Title

Fluid--Structure Interaction : : Physiologic Simulation of Pulsatile Ventricular Assist Devices using Isogeometric Analysis

Author

Long, Christopher Curtis

Publication Date

2013

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Fluid–Structure Interaction: Physiologic Simulation of Pulsatile Ventricular Assist Devices using Isogeometric Analysis

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Engineering Science with Specialization in Computational Science

by

Christopher Curtis Long

Committee in charge:

Professor Alison L. Marsden, Chair
Professor Yuri Bazilevs, Co-Chair
Professor David J. Benson
Professor Juan Carlos del Álamo
Professor Daniel M. Tartakovsky

2013

The dissertation of Christopher Curtis Long is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Co-Chair

Chair

University of California, San Diego

2013

DEDICATION

To my wife, Kristin Barnes. Your unending patience has made this possible; this is your work as much as mine.

Thank you.

EPIGRAPH

If people do not believe that
mathematics is simple, it is
only because they do not realize
how complicated life is.
—John von Neumann

And if the computer gives you any back
talk, pour some well-sugared office
coffee into its evil little silicon brain.
—Ed Abbey

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would first like to thank my two advisors, Alison Marsden and Yuri Bazilevs, for their support, expertise, and encouragement throughout this project. Their patient instruction during my stay will benefit me for years to come. I would also like to thank all of my lab members whom I shared my time with: Mahdi Esmaily-Moghadam, Abhay Ramachandra, Sethuraman Sankaran, Weiguang Yang, Dibyendu Sengupta, Jessica Oakes, Matthew Bockman, Ethan Kung, Daniele Schiavazzi, Ming-Chen Hsu, Ido Akkerman, and Shahrouz Alimohammadi. All have offered me both technical knowledge and moral/social support, for which I am very grateful. I would also like to thank Jeff Feinstein and Mikhail Shashkov, who have also provided invaluable time and expertise.

I would also like to acknowledge all of the funding agencies which have provided financial or computational support. The computational support from XSEDE, NICS, San Diego Supercomputer Center, and Oak Ridge National Laboratory made much of this work possible. Funding from AFOSR, Los Alamos National Laboratory, and Burroughs Wellcome Fund have provided for my tuition and stipend, and I am grateful for their faith and support in this project. I would also like to acknowledge the University of California, which has provided support through teaching assistantships, facilities, and educational opportunities.

For working patiently with me, providing valuable insights, and allowing me to reprint published material as necessary, I would like to extend my gratitude to all of my co-authors: Jeff Feinstein, Ido Akkerman, Ming-Chen Hsu, David Benson, and Mikhail Shaskov. The publishers at Wiley and Springer have granted licenses to reuse published material in this dissertation, and I also acknowledge this generosity. Below is a list of the utilized published and/or submitted works:

Chapter 2, in full, is a reprint of the material as it appears in the article "Fluid–structure interaction simulations of the Fontan procedure using variable wall properties", which was published in May 2012 in Volume 28, Issue 5 of **International Journal for Numerical Methods in Biomedical Engineering**. Authors include: C.C. Long, M.-C. Hsu, J.A. Feinstein, and A.L. Marsden.

Chapter 4, in full, is a reprint of the material as it was submitted in the article “Isogeometric Analysis of Lagrangian Hydrodynamics: Axisymmetric Formulation in the rz-Cylindrical Coordinates”, submitted to **Journal of Computational Physics**. Authors include: Y. Bazilevs, C.C. Long, I. Akkerman, D.J. Benson, and M.J. Shaskov.

Chapter 5, in full, is a reprint of the material as it appears in the article “Fluid–Structure Intercation Simulation of Pulsatile Ventricular Assist Devices,” published online April 2013 in **Computational Mechanics**. Authors include: C.C. Long, A.L. Marsden, and Y. Bazilevs.

And to my wife, Kristin Barnes, the love of my life. The most important support of all has undoubtedly been her unshakable faith in me. Her smile has made many difficult days easier to bear, and she has made my best days in San Diego possible.

VITA

2005	B. E. in Mechanical Engineering and Mathematics *cum laude*, Vanderbilt University
2005-2007	Graduate Research Assistant, University of Arizona, Tucson
2007	M. S. E. in Mechanical Engineering, University of Arizona, Tucson
2008-2009	Software Development Engineer, Ziebel Inc., Tulsa
2009-2013	Graduate Research Assistant, University of California, San Diego
2013	Ph. D. in Engineering Science with Specialization in Computational Science, University of California, San Diego

PUBLICATIONS

C.C. Long, V.V. Krivets, J.A. Greenough, J.W. Jacobs, “Shock tube experiments and numerical simulation of the single-mode, three-dimensional Richtmyer-Meshkov instability,” *Physics of Fluids*, 21(11), 2009.

C.C. Long, M.-C. Hsu, Y. Bazilevs, J.A. Feinstein, A.L. Marsden, “Fluid-structure interaction simulations of the Fontan procedure using variable wall properties,” *International Journal of Numerical Methods in Biomedical Engineering*, 28(5), 2012

C.C. Long, A.L. Marsden, Y. Bazilevs, “Fluid–structure interaction simulation of pulsatile ventricular assist devices,” *Computational Mechanics*, published online April 2013.

Y. Bazilevs, C.C. Long, I. Akkerman, D.J. Benson, M.J. Shashkov, “Isogeometric analysis of Lagrangian hydrodynamics: Axisymmetric formulation in the *rz*-cylindrical coordinates,” submitted for publication at *Journal of Computational Physics*

A.L. Marsden, Y. Bazilevs, C.C. Long, M. Behr, “Recent advances in computational methodology for simulation of mechanical circulatory assist devices,” submitted for publication at *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*

ABSTRACT OF THE DISSERTATION

Fluid–Structure Interaction: Physiologic Simulation of Pulsatile Ventricular Assist Devices using Isogeometric Analysis

by

Christopher Curtis Long

Doctor of Philosophy in Engineering Science with Specialization in Computational Science

University of California, San Diego, 2013

Professor Alison L. Marsden, Chair
Professor Yuri Bazilevs, Co-Chair

This dissertation describes key contributions in Fluid–Structure Interaction (FSI) simulations using Isogeometric Analysis (IGA). We first describe the state of the art in both FSI and IGA. Several new contributions to these fields are developed, which enable novel simulations in the fields of cardiovascular fluid flow and shock hydrodyanmics.

In our first application, a hyperelastic material model is integrated with the FSI solver and applied to bloodflow through a Fontan surgical conduit. These conduits are created in patients with otherwise fatal congenital heart defects. The junction is comprised of the native venous vessels which return blood to the heart, the vena cava, which are anastamosed directly to the pulmonary arteries, resulting in a junction with several material regions. A hyperelastic material model is applied to the disparate native tissues, and FSI simulations are performed which respect unique material regions. Parameters of clinical interest, such as efficiency, flow split, and wall shear are presented. FSI was required to accurately predict wall shear, but did not significantly impact other hemodynamic properties.

