SIMULATION OF CARDIAC THERMAL AND FLUID-STRUCTURE INTERACTION PHYSICS USING THE FINITE ELEMENT METHOD: METHODS, DEVELOPMENT AND APPLICATIONS

By

Tejas Patel

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Mechanical Engineering – Doctor of Philosophy

2023

ABSTRACT

The human heart is a highly complex organ, and its primary function is to pump blood through the arteries, veins and to perfuse all other body tissues and organs, including itself. In the last decade, cardiac simulations have become increasingly crucial to gain clinical insight into cardiac function, treatment, and testing. Nowadays, multi-physics cardiovascular simulations applied to patient-specific modeling can help in the diagnosis of cardiovascular diseases and in studying relevant clinical treatments. Hence, our central objective here is to develop a generalized multi-physics finite element (FE) framework that includes thermal-fluid structure interaction coupling to study cardiac function and treatments.

First, we developed a stabilized FE based flow solver with heat transfer to study hemodynamics. A python based open-source FE library (FEniCS) is used from ground-up to custom-build the solver. We benchmark and validate the solver and study convergence for classical test cases at intermediate Reynolds and Peclet number.

Second, we utilize the solver to investigate cryoballoon ablation (CBA), which is a minimally invasive surgery that uses freezing or cryoenergy to treat atrial fibrillation (AF). To begin with, we use a patient-specific left atrium (LA) geometry and realistic pulmonary vein (PV) blood flow boundary conditions to validate hemodynamics of the LA chamber. Next, we position a cryoballoon (CB) at the pulmonary vein ostium to simulate incomplete occlusion during cryotherapy and investigate the factors affecting lesion formation. We observe that lesion size is highly sensitive to the CB position and balloon tissue contact area. The threshold gap for lesion formation is 2.4 mm. We also note that as the balloon tissue contact area increases, the surgery is more effective, and the power absorbed across the CB reduces.

Third, we extend our development to a fully coupled fluid-structure interaction (FSI) solver with heat transfer using FEniCS. The FSI solver (named vanDANA) that uses the immersed boundary (IB) method is based on the Distributed Lagrange Multiplier based Fictitious Domain method and the interpolation of variables is conducted using the smeared delta-functions. Additionally, the structure can be set as incompressible or compressible. We benchmark our solver and analyze the scalability on HPC. This builds a solid foundation for the future use of this solver.

This thesis is dedicated to the patience and all the sacrifices made by my mom and dad, to friends in Michigan, and to my personal growth during this journey.
Thank you for always being present - knowingly/unknowingly for me.

PREFACE

The content included in Chapter 3, Chapter 4, Appendix A and Appendix B of this thesis has contributed towards the following publication:

Patel, Tejas, Chris Li, Farshad Raissi, Ghassan S. Kassab, Tong Gao, and Lik Chuan Lee. "Coupled thermal-hemodynamics computational modeling of cryoballoon ablation for pulmonary vein isolation." Computers in Biology and Medicine (2023): 106766.

TABLE OF CONTENTS

CHAPTER 1 : GENERAL BACKGROUND

1.1 Heart Anatomy and cardiac cycle diagram

The human heart is a highly complex organ, and its primary function is to pump blood through the arteries, veins and to perfuse all other body tissues and organs, including itself. This process facilitates the exchange of gases, fluids, electrolytes and heat exchange between the cells and the ambient environment [1]. The cardiac anatomy mainly comprises of two atriums and two ventricles separated by atrio-ventricular (AV) valves, and other arteries and veins that deliver blood to and

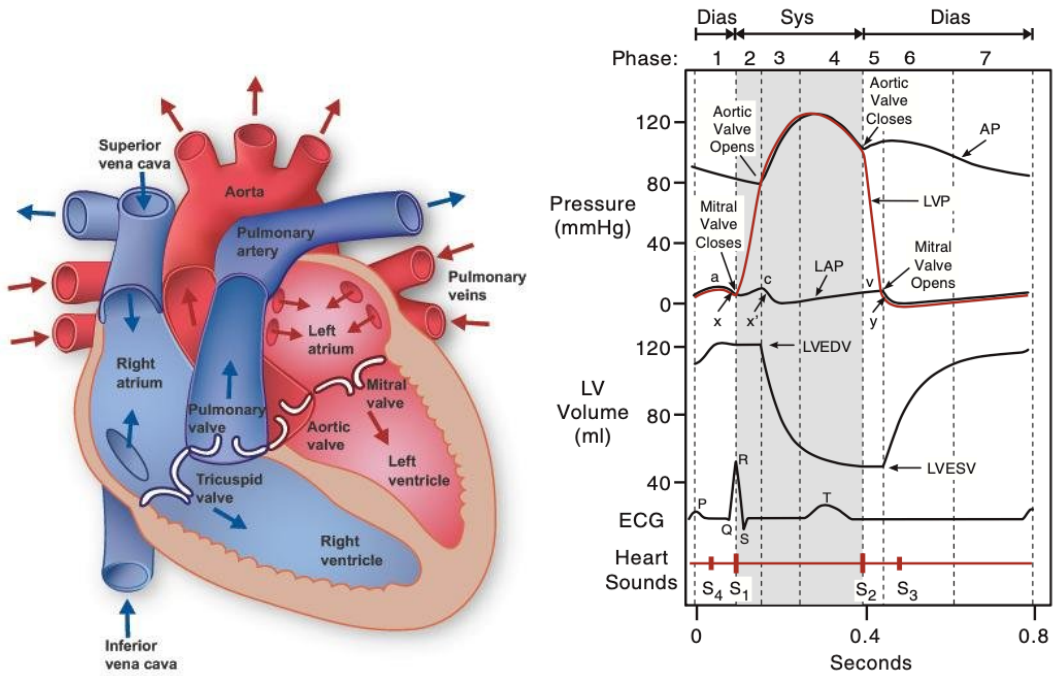

Figure 1.1: a) Heart anatomy with four chambers, two atria and two ventricles, separated by AV valves. The red color on the left and blue color on the right side depicts the flow of oxygenated blood and deoxygenated blood respectively. b) The Wiggers diagram. It demonstrates pressure-volume relationship for different heart chambers in sync with a healthy electrocardiogram (ECG). Figure adopted from Klabunde [1].

away from the heart chambers (see **Figure 1.1a**). The right heart comprises of the right atrium (RA) and right ventricle (RV) that are connected by a tricuspid AV valve and the left side comprises of the left atrium (LA) and the left ventricle (LV) that are connected by a mitral valve (MV). These valves function with the help of chordae tendineae and papillary muscles. Functionally, the heart

can be viewed as two pumps connected in series, one responsible for systemic circulation and the other one for pulmonary circulation. The systemic circulation comprises of the blood vessels inside and outside of all the organs excluding the lungs and the pulmonary circulation denotes blood flow within the lungs that allows the exchange of gases and oxygen in the alveoli. Deoxygenated blood collected from the body in the systemic circulation passes through the inferior and superior vena cava, and then enters the RA and the right ventricle (RV). The deoxygenated blood is then pumped by the RV into the pulmonary artery and lungs for oxygenation. The oxygenated blood from the lungs enters the LA through the pulmonary veins (PV's) in the left heart. The oxygenated blood enters the LV, which then pumps the blood into the aorta and systemic circulation. The pumping capacity of the heart is usually expressed by **cardiac output** = amount of blood pumped during each contraction (stroke volume) * heart rate (HR).

In **Figure 1.1b**, the cardiac cycle diagram (aka. The Wiggers Diagram) illustrates the pressure and volume waveforms in the left heart, which is normally divided into two different phases, namely systole and diastole. Systole refers to ventricular contraction and ejection whereas diastole refers to ventricular relaxation and filling. In the left heart**,** the MV closes when pressure in the LV is higher than that in the LA, marking the beginning of systole. This onset of systole is denoted by the QRS complex which represents ventricular depolarization. Next, with all the valves closed, the intraventricular pressure begins to rise until there is a pressure difference across the LV and aorta. This pressure difference causes the aortic valve to open, and blood is then ejected out of the ventricles. About approximately 180 ms after the QRS complex, a T-wave initiates ventricular repolarization. The ventricle muscle starts to relax, and the intraventricular pressures start to drop. This marks the beginning of diastole and soon after, the aortic valve closes, and the LV volume remains constant (since all valves are closed). This volume is called End-Systolic Volume (ESV).

In healthy humans, the ESV is approximately 50 mL. Pressure in the LV continues to drop, and when the LV pressure falls below the LA pressure, the MV opens. This initiates ventricular filling which is mainly divided into two phases: passive filling without atria contraction and filling because of atria contraction. Passive filling occurs before the P-wave in the ECG, and this accounts for the bulk of LV filling. As the LV continues to fill up with blood, they become stiffer (less compliant) and the intraventricular pressures start to rise. The P-wave in the ECG is associated with electrical depolarization of the atria, which results in atrial contraction and an increase in LA pressure as marked by the a-wave in the LA pressure curve in **Figure 1.1b.** Atria contraction (sometimes known as atrial kick) forces more blood into the LV before the MV closes at the end of diastole. Filling due to atrial contraction accounts for only 10% of ventricular filling. This behavior is reflected in the spectral doppler MV velocity graph in **Figure 1.2**, wherein the E peak represents rapid filling during LV relaxation and A peak represents filling due to the atrial kick. In heathy human subjects, the average E/A ratio ranges between 0.75 to 1.5.

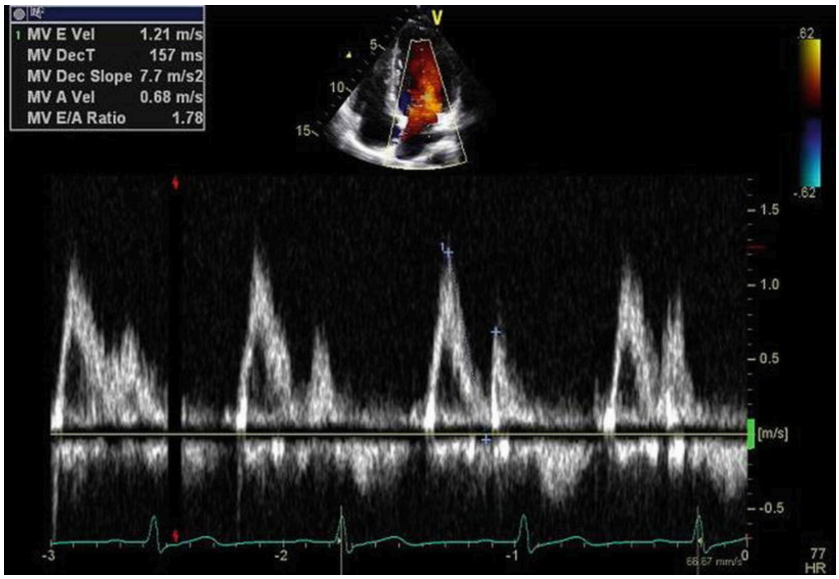

Figure 1.2: Normal Mitral Valve (MV) inflow velocity measured for a healthy subject using spectral doppler echocardiography. Here Deceleration Time (DecT) is defined as the time taken for the maximum velocity at the E peak to drop to zero. Figure adopted from [2].

1.2 Background of this dissertation

1.2.1 Atrial Fibrillation

Atrial fibrillation (AFib) is one of the most common chronic cardiac arrythmias where the upper chambers of the heart (atria) beat out of synchronization with the lower chambers i.e., the ventricles. This increases the risk of stroke and heart failure, and the cardiac output reduces by approximately 30%. In healthy humans, the ECG comprises of the P-wave, QRS complex and the T-wave. For patients suffering from AFib however, the P-waves are missing, and the heartbeat is highly erratic, see **Figure 1.3**. In these patients, electrical depolarization currents arise from multiple sites throughout the atria, resulting to quiver and uncoordinated beating with no discernable P-waves. The symptoms of AFib range from nothing to fatigue, shortness of breath, dizziness, palpitations, or chest pain. Although AFib is not life threatening by itself, it can be seriously uncomfortable, is associated with many cardiovascular diseases and requires medical treatment [3]. There are different classifications of AFib [4]:

- **Paroxysmal Atrial Fibrillation:** Intermittent episodes occur and usually stop within 24 hours, without any treatment.
- **Persistent Atrial Fibrillation:** Last longer than seven days and may or may not cease on its own.
- **Long-standing Atrial Fibrillation:** Last longer than a year and changes the cardiac structure.
- **Permanent Atrial Fibrillation:** Permanent and can cannot be treated by medical practices and can severely diminish quality of life.

Depending on the patients' symptoms, AFib is treated with medicine to control the heart rhythm and using blood thinners to prevent clotting. If medications do not help, the patient often undergo cardioversion or a cardiac ablation treatment.

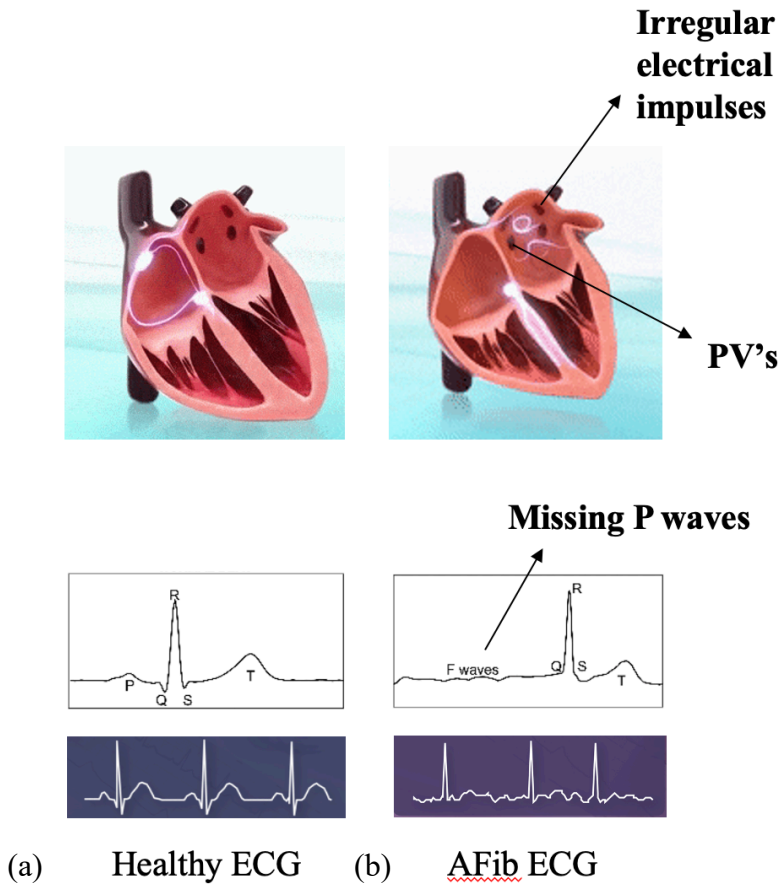

Figure 1.3: Pathway of electrical signals and ECG of (a) a healthy patient vs (b) an AFib patient. Note how irregular electrical impulses are generated at the PV opening in case of an AFib heart.

1.2.2 Cryoballoon ablation

In AFib patients, irregular electro-physiological signals usually begin where the PV is attached to the LA. In these patients, it becomes very important to cease the erratic beating and hence the transmission of irregular electrical activity from the PV to the rest of the cardiac muscle in the atria. There are non-invasive (e.g., cardioversion) and invasive AFib treatments that include catheter ablation or the maze procedure. Cardioversion supplies low-energy shocks to regulate a

patient's heartbeat [5]. However, this is only a temporary solution and will not cure the underlying cause. There are two types of minimally invasive procedures for treating AFib, namely, radiofrequency ablation (RFA) and cryoballoon ablation (CBA). Radiofrequency ablation or point-by point focal ablation is a more traditional approach that uses heat to destroy the heart tissue. Cryoballoon ablation, on the other hand, is a recent cryo-energy based technique. It has become one of the most prominent surgical procedures to treat AFib [6]. Medtronic, based out of Minneapolis MN, is the leading producer of devices used for CBA surgery. Since the release of CB2 (second-generation cryoballoon) by Medtronic [7, 8]: Arctic Front Advance in 2012 and Arctic Front Advance Pro in 2020, efficiency of CBA surgery has increased drastically.

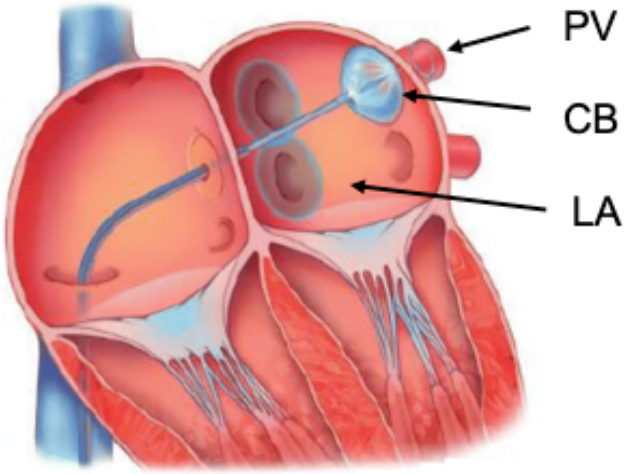

Figure 1.4: Cryoballoon Ablation catheter. For CBA, vascular access is obtained at the femoral vein and the LA is instrumented via a trans-septal puncture. Figure taken from Patel et al. [9].

Ideally, CBA is conducted stepwise, first for the left PV's and then for the right PV's. During CBA, the electrophysiologist positions the balloon and then keep a heavy forward pressure to occlude the PV ostium (see **Figure 1.4**) [8, 9]. Before ablation, the cryoballoon (CB) is manually flushed back and forth in the PV antrum to eliminate any trapped air-bubbles in the blood stream. An 8-pole circular recording catheter attached to the frontal tip of the CB continuously monitors PV

potentials in real time. By exploiting the Joule-Thompson effect, liquid nitrous oxide is injected in the CB and a phase change from liquid-to-gas gradually absorbs heat from the LA. When the temperature in the cardiac tissue falls below 23^0C, electrical dormancy is observed, and isolation is confirmed by the loss of PV potentials. A conduction delay (or a conduction block) is normally observed at the catheter [10] and the time taken to achieve this isolation is called time to isolation (TTI), which is normally between 30-60 sec.

After TTI, the physiologist ideally holds the CB in place as per the recommended dosing protocol - about 2 to 4 minutes per freeze. Permanent cellular damage is detected at temperatures below -20^0C, if held for 60 seconds. On the other hand, if the temperature is less than -40^0C (necrosis temperature), the hold duration is not important, and the cardiac tissue becomes irreversibly nonviable. At this stage, irreparable lesion formation is initiated at the balloon-tissue contact surface and for an optimal scar/lesion, temperature monitored at the CryoConsole normally ranges between -40^0C to -50^0C [10]. However, this is only the return cryogenic gas temperature and not the actual temperature of the cryoballoon. Nadir temperatures at the CB surface can go as low as -60^0C to -70^0C. The most important factors controlling lesion thickness and depth for a successful CBA surgery is namely, balloon-tissue contact area (particularly no leakage at all), time of freeze and the source of cryo-energy. Overall, CBA surgery forms a contiguous lesion and is less time consuming (lower TTI) compared to RFA, which means that the patient is on general anesthesia for a shorter duration. This makes CBA safer than RFA. Cryoballoon ablation has also led to more predictable surgeries with less reliance on electrophysiologist skills and improved quality of life post-surgery. Due to its obvious advantages over RFA, CBA has become the preferred choice of ablation surgery to treat AFib patients.

1.2.3 Fluid Structure Interactions

Fluid structure interaction (FSI) is the interaction of single or multiple flexible structures (bodies) in adjacent fluid media. Depending on the physical nature, geometry and the incompressibility of the media, the structural deformation can be insignificant or large. Such interactions between the fluid and structure can be found in many day-to-day applications such as aerodynamic flows, bioengineering, marine applications, and turbomachinery. Almost all bio-driven physics in the human body e.g., hemodynamics of aneurysms, arterial stenosis, blood flow in the heart chambers, circle of Willis, and the flow of urine in kidneys and the bladder involve FSI. Developing numerical models to study FSI is crucial and can be used as a practical tool to probe the physical behavior of biosystems. In problems involving FSI, one needs to solve the Navier-Stokes (NS) equations [11] for fluid flow as well as the structural mechanics equations describing deformation of the flexible body. The fluid and structure equations are coupled in both directions, meaning each medium exerts a force on the other and subsequently influences their temporal behavior. For example, in vortex induced vibrations (VIV) over bluff bodies, a condition is reached when the vortex shedding frequency is close enough to the body's natural frequency and the body begins to oscillate naturally. The forces resulting from vortex shedding cause the body to oscillate and subsequently the body's oscillation affects the nature of the flow around it. Hence to build a stable design, one needs to consider vortex shedding, forces acting on the body and the natural frequency and amplitude of the body's oscillation at various flow speeds. Vortex induced vibrations is a very fundamental FSI phenomenon and is widely studied to investigate vibrations caused due to flow over cables, bridges, heat exchanger tubes, ocean drilling risers etc.

Another widely studied FSI is the cyclic pumping of blood through arteries. Arteries are compliant and deformable, and the blood flow exerts pressure on the walls causing them to stretch

and deform. However, if the shear stress is too high or cyclic it can lead to endothelial dysfunction that may contribute to the development of atherosclerosis. This is because increased shear stress can cause damage to the endothelial cell lining, leading to the formation of plaque and narrowing of the blood vessel. Simulations assuming that the arterial wall is rigid usually overestimates the true wall shear stress [12]. To address this issue, simulations should consider FSI to accurately estimate the wall shear stress and help identify areas of the arterial system that are at a higher risk of developing endothelial damage. **Figure 1.5** shows an idealized flow pattern over plaque in an artery. Here, on the distal side, the shear stress is oscillatory and this compounds to more plaque formation over time. Also in real practice, it is fundamentally imperative to include patient specific geometries for such simulations. Overall, understanding how shear stress affects arterial walls can help identify risk factors associated with diseases and help optimize the design of treatments.

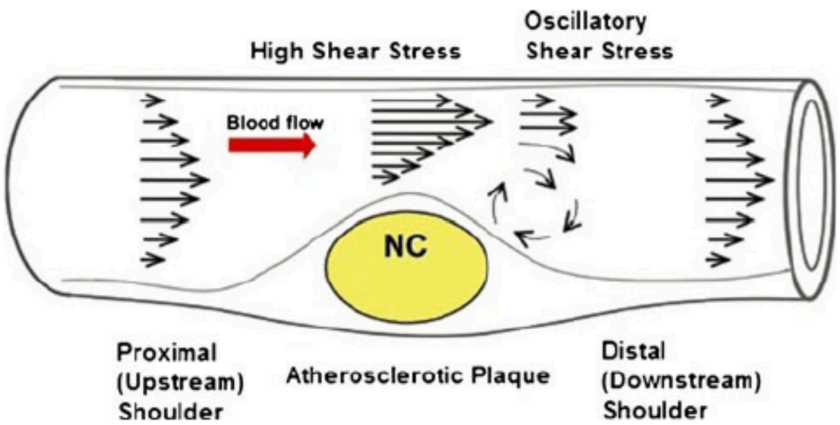

Figure 1.5: Ideal distribution of wall shear stress an arterial segment with atherosclerotic plaque formation. Figure adopted from Young Cho et al [13].

1.2.4 Coding in FEniCS

The FEniCS project (https://fenicsproject.org [14]), initiated in 2003, is an open-source python platform to solve partial differential equations (PDE's) using the finite element method (FEM). The backend of FEniCS mainly comprises of fundamental building blocks: DOLFIN, FIAT, FFC (FEniCS Form Compiler), UFL (Unified Form Language) and UFC (Unified Form Assembly

Code). FEniCS includes automated code generation capabilities, which generate C++ code for a variational weak form written in high level UFL language. This eliminates the need for manually coding complex subroutines and allows one to quickly generate code in just a few lines of python. This permits quick prototyping, debugging, and testing of the physical problem at hand. Moreover, DOLFIN also provides a set of preconditioned Krylov solvers and PETSc as a linear algebra backend, that support easy assembly and conversion to matrix-vector form before solving it iteratively. In addition, for experienced programmers, FEniCS offers straightforward compilation of custom-built C++ codes into their high-level coding structure. Due to all these advantages, we choose FEniCS to develop our FEM codes for modeling FSI. **Figure 1.6** shows an example FEniCS script for solving a 3D elastic deformation problem in case of a hyper-elastic material. For post-processing, the results can be easily visualized on the opensource ParaView (Kitware Inc.) [15].

```python
from dolfin import *

# Create mesh and define function space
mesh = UnitCubeMesh(24, 16, 16)
V = VectorFunctionSpace(mesh, "Lagrange", 1)

# Define functions
du = TrialFunction(V)            # Incremental displacement
v  = TestFunction(V)             # Test function
u  = Function(V)                 # Displacement from previous iteration
B  = Constant((0.0, -0.5, 0.0))  # Body force per unit volume
T  = Constant((0.1,  0.0, 0.0))  # Traction force on the boundary

# Kinematics
d = u.geometric_dimension()
I = Identity(d)             # Identity tensor
F = I + grad(u)             # Deformation gradient
C = F.T*F                   # Right Cauchy-Green tensor

# Invariants of deformation tensors
Ic = tr(C)
J  = det(F)

# Elasticity parameters
E, nu = 10.0, 0.3
mu, lmbda = Constant(E/(2*(1 + nu))), Constant(E*nu/((1 + nu)*(1 -
2*nu)))

# Stored strain energy density (compressible neo-Hookean model)
psi = (mu/2)*(Ic - 3) - mu*ln(J) + (lmbda/2)*(ln(J))**2

# Total potential energy
Pi = psi*dx - dot(B, u)*dx - dot(T, u)*ds

# Compute first variation of Pi (directional derivative about u in the
direction of v)
F = derivative(Pi, u, v)

# Solve variational problem
solve(F == 0, u, bcs, form_compiler_parameters=ffc_options)

# Save solution in VTK format
file = File("displacement.pvd");
file << u;
```

Figure 1.6: Simple FEniCS script to solve a 3D elasticity problem for a hyper-elastic cube. This python code is copied from the FEniCS tutorial: <u>demo hyperelasticity.py</u>.

1.3 Objectives of this dissertation

This dissertation is organized as follows:

1. **Chapter 3:** Develop a computational modeling framework in FEniCS to investigate flow and heat transfer to investigate cardiovascular flows.
2. **Chapter 4:** Simulate patient specific CBA surgery for pulmonary vein isolation.
   - Validate present computational framework for patient-specific left atrium hemodynamics.
   - Investigate the factors affecting CBA surgery.
   - Analyze temperature distribution and efficiency for various cryoballoon positions in the PV antrum.
3. **Chapter 5:** Develop an immersed boundary (IB) distributed Lagrange multiplier (DLM) based fictitious domain (FD) method in FEniCS to model FSI for biophysical flows.
   - Create a user-friendly open-source solver and publish/document it on GitHub: **vanDANA** (https://github.com/patelte8/vanDANA).
   - Validate and measure HPC-scalability of **vanDANA** solver (IB-FSI algorithm) using classical benchmark problems.

