penalty formulations.
Chapter 5
Contact Algorithm

In this chapter, I will present a way to resolve the dynamic contact issues encountered in fluid structure interaction simulations. I will start with a brief introduction of contact problems in Section 5.1. In Section 5.2, I will present a few real life examples where one needs a contact model to continue the fluid-structure interaction (FSI) simulations. In Section 5.3, I will discuss a contact model that was developed to deal with the dynamic contact issues. In Section 5.4, the contact model will be successfully demonstrated for a 2D test problem. Finally, I will present the results for the FSI simulations for a real life problem.

5.1 Introduction

Contact phenomena are commonly observed in the operation of parachute systems. The effects of contact on parachute performance is poorly understood and difficult to study experimentally. Contact can occur in a variety of situations including parachute inflation, a parachute cluster, the collapse of a gore, and a parachute under foreign object impact. My goal is to mainly concentrate on two types of contact – parachute-to-parachute and gore-to-gore contact. I will address these challenges with innovative numerical methods which are suitable for a broad range of parachute applications.

Finite element analysis of the contact problem has long been a difficult subject in computational mechanics. Many researchers have developed suitable contact algorithms and have applied them to a variety of problems. Laursen and Simo [15] have developed a contact stiffness matrix for geometrically nonlinear problems. Using this contact stiffness matrix, Xu et al. [16,18] developed a parallel contact model
to study the structural dynamics of the parachute-contact problem under different circumstances. These models, however, cannot be applied to fluid structure interaction problems because they allow two contacting surfaces to contact or penetrate.

Contact is very difficult to model in FSI problems. When two surfaces come close to each other, the fluid-mesh between them becomes skewed. A fluid-mesh cannot exist when they are in contact. Neither of these two cases are acceptable for FSI simulations. A contact model devised here will prevent the complete contact and the penetration of two contacting surfaces.

A geometrically nonlinear dynamic contact algorithm has been developed in recent work by Gupta et al. [17] and has proven effective for gore-to-gore and parachute-to-parachute contact in stand alone parachute structural dynamics computations. While these algorithms permit simulations of parachute FSI problems with contact, new mesh moving challenges will be encountered in these problems as gore-to-gore or canopy-canopy contact is encountered. Recent enhancements to the mesh moving methods are expected to assist in tackling parachute FSI involving contact.

5.2 Why Contact Model?

As discussed in Chapter 4 and also presented by Stein et al. in [51], dynamic contact phenomena can be experienced in FSI simulations of parachutes. Contact issues present an important challenge and are current barriers in FSI modeling efforts of parachutes. Contact can occur between the gores in a single canopy (see Fig. 5.1) or between two canopies as seen in parachute clusters (see Fig. 5.2). In each situation, contact algorithms are necessary to prevent contact or penetration of the surfaces. In Fig. 5.1, the fluid mesh between the two parachutes is thinly squeezed, and hence the
elements become very skewed. The skewness of the element determines the accuracy of the fluid dynamics (FD) solutions. In the case of gore-to-gore contact as shown in Fig. 5.2, the projection algorithm, needed to project the matching information between fluid and structure, fails. In both of these scenarios, the convergence of Newton-Raphson iterations is badly affected and the automatic mesh generator fails to remesh. So, a contact model is needed which will keep the two contacting surfaces sufficiently far from each other, but will allow them to come sufficiently close enough, so that the effect of the model on the overall aerodynamics is acceptable.
Figure 5.1  FSI of cluster of three G-12 parachutes without a contact model. At the end of computations, two parachutes come in contact.

Figure 5.2  FSI of cluster of three G-12 parachutes without a contact model. At the end of computations, two parachutes come in contact.
5.3 Contact Models

Here, I propose a contact model based on a pressure penalty formulation. When the distance between the two contacting surfaces become less than a threshold limit $\delta$, the contact model is activated. A penalty pressure is applied on both contacting objects to pull them apart. The closer the distance is between them, the larger the penalty force is going to be. The penalty pressure is computed from equations (5.1) and (5.2). This could be thought of as an imaginary nonlinear spring placed between two surfaces. The two corners of the spring are attached to each of the contacting surfaces. A schematic diagram of this contact model is shown in Fig. 5.3. As a

![Diagram of contact model](image)

**Figure 5.3** Schematic diagram of the contact model.

generalization to this model in the context of bearing problems, the spring constant is related to the compressibility of the lubricant fluid between the two contacting surfaces. Huebner *et al.* (equation (8.19) in [52]) carried out a theoretical analysis
for lubricant problems by simplifying the Reynolds equations. They found that the contact pressure is inversely proportional to the distance between contact surfaces when the body forces, the rate of change of distance, and the relative velocity are negligible.

The governing equations to compute the penalty pressure $\Delta p_{\text{penalty}}$ are given by

$$\Delta p_{\text{penalty}} = \frac{1}{2}\rho_{\infty}U_{\infty}^2 L^2 \left( \frac{1}{(r-\epsilon)^2} - \frac{1}{(\delta-\epsilon)^2} \right) \text{ if } 0 < \epsilon < r < \delta, \quad (5.1)$$

$$= 0.0 \text{ if } r > \delta, \quad (5.2)$$

where $\kappa$ is the contact constant and $L$ is the characteristic length. The constant $\epsilon$ is the minimum distance between two contacting surfaces. The penalty pressure goes to infinity when $r$, the distance between the two contacting surfaces, is equal to $\epsilon$. This contact model will neither allow the contact nor permit the penetration of two surfaces. However, the primary issue is how to estimate the parameters $\delta$, $\epsilon$, and $\kappa$. The constant $\delta$ determines when the contact model should be activated. In other words, it determines how close two surfaces should be allowed to get. For most practical applications, I propose to use its value equal to three times the boundary layer thickness. Beyond the boundary layer thickness, it becomes difficult to manage the health of boundary layer elements. If there are no boundary layer elements defined near the contacting surfaces, the maximum element length on the surface is used in place of boundary layer thickness. The constant $\kappa$ is the stiffness constant of the spring. It determines the sharpness of the penalty force as shown in Fig.5.4. The smaller its value, the smoother and wider the penalty force.
Figure 5.4  $k$-chart: Variation of penalty pressure with respect to the distance for various $\kappa$.

5.4 Numerical Examples

To demonstrate the contact model, I considered two problems. The first is a simple problem; a cluster of two 2D parachutes. The second is a real-life model of a cluster of three G–12 parachutes as discussed in the previous chapter.

5.4.1 Cluster of two 2D parachutes

I considered a cluster of two 2D parachutes. Though it is a 2D problem, I modeled it using the existing 3D incompressible FSI solver by introducing a layer of one element thickness in the third dimension. The two dimensional mesh consists of about 10,000 triangular elements and 5,000 nodes, resulting in about 40,000 unknowns. Most of the elements are clustered close to the parachute where gradients of fluid properties are high as shown in Fig. 5.5. The structure mesh consists of 100 membrane elements and 44 cable elements. Payload is pinned i.e. it is not allowed to translate in any
direction. The initial angle between the two parachutes is assumed to be $60^\circ$. First, a stand-alone fluid simulation is carried out. The stand-alone fluid results are used as the initial conditions for the FSI simulations. I forced these two parachutes to make contact by pulling the inner cables. The findings are presented in Fig. 5.6. A contact

![Figure 5.5](image)

**Figure 5.5** Fluid mesh (left), Close-up of fluid mesh(right top), and Structure mesh(right bottom).

is observed in Fig. 5.6(middle). The fluid-mesh between the two parachutes gets squeezed, and hence the mesh gets very skewed. The FSI simulations were deadlocked beyond this point in the absence of a contact model. I applied the pressure penalty based contact model and was successfully able to prevent the contact as observed in
Fig. 5.6(bottom).

The value of the contact coefficients is as follows: $\kappa$ is $1 \times 10^{-3}$ and $\epsilon$ is 0.0. This case required remeshing two times to prevent the contact. Further computations were also carried out with different values for $\kappa$ ($1 \times 10^{-4}$, $1 \times 10^{-5}$, and $1 \times 10^{-7}$) at $\epsilon = 3 \times dL$ where $dL$ is the boundary layer element thickness ($=0.01L$). The smaller the $\kappa$, the more the number of remeshes required to prevent the contact and to continue the FSI simulations. I believe that $\kappa = 1 \times 10^{-3}$ is a suitable choice for this problem. In this case, when the gap between the contacting surfaces is about two times the boundary layer element thickness ($0.04L$), the penalty pressure is close to the stagnation pressure. This implies that the contacting surfaces should remain stationary at this point.

5.4.2 Cluster of Three G–12 Parachutes

In Section 4.3, I presented results from the FSI simulations of a cluster of three G–12 parachutes. I found that two of the parachutes came in contact. Here, I will apply the pressure penalty based contact model as discussed earlier and will present results.

The results from the FSI simulations of a cluster of three G–12 parachutes using the pressure penalty based contact model are presented in Fig. 5.7. On the left side is shown a mesh colored with pressure on a cross sectional plane sweeping through the fluid-mesh and passing through the parachutes. The red color indicates maximum pressure, the blue indicates the minimum pressure, and the other colors are interpolated values between max-min pressure accordingly. On the right side, parachutes and a close-up view of the contact-zone are shown.
Figure 5.6 In the beginning of FSI simulations before contact (top), after contact without contact model (middle), and after contact with contact model (bottom). Left: Fluid mesh colored with pressure, Right: Structure mesh.
In Fig. 5.7(top), the parachutes are separated from each other. However, the dynamic nature of the cluster pushes two of the parachutes in contact as observed in the FSI simulations of the cluster in Fig. 5.7(middle). The FSI simulations beyond this point were not possible because the automatic volume mesh generator failed. In order to carry out further FSI simulations, beyond the results shown in Fig. 5.7(middle), a contact model is required which will keep the two contacting parachutes sufficiently far apart during the contacting period.

I used the contact model proposed in Section 5.3 and successfully applied it to this problem. From Fig. 5.7, it is found that the contact of two parachutes was completely prevented and the barriers in the FSI simulations were addressed. The constant $\delta$ is 0.2$L$ where $L$ is the characteristic length and the $\kappa$ is $10^{-4}$. The constant $\epsilon$ is chosen as zero. The characteristic length used in this simulation is $D/2$ where $D$ is the diameter of the G–12 parachute canopy in flat configuration. Parachutes deformed near the contacting area (Fig. 5.7(bottom, right)) as one would expect, without making any physical contacts. I believe that, by choosing a smaller tolerance $\delta$ and small spring constant $\kappa$, this contact model had little influence on the overall aerodynamics of the system and allowed me to continue the FSI simulations without any deadlock. Soon after, the parachutes were separated far enough from each other that the contact model was not playing any role in the FSI simulations.