In a second application, we simulate physiologic operation of a Pulsatile Ventricular Assist Device (PVAD). Patients with severe congenital or acquired heart diseases may require cardiac transplantation to ensure survival. In critical cases, a PVAD is implanted in a bridge-to-transplant scenario. This device is a mechanical displacement pump which delivers blood systemically to the patient. PVADs exhibit large structural deformations, creating significant modeling difficulties. IGA is used to model the structural membrane, which is a key component of PVAD performance. Physiologic simulations are carried out for the first time, and results are analyzed for thrombogenic risks, such as residence time. Preliminary results indicate pediatric models have higher thrombotic risks than adult models, matching clinical experience.

The use of IGA without FSI is also explored, and is applied to several classic shock hydrodynamics problems in the axisymmetric frame. The retention of radially symmetric solutions are crucial in this field, and difficult to obtain. Simulations of the Noh problem, Coggeshall problem, and a ‘multi-material’ problem are presented, and in each case IGA exhibits the best known symmetric performance.

Chapter 1

Fundamentals of Fluid-Structure Interaction

In this section, we introduce the basic aspects of fluid structure interaction (FSI) used throughout this dissertation. The basic formulation of moving domain fluid problems is derived, along with the basic equations governing the structural domain. The coupling of the Fluid–Structure problem is also introduced, and further explored in Chapter 5 to solve the coupled motion between domains with non-matching discretizations and inconsistent shape function definitions.

1.1 Introduction

Fluid-structure Interactions may be solved using a strongly coupled finite element approach. For this approach to work, we require a robust finite element solver for both the fluid and structural components of the problem. For many fluid problems, including those presented in Chapters 2 and 5, the flow is advection dominated, which requires a stabilized finite element method to be used. [37, 85, 169, 168, 161, 89, 170, 79] These methods have been well-studied and tested, and are known to be stable and convergent for both advective and diffusive flows.

More recently, the variational multiscale method has been introduced [80], in which the stabilization terms on the fluid formulation include aspects of subgridscale modeling. The FSI problems presented herein use a residual-based VMS method, or RBVMS method [15]. This method has been shown to work well for both laminar and turbulent flows.

In standard Finite Element Methods (FEM), and herein, the physical domain is denoted as Ω. The physical domain is discretized into a finite number of subdomains called *elements*, which occupy the physical domain, Ω, and comprise the computational domain, denoted $\hat{\Omega}$ or Ω^h. In this work, these elements are tetrahedral unless otherwise noted. The formal definition of $\hat{\Omega}$ is as follows

$$\hat{\Omega} = \bigcup_{e=1}^{N_{el}} \hat{\Omega}^e, \tag{1.1}$$

where e is the element number index, N_{el} is the total number of elements used in the discretization, and $\hat{\Omega}^e$ is the volume occupied by a single element, e.

For moving domain problems, it is clear that $\hat{\Omega}|_{t=0} \neq \hat{\Omega}|_{t>0}$. By our definition, this implies that the individual nodal points and elements must move to accomodate the motion of the computational domain. Furthermore, it introduces the concept of a *reference* configuration, and a *current* configuration. The current configuration is denoted by Ω_t, or $\hat{\Omega}_t$, where t is the current time index. The reference configuration is fixed, and is the special case when $t = 0$. It is denoted by Ω_0 or $\hat{\Omega}_0$, as appropriate.

If the mesh motion is defined to be equal to the advective velocity of the fluid, we refer to the problem as *Lagrangian.* In this case, the convective terms drop out of the fluid formulation. Several examples of Lagrangian fluid flow simulations are presented in Chapter 4. The case whereby the mesh is rigidly fixed for all time (e.g., the reference and current configurations are equivalent $\forall t$) is referred to as an *Eulerian* formulation, and is the standard for most fluid dynamics problems which do not incorporate FSI. A third option exists that is convenient for many FSI problems, in which the mesh motion does not exactly match the advective velocity of the fluid. In this case, the reference and current configurations are not equivalent, and convective terms in the Navier–Stokes equations persist. The resulting formulation of the Navier–Stokes equations is referred to as the *Arbitrary Lagrangian-Eulerian* (ALE) form, and will be derived in Sec. 1.2.1.

Chapters 2 and 5 present examples of moving domain FSI problems. For these cases, we couple the ALE fluid equations with the RBVMS framework. This approach is called the ALE-VMS method [153, 24].

1.2 Fluid Formulation in Moving Domains

In this section, we derive the governing equations of fluid motion in a moving domain, and briefly present the RBVMS-stabilized formulation of an FEM based fluid solver. A full description of the stabilization on these equation may be found in the literature [28, 153, 24].

1.2.1 ALE Navier–Stokes

We first introduce the traditional strong form of the Navier–Stokes equations, as will be used throughout this dissertation:

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} - \sigma) - \rho \mathbf{f} = \mathbf{0}, \tag{1.2}$$

$$\nabla \cdot \mathbf{u} = \mathbf{0}, \tag{1.3}$$

where $\mathbf{u}$ is the unknown fluid velocity, ρ is the fluid density, $\mathbf{f}$ is a body force acting on the fluid, ∇ is the traditional differential operator, and σ is the fluid stress tensor, as shown

$$\sigma = -p\delta_{ij} + 2\mu\epsilon_{ij}, \tag{1.4}$$

where p is pressure, δ_{ij} is the Kronecker delta function, μ is fluid viscosity, and ϵ_{ij} is the symmetric differential operator acting on fluid velocity defined as follows:

$$\epsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}). \tag{1.5}$$

We wish to pose this in a modified form amenable to discrete analysis for the moving domain. An integration of the momentum equation (Eq. 1.2) with respect to the moving domain Ω_t, and for an arbitrary time period such that $0 \leq t \leq T$, yields:

$$\int_0^T \int_{\Omega_t} \frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} - \sigma) - \rho \mathbf{f} \mathrm{d}\Omega \mathrm{d}t = \mathbf{0}. \tag{1.6}$$

For convenience, the result is rewritten using standard index notation:

$$\int_0^T \int_{\Omega_t} (\rho u_i)_{,t} + \left(\rho u_i u_j - \sigma_{ij}\right)_{,j} - \rho f_i \mathrm{d}\Omega \mathrm{d}t = \mathbf{0}. \tag{1.7}$$