CHAPTER 2 : REVIEW OF THE STATE OF ART

2.1 Challenges, complications, and the need for cryotherapy simulations

One of the major challenges in a CBA surgery is to achieve complete occlusion at the PV ostium. In case of a leakage at the PV ostium, energy (cold) is convected into the blood stream, and this hinders lesion formation. Another critical challenge is to monitor the real time evolution and distribution of temperature (cold) in the PV antrum. This is very critical in preventing complications like the phrenic nerve and esophageal injury. Phrenic nerve injury (PNI) is not a common occurrence in RFA. Due to the anatomical proximity of the phrenic nerve to the right PV's and more contiguous lesions in CBA, however, many studies [16, 17, 18] have reported that the risks of PNI for CB2's are increased up to 8% - 10%. On the other hand, esophageal injury is observed for both RFA and CBA due to proximity of the esophagus to the posterior LA wall (see **Figure 2.1**). Hence, it is very important for the electrophysiologist to monitor real time temperature in the esophagus by placing an esophageal probe near the CB [19]. If the balloon and tissues are not in proper contact, then TTI could be longer than 60 sec or there could be no-isolation at all. In such conditions, a better and safer strategy is to abort the freeze and reposition the CB at another position on the PV ostium. Considering all the risks involved, optimum CB positioning is very crucial for an effective CBA surgery [20].

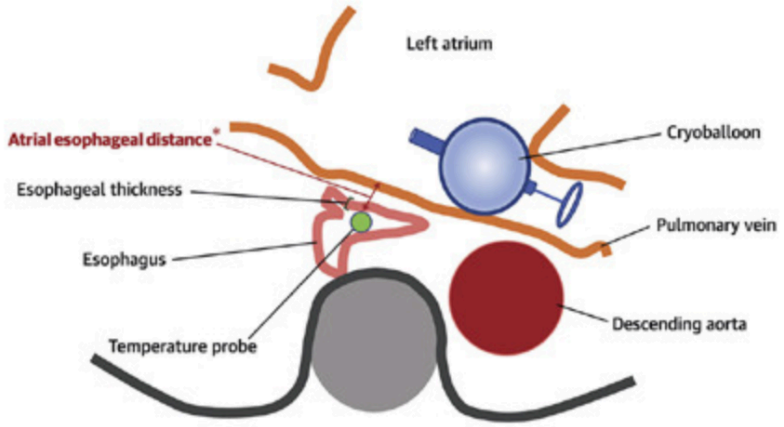

Figure 2.1: Figure adopted from Sarairah S.Y. et al. [19].

Today with the growth in computational modeling, these challenges to CBA can be addressed by conducting simulations of the temperature field for patient specific geometries prior to the surgery. Numerical models to simulate cryotherapy can be developed to investigate patient specific LA geometries. Some numerical studies have been conducted to investigate the thermal field in the PV antrum. Getman et. al. [7] studied the relationship between TTI and freeze duration where they modeled the temperature at various tissue depths using COMSOL. Their simulations for both the 28 mm CB2's: Arctic Front Advance and Arctic Front Advance Pro demonstrated similar TTI's across various cardiac tissue depths up to 5 mm, where they observed a TTI of about 60 sec for a lesion depth of 3 mm. Similarly, Xia et al. [21] used a more simplistic 2D-axisymetric geometry for the balloon-PV contact surface and they compare their numerical findings with an in-house experimental balloon setup surrounded by a freshly cut porcine myocardium tissue. They conducted a thorough tissue damage analysis and found that if the temperature of the tissue drops below $-40^0\mathrm{C}$ (necrosis temperature), the duration of hold is unimportant due to direct cell damage [22]. Whereas, for temperatures less than $-20^0\mathrm{C}$, they found that it is important to hold the CB for at least 60 sec to produce necrosis of the tissue. They observe that the necrosis $-40^0\mathrm{C}$ isotherm only reaches a depth of about 2 mm after 240 sec of ablation. At depths greater than 2 mm, the hold time becomes decisive to lesion/scar formation. Both these studies [7, 21] model the cold by solving for the Pennes heat conduction (or energy) equation in time and accounting for convection due to blood flow in the LA antrum.

Today, with the advent of CBA surgery and the diverse range of governing factors affecting the surgery, there is a need for patient-specific computational modeling within a LA geometry to simulate the coupling between blood flow and heat transfer during CBA. Our goal mainly lies in

investigating how the presence of a CB affects hemodynamics in the LA, and how incomplete occlusion (leakage) affects temperature distribution at the PV ostium and lesion formation.

2.2 Fluid structure interactions: complexities, algorithms, and applications

To simulate real-world FSI, requires a robust, accurate, and flexible algorithm that can handle complex 3D moving geometries. In such geometries, the implementation of boundary conditions and mesh generation are major challenges and also the fundamental reason that influence the accuracy of results. Mesh generation is critical especially when the objects are moving or deforming and if not done efficiently, can lead to high computational overheads [23]. Most FSI mesh-based algorithms can be classified into two families: boundary-fitted methods and non-boundary-fitted methods, according to whether the boundary-fitted mesh is used for the solution of the flow field [24, 25]. The boundary fitted method is the traditional approach where the mesh conforms to the boundary of the domain and remeshing is required as the interface deforms with time. The classical boundary fitted method is Arbitrary-Lagrangian Eularian (ALE) method [26]. Besides a fine grid resolution at the FSI interface, the ALE method also requires discretization of the governing equations on grids aligned with the boundary. The other class of methods are the non-boundary fitted methods, where the flow field is computed on a stationary Eulerian grid and remeshing is not required for the fluid grid. This saves a considerable chuck of the computational cost in terms of remeshing the flow grid. Sometimes, remeshing might be required for the structural grid if the extent of deformation is large which is computationally less expensive as the solid grid is less in volume as compared to the fluid grid. The immersed boundary (IB) and fictitious domain (FD) algorithms are examples of non-boundary fitted methods.

The IB method typically loses accuracy at the solid-fluid interface due to the numerical difficulty in constructing a conservative mapping function between the Eulerian (fluid) mesh and

the Lagrangian (solid) mesh. As a result, IB methods cannot resolve the fine details of boundary flow for accurate shear stress computation. Today, there are two different FD based schemes in the literature: non-body-force-based scheme and the body-force-based scheme. The method used by Farnell et al. [27] for the simulation of a filament in a flowing soap film belongs to the first category. The second, is the IB method proposed by Peskin [28] which is a body force based FD method that has been applied to a wide range of FSI problems (Zhu and Peskin [29], Eggleton and Popel [30]). In this method, the flexible body moves at the same velocity as the local fluid, and then affects the fluid motion through an elastic force that is calculated with the known deformation of the body and is introduced into the fluid momentum equation as a pseudo body force. Here the key idea is to virtually extend the exterior fluid domain into the solid where a distribution of pseudo body forces is employed to enforce the structural movement through Lagrange multipliers. Such methods have been implemented by Kamensky et al. [31], Kapada et al. [32] and Yu [33].

Instead of using the macroscopic NS equations to solve for the fluid flow, one can flexibly combine particle based methods with the immersed boundary method. Particle-based methods model the fluid as a collection of discrete particles rather than as a continuous fluid. Two commonly used particle-based methods are Smoothed Particle Hydrodynamics (SPH) and Lattice Boltzmann Method (LBM). SPH [34] is a Lagrangian method that models the fluid as a collection of particles, where each particle is assigned a mass, velocity, and density. The fluid equations are solved by computing the forces between the particles using a kernel function that smooths out the interaction between neighboring particles. LBM [35, 36, 37, 38] is a mesh-free method that discretizes the fluid domain into a lattice of cells, where each cell represents a small volume of fluid. The fluid equations are solved by tracking the distribution of particles within each cell using a probabilistic approach [39]. LBM is well-suited for simulating complex fluid behavior such as

turbulence and multiphase flows. Recently, particle-based methods have become more and more popular because it offers advantages in terms of parallelization (to efficiently model large-scale simulations), simple implementation, and robustness for FSI applications.

Models describing FSI in cardiovascular systems

In the last two decades, the use of FSI modeling has increased in cardiovascular engineering, especially in the field of arterial blood flow and cardiac hemodynamics. Computational models of FSI are used to model the abdominal aorta [40], carotid bifurcation [41], heart valves [42] and cerebral aneurysm [43]. Torii, Wood et al. [44] has studied the effects of wall compliance on a patient-specific right coronary artery with severe stenosis and found noticeable differences in the instantaneous wall shear stress produced by the FSI and rigid wall models. Apart from the difference in patient geometries, mechanical properties of the vessel wall also vary significantly. The larger arteries are more elastic than the smaller ones. Deformations up to 10% of the vessel radius are common in large arteries. This elasticity gives rise to the Windkessel effect and provides a passive mechanism for smoothing the pulsatile blood flow from the heart. The Windkessel effect is mainly attributed to the distension of large arteries during systole, functioning as a buffering reservoir for blood [45]. Besides arteries, lately FSI models have also been used to simulate heart valve dynamics [46] and cardiac chamber flows [47]. The earliest coupled heart-valve FSI model was developed by Peskin and McQueen's in the 1970s [48, 49, 50] using the classical IB approach. Chandran and Kim [51] reported a study with MV dynamics in a simplified LV chamber model during diastolic filling using an immersed interface-like approach. One of the main limitations of this coupled models is the simplified representation of the biomechanics of the LV wall. Recently, Gao et al. [47] have developed a MV–LV model which has full FSI and is based on a realistic geometry and an experimental model for soft tissue mechanics. Such complex FSI simulations

require a robust multiphysics coupling algorithm that allows large deformations and incorporates turbulence modeling and contact mechanics.

Given the complexities and challenges, we aim to develop an open-source robust IB based FSI solver using the delta-functions which can simulate any complex 3D geometry. Moreover, we also intend to benchmark, document, and measure the scalability of our IB-FSI algorithm on HPC. This will enable us to create a solid foundation for the future use of this solver.

CHAPTER 3 : DEVELOPMENT OF STABILIZED FINITE ELEMENT MODEL TO INVESTIGATE THERMAL-HEMODYNAMICS IN CARDIAC CHAMBERS

3.1 Methods

3.1.1 Thermal-Hemodynamics coupling model

Hemodynamics and heat transfer in the LA are described by solving the continuity, incompressible NS and energy conservation equations. To non-dimensionalize the governing equations [11], we introduce a characteristic length L_{sc} and velocity V_{sc} to define the non-dimensional control parameters ($\mathbb{C}$): Reynolds number $Re = \frac{V_{sc}\, L_{sc}}{\upsilon}$ and Prandtl number $Pr = \frac{\upsilon}{\alpha}$, where υ is the kinematic viscosity (m^2/s) and α is the thermal diffusivity (m^2/s) of the fluid. We also define the non-dimensional Peclet number $Pe = Re \cdot Pr$, which is a measure of the ratio of advective to diffusive transport of temperature. We restrict our modeling to moderate Re and assume laminar flow everywhere. Correspondingly, the complete non-dimensional governing equations is given as follows:

In the fluid domain Ω:

$$\frac{\partial \boldsymbol{u}}{\partial \tau} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nabla \cdot \boldsymbol{\sigma} - \boldsymbol{f} = \boldsymbol{0}, \tag{3.1}$$

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

$$\frac{\partial \mathrm{T}}{\partial \tau} + \boldsymbol{u} \cdot \nabla \mathrm{T} - \frac{\nabla^2 \mathrm{T}}{Pe} - Q = 0, \tag{3.3}$$

Constitutive model - Newtonian Fluid:

$$\boldsymbol{\sigma}(\boldsymbol{u}, p) = -p\boldsymbol{I} + \frac{2}{Re}\boldsymbol{E}(\boldsymbol{u}), \tag{3.4}$$

$$\boldsymbol{E}(\boldsymbol{u}) = \frac{1}{2}(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T), \tag{3.5}$$

On the boundaries:

$$\boldsymbol{u} = \boldsymbol{u}_{\boldsymbol{\omega}} \quad at \quad \partial\Omega_D^{\boldsymbol{u}}; \qquad \boldsymbol{\nabla u} \cdot \boldsymbol{n} = \overline{\boldsymbol{u}}_{\boldsymbol{\omega}} \quad at \quad \partial\Omega_N^{\boldsymbol{u}}, \tag{3.6}$$

$$p = p_\omega \quad at \quad \partial\Omega_D^p; \qquad \nabla p \cdot \boldsymbol{n} = \overline{p}_\omega \quad at \quad \partial\Omega_N^p, \tag{3.7}$$

$$\mathrm{T} = \mathrm{T}_\omega \quad at \quad \partial\Omega_D^{\mathrm{T}}; \qquad \nabla \mathrm{T} \cdot \boldsymbol{n} = \overline{\mathrm{T}}_\omega \quad at \quad \partial\Omega_N^{\mathrm{T}}. \tag{3.8}$$

In the above equations, $\boldsymbol{u}$, p and T are the primary unknown variables defining the fluid velocity, pressure, and temperature fields, respectively. The blood is assumed to behave as a Newtonian fluid. Correspondingly in Eqs. (3.4) and (3.5), $\boldsymbol{\sigma}$ and $\boldsymbol{E}$ are, respectively, the stress and strain rate tensors. In Eq. (3.1), $\boldsymbol{f}$ is the prescribed body force, τ is the non-dimensional time and $\nabla(\cdot)$ is the gradient operator, whereas in Eq. (3.3), Q is a prescribed heat source. The Dirichlet and Neumann conditions are prescribed for the unknown variables $\boldsymbol{u}$, p and T at the boundary in Eqs. (3.6-3.8), with $\boldsymbol{n}$ denoting the vector normal to the boundary and $(\boldsymbol{u}_\omega, \overline{\boldsymbol{u}}_\omega, p_\omega, \overline{p}_\omega, \mathrm{T}_\omega, \overline{\mathrm{T}}_\omega)$ denoting the prescribed values. In these equations, the boundary is denoted by $\partial\Omega^{\boldsymbol{\phi}} = \partial\Omega_D^{\boldsymbol{\phi}} \cup \partial\Omega_N^{\boldsymbol{\phi}}$ such that $\partial\Omega_D^{\boldsymbol{\phi}} \cap \partial\Omega_N^{\boldsymbol{\phi}} = \{\,\}$, where $\boldsymbol{\phi} \subseteq \{\, \boldsymbol{u}, p, \mathrm{T} \,\}$.

3.1.2 Stabilized Finite element method

The FE method is used for spatial discretization of the governing equations. Denoting $\boldsymbol{v}$, q and w as the respective test functions for $\boldsymbol{u}$, p and T, the resultant Galerkin weak formulation is given as follows:

$$\langle \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{\tau}}, \boldsymbol{v} \rangle + \langle \boldsymbol{u} \cdot \boldsymbol{\nabla} \boldsymbol{u}, \boldsymbol{v} \rangle + \langle \boldsymbol{\sigma}, \boldsymbol{\nabla} \boldsymbol{v} \rangle - \langle \boldsymbol{\sigma} \cdot \boldsymbol{n}, \boldsymbol{v} \rangle_{\partial\Omega} - \langle \boldsymbol{f}, \boldsymbol{v} \rangle = \boldsymbol{0}, \tag{3.9}$$

$$\langle \boldsymbol{\nabla} \cdot \boldsymbol{u}, q \rangle = 0, \tag{3.10}$$

$$\langle \frac{\partial \mathrm{T}}{\partial \tau}, w \rangle + \langle \boldsymbol{u} \cdot \nabla \mathrm{T}, w \rangle + \frac{1}{Pe} \left[\langle \nabla \mathrm{T}, \nabla w \rangle - \langle \overline{\mathrm{T}}_\omega, w \rangle_{\partial\Omega_N^{\mathrm{T}}} \right] - \langle Q, w \rangle = 0, \tag{3.11}$$

where $\{\boldsymbol{u} \in H^1(\Omega),\ \boldsymbol{u} = \boldsymbol{u_\omega}\ at\ \partial\Omega_D^{\boldsymbol{u}}\}; \{p \in H^1(\Omega), p = p_\omega\ at\ \partial\Omega_D^p\}; \{\mathrm{T} \in H^1(\Omega), \mathrm{T} = \mathrm{T}_\omega\ at\ \partial\Omega_D^{\mathrm{T}}\}; \{\boldsymbol{v} \in H^1(\Omega), \boldsymbol{v} = \boldsymbol{0}\ at\ \partial\Omega_D^{\boldsymbol{u}}\}; \{q \in H^1(\Omega), q = 0\ at\ \partial\Omega_D^p\}$ and $\{w \in H^1(\Omega), w = 0\ at\ \partial\Omega_D^{\mathrm{T}}\}$. In the above equations, $\langle \cdot, \cdot \rangle$ and $\langle \cdot, \cdot \rangle_{\partial\boldsymbol{\Omega}}$ denote the inner product of two functions i.e., $\langle z_1, z_2 \rangle = \int_{\boldsymbol{\Omega}} z_1 z_2 \cdot d\Omega$ and $\langle z_1, z_2 \rangle_{\partial\Omega} = \int_{\partial\Omega} z_1 z_2 \cdot ds$ for functions z_1 and z_2.

Because of the high Re ($\sim 10^3$), Pe($\sim 10^5$) and velocities, hemodynamics in the heart chambers is dominated by advection. As such, it is necessary to incorporate appropriate stabilization schemes to the Galerkin weak formulation to ensure numerical stability [52]. To do so, we applied the well-established Streamline-Upwind Petrov Galerkin (SUPG) [53, 54] and the crosswind stabilization scheme developed by Codina [55]. The SUPG stabilization scheme is applied by introducing the following to Eqs. (3.9) and (3.11):

$$\int_\Omega \gamma_{SUPG}(\boldsymbol{u})\ P(\boldsymbol{u}, \boldsymbol{m}) \cdot \boldsymbol{R}^{\boldsymbol{\phi}}\, d\Omega, \tag{3.12}$$

where,

$$\gamma_{SUPG}(\boldsymbol{u}) = \alpha \left[\left(\frac{2}{\Delta\tau}\right)^2 + \left(\frac{2\,||\boldsymbol{u}||}{h}\right)^2 + 9\left(\frac{4}{\mathbb{C}\,h^2}\right)^2 \right]^{-\frac{1}{2}}, \tag{3.13}$$

$$P(\boldsymbol{u}, \boldsymbol{m}) = \boldsymbol{u} \cdot \nabla \boldsymbol{m}\,. \tag{3.14}$$

In the above, h denotes the cell diameter, $\mathbb{C}$ denotes the respective control parameter (i.e., Re or Pe) and the coefficient α is prescribed with a value of 0.85.

Next, we introduce another residual pressure-based stabilization term to Eq. (3.9), namely the pressure stabilizing Petrov Galerkin (PSPG) scheme [19] that is given as:

$$\int_\Omega \gamma_{PSPG}(\boldsymbol{u})\, \nabla q \cdot \left(\frac{\partial \boldsymbol{u}}{\partial \tau} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nabla \cdot \boldsymbol{\sigma} - \boldsymbol{f}\right) d\Omega, \tag{3.15}$$

where $\gamma_{PSPG} = \gamma_{SUPG}$. The crosswind stabilization scheme adopted from Codina [18] is also applied by introducing the following as additional term to the weak formulation into Eqs. (3.9) and (3.11):

$$\int_{\Omega} \left(\gamma_{CW}(\boldsymbol{\phi})\ \Lambda(\boldsymbol{u},\boldsymbol{\phi})\right) : \boldsymbol{\nabla m}\, d\Omega, \tag{3.16}$$

where

$$\Lambda(\boldsymbol{u},\boldsymbol{\phi}) = \boldsymbol{\Gamma}(\boldsymbol{u}) \cdot \nabla\boldsymbol{\phi}, \tag{3.17}$$

$$\gamma_{CW}(\boldsymbol{\phi}) = \frac{1}{2}\beta_c h \frac{\left|\left|\boldsymbol{R}^{\phi}\right|\right|}{\left|\left|\nabla\boldsymbol{\phi}\right|\right|} \tag{3.18}$$

In the above, $\boldsymbol{\Gamma}(\boldsymbol{u}) = \begin{Bmatrix} \boldsymbol{I} - \frac{\boldsymbol{u}\otimes\boldsymbol{u}}{|\boldsymbol{u}|^2} & if\ \boldsymbol{u} \neq \boldsymbol{0} \\ \boldsymbol{0} & if\ \boldsymbol{u} = \boldsymbol{0} \end{Bmatrix}$, $\boldsymbol{\phi}$ is the unknown variable, and $\beta_c = \max\left(0, \zeta - \frac{2\,||\boldsymbol{\nabla\phi}||}{\mathbb{C}\,h\,\left|\left|\boldsymbol{R}^{\phi}\right|\right|}\right)$ [17]. In both Eqs. (3.12) and (3.16), $\boldsymbol{R}^{\boldsymbol{\phi}}$ and $\boldsymbol{m}$ are the residual and test function. The crosswind stabilization scheme works in such a way that a non-linear numerical dissipation proportional to the element residual is added in the crosswind direction in regions where the convective effects are dominant. It should be noted that SUPG, PSPG and crosswind stabilizations are highly consistent because they are all residual based terms and vanish as the residual approaches to zero.

Next, we note that when the control parameters (Re, Pe) are large, applying only the SUPG, PSPG and crosswind stabilization schemes is not sufficient to prevent non-physical oscillations. Hence, to further increase stability and accuracy of the flow solver, we also include two additional stabilization terms to Eq. (3.9). Specifically, the Least-Squares on Incompressibility Constraint (LSIC) stabilization is used to improve the accuracy and conditioning of mass balance in the LA. This stabilization scheme is implemented by adding the following to Eq. (3.9):

$$\int_{\Omega} \gamma_{LSIC} \, (\nabla \cdot \boldsymbol{u}) \, (\nabla \cdot \boldsymbol{v}) \, d\Omega, \tag{3.19}$$

where $\gamma_{LSIC} = \frac{2}{3\,Re}$. To address the numerical instability arising from flow reversal at the boundaries with high Re that is common in biological flow problems [20], we include the following penalty-based backflow stabilization term as:

$$K_p \int_{\partial\Omega_D^p} \big((\boldsymbol{I} - \boldsymbol{n}\otimes\boldsymbol{n})\, \boldsymbol{u}\big) \cdot \boldsymbol{v} \, ds, \tag{3.20}$$

which is active only on the boundaries $\partial\Omega_D^p$ into Eq. (3.9). The addition of this term is to confine the flow to be in the normal direction at the boundary where pressure is imposed. In the above equation, $\boldsymbol{I}$ is the Identity matrix, $\otimes$ is the cartesian outer product on vectors and K_p is a penalty factor, which we prescribed a value of 10^4.

3.1.3 Time discretization

We use the standard first order forward Euler scheme for time-discretization of the inertia term in Eq. (3.9). Because of the high Re and Pe, and to avoid any strict time step restriction [56], we use the unconditionally stable Crank Nicolson time-discretization scheme for terms associated with the advection of velocity and temperature, as well as the temperature diffusion and (elliptic) viscous terms.

In the non-linear advection, crosswind, SUPG and PSPG stabilization terms, the convective velocities, and constants γ_{SUPG}, γ_{CW}, γ_{PSPG} are discretized using an explicit time-integration scheme. The convective velocity in the backflow stabilization and LSIC terms, however, are discretized using an implicit time-integration scheme. The complete time discretized weak form with all the stabilization terms is given below:

$$
\begin{aligned}
\langle \frac{\boldsymbol{u}^{\theta+1}-\boldsymbol{u}^{\theta}}{\Delta\tau}, \boldsymbol{v}\rangle &+ \langle \boldsymbol{u}^{\theta}\cdot\nabla\boldsymbol{u}_{ck}^{\theta+1}, \boldsymbol{v}\rangle + \langle \boldsymbol{\sigma}\left(\boldsymbol{u}_{ck}^{\theta+1}, p^{\theta}\right), \nabla\boldsymbol{v}\rangle - \langle \boldsymbol{\sigma}\left(\boldsymbol{u}_{ck}^{\theta+1}, p^{\theta}\right)\cdot\boldsymbol{n}, \boldsymbol{v}\rangle_{\partial\Omega} \\
&- \langle \boldsymbol{f}, \boldsymbol{v}\rangle \\
&+ \int_{\Omega} \gamma_{SUPG}\left(\boldsymbol{u}^{\theta}\right)\left(P\left(\boldsymbol{u}^{\theta}, \boldsymbol{v}\right) + \nabla q\right) \\
&\cdot \left(\frac{\boldsymbol{u}^{\theta+1}-\boldsymbol{u}^{\theta}}{\Delta\tau} + \boldsymbol{u}^{\theta}\cdot\nabla\boldsymbol{u}_{ck}^{\theta+1} - \nabla\cdot\boldsymbol{\sigma}\left(\boldsymbol{u}_{ck}^{\theta+1}, p^{\theta}\right) - \boldsymbol{f}\right) dx \\
&+ \int_{\Omega} \left(\gamma_{CW}\left(\boldsymbol{u}^{\theta}\right)\ \Lambda\left(\boldsymbol{u}^{\theta}, \boldsymbol{u}^{\theta+1}\right)\right) : \boldsymbol{\nabla v}\, dx \\
&+ \int_{\Omega} \gamma_{LSIC}\left(\nabla\cdot\boldsymbol{u}^{\theta+1}\right)\left(\nabla\cdot\boldsymbol{v}\right) dx + \langle 10^{4}\left(\boldsymbol{I}-\boldsymbol{n}\otimes\boldsymbol{n}\right)\boldsymbol{u}^{\theta+1}, \boldsymbol{v}\rangle_{\partial\Omega_D^p} = \boldsymbol{0}
\end{aligned}
\tag{3.21}
$$

$$
\langle \nabla\cdot\boldsymbol{u}^{\theta+1}, q\rangle = \boldsymbol{0} \tag{3.22}
$$

$$
\begin{aligned}
\langle \frac{\mathrm{T}^{\theta+1}-\mathrm{T}^{\theta}}{\Delta\tau}, w\rangle &+ \langle \boldsymbol{u}^{\theta+1}\cdot\nabla\mathrm{T}_{ck}, w\rangle + \frac{1}{Pe}\left[\langle \nabla\mathrm{T}_{ck}, \nabla w\rangle - \langle \overline{\mathrm{T}}_{\omega}, w\rangle_{\partial\Omega_N^{\mathrm{T}}}\right] - \langle Q, w\rangle \\
&+ \int_{\Omega} \gamma_{SUPG}\left(\boldsymbol{u}^{\theta+1}\right) P\left(\boldsymbol{u}^{\theta+1}, w\right) \\
&\cdot \left(\frac{\partial\mathrm{T}^{\theta+1}}{\partial\tau} + \boldsymbol{u}^{\theta+1}\cdot\nabla\mathrm{T}_{ck} - \frac{\nabla^2\mathrm{T}_{ck}}{Pe} - Q\right) dx \\
&+ \int_{\Omega} \left(\gamma_{CW}\left(\mathrm{T}^{\theta}\right)\ \Lambda\left(\boldsymbol{u}^{\theta+1}, \mathrm{T}^{\theta+1}\right)\right) : \nabla w\, dx = 0.
\end{aligned}
\tag{3.23}
$$

In the above equation, $\boldsymbol{u}_{ck}^{\theta+1} = \frac{\boldsymbol{u}^{\theta+1}+\boldsymbol{u}^{\theta}}{2}$ and $\mathrm{T}_{ck} = \frac{\mathrm{T}^{\theta+1}+\mathrm{T}^{\theta}}{2}$ are the Crank-Nicolson velocity and temperature, respectively. Quantities with superscripts $\theta+1$ and θ denote the current and previous timesteps, respectively.