5.5 Conclusion

In this chapter, a contact model based on pressure penalty formulations was discussed. The capabilities of the contact model was successfully demonstrated by solving a cluster of three G–12 parachutes where a parachute-to-parachute type of contact
Figure 5.7  Mesh plane at $z = 3.3H$ colored with pressure field (left) and parachute geometry (right), at the beginning (top), before they come in contact without contact model (middle), and with contact model (bottom).
was observed.

In the next chapter, I will discuss effects of porosity on the aerodynamic forces of parachutes.
Chapter 6
Effect of Porosity on Fluid Dynamics Forces

Dropping a payload at a target or landing at a specified location using round parachutes has been always been a major challenge for the Army. There are many factors which can affect the decent trajectory of the parachute such as wind, heat, humidity, hostile terrains, and the wake of the airplane. In recent years, much time and effort has gone into developing controlled parachute systems to improve the accuracy of unmanned airdrops. Numerous research efforts have gone into developing a low-cost guidance, navigation and control (GN&C) [53, 9], such as Affordable Guided Airdrop System (AGAS), for fielded cargo air delivery systems. Additionally, parachutes are made of fabric materials which have natural porosity. In this chapter, I will study the effect of porosity on the aerodynamic forces of a parachute. Porosity features can be used for a better GN&C. I will use a porosity model based on Darcy’s law [54] to demonstrate the effect of porosity on a cross parachute and a US Army T-10 personnel parachute. I will start with a brief introduction followed by a porosity model and numerical examples.

6.1 Introduction

Parachutes are made of nylon fabrics and other textile-type materials. The flow of a fluid through textiles has been studied for a number of years. Understanding such flow phenomena along with the corresponding environmental changes (pressure drop, concentration gradient, etc.) becomes important in many fields such as flow through fabrics [55], turbulent flow in porous media [56], and liquid transport through geomembrane. Porosity of a fabric is defined by the ratio of free space to fiber in a
given volume of fabric. A fabric with a porosity of 0.25 has less free space than a fabric with a porosity of 0.75. Historically, porosity and permeability have been interchanged when describing textile fabrics.

6.2 Porosity Model

Many variables affect the flow through such fabrics, and there are many mathematical ways to model porosity. One such approach is Darcy’s law, as described in [55]. Darcy’s law dictates that the velocity through the porous medium is directly proportional to the pressure drop across the boundary. It is given by

\[ V = \frac{B \Delta p}{\mu t}, \]  

(6.1)

where \( \mu \) is the viscosity of the fluid, \( t \) is the thickness of the fabric membrane, \( V \) is the velocity of the fluid, \( \Delta p \) is the pressure drop across the membrane, and \( B \) is the permeability (or Darcy’s) constant.

6.3 Boundary Conditions for FSI Computations with Porous Boundary

I define a porosity boundary condition on the canopy surface. Through this boundary condition, a velocity boundary condition is specified in the porous section of the boundary. In the case of fluid-structure interaction problems, the FSI velocity of parachute \( \mathbf{u}_{FSI} \) is augmented by the velocity contribution from porosity \( \mathbf{u}_p \) as given below

\[ \mathbf{u}_{FSI} \leftarrow \mathbf{u}_{FSI} + \mathbf{u}_p. \]  

(6.2)

The variable \( \mathbf{u}_p \) is normal to the surface and proportional to the pressure drop across the canopy as described in equation (6.3). A nondimensionalized version of equation
(6.1) is used to calculate the velocity through the canopy, \( \mathbf{u}_p \). It is given by

\[
\frac{\mathbf{u}_p}{U_\infty} = \frac{k_p}{\frac{1}{2} \rho_\infty U_\infty^2} \Delta p \hat{n},
\]

\[
= \frac{k_p}{\frac{1}{2} \rho_\infty U_\infty^2} \Delta p (\hat{t}_1 \times \hat{t}_1), \tag{6.3}
\]

where \( k_p \) is the material porosity coefficient, \( \Delta p \) is the pressure drop across the membrane, and \( \hat{t}_1 \) and \( \hat{t}_2 \) are unit vectors tangent to the membrane surface. The vector \( \hat{n} \) is a unit vector normal to the surface. The parameters \( U_\infty \) and \( \rho_\infty \) are freestream velocity and density, respectively. The porosity coefficient is related to other material properties as follows:

\[
k_p = \frac{1}{2} \rho_\infty U_\infty \frac{B}{\mu t}. \tag{6.4}
\]

The differential pressure is calculated from the fluid simulations. Using this pressure, the velocity computed from equation (6.3) is specified as the Dirichlet type boundary conditions on the fixed boundary or is augmented to the FSI velocity on the deforming interface boundary. One can also combine equation (6.3) with the Navier-Stokes equations by treating the velocities on the parachute surface as unknowns and solving the entire equation system. Each of these approaches are summarized in the next section.

### 6.4 Approach

Three slightly different approaches are discussed on how to modify the parachute canopy velocity boundary condition to include porosity effects. The parachute canopy is treated as a membrane structure. First, the most straightforward approach will define a porosity contribution based on the time-lagged pressure distribution. Implementation here is easiest but the approach may result in instabilities due to the
explicit approach for defining the porosity contribution. Second, an implicit approach is taken and the porosity contribution is recalculated at each nonlinear iteration. However, the contributions to the tangent stiffness matrix are not included and thus, convergence could present difficulties. Third and finally, a fully implicit method is another valid approach. Here, the velocity at the canopy surface nodes is treated as an additional unknown (rather than a Dirichlet boundary condition) which is coupled to the pressure field. The studies of a fully implicit approach has not been studied in this thesis.

6.4.1 Explicit Approach

I compute the porosity contributions from the time-lagged pressure field. The normal at surface nodes is computed based on a weighted average of the element face normals for the faces in which the node lies. A sample logic to program this is given as follows:

\begin{verbatim}
    do i = 1, nit        ! Beginning of nonlinear iteration
        ...
        Calculate tangent vectors t1 and t2, and normal vector n
        Calculate pressure drop and element level normal velocity
        Compute nodal $V_n$ from element level velocity and apply BC
        ...
    enddo                ! End of nonlinear iteration
\end{verbatim}

6.4.2 Implicit Approach

In the implicit approach, porosity contributions are computed in the same way as in the explicit approach, except they are calculated from the current pressure field
and are recalculated at each nonlinear iteration.

6.4.3 Fully Implicit Approach

In the fully-implicit approach, the FSI solver is modified to treat the velocity on the boundary surface as an unknown. The additional equations can be written as

\[ \int_{\Gamma_{\text{canopy}}} \mathbf{w} \cdot \mathbf{u} \, d\Gamma = \int_{\Gamma_{\text{canopy}}} \mathbf{w} \cdot (\mathbf{u}_{\text{FSI}} - k_p \Delta p \hat{n}) \, d\Gamma, \]  

(6.5)

where \( \mathbf{w} \) is the weighting function corresponding to the velocity \( \mathbf{u} \) on the canopy surface.

6.5 Test Problem: Flow Past Cross Parachute

6.5.1 Problem Definition

I consider a simple cross-parachute problem [40] to understand and demonstrate my model. The fluid mesh is composed of about 120,000 tetrahedral elements and 20,000 nodes in fluid mesh, and the structure mesh contains 60 9-node (bi-quadratic) elements with 140 nodes. The domain box for the volume mesh, a section of the volume mesh, and the structure mesh for FSI computations are shown in Fig. 6.1.

Half of the parachute is considered to have porosity \( k_{p1} \) and the other half is assigned porosity \( k_{p2} \) as shown in Fig. 6.2.

6.5.2 Implicit-Explicit Convergence

Fig. 6.3 presents the convergence of nonlinear iterations. Flow was computed for eight nonlinear Newton-Raphson iterations. The explicit approach gives two orders of convergence in eight nonlinear iterations for both low (\( k_p = 0.05 \)) and high (\( k_p = 0.10 \)) porosity cases whereas the implicit approach gives poor convergence (\( k_p = 0.05 \))
Figure 6.1  Domain box for cross-parachute volume mesh(left), a section of volume mesh(center), structure mesh(right).

Figure 6.2  Definition of porous boundary.
case) or no-convergence ($k_p = 0.10$ case). I believe that this is due to the increase in nonlinear behavior with the implicit approach. The implicit approach will give a better convergence if the tangent stiffness matrix is also accounted for.

![Graph showing convergence comparison](image)

**Figure 6.3** Convergence comparison for implicit and explicit algorithm with $k_{p1} = 0.0$ and $k_{p2} = k_p$.

### 6.5.3 Results

Porous media permit the passage of fluid through the medium. Here “the fluid” is air and “the medium” is the membrane-type parachute structure. As a result of porosity, there is leak in the differential pressure across the membrane as seen in Fig. 6.4. The higher the porosity coefficient, the larger the magnitude of velocity through the porous media, as expected (see Fig. 6.4[a-c]). Consequently, it gives rise to a gliding force towards the $+x$-direction as observed in Fig. 6.5. This force could be used to steer the parachute left (Cases $k_p = 0.05^+/0.10^+$) or right (Cases $k_p = 0.05^-/0.10^-$). A larger porosity coefficient gives more drift as observed in the Fig. 6.8. Two cases ($k_p = 0.05^-/0.05^+$) were computed for a longer duration (as shown in Fig. 6.8) to further predict the behavior of the parachute. The results are
consistent i.e. in the case where $k_{kp} = 0.05^-$, the parachute is carried more towards the right ($+x$-direction) and in the case where $k_{kp} = 0.05^+$, the parachute is carried more towards the left ($-x$-direction) with respect to a “no porosity” case.
Figure 6.4 Parachute colored with pressure (left) and velocity magnitude (right) for various values of porosity.
Figure 6.5  Force coefficient along x-direction.

Figure 6.6  Force coefficient along z-direction.

Figure 6.7  The $+x$-position of payload v/s time.
Figure 6.8  The $+x$-position of payload v/s time. Cases ($k_{kp} = 0.05^-/0.05^+$) were computed for longer duration.

6.6 Simulations of a T–10 Parachute with Porous Gores

In this section, I will present the results from the FSI simulations of a fully inflated round parachute when a few gores are made porous. I will consider the T–10 parachute system (see Fig. 6.9). The T–10 is a standard US Army personnel parachute [1, 40]. Fig. 6.9 presents a snapshot from a real life paratrooper jump. The objectives are to study the lateral forces generated as a result of making the parachute partially porous.

6.7 Problem Definition

This parachute system is composed of a 35-foot diameter canopy and 30 suspension lines which are each 29.4 ft long. The canopy is called a “flat extended skirt canopy” because in its constructed (unstressed) configuration it is composed of a main circular section with a circular vent at the apex and an inverted flat ring section, which lies under the main section and is connected to the main section at the outer radius. A schematic diagram of T-10 gore-construction is given in Fig. 6.10. The lines are connected to the payload (a paratrooper in this case) with four risers. The suspension
Figure 6.9  T-10 personnel parachute (courtesy of the US Army).