A *space-time Piola transform* is performed on Eq. 1.7, which effectively changes variables from $\Omega_t \rightarrow \Omega_0$. This transform is well-known in structural mechanics [178], and the space-time variant has been explored for fluids in [16]. Two parameters, $\hat{\gamma}_0$ and $\hat{\gamma}_I$ are defined as:

$$\left\{ \begin{array}{c} \hat{\gamma}_0 \\ \hat{\gamma}_I \end{array} \right\} = \left\{ \begin{array}{c} \hat{J}\gamma_0 \\ \hat{J}\hat{F}_{Ii}^{-1}(\gamma_i - \gamma_0 \hat{u}_i) \end{array} \right\}, \tag{1.8}$$

where γ_0 and $\gamma_\mathbf{i}$ are parameters to be chosen from Eq. 1.7, and $\hat{\mathbf{u}}$ is the domain velocity. The value $\hat{J}$ is defined as $\hat{J} = \det(\hat{\mathbf{F}})$, where $\hat{\mathbf{F}}$ is the *deformation gradient*, which provides a relationship between the reference and current domains such that:

$$\hat{\mathbf{F}} = \mathbf{I} + \frac{\partial \hat{\mathbf{y}}}{\partial \hat{\mathbf{x}}}, \tag{1.9}$$

where $\hat{\mathbf{y}}$ is a time-dependent vector field which contains the displacement of the reference domain. By choosing variables $\gamma_0 = \rho u_i$, and $\gamma_j = \rho u_i u_j - \sigma_{ij}$, we can rewrite Eq. 1.7 as shown:

$$\int_0^T \int_{\Omega_0} (\hat{J}\rho u_i)_{,t}|_{\hat{\mathbf{x}}} + \left(\hat{J}\left(\rho u_i(u_j - \hat{u}_j) - \sigma_{ij}\right)\hat{F}_{Jj}^{-1}\right)_{,J} - \hat{J}\rho f_i \mathrm{d}\Omega \mathrm{d}t = \mathbf{0}. \tag{1.10}$$

This change of variable permits a permutation of integrands, as the domain Ω_0 has no time dependence. That is, we can rewrite it such that $\int_T \int_{\Omega_0} \cdot \mathrm{d}\Omega \mathrm{d}t = \int_{\Omega_0} \int_T \cdot \mathrm{d}t \mathrm{d}\Omega$. Furthermore, since the relation holds for all time, the time integration term can be eliminated as shown:

$$\int_{\Omega_0} (\hat{J}\rho u_i)_{,t}|_{\hat{\mathbf{x}}} + \left(\hat{J}\left(\rho u_i(u_j - \hat{u}_j) - \sigma_{ij}\right)\hat{F}_{Jj}^{-1}\right)_{,J} - \hat{J}\rho f_i \mathrm{d}\Omega = 0. \tag{1.11}$$

A second Piola transform can then be used to change variables from $\Omega_0 \rightarrow \Omega_t$. In this instance, a purely spatial tranform may be used, where $\hat{\gamma} = \hat{J}\hat{\mathbf{F}}^{-1}\gamma$. The resulting form is:

$$\int_{\Omega_t} \frac{1}{\hat{J}}(\hat{J}\rho u_i)_{,t}|_{\hat{\mathbf{x}}} + \left(\rho u_i(u_j - \hat{u}_j) - \sigma_{ij}\right)_{,j} - \rho f_i \mathrm{d}\Omega = 0. \tag{1.12}$$

After reinstating vector notation, the equations may be localized as:

$$\frac{1}{\hat{J}}\left(\frac{\partial \hat{J}\rho\mathbf{u}}{\partial t}\right)|_{\hat{\mathbf{x}}} + \nabla\cdot(\rho\mathbf{u}\otimes(\mathbf{u}-\hat{\mathbf{u}})-\sigma)-\rho\mathbf{f}=\mathbf{0}. \tag{1.13}$$

Expansion of the derivative terms yields:

$$\frac{\rho}{\hat{J}}\frac{\partial\hat{J}}{\partial t}|_{\hat{\mathbf{x}}}\mathbf{u}+\rho\frac{\partial\mathbf{u}}{\partial t}|_{\hat{\mathbf{x}}}+\rho(\mathbf{u}\nabla\cdot(\mathbf{u}-\hat{\mathbf{u}})+(\mathbf{u}-\hat{\mathbf{u}})\cdot\nabla\mathbf{u})-\nabla\cdot\sigma-\rho\mathbf{f}=\mathbf{0}, \tag{1.14}$$

and we may use Eq. 1.3 and the identity

$$\frac{\partial\hat{J}}{\partial t}|_{\hat{\mathbf{x}}}=\hat{J}\nabla\cdot\hat{\mathbf{u}} \tag{1.15}$$

to reduce Eq. 1.14 to its final form, as shown:

$$\rho\left(\frac{\partial\mathbf{u}}{\partial t}|_{\hat{\mathbf{x}}}+(\mathbf{u}-\hat{\mathbf{u}})\cdot\nabla\mathbf{u}-\mathbf{f}\right)-\nabla\cdot\sigma=\mathbf{0}. \tag{1.16}$$

Equation 1.16, coupled with Eq. 1.3 are the standard form of the Navier–Stokes equations in a moving domain, and are the standard for use in the ALE framework. We note that the ∇ operator acts with respect to the *current* configuration, unless otherwise denoted.

1.2.2 Fluid RHS

The strong form of the Navier–Stokes equations, as given in Eqs. 1.16 and 1.3, are generally more familiar, but finite elements require the formulation to be posed in the variational, or weak form. A basic familiarity with the finite element method (FEM) is presumed. For more information on basis functions, natural and essential boundary conditions, parametric spaces and mappings, differentiability requirements, and IEN arrays, the reader is referred to Hughes [81]. To pose the ALE equations in the weak form, we define solution spaces S_u and S_p, as well as the variational spaces V_u and V_p. The solution spaces are sets of trial functions, and are comprised of all functions that meet the differentiability requirements in Ω. The variational space is defined in Ω in a similar way. The boundary of the domain Ω is denoted with Γ. We further assert that Γ_h refers to the portion of the boundary with natural boundary conditions, and that Γ_g referes to the portion of the boundary with essential boundary conditions. It is required that $\Gamma = \Gamma_h \cup \Gamma_g$, and further that $\Gamma_h \cap \Gamma_g = \emptyset$. We then build the essential boundary conditions into the definition of our function space:

$$u = g \;\text{ on }\; \Gamma_g, \forall u \in S, \tag{1.17}$$

$$w = 0 \;\text{ on }\; \Gamma_g, \forall w \in V, \tag{1.18}$$

where u and w are members of the function sets S and V respectively, and g is a given function.