The time-discretized equations with the stabilization terms can be solved either using a coupled scheme or a decoupled fractional stepping algorithm. Here, we use the fractional stepping algorithm that is computationally more efficient [57]. In a traditional fractional stepping algorithm (**Table 3.1**), the computation is split into two separate steps where Eq. (3.21) is decoupled from Eq. (3.22) and solved separately for $\boldsymbol{u}$, p.

Table 3.1: Algorithm of the fractional stepping scheme

Predictor step :	Solve for intermediate velocity ($\boldsymbol{u}^*$) by ignoring the pressure gradient ($\nabla p,$) in Eq. (3.9).	
Projection step :	a.	Project the pressure field onto a divergence free velocity space (using the intermediate velocity) i.e., $\boldsymbol{\nabla} \cdot \boldsymbol{u}^* = 0$. Solve iteratively and minimize pressure residual.
	b.	Perform velocity correction to accommodate for the pressure gradient.

In our problem setting, we adopted the Incremental Pressure Correction Scheme (IPCS) which uses pressure gradient from the previous time step at the predictor step, along with semi-implicit formulations for the advection terms as we found that the scheme is robust, accurate and can accommodate larger time steps that reduces the overall simulation time [58]. The two step - IPCS scheme is given as follows:

Step 1 - Velocity prediction:

$$\begin{aligned}\langle \frac{\boldsymbol{u}^* - \boldsymbol{u}^{\boldsymbol{\theta}}}{\Delta\tau}, \boldsymbol{v}\rangle + \langle \boldsymbol{u}^{\boldsymbol{\theta}} \cdot \nabla \boldsymbol{u}^*_{\boldsymbol{ck}}, \boldsymbol{v}\rangle + \langle \boldsymbol{\sigma}\left(\boldsymbol{u}^*_{\boldsymbol{ck}}, p^\theta\right), \nabla \boldsymbol{v}\rangle - \langle \boldsymbol{\sigma}\left(\boldsymbol{u}^*_{\boldsymbol{ck}}, p^\theta\right) \cdot \boldsymbol{n}, \boldsymbol{v}\rangle_{\partial\boldsymbol{\Omega}} - \langle \boldsymbol{f}, \boldsymbol{v}\rangle \\ + \int_\Omega \gamma_{SUPG}\left(\boldsymbol{u}^{\boldsymbol{\theta}}\right) P\left(\boldsymbol{u}^{\boldsymbol{\theta}}, \boldsymbol{v}\right) \\ \cdot \left(\frac{\boldsymbol{u}^* - \boldsymbol{u}^{\boldsymbol{\theta}}}{\Delta\tau} + \boldsymbol{u}^{\boldsymbol{\theta}} \cdot \nabla \boldsymbol{u}^*_{\boldsymbol{ck}} - \nabla \cdot \boldsymbol{\sigma}\left(\boldsymbol{u}^*_{\boldsymbol{ck}}, p^\theta\right) - \boldsymbol{f}\right) dx \\ + \int_\Omega \left(\gamma_{CW}\left(\boldsymbol{u}^{\boldsymbol{\theta}}\right) \Lambda\left(\boldsymbol{u}^{\boldsymbol{\theta}}, \boldsymbol{u}^*\right)\right) : \nabla \boldsymbol{v}\, dx + \int_\Omega \gamma_{LSIC} \left(\nabla \cdot \boldsymbol{u}^*\right) \left(\nabla \cdot \boldsymbol{v}\right) dx \\ + \langle 10^4 \left(\boldsymbol{I} - \boldsymbol{n} \otimes \boldsymbol{n}\right) \boldsymbol{u}^*, \boldsymbol{v}\rangle_{\partial\Omega_D^p} = \boldsymbol{0}\end{aligned} \tag{3.24}$$

Step 2 - Pressure projection & velocity correction:

$$\begin{aligned}\langle \nabla\left(p^{\theta+1} - p^\theta\right), \nabla q\rangle + \frac{1}{\Delta\tau}\langle \nabla \cdot \boldsymbol{u}^*, q\rangle \\ + \int_\Omega \gamma_{PSPG}(\boldsymbol{u}^*) \nabla q \left(\frac{\boldsymbol{u}^{\boldsymbol{\theta}} - \boldsymbol{u}^{\boldsymbol{\theta-1}}}{\Delta\tau} + \boldsymbol{u}^{\boldsymbol{\theta}} \cdot \nabla \boldsymbol{u}^{\boldsymbol{\theta}}_{\boldsymbol{ck}} - \nabla \cdot \boldsymbol{\sigma}\left(\boldsymbol{u}^{\boldsymbol{\theta}}_{\boldsymbol{ck}}, p^\theta\right)\right) dx = 0\end{aligned} \tag{3.25}$$

$$\langle \frac{\boldsymbol{u}^{\theta+1} - \boldsymbol{u}^*}{\Delta\tau}, \boldsymbol{v} \rangle + \langle \nabla\left(p^{\theta+1} - p^{\theta}\right), \boldsymbol{v} \rangle = 0. \tag{3.26}$$

In the above equations, $\boldsymbol{u}^*$ is the intermediate velocity and $\boldsymbol{u}^*_{\boldsymbol{ck}} = \frac{\boldsymbol{u}^* + \boldsymbol{u}^{\boldsymbol{\theta}}}{2}$ is the Crank-Nicolson velocity. We used (P2-P1) discretization for the flow velocity field $\boldsymbol{u}$ and pressure field p, and P2 discretization for the temperature field T. Correspondingly, we select $\zeta = 0.7$ in the crosswind stabilization constant γ_{CW} [55]. The system of equations given in Eq. (3.24-2.26) and Eq. (3.23) are solved using the high-level open-source FE library FEniCS [14]. The code can be found in https://github.com/patelte8/vanDANA.

3.2 Numerical details

We chose to make use of a modified fractional time-stepping scheme to solve for the pressure-velocity coupling instead of a coupled scheme. The coupled scheme is fully implicit, where Eq's. (3.9) and (3.10) are solved simultaneously. Compared to the IPCS fractional time-stepping scheme, the coupled scheme is more stable and can accommodate larger time steps. However, the coupled scheme is computationally more expensive because of larger matrix sizes and requires stable pairs of FE discretization that satisfy the inf-sup or Ladyzhenskaya-Babuska-Breezi (LBB) condition [59, 60]. The Taylor-Hood (TH) element is one such pair, in which the velocity field $\boldsymbol{u}$ and pressure field p are, respectively, discretized using quadratic (P2) and linear (P1) Lagrange-type elements. Equal order (P1-P1) discretization that does not obey the LBB condition can be used only with pressure stabilization PSPG [61, 62] - Eq. (3.15), to circumvent pressure instabilities arising due to the unstable LBB pairs. Here, it is crucial to understand that the SUPG stabilization (Eq. (3.12)) is responsible to stabilize the node-to-node oscillations present due to the high advective nature of the flow and is inherently required regardless of whether TH elements or PSPG stabilization are used. We mainly introduce the PSPG stabilization because we observe that it improves the

convergence rate of pressure projection equation (Eq. (3.25)) [62]. On the other hand, the fractional time-stepping scheme theoretically allows one to circumvent the LBB condition and use equal order (P1-P1) discretization without the need for PSPG stabilization. Hence, we can also use P1-P1-P1 discretization for $\boldsymbol{u}$, p and T. Furthermore, we also introduce crosswind stabilization into our weak formulation. The crosswind stabilizing operator is introduced to prevent undershooting/overshooting of the field variables in the presence of sharp gradients, especially in the vicinity of boundary layers [63]. The crosswind stabilization operator adds a non-linear numerical dissipation proportional to the element residual in the direction perpendicular to streamlines in regions where the convective effects are dominant. To better understand the effects of crosswind, LSIC and PSPG stabilizations, we also conducted additional numerical tests (see Appendix).

3.2.1 Implementation in FEniCS

FEniCS allows us to write the weak form in the high level UFL format and later assemble it into matrix-vector form to solve it iteratively. However, for a large problem size, using the naive weak form for assembly can be computationally expensive. Hence it is good practice to split the weak form into constant time-independent matrices and vectors wherever possible and perform a pre-assembly before solving for a transient problem. For the discretization in Eq. (3.24), we consider the velocity field $\boldsymbol{u}$ as piecewise quadratic (P2): $\boldsymbol{u} = \sum_1^{K_u} \hat{\psi}_i(x)\boldsymbol{u_i}$ and the pressure field p piecewise linear (P1): $p = \sum_1^{K_p} \psi_i(x)p_i$. Here, $\hat{\psi}$ and ψ are quadratic and linear FE basis functions, K_u and K_p are the Degrees of Freedom (DOF), and $\boldsymbol{u_i}$ and p_i are the nodal values for velocity and pressure respectively. To decompose the weak form, we predefine the following set of constant matrices:

$$A_{ij} = \sum_{1}^{K_u} \int_{\Omega^e} \hat{\psi}_i \cdot \hat{\psi}_j \cdot dx \qquad K_{ij} = \sum_{1}^{K_u} \frac{2}{Re} \int_{\Omega^e} \left(\frac{\nabla\hat{\psi}_i + \nabla\hat{\psi}_i^{\,T}}{2} \right) : \nabla\hat{\psi}_j \cdot dx$$

$$O_{ij} = \sum_{1}^{K_u} \int_{\Omega^e} \boldsymbol{I} : \nabla\hat{\psi}_j \cdot dx \qquad F_{ij} = \sum_{1}^{K_u} \int_{\Omega^e} \boldsymbol{f} \cdot \hat{\psi}_j \cdot dx$$

$$B_{ij} = \sum \int_{\partial\Omega^e} (\psi\, \boldsymbol{n}) \cdot \hat{\psi}_j \cdot ds \qquad M_{ij} = \sum \frac{2}{Re} \int_{\partial\Omega^e} \left(\frac{\nabla\hat{\psi}_i + \nabla\hat{\psi}_i^{\,T}}{2} \right) \boldsymbol{n} \cdot \hat{\psi}_j \cdot ds$$

$$LS_{ij} = \sum_{1}^{K_u} \int_{\Omega^e} \gamma_{LSIC} \left(\boldsymbol{\nabla} \cdot \hat{\psi}_j\right) \left(\boldsymbol{\nabla} \cdot \hat{\psi}_j\right) \cdot dx \quad BS_{ij} = \sum 10^4 \int_{\partial\Omega_D^{p\,e}} (\boldsymbol{I} - \boldsymbol{n} \otimes \boldsymbol{n})\, \hat{\psi}_i \cdot \hat{\psi}_j \cdot ds \qquad \textbf{(3.27)}$$

Other terms that depend on the evolving solution are assembled separately in the temporal loop. Their corresponding matrices are,

Convective term: $\boldsymbol{v_j}\, Y_{ij}\left(\hat{\psi}, \boldsymbol{u^\theta}\right) \dfrac{\left(\boldsymbol{u_i^*} + \boldsymbol{u_i^\theta}\right)}{2}$

SUPG stabilization: $\boldsymbol{v_j} \left[US_{ij}\left(\psi, \hat{\psi}, \boldsymbol{u^\theta}\right) \boldsymbol{u_i^*} - \chi_j\left(\psi, \hat{\psi}, \boldsymbol{u^\theta}\right) \right]$

Crosswind stabilization: $\boldsymbol{v_j}\, CW_{ij}\left(\hat{\psi}, \boldsymbol{u^\theta}\right) \boldsymbol{u_i^*}$ **.**

(3.28)

In the above equation, χ_j is a vector of size K_u. Next, we use all the matrices in Eq. (3.27) and (3.28) to rewrite the weak form:

$$\boldsymbol{v_j} \Bigg[\frac{A_{ij}}{\Delta\tau} \left(\boldsymbol{u_i^*} - \boldsymbol{u_i^\theta}\right) + Y_{ij}\left(\hat{\psi}, \boldsymbol{u^\theta}\right) \frac{\left(\boldsymbol{u_i^*} + \boldsymbol{u_i^\theta}\right)}{2} - O_{ij} p_i^\theta + K_{ij} \frac{\left(\boldsymbol{u_i^*} + \boldsymbol{u_i^\theta}\right)}{2} + B_{ij} p_i^\theta$$

$$- M_{ij} \frac{\left(\boldsymbol{u_i^*} + \boldsymbol{u_i^\theta}\right)}{2} + F_{ij} + US_{ij}\left(\psi, \hat{\psi}, \boldsymbol{u^\theta}\right) \boldsymbol{u_i^*} - \chi_j\left(\psi, \hat{\psi}, \boldsymbol{u^\theta}\right)$$

$$+ CW_{ij}\left(\hat{\psi}, \boldsymbol{u^\theta}\right) \boldsymbol{u_i^*} + LS_{ij}\, \boldsymbol{u_i^*} + BS_{ij}\, \boldsymbol{u_i^*} \Bigg] = 0$$

(3.29)

On further simplification,

$$\left[\left[\frac{A_{ij}}{\Delta\tau}+\frac{K_{ij}}{2}-\frac{M_{ij}}{2}+\frac{Y_{ij}}{2}\right]+US_{ij}+CW_{ij}+LS_{ij}+BS_{ij}\right]\boldsymbol{u}_{\boldsymbol{i}}^{*}$$

$$=\left[\left[\frac{A_{ij}}{\Delta\tau}-\frac{K_{ij}}{2}+\frac{M_{ij}}{2}\right]-\frac{Y_{ij}}{2}\right]\boldsymbol{u}_{\boldsymbol{i}}^{\boldsymbol{\theta}}+[O_{ij}-B_{ij}]\,p_i^{\theta}+\chi_j \qquad \textbf{(3.30)}$$

An efficient algorithm can be constructed for Eq. (3.30) as follows:

- Pre-set sparsity pattern of all the matrices (since mesh is stationary).
- Preassemble the following matrix calculations: $(LS_{ij}+BS_{ij})$, $(O_{ij}-B_{ij})$, $\frac{A_{ij}}{\Delta\tau}$ and $\left(\frac{A_{ij}}{\Delta\tau}-\frac{K_{ij}}{2}+\frac{M_{ij}}{2}\right)$.

Time loop: t < total time:

- Assemble matrices: US_{ij}, CW_{ij} and $-\frac{Y_{ij}}{2}$ and vector χ_j.
- Evaluate $Y_{ij}\leftarrow-\frac{Y_{ij}}{2}+\left(\frac{A_{ij}}{\Delta\tau}-\frac{K_{ij}}{2}+\frac{M_{ij}}{2}\right)$

$$\lambda_j=Y_{ij}\boldsymbol{u}_{\boldsymbol{i}}^{\boldsymbol{\theta}}+[O_{ij}-B_{ij}]\,p_i^{\theta}+\chi_j$$

- Evaluate $Y_{ij}\leftarrow-Y_{ij}+2\frac{A_{ij}}{\Delta\tau}+CW_{ij}+US_{ij}+\left(LS_{ij}+BS_{ij}\right)$
- Solve $Y_{ij}\boldsymbol{u}_{\boldsymbol{i}}^{*}=\lambda_j$ using BICGSTAB with a jacobi preconditioner and appropriate boundary conditions.
- Solve Eq. (3.25), (3.26) and (3.23) and repeat.

The above optimized algorithm is implemented in FEniCS, and it is highly efficient. The matrices that depend on the evolving solution (Eq. (3.28), 1st step in time loop) are assembled using the naive approach in FEniCS. This is very straight-forward, since one can use the lhs( ) and rhs( ) functions in conjunction with the assemble( ) function call. Note that we use a constant time step $\Delta\tau$ for our solver. A similar optimization is used for the weak form of energy equation Eq. (3.23)

and is not explained here for the sake of simplicity. FEniCS makes it easy to implement the above optimized algorithm in a few lines of high-level python code. The accuracy and convergence of our algorithm has been well tested, and a few benchmark results are presented in the next section. The code's performance shows good strong scalability up to 84 cores on Michigan State University's - AMD EPYC 7H12 cluster.

3.3 Benchmarks

3.3.1 2D Lid-driven cavity

We solve the standard 2D lid-driven cavity case for $Re = 1000$ and $Re = 2500$. Our domain size is (1,1) and the mesh resolution and time step are 200 x 200 and 0.005 respectively. The mesh is uniform throughout and is composed of triangular elements. The top boundary has a velocity $U = 1$ in the positive x-direction and the other boundaries are no-slip. We use wall boundary condition for pressure at all the four walls. For this setup, we do not solve for the energy equation. **Figure 3.1** shows the center line velocity (U_x, U_y) for $Re = 1000$ and 2500 respectively. These are an excellent match with Botella and Peyret [64].

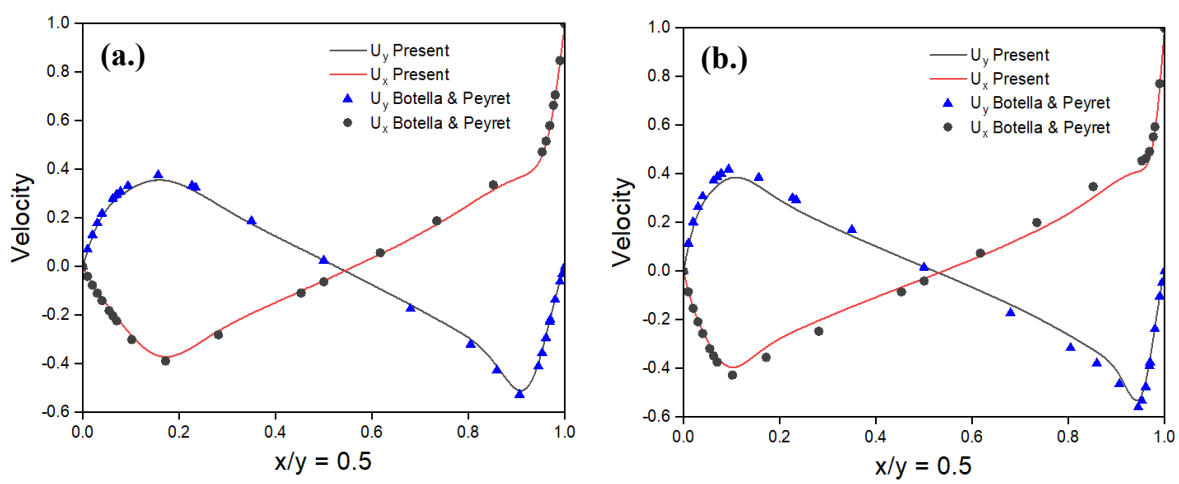

Figure 3.1: A comparison of velocity profiles at the center line of the domain with Botella and Peyret [64] for (a) Re = 1000 (b) Re = 2500.

3.3.2 3D flow over a heated sphere

We simulate flow over a 3D heated sphere in a cylindrical domain with dimensions: $L = 31$ and $r = 10$. The Re is defined with respect to the sphere diameter and inlet velocity and is set as $Re = 100, 300$ and 1000. The size of the cylindrical domain is considered such that it has negligible effect on the simulation results. We run all the Re cases for $Pr = 0.7$ and compare our results with Gao et al. [65] and Rodriguez et al. [66]. Note that we do not consider backflow stabilization for this flow problem.

The boundary conditions at the inflow and circumferential boundaries are uniform velocity $U = (1, 0, 0)$ and at the sphere surface is no-slip. An outflow boundary condition is applied at the outlet. For temperature: at the inflow $\mathrm{T} = 0$, at the sphere surface $\mathrm{T} = 1$ and for the outlet and circumferential boundaries a Neumann condition is used. We use a non-uniform tetrahedral mesh which is fine at the sphere surface, and it gets coarse as we move downstream and at the far circumferential boundaries. The time step is calculated such that the maximum runtime courant number stays below 0.3. A comparison of our drag coefficient and Nusselt number, with the existing literature is given in **Table 3.2**. Our results showcase a good match at all the Re.

Table 3.2: Drag coefficient and Nusselt number at Re = 100, 300, 1000 and Pr = 0.7.

Re	C_d		Nu (Pr = 0.7)	
	present	Benchmark [65, 66]	present	Benchmark [66]
100	1.1	1.09	6.91	-
300	0.667	0.659	10.84	10.63
1000	0.468	0.466	17.78	17.4

Next, we also perform a convergence study for the case of $Re = 1000$, and the code is run for a range of mesh sizes. **Table 3.3** shows order of convergence and error norm with respect to the benchmark values for all the tetrahedral mesh sizes simulated. Here h is the average cell size of

the elements close to the sphere and the order of convergence $\boldsymbol{p}$, is calculated as $ln\left(\frac{E_1}{E_2}\right)/ln\left(\frac{h_1}{h_2}\right)$.

We observe that our solver achieves fourth and second order accuracy for velocity and temperature respectively.

Table 3.3: Error norm and order of convergence for different mesh sizes. The velocity and temperature are both quadratic (P2).

h	C_d error	**p**	Nu error	**p**
3.75×10^{-2}	1.61×10^{-1}	---	1.23×10^{-1}	---
3×10^{-2}	6.83×10^{-2}	3.84	8.32×10^{-2}	1.75
2.25×10^{-2}	2.18×10^{-2}	3.97	4.83×10^{-2}	1.89
1.5×10^{-2}	4.30×10^{-3}	4.00	2.18×10^{-2}	1.96

CHAPTER 4 : PATIENT SPECIFIC COMPUTATIONAL MODELING OF CRYOBALLOON ABLATION FOR PULMONARY VEIN ISOLATION

4.1 Patient specific geometry and physical setup

To investigate CBA, simulations are performed on a patient specific LA geometry reconstructed from the computed tomographic images of a 54-year-old female patient (Patient III in [67]) suffering from paroxysmal atrial fibrillation (**Figure 4.1**). The model dimensions given in **Table 4.1**.

Table 4.1: LA patient specific model dimensions

	LSPV	LIPV	RSPV	RIPV	MV
Area (mm^2)	322.5	209.9	188.35	437.6	2061.1
Av. Dia. (mm)	20.26	16.34	15.48	23.6	51.24

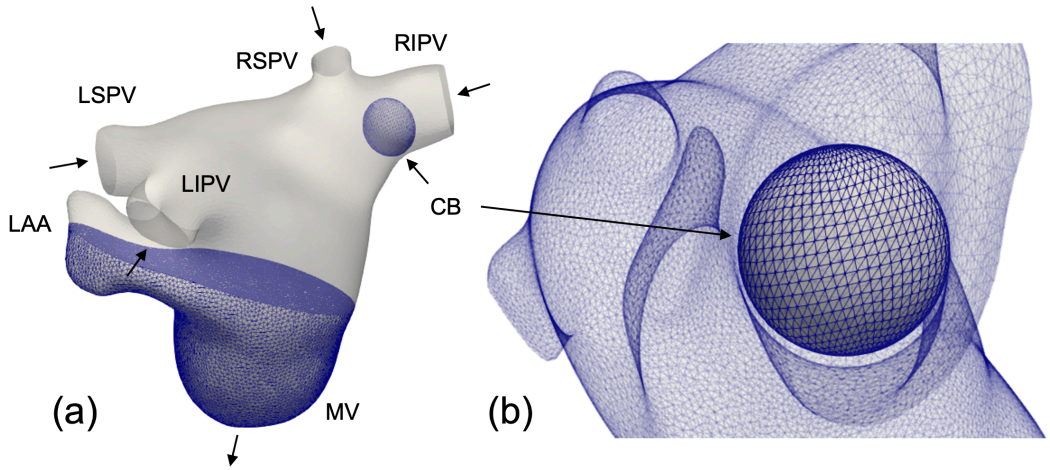

Figure 4.1: (a) Schematic of LA geometry with an ellipsoidal CB at the RIPV; the four PV's and MV serve as inlets and outlet for blood flow in the LA respectively. (b) Close up view of the CB; LAA: LA appendage.

Flow rate measurements (at a HR - 63 bpm) of 4 PVs acquired in healthy patients over 1 cardiac cycle [68] are imposed as boundary conditions at the 4 PV ostium serving as inlets to the LA (**Figure 4.2a**). No-slip boundary condition is applied on the LA walls. Pressure is set as 0 at the outlet of the LA (i.e., MV annulus), whereas for all the other boundaries, we prescribed $\frac{\partial p}{\partial \boldsymbol{n}} = 0$. A temperature of 37°C is prescribed as Dirichlet boundary condition at the four PV ostia. Neumann

boundary condition with zero heat flux (i.e., $\overline{\mathrm{T}}_{\omega} = 0$) is applied at the MV outlet. Initial temperature in the problem domain is set as 37°C. The diameter and velocity of the LSPV inlet is taken as the characteristic length and velocity scales, respectively (i.e., $L_{sc} = 20.26$ mm and $V_{sc} = 9.3\ cm/s$). The corresponding control parameters are $Re \approx 628$ and $Pr = 27$ for this problem. Values of the parameters are given in **Table 4.2**.

Table 4.2: Physical properties of blood at 37^{0}C.

Density (kg/m^3)	Kinematic viscosity (m^2/s)	Thermal conductivity (W/m^{0}C)	Specific heat (J/kg^{0}C)
1060	3×10^{-6}	0.52	3800

We consider two baseline cases, one without the CB (**LA-only** case) and the other with a CB to simulate CBA (**LA+CB** case). In the latter, the cryoballoon (Medtronic CB2) geometry is idealized as an ellipsoid with major/minor diameter = 23/19.85 mm located in the right inferior pulmonary vein (RIPV) [7]. The CB is prescribed a constant minimum temperature of -70°C. At the LA wall, we prescribed a temperature dependent heat source $Q = \beta(1 - \mathrm{T})$ to describe heat transfer due to blood perfusion in the LA wall. Here, β is the coefficient proportional to the perfusion rate and its quadratic dependence on non-dimensional temperature T is given as follows:

$$\beta = \begin{Bmatrix} 6.18\,\mathrm{T}^2 - 7.39\,\mathrm{T} + 2.21 & for\ \ \mathrm{T} \geq 0.725 \\ 0.1 & for\ \ \mathrm{T} < 0.725 \end{Bmatrix}. \quad \textbf{(4.1)}$$

The maximum value of β at 37°C is taken as 14% (mass fraction of LA) of the rate of coronary perfusion: 0.85 ml/gm/min. At lower temperatures, we assume that perfusion remains constant and reduces to 10% of its original baseline value. While the dependence of vascular perfusion β with temperature T is not patient-specific, the reduction of β with T should hold in general for every patient. Note that the non-dimensional temperature T used above is different from the actual

temperature T_{act}, which is dimensional. So, T_{act} lies between the temperature $T_{min} = -70°C$ and $T_{max} = 37°C$, and is given by:

$$T_{act} = T_{min} + (T_{max} - T_{min}) \cdot T. \quad \textbf{(4.2)}$$

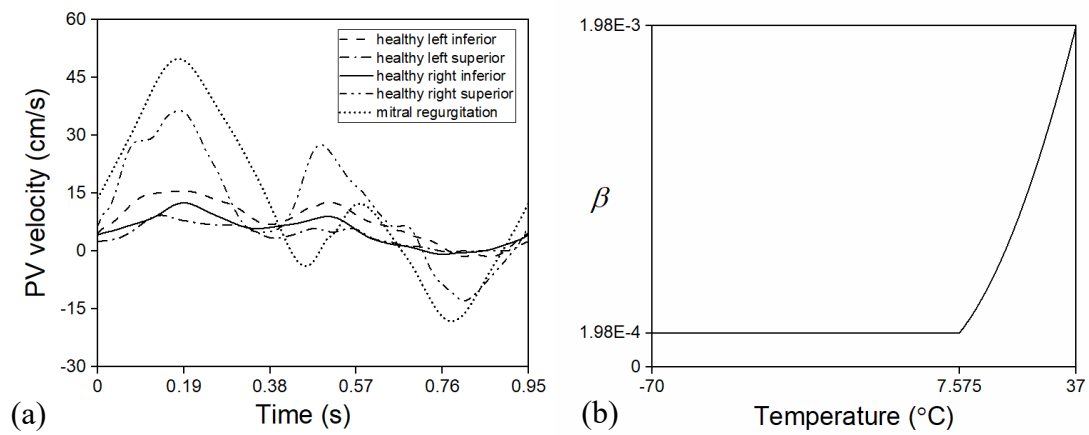

Figure 4.2: Prescribed patient-specific blood flow velocity waveforms for the (a) PV inlets and mitral regurgitation (MR) [69]. (b) Perfusion constant β as a function of temperature T_{act}.