Figure 6.10  T-10 parachute: Gore and constructed configuration.
lines continue as 30 gore-to-gore reinforcements through the parachute canopy and meet at the apex. The vent diameter and the width of the skirt are 10% of the canopy diameter. The material properties of T–10 parachute system is given table 6.1. The

<table>
<thead>
<tr>
<th></th>
<th>Membranes</th>
<th>Suspension lines</th>
<th>Riser cables</th>
<th>Payload trusses</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Young's modulus</strong> (lb/ft$^2$)</td>
<td>$2.0 \times 10^5$</td>
<td>$4.32 \times 10^6$</td>
<td>$2.16 \times 10^7$</td>
<td>$4.65 \times 10^9$</td>
</tr>
<tr>
<td><strong>Density</strong> (slugs/ft$^3$)</td>
<td>6</td>
<td>6</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td><strong>Thickness (ft)</strong></td>
<td>$10^{-4}$</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td><strong>Cross sectional area (ft$^2$)</strong></td>
<td>-</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
<td>$10^{-4}$</td>
</tr>
</tbody>
</table>

**Table 6.1** T–10 parachute system: material properties.

payload is modelled by four point masses using 6-truss elements. I used the same cables for suspension lines and radial reinforcements.

 Constructed geometries of a T–10 parachute in flat (unstressed) and inflated (pre-stressed) configurations are shown in Fig. 6.11. The 3D mesh with tetrahedral elements (with about 140,000 nodes and 860,000 elements) was generated using, as an interior boundary, the inflated canopy (Fig. 6.11(right)) from the stand-alone structural dynamics simulation. Initial unsteady flow solutions were obtained for the fixed canopy configuration at Reynolds number $5 \times 10^6$ using semi-discrete stabilized formulations. To obtain the flow solutions, the canopy is initially treated as a zero-porosity material and is assigned a no-slip boundary condition. The inflow boundary
below the parachute is assigned a prescribed velocity condition which is equivalent to
the desired Reynolds number. Side boundaries are assigned slip conditions, and the
outflow boundary above the parachute is assigned a traction-free condition. Using
the results from semi-discrete formulations when the flow was fully developed, the
DSD/SST (more cost-intensive than semi-discrete) procedure is used to obtain the
starting fluid dynamics conditions for the simulation of a canopy with porous gores.
The flow solutions of the canopy with non-porous gores are used as the initial con-
ditions for the flow simulations of the canopy with porous gores. The flow solutions
from the DSD/SST simulations of the canopy with porous gores are used as start up
conditions for fluid structure simulations of the parachute.

6.7.1 Definition of the Porous Section

To demonstrate the capabilities of this model, I considered three cases. In these
cases, three, five and seven gores are made porous as shown in Fig. 6.12.

6.7.2 Results

The simulated results from the FSI simulations for varying numbers of porous
gores with porosity constant $K_p = 0.10$ are shown in Fig. 6.13. First, I obtained the
results from the FSI simulations of the parachute with the lower porosity set at $K_p$
= 0.01, and then a porosity value of $K_p = 0.05$. Flows results from lower porosity
cases were used as initial conditions for higher porosity cases. This step was necessary
to get good convergence of nonlinear iterations for higher porosity cases. Doing so,
one reduces the mismatch in the initial conditions and, hence, arrives at a better
convergence of the nonlinear systems.

Since the lower-right section of the parachute (as shown in Fig. 6.12) is made
Figure 6.11  T-10 parachute before inflation (left) and after inflation (right).

Figure 6.12  T-10 parachute geometry with 3 porous gores (left), 5 porous gores (center) and 7 porous gores(right).
porous, I expect that more fluid will pass through this section of the canopy. So, the magnitude of the pressure drop across the canopy in these areas will be lowered. The loss in the differential pressure can be thought of as result of a leak in the pressure in the porous area of the canopy. The leak in the pressure leads to a negative force acting on the parachute normal to the canopy. By breaking this force component in three directions, I expected an increase in the magnitude of the force in the $-x$-direction and in the $+y$-direction. This is reflected and confirmed in Figs. 6.13. In Fig. 6.13(upper), I notice that the payload with porous gores drifts more towards the $-x$-direction than with the non-porous case. The larger the number of porous gores, the larger the drift. I made a similar observation for the $+y$-direction in Fig. 6.13(lower). As more gores are made porous, the parachute drifts more towards the $+y$-direction. Here, the difference in the extent of drift is more visible due to a larger impact of porosity in the $+y$-direction.

6.8 Conclusion

In this chapter, a porosity model, based on Darcy's law, was implemented in the FSI solver. It was observed that porosity could be used to generate additional sidewise aerodynamic forces. These force can be used for a better guidance, navigation, and control.

In the next chapter, I will discuss the compressible fluid–structure interaction. I will propose an efficient shock capturing algorithm which will be followed by a methodology on how to solve compressible fluid–structure interaction problems.
Figure 6.13  Horizontal position of payload versus time $t$ for different number of porous gores at porosity constant $K_p=0.10$. Upper: Position $x$ v/s time $t$. Lower: Position $y$ v/s time $t$. Both position and time are nondimensionalized by $D/2$ and $D/2U$, respectively.
Chapter 7
Compressible Fluid Structure/Object Interaction and Challenges

Compressible flows occur frequently in engineering applications. Common examples include compressed air systems used to power shop tools and dental drills, transmission of gases in pipelines at high pressure, and pneumatic or fluidic control and sensing systems. Compressibility effects are very important in the design of modern

Figure 7.1 Artistic painting showing parachute deployment sequence of the Galileo probe. With the deployment of the drogue parachute, the probe velocity decreased from 1,000mph to 430mph (courtesy of NASA Ames).

high-speed aircraft and missiles, power plants, fans, and compressors. Many airdrops also occur in a compressible flow regime. Space probes like the one shown in Fig. 7.1 for the Galileo probe of Jupiter fall into this category. Another example is the drogue parachute deployment during the airdrop stage of the proposed X-38 crew return
vehicle for the international space station. Here, my goal is to study the physics of compressible airdrop problems. Numerical simulations of these airdrop technologies require development of a compressible FSI solver.

In compressible flows, the density variations within a flow are not negligible. Therefore, the energy equation in the Navier-Stokes equations of a Newtonian fluid is no longer decoupled from momentum and continuity equations. As a result, all five (one continuity, three momentum, and one energy) equations in three dimensional problems need to be solved simultaneously. The increased complexity, owing to additional equations compared to the incompressible flows, introduces additional difficulties in performing accurate and efficient numerical simulations.

In this chapter, I will describe some of the challenges involved in numerical modeling of compressible flows. Then, I will present an efficient method to capture shocks in Section 7.2 followed by compressible fluid structure/object interaction formulations and examples in Section 7.3.

7.1 Challenges in Modeling Compressible Flows

Accurate numerical treatment of compressible fluid flow problems faces many computational challenges. A brief description of some of them are given below.

7.1.1 Capturing Shocks in Supersonic Flows

In supersonic flows past a bluff body, a discontinuity in the solution is observed. This discontinuity is popularly known as “shock”. Standard numerical methods cannot accurately handle any kind of discontinuity in the flow. So, a modeling technique is needed to appropriately account for the discontinuities in the flow field. In the next section, I propose an efficient modeling technique that introduces an optimum degree
of numerical diffusion to capture the shocks accurately while maintaining acceptable convergence of the nonlinear iteration procedure.

7.1.2 Fluid-Structure Interaction

Fluid-structure interaction problems are encountered when the structure either deforms or is set into motion as a result of aerodynamic forces from the fluid. In each case, the fluid-mesh needs to be updated according to the structure's deformation or motion. Shocks (in supersonic flows) will also interact with the deforming or moving geometries. These interactions make the problem highly nonlinear in nature and lead to difficulties with the convergence of nonlinear iterations.

7.1.3 Turbulence Modeling

High Reynolds number flows are usually turbulent which can often be triggered by the strong shock-object interactions (in supersonic flows). Upon interaction with a shock, an object gets impulsively accelerated resulting in the generation of vorticity due to the misalignment of pressure and density gradients ultimately producing a turbulent flow. Appropriate turbulence modeling techniques or direct numerical simulations are needed to understand the turbulence behavior.

7.1.4 Incompressibility Limit

Commonly used compressible solvers that address the issues related to conservation, entropy production, upwinding, and the changing character of the system of partial differential equations, fail to adequately handle the incompressibility constraint. In practical applications, however, mixed types of flows are often observed where both compressible and incompressible (presumably when Mach number tends
to zero) flows co-exist. A mixed type of formulation, which can deal with both compressible and incompressible flows simultaneously, is needed. See [57] for further details.

7.2 Stabilization and Shock Capturing Parameters

In this section, stabilization parameters for the compressible flows introduced in Section 3.2.3, are discussed. Stabilized formulations such as the Streamline-Upwind/Petrov-Galerkin (SUPG) [58–60] and Pressure-Stabilizing/Petrov-Galerkin (PSPG) [13] formulations are widely used in finite element flow computations. These formulations prevent numerical oscillations and other instabilities in solving problems with high Reynolds and/or Mach numbers. They also allow equal-order interpolation functions for velocity, pressure, and other unknowns. It was shown by Tezduyar et al. [61] that these stabilized formulations also substantially improve the convergence rate in the iterative solution of the large matrix systems that need to be solved at every Newton-Raphson iteration.

The SUPG formulation for compressible flows was first introduced, in the context of conservative variables, by Tezduyar et al. in [59,60]. Subsequently, several SUPG-like methods for compressible flows were also developed. The Taylor-Galerkin method (Donea [62]), for example, is very similar and, under certain conditions, is identical to one of the SUPG-like methods [59,60]. Another example of a SUPG-like method for compressible flows in the conservative variables is the streamline-diffusion method described by Johnson et al. [63]. Later, the SUPG formulation for compressible flows was recast in entropy variables and supplemented with a shock-capturing term by Hughes et al. [64]. It was shown by Le Beau and Tezduyar [65] that the SUPG
formulation introduced by Tezduyar et al. [59,60], when supplemented with a similar
shock-capturing term, is comparable in accuracy to the one that was recast in entropy
variables. Later, 2D test computations for inviscid flows reported by Le Beau et al. [66]
showed that the SUPG formulation in the conservative and the entropy variables yield
indistinguishable results.

The stabilization parameters used in the SUPG formulation for compressible flows
were first introduced by Tezduyar et al. [59,60]. In this section, I will call the SUPG
formulation introduced in [59,60] for compressible flows \((SUPG)_{82}\), and the set of
\(\tau\)'s introduced in conjunction with that formulation \(\tau_{82}\). The stabilized formulation
introduced by Tezduyar et al. in [67] for advection-diffusion equations include a shock-
capturing term and a \(\tau\) definition that takes into account the interaction between the
shock-capturing term and the SUPG term. This \(\tau\) definition, for example, precludes
"compounding" i.e. the augmentation of SUPG effects by the shock-capturing effects
that occur when advection and shock capturing directions coincide. The \(\tau\) used by
Le Beau et al. in [65] with \((SUPG)_{82}\) is an extensively modified version of \(\tau_{82}\). A
shock-capturing parameter, which I will refer to as \(\delta_{91}\), was embedded in the shock-
capturing term used by Le Beau in [65]. Subsequent modifications to \(\tau_{82}\) account for
the interaction between the shock-capturing and the \((SUPG)_{82}\) in a similar fashion
following Tezduyar and Park [67] for the advection-diffusion-reaction equations. All
of these slightly modified versions of \(\tau_{82}\) have always been used with the same \(\delta_{82}\).