Given the above definitions, we can pose the variational form of the incompressible ALE Navier–Stokes equations. Find $\mathbf{u} \in S_u$ and $p \in S_p$, such that for all $\mathbf{w} \in V_u$ and $q \in V_p$, the following is satisfied:

$$\begin{aligned}\int_{\Omega_t} \mathbf{w} \cdot \rho \left(\frac{\partial \mathbf{u}}{\partial t}|_{\hat{x}} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla \mathbf{u} - \mathbf{f} \right) \mathrm{d}\Omega + \int_{\Omega_t} \epsilon(\mathbf{w}) : \sigma(\mathbf{u}, p) \mathrm{d}\Omega \\ - \int_{\Gamma_{ht}} \mathbf{w} \cdot \mathbf{h} \mathrm{d}\Gamma - \int_{\Omega_t} q \nabla \cdot \mathbf{u} \mathrm{d}\Omega = \mathbf{0}.\end{aligned} \tag{1.19}$$

And finally, we discretize:

$$\begin{aligned}\int_{\Omega_t^h} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t}|_{\hat{x}} + (\mathbf{u}^h - \hat{\mathbf{u}}^h) \cdot \nabla \mathbf{u}^h - \mathbf{f}^h \right) \mathrm{d}\Omega \\ + \int_{\Omega_t^h} \epsilon(\mathbf{w}^h) : \sigma(\mathbf{u}^h, p^h) \mathrm{d}\Omega - \int_{\Gamma_{h_t}^h} \mathbf{w}^h \cdot \mathbf{h}^h \mathrm{d}\Gamma - \int_{\Omega_t^h} q^h \nabla \cdot \mathbf{u}^h \mathrm{d}\Omega = \mathbf{0}.\end{aligned} \tag{1.20}$$

The superscript h denotes the local element-level evaluation. Local integration is carried out using Gauss quadrature, and global integration is then performed as a sum over the total number of discrete elements. That is, $\Omega_t \approx \Omega_t^h = \bigcup_{e=1}^{N_{el}} \Omega_e$, where Ω_e denotes the domain occupied by a particular element, e is an element index, and N_{el} is the number of elements in the discretized domain. The integration over Ω_t^h may be therefore written as:

$$\int_{\Omega_t^h} \cdot \mathrm{d}\Omega \approx \sum_{e=1}^{N_{el}} \int_{\Omega_e} \cdot \mathrm{d}\Omega \tag{1.21}$$

where the integration on a single element domain Ω_e is carried out with Gaussian quadrature, as described in [81]. Furthermore, the respective variables $\mathbf{u}^h, p^h, \mathbf{w}^h$, and q^h are element-level values which can be written as:

$$\mathbf{u}^h = \sum_{N_{bf}} \mathbf{u}_A N_A, \tag{1.22}$$

$$\mathbf{w}^h = \sum_{N_{bf}} \mathbf{w}_A N_A, \tag{1.23}$$

$$p^h = \sum_{N_{bf}} p_A N_A, \tag{1.24}$$

$$q^h = \sum_{N_{bf}} q_A N_A. \tag{1.25}$$

The variables $\mathbf{u}_A$ and p_A are trial functions for the velocity and pressure solutions, respectively, while the variables $\mathbf{w}_A$ and q_A are the respective test functions. The N_A terms refer to the basis function evaluations at the same point, and the parameter N_{bf} refers to the number of basis functions used. From henceforth, summation is implied on repeated indices to simplify notation. In this formulation, the test functions have no time-dependence.

To compute the right-hand-side (RHS) residual vector of the N-S equations, we simply plug prospective solution values for $\mathbf{u}^h$ and p^h into Eq.1.20. As shown, we desire the result to be $\mathbf{0}$ throughout the domain. A non-zero value indicates that the prospective solution is not exactly correct, and this computed value is the *residual* RHS, which we would like to minimize. Two residual vectors are formed, one for the momentum equation, and the other for the contiuity equation. Using definitions given in Section 1.2.1, we more formally describe the RHS residual vectors as follows:

$$\mathbf{R}_A^{cont} = \int_{\Omega_t} N_A \nabla \cdot \mathbf{u}^h \mathrm{d}\Omega \tag{1.26}$$

$$\begin{aligned}\mathbf{R}_{A_i}^{mom} &= \int_{\Omega_t} N_A \hat{\mathbf{n}}_i \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t}|_{\hat{x}} + (\mathbf{u}^h - \hat{\mathbf{u}}^h) \cdot \nabla \mathbf{u}^h - \mathbf{f}^h \right) \mathrm{d}\Omega \\ &+ \int_{\Omega_t} \epsilon(N_A \hat{\mathbf{n}}_i){:}\sigma(\mathbf{u}^h, p^h) \mathrm{d}\Omega - \int_{\Gamma_{h_t}} N_A \hat{\mathbf{n}}_i \cdot \mathbf{h}^h \mathrm{d}\Gamma \end{aligned} \tag{1.27}$$

where $\hat{\mathbf{n}}_i$ represents the standard cartesian unit vector.

1.2.3 Stability Terms

The RBVMS method of stabilization defines a subgrid scale definition of the trial and test functions that attempts to capture modes of the final solution that are not resolved. We denote these terms with a superscript $'$, and define them as follows:

$$\mathbf{u} = \mathbf{u}^h + \mathbf{u}', \tag{1.28}$$

$$p = p^h + p', \tag{1.29}$$

$$\mathbf{w} = \mathbf{w}^h + \mathbf{w}', \tag{1.30}$$

$$q = q^h + q'. \tag{1.31}$$

By ignoring the fine scale terms in the test functions (i.e. $\mathbf{w} = \mathbf{w}^h$, $q = q^h$), we can use Eqs. 1.28 and 1.29 in the weak form (Eq. 1.19) [15]. Then, using integration by parts, the derivatives can be moved onto the test functions, and we arrive at the following form:

$$\mathbf{u}' = -\frac{\tau_{SUPS}}{\rho}\mathbf{r}^{mom}(\mathbf{u}^h, p^h), \tag{1.32}$$

$$p' = -\rho\nu_{LSIC}r^{cont}(p^h), \tag{1.33}$$

where $\mathbf{r}^{mom}$ and r^{cont} are the residuals of the unstabilized continuity and momentum equations, defined pointwise in the form:

$$\mathbf{r}^{mom}(\mathbf{u}^h, p^h) = \rho\left(\frac{\partial \mathbf{u}}{\partial t}|_{\hat{\mathbf{x}}} + (\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla\mathbf{u} - \mathbf{f}\right) - \nabla \cdot \sigma(\mathbf{u}^h, p^h), \tag{1.34}$$

$$r^{cont}(\mathbf{u}^h) = \nabla \cdot \mathbf{u}^h. \tag{1.35}$$

The RHS described in Eqs. 1.26 and 1.27 can then be rewritten with additional terms. The full derivation is available in [15], and the final result is shown:

$$\mathbf{R}_A^{cont} = \int_{\Omega_t} N_A \nabla \cdot \mathbf{u}^h \mathrm{d}\Omega + \sum_{e=1}^{N_{el}} \int_{\Omega_e} \tau_{SUPS} \frac{\nabla N_A}{\rho} \cdot \mathbf{r}^{mom}(\mathbf{u}^h, p^h) \mathrm{d}\Omega \tag{1.36}$$