The behavior of perfusion coefficient β given in Eq. (4.1) with actual temperature (T_{act}) is shown in **Figure 4.2b**. We note that a similar vascular perfusion-temperature relationship was also employed by Pennes [70] and Getman et. al. [7]. Moreover, we note that Eq. (3.3) reduces to the Pennes heat conduction equation [70] at the LA wall boundary because the conductivity and density of cardiac tissue is very close to that of blood (**Table 4.2**) and the velocity at the LA wall boundaries is zero here. Therefore, the inclusion of a heat source at the boundary is more physical than prescribing a Neumann boundary condition. We neglect metabolic heat generation of the cardiac tissue and assume it to be negligible when compared to the cryogenic heat flux across the CB.

4.2 Validation for Left Atrium hemodynamics

To validate the model for LA hemodynamics, a simulation was performed for 25 heart cycles in the **LA-only** case until the MV flow reaches periodic steady state. In this simulation, the FE tetrahedral mesh has 1.16 million degrees of freedom (DOF) and is graded so that it is fine close to the LA wall. The time step is selected so that the maximum Courant number throughout the simulation runtime is less than 0.12. Note that here, we do not solve for the energy Eq. (3.3).

For this case, **Figure 4.3a** shows the velocity field vectors at the MV plane. **Figure 4.3b** shows a comparison of our instantaneous MV velocity (averaged over the MV surface area given in **Table 4.1**) obtained from the **LA-only** simulation case with that measured from a healthy patient using spectral doppler echocardiography [71]. The simulated MV velocity waveform is close to the measurement, showing an E peak associated with the passive LV filling and an A peak associated with the contraction of the atria (i.e., atrial kick) during LV filling. The model-predicted E/A ratio and deceleration time (DT) are 1.4 and 240 ms, respectively. The greatest difference between the simulated MV velocity waveform and measurement occurs during systole (0.76 – 0.95ms), where the model predicted some (negative) backflow when the MV is closed during systole. This behavior is attributed to the pressure boundary conditions imposed and the assumption of a rigid LA wall geometry.

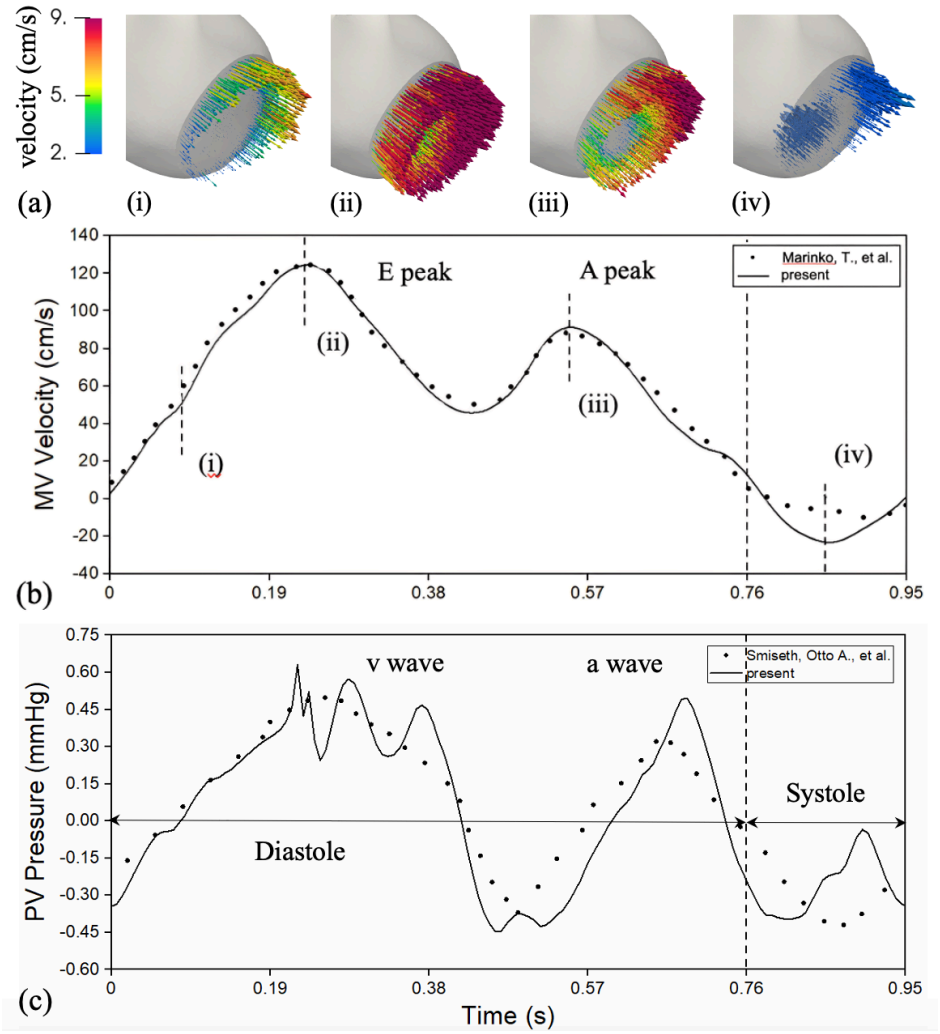

Figure 4.3: Simulation results from LA-only case (a) Velocity vectors at the MV plane. Vectors are scaled in (iv) to illustrate the presence of back/reverse flow. Comparison of (b) simulated MV velocity with measurements from spectral doppler images [71]; and (c) simulated pulmonary vein (PV) pressure waveforms with that measured based on the ultrasound transit time by Smiseth, Otto et. al. [72]. Note: measurements are scaled uniformly based on the prescribed heart rate, velocity, and pressure.

Figure 4.3c shows a comparison between the simulated PV pressure waveforms with that measured from a healthy patient based on the ultrasound transit time by Smiseth et. al. [72]. Similar to the MV velocity waveform, the simulated PV pressure waveform is consistent with the measurements during diastole. The largest difference between the model prediction and measurement occurs during systole, where the simulated result shows a slight pressure rise (small peak during systole in **Figure 4.3c**) that is not found in the measurements.

In **Table 4.3**, we compare the simulated results with the measurement, during diastole and systole using a standard Welch t-Test. We note that the mean for both MV velocity and PV pressure match more closely during diastole as compared to systole. The same is reflected by the higher p-values in case of diastole. Also, despite the deviation of simulated results during systole, the absolute t-value is still less than t-critical which implies a good statistical match.

Table 4.3: Comparison of MV velocity and PV pressure for LA-only case during diastole and systole using a standard Welch t-Test. Note: for p-values < 0.05, the comparison is statistically different.

		Mean	Variance	t-value (t-critical)	p-value
MV velocity					
Diastole	present	70.81	1000.79	0.285 (1.98)	0.78
	measured [71]	68.83	1098.01		
Systole	present	-10.45	97.59	-2.05 (2.14)	0.06
	measured [71]	-4.19	8.18		
PV pressure					
Diastole	present	0.16	0.09	0.43 (1.99)	0.67
	measured [72]	0.13	0.06		
Systole	Present	-0.23	0.013	1.67 (2.16)	0.11
	measured [72]	-0.31	0.010		

4.3 LA+CB results

Having validated the model, CBA was simulated in the **LA+CB** case where the CB was positioned at a fixed position that is close to the lower end of the RIPV ostium (**Figure 4.1b**). The simulation was performed for 40 heart cycles until a periodic steady state for the power absorbed across the

CB is obtained. Because a graded fine mesh is necessary to resolve the gap between the CB and the LV wall, the FE tetrahedral mesh for this case has 1.88 million DOF (**Figure 4.1b**). The time step for this simulation case is selected such that the maximum Courant number throughout the runtime is less than 0.25. The computing wall time is approximately 46 hours for simulation of the **LA+CB** case.

Figure 4.4a shows the temperature distribution at the RIPV ostium after periodic steady state is reached after about 20 seconds in the **LA+CB** case. By defining the scar/lesion as the region in the LA wall with temperature less than -20^0C [21], the model predicted the lesion area $\boldsymbol{A_l}$ to be $296.63\,mm^2$. The maximum lesion width along the flow direction is 6.8 mm. To determine a relationship between the local gap g (defined as the minimum distance to the LA wall for any given point on the CB) and the local LA wall temperature T, we project both the local g and T onto the CB surface ($\partial\Omega_{CB}$) (**Figure 4.4b and c**) and plot out their distribution (i.e., $g - T$ distribution) in **Figure 4.4d**. The distribution shows that the temperature plateau when the gap g is above approximately 6mm. On the other hand, the gap threshold at which the temperature falls below that associated with the lesion formation (i.e., -20^0C) is approximately 2.5 mm. By defining the balloon-tissue contact area A_c as the area where the gap $g < 3$mm and an efficiency factor $\boldsymbol{e}$ given by the ratio between A_l and A_c, we find that $A_c = 422.8\,\text{mm}^2$ and $\boldsymbol{e} = 0.7$ in the **LA+CB** case. **Figure 4.4e** shows the rate of energy transfer (power) across the CB with time. At the initiation of CBA, the power is high, then it drops and flattens after about 24s once a periodic steady state is reached. The average value of power absorbed across the CB is approximately 94 W in the ***LA+CB*** case.

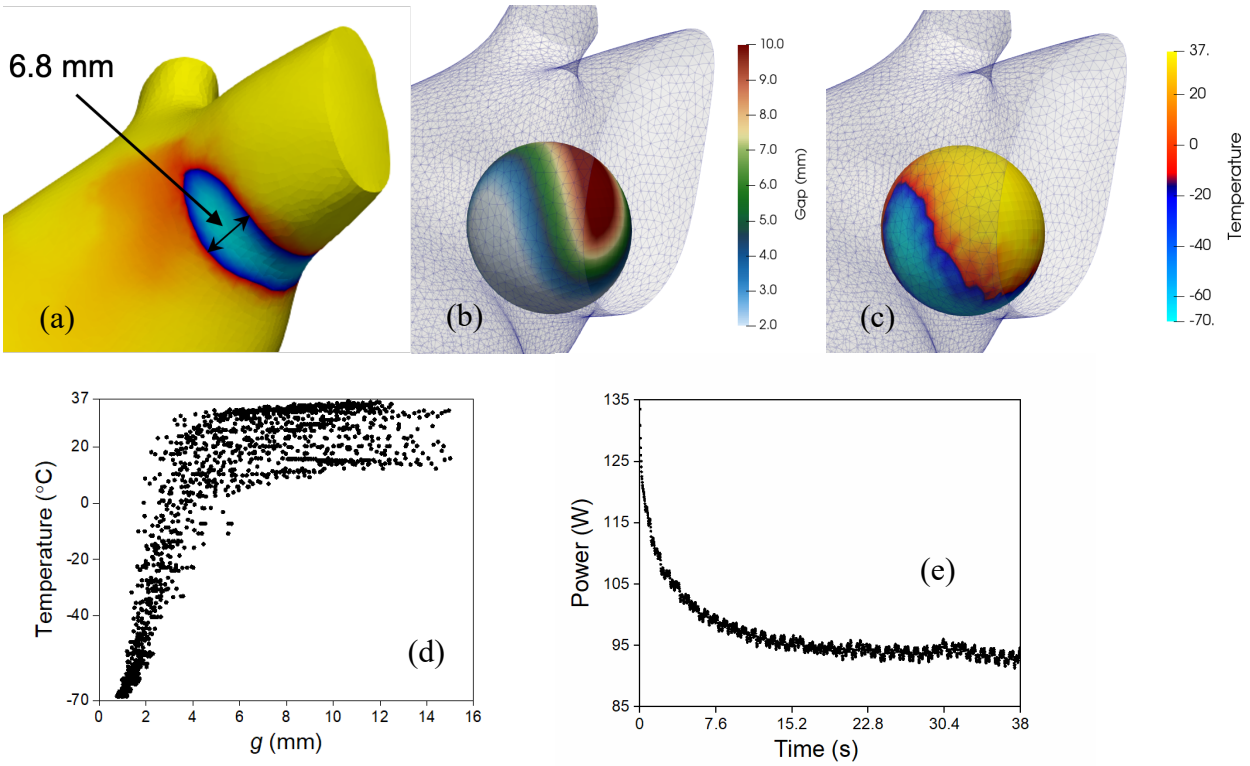

Figure 4.4: Simulation results from LA+CB case (a) Periodic steady state temperature profile at the RIPV ostium. (b) Gap distribution projected onto the CB. (c) Temperature at neighboring RIPV ostium wall (corresponding to the gap) projected onto the CB. (d) Plot for local temperature vs local gap distribution; extracted from Figure 4.4c and 4.4b respectively. (e) Power absorbed across the CB vs time during the cryoablation therapy.

Using the **LA+CB** case as baseline, we then perform simulations with:

1) 12 distinct CB positions where the RIPV ostium is not completely occluded,

2) 4 different PV blood velocities (for healthy patients) at 120%, 90%, 75% and 60% of that given in the **Figure 4.2a.** This flow rate measurement is applied to all inlets associated with the PVs.

3) Flow reversal at the MV to investigate whether mitral regurgitation (MR) affects the formation of lesion in CBA. To simulate flow reversal at the MV, we use the blood flow rate measurements in a PV of a patient suffering from severe MR [69] (**Figure 4.2a**).

Similar temperature boundary conditions are applied in all the simulations.

4.3.1 Effects of CB position on lesion formation

Figure 4.5 shows the local $g - T$ distribution for 3 CB positions out of the 12 simulated CB positions. Here P1, P8 and P11 are notably different in terms of the CB position and therefore, form lesions at different locations at the RIPV ostium. The results show that the temperature fall below that for lesion formation (-20^0C) at locations where the gap is less than 2.5 mm at the RIPV ostium irrespective of the CB position. At locations with $g > 5 - 6$ mm, the temperature plateaued at approximately the blood temperature (37^0C).

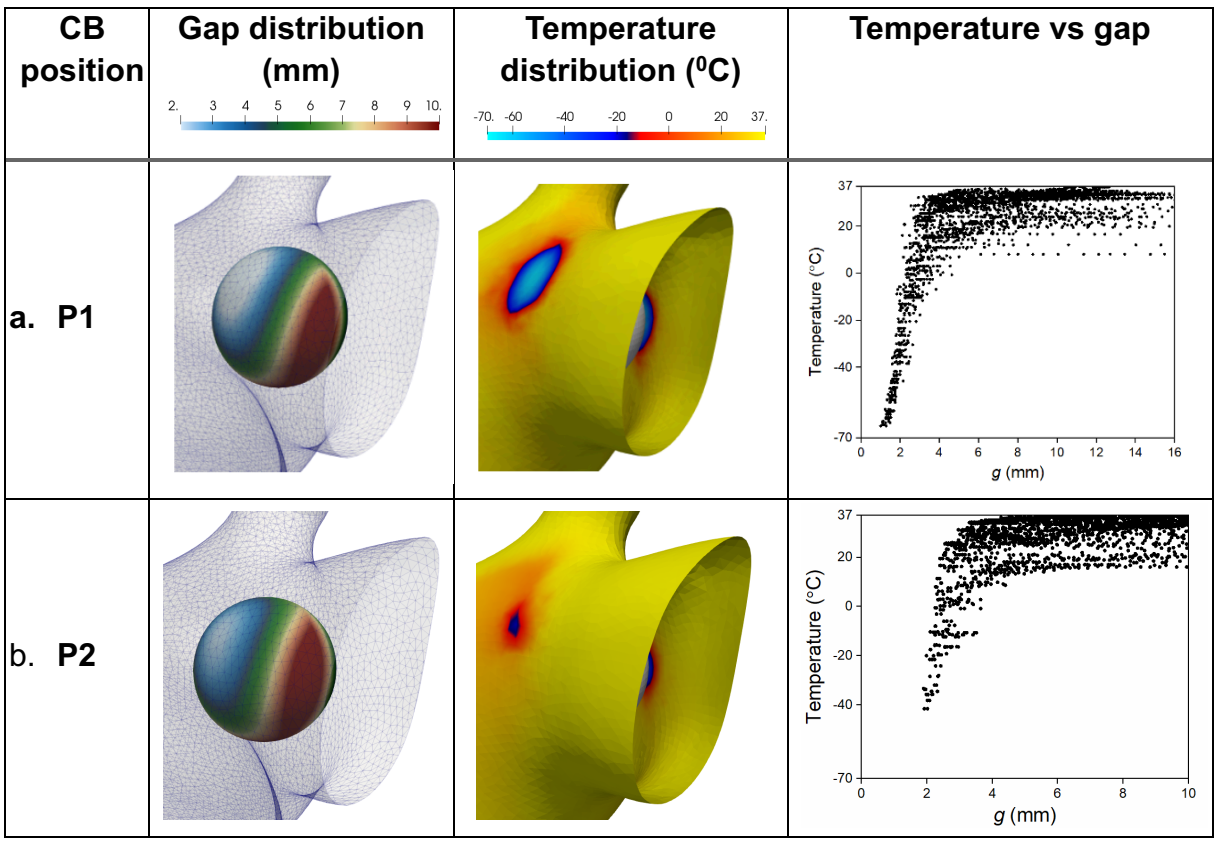

Figure 4.5: CB position, P1 to P11 characterized by gap distribution (projected on the CB) and its effect on corresponding temperature distribution at RIPV ostium. The lesion area A_l, is the total area on either side of the RIPV ostium where $T_{act} < -20^0$C. Also, a plot for local temperature vs gap shows that lesion formation starts if g < 2.5 mm.

Figure 4.5 (cont'd)

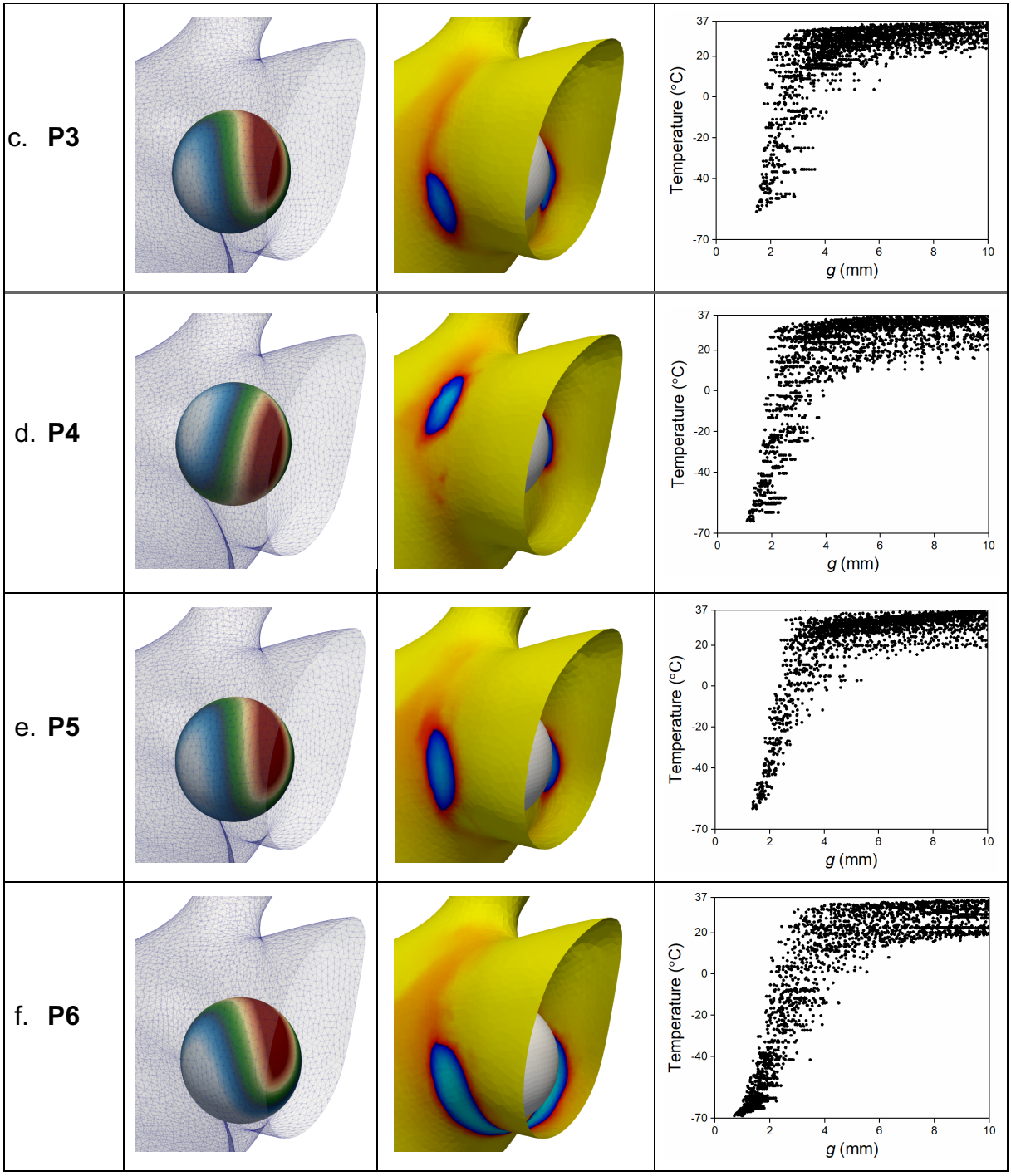

Figure 4.5 (cont'd)

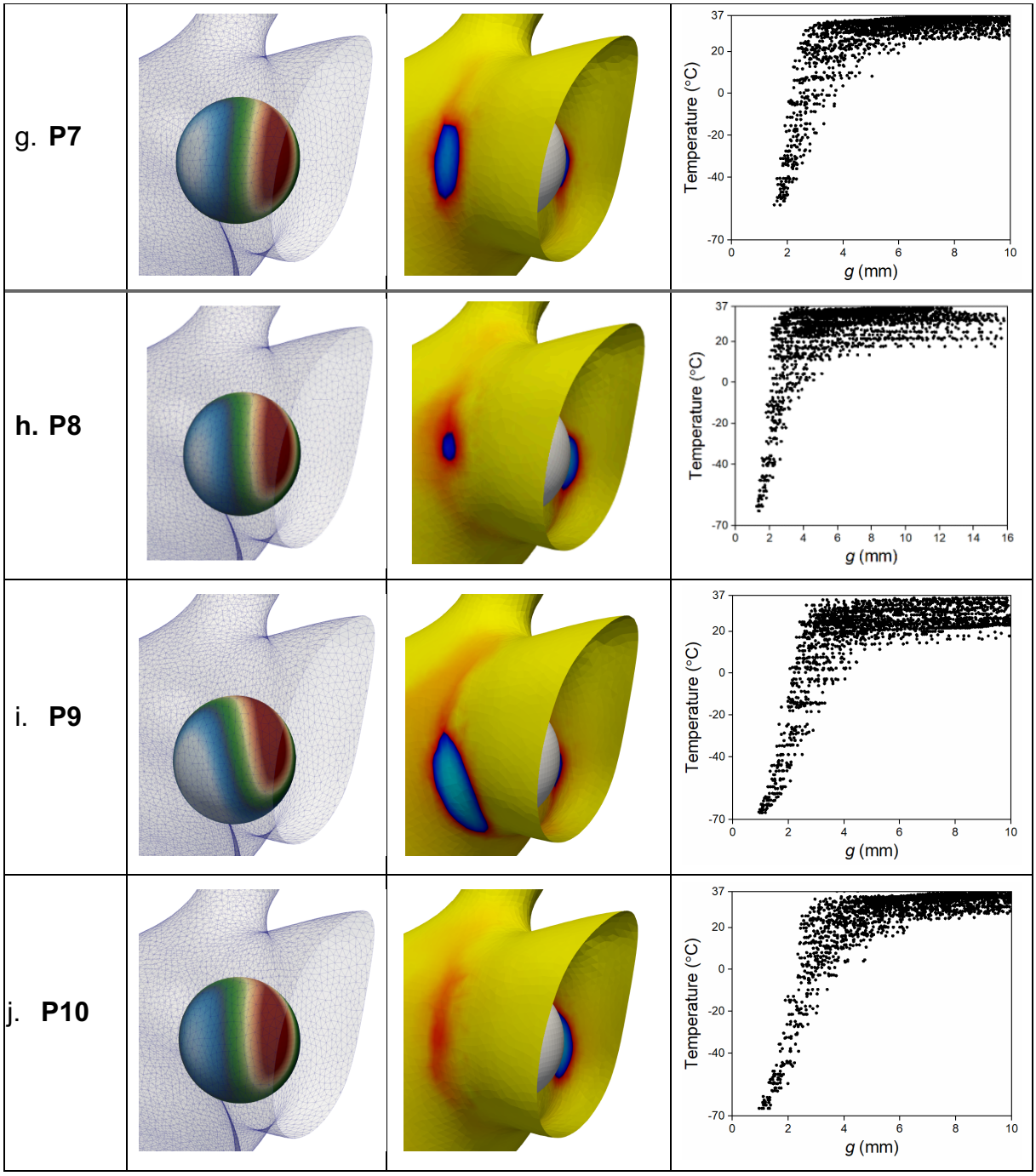

Figure 4.5 (cont'd)

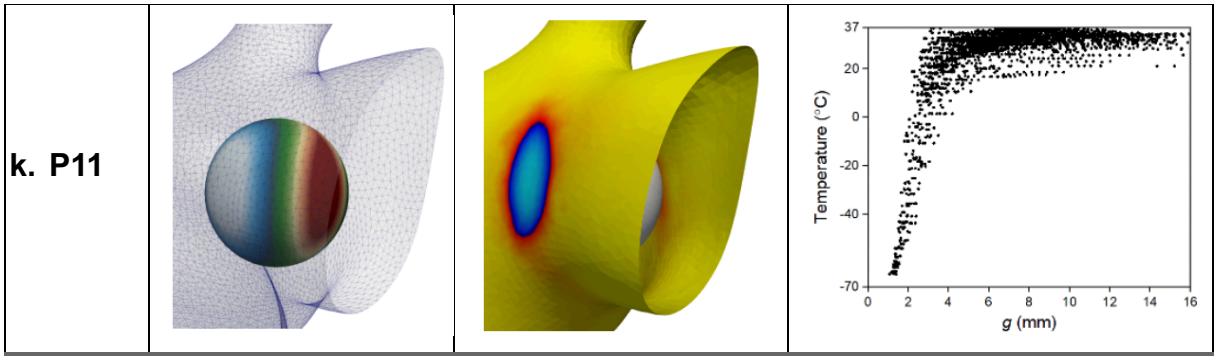

Figure 4.6 shows the efficiency factor $\boldsymbol{e}$ and average rate of energy transfer (power absorbed in watts) across the CB for all the 12 simulated positions arranged with increasing $\boldsymbol{A_c}$. The figure shows that the lesion area $\boldsymbol{A_l}$ and $\boldsymbol{e}$ are increased with increasing $\boldsymbol{A_c}$. There is also considerable decrease in power as the efficiency factor increases. These results imply that less energy per unit area is required to maintain the low nadir temperatures of the CB if it is positioned in such a manner that $\boldsymbol{A_c}$ is high.