7.2.1 SUPG Stabilization and Shock Capturing Parameters Based on the
Entropy

The SUPG matrix \(\mathbf{\tau}\), used by Le Beau et al. [65] and Aliabadi et al. [21], is given
by
\[
\tau = \tau_{\text{entropy}} = \max[0, \tau_t + \zeta(\tau_a - \tau_\delta - \tau_d)],
\]

where \(\tau_a\) is the stabilization matrix corresponding to the advection term, and \(\tau_\delta\) and \(\tau_d\) matrices are corrections. Hence, they are subtracted from \(\tau_a\) to account for the presence of the shock-capturing term and physical diffusion. Without the correction terms, the formulation tends to be overly diffusive [21]. The \(\tau_t\) comes from the time-dependent terms. These matrices are defined as

\[
\tau_a = \frac{h}{2(c + |u \cdot j|)} I,
\]

\[
\tau_t = \frac{2}{3(1 + 2\alpha C_r)} \tau_a,
\]

\[
\tau_\delta = \frac{\delta}{(c + |u \cdot j|)^2} I,
\]

\[
\tau_d = \frac{\sum_{i=1}^{n_{en}} j_i^2 \text{diag}(K_{ii})}{(c + |u \cdot j|)^2}.
\]

The direction \(j\) is defined as

\[
j = \frac{\Delta ||U||_2}{||\Delta ||U||_2||}.
\]

The element length \(h\) is calculated along \(j\) and is given by

\[
h = \frac{2}{\sum_{a=1}^{n_{en}} j \cdot \nabla N_a},
\]

where \(n_{en}\) is the total number of nodes per element and \(N_a\) is the shape (or test) function at a local node “a” for a given element.

The effective courant number \(C_r\), based on element length, \(h\), and the speed of sound, \(c\), is given by
\[ C_r = \frac{(c + |\mathbf{u} \cdot \mathbf{j}|)\Delta t}{h}, \]  

where \( \Delta t \) is the time step size.

The time-marching weighting coefficient, \( \zeta \), is defined as

\[ \zeta = \frac{2\alpha C_r}{1 + 2\alpha C_r}, \]  

where \( \alpha \) is a time-marching coefficient.

The shock-capturing coefficient, \( \delta \) is defined as

\[ \delta = \left[ \frac{\| J_{ik} \frac{\partial \mathbf{U}}{\partial x} \|_{A_0^{-1}} + \| J_{j} \frac{\partial \mathbf{U}}{\partial x} \|_{A_0^{-1}} + \| J_{j} \frac{\partial \mathbf{U}}{\partial x} \|_{A_0^{-1}}}{\| J_{ik} \frac{\partial \mathbf{U}}{\partial x} \|_{A_0^{-1}} + \| J_{j} \frac{\partial \mathbf{U}}{\partial x} \|_{A_0^{-1}} + \| J_{j} \frac{\partial \mathbf{U}}{\partial x} \|_{A_0^{-1}}} \right]^{\frac{1}{2}}, \]  

where \( J_{jk} \) are components of the Jacobian of the transformation from the physical to element reference coordinates (the inverse transformation). This expression attempts to recreate a successful design of the shock-capturing operator in the context of conservative variables [68].

This entropy based discontinuity capturing is discussed in detail in [69]. It uses the inverse of the entropy variables transformation matrix given by

\[ A_0^{-1} = \frac{-1}{\rho i V_5} \begin{pmatrix}
  k_1^2 + \gamma & k_1 V_2 & k_1 V_3 & k_1 V_4 & (k_1 + 1)V_5 \\
  k_1 V_2 & V_2^2 - V_5 & -d_1 & -d_2 & e_1 \\
  k_1 V_3 & -d_1 & V_2^2 - V_5 & -d_3 & e_2 \\
  k_1 V_4 & -d_2 & -d_3 & V_2^2 - V_5 & e_3 \\
  (k_1 + 1)V_5 & e_1 & e_2 & e_3 & V_5^2
\end{pmatrix}, \]  

where the entropy variables, \( V_{(1-5)} \) are defined as
\[
\begin{pmatrix}
V_1 \\
V_2 \\
V_3 \\
V_4 \\
V_5
\end{pmatrix} = \frac{1}{\rho_i} \begin{pmatrix}
\rho e + \rho i (\gamma + 1 - s) \\
\rho u_1 \\
\rho u_2 \\
\rho u_3 \\
-\rho
\end{pmatrix}.
\] (7.12)

The entropy \( s \) is given by

\[
s = \ln \left[ \frac{(\gamma - 1)\rho i}{\rho^\gamma} \right]
\] (7.13)

and the remaining auxiliary variables appearing in right-hand side of equation (7.11) are given by

\[
k_1 = \frac{V_2^2 + V_3^2 + V_4^2}{2V_5},
\] (7.14)
\[
d_1 = -V_2 V_3, \quad d_2 = -V_2 V_4, \quad d_3 = -V_3 V_4,
\] (7.15)
\[
e_1 = V_2 V_5, \quad e_2 = V_3 V_5, \quad e_3 = V_4 V_5.
\] (7.16)

The matrix A-norm of a vector \( \mathbf{x} \) is defined as

\[
\|\mathbf{x}\|_A = [\mathbf{x}^T \mathbf{A} \mathbf{x}]^{\frac{1}{2}}.
\] (7.17)

### 7.2.2 SUPG Stabilization and Shock Capturing Parameters Based on the Gradient of Density

In this thesis, in the context of the \((SUPG)_{s2}\) formulation, I will propose an alternative way of calculating the stabilization and the shock-capturing parameters.
Given the conservation variable vector \( \mathbf{U} = (\rho, \rho \mathbf{u}, \rho e) \), a test vector-function \( \mathbf{W} \) associated with it, and the acoustic speed \( c \), I define a unit direction \( \mathbf{j} \) along the gradient of density as follows

\[
j = \frac{\nabla \rho^h}{||\nabla \rho^h||}.
\]  
(7.18)

In much of the area away from the shock, the \( ||\nabla \rho|| \) is small. The smaller the \( ||\nabla \rho|| \), the more difficult it is to find the direction \( \mathbf{j} \). In this circumstance, the vector \( \mathbf{j} \) is defined along the velocity as below

\[
j = \frac{\mathbf{u}}{||\mathbf{u}||} \quad \text{if} \quad ||\nabla \rho|| < \epsilon,
\]  
(7.19)

where \( \epsilon \) is a very small number. For practical purposes, I chose \( \epsilon = 10^{-8} \).

As a first alternative to computing SUPG terms for each component of the test vector-function \( \mathbf{W} \), I propose to define \( \tau^\rho_{SUPG1} \), \( \tau^u_{SUPG1} \) and \( \tau^e_{SUPG1} \) (associated with \( \rho \), \( \rho \mathbf{u} \) and \( \rho e \), respectively) as

\[
\tau^\rho_{SUPG1} = \tau^u_{SUPG1} = \tau^e_{SUPG1} = \left( \sum_{a=1}^{n_{en}} |\mathbf{u}^h \cdot \nabla N_a| \right)^{-1}
\]  
(7.20)

As a second alternative, I propose to use the following definitions

\[
\tau^\rho_{SUPG1} = \tau^u_{SUPG1} = \tau^e_{SUPG1} = \left( \sum_{a=1}^{n_{en}} c|\mathbf{j} \cdot \nabla N_a| + |\mathbf{u}^h \cdot \nabla N_a| \right)^{-1}
\]  
(7.21)

In computing \( \tau_{SUPG2} \), I propose the following expression

\[
\tau^\rho_{SUPG2} = \tau^u_{SUPG2} = \tau^e_{SUPG2} = \frac{\Delta t}{2}.
\]  
(7.22)

In computing \( \tau_{SUPG3} \), I define \( \tau^u_{SUPG3} \) as
\[
\tau_{SUPG3}^u = \frac{h_{RGN}^2}{4\nu}.
\] (7.23)

I define \(\tau_{SUPG3}^\epsilon\) as

\[
\tau_{SUPG3}^\epsilon = \frac{(h_{RGN}^\epsilon)^2}{4\nu^\epsilon} c,
\] (7.24)

where \(\nu^\epsilon\) is the kinematic viscosity in the energy equation. Here, \(h_{RGN}^\epsilon\) is calculated along the gradient of temperature and is given by

\[
h_{RGN}^\epsilon = 2 \left( \sum_{\alpha=1}^{n_{en}} |r^\epsilon \cdot \nabla N_\alpha| \right)^{-1},
\] (7.25)

\[
r^\epsilon = \frac{\nabla \theta^h}{\|\nabla \theta^h\|},
\] (7.26)

where \(\theta\) is the temperature. I define \((\tau_{SUPG}^\rho)_{UGN}\), \((\tau_{SUPG}^u)_{UGN}\) and \((\tau_{SUPG}^\epsilon)_{UGN}\) by using the “\(r\) – switch” as

\[
(\tau_{SUPG}^\rho)_{UGN} = \left( \frac{1}{(\tau_{SUPG1}^\rho)^r} + \frac{1}{(\tau_{SUPG2}^\rho)^r} \right)^{-\frac{1}{r}},
\] (7.27)

\[
(\tau_{SUPG}^u)_{UGN} = \left( \frac{1}{(\tau_{SUPG1}^u)^r} + \frac{1}{(\tau_{SUPG2}^u)^r} + \frac{1}{(\tau_{SUPG3}^u)^r} \right)^{-\frac{1}{r}},
\] (7.28)

\[
(\tau_{SUPG}^\epsilon)_{UGN} = \left( \frac{1}{(\tau_{SUPG1}^\epsilon)^r} + \frac{1}{(\tau_{SUPG2}^\epsilon)^r} + \frac{1}{(\tau_{SUPG3}^\epsilon)^r} \right)^{-\frac{1}{r}}.
\] (7.29)

In defining the shock-capturing term, I first define the “shock-capturing viscosity” \(\nu_{SHOC}\) as

\[
\nu_{SHOC} = \tau_{SHOC}(u_{int})^2,
\] (7.30)
where

$$
\tau_{SHOC} = \frac{h_{SHOC}}{2u_{cha}} \left( \frac{\|\nabla \rho\| h_{SHOC}}{\rho_{ref}} \right)^\beta, 
$$
(7.31)

$$
h_{SHOC} = h_{jgm},
$$
(7.32)

$$
h_{jgm} = 2 \left( \sum_{a=1}^{n_{en}} |j \cdot \nabla N_a| \right)^{-1}.
$$
(7.33)

Here, \(\rho_{ref}\) is the reference density (such as \(\rho\) at the inflow, or the difference between the estimated maximum and minimum values of \(\rho\)), \(u_{cha}\) is a characteristic velocity (such as \(u_{ref}\) or \(\|u\|\) or acoustic speed, \(c\)), and \(u_{int}\) is an intrinsic velocity (such as \(u_{cha}\) or \(\|u\|\) or acoustic speed). I propose to set \(u_{int} = u_{cha} = u_{ref}\). The constant \(\beta\) is a free parameter.