$$\begin{aligned}
\mathbf{R}_{A_i}^{mom} = &\int_{\Omega_t} N_A \hat{\mathbf{n}}_i \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t}|_{\hat{x}} + (\mathbf{u}^h - \hat{\mathbf{u}}^h) \cdot \nabla \mathbf{u}^h - \mathbf{f}^h \right) \mathrm{d}\Omega \\
&+ \int_{\Omega} \epsilon(N_A \hat{\mathbf{n}}_i){:}\sigma(\mathbf{u}^h, p^h) \mathrm{d}\Omega - \int_{\Gamma_{h_t}} N_A \hat{\mathbf{n}}_i \cdot \mathbf{h}^h \mathrm{d}\Gamma \\
&+ \sum_{e=1}^{N_{el}} \int_{\Omega_e} \tau_{SUPS} (\mathbf{u}^h \cdot \nabla N_A \hat{\mathbf{n}}_i) \cdot \mathbf{r}^{mom}(\mathbf{u}^h, p^h) \mathrm{d}\Omega \\
&+ \sum_{e=1}^{N_{el}} \int_{\Omega_e} \rho \nu_{LSIC} (\nabla \cdot N_A \hat{\mathbf{n}}_i) r^{cont}(\mathbf{u}^h) \mathrm{d}\Omega \\
&- \sum_{e=1}^{N_{el}} \int_{\Omega_e} \tau_{SUPS} N_A \hat{\mathbf{n}}_i \cdot (\mathbf{r}^{mom}(\mathbf{u}^h, p^h) \cdot \nabla \mathbf{u}^h) \mathrm{d}\Omega \\
&- \sum_{e=1}^{N_{el}} \int_{\Omega_e} \frac{\nabla N_A \hat{\mathbf{n}}_i}{\rho} : (\tau_{SUPS} \mathbf{r}^{mom}(\mathbf{u}^h, p^h)) \otimes (\tau_{SUPS} \mathbf{r}^{mom}(\mathbf{u}^h, p^h)) \mathrm{d}\Omega.
\end{aligned} \tag{1.37}$$

The parameters τ_{SUPS} and ν_{LSIC} are taken from literature involving stabilized finite element methods for fluids (see, e.g. [37, 85, 169]). There are several previously defined variations of these parameters, and we employ the following definition:

$$\tau_{SUPS} = \left(\frac{4}{\Delta t^2} + \mathbf{u}^h \cdot \mathbf{G} \mathbf{u}^h + C_I \nu^2 \mathbf{G} : \mathbf{G} \right)^{\frac{-1}{2}}, \tag{1.38}$$

$$\nu_{LSIC} = (\mathrm{tr}(\mathbf{G}) \tau_{SUPS})^{-1}. \tag{1.39}$$

The variable $\mathbf{G}$ is the element metric tensor, defined as

$$\mathbf{G} = \frac{\partial \xi}{\partial \mathbf{x}}^T \frac{\partial \xi}{\partial \mathbf{x}}, \tag{1.40}$$

where ξ represents the parametric coordinates of the element. The variable C_I is a scalar constant which is dependent on element topology and polynomial order, but remains independent of size. Details are available in [81, 94]. Lastly, the variable $\mathrm{tr}(\mathbf{G})$ is the trace of the element metric tensor $\mathbf{G}$.

1.2.4 Generalized-α

The evaluation of the Left-Hand-Side (LHS) matrix and RHS vector will depend on the specific choice of time integration scheme. For all FSI simulations in this dissertation, we employ a generalized-α method (often referred to as the θ method in the finite volume literature) to advance in time. For notational simplicity, we will from here on replace the term $\frac{\partial \mathbf{u}^h}{\partial t}|_{\hat{x}}$, as seen in Eq. 1.27, with $\dot{\mathbf{u}}$.

Time integration in general assumes a given set of values at nodal points at time t_n, where n is our time stepping index. In this case, these values are: $\dot{\mathbf{u}}_n$, $\mathbf{u}_n$, and p_n. These values are then used to determine a new set of values at some future point in time, t_{n+1}, expressed as: $\dot{\mathbf{u}}_{n+1}$, $\mathbf{u}_{n+1}$, and p_{n+1}. We note that we have 4 equations, and 7 unknowns in the current formulation. To reduce the number of unknowns, we employ an approximation known as the Newmark formula (see [81]), as shown:

$$\mathbf{u}_{n+1} = \mathbf{u}_n + \Delta t((1-\gamma)\dot{\mathbf{u}}_n + \gamma \dot{\mathbf{u}}_{n+1}). \tag{1.41}$$

We introduce several user-defined variables here, where Δt is the timestep size, and γ is a non-dimensional parameter, which affects the stability and accuracy of the solution. A choice of $\gamma = 1$, for example, is analogous to the implicit Euler method, while $\gamma = \frac{1}{2}$ yields the Crank-Nicholson method.

The generalized-α method defines two additional parameters, α_f and α_m, which also affect the stability and accuracy of the time integration. We define two intermediate points in time, $t_{n+\alpha_f}$ and $t_{n+\alpha_m}$, and define the following relations:

$$\mathbf{u}_{n+\alpha_f} = \mathbf{u}_n + \alpha_f(\mathbf{u}_{n+1} - \mathbf{u}_n), \tag{1.42}$$

$$\dot{\mathbf{u}}_{n+\alpha_m} = \dot{\mathbf{u}}_n + \alpha_m(\dot{\mathbf{u}}_{n+1} - \dot{\mathbf{u}}_n). \tag{1.43}$$

We now have three variables which affect our numerical algorithm, and the user may choose the values as appropriate. For example, choosing values of $\gamma = \alpha_f = \alpha_m = 1$ yields the implicit euler method, while the choice $\gamma = \alpha_f = \alpha_m = 0$ will yield the explicit euler method. It has been previously shown in [90] that using any choice such that $\gamma = 1/2 + \alpha_m - \alpha_f$ and $\alpha_m \geq \alpha_f \geq 1/2$ will yield second order accuracy in time, and be unconditionally stable.