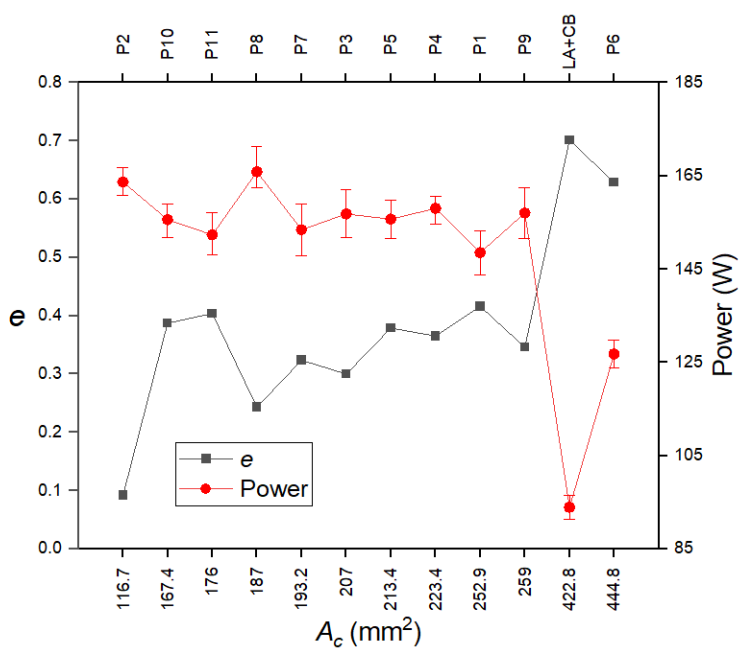

Figure 4.6: Efficiency factor $\boldsymbol{e}$ and average power absorbed across the CB surface (watts) with increasing balloon-tissue contact area $\boldsymbol{A_c}$ for 12 different CB positions in the RIPV ostium.

Figure 4.7 shows the scatter plot between the LA wall temperature T_{act} (see Eq. (4.3)) with the local gap g and the balloon-tissue contact area $\boldsymbol{A_c}$. The results show that the LA wall temperature is less sensitive to $\boldsymbol{A_c}$ than g. The non-linear relationship between T_{act} with g and $\boldsymbol{A_c}$ can be described by the fitted equation:

$$T_{act} = \left(\frac{-4268.6 + 1971.47g - 4.82A_c}{3666.34g + 2.44A_c}\right) \cdot 10^2 \quad for\ \ 1.25\ mm \leq g \leq 9.75\ mm \ \ \& \ 115\ mm^2 \leq A_c \leq 450\ mm^2. \tag{4.3}$$

The coefficient of determination (R^2) for the fit is 0.79.

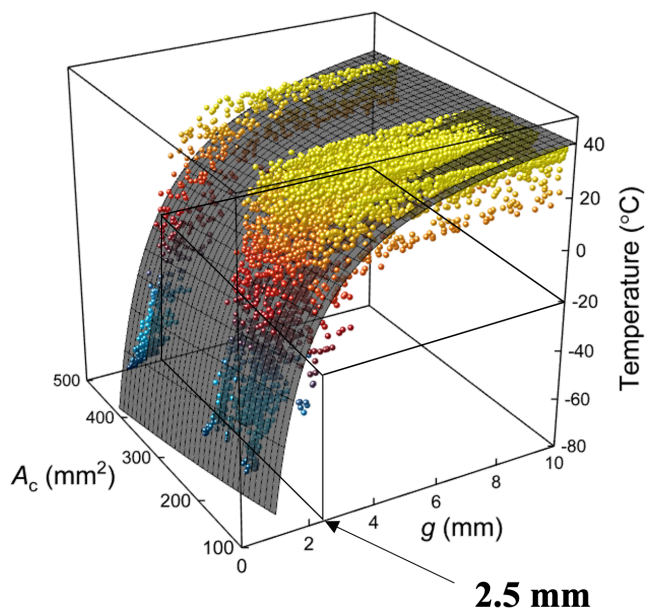

Figure 4.7: Non-linear regression fit of the LA wall temperature T_{act} with the local gap g and the balloon-tissue contact area A_c.

4.3.2 Effects of PV blood velocity and mitral regurgitation on lesion formation

Compared to the position of CB, the effects of PV blood velocity and MR are less substantial. Specifically, when the velocity magnitude in the PV is halved, the power absorbed changes by only up to 6%, and the efficiency factor e (or lesion area $\boldsymbol{A_l}$) increases anywhere between 5% to 38% depending on the CB position. The changes in e and $\boldsymbol{A_l}$ are larger when $\boldsymbol{A_c}$ is low. For

example, in the case of P3 (**Figure 4.8**), power drops only by 3.2% and $\boldsymbol{A_l}$ increases by only 8.5% when the velocity magnitude is halved. In the presence of MR, we observe only a 2% change in lesion size and approximately 3.9% change in power absorbed across the CB.

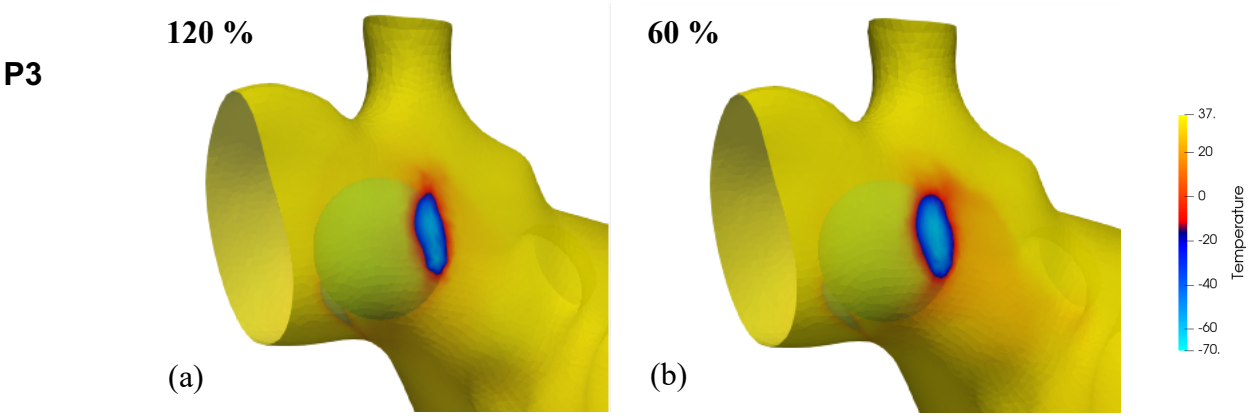

Figure 4.8: Temperature distribution and lesion area A_l at the RIPV ostium for CB position P3 when the PV blood flow velocity magnitude is (a) 120 % (b) 60 % of that given in Figure 4.2a.

4.4 Discussion

We investigated 3 main factors that can affect the lesion size for CBA, i.e., catheter balloon position, PV blood velocity and MR. By prescribing PV inlet flow boundary conditions measured in human subjects, our model can predict MV velocity profile that agrees with those measured in humans [71], especially during diastole. The framework also predicts both the E and A peak (E/A ratio = 1.4) and the deceleration time (DT = 240 ms) that are within the normal value range (0.75 < E/A < 1.5 and DT > 140 ms) found in healthy subjects.

4.4.1 Patient-specific model calibration for cryoablation simulations

To simulate patient-specific cryotherapy, the governing parameters in Eq. (3.1-3.8) can be made patient-specific by prescribing the Re, Pr, flow and temperature boundary conditions based on measurements obtained in each patient. Specifically, the blood flow velocity at the PV ostia can be measured using doppler ultrasound or patient specific segmentation can be performed using a

4D (3D - time resolved) MRI or CT imaging scan [68]. Moreover, the pressure and temperature in the LA can in principle, be measured using a catheter. In previous studies, Gonzalo et al. [73] and Morales et al. [74] have used in-vivo 4D-CT and 4D-MRI measurements to segment and resolve the LA geometry and flow for their hemodynamic simulations. Using such imaging, it can be very difficult to precisely model areas with less flow for e.g., the LAA, however, to measure flow velocities at major vessels like the PV's, a set of planes perpendicular to the vessel of interest can be placed to extract the normal projection of the velocity. For medical imaging and segmentation one can utilize tools like MeVisLab [75] or 3D Slicer [76]. Thereafter, these images can also be used to calculate the average diameter of the PV's and MV.

4.4.2 Factors affecting cryoablation

Of the 3 factors that may affect the effectiveness of CBA, namely, CB position, PV inlets velocity and MR, we found that the lesion size is most sensitive to CB position, which in turn, depends on the patient specific PV anatomy. The lesion size depends directly on the size of the balloon-tissue contact area (defined as per area where gap less than 3 mm) and the power absorbed by the CB. Correspondingly, we find that a gap less than 2.5 mm is a threshold to produce a temperature $< -20^0\mathrm{C}$ for lesion formation. This supports the hypothesis that lesion formation is largely influenced by balloon-tissue contact area A_c and time of freeze [10].

In the **LA+CB case** that has a high A_c, the average lesion width of 6.8 mm for a 10 cm/s RIPV velocity is comparable with experimental observations in a porcine heart by Parvez et. al. [77]. Their experiments revealed similar lesion widths for a vertical orientation of the CB at 6g contact force and 40 cm/s PV velocity. They noted an increase in lesion size with a decrease in PV velocity. For instance, lesion width increased by 39% from 10.4 mm to 14.5 mm when the PV velocity is

halved (from 40 cm/s to 20 cm/s). The same quantitative behavior is observed for other CB positions simulated in **Figure 4.5**.

4.4.3 Comparison with previous models of cryoablation and LA flow

Computational models simulating cryoablation in the LA are limited. Most models [7, 21, 78] are based on idealized geometries as well as boundary conditions and consider only heat transfer across the LA wall to analyze transmural lesion depth. Specifically, a numerical simulation performed by Mussig et al. [78] with the electromagnetic and thermal simulation software CST Darmstadt, ignores the convective effects of blood flow in the PV antrum. The CB geometry in that model, however, is realistic and replicates the model used by Medtronic [7]. For a 23 mm CB set at -50^0C, they observed that after 20 sec of ablation the temperature is -50^0C and -24^0C at a depth of 0.5 and 1 mm respectively. Instead of prescribing a constant CB temperature, Xia et al. [21] applied a constant heat source of -30 W to the CB contact area of a segment of the PV antrum wall. They predicted about 45 s for the temperature to reach -20^0C at a depth of 0.5 mm. Also, at depths greater than 2 mm, temperature remains between -20^0C to -40^0C and does not drop below -40^0C, suggesting that the hold time becomes important for lesion/scar formation at that depth.

To the best of our knowledge, there is only one computational model [7] that consider blood flow and heat transfer in modeling CBA. That model, however, is based on an idealized geometry of a PV ostium and does not consider flow leakage across the CB. In that study, Getman et al. [7] modelled the temperature at various tissue depths using the non-isothermal flow module in COMSOL. They considered balloon-tissue interactions with the CB set at -60^0C. For tissue depths less than 0.5 mm, they predicted TTI ($T_{act} < 23^0$C) and the time required for optimal lesion ($T_{act} < -20^0$C) are 33 and 40 s, respectively. In comparison to these studies, we predicted a shorter time of (~ 24 sec) to steady state lesion formation in the ***LA+CB*** case. This difference may

be due to **(a)** A lower nadir CB temperature; **(b)** Zero thickness of the PV ostium; **(c)** Difference in boundary conditions at the PV inlets. We also note that accounting for LA wall thickness will increase the power absorbed by the CB and increase the duration required for lesion formation.

For power absorbed by the CB during the procedure, the model predicted it to be approximately 165 W for most of the CB positions considered (**Figure 4.6**). This value is approximately half of that measured in a recent study that uses an ex-vivo calorimetric assessment on a bovine epicardial tissue [79]. That study revealed an ablative power of more than 325 W to generate 2 cm diameter freeze zones in less than 120 s. In the case of excess leakage past the CB, our model also predicts a substantial increase in the ablative power. Similar behavior was observed by Ghosh and McGuire [80]. Data for cooling power, however, is typically not reported for cryoablation probes. Nevertheless, most nitrous oxide probes provide approximately 8.5 W/mm with a range of 50 – 130 W depending on the size/length of the cryocatheter [79, 81]. We also note that for a fixed ablative power, nadir CB temperatures is inversely proportional to the amount of leakage past the CB.

4.5 Conclusion

We show that our framework can reproduce measurements of MV velocity and PV pressure waveforms measured in human subjects. The modeling framework predicts that the effectiveness of cryoablation therapy based on the lesion size is very sensitive to the CB position, while PV inlet velocity and MR do not have substantial effects on the lesion size. The framework can be applied in future clinical investigations to optimize cryoablation therapy in patient specific LA-PV geometries.

CHAPTER 5 : DEVELOPMENT OF IMMERSED BOUNDARY BASED FICTITIOUS DOMAIN METHOD FOR MODELING OF FLUID STRUCTURE INTERACTIONS WITH HEAT TRANSFER

5.1 Methods

5.1.1 Mathematical formulation

We use a Eulerian description to solve for the fluid motion and a Lagrangian description to solve for the solid deformation and stresses in the Immersed Boundary (IB) method. We employ the governing continuity, momentum, and energy equations along with kinematic constraints in each domain. A schematic is given in **Figure 5.1** and the corresponding boundary value problem (b.v.p.) in dimensional form is given as:

For the fluid domain: $\Omega/P(t)$

$$\rho_f \frac{d\boldsymbol{u_f}}{dt} = \nabla.\boldsymbol{\sigma_f} + \rho_f \boldsymbol{f_f} \tag{5.1}$$

$$\nabla.\boldsymbol{u}_f = \boldsymbol{0} \tag{5.2}$$

$$\rho_f C_{pf} \frac{d\mathrm{T}_f}{dt} = k_f \nabla^2 \mathrm{T}_f + Q_{\boldsymbol{f}} \tag{5.3}$$

For the solid domain: $P(t)$

$$\rho_s \frac{d\boldsymbol{u_s}}{dt} = \nabla.\boldsymbol{\sigma_s} + \rho_s \boldsymbol{f_s} \tag{5.4}$$

$$J\rho_s = \rho_0 \tag{5.5}$$

$$\rho_s C_{ps} \frac{d\mathrm{T}_{\boldsymbol{s}}}{dt} = k_s \nabla^2 \mathrm{T}_s + Q_{\boldsymbol{s}} \tag{5.6}$$

$$\boldsymbol{\sigma_s} = -p_s \boldsymbol{I} + G(\boldsymbol{B} - \boldsymbol{I}) \tag{5.7}$$

Kinematic constraints: $P(t)$

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{u_s} \tag{5.8}$$

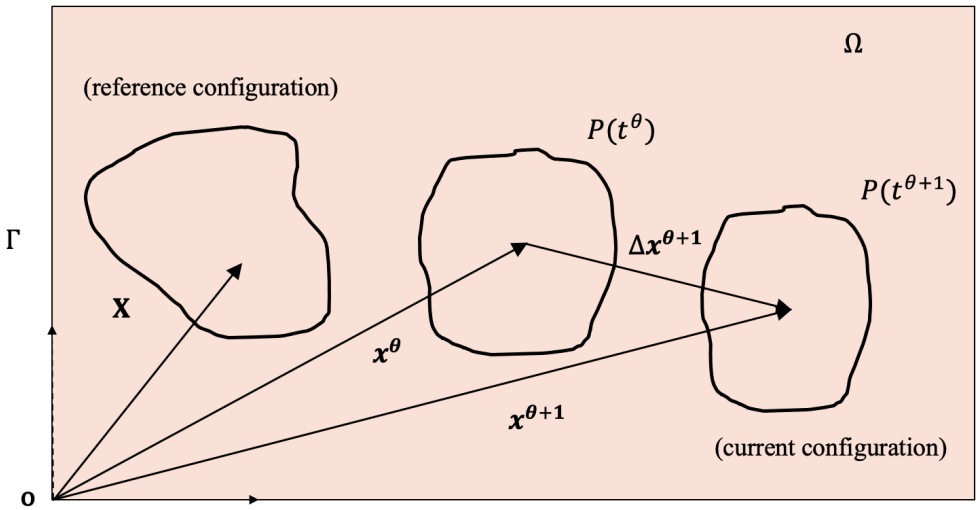

Figure 5.1: A schematic showing the FSI domain and respective boundaries along with the reference and current configuration for the solid.

In the above equations we use the constitutive law for a Newtonian fluid (Eq. (3.4)) and a hyper-elastic solid (Eq. (5.6)) and Ω is the overall fluid domain including the fictitious fluid and $P(t)$ is the solid current configuration. In Eqs. (5.1-5.3), $\boldsymbol{u_f}, p_f, \mathrm{T}_{\boldsymbol{f}}, \rho_f$, $\boldsymbol{f}_f$, $\boldsymbol{\sigma_f}$ are the fluid velocity, pressure, temperature, density, body force and stress tensor respectively. In Eqs. (5.4-5.7), $\boldsymbol{u_s}, p_s, \mathrm{T}_{\boldsymbol{s}}, \rho_s, \boldsymbol{f}_s$, $\boldsymbol{\sigma_s}$ denotes the solid velocity, pressure, temperature, density, body force and Cauchy stress tensor at the current configuration $\boldsymbol{x^{\theta+1}}$. Also, J is the determinant of deformation (gradient) tensor F which describes the material deformation with respect to the reference configuration and is given as $F = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \boldsymbol{\nabla x}$. In the solid constitutive law Eq. (5.7), $\boldsymbol{B} = \boldsymbol{F} \cdot \boldsymbol{F^T}$ is the finger tensor and $\boldsymbol{G}$ is the shear modulus of the solid material. Additionally, in the energy Eqs. (5.3) and (5.6), k_f, k_s are the thermal conductivity, C_{pf}, C_{ps} are the specific heats, $Q_{\boldsymbol{f}}$, $Q_{\boldsymbol{s}}$ are the heat sources for the fluid and solid domain respectively.

On the boundaries:

$$\boldsymbol{u_f} = \boldsymbol{u_\Gamma}\,; \quad \mathrm{T}_{\boldsymbol{f}} = \mathrm{T}_{\boldsymbol{\Gamma}} \qquad on \;\; \boldsymbol{\Gamma} \tag{5.9}$$

On the FSI interface:

$$\boldsymbol{u_f} = \boldsymbol{u}_s\,; \qquad \boldsymbol{\sigma_s}.\boldsymbol{n} = \boldsymbol{\sigma_f}.\boldsymbol{n} + \boldsymbol{F^e}\,; \qquad \mathrm{T}_{\boldsymbol{f}} = \mathrm{T}_{\boldsymbol{s}}\,; \qquad k_f \frac{\partial \mathrm{T}_f}{\partial \boldsymbol{n}} = k_s \frac{\partial \mathrm{T}_s}{\partial \boldsymbol{n}} \quad on \;\; \partial P \tag{5.10}$$

Here, $\boldsymbol{n}$ denotes the vector normal to the FSI interface and $\boldsymbol{F^e}$ is the non-hydrodynamic forces at the fluid-solid interface, like external contact forces. Note that in Eq. (5.5), ρ_0 is the solid density in the reference configuration and if the solid is incompressible i.e., $J = 1$, we also need to compute the solid pressure p_s. For a compressible hyper-elastic solid, however, we can replace p_s by $\frac{\lambda_0 lnJ}{J}$, where λ_0 is related to the bulk compressibility of the solid, and compute only the solid deformation. In our model, we implement both the incompressible and compressible formulations for the solid constitutive law. We also note that at high values of $\lambda_0 \sim 10^2 G$, the solid becomes close to incompressible ($J \sim 1$).

5.1.2 Finite element discretization

The FEM is used for spatial discretization of the governing equations in both the fluid and solid domain. Let $\boldsymbol{v}$, $\boldsymbol{\omega}$ be the test functions for the momentum, q be the test function for the continuity, and γ, γ_s be the test functions for the energy equations - for the fluid and solid domains respectively; then the Galerkin weak formulation is given as:

Momentum:

$$\langle \rho_f \frac{d\boldsymbol{u_f}}{dt} - \rho_f \boldsymbol{f_f}\,, \boldsymbol{v}\rangle_{\Omega/P} + \langle \boldsymbol{\sigma_f}\,, \nabla \boldsymbol{v}\rangle_{\Omega/P} + \langle \rho_s \frac{d\boldsymbol{u_s}}{dt} - \rho_s \boldsymbol{f_s}\,, \boldsymbol{\omega}\rangle_P + \langle \boldsymbol{\sigma_s}\,, \nabla \boldsymbol{\omega}\rangle_P = \int_{\partial P} \boldsymbol{F^e} \cdot \boldsymbol{\omega}\, \boldsymbol{dx} \tag{5.11}$$

Continuity:

$$\langle q\,, \nabla \cdot \boldsymbol{u_f}\rangle_{\Omega/P} = 0 \tag{5.12}$$

Energy:

$$\langle \rho_f C_{pf} \frac{d\mathrm{T}_f}{dt} - Q_f\,, \gamma\rangle_{\Omega/P} + \langle k_f \nabla \mathrm{T}_f, \nabla \gamma\rangle_{\Omega/P} + \langle \rho_s C_s \frac{d\mathrm{T}_s}{dt} - Q_s\,, \gamma_s\rangle_P + \langle k_s \nabla \mathrm{T}_s, \nabla \gamma_s\rangle_P = 0 \tag{5.13}$$

where $\{\boldsymbol{u_f}, \boldsymbol{u_s}, \mathrm{T_f}, \mathrm{T_s}, p_s, p_f \in H^1(\Omega)\}$ and $\{\boldsymbol{v}, q, \boldsymbol{\omega}, \gamma, \gamma_s \in H^1(\Omega)\}$. In the above equations, $\langle \cdot , \cdot \rangle$ denotes the Euclidean inner product of two functions in their respective domains.

Fictitious Domain (FD) / Distributed Lagrange Multiplier (DLM) formulation

Next, using the fictitious domain (FD) formulation (initially proposed by Yu [33]), we extend the fluid domain into the immersed solid and assume that the interior of the solid is filled with a fictitious fluid (see **Figure 5.2**). This fictitious fluid is constrained to move with the same velocity as the solid by enforcing a distributed Lagrange multiplier λ which acts as a pseudo volumetric force. For coupling between both the domains we use the Dirac-delta function kernels to calculate the pseudo forces.

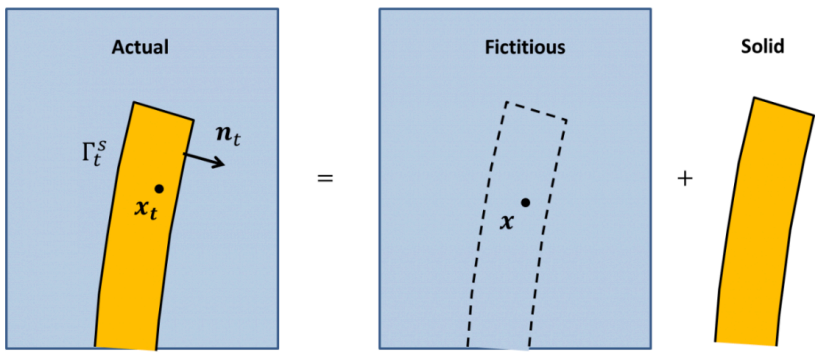

Figure 5.2: Fictitious domain formulation where the fluid is extended into the solid domain and $\boldsymbol{u_f} = \boldsymbol{u_s}$ in the overlapping region.

Using the FD/DLM formulation, we get,

Fluid:

$$\langle \rho_f \frac{d\boldsymbol{u_f}}{dt} - \rho_f \boldsymbol{f_f}, \boldsymbol{v} \rangle_\Omega + \langle \boldsymbol{\sigma_f}, \nabla \boldsymbol{v} \rangle_\Omega + \langle q, \nabla \cdot \boldsymbol{u_f} \rangle_\Omega = \langle \boldsymbol{\lambda}, \boldsymbol{v} \rangle_P \tag{5.14}$$

Fluid temperature:

$$\langle \frac{\partial \mathrm{T_f}}{\partial t} - \boldsymbol{u_f} \cdot \nabla \mathrm{T_f} - Q_f, \gamma \rangle_\Omega + \frac{1}{RePr} \langle \nabla \mathrm{T_f}, \nabla \gamma \rangle_\Omega = \langle \lambda_T, \gamma \rangle_P \tag{5.15}$$

Solid:

$$\langle (\rho_s - \rho_f) \frac{d\boldsymbol{u_s}}{dt} - (\rho_s \boldsymbol{f_s} - \rho_f \boldsymbol{f_f}), \boldsymbol{\omega} \rangle_P + \langle \boldsymbol{\sigma_s} - \boldsymbol{\sigma_f}, \nabla \boldsymbol{\omega} \rangle_P = \int_{\partial P} \boldsymbol{F^e} \cdot \boldsymbol{\omega}\, \boldsymbol{dx} - \langle \boldsymbol{\lambda}, \boldsymbol{\omega} \rangle_P \tag{5.16}$$

Lagrange multiplier constraint:

$$\langle \boldsymbol{u_f} - \boldsymbol{u_s}, \boldsymbol{\xi} \rangle_P = 0 \tag{5.17}$$

Solid temperature Lagrange multiplier:

$$\langle(\rho_r C_{pr} - 1)\frac{d\mathrm{T}_s}{dt} - (Q_s - Q_f), \gamma_s\rangle_P + \langle\frac{k_r - 1}{Pe}\nabla\mathrm{T}_s, \nabla\gamma_s\rangle_P = -\langle\lambda_T\,, \gamma_s\rangle_P \tag{5.18}$$

where ξ and γ_s are the test functions on the Lagrange multiplier space. Here, the distributed Lagrange multiplier (DLM), is introduced in the FD formulation to separate the fluid and solid equations from Eq. (5.11) and relax the velocity constraint. Note that, λ_T is the temperature-based Lagrange multiplier, which acts as a pseudo heat source [82], besides λ is the momentum based Langrange multiplier which acts as a pseudo body force in the momentum equation. To impose incompressibility constraint for the solid, we solve for solid pressure and explicitly add the following to Eq. (5.16):

$$\int_P (J - 1).\zeta dx = 0 \tag{5.19}$$

It is worth pointing out that considering the system as a whole, in the pair of Eqs. (5.14, 5.16) and Eqs. (5.15, 5.18), the fictitious Lagrange multiplier terms cancel out.