Then, the shock-capturing term is defined as

$$
S_{SHOC} = \sum_{e=1}^{n_{en}} \int_{\Omega_e} \nabla W^h : (\kappa_{SHOC} \cdot \nabla U^h) d\Omega,
$$
(7.34)

where \(\kappa_{SHOC}\) is defined as

$$
\kappa_{SHOC} = \nu_{SHOC} I,
$$
(7.35)

where \(I\) is the identity matrix.

As a possible alternative, I propose

$$
\kappa_{SHOC} = \nu_{SHOC} j j.
$$
(7.36)

In an attempt to preclude “compounding” i.e. augmentation of the SUPG effect by the shock-capturing effect when the advection and shock directions coincide, I propose to modify \(\nu_{SHOC}\) as follows,
\[ \nu_{SHOC} \leftarrow \nu_{SHOC} - \text{switch}(\tau_{SUPG}(j \cdot u)^2, \tau_{SUPG}(|bj \cdot u| - c)^2, \nu_{SHOC}) \]  \hspace{1cm} (7.37)

where the “switch” function is defined as a “min” function where the function “min” stands for “minimum”. For viscous flows, the above modification would be made separately for \( \tau_{SUPG}^\rho \), \( \tau_{SUPG}^\mu \), and \( \tau_{SUPG}^\varepsilon \), and thereby, resulting in \( \nu_{SHOC} \) becoming a diagonal matrix.

### 7.2.3 Numerical Examples

I will consider two examples to demonstrate my technique. For both of these two problems, exact analytical solutions are available.

#### Shock Reflection

This is a two dimensional inviscid steady problem. It involves three regions separated by an oblique shock and its reflection from a wall as shown in Fig. 7.2. This is a standard benchmark problem. For details, readers are referred to the work by Le Beau and Tezduyar [65] and Shakib [28]. The computational domain is a rectangular region of dimensions 4.1 along the \( x \) direction and 1.0 along the \( y \) direction. The mesh consists of 100 \( \times \) 40 uniform rectangular elements. On the left and the top boundaries, flow data corresponding to the exact solution are prescribed as listed in Table 7.1. At the lower boundary the velocity, normal to the wall, is assigned to zero. The initial conditions for the flow data equal to the flow data in region 1. The \( \rho_{ref} \) and the \( u_{ref} \) are 2.9 and 1.0, respectively. Although it is a steady state problem, I use a time dependent flow solver to study this steady state problem because it helps to get converged solutions.
Figure 7.2  Shock-reflection problem: problem description.

<table>
<thead>
<tr>
<th></th>
<th>Region 1</th>
<th>Region 2</th>
<th>Region 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>2.9</td>
<td>2.378</td>
<td>1.942</td>
</tr>
<tr>
<td>$\rho$</td>
<td>1</td>
<td>1.7</td>
<td>2.687</td>
</tr>
<tr>
<td>$u_1$</td>
<td>2.9</td>
<td>2.619</td>
<td>2.401</td>
</tr>
<tr>
<td>$u_2$</td>
<td>0</td>
<td>-0.506</td>
<td>0</td>
</tr>
<tr>
<td>$p$</td>
<td>0.714</td>
<td>1.528</td>
<td>2.934</td>
</tr>
</tbody>
</table>

Table 7.1  Shock-reflection problem: exact solution.
In Fig. 7.3(top), density, for various $\beta$ at $y=0.5$ line, is compared with the analytical solution. The $\beta$ is defined in equation (7.33). The SUPG stabilization terms are given in equation (7.20). Here, I notice that the predicted shock becomes sharper as $\beta$ increases from 0.5 to 4.0. At $\beta = 0.5$, the solution is more diffusive and hence, the shock is smeared out near the shock. At $\beta = 4.0$, the formulation does not provide enough stabilization to suppress the numerical oscillations near the shock. I propose to use $\beta = 1.0$ which, I believe, provides the right amount of stabilization.

In Fig. 7.3(middle) I analyze the effect of parameter $r$, as defined in equations (7.27)-(7.29). For this case, the parameter $\beta$ is 1.0. I observe that the $r = 0.5$ case is more oscillatory than cases with a larger $r$. Also, the magnitude of overshoots and undershoots near the shock is larger. However, the impact of the parameter $r$ is not as much as the impact of parameter $\beta$. I believe that this is because of the dominance of the shock capturing terms over the SUPG stabilization terms in the area where shock exists. I propose to use $r = 2.0$.

In Fig. 7.3(bottom), I compare the results of $\tau_{SUPG_1}$ (equation (7.20)), $\tau_{SUPG ugn}$ (equation (7.27)-(7.29)), and $\nu_{shock-switch}$ (equation (7.37)) cases with the $\tau_{entropy}$ (equation (7.1)) case. The $\tau_{entropy}$ case does not capture the shock as well as the other cases. The $\tau_{entropy}$ case seems to over-stabilize the system. Meanwhile, the other cases appear to be more successful in capturing the shock. The cases $\tau_{SUPG ugn}$ and $\nu_{shock-switch}$ give indistinguishable results because the advection and the shock directions are not coinciding. I believe that the $\nu_{shock-switch}$ will predict a sharper shock than the $\tau_{SUPG ugn}$ case when the advection and the shock wave are in the same direction.
Figure 7.3  Shock-reflection problem: density v/s x at y = 0.5 for various $\beta$ (top), for various $\tau$ (middle), and for various $\tau$ definitions (bottom) in comparison with analytical solution.
Oblique Shock

This is a two dimensional, inviscid, steady state problem. In this, a uniform flow at mach 2.0 is obstructed by a sharply turned wall at an angle $10^\circ$ as shown in Fig. 7.4. The exact solutions are given in Table 7.2. For details, readers are referred to Hauke [57]. The computational domain $(0 \leq x \leq 1, \ 0 \leq y \leq 1)$ is discretized by $20 \times 20$ rectangular elements. This problem shows trends similar to the shock-reflection problem discussed in the previous section. The smaller the $\beta$, the larger the numerical dissipation. The larger the numerical dissipation, the more smeared out the solutions are going be as observed in Fig. 7.5(top). Density gradient based stabilization cases appear to predict a sharper shock than the entropy case as observed in Fig. 7.5(bottom). The effect of the parameter $r$ is shown in Fig. 7.5(middle).

Figure 7.4 Oblique-shock problem: problem description
<table>
<thead>
<tr>
<th>Inflow</th>
<th>Outflow</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>2</td>
</tr>
<tr>
<td>$\rho$</td>
<td>1</td>
</tr>
<tr>
<td>$u_1$</td>
<td>$\cos(10^\circ)$</td>
</tr>
<tr>
<td>$u_2$</td>
<td>$-\sin(10^\circ)$</td>
</tr>
<tr>
<td>$p$</td>
<td>0.17857</td>
</tr>
</tbody>
</table>

Table 7.2 Oblique-shock problem: analytical solution.
Figure 7.5  Oblique-shock problem: density v/s x at y = 0.2 for various β (top), for various τ (middle), and for various τ definitions (bottom) in comparison with analytical solution.
7.3 Compressible Fluid Structure/Object Interactions

Fluid structure/object interactions occur in various engineering problems and pose several numerical difficulties, especially in problems where the interaction between fluid and structure (or object) is strong such as in FSI simulations of parachutes and Vortex Induced Vibrations (VIV) of flexible cylinders. Various researchers around the globe have studied such problems in the incompressible flow regime. For example, flow induced vibration of circular cylinders are well documented in the works of S. Mittal and V. Kumar [3, 70–72]. They observed that “lock-in” phenomena occur in the oscillation of circular cylinders [71, 72] for a given natural harmonics and Reynolds number, and suggested ways to counter the “lock-in”. In many engineering problems such as off-shore oil platforms, the “lock-in” (similar to resonance) can be catastrophic. The basic capabilities of the method for parachute problems in incompressible flow regime is presented by Karlo et al. in [73] and Stein et al. in [14].

Fluid-object interaction for ballistics in compressible flow regime, with automatic mesh updating techniques [49], is presented by Ray et al. in [74]. Kroyer [75] carried out the FSI simulations of a 2D-fin profile in a supersonic flow regime where aerelastic instability effects may occur. However, little is known about fluid structure interaction phenomena of parachute structures in compressible flow regimes while there exist several practical applications where it is crucial to understand such FSI processes.

In this section, I will present a methodology to solve fluid-structure and fluid-object interaction problems in a compressible flow regime. First, I will discuss the coupling mechanism associated with the fluid and the object. Then, I will present formulations for fluid-object and fluid-structure interactions which will be followed
by numerical examples.

7.3.1 Coupling Mechanism

Fluid and structure/object are coupled through the exchange of force and displacement information at the interface surface as shown in Fig. 7.6. The coupling is achieved in an iterative fashion where transfer of force, displacement, and velocity information between the fluid and structure/object takes place within the Newton-Raphson nonlinear iteration loop. Multiple iterations improve the convergence of the coupled system. I used an implicit time-integration method for both the dynamics of the fluid as well as the structure. Displacement from structural dynamics is used as Dirichlet-type boundary conditions for the mesh moving scheme. Displacement rate or velocity is treated as the boundary condition for flow. For the fluid dynamics, it is ensured that the fluid velocity normal to the structure surface equals the mesh velocity of the structure boundary. In return, fluid forces on the interface are used as distributed forces for structural dynamics.

Information between structure and fluid can be transferred using compatible or incompatible meshes at the interface. In the case of a compatible mesh, nodes in the structure mesh exactly match fluid nodes. In the case of an incompatible mesh, the two meshes do not exactly match each other. Coupling information in incompatible meshes is transferred using a least-square projection technique. Incompatible meshes have many advantages over compatible meshes. They provide an opportunity to choose different orders of elements, different refinements etc. For example, one can use a 9-noded bi-quadratic elements to model the structural dynamics and a 3-noded surface elements to model the fluid dynamics. Structural dynamics usually behave
better with higher order elements. However, the disadvantages of using incompatible meshes are that they increase the cost of computations and introduce projection errors.

![Diagram](image)

**Figure 7.6** Coupling mechanism for compressible fluid–structure interaction problems.

Using the fluid forces, I compute the displacement and velocity of the structure. For the structure, I use the structural dynamics equations as described in Section 3.3 to compute displacement and velocity of a deformable object. In the case of a rigid body object, I use the force balance equations given in Section 7.3.2 to compute the displacement. Computed displacement is used to move the mesh. All of these steps are repeated over nonlinear iterations.