We can now state the generalized-α method more formally: Given $\dot{\mathbf{u}}_n, \mathbf{u}_n, p_n$, find $\dot{\mathbf{u}}_{n+1}, \mathbf{u}_{n+1}, p_{n+1}$ such that

$$\mathbf{R}^{mom}\left(\dot{\mathbf{u}}_{n+\alpha_m}, \mathbf{u}_{n+\alpha_f}, p_{n+1}\right) = \mathbf{0}, \tag{1.44}$$

$$\mathbf{R}^{cont}\left(\dot{\mathbf{u}}_{n+\alpha_m}, \mathbf{u}_{n+\alpha_f}, p_{n+1}\right) = \mathbf{0}. \tag{1.45}$$

Now, we note that by Eqs. 1.41 and 1.42, we obtain:

$$\mathbf{u}_{n+\alpha_f} = \mathbf{u}_n + \alpha_f \Delta t(1-\gamma)\dot{\mathbf{u}}_n + \alpha_f \gamma \Delta t \dot{\mathbf{u}}_{n+1}. \tag{1.46}$$

We can then rewrite Eqs. 1.26 and 1.27 in terms of $\dot{\mathbf{u}}_{n+\alpha_m}$ and $\mathbf{u}_{n+\alpha_f}$, as follows:

$$\mathbf{R}_A^{cont} = \int_{\Omega_{t+\alpha_f}} N_A \nabla \cdot \mathbf{u}^h_{n+\alpha_f} \mathrm{d}\Omega, \tag{1.47}$$

$$\begin{aligned}\mathbf{R}_{A_i}^{mom} = &\int_{\Omega_{t+\alpha_f}} N_A \hat{\mathbf{n}}_i \cdot \rho \left(\dot{\mathbf{u}}^h_{n+\alpha_m} + (\mathbf{u}^h_{n+\alpha_f} - \hat{\mathbf{u}}^h_{n+\alpha_f}) \cdot \nabla \mathbf{u}^h_{n+\alpha_f} - \mathbf{f}^h\right) \mathrm{d}\Omega \\ &+ \int_{\Omega_{t+\alpha_f}} \epsilon(N_A \hat{\mathbf{n}}_i){:}\sigma(\mathbf{u}^h_{n+\alpha_f}, p^h)\mathrm{d}\Omega - \int_{\Gamma_{h_{t+\alpha_f}}} N_A \hat{\mathbf{n}}_i \cdot \mathbf{h}^h \mathrm{d}\Gamma.\end{aligned} \tag{1.48}$$

This is the RHS formulation assumed throughout this dissertation whenever the generalized-α method is used. The stabilized terms from Eqs. 1.37 and 1.36 are amended in a similar fashion.

1.2.5 Left-Hand-Side matrices

With a computationally feasible RHS, we can solve the following system incrementally on velocity and pressure:

$$\frac{\partial \mathbf{R}^{mom}}{\partial \dot{\mathbf{u}}_{n+1}} \Delta \dot{\mathbf{u}}_{n+1,i} + \frac{\partial \mathbf{R}^{mom}}{\partial p_{n+1}} \Delta p_{n+1,i} = -\mathbf{R}^{mom}_{i-1} \tag{1.49}$$

$$\frac{\partial \mathbf{R}^{cont}}{\partial \dot{\mathbf{u}}_{n+1}} \Delta \dot{\mathbf{u}}_{n+1,i} + \frac{\partial \mathbf{R}^{cont}}{\partial p_{n+1}} \Delta p_{n+1,i} = -\mathbf{R}^{cont}_{i-1}. \tag{1.50}$$

Each of the partial derivatives in the system of equations represented by Eqs. 1.49 and 1.50 represents a matrix, or *block* in the linear system. The linear system can then be more succinctly represented by rewriting it in the following form:

$$\begin{pmatrix} \mathbf{K} & \mathbf{G} \\ \mathbf{D} & \mathbf{L} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{u}}_{n+1} \\ p_{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{R}^{mom} \\ \mathbf{R}^{cont} \end{pmatrix}, \tag{1.51}$$

where we introduce the following definitions for **K**, **G**, **D**, and **L**:

$$\mathbf{K} = \frac{\partial \mathbf{R}^{mom}}{\partial \dot{\mathbf{u}}_{n+1}}, \tag{1.52}$$

$$\mathbf{G} = \frac{\partial \mathbf{R}^{mom}}{\partial p_{n+1}}, \tag{1.53}$$

$$\mathbf{D} = \frac{\partial \mathbf{R}^{cont}}{\partial \dot{\mathbf{u}}_{n+1}}, \tag{1.54}$$

$$\mathbf{L} = \frac{\partial \mathbf{R}^{cont}}{\partial p_{n+1}}. \tag{1.55}$$

Within each Newton iteration, we hold convective velocity fixed in order to linearize the equations and assemble the appropriate matrices. We will denote this variable as $\bar{\mathbf{u}}$. These are updated at each Newton step, and thus the LHS matrices are reassembled at every Newton iteration. The matrices are also assembled at the relevant α step level. We can carry out the differentiation for these matrices, and the Galerkin form of the matrix **K** becomes

$$\begin{aligned}\mathbf{K} =& \frac{\partial}{\partial \dot{\mathbf{u}}_{n+1}} \left(\int_{\Omega_{t+\alpha_f}} N_A \hat{\mathbf{n}}_i \cdot \rho \left(\dot{\mathbf{u}}^h_{n+\alpha_m} + \bar{\mathbf{u}}^h_{n+\alpha_f} \cdot \nabla \mathbf{u}^h_{n+\alpha_f} - \mathbf{f}^h \right) \mathrm{d}\Omega \right) \\ &+ \frac{\partial}{\partial \dot{\mathbf{u}}_{n+1}} \left(\int_{\Omega_{t+\alpha_f}} \epsilon(w^h) : \sigma(\mathbf{u}^h_{n+\alpha_f}, p^h) \mathrm{d}\Omega \right) \\ &- \frac{\partial}{\partial \dot{\mathbf{u}}_{n+1}} \left(\int_{\Gamma_{h_{t+\alpha_f}}} N_A \hat{\mathbf{n}}_i \cdot \mathbf{h}^h \mathrm{d}\Gamma \right). \end{aligned} \tag{1.56}$$

Incorporating Eq. 1.46, we differentiate Eq. 1.56 term-by-term. The first term becomes:

$$\begin{aligned} &\frac{\partial}{\partial \dot{\mathbf{u}}_{n+1}} \left(\int_{\Omega_{t+\alpha_f}} N_A \hat{\mathbf{n}}_i \cdot \rho \left(\dot{\mathbf{u}}^h_{n+\alpha_m} + \bar{\mathbf{u}}^h_{n+\alpha_f} \cdot \nabla \mathbf{u}^h_{n+\alpha_f} - \mathbf{f}^h \right) \mathrm{d}\Omega \right) \\ &\quad = \int_{\Omega_{t+\alpha_f}} \rho \alpha_m N_A \hat{\mathbf{n}}_i \cdot N_B \hat{\mathbf{n}}_j \delta_{ij} \mathrm{d}\Omega \\ &\quad + \int_{\Omega_{t+\alpha_f}} \alpha_f \gamma \Delta t N_A \bar{\mathbf{u}}^h_{n+\alpha_f} \cdot (\nabla N_B) \hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_j \delta_{ij} \mathrm{d}\Omega. \end{aligned} \tag{1.57}$$