5.1.3 Numerical scheme

To solve for the fluid velocity-pressure coupling in Eq. (5.14), we use the optimized algorithm for incremental pressure correction scheme (IPCS) described in section 3.1.3 and 3.2.1 and also introduce stabilization schemes, namely: SUPG (Eq. (3.12)), PSPG (Eq. (3.15)) and crosswind stabilization (Eq. (3.16)). Detailed information on the need for stabilization schemes is provided in section 3.2. Further, to non-dimensionalize the governing equations, we introduce the following characteristic length L_{sc} and velocity V_{sc} scales, which changes the variable unknowns to $\overline{\boldsymbol{u_f}} = \frac{\boldsymbol{u_f}}{V_{sc}}$, $\overline{\boldsymbol{u_s}} = \frac{\boldsymbol{u_s}}{V_{sc}}$, $\overline{p_f} = \frac{p_f}{\rho_f {V_{sc}}^2}$, $\overline{p_s} = \frac{p_s}{\rho_f {V_{sc}}^2}$ and $\overline{\boldsymbol{x}} = \frac{\boldsymbol{x}}{\boldsymbol{L_{sc}}}$. The subsequent non-dimensional solid parameters are $\rho_r = \frac{\rho_s}{\rho_f}$, $C_{pr} = \frac{C_{ps}}{C_{pf}}$, $k_r = \frac{k_s}{k_f}$, $\overline{\lambda_0} = \frac{\lambda_0}{\rho_f {V_{sc}}^2}$ and $\bar{G} = \frac{G}{\rho_f {V_{sc}}^2}$. To summarize, we have

a set of four independent control parameters for our IB-FSI formulation: $Re = \frac{V_{sc}\, L_{sc}}{\nu}$, $Fr = \frac{V_{sc}}{\sqrt{g L_{sc}}}$, $Pr = \frac{\nu}{\alpha}$, $Ec = \frac{{V_{sc}}^2}{C_{pf}(\mathrm{T}_h - \mathrm{T}_l)}$, where ν is the kinematic viscosity given as $\nu = \frac{\mu}{\rho_f}$, g is the acceleration due to gravity, α is the thermal diffusivity and we introduce Ec, that accounts for heat generation due to viscous dissipation of the fluid. Another important control parameter that appears in the energy equation is the Peclet number; $Pe = Re \cdot Pr$. This completes the dimensionless FD/DLM formulation with FE stabilizations. The numerical algorithm in order of solving the equations and its corresponding numerical discretization is given below:

Fluid:

$$\begin{aligned}\langle \frac{\boldsymbol{u_f}^* - \boldsymbol{u_f}^{\boldsymbol{\theta}}}{\Delta \boldsymbol{\tau}} + \frac{\mathbf{3}\boldsymbol{u_f}^{\boldsymbol{\theta}} - \boldsymbol{u_f}^{\boldsymbol{\theta-1}}}{\mathbf{2}} \cdot \boldsymbol{\nabla u_{ck}^*}, \boldsymbol{v} \rangle_\Omega \\ + \langle -{p_f}^\theta \boldsymbol{I} + \frac{\boldsymbol{\nabla u_{ck}^*} + (\boldsymbol{\nabla u_{ck}^*})^T}{Re}, \nabla \boldsymbol{v} \rangle_\Omega \\ + \langle {p_f}^\theta \cdot \boldsymbol{n}, \boldsymbol{v} \rangle_{\partial\Omega} - \langle \frac{\widehat{\boldsymbol{g}}}{Fr^2}, \boldsymbol{v} \rangle_\Omega \\ + \langle \gamma_{SUPG}(\boldsymbol{u_f}^{\boldsymbol{\theta}})\, P(\boldsymbol{u_f}^{\boldsymbol{\theta}}, \boldsymbol{v}), R^\theta \rangle_\Omega \\ + \langle \gamma_{CW}(\boldsymbol{u_f}^{\boldsymbol{\theta}})\, \Lambda(\boldsymbol{u_f}^{\boldsymbol{\theta}}, \boldsymbol{u_f}^*), \nabla \boldsymbol{v} \rangle_\Omega = \langle \boldsymbol{\lambda^\theta}, \boldsymbol{v} \rangle_P\end{aligned} \tag{5.20}$$

$$\langle \nabla\big({p_f}^{\theta+1} - {p_f}^\theta\big), \nabla q \rangle_{\boldsymbol{\Omega}} + \frac{1}{\Delta\tau} \langle \nabla \cdot \boldsymbol{u_f}^*, q \rangle_{\boldsymbol{\Omega}} + \langle \gamma_{PSPG}\big(\boldsymbol{u_f}^*\big) \nabla q, R^\theta \rangle_{\boldsymbol{\Omega}} = 0 \tag{5.21}$$

$$\langle \frac{\boldsymbol{u_f}^{\boldsymbol{\theta+1}} - \boldsymbol{u_f}^*}{\Delta\tau}, \boldsymbol{v} \rangle + \langle \nabla\big({p_f}^{\theta+1} - {p_f}^\theta\big), \boldsymbol{v} \rangle_{\boldsymbol{\Omega}} = 0. \tag{5.22}$$

Fluid temperature:

$$\begin{aligned}\langle \frac{\mathrm{T}_{\boldsymbol{f}}^{\theta+1} - \mathrm{T}_{\boldsymbol{f}}^{\theta}}{\Delta\tau}, \gamma \rangle_{\boldsymbol{\Omega}} + \langle \boldsymbol{u_f}^{\boldsymbol{\theta+1}} \cdot \nabla \mathrm{T}_{ck}, \gamma \rangle_{\boldsymbol{\Omega}} + \frac{1}{Pe} \langle \nabla \mathrm{T}_{ck}, \nabla \gamma \rangle_{\boldsymbol{\Omega}} \\ - \langle \frac{2Ec}{Re} \langle \boldsymbol{E}(\boldsymbol{u_f}), \nabla \boldsymbol{u_f} \rangle^{\boldsymbol{\theta+1}}, \gamma \rangle_\Omega \\ + \langle \gamma_{SUPG}\big(\boldsymbol{u_f}^{\boldsymbol{\theta+1}}\big)\, P\big(\boldsymbol{u_f}^{\boldsymbol{\theta+1}}, \gamma\big), R^\theta \rangle_\Omega \\ + \langle \gamma_{CW}\big(\mathrm{T}_{\boldsymbol{f}}^{\theta}\big)\ \Lambda\big(\boldsymbol{u_f}^{\boldsymbol{\theta+1}}, \mathrm{T}_{\boldsymbol{f}}^{\theta+1}\big), \nabla \gamma \rangle_\Omega = \langle {\lambda_T}^\theta, \gamma \rangle_P\end{aligned} \tag{5.23}$$

Solid:

$$\langle \rho_r \frac{\Delta \boldsymbol{x_s}^{\boldsymbol{\theta+1}}}{\Delta \boldsymbol{\tau^2}}, \boldsymbol{\omega} \rangle_{P_0} + \langle (\nabla_0 \boldsymbol{\omega})^T, -p_s \boldsymbol{F^{-1}} + G(\boldsymbol{F^T} - \boldsymbol{F^{-1}}) \rangle_{P_0}{}^{\theta+1}$$
$$= \langle \frac{\boldsymbol{u_f}^{\boldsymbol{\theta+1}}}{\Delta \boldsymbol{\tau}} + (\rho_r - 1) \frac{\Delta \boldsymbol{x_s}^{\boldsymbol{\theta}}}{\Delta \boldsymbol{\tau^2}}, \boldsymbol{\omega} \rangle_{P_0} + \langle J^{\theta+1} - 1, \zeta \rangle_{P_0} \tag{5.24}$$
$$+ \langle (\rho_r - 1) \frac{\widehat{\boldsymbol{g}}}{Fr^2}, \boldsymbol{\omega} \rangle_{P_0} - \langle \boldsymbol{\lambda^\theta}, \boldsymbol{\omega} \rangle_{P_0}$$

DLM Constraint:

$$\langle \boldsymbol{u_s}^{\boldsymbol{\theta+1}} - \frac{\Delta \boldsymbol{x_s}^{\boldsymbol{\theta+1}}}{\Delta \boldsymbol{\tau}}, \boldsymbol{\xi} \rangle_P = \boldsymbol{0} \tag{5.25}$$

$$\langle \frac{\boldsymbol{u_s}^{\boldsymbol{\theta+1}} - \boldsymbol{u_f}^{\boldsymbol{\theta+1}}}{\Delta \boldsymbol{\tau}}, \boldsymbol{\omega_s} \rangle_P = \langle \lambda^{\theta+1} - \lambda^\theta, \boldsymbol{\omega_s} \rangle_P \tag{5.26}$$

Solid temperature Lagrange multiplier:

$$\langle \rho_r C_{pr} - 1 \frac{\mathrm{T_s}^{\theta+1} - \mathrm{T_s}^{\theta}}{\Delta \tau}, \gamma_s \rangle_P + \langle \frac{2Ec}{Re} \langle \boldsymbol{E}(\boldsymbol{u_f}), \nabla \boldsymbol{u_f} \rangle^{\boldsymbol{\theta+1}}, \gamma_s \rangle_P$$
$$+ \langle \frac{k_r - 1}{Pe} \nabla \mathrm{T}_{s,ck}, \nabla \gamma_s \rangle_P = -\langle \lambda_T{}^{\theta+1}, \gamma_s \rangle_P \tag{5.27}$$

In the above equations, $\boldsymbol{u^*_{ck}}$ and T_{ck} is the Crank-Nicolson velocity and temperature given as $\boldsymbol{\nabla u^*_{ck}} = 0.5\left(\boldsymbol{u_f}^* + \boldsymbol{u_f}^{\boldsymbol{\theta}}\right)$ and $\mathrm{T}_{ck} = 0.5\,(\mathrm{T_f}^{\theta+1} + \mathrm{T_f}^{\theta})$ and $\Delta \boldsymbol{x_s}$ is the incremental solid displacement. Also, Eq. (5.22) is solved in the reference configuration P_0 and the gradients in both the current and reference configuration are related as $\boldsymbol{\nabla} = \boldsymbol{\nabla}_0(\boldsymbol{F^{-1}})$. Note that in the above equations we drop the bar on top of the non-dimensional variables and neglect any external hydrodynamic forces at the fluid-structure interface i.e., $\boldsymbol{F^e} = \boldsymbol{0}$ for the sake of simplicity. Also, one needs to keep in mind that Eq. (5.22) applies to a perfectly incompressible solid and we compute both the solid pressure p_s and displacement $\boldsymbol{x_s}$. However, if the solid is compressible, we replace the solid pressure by $\frac{\lambda_0 lnJ}{J}$ and remove $\langle J - 1, \zeta \rangle_{P_0}$ term from Eq. (5.24) and solve only for the incremental solid displacement $\Delta \boldsymbol{x_s}$.

To solve Eq. (5.20) and Eq. (5.23), we use the stabilized biconjugate gradient iterative method with hypre-amg preconditioners at an absolute tolerance value of 10^{-8}. To solve the solid problem

Eq. (5.24), we use the non-linear Newton's iteration for a relative tolerance value of 10^{-6}. The $\Delta \boldsymbol{x_s}$ from the previous time step is provided as initial guess to the Newton solver and we observe that it takes about 3-4 iterations to converge.

5.1.4 Interpolation functions

To interpolate fictitious forces, heat source, velocity, or temperature from the Eulerian mesh to the Lagrangian mesh or vice-versa, we use the 4-point conservative piecewise-delta function [83] given below:

$$u^{\phi}(X) = \int_{\Omega} u^{\phi}(x)\delta(x - X)\, dx\,; \qquad \lambda(x) = \int_{P} \lambda(X)\delta(x - X)\, dx\,;$$
$$\mathrm{T}_s(X) = \int_{\Omega} \mathrm{T}_s(x)\delta(x - X)\, dx; \qquad \lambda_T(x) = \int_{P} \lambda(X)\delta(x - X)\, dx \tag{5.28}$$

$$\delta(\boldsymbol{x} - \boldsymbol{X}) = \frac{1}{h^3}\emptyset\left(\frac{x - X}{h}\right)\emptyset\left(\frac{y - Y}{h}\right)\emptyset\left(\frac{z - Z}{h}\right) \tag{5.29}$$

$$\emptyset(r) = \begin{cases} \frac{1}{8}(3 - 2|r| - \sqrt{1 + 4|r| - 4|r|^2}) & |r| \le 1 \\ \frac{1}{8}(5 - 2|r| - \sqrt{-7 + 12|r| - 4|r|^2}) & 1 \le |r| \le 2 \\ 0 & 2 \le |r| \end{cases} \tag{5.30}$$

In the IB-FSI solver, the delta functions essentially determine the order of accuracy of the IB method. Due to the inherent nature of the delta functions, it is obligatory that the fluid grid surrounding the solid is structured and for good accuracy, we recommend that the solid mesh size stays between $0.8h$ - $1.5h$ at all times where h is the uniform fluid grid size [33, 84]. Here, note that the solid grid can be unstructured and non-uniform. Also, the delta functions utilize piecewise functions $\emptyset$ with different support areas and continuity properties which directly dictate their behavior in terms of numerical stability and accuracy [85]. We use the 4-point piecewise function (Eq. (5.30)) since we observe that it provides a good balance between stability and accuracy.

5.2 vanDANA solver

Our code is open-source and available to use in the form a GitHub repository: https://github.com/patelte8/vanDANA. vanDANA is an efficient FEM Immersed Boundary (IB) based Flow-thermal FSI solver utilizing the FEniCS library and it is a python package with main executable (vanDANA.py) and three submodules (common, utilities, user_inputs). The directory tree for vanDANA code is given below in **Figure 5.3**.

```
vanDANA-IB-FSI
 |
 ├ __init__.py
 ├ vanDANA.py
 ├ LICENSE.md
 |
 ├ common
 | ├ __init__.py
 | ├ constitutive_eq.py
 | ├ delta_interpolation.py
 | ├ fem_stabilizations.py
 | ├ flow_temperature_variational_problem.py
 | ├ flow_variational_problem.py
 | ├ functions.py
 | ├ lagrange_variational_problem.py
 | ├ solid_variational_problem.py
 | └ solver_options.py
 |
 ├ user_inputs
 | ├ __init__.py
 | ├ boundary_initial_conditions.py
 | ├ problem_specific.py
 | ├ user_parameters.py
 | └ ...
```

```
 ├ utilities
 | ├ __init__.py
 |.├ utils.py
 | ├ write.py
 | └ read.py
 |
 ├ src
 | ├ flag.geo
 | ├ mesh.py
 | ├ fenics.sb
 | ├ fenics_2019_dev
 | └ ...
 |
 └ results
   ├ HDF5_files
   ├ mesh_files
   ├ restart_variables
   ├ XDMF_files
   └ text_files
     ├ flow_data.txt
     ├ flow_temp_data.txt
     ├ lagrange_data.txt
     ├ log_info.txt
     ├ restart.txt
     ├ runtime_stats.txt
     ├ solid_data.txt
     ├ solid_mesh_quality.txt
     └ solid_temp_data.txt
```

Figure 5.3: Directory tree for vanDANA IB-FSI code.

1. The **common** module is the heart of the solver and mainly comprises of different variational problems: flow_variational_problem.py, flow_temperature_variational_problem.py, solid_variational_problem.py, lagrange_variational_problem.py. Each of these files contain separate class definitions and subroutines for pre-assembly, run-time assembly, solvers, and post-processing for their respective variational problems. We also include some basic post-processing subroutines for calculation of drag, lift, Nusselt number, Jacobian, and vorticity.

Hence, the user need not require making any changes to the common module unless one needs to implement a custom user-defined subroutine.

2. The **utilities** module deals with the handling of background functions for the code and comprises of the files: utils.py, read.py and write.py. Almost all the functionality for read, write, restart, MPI, counters, memory usage and timing of different modules is initialized and carried out using this module.

3. Next, the **user_inputs** module provides input to the code mainly in terms of control parameters, time step, meshes, initial and boundary conditions for any given physical problem. Here, we allow user control from two main files namely: user_parameters.py and boundary_initial_conditions.py. A sample example of user parameters.py is given below in **Figure 5.4**. All parameters are grouped in the form of python dictionaries which makes them easy to append during runtime.

```
restart = False                          # Restart parameter

# Physics of the problem
# ---------------------------------------------------------------------
problem_physics = dict(
                                    solve_temperature = True,
                                    solve_FSI = True,
                                    compressible_solid = True,
                                    viscous_dissipation = False,
                                    body_force = False,
                                  )

def f_dir(dim):                          # Body force direction
      n = -1*tensors.unit_vector(1, dim)
      return n

interpolation_fx = 'phi4'                # Delta-function interpolation

# FEM stabilization and constants
# ---------------------------------------------------------------------
stabilization_parameters = dict(
```

Figure 5.4: All control parameters are provided as input to the vanDANA code from user_parameters.py.

Figure 5.4 (cont'd)

```
        # Navier-stokes
        SUPG_NS = False,                    # explicit
        PSPG_NS = False,                    # explicit
        crosswind_NS = False,               # implicit

        # Energy-equation
        SUPG_HT = False,                    # explicit
        crosswind_HT = False                # implicit
)

alpha = Constant(0.85)                      # SUPG/PSPG stabilization constant
C_cw = Constant(0.7)                         # Crosswind stabilization constant

# Physical parameters
# ------------------------------------------------------------------
physical_parameters = dict(

        g = 9.81,                           # Gravity (m/s2)

        # Fluid
        rho_f = 1,                          # Density (kg/m3)
        nu = 1,                             # Dynamic viscosity (kg/m.s)
        Spht_f = 1,                         # Specific heat (J/kg.C)
        K_f = 1,                            # Thermal conductivity (W/m.C)

        # Solid
        rho_s = 10,                         # Density (kg/m3)
        Sm = 0,                             # Shear modulus (N/m2)
        Ld = 0,                             # Compressibility (N/m2)
        Spht_s = 0.11,                      # Specific heat (J/kg.C)
        K_s = 1.2                           # Thermal conductivity (W/m.C)
)

def calc_non_dimensional_solid_properties(g, rho_f, nu, Spht_f, K_f, rho_s, Sm, Ld,
Spht_s, K_s, Lsc, Vsc, T0, Tm, Tsc):

        rho = rho_s/rho_f
        Spht = Spht_s/Spht_f
        K = K_s/K_f
        Ld = 2000 # Ld/(rho_f*Vsc*Vsc)
        Sm = 500 # Sm/(rho_f*Vsc*Vsc)

        return rho, Spht, K, Ld, Sm

# Characteristic scales
# ------------------------------------------------------------------
characteristic_scales = dict(

        Lsc = 1,                            # m
        Vsc = 1,                            # m/s
```

Figure 5.4 (cont'd)

```
        T0 = -1*52,                          # lower_temp (C)
        Tm = 37                              # higher_temp (c)
    )

    # Temporal control
    # ------------------------------------------------------------------
    time_control = dict(
                            C_no = 0.35,       # Maximum possible Courant number
                            dt = 0.0025,       # Time-step
                            T = 100,           # Total runtime
                            adjustable_timestep = True
                    )

    # FEM degree of variables
    # ------------------------------------------------------------------
    fem_degree = dict(
                            velocity_degree = 2,
                            pressure_degree = 1,
                            temperature_degree = 2,
                            displacement_degree = 2,
                            lagrange_degree = 1
                   )

    # Non-dimensional numbers
    # ------------------------------------------------------------------
    def calc_non_dimensional_numbers(g, rho_f, nu, Spht_f, K_f, rho_s, Sm, Ld, Spht_s, K_s,
    Lsc, Vsc, T0, Tm, Tsc):

        Re = rho_f*(Vsc*Lsc)/nu
        Pr = Spht_f*nu/K_f
        Ec = (Vsc*Vsc)/(Spht_f*(Tm-T0))
        Fr = Vsc/sqrt(g*Lsc)

        return Re, Pr, Ec, Fr

    # Enter "True" if you want to post-process data
    # ------------------------------------------------------------------
    post_process = True

    # File printing / solid-remeshing control
    # ------------------------------------------------------------------
    print_control = dict(
                      a = 40,         # for printing variables and restart files
                      b = 20,         # for post processing data
                      c = 20,         # for simulation_wall_time text file
                      d = 5,          # for remeshing solid current-configuration mesh
                      e = 20          # for runtime_tsp_courant_no_stats text file
                    )
```

In user_parameters.py, we have provided the following important features, namely:

- The restart parameter, if "True" will restart the simulation from the last saved time step.
- The order of finite-element basis functions for all variables can be specified using the fem_degree dictionary.
- In the time_control dictionary if adjustable_timestep is set as True, the code will automatically calculate the time step during run-time by limiting the maximum Courant number to time_control['C_no']. Otherwise, if adjustable_timestep is set as False, it will use time_control['dt'] as a constant time step.
- The calc_stream_function flag if True, enables calculation of vorticity and stream function only for a 2D problem setup. This is because the steam function is defined only in 2D, using $\nabla^2\psi = -\omega$, where ω is the vorticity.
- The effect of stabilization schemes can also be tuned using stabilization constants: **alpha** and **C_cw**.

To execute the code, vanDANA.py is the main executable file that pulls information from all the three modules and runs the main function **vanDANA_solver (args).** This sets the entire workflow defined in Eqs. (5.20-5.27). Overall, vanDANA code is aimed to be user-friendly and hence to gain more user-control for ease of testing and restarting, we use the argparse library [86] to add keywords from the command line, for e.g.: mpirun.mpich -n 64 python3 vanDANA.py -restart=True -T=20.

5.3 Benchmarks

In this section we validate four different FSI benchmarks. In all these benchmark cases, we used second order (quadratic) elements for velocity and solid displacement and linear elements for pressure and Lagrange multiplier.

5.3.1 Transverse flow over a flexible beam

This problem was introduced by Baaijens [34] and later also studied by Yu [33]. A thin flexible leaflet is located transverse to the flow direction at the center of the 2D channel as shown in **Figure 5.5**. For velocity boundary conditions, a no-slip at Γ_2, free slip at Γ_4 and a parabolic time varying velocity profile given as $U = 1.5y(2-y)\sin\left(\frac{2\pi}{T}t\right)$ is provided at Γ_1 and Γ_3, where the origin is defined as the bottom left corner and the total runtime is $T = 10$. For pressure we provide wall boundary conditions on all the sides. The channel length is 10 and height $H = 1$. The density ratio is set as $\rho_r = 1$ and the solid is considered incompressible with stiffness $G = 10^3$. Note that since this is an incompressible leaflet i.e., $J = 1$, we also solve for the solid pressure.

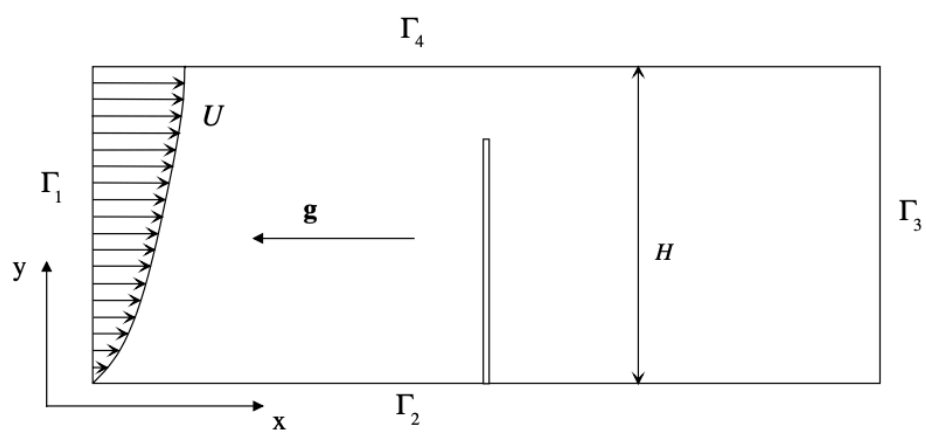

Figure 5.5: Schematic geometry for transverse flow over a flexible plate.

The leaflet is held stationary at its bottom end and its length and height are 0.8 and 0.0212 respectively. All body forces are neglected. The Reynolds number considered here is 100 and the fluid mesh is uniform with a mesh size of $h_f = 1/128$. The solid mesh size is $h_s = 1/100$ along the length of the leaflet and it has 3 grid cells along its thickness. The time step is set as 0.005. **Figure 5.6** shows our leaflet position in the channel at 8 different time instants over one cycle of oscillation. These results depict and an excellent qualitative match with those of Yu [33].

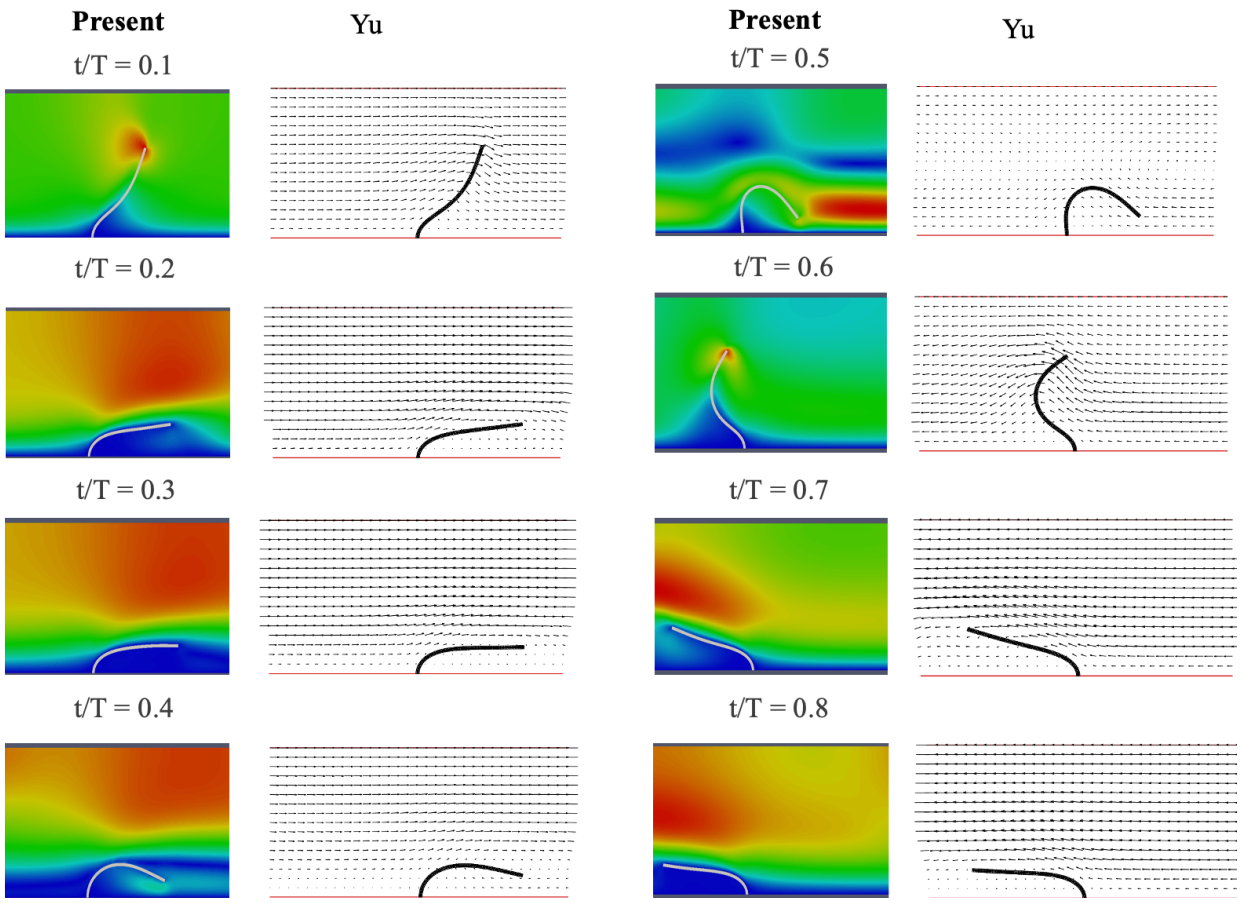

Figure 5.6: Transverse flow over a flexible leaflet: Re 100, $\rho_r = 1$, $G = 10^3$. A comparison of present results with Yu [33] for leaflet positions in the channel over one periodic cycle.