Using the ideal gas equation, the pressure, using the conservative (primitive) variables which are computed by solving fluid dynamics equations, is computed as follows
\[ p = \rho R \theta, \tag{7.38} \]

\[ = (\gamma - 1) \rho i, \tag{7.39} \]

\[ = (\gamma - 1) \left[ \rho e - \frac{\|\rho u\|^2}{2\rho} \right], \tag{7.40} \]

where \( R \) is the ideal gas constant, \( i \) is the internal energy, \( \theta \) is the temperature, and \( \gamma \) is the ratio of specific heat (usually \( = 1.4 \) for mono atomic gases).

The boundary conditions for fluid velocity on the structure interface boundaries are given by

\[
\begin{bmatrix}
U_2 \\
U_3 \\
U_4
\end{bmatrix}
= 
\begin{bmatrix}
\rho u_{F1} \\
\rho u_{F2} \\
\rho u_{F3}
\end{bmatrix},
\tag{7.41}
\]

where \( u_{Fi}, i = 1:3 \), are the velocity of the fluid-structure interface boundary.

In the case of fluid-object interactions, the coupling mechanism for velocity and displacement is very straightforward. Here, velocity and displacement computed from object dynamics (as discussed in 7.3.2) are directly applied to the fluid-object interface boundary in the fluid dynamics.

Additional coupling issues have been discussed in Section 4.1 and in Stein [40].

### 7.3.2 Fluid Object Interaction

In this section, I will discuss the governing equations for object dynamics. Objects can go through translation, rotation or both. The fluid mesh follows the object and deforms accordingly. The mesh is treated as a pseudo-solid elastic material and has no physical significance. Therefore, a smart meshing scheme which reduces the number
of remeshing would be better suited for such problems. I used an element based stiffening mesh moving scheme discussed by Stein in [40].

**Equilibrium Equations**

Let us consider a body of mass $m$ and moment of inertia $J$ with respect to the center of mass at the "CM". This body is acted upon by a force $F$ and moment $M$ about the axis of rotation at the origin $O$ as shown in Fig. 7.7.

![Figure 7.7](image)

**Figure 7.7** Rotation of body.

The fluid dynamic force on the body is computed as

$$F = \int \sigma \cdot n \, dA,$$

(7.42)

where $n$ is the normal unit vector of differential area element $dA$ and $\sigma$ is the stress
tensor which includes shear stress and pressure terms. The shear stress $\sigma$ is given by

$$\sigma = -pI + \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T),$$  

(7.43)

where $I$ is the identity matrix and $\mathbf{u}$ is the velocity of fluid. The moment about the origin $O$ is given by

$$\mathbf{M} = \int \mathbf{r} \times (\sigma \cdot \mathbf{n})dA,$$

(7.44)

where $\mathbf{r}$ is position vector of a differential area element $dA$.

Using the moment and force balance equations, the angular acceleration $\dot{\omega}$ and translational acceleration $\dot{v}$ are given by

$$
\begin{bmatrix}
\dot{v}_1 \\
\dot{v}_2 \\
\dot{v}_3 \\
\dot{\omega}_1 \\
\dot{\omega}_2 \\
\dot{\omega}_3
\end{bmatrix} =
\begin{bmatrix}
m & 0 & 0 & 0 & 0 & 0 \\
0 & m & 0 & 0 & 0 & 0 \\
0 & 0 & m & 0 & 0 & 0 \\
0 & 0 & 0 & J_{11} & J_{12} & J_{13} \\
0 & 0 & 0 & J_{21} & J_{22} & J_{23} \\
0 & 0 & 0 & J_{31} & J_{32} & J_{33}
\end{bmatrix}
^{-1}
\begin{bmatrix}
F_1 \\
F_2 \\
F_3 \\
M_1 \\
M_2 \\
M_3
\end{bmatrix}.
$$  

(7.45)

Velocity at $(n+1)^{th}$ level is computed as

$$v_{n+1} = v_n + \dot{v}\Delta t,$$

(7.46)

$$\omega_{n+1} = \omega_n + \dot{\omega}\Delta t.$$

(7.47)

The displacement of the body is computed by the following equations

$$\Delta x = \frac{v_{n+1} + v_n}{2} \Delta t + \mathbf{r} \times \frac{\omega_{n+1} + \omega_n}{2} \Delta t.$$

(7.48)
Numerical Examples

**Moving wedge:** I compare the computed results for the flow past a wedge problem using the fluid-object interaction code with available analytical results. The mesh, used to simulate this problem, is shown in Fig. 7.8. First, I compute for the flow results at freestream Mach number \(M = 3.0\) to arrive at the steady state solution (see Fig. 7.9(left)). Then, I start moving the wedge in the downstream direction such that the equivalent freestream Mach number is 2.8. Note that this problem does not require any fluid object interaction (FOI) computation. The FOI solver has been used for demonstration purposes only. The computed values of mach cone angles with the analytical values are compared. There is a good agreement between these two results as shown in table 7.3.

![Figure 7.8 Mesh for wedge problem.](image-url)
<table>
<thead>
<tr>
<th></th>
<th>Computed ($\theta_c$)</th>
<th>Analytical ($\theta_a$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed Wedge</td>
<td>54.5166°</td>
<td>54.7888°</td>
</tr>
<tr>
<td>Moving Wedge</td>
<td>52.0013°</td>
<td>52.4206°</td>
</tr>
</tbody>
</table>

**Table 7.3**  Mach cone angle with moving wedge

![Density comparison](image)

**Figure 7.9**  Density at the beginning (left) and end (right) of computations.

**Jet Intake:**  The intake of a jet engine flying at variable supersonic Mach number is made of an adjustable spool. The spool is adjusted in such a way that the supersonic cone is aligned with the tip of the outer casing. The freestream Mach number is 2 and the Reynolds number based on the freestream values and the gap size is $0.8 \times 10^6$. This problem has been described in more detail by Aliabadi in [37].

The Gambit [48] automatic mesh generator was used to generate the unstructured mesh shown in Fig. 7.10. Because of the axisymmetric nature of the solution this problem can be reduced to a 2D problem. The spool in the center is prescribed a sinusoidal motion of $-0.2 \sin(0.5\pi(t - 1))$ in the time interval $1 \leq t \leq 2$ as described by Aliabadi in [37].

The Mach number distribution at different instances in one quarter of a cycle ($1 \leq t \leq 2$) is shown in Fig. 7.11. First, I computed the supersonic flow without
Figure 7.10  The fluid mesh for Jet Intake.

fluid-object interaction. After reaching the steady state solution, the fluid-object interaction solver were used to allow the spool to move. The results, shown in Fig.7.11, was verified with the results obtained by Aliabadi in [37].
Figure 7.11  Supersonic flow through air intake of a jet engine: Mach number distribution at different instances in one quarter of a cycle. Only half of the geometry shown here was used in the computational domain.
7.3.3 Fluid-Structure Interactions

In this section, I am going to discuss the fluid-structure interaction problems in compressible flow.

Unlike the fluid-object interaction, the structure here is allowed to go through deformations. The coupling between structure and fluid is dealt with in an iterative fashion as discussed in Section 7.6. The structure is modeled as a membrane type structure. The structure can also be replaced by some other geometry which undergoes deformation as a result of fluid dynamics forces. One such example is the "flow past a thin flexible beam." First, I will present results from the "flow past a thin beam" followed by results from the "flow past a cross-parachute."

Numerical examples

Flow past a thin flexible beam: At the inflow boundary, I have velocity $U = 1.0 \text{ m/s}$, density $\rho = 1.0 \text{ kg/m}^{-3}$, and kinematic viscosity $\nu = 1.792 \times 10^{-6} \text{ N.s/m}^2$. The beam is made of a material with Young’s Modulus $Y = 75 \text{ GPa}$ and density $\rho = 2.75 \text{ kg/m}^3$. It is 1.8 mm thick and 2 m long. At the center point, it is simply supported as shown in Fig. 7.12. I used approximately 9000 rectangular fluid elements to solve this problem. Mesh is refined near the beam as shown in Fig. 7.13 to accurately capture the physics. Here, I present results for two cases – slightly compressible flows at inflow Mach number $M = 0.3$ and supersonic inflow at $M = 3.0$.

M=0.3, Slightly compressible: Fig. 7.14 shows the results obtained for the slightly compressible case. Here, I observed alternate vortex shedding from the top and the bottom tips of the beam. This result is consistent with the results obtained
Figure 7.12  Flow past a thin flexible beam: Problem definition.

Figure 7.13  Flow past a thin flexible beam: Fluid mesh.
from a subsonic flow solver. The frequency of vortex shedding and the frequency of beam-vibration are close to each other. This implies that the beam is vibrating at the vortex shedding frequency.

![Image](image_url)

**Figure 7.14** Pressure at M=0.3 (top), and displacement-time plot (bottom).

**M=3.0, Supersonic flow:** The freestream Mach number was increased to a supersonic Mach number M=3.0. At this high Mach number, I observed a steady state flow as shown in Fig. 7.15. In front of the beam, a detached normal shock is
observed (Fig. 7.15(top)) as expected for a bluff body at high Mach number flow. At the downstream side tip-ends of the beam, expansion waves are formed.

In the early phase, the tips of beam go through oscillations due to the start-up conditions as observed in Fig. 7.15(bottom). However, these oscillations are quickly damped out with time because fluid acts as a natural damper. Using the steady state load from fluid on the beam, I used stand-alone over damped time dependent Euler beam equations to compute tip displacement and compared it with the one computed from the FSI results as shown in Fig. 7.15(bottom). Both of the results match well with each other at steady state as expected.
Figure 7.15  Density at M=3.0 (top), displacement-time plot(bottom).
**Deformable cross-parachute:** Here, I will present the results for FSI simulations of a cross-parachute falling at Mach number $M=3.0$. The payload is pinned. The problem setup for the cross-parachute is described in Chapter 6. It required several preparatory stage simulations before starting the FSI simulations. First, a stand-alone fluid dynamics simulation using the compressible flow solver was carried out to obtain a fully developed solution. I used the prestressed structure geometry from the simulations discussed in Chapter 6. The FSI results are presented in Fig. 7.16. At time $t=4$ (Fig. 7.16(top)), the parachute geometry is close to the initial prestressed geometry that I assumed. The parachute is very wide. A bow shock is formed in front of the parachute. At time $t=4$ (Fig. 7.16(middle)), the width of the parachute is reduced as a result of aerodynamic forces. Consequently, the size of the bow shock also gets narrower. The shock follows the object as expected. The width of the subsonic downstream wake behind the parachute reduces as the width of the parachute becomes smaller. Slowly, the oscillations in the canopy become insignificant. The parachute has nearly reached steady state solution at time $t=45$.