In the second term, we first expand the ϵ and σ terms as shown

$$
\begin{aligned}
&\frac{\partial}{\partial \dot{\mathbf{u}}_{n+1}} \left(\int_{\Omega_{t+\alpha_f}} \epsilon(N_A \hat{\mathbf{n}}_i){:}\sigma(\mathbf{u}^h_{n+\alpha_f}, p^h) \mathrm{d}\Omega \right) \\
&\qquad = \frac{\partial}{\partial \dot{\mathbf{u}}_{n+1}} \int_{\Omega_{t+\alpha_f}} \frac{1}{2}(\nabla w^h + \nabla w^{h^T}){:}\mu \alpha_f \gamma \Delta t(\nabla \dot{\mathbf{u}}^h_{n+1} + \nabla {\dot{\mathbf{u}}^h_{n+1}}^T)) \mathrm{d}\Omega \qquad (1.58) \\
&\qquad = \frac{\partial}{\partial \dot{\mathbf{u}}_{n+1}} \int_{\Omega_{t+\alpha_f}} \frac{\mu \alpha_f \gamma \Delta t}{2} (\nabla w^h{:}\nabla \dot{\mathbf{u}}^h_{n+1} + \nabla w^{h^T}{:}\nabla {\dot{\mathbf{u}}^h_{n+1}}^T \\
&\qquad + \nabla w^h{:}\nabla {\dot{\mathbf{u}}^h_{n+1}}^T + \nabla w^{h^T}{:}\nabla \dot{\mathbf{u}}^h_{n+1}) \mathrm{d}\Omega. \qquad (1.59)
\end{aligned}
$$

It should be noted here that $(\nabla w^{h^T}{:}\nabla {\dot{\mathbf{u}}^h_{n+1}}^T)^T = \nabla w^h{:}\nabla \dot{\mathbf{u}}^h_{n+1}$, and since the quantity is a scalar, we can combine terms in Eq. 1.59 to get:

$$
\begin{aligned}
&\frac{\partial}{\partial \dot{\mathbf{u}}_{n+1}} \left(\int_{\Omega_{t+\alpha_f}} \epsilon(N_A \hat{\mathbf{n}}_i){:}\sigma(\mathbf{u}^h_{n+\alpha_f}, p^h) \mathrm{d}\Omega \right) \\
&\qquad = \frac{\partial}{\partial \dot{\mathbf{u}}_{n+1}} \int_{\Omega_{t+\alpha_f}} \mu \alpha_f \gamma \Delta t \left(\nabla w^h{:}\nabla \dot{\mathbf{u}}^h_{n+1} + \nabla w^h{:}(\nabla {\dot{\mathbf{u}}^h_{n+1}}^T) \right) \mathrm{d}\Omega. \qquad (1.60)
\end{aligned}
$$

We can now perform the differentiation on each entry in Eq. 1.60. The first can be written as

$$
\begin{aligned}
&\frac{\partial}{\partial \dot{\mathbf{u}}_{n+1}} \nabla w^h{:}\nabla \dot{\mathbf{u}}^h_{n+1} \\
&\qquad = \frac{\partial}{\partial \dot{\mathbf{u}}_{B,j}} N_{A,j} \hat{\mathbf{n}}_i N_{B,k} \hat{\mathbf{n}}_l \dot{\mathbf{u}}_{B,l} \delta_{il} \delta_{jk} \qquad (1.61) \\
&\qquad = N_{A,j} \hat{\mathbf{n}}_i N_{B,j} \hat{\mathbf{n}}_i \frac{\partial}{\partial \dot{\mathbf{u}}_B, j} \dot{\mathbf{u}}_{B,i} \qquad (1.62) \\
&\qquad = \nabla N_A \cdot \nabla N_B \delta_{ij}. \qquad (1.63)
\end{aligned}
$$

Similarly, we can rewrite the transposed term from Eq. 1.60 as

$$
\begin{aligned}
&\frac{\partial}{\partial \dot{\mathbf{u}}_{n+1}} \nabla w^h{:}(\nabla {\dot{\mathbf{u}}^h_{n+1}}^T) \\
&\qquad = \frac{\partial}{\partial \dot{\mathbf{u}}_{n+1,j}} N_{A,j} \hat{\mathbf{n}}_i N_{B,k} \hat{\mathbf{n}}_l \dot{\mathbf{u}}_{B,l} \delta_{ik} \delta_{jl} \qquad (1.64) \\
&\qquad = N_{A,j} \hat{\mathbf{n}}_i N_{B,i} \hat{\mathbf{n}}_j \frac{\partial}{\partial \dot{\mathbf{u}}_{n+1,j}} \dot{\mathbf{u}}_{B,j} \qquad (1.65) \\
&\qquad = \nabla N_A \cdot \hat{\mathbf{n}}_j \nabla N_B \cdot \hat{\mathbf{n}}_i \qquad (1.66)
\end{aligned}
$$

We finally combine Eqs. 1.63 and 1.66 to rewrite the second term in Eq. 1.56 as

$$\int_{\Omega_{t+\alpha_f}} \mu\alpha_f\gamma\Delta t\left(\delta_{ij}\nabla N_A\cdot\nabla N_B + (\nabla N_A\cdot\hat{\mathbf{n}}_j)(\nabla N_B\cdot\hat{\mathbf{n}}_i)\right)\mathrm{d}\Omega. \tag{1.67}$$

The final term in Eq. 1.56 has no dependence on $\dot{\mathbf{u}}_{n+1}$, and drops out of the LHS. We finally combine Eqs.1.59 and 1.67, and get the final form of the Galerkin $\mathbf{K}$:

$$\begin{aligned}\mathbf{K}_{ij} = &\int_{\Omega_{t+\alpha_f}} \delta_{ij}\rho\alpha_m N_A N_B \mathrm{d}\Omega\\ &+\int_{\Omega_{t+\alpha_f}} \delta_{ij}\alpha_f\gamma\Delta t N_A\bar{\mathbf{u}}^h_{n+\alpha_f}\cdot(\nabla N_B)\,\mathrm{d}\Omega\\ &+\int_{\Omega_{t+\alpha_f}} \delta_{ij}\mu\alpha_f\gamma\Delta t(\nabla N_A\cdot\nabla N_B)\mathrm{d}\Omega\\ &+\int_{\Omega_{t+\alpha_f}} \mu\alpha_f\gamma\Delta t(\nabla N_A\cdot\hat{\mathbf{n}}_j)(\nabla N_B\cdot\hat{\mathbf{n}}_i)\mathrm{d}\Omega.\end{aligned} \tag{1.68}$$