5.3.2 Turek and Hron (FSI2)

Next, we consider the classical FSI2 benchmark by Turek and Hron [87] - flow over an elastic beam attached behind a rigid cylinder. The schematic is given in **Figure 5.7**. Here D, the diameter of the rigid cylinder is defined as the characteristic length scale L_{sc} and is chosen as $D = 1$. In **Figure 5.7,** other model dimensions are given as $H = 4.1$, $L = 11$ and $l = 3.5$, $h = 0.2$. The center of the rigid cylinder is (2,2) and at time $t = 0$, the flexible beam is symmetric with respect to the cylinder center. The boundary conditions for the top and bottom boundary are no-slip, left boundary is a constant inlet velocity profile $U = \frac{6U_0 y(H-y)}{H^2}$ and the right boundary is outflow. Here, $U_0 = 1$ is defined as the characteristic velocity scale V_{sc}. For pressure, we apply wall boundary

condition at all the boundaries, except the at the right boundary the pressure is set as 0. The Re is defined with respect to D and U_0, and is set as 100.

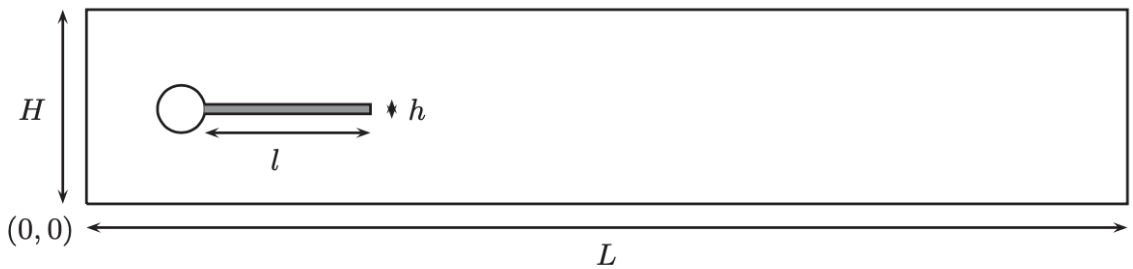

Figure 5.7: A schematic diagram for the FSI2 benchmark [87]: Flow past a flexible beam (compressible) attached behind a rigid cylinder.

For the properties of the flexible beam, the non-dimensional shear modulus is G = 500, density ratio is $\rho_r = 10$, and the solid is considered compressible with non-dimensional compressibility as $\lambda_0 = 2000$. The fluid mesh is 575 x 215, and the flexible beam has a resolution of 175 x 10. The time step is set such that the average courant number during runtime is around 0.35 and the total non-dimensional run time is T = 100 sec. The total DOF's for the case setup is 1.15 million.

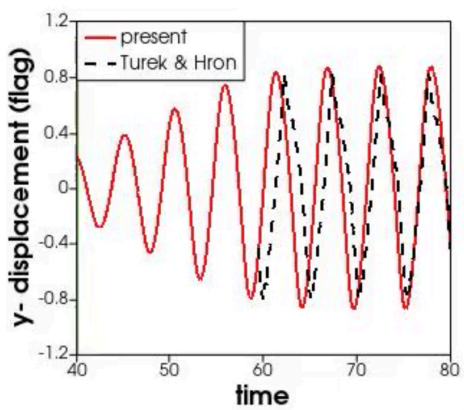

Figure 5.8: Comparison of present y-displacement of trailing tip of the flexible beam with Turek and Hron [87].

Our results showcase an excellent match for the y-displacement of the trailing tip of the flexible beam (**Figure 5.8**), strouhal number and average drag coefficient (**Table 5.1**) with results from the

literature. The drag coefficient is calculated as $C_D = \frac{F_x}{0.5\rho_f U_0^2 D}$, where $F_x = \boldsymbol{\sigma_f} \cdot \boldsymbol{n_1}$ and n_1 is the unit vector along the flow direction. **Figure 5.9** illustrates the instantaneous beam position and velocity and pressure contours at a particular time instant during the simulation.

Table 5.1: Comparison of present average drag coefficient, Strouhal number and trailing tip displacement amplitude.

	present	Turek and Hron [87]	Tian et al. [88]
C_D	4.10	4.13	4.11
Amplitude	0.82	0.83	0.78
Strouhal no.	0.19	0.19	0.19

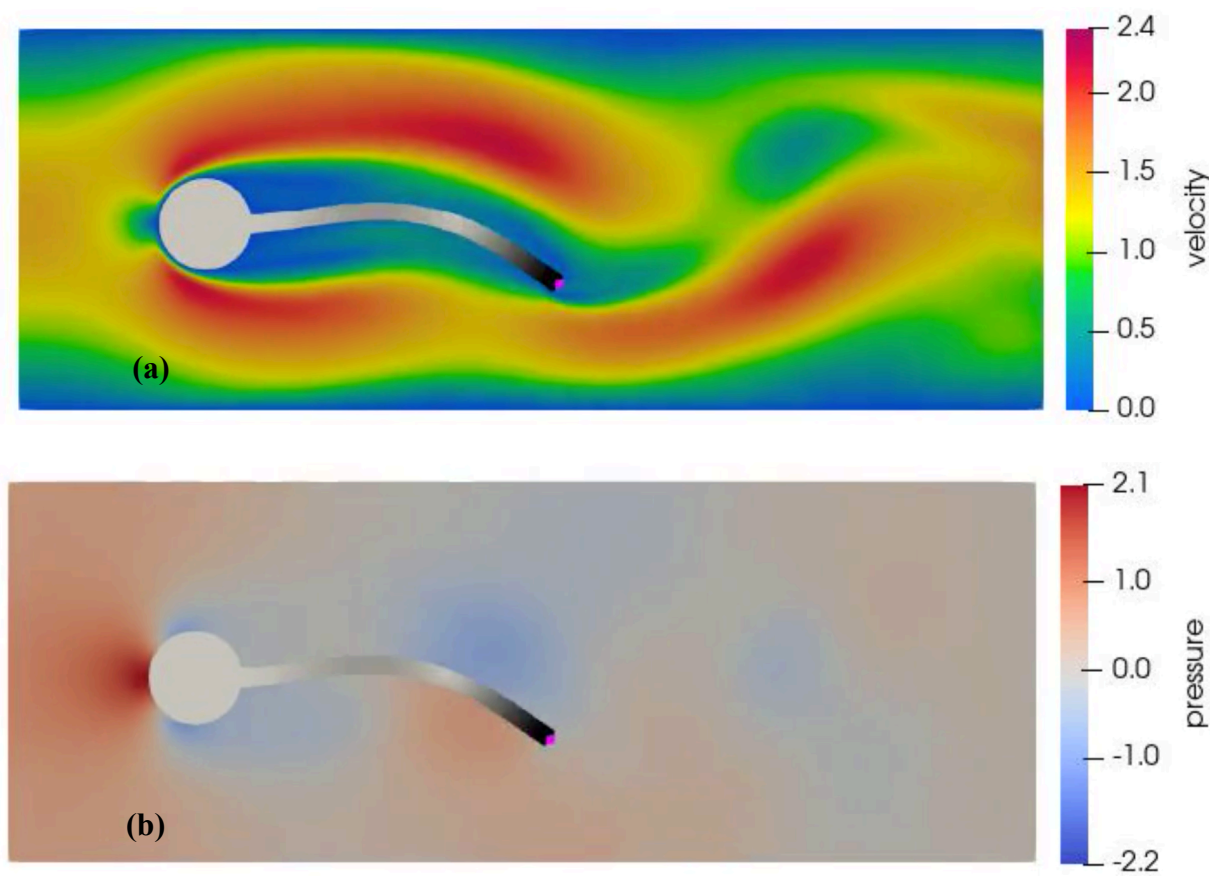

Figure 5.9 (a) Velocity and (b) pressure contours for flow over a flexible beam attached behind a rigid cylinder at T = 71.

5.3.3 Flapping of a 3D flexible plate in uniform flow

Next, we consider the self-sustained flapping of a 3D thin elastic plate in uniform viscous flow and compare our simulation results with Yu et al. [89]. The computational domain is a cuboid domain of size (4, 2, 1) and the flexible plate has dimensions (1, 2/3, 1/30). The plate density ratio is $\rho_r = 16$, and the shear modulus is set as $G = 50$. The plate is treated incompressible by providing a high value of $\lambda_0 = 100G$. The fluid mesh is 200 x 100 x 50 and is uniform in all three directions, whereas the plate mesh is 50 x 30 x 4 and the initial configuration of the plate is inclined with the streamwise direction by an angle $\tan^{-1}(0.2)$. A problem schematic is given in **Figure 5.10** where the flow in the streamwise direction (+ve x-direction) at a $Re = 500$ and the plate is clamped at its leading edge. The flow boundary conditions are uniform flow at the inlet (x=0), top and bottom boundaries. On the spanwise boundaries (z=0 and z=1) we impose periodic boundary conditions and at the outlet we specific outflow or Neumann boundary condition. We use a variable time step during runtime such the convection CFL number stays below 0.25. **Figure 5.11** shows the time development of point y_1 on the elastic plate. Our present results showcase an excellent match with Yu et al. [89]. Here we note that the simulation is stable only if the SUPG stabilization (Eq. (5.20)) is enabled for the flow solver.

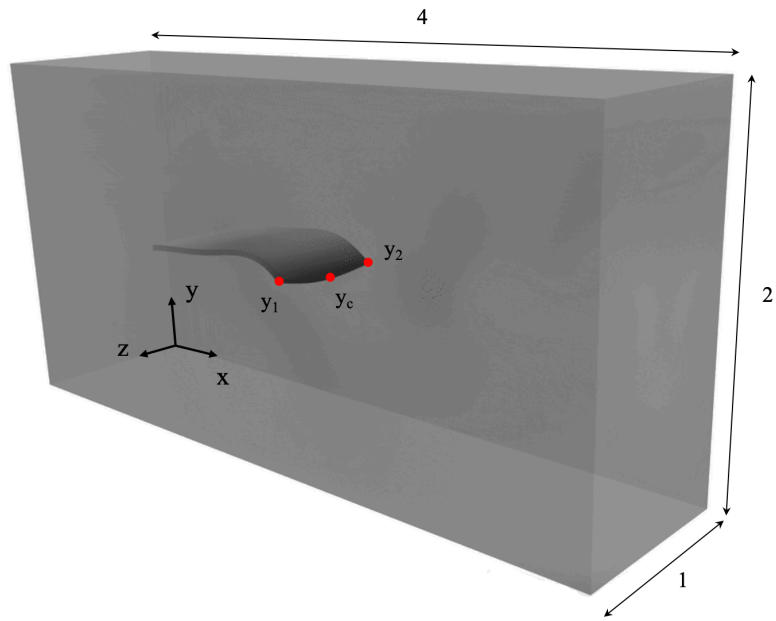

Figure 5.10: Problem schematic for the flapping of a 3D flexible plate in uniform flow. The plate is positioned equidistant from the front/back and top/bottom faces of the fluid domain and its initial configuration is at an angle of $\tan^{-1}0.2$ with respect to the x-direction.

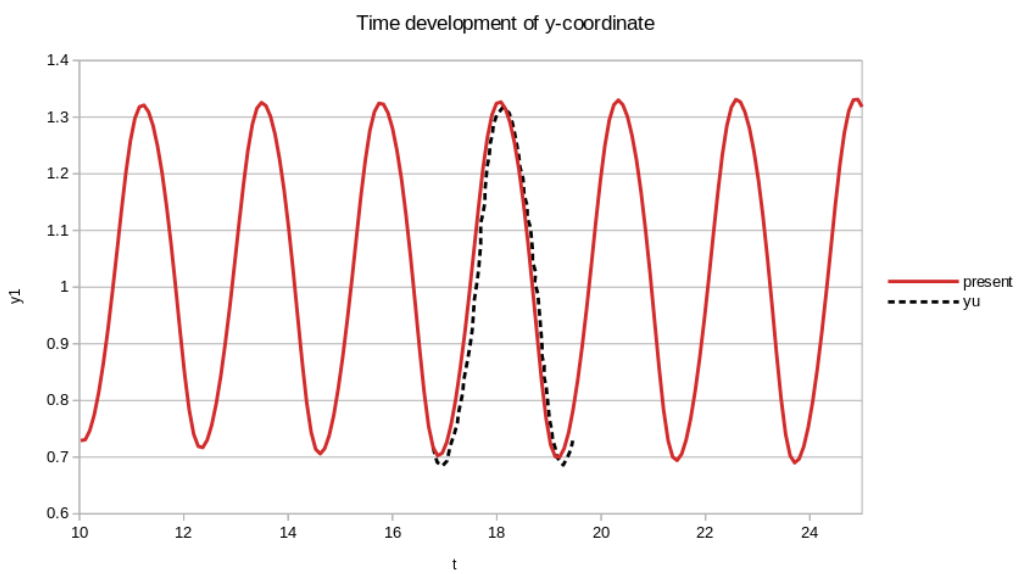

Figure 5.11: A comparison of the present time development of point y_1 on the elastic plate with Yu et al. [89].

Moreover, we also increase the plate shear modulus to G = 100 and 250 and **Table 5.2** shows the oscillation amplitude and Strouhal number comparison with Yu et al [89] and we observe that the plate becomes almost stationary as the shear modulus is increased up to G = 250.

Table 5.2: Comparison of y_1 displacement amplitude and Strouhal number at two different shear moduli of the elastic plate.

	present	Yu et al. [89]
***G* = 100**		
y_1 amplitude	0.26	0.27
Strouhal no.	0.44	0.44
***G* = 250**		
y_1 amplitude	0.02	0

5.3.4 Hagen-Poiseuille flow

Next, we consider an internal pipe flow problem using an immersed boundary setup. For a standard 3-D pipe flow problem the analytical fully developed velocity profile is given by

$$v(r) = 2U\left(1 - \frac{r^2}{R^2}\right) \tag{5.31}$$

Moreover, at steady state, the pressure drop in the pipe is given as

$$\Delta P = \frac{8\mu LQ}{\pi R^4} \tag{5.32}$$

In the above equations, v is the velocity at any given radius r, U is the mean velocity and Q is the flow rate through the pipe cross-section, μ is the dynamic viscosity in $Pa \cdot s$ and L and R are the length and radius of the pipe. On converting Eq. (5.32) into non-dimensional form the pressure drop is

$$\Delta\bar{P} = \frac{16}{Re}\left(\frac{L}{R}\right) \tag{5.33}$$

where $\bar{P} = \frac{P}{\rho U^2}$ is the non-dimensional pressure and $Re = \frac{\rho U(2R)}{\mu}$. For the computational setup, we consider the fluid domain to be a cuboid of dimensions (3.2, 2, 2) with a uniform mesh of 84 X 56 X 56. To model the solid pipe, we consider a hollow cylinder with an inner radius of $R = 0.5$, thickness 0.1 and length 3.2. The pipe material is a compressible neo-Hookean solid (Eq. (5.7)) with $G = 1000$ and $\lambda_0 = 100G$. For boundary conditions, the two pipe ends are clamped and are prescribed with the fully developed analytic velocity profile using Eq. (5.31). The remaining parts of the boundary are prescribed zero-velocity boundary condition. The solid pipe is meshed using 3-D tetrahedral elements and the shear-modulus is increased to model the pipe as a stiff hollow-cylinder. We simulate Hagen-Poiseuille flow at three different Re and benchmark the simulated pressure drop across the length of the pipe. **Figure 5.12** shows the cross-sectional steady state velocity contours in the fluid domain for $Re = 100$. **Table 5.3** shows the comparison of simulated pressure drop vs. the analytical pressure drop across the pipe length along with the percentage error at different Re. The percentage error is less than 8% for all the Re considered.

Table 5.3: A comparison of simulated pressure drop with the analytical pressure drop and the percentage error for Re = 10, 100, 1000.

Re	Present ΔP	Analytical ΔP	Percentage error (%)
10	10.98	10.24	7.22
100	1.028	1.024	0.4
1000	0.1101	0.1024	7.52

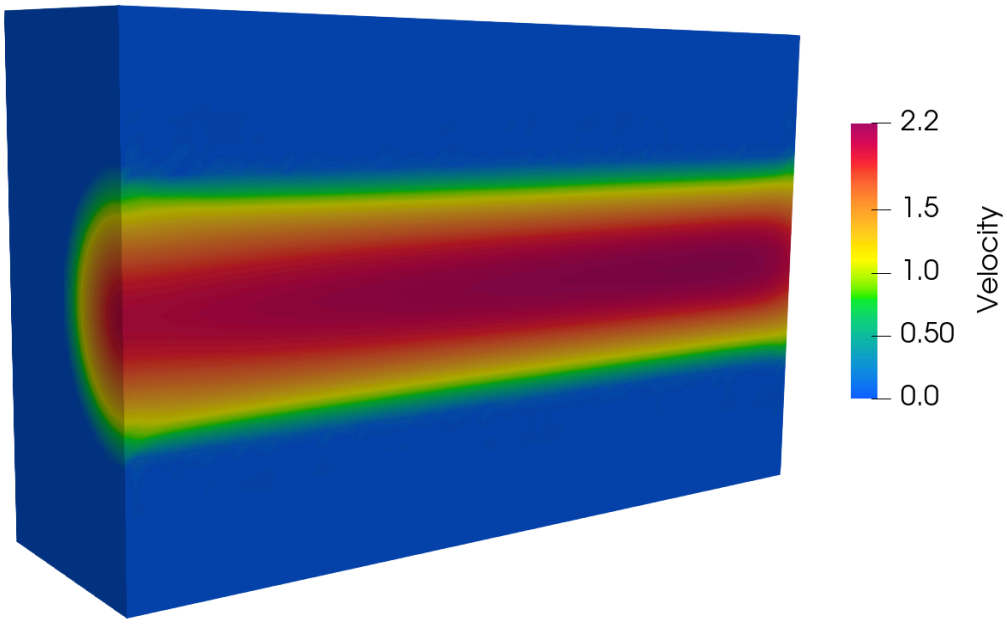

Figure 5.12: Velocity contours for Hagen-Poiseuille flow using an immersed boundary setup at Re=100.

5.4 HPC performance

To test the HPC performance of vanDANA solver we have performed a strong scaling test for the 2D Turek and Hron benchmark using a relatively large number of parallel MPI tasks for varying Degrees of Freedom (DOF's). The test runs are conducted on Michigan State University's - **AMD EPYC 7H12** node and the wall timings are shown in **Figure 5.13** for four different degrees of freedom. At present, the delta-interpolation is the bottleneck for the vanDANA solver since the interpolation operation is conducted four times during each time step.

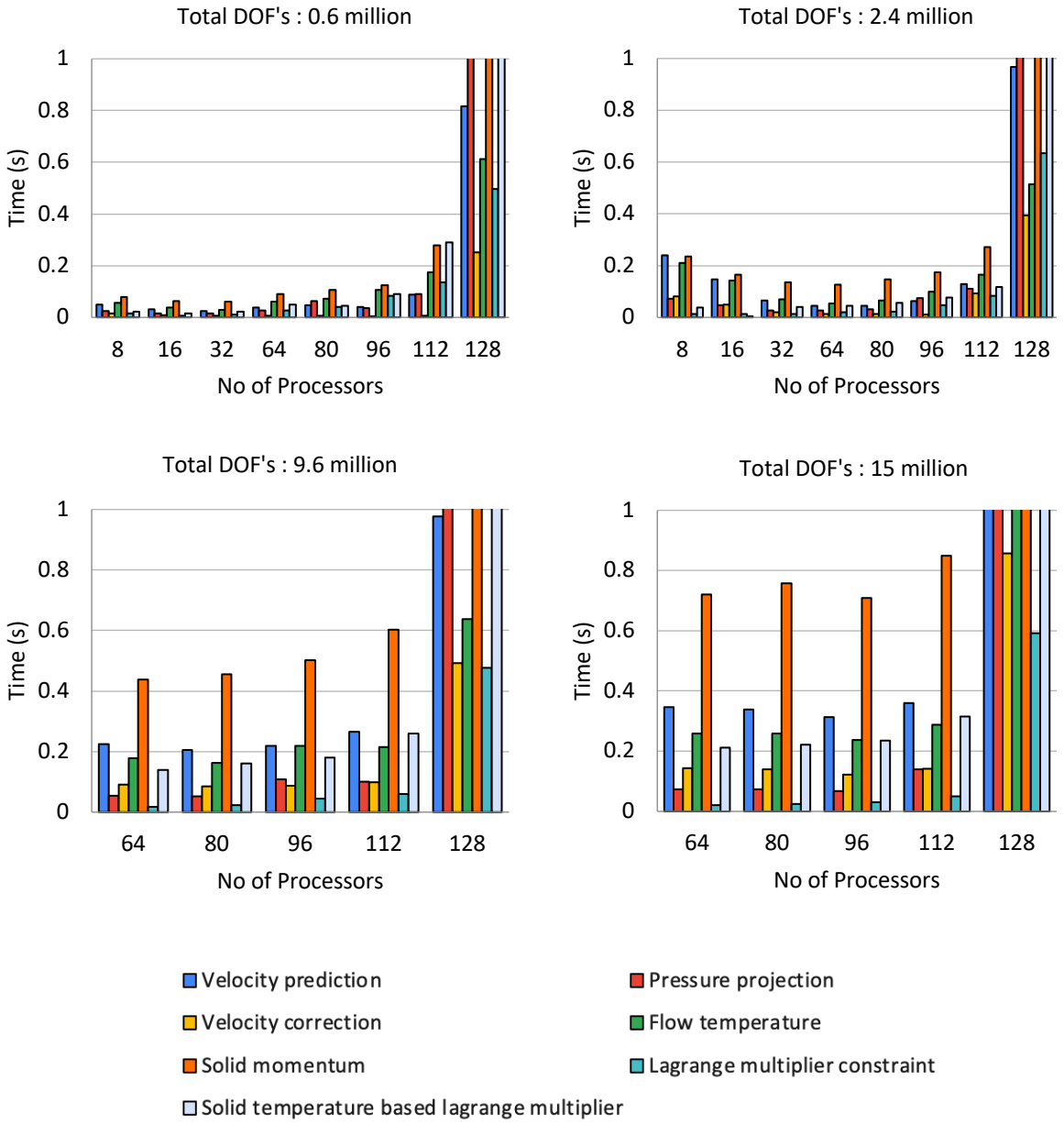

Figure 5.13: Strong scaling results for the Turek and Hron FSI2 benchmark. The legends showcase the timings for the 7 steps in sequence i.e., Eqs. (5.18-5.25). The timers in utilities/utils.py are used to measure each module.

CHAPTER 6 : SUMMARY

6.1 Limitations

Despite our successful attempt to simulate CBA surgery for PV isolation, and to develop a fully coupled FSI framework along with heat transfer, our study still faces a few limitations.

- **Limitations for the patient specific CBA simulation**

  First, the LA wall is considered rigid, and its contraction and relaxation are not considered. **Second,** we only consider laminar flow and do not consider the transition from laminar to turbulence that may occur in the LA [90, 91]. To capture flow turbulence and its effects on hemodynamics and temperature, one can augment our current algorithm with other readily available turbulence models (LES-Smagorinsky or LES-Kinetic energy SGS). **Third**, we neglected the thickness of the LA wall and the metabolic heat generation within the wall. Metabolic heat generation for cardiac tissue is, however, estimated to be 1 W/cm^3 [92] for $\boldsymbol{A_c} < 5\ cm^2$ and a LA wall thickness < 0.4 cm [63]. This value is insignificant in comparison to the power absorbed across the CB (**Figure 4.6**). **Fourth**, we prescribed a temperature boundary condition on the CB for simplicity. In practice, however, a system controls the delivery of refrigerant (liquid N_2O) to the CB and the temperature may drop depending on the degree of occlusion (or leak) [93]. **Fifth,** we do not consider phase changes arising from the formation of stable ice caps on the CB [94, 95]. Ice formation can contribute to PV occlusion and change the effective geometry of the CB, which in turn, can affect the hemodynamics. **Last**, the forward manual force applied on the catheter up to freezing will make the CB position and contact static. This would change the PV ostium geometry and significantly affect above-mentioned factors. Such effects, however, would be very difficult to quantify in clinical measurements and simulation models.

- **Limitations for the development of coupled FSI-thermal framework**

  First, our implementation of the delta-function interpolation kernel is the computational bottleneck of the vanDANA solver, and it takes about 10 seconds or longer for DOF's > 12 million. **Second,** while vanDANA using FEniCS, both the fluid and solid mesh are split into N-MPI partitions each. However, in such IB-FSI simulations in most cases the solid mesh is smaller in volume than the fluid mesh and such a partitioning strategy retards the scalability of the solver as a whole. **Third,** the present Lagrange multiplier based heat transfer algorithm is unstable at higher values of $\rho_r C_{pr} - 1$ and needs implicit treatment of the unsteady term in Eq. (5.25) [82]. **Fourth,** solving for physical problem cases with large values of k_r, can be unstable. However, this can be tackled by using a smaller time step. **Last,** the IB algorithm in conjunction with the delta-function interpolation kernel requires the fluid cartesian mesh to be uniform in all three directions at least in the region overlapping the solid. In case of an internal flow problem, this uses a larger fluid mesh volume. Here, one needs to judiciously provide physical flow boundary conditions since it can instigate solver divergence.

6.2 Concluding remarks

This thesis can mainly be summarized in two parts:

- **Patient specific CBA modeling:** We have developed a computational stabilized FE framework to simulate the dynamic thermal-fluid coupling in a realistic LA geometry during cryoablation therapy. We show that the framework can reproduce measurements of MV velocity and PV pressure waveforms measured in healthy human subjects. The modeling framework predicts that the effectiveness of cryoablation therapy based on the lesion size is very sensitive to the CB position, while PV inlet velocity and MR do not have substantial effects on the lesion size.

The framework can be applied in future clinical investigations to optimize cryoablation therapy in patient specific LA-PV geometries.

- **Development of vanDANA IB-FSI solver:** We have developed a fully coupled FE based immersed boundary FSI code using the Distributed Lagrange Multiplier based fictitious domain method and have extended it to deal with heat transfer. We have used FEniCS, and our code is open source, documented, user-friendly and is available to use in the form of a GitHub repository: vanDANA. The vanDANA code is benchmarked against complex FSI problem cases and its scalability has been well tested on HPC. In future, our computational framework can be used to simulate fully coupled FSI physics (including heat transfer) for complex cardiovascular flows.