### 7.4 Conclusion

In this chapter, compressible fluid–structure interaction methodologies were developed. A shock capturing algorithm based on density gradient was discussed. It was found that the density-gradient based shock capturing algorithm predicts a sharper shock than the entropy based shock capturing algorithm. The capabilities of compressible fluid structure/object interaction solver was successfully demonstrated by solving the supersonic flow past a jet engine with adjustable spool and a deformable cross–parachute with fixed payload.
Figure 7.16  Compressible FSI simulations of cross parachute at different instances. Time is nondimensionalized by D/2U where D is the width of the cross parachute in flat configuration and U is the freestream velocity.
In the next chapter, I will summarize the findings of this thesis with possible future directions.
Chapter 8
Conclusion

Fluid dynamics is very important in the study of many real life problems. In most of the practical problems, the physics behind the fluid dynamics is coupled with the physics of other areas. One such example is fluid–structure interaction (FSI) of parachute systems during an airdrop. The parachute is represented as a structure composed of membranes, cables, and concentrated masses. The cables and membranes are assumed to have no flexural rigidity and experience large displacements and rotations. As a result, the interaction between a parachute system and the surrounding flow field is dominant in most of the parachute operations. Thus, the ability to predict parachute FSI is a challenge that must be faced in airdrop systems modeling. The larger the parachute, the stronger the interaction between fluid and structure. The stronger the interaction between fluid and structure, the more difficult it is to accurately and efficiently carry out the FSI simulations.

In this thesis, simulation and modeling techniques for FSI problems in both incompressible and compressible flow regime were discussed. FSI simulations consist of the following primary components: a solution method for the fluid dynamics, a solution method for the structural dynamics, an intelligent way to efficiently move and update a finite element mesh of fluid, and strategies for the coupling of fluid and structural dynamics along the fluid–structure interface. The coupling is achieved in staggered fashion, with the fluid and structure coupled iteratively within a nonlinear
iteration loop, and with multiple nonlinear iterations improving the convergence of the coupled system. The mesh domain was treated as a pseudo-elastic solid.

In most of the FSI simulations, the structure undergoes large deformations. The deformations depend on time. Therefore, the fluid dynamics (FD) solver requires a methodology which can handle time-dependent spatial domain deformations. The Deforming-Spatial Domain / Stabilized Space-Time (DSD/SST) method is one such method which has a built-in capability to handle deforming structures. So, the DSD/SST was used for the FD solver which is well suited for such problems. Semi-discrete formulations were used to carry out structural dynamics (SD) simulations which are based on the principle of virtual work. The SD model usually consists of membranes along with cables and payloads.

Results from the FSI simulations of a cluster of two G–12 parachutes under three conditions – fixed payload, airdrop, and soft-landing, were presented. The G–12 is a very large cargo parachute and its analysis presents convergence difficulties at the start of FSI simulations. A pressure ramping technique was proposed to deal with this start-up mismatch in the information exchange between the parachute’s canopy and the fluid.

Results for FSI simulations of a cluster of three G–12 parachutes with fixed payload were presented. Two of the parachutes in this case came very close to contact. The FSI simulations faced a numerical barrier on the onset of contact. Similar contact issues were observed in the case of two parachute simulations when one or more gores collapsed. A contact algorithm based on pressure penalty was proposed and was successfully applied to a real life problem involving the cluster of a three parachute case. It was qualitatively shown that this contact model preserved the basic behavior
of the canopy without letting them touch but allowing them to come very close to each other. I believe that the contact model had a minimal effect on the overall aerodynamics of the parachutes and preserved the basic overall shape of the contacting surfaces while allowing one to continue the FSI simulations.

Paratroopers need to be able to maneuver the chute to land in the radius of safety zones. Round parachutes have little maneuvering capabilities. A porosity model, based on Darcy’s law, was presented and showed that this feature could be used to fine tune the trajectory. This feature could also be used to model the permeability effect of the fabric materials on the overall performance of the parachute system.

All of these FSI simulations were carried out with a Navier-Stokes solver written for incompressible flows when the Mach number is, ideally speaking, assumed to be less than 0.3. However, for many applications such as the drogue parachute deployment stage during the airdrop of the X-38 spacecraft, the Mach number is more than 0.3. These kinds of flows fall into the compressible flow regime, hence the incompressible FSI solver cannot be used to accurately model them. The existing capabilities of the FSI solver are enhanced and a compressible module is added to perform the analysis of compressible fluid–structure interaction problems. First, a density gradient based shock-capturing method is proposed to efficiently capture the shocks and accurately predict their location in supersonic flows. It is successfully applied to a test problem for which the analytical solution is available and it is shown that it gives a better result than using the entropy based shock-capturing terms. Then, the compressible FSI solver is discussed and is successfully applied to the Euler beam problem. Along with this, a compressible fluid–object interaction is also presented and successfully applied to a jet engine intake problem. The results from the com-
pressible FSI simulations of a cross-parachute problem was also presented and it was shown that the shock waves follow the deforming geometry.

8.1 Future Directionss

Numerical simulations of real life problems always suffer from many numerical and modeling difficulties. Often the physics of a flow, for which accurate representation is not possible or is beyond the scope of numerical capabilities, is numerically modeled. Further studies are needed to improve the capabilities of numerical modeling for a given amount of computational resources to best represent the real life problem. Some of them along the line of my research which were presented in this thesis are as follows:

- Turbulence: Little is known about the behavior of turbulence modeling with stabilized finite element formulations. A better understanding of turbulence modeling to appropriately account for all turbulence scales in association with stabilization parameters is needed.

- Clusters of parachutes: Further simulations of parachute clusters are required to better understand their dynamic behavior. Larger parachute clusters (namely four, five, six, and twelve used by the US Army) are needed. Alternative coupling strategies should be explored, including direct coupling or a variant of direct coupling to deal with strong interactions.

- Contact: A new contact model, to study gore-to-gore contact phenomena, is needed. Additionally, when contact occurs there is no fluid between the contacting surfaces. A model which should account for no-fluid zone areas along contacting surfaces is needed.
• Porosity: Further studies using the real porosity of the fabric materials need to be completed. In real life, the porosity of a parachute is changed by heating the canopy or reorienting the fabric structure. A more sophisticated model which depends on temperature change along with pressure drop across the canopy is needed.

• Compressible FSI: The FSI simulations of an object airdropped at a high Mach number (usually the stage with drogue parachute deployment) need to be carried out. The FSI solver needs to be improved to account for the traction forces on structures from fluid which, I believe, will improve nonlinear convergence of Newton-Raphson iteration.

• Linear iterative solver: For highly nonlinear systems, the condition number of matrix($A$) of the linearized system becomes very large. The extent of nonlinearity in the physics could surpass the applicability of a given preconditioned GMRES iterative solver when the coupling between the fluid and the structure becomes very severe, when a large time step size is used resulting in a larger Courant number or when the Mach number is very large resulting in sharper shocks. I used a diagonal preconditioned GMRES and experimented with a block-diagonal preconditioned GMRES. I found that the block-diagonal preconditioner works better than diagonal-preconditioner. I believe that a better preconditioner based on flow physics is needed. One example for the preconditioner is to assume a full matrix in the highly nonlinear zone and a diagonal everywhere else. Another issue arising from the localized full matrix assumption is the performance of the iterative solver on parallel computers. More
studies are required to better understand such preconditioners and their effect on performance.

- Everything parallelized: In the current FSI solver, the automatic mesh-generator, and the data-projection algorithm between two fluid-meshes at the time of remeshing or between incompatible interface meshes in FSI simulations are carried out on a serial processor. This negatively affects the performance of parallel computations. A parallel mesh generator and a parallelized version of the data-projection algorithm is needed to achieve good performance for large systems such as some of those treated in this thesis.
References


59. T.E. Tezduyar and T.J.R. Hughes, “Development of time-accurate finite element techniques for first-order hyperbolic systems with particular emphasis on


Appendix A

Compressible flows

A.1 Advective and Diffusivity Coefficient Matrices

The coefficient matrices of a generalized advective-diffusive system used in section (2.23) is described here. For simplicity, We define the following parameters:

\[ \overline{\gamma} = \gamma - 1 \]  \hspace{1cm} (A.1)
\[ \overline{\mu} = \mu - \frac{\kappa}{C_v} \]  \hspace{1cm} (A.2)
\[ c_1 = \overline{\gamma}||u||^2 - \gamma e \]  \hspace{1cm} (A.3)
\[ c_2 = \gamma e - \frac{\overline{\gamma}}{2}||u||^2 \]  \hspace{1cm} (A.4)
\[ c_3 = \frac{\kappa}{C_v} \left( \frac{1}{2}||u||^2 - i \right) - \mu||u||^2 \]  \hspace{1cm} (A.5)

A.1.1 Derivation of Coefficient Matrices for 3D Compressible Flow

In order to derive the matrices \( A_1, A_2 \) and \( A_3 \), one has to express fluxes \( F_1, F_2 \) and \( F_3 \) solely in terms of independent variables \( U \):

\[
F_1 = \begin{pmatrix}
  u_1 \rho \\
  u_1 \rho u_1 + p \\
  u_1 \rho u_2 \\
  u_1 \rho u_3 \\
  u_1 (\rho e + p)
\end{pmatrix}
\]  \hspace{1cm} (A.6)

\[
F_1 = \begin{pmatrix}
  u_1 \rho \\
  u_1 \rho u_1 + \overline{\gamma} \rho (e - \frac{1}{2}||u||^2) \\
  u_1 \rho u_2 \\
  u_1 \rho u_3 \\
  u_1 (\rho e - \frac{1}{2} \overline{\gamma} ||u||^2)
\end{pmatrix}
\]  \hspace{1cm} (A.7)

\[
F_1 = \begin{pmatrix}
  u_1 \rho \\
  (\rho u_1)^2 / \rho + \overline{\gamma} \rho e - \frac{\overline{\gamma}}{2}((\rho u_1)^2 + (\rho u_2)^2 + (\rho u_3)^2) / \rho \\
  (\rho u_1)(\rho u_2) / \rho \\
  (\rho u_1)(\rho u_3) / \rho \\
  \gamma (\rho u_1)(\rho e) / \rho - \frac{\overline{\gamma}}{2}((\rho u_1)^2 + (\rho u_2)^2 + (\rho u_3)^2) / (\rho)^2
\end{pmatrix}
\]  \hspace{1cm} (A.8)
Similarly,

\[
F_2 = \begin{pmatrix}
  u_2\rho \\
  u_2\rho u_1 \\
  u_2\rho u_2 + p \\
  u_2\rho u_3 \\
  u_2(\rho e + p)
\end{pmatrix}
\]

(\text{A.9})

\[
= \begin{pmatrix}
  u_2\rho \\
  (\rho u_2)/(\rho e^2 - \frac{\gamma}{2}(\rho u_1)^2 + (\rho u_2)^2 + (\rho u_3)^2)/\rho \\
  (\rho u_2)/(\rho u_3)/\rho \\
  \gamma(\rho u_2)/(\rho e - \frac{\gamma}{2}(\rho u_1)^2 + (\rho u_2)^2 + (\rho u_3)^2)/\rho^2
\end{pmatrix}
\]