The future direction for this project should be focused on addressing the limitations and improving the computational speed and scalability of the vanDANA solver. Other computational models (or subroutines) that incorporate turbulence modeling, active strain, contact mechanics, and non-Newtonian models for blood flow can be implemented into the vanDANA solver. This will be of use to capture complex flow physics and investigate cardiovascular diseases in patient specific geometries more accurately.

BIBLIOGRAPHY

[1] R. E. Klabunde, Cardiovascular physiology concepts, Lippincott Williams & Wilkins, 2011.

[2] "Assessment of Mitral Valve Function | Radiology Key.," July 3, 2022. [Online]. Available: https://radiologykey.com/assessment-of-mitral-valve-function.

[3] V. Markides and R. J. Schilling, "Atrial fibrillation: classification, pathophysiology, mechanisms and drug treatment," Heart 89, no. 8 , pp. 939-943, 2003 .

[4] Z. Nesheiwat, A. Goyal and M. Jagtap, Atrial Fibrillation (Nursing), Treasure Island (FL): StatPearls Publishing. PMID: 30252328, 2022 Jul 31.

[5] M. Sucu, V. Davutoglu and O. Ozer, "Electrical cardioversion," Annals of Saudi medicine 29, no. 3 , pp. 201-206, 2009.

[6] O. M. Wazni, G. Dandamudi, N. Sood, R. Hoyt, J. Tyler, S. Durrani and M. Niebauer, "Cryoballoon ablation as initial therapy for atrial fibrillation," New England Journal of Medicine 384, no. 4 , pp. 316-324, 2021 .

[7] M. Getman, E. Wissner, R. Ranjan and J. Lalonde, "Relationship between time-to-isolation and freeze duration: Computational modeling of dosing for Arctic Front Advance and Arctic Front Advance Pro cryoballoons," J Cardiovasc Electrophysiol. 30, p. 2274–2282, 2019.

[8] M. Yacoub and R. Sheppard, "Cryoballoon Pulmonary Vein Catheter Ablation of Atrial Fibrillation," StatPearls, 2021.

[9] N. Patel, J. Maradey and P. Bhave, "Atrial Fibrillation Ablation: Indications and Techniques," Curr Treat Options Cardio Med. 21, p. 43, 1936.

[10] W. Su, R. Kowal, M. Kowalski, A. Metzner and J. Svinarich, "Best practice guide for cryoballoon ablation in atrial fibrillation: The compilation experience of more than 3000 procedures," Heart Rhythm. 12, p. 1658–1666, 2015.

[11] T. Patel and A. Lakdawala, "A dual grid, dual level set based cut cell immersed boundary approach for simulation of multi-phase flow," Chemical Engineering Science. 177, p. 180–194, 2018.

[12] A. Athani, N. N. N. Ghazali, I. A. Badruddin, S. Kamangar, A. E. Anqi and A. Algahtani, "Investigation of two-way fluid-structure interaction of blood flow in a patient-specific left coronary artery," Bio-Medical Materials and Engineering 33 no. 1 , pp. 13-30, 2022.

[13] Y. I. Cho, D. J. Cho and R. S. Rosenson, "Endothelial shear stress and blood viscosity in peripheral arterial disease," Current atherosclerosis reports 16, pp. 1-10, 2014.

[14] A. Logg, K. Mardal and G. Wells, Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book, Manual, 2012.

[15] A. Henderson, ParaView Guide, A Parallel Visualization Application, Kitware Inc., 2007.

[16] V. Parikh and M. Kowalski, "Comparison of phrenic nerve injury during atrial fibrillation ablation between different modalities, pathophysiology and management," Journal of atrial fibrillation 8, no. 4 , 2015.

[17] D. L. Packer, R. C. Kowal, K. R. Wheelan, J. M. Irwin, J. Champagne, P. G. Guerra and M. Dubuc, "Cryoballoon ablation of pulmonary veins for paroxysmal atrial fibrillation: first results of the North American Arctic Front (STOP AF) pivotal trial," Journal of the American College of Cardiology 61, no. 16, pp. 1713-1723, 2013.

[18] R. Casado-Arroyo, G.-B. Chierchia, G. Conte, M. Levinstein, J. Sieira, M. Rodriguez-Mañero and G. D. Giovanni, "Phrenic nerve paralysis during cryoballoon ablation for atrial fibrillation: a comparison between the first-comparison between the first-and second-generation balloon," Heart Rhythm 10, no. 9 , pp. 1318-1324, 2013.

[19] S. Y. Sarairah, B. Woodbury, N. Methachittiphan, D. M. Tregoning, A. R. Sridhar and N. Akoum, "Esophageal thermal injury following cryoballoon ablation for atrial fibrillation," Clinical Electrophysiology 6, no. 3, pp. 262-268., 2020.

[20] N. Kulkarni, W. Su and R. Wu, "How to prevent, detect and manage complications caused by cryoballoon ablation of atrial fibrillation," Arrhythmia & Electrophysiology Review 7, no. 1, p. 18, 2018.

[21] Y. Xia, B. Liu, P. Ye and B. Xu, "Thermal field and tissue damage analysis of cryoballoon ablation for atrial fibrillation," Applied Thermal Engineering. 142, p. 524–529, 2018.

[22] A. A. Gage, J. M. Baust and J. G. Baust, "Experimental cryosurgery investigations in vivo," Cryobiology 59, no. 3 , pp. 229-243., 2009.

[23] P. Crosetto, S. Deparis, G. Fourestey and A. Quarteroni, "Parallel algorithms for fluid-structure interaction problems in haemodynamics."," SIAM Journal on Scientific Computing 33, no. 4 , pp. 1598-1622., 2011.

[24] A. M. Hess, Simulation and Design of Soft Robotic Swimmers with Artificial Muscle, Michigan State University, 2019.

[25] Z. Lin, A. Hess, Z. Yu, S. Cai and T. Gao, "A fluid–structure interaction study of soft robotic swimmer using a fictitious domain/active-strain method," Journal of Computational Physics 376 , pp. 1138-1155., 2019.

[26] A. W. Bergersen, A. Slyngstad, S. Gjertsen, A. Souche and K. Valen-Sendstad, "turtleFSI: A robust and monolithic FEniCS-based fluid-structure interaction solver," Journal of Open Source Software 5, no. 50, p. 2089, 2020.

[27] D. Farnell, T. David and D. Barton, "Numerical simulations of a filament in a flowing soap film," International Journal for Numerical Methods in Fluids 44, no. 3, pp. 313-330., 2004.

[28] C. S. Peskin, "Numerical analysis of blood flow in the heart," Journal of computational physics 25, no. 3, pp. 220-252., 1977.

[29] L. Zhu and C. S. Peskin, "Simulation of a flapping flexible filament in a flowing soap film by the immersed boundary method," Journal of Computational Physics 179, no. 2, pp. 452-468., 2002.

[30] C. D. Eggleton and A. S. Popel, "Large deformation of red blood cell ghosts in a simple shear flow," Physics of fluids 10, no. 8 , pp. 1834-1845., 1998.

[31] D. Kamensky, M.-C. Hsu, D. Schillinger, J. A. Evans, A. Aggarwal, Y. Bazilevs, M. S. Sacks and T. J. Hughes, "An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves," Computer methods in applied mechanics and engineering 284 , pp. 1005-1053, 2015.

[32] C. Kadapa, W. G. Dettmer and D. Perić, "A fictitious domain/distributed Lagrange multiplier based fluid–structure interaction scheme with hierarchical B-spline grids," Computer Methods in Applied Mechanics and Engineering 301, pp. 1-27, 2016.

[33] Z. Yu, "A DLM/FD method for fluid/flexible-body interactions," Journal of computational physics 207, no. 1 , pp. 1-27., 2005.

[34] C. Antoci, M. Gallati and S. Sibilla, "Numerical simulation of fluid–structure interaction by SPH," Computers and Structures 85, p. 879–890, 2007.

[35] S. Chen and G. Doolen, "Lattice boltzmann method for fluid flows," Annu. Rev. Fluid Mech. 30, pp. 329-364, 1998.

[36] X. He and L. Luo, "Lattice boltzmann model for the incompressible navier–stokes equation," Journal of Statistical Physics 88, pp. 927-944, 1997.

[37] H. Huang and X.-Y. Lu, "An ellipsoidal particle in tube Poiseuille flow," Journal of Fluid Mechanics 822, pp. 664-688, 2017.

[38] Q. Wei, X. Yuan-Qing, T. Fang-Bao, T.-X. Gao, X.-Y. Tang and W.-H. Zu, "IB-LBM simulation on blood cell sorting with a micro-fence structure," Bio-Medical Materials and Engineering 24, no. 1, pp. 475-481, 2014.

[39] S.-G. Cai, Computational fluid-structure interaction with the moving immersed boundary method, Doctoral dissertation, Compiègne, 2016.

[40] C. M. Scotti, A. D. Shkolnik, S. C. Muluk and E. A. Finol, "Fluid-structure interaction in abdominal aortic aneurysms: effects of asymmetry and wall thickness," Biomedical engineering 4 , pp. 1-22., 2005.

[41] S. Tada and J. M. Tarbell, "A computational study of flow in a compliant carotid bifurcation–stress phase angle correlation with shear stress," Annals of biomedical engineering 33, pp. 1202-1212., 2005.

[42] J. H. Lee, A. D. Rygg, E. M. Kolahdouz, S. Rossi, S. M. Rett, N. Duraiswamy, L. N. Scotten, B. A. Craven and B. E. Griffith, "Fluid–structure interaction models of bioprosthetic heart valve dynamics in an experimental pulse duplicator," Annals of biomedical engineering 48, pp. 1475-1490., 2020.

[43] T. E. Tezduyar, K. Takizawa, T. Brummer and P. R. Chen, "Space–time Fluid– structure Interaction Modeling of Patient-Specific Cerebral Aneurysms," Int. J. Numer. Methods Biomed. Eng., 27(11), p. pp. 1665–1710. , 2011.

[44] T. R, W. NB, H. N, D. AW, W. AR, H. AD, D. J, F. DP, M. J, Y. GZ and T. SA., "Fluid–structure Interaction Analysis of a Patient-Specific Right Coronary Artery with Physiological Velocity and Pressure Waveforms," Commun. Numer. Methods Eng., 25(5), p. 565–580., 2009.

[45] L. Taelman, J. Degroote, P. Verdonck, J. Vierendeels and P. Segers, "Modeling hemodynamics in vascular networks using a geometrical multiscale approach: numerical aspects," Annals of biomedical engineering 41 , pp. 1445-1458, 2013.

[46] G. Luraghi, F. Migliavacca and J. F. R. Matas, "Study on the accuracy of structural and FSI heart valves simulations," Cardiovascular Engineering and Technology 9, pp. 723-738, 2018.

[47] H. Gao, L. Feng, N. Qi, C. Berry, B. E. Griffith and X. Luo, "A coupled mitral valve—left ventricle model with fluid–structure interaction," Medical engineering and physics 47 , pp. 128-136., 2017.

[48] C. S. Peskin, "Flow patterns around heart valves: a numerical method," Journal of computational physics 10, no. 2, pp. 252-271, 1972.

[49] C. S. Peskin, "Numerical analysis of blood flow in the heart," Journal of computational physics 25, no. 3 , pp. 220-252., 1977.

[50] M. D. M., C. S. Peskin and E. L. Yellin, "Fluid dynamics of the mitral valve: physiological aspects of a mathematical model," American Journal of Physiology-Heart and Circulatory Physiology 242, no. 6, pp. H1095-H1110, 1982.

[51] K. B. Chandran and H. Kim, "Computational mitral valve evaluation and potential clinical applications," Annals of biomedical engineering 43 , pp. 1348-1362, 2015.

[52] E. Oñate, M. Manzan, E. Oñate and M. Manzan, "Stabilization Techniques for Finite Element Analysis of Convection-Diffusion Problems," International Center for Numerical Methods in Engineering, 2000.

[53] K. Hansen, A. Arzani and S. C. Shadden, "Finite element modeling of near-wall mass transport in cardiovascular flows," International journal for numerical methods in biomedical engineering 35, no. e3148, 2019.

[54] V. John and P. Knobloch, "A computational comparison of methods diminishing spurious oscillations in finite element solutions of convection-diffusion equations," p. 122–136, 2006.

[55] R. Codina, "A discontinuity-capturing crosswind-dissipation for the finite element solution of the convection-diffusion equation," Computer Methods in Applied Mechanics and Engineering. 110, p. 325–342, 1993.

[56] W. Kress and P. Lötstedt, "Time step restrictions using semi-explicit methods for the incompressible Navier–Stokes equations," Computer Methods in Applied Mechanics and Engineering. 195 , p. 4433–4447, 2006.

[57] P. Ryzhakov and J. Marti, "A Semi-Explicit Multi-Step Method for Solving Incompressible Navier-Stokes Equations," Applied Sciences. 1 , 2018.

[58] K. Selim, A. Logg and M. Larson, "An Adaptive Finite Element Splitting Method for the Incompressible Navier-Stokes Equations," Computer Methods in Applied Mechanics and Engineering. 209–212, p. 54–65, 2012.

[59] T. Gelhard, G. Lube, M. Olshanskii and J. Starcke, "Stabilized finite element schemes with LBB-stable elements for incompressible flows,," Journal of Computational and Applied Mathematics, vol. 177, p. 243–267., 2005.

[60] D. Boffi, F. Brezzi and M. Fortin, Mixed Finite Element Methods and Applications, 44, 2013.

[61] T. Tezduyar and S. Sathe, "Stabilization Parameters in SUPG and PSPG Formulations," Journal of Computational and Applied Mechanics. 4, p. 71–88, 2003.

[62] J. Peterson, A. Lindsay and F. Kong, "Overview of the incompressible Navier–Stokes simulation capabilities in the MOOSE framework," Advances in Engineering Software. 119, p. 68–92., 2018.

[63] S. Haßler, A. M. Ranno and M. Behr, "Finite-element formulation for advection-reaction equations with change of variable and discontinuity capturing," Computer Methods in Applied Mechanics and Engineering 369, no. 113171, 2020 .

[64] O. Botella and R. Peyret, "Benchmark spectral results on the lid-driven cavity flow," Computers & Fluids. 27 , p. 421–433, 1998.

[65] T. Gao, Y. Tseng and X. Lu, "An improved hybrid Cartesian/immersed boundary method for fluid-solid flows," International Journal for Numerical Methods in Fluids. 55, p. 1189–1211, 2007.

[66] I. Rodriguez, O. Lehmkuhl, M. Soria, S. Gómez, M. Domínguez-Pumar and L. Kowalski, "On the boundary layer development and heat transfer from a sphere atmoderate Reynolds numbers," Tenth International Conference on Computational Fluid Dynamics, p. 9–13, 2018.

[67] T. Fastl, C. Tobon-Gomez, A. Crozier, J. Whitaker, R. Rajani, K. McCarthy, D. Sanchez-Quintana, S. Ho, M. O'Neill, G. Plank, M. Bishop and S. Niederer, "Personalized computational modeling of left atrial geometry and transmural myofiber architecture," Medical Image Analysis. 47 , p. 180, 2018.

[68] J. Lantz, V. Gupta, L. Henriksson, M. Karlsson, A. Persson, C. Carlhäll and T. Ebbers, "Impact of Pulmonary Venous Inflow on Cardiac Flow Simulations: Comparison with In Vivo 4D Flow MRI," Annals of Biomedical Engineering. 47, p. 413, 2019.

[69] A. Klein, W. Stewart, J. Bartlett, G. Cohen, F. Kahan, G. Pearce, K. Husbands, A. Bailey, E. Salcedo and D. Cosgrove, "Effects of mitral regurgitation on pulmonary venous flow and left atrial pressure: An intraoperative transesophageal echocardiographic study," J Am Coll Cardiol. 20, p. 1345–1352., (992.

[70] H. Pennes, "Analysis of tissue and arterial blood temperatures in the resting human forearm," J Appl Physiol. 1 , p. 93–122., 1948.

[71] T. Marinko, J. Dolenc and C. Bilban-Jakopin, "Cardiotoxicity of concomitant radiotherapy and trastuzumab for early breast cancer," Radiology and oncology 48, vol. no. 2, p. 105. , 2014.

[72] O. Smiseth, C. Thompson, K. Lohavanichbutr, H. Ling, J. Abel, R. Miyagishima, S. v. Lichtenstein and J. Bowering, "The pulmonary venous systolic flow pulse—its origin and relationship to left atrial pressure," J Am Coll Cardiol. 34, p. 802–809, 1999.

[73] A. Gonzalo, M. García-Villalba, L. Rossini, E. Durán, D. Vigneault, P. Martínez-Legazpi and O. Flores, "Non-Newtonian Patient-specific Simulations of Left Atrial Hemodynamics," bioRxiv, pp. 2021-06., 2021.

[74] X. Morales, J. Mill, G. Delso, F. Loncaric, A. Doltra, X. Freixa, M. Sitges, B. Bijnens and O. Camara, "4D flow magnetic resonance imaging for left atrial haemodynamic characterization and model calibration," In Statistical At Atlases and Computational Models of the Heart. M&Ms and EMIDEC Challenges: 11th International Workshop, STACOM 2020, Springer International Publishing, pp. 156-165., 2021.

[75] J. Egger, M. Gall, J. Wallner, P. Boechat and A. Hann, "HTC Vive MeVisLab integration via OpenVR for medical applications," PLOS ONE 12(3): e0173972., 2017.

[76] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J.-C. Fillion-Robin, S. Pujol and C. Bauer, "3D Slicer as an image computing platform for the Quantitative Imaging Network," Magnetic resonance imaging 30, vol. no. 9 , pp. 1323-1341., 2012.

[77] B. Parvez, V. Pathak, C. Schubert and M. Wood, "Comparison of Lesion Sizes Produced by Cryoablation and Open Irrigation Radiofrequency Ablation Catheters," Journal of Cardiovascular Electrophysiology. 19, p. 528–534., 2008.

[78] R. Müssig, M. Heinke and J. Hörth, "Cryoballoon model and simulation of catheter ablation for pulmonary vein isolation in atrial fibrillation, Current Directions in Biomedical Engineering. 4," p. 473–475., 2018.

[79] J. Baust, A. Robilotto, K. Snyder, R. v. Buskirk and J. Baust, "Evaluation of a new epicardial cryoablation system for the treatment of Cardiac Tachyarrhythmias," Research Article Trends Med. 18 , p. 1–9, 2018.

[80] J. Ghosh and M. McGuire, "Atrial flow dynamics as a determinant of tissue temperature during balloon cryoablation," EP Europace. 20, p. f451–f457., 2018.

[81] J. Baust, A. Robilotto, P. Guerra, K. Snyder, R. v. Buskirk, M. Dubuc and J. Baust, "Assessment of a novel cryoablation device for the endovascular treatment of cardiac tachyarrhythmias," SAGE Open Med., vol. 6, 2018.

[82] Z. Yu, X. Shao and A. Wachs, "A fictitious domain method for particulate flows with heat transfer," Journal of Computational Physics 217, vol. no. 2, pp. 424-452, 2006.

[83] Y. Yu, "Numerical Methods for Fluid-Structure Interaction: Analysis and Simulations," PhD diss., Brown University, 2014.

[84] X. Wang and L. T. Zhang, "Interpolation functions in the immersed boundary and finite element methods," Computational Mechanics 45, no. 4 , pp. 321-334, 2010.

[85] X. Yang, Z. L. Xing Zhang and G.-W. He, "A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations," Journal of Computational Physics 228, no. 20, pp. 7821-7836, 2009.

[86] [Online]. Available: https://docs.python.org/3/library/argparse.html.

[87] S. Turek and J. Hron, "Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow," Springer Berlin Heidelberg, 2006.

[88] F.-B. Tian, H. Dai, H. Luo, J. F. Doyle and B. Rousseau, "Fluid–structure interaction involving large deformations: 3D simulations and applications to biological systems," Journal of computational physics, no. 258, pp. 451-469, 2014.

[89] Z. Yu, Y. Wang and X. Shao, "Numerical simulations of the flapping of a three-dimensional flexible plate in uniform flow," Journal of Sound and Vibration 331, no. 20 , pp. 4448-4463, 2012.

[90] É. d. Madron, V. Chetboul and C. Bussadori, Clinical Echocardiography of the Dog and Cat, 2015, p. 1–348..

[91] G. D'Angelo, E. Moro, G. Nicolosi, V. Dall'Aglio, R. Mimo, S. Mangano and D. Zanuttini, "Color Doppler identification of early diastolic turbulence in the left atrium in patients with mitral valve insufficiency: persistence of regurgitation or inertia phenomenon?," Giornale Italiano Di Cardiologia. 20 , p. 700–704, 1990.

[92] S. Singh and S. Kumar, "A study on the effect of metabolic heat generation on biological tissue freezing," ScientificWorldJournal, 2013.

[93] "Ablation Products for Atrial Fibrillation - CryoConsole Cardiac Cryoablation System | Medtronic, (n.d.).," [Online]. Available: https://www.medtronic.com/us-en/healthcare-professionals/products/cardiac-rhythm/ablation-atrial-fibrillation/cryoconsole-cardiac-cryoablation.html.

[94] N. V. C. Gillespie, A. Lin and B. Knight, "Ice formation in the left mainstem bronchus during cryoballoon ablation for the treatment of atrial fibrillation," Heart Rhythm. 13 , p. 814–815, 2016.

[95] S. Bordignon, A. Fürnkranz, B. Schmidt and K. Chun, "Remaining ice cap on second-generation cryoballoon after deflation," Circulation: Arrhythmia and Electrophysiology. 5, 2012.

[96] A. Odutayo, C. X. Wong, A. J. Hsiao, S. Hopewell, D. G. Altman and C. A. Emdin, "Atrial fibrillation and risks of cardiovascular disease, renal disease, and death: systematic review and meta-analysis," bmj 354, 2016.

APPENDIX A : EFFECTS OF STABILIZATION SCHEMES

First, we consider the **LA+CB** case without the LSIC and PSPG stabilization schemes for the momentum equation. The results for both lesion area (**Figure A.1a**) and power absorbed across the CB are similar to that shown in **Figure 4.4a** and **e**. Subsequently, the case without LSIC and PSPG stabilization for the momentum equation is utilized to investigate the isolated effects of crosswind stabilization on the energy equation (**Figure A.1b**) and the momentum equation (**Figure A.1c**). When the crosswind stabilization is disabled for the energy equation**,** localized undershoots (-132^0C) /overshoots (216^0C) are observed in the temperature field in the vicinity of the CB that is outside the problem's temperature range (between -70^0C to 37^0C, see **Figure A.1b)**. On the other hand, when crosswind stabilization is disabled for the momentum equation, we observe an unsteady lesion, and the position of the lesion oscillates in way that is non-physical even though temperature gradients downstream of the CB are better captured without undershooting/overshooting of the temperature (**Figure A.1c**). Based on these simulations, we conclude that the crosswind stabilizing operator (Eq. (3.16)) is necessary and serves mainly to capture sharp gradients in the field variable.

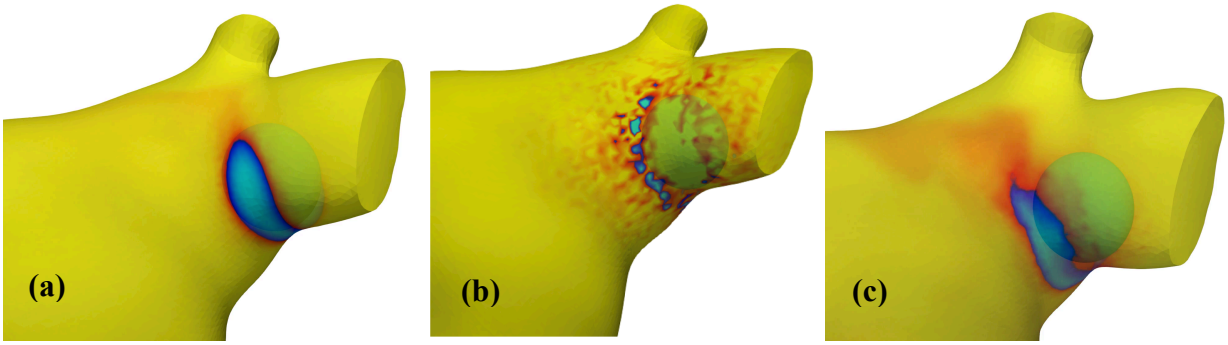

Figure A.1 Effect of crosswind stabilization: (a) LA+CB case without PSPG and LSIC stabilization for the momentum equation (b) LA+CB case without PSPG and LSIC stabilization for the momentum equation and without crosswind stabilization for the energy equation (c) LA+CB case without PSPG, LSIC and crosswind stabilization for the momentum equation.

APPENDIX B : CRYOBALLOON POSITIONING

Under identical simulation conditions (section 4.1), a total of 12 CB positions (**LA+CB** case + P1 - P11) are simulated to study the effects of CB position on lesion formation. The exact positions of the CB (center O) with respect to 3 constant points A, B and C on the LA surface are given in **Table B.1**. Here, we also introduce α which is the angle between the CB's minor axis and OA (see **Figure B.1**). Note that point A is located centrally at the RIPV opening, and point B and C are selected arbitrary at the top and side of the RIPV where it merges with the LA. Positioning the CB in this manner provides a range of different balloon-tissue contact areas $\boldsymbol{A_c}$ to analyze incomplete occlusion in case of cryotherapy.

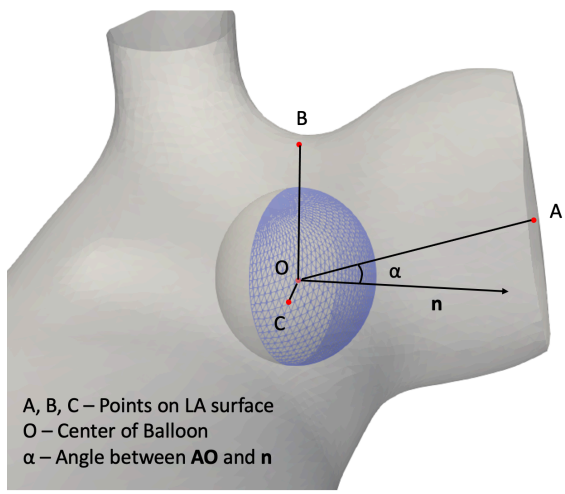

Figure B.1: Geometrical positioning of the CB (center O) in the RIPV ostium. Point A is located centrally at the RIPV opening. n is normal to the plane passing through the two major axes of the CB and the angle between n and OA is defined as α. Point B and C are selected arbitrary at the top and side of the RIPV where it merges with the LA.

Table B.1: The distance OA, OB, OC, angle α and balloon tissue contact area A_c for all CB positions.

CB position	OA (mm)	OB (mm)	OC (mm)	α (deg)	A_c (mm^2)
P1	31.69	16.25	15.36	7.37	252.9
P2	33.95	18.67	16.66	20.16	116.7
P3	30.67	21.88	16.31	14.27	207
P4	30.11	17.43	14.81	9.18	223.4

Table B.1 (cont'd)

P5	29.67	21.30	15.72	14.30	213.4
P6	28.45	23.40	16.52	16.26	444.8
P7	29.71	19.74	14.94	19.85	193.2
P8	30.14	20.74	15.82	10.94	187
P9	31.14	23.18	16.66	11.79	259
P10	30.52	20.54	16.10	4.21	167.4
P11	31.09	19.67	14.81	14.52	176