(\text{A.10})

\[
F_3 = \begin{pmatrix}
  u_3\rho \\
  u_3\rho u_1 \\
  u_3\rho u_2 \\
  u_3\rho u_3 + p \\
  u_3(\rho e + p)
\end{pmatrix}
\]

(\text{A.11})

\[
= \begin{pmatrix}
  u_3\rho \\
  (\rho u_3)/(\rho e^2 - \frac{\gamma}{2}(\rho u_1)^2 + (\rho u_2)^2 + (\rho u_3)^2)/\rho \\
  (\rho u_3)/(\rho u_2)/\rho \\
  \gamma(\rho u_3)/(\rho e - \frac{\gamma}{2}(\rho u_1)^2 + (\rho u_2)^2 + (\rho u_3)^2)/\rho^2
\end{pmatrix}
\]

(\text{A.12})

The advective coefficient matrices can be derived as:

\[
A_1 = \begin{pmatrix}
  0 & 1 & 0 & 0 & 0 \\
  \frac{\gamma}{2}||u||^2 - u_1 u_1 & (3 - \gamma)u_1 & -\gamma u_2 & -\gamma u_3 & \gamma \\
  -u_1 u_2 & u_2 & u_1 & 0 & 0 \\
  -u_1 u_3 & u_3 & 0 & u_1 & 0 \\
  c_1 u_1 & c_2 - \gamma u_1 u_1 & -\gamma u_1 u_2 & -\gamma u_1 u_3 & \gamma u_1
\end{pmatrix}
\]

(\text{A.13})

\[
A_2 = \begin{pmatrix}
  0 & 1 & 0 & 0 & 0 \\
  -u_2 u_1 & u_2 & u_1 & 0 & 0 \\
  -u_2 u_2 & -\gamma u_1 & (3 - \gamma)u_2 & -\gamma u_3 & \gamma \\
  -u_2 u_3 & 0 & u_3 & u_2 & 0 \\
  c_1 u_2 & -\gamma u_2 u_1 & c_2 - \gamma u_2 u_2 & -\gamma u_2 u_3 & \gamma u_2
\end{pmatrix}
\]

(\text{A.14})
\[ A_3 = \begin{pmatrix}
0 & 0 & 0 & 1 & 0 \\
-u_3 u_1 & u_3 & 0 & u_1 & 0 \\
-u_3 u_2 & 0 & u_3 & u_2 & 0 \\
\frac{\bar{\gamma}}{2} ||u||^2 - u_3 u_3 & -\bar{\gamma} u_1 & -\bar{\gamma} u_2 & (3 - \gamma) u_3 & \bar{\gamma} \\
c_1 u_3 & -\bar{\gamma} u_3 u_1 & -\bar{\gamma} u_3 u_2 & c_2 - \bar{\gamma} u_3 u_3 & \gamma u_3 \\
\end{pmatrix} \]  
(A.15)

Diffusive coefficient matrices are obtained by expanding the diffusive flux terms. They are given as:

\[ K_{11} = \frac{1}{\rho} \begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
-\frac{4\mu}{3} u_1 & \frac{4\mu}{3} & 0 & 0 & 0 \\
-\mu u_2 & 0 & \mu & 0 & 0 \\
-\mu u_3 & 0 & 0 & \mu & 0 \\
c_3 - \frac{\bar{\mu}}{3} u_1 u_1 & (\frac{\bar{\mu}}{3} + \bar{\mu}) u_1 & \bar{\mu} u_2 & \bar{\mu} u_3 & \frac{\kappa}{C_v} \\
\end{pmatrix} \]  
(A.16)

\[ K_{12} = \frac{1}{\rho} \begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
\frac{2\mu}{3} u_2 & 0 & -\frac{2\mu}{3} & 0 & 0 \\
-\mu u_1 & \mu & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
-\frac{\mu}{3} u_1 u_2 & \mu u_2 & -\frac{2\mu}{3} u_1 & 0 & 0 \\
\end{pmatrix} \]  
(A.17)

\[ K_{13} = \frac{1}{\rho} \begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
\frac{2\mu}{3} u_2 & 0 & 0 & -\frac{2\mu}{3} & 0 \\
0 & 0 & 0 & 0 & 0 \\
-\mu u_1 & \mu & 0 & 0 & 0 \\
-\frac{\mu}{3} u_1 u_3 & \mu u_3 & 0 & -\frac{2\mu}{3} u_1 & 0 \\
\end{pmatrix} \]  
(A.18)

\[ K_{21} = \frac{1}{\rho} \begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
-\mu u_2 & 0 & \mu & 0 & 0 \\
\frac{2\mu}{3} u_1 & -\frac{2\mu}{3} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
-\frac{\mu}{3} u_2 u_1 & -\frac{2\mu}{3} u_2 & \mu u_1 & 0 & 0 \\
\end{pmatrix} \]  
(A.19)

\[ K_{22} = \frac{1}{\rho} \begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
-\mu u_1 & \mu & 0 & 0 & 0 \\
-\frac{4\mu}{3} u_2 & 0 & \frac{4\mu}{3} & 0 & 0 \\
-\mu u_3 & 0 & 0 & \mu & 0 \\
c_3 - \frac{\bar{\mu}}{3} u_2 u_2 & \bar{\mu} u_1 & (\frac{\bar{\mu}}{3} + \bar{\mu}) u_2 & \bar{\mu} u_3 & \frac{\kappa}{C_v} \\
\end{pmatrix} \]  
(A.20)
\[
K_{23} = \frac{1}{\rho} \begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
\frac{2\mu}{3} u_3 & 0 & 0 & -\frac{2\mu}{3} & 0 \\
-\mu u_2 & 0 & \mu & 0 & 0 \\
-\frac{\nu}{3} u_2 u_3 & 0 & \mu u_3 & -\frac{2\mu}{3} u_2 & 0 \\
\end{pmatrix}
\]
(A.21)

\[
K_{31} = \frac{1}{\rho} \begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
-\mu u_3 & 0 & 0 & \mu & 0 \\
0 & 0 & 0 & 0 & 0 \\
\frac{2\mu}{3} u_1 & -\frac{2\mu}{3} & 0 & 0 & 0 \\
-\frac{\nu}{3} u_3 u_1 & -\frac{2\mu}{3} u_3 & 0 & \mu u_1 & 0 \\
\end{pmatrix}
\]
(A.22)

\[
K_{32} = \frac{1}{\rho} \begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
-\mu u_3 & 0 & 0 & \mu & 0 \\
0 & 0 & 0 & 0 & 0 \\
\frac{2\mu}{3} u_2 & 0 & -\frac{2\mu}{3} & 0 & 0 \\
-\frac{\nu}{3} u_2 u_3 & 0 & -\frac{2\mu}{3} u_3 & \mu u_2 & 0 \\
\end{pmatrix}
\]
(A.23)

\[
K_{33} = \frac{1}{\rho} \begin{pmatrix}
0 & 0 & 0 & 0 & 0 \\
-\mu u_1 & \mu & 0 & 0 & 0 \\
0 & 0 & \frac{4\mu}{3} & 0 & 0 \\
\frac{\nu}{3} u_2 u_2 & 0 & 0 & \mu & 0 \\
c_3 - \frac{\nu}{3} u_2 u_2 & \mu u_1 & (\frac{\nu}{3} + \mu) u_2 & \mu u_3 & \frac{\nu}{c_e} \\
\end{pmatrix}
\]
(A.24)

(A.25)

### A.1.2 Derivation of Coefficient Matrices for 2D Compressible Flow

By carrying out the similar analysis as discussed for 3D case, we arrive at the following coefficient matrices.

Adveective matrices are given by:

\[
A_1 = \begin{pmatrix}
0 & \frac{\nu}{2} ||\mathbf{u}||^2 - u_1 u_1 & 1 & 0 & 0 \\
\frac{\nu}{2} ||\mathbf{u}||^2 - u_1 u_1 & (3 - \gamma) u_1 & -\gamma u_2 & \gamma \\
-u_1 u_2 & u_2 & u_1 & 0 \\
c_1 u_1 & c_2 - \gamma u_1 u_1 & -\gamma u_1 u_2 & \gamma u_1 \\
\end{pmatrix}
\]
(A.26)

\[
A_2 = \begin{pmatrix}
0 & 0 & 1 & 0 \\
-u_2 u_1 & u_2 & u_1 & 0 \\
\frac{\nu}{2} ||\mathbf{u}||^2 - u_2 u_2 & -\gamma u_1 & (3 - \gamma) u_2 & \gamma \\
c_1 u_2 & -\gamma u_2 u_1 & c_2 - \gamma u_2 u_2 & \gamma u_1 \\
\end{pmatrix}
\]
(A.27)

Diffusive coefficient matrices given as:
\[
\mathbf{K}_{11} = \frac{1}{\rho} \begin{pmatrix}
0 & 0 & 0 & 0 \\
-\frac{4\mu}{3} u_1 & \frac{4\mu}{3} & 0 & 0 \\
-\mu u_2 & 0 & \mu & 0 \\
c_3 - \frac{\mu}{3} u_1 u_1 (\frac{\mu}{3} + \bar{\mu} + \bar{u}) & (\frac{\mu}{3} + \bar{\mu}) u_1 & \bar{\mu} u_2 & \frac{\nu}{c_v}
\end{pmatrix}
\]  
(A.28)

\[
\mathbf{K}_{12} = \frac{1}{\rho} \begin{pmatrix}
0 & 0 & 0 & 0 \\
\frac{2\mu}{3} u_2 & 0 & -\frac{2\mu}{3} & 0 \\
-\mu u_1 & \mu & 0 & 0 \\
-\frac{\mu}{3} u_1 u_2 & \mu u_2 & -\frac{2\mu}{3} u_1 & 0
\end{pmatrix}
\]  
(A.29)

\[
\mathbf{K}_{21} = \frac{1}{\rho} \begin{pmatrix}
0 & 0 & 0 & 0 \\
-\mu u_2 & 0 & \mu & 0 \\
\frac{2\mu}{3} u_1 & -\frac{2\mu}{3} & 0 & 0 \\
-\frac{\mu}{3} u_1 u_2 & -\frac{2\mu}{3} u_2 & \mu u_1 & 0
\end{pmatrix}
\]  
(A.30)

\[
\mathbf{K}_{22} = \frac{1}{\rho} \begin{pmatrix}
0 & 0 & 0 & 0 \\
-\mu u_1 & \mu & 0 & 0 \\
-\frac{4\mu}{3} u_2 & 0 & \frac{4\mu}{3} & 0 \\
c_3 - \frac{4\mu}{3} u_2 u_2 & \bar{\mu} u_1 (\frac{\mu}{3} + \bar{\mu} + \bar{u}) & \frac{\nu}{c_v}
\end{pmatrix}
\]  
(A.31)