The addition of stabilization terms is straightforward, and involves differentiating the extra RHS terms. This results in the following form for $\mathbf{K}$, which is exactly the Galerkin formulation with additional stability terms:

$$\begin{aligned}\mathbf{K}_{ij} = &\int_{\Omega_{t+\alpha_f}} \delta_{ij}\rho\alpha_m N_A N_B \mathrm{d}\Omega\\ &+\int_{\Omega_{t+\alpha_f}} \delta_{ij}\alpha_f\gamma\Delta t N_A\bar{\mathbf{u}}^h_{n+\alpha_f}\cdot(\nabla N_B)\,\mathrm{d}\Omega\\ &+\int_{\Omega_{t+\alpha_f}} \delta_{ij}\mu\alpha_f\gamma\Delta t(\nabla N_A\cdot\nabla N_B)\mathrm{d}\Omega\\ &+\int_{\Omega_{t+\alpha_f}} \mu\alpha_f\gamma\Delta t(\nabla N_A\cdot\hat{\mathbf{n}}_j)(\nabla N_B\cdot\hat{\mathbf{n}}_i)\mathrm{d}\Omega\\ &+\int_{\Omega_{t+\alpha_f}} \delta_{ij}\alpha_m\tau_{SUPS}\bar{\mathbf{u}}^h_{n+\alpha_f}\cdot\nabla N_B\mathrm{d}\Omega\\ &+\int_{\Omega_{t+\alpha_f}} \delta_{ij}\rho\alpha_f\gamma\Delta t\tau_{SUPS}(\bar{\mathbf{u}}^h_{n+\alpha_f}\cdot\nabla N_A)(\bar{\mathbf{u}}^h_{n+\alpha_f}\cdot\nabla N_B)\mathrm{d}\Omega\\ &+\int_{\Omega_{t+\alpha_f}} \rho\alpha_f\gamma\Delta t\nu_{LSIC}(\nabla N_A\cdot\hat{\mathbf{n}}_i)(\nabla N_B\cdot\hat{\mathbf{n}}_j)\mathrm{d}\Omega.\end{aligned} \tag{1.69}$$

The blocks $\mathbf{G}$, $\mathbf{D}$, and $\mathbf{L}$ are similarly computed, and the end result is

$$\begin{aligned}\mathbf{G} = &-\int_{\Omega_{t+\alpha_f}} \nabla N_A \cdot \hat{\mathbf{n}}_i N_B \mathrm{d}\Omega \\ &+ \int_{\Omega_{t+\alpha_f}} \tau_{SUPS}(\bar{\mathbf{u}}^h_{n+\alpha_f} \cdot \nabla N_A)(\nabla N_B \cdot \hat{\mathbf{n}}_i)\mathrm{d}\Omega, \qquad (1.70)\\ \mathbf{D} = &\int_{\Omega_{t+\alpha_f}} \alpha_f \gamma \Delta t N_A \nabla \cdot (N_B \hat{\mathbf{n}}_i)\mathrm{d}\Omega \\ &+ \int_{\Omega_{t+\alpha_f}} \alpha_f \gamma \Delta t \tau_{SUPS}(\nabla N_A \cdot \hat{\mathbf{n}}_j)(\bar{\mathbf{u}}^h_{n+\alpha_f} \cdot \nabla N_B)\mathrm{d}\Omega \\ &+ \int_{\Omega_{t+\alpha_f}} \alpha_m \tau_{SUPS} \nabla N_A \cdot \hat{\mathbf{n}}_j N_B \mathrm{d}\Omega, \qquad (1.71)\\ \mathbf{L} = &\int_{\Omega_{t+\alpha_f}} \frac{\tau_{SUPS}}{\rho} \nabla N_A \cdot \nabla N_B \mathrm{d}\Omega. \qquad (1.72)\end{aligned}$$

Interestingly, the $\mathbf{L}$-block of Eq. 1.51 is identically $\mathbf{0}$ in the Galerkin formulation, as p does not explicitly appear in the continuity formulation. However, the stability terms populate $\mathbf{L}$ as shown. Further details of these derivations may be found in [28].

1.3 Structural Mechanics

In this section, we briefly introduce the concepts and governing equations of structural mechanics, and derive an appropriate LHS matrix and RHS vector for computation. We continue to utilize the concepts of a *reference* and *current* domain, denoted Ω_0 and Ω_t, respectively. The variable $\mathbf{X}$ is then used to denote the set of physical coordinates in the reference domain. The following relation then follows:

$$\mathbf{x}(\mathbf{X}, t) = \mathbf{X} + \mathbf{d}(\mathbf{X}, t), \qquad (1.73)$$

where $\mathbf{d}$ is a time-varying vector field on Ω_0, which acts as a mapping such that $\mathbf{d} : \Omega_0 \Rightarrow \Omega_t$. A field of instantaneous structural deflections at a given time point, t, is equivalent to $\mathbf{d}$. The standard definitions employed for structural mechanics are used, and shown below:

$$\ddot{\mathbf{d}} = \frac{\mathrm{d}^2\mathbf{d}}{\mathrm{d}t^2}, \qquad \mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}}, \qquad \mathbf{C} = \mathbf{F}^T\mathbf{F},$$
$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I}), \qquad J = \mathrm{det}\mathbf{F}. \tag{1.74}$$

In these definitions, we use $\ddot{\mathbf{d}}$ to refer to the structural acceleration, and the derivative $\frac{\mathrm{d}^2}{\mathrm{d}t^2}$ is assumed to operate with regard to the reference domain. The deformation gradient is denoted by $\mathbf{F}$, the Cauchy-Green tensor by $\mathbf{C}$, the Green-Lagrange strain by $\mathbf{E}$, and we use J to denote the determinant of the deformation gradient.

The governing equations of structural mechanics are predicated on the principal of *virtual work* [29], which states:

$$\delta W = \delta W_{int} + \delta W_{ext} = 0, \tag{1.75}$$

where δW is the total work, and δW_{int}, δW_{ext} are the internal and and external work, respectively. The symbol δ denotes the variation with regard to a *virtual displacement*, which we denote as $\mathbf{w}$. Internal work is defined as the virtual work done by internal stresses, and external work is defined as the virtual work done by body forces and surface traction. The external work can then be written as:

$$\delta W_{ext} = \int_{\Omega_t} \mathbf{w} \cdot \rho(\mathbf{f} - \ddot{\mathbf{d}})\mathrm{d}\Omega + \int_{(\Gamma_t)_h} \mathbf{w} \cdot \mathbf{h}\mathrm{d}\Gamma, \tag{1.76}$$

where $\mathbf{f}$ is the body force, $\mathbf{h}$ is the surface traction vector, and ρ is the structural density in the current configuration. The internal virtual work, δW_{int}, is similarly shown as:

$$\delta W_{int} = -\int_{\Omega_0} \delta\mathbf{E} : \mathbf{S}\mathrm{d}\Omega, \tag{1.77}$$

where $\mathbf{S}$ is the second Piola-Kirchhoff stress tensor, which is written as:

$$\mathbf{S} = J\mathbf{F}^{-1} \cdot \sigma \cdot \mathbf{F}^{-T}, \tag{1.78}$$

where σ is the familiar Cauchy stress tensor. Combining Eqs. 1.75, 1.77, and 1.76 yields:

$$\int_{\Omega_t} \mathbf{w}\rho \cdot \mathbf{f}\mathrm{d}\Omega - \int_{\Omega_t} \mathbf{w}\rho \cdot \ddot{\mathbf{d}}\mathrm{d}\Omega + \int_{(\Gamma_t)_h} \mathbf{w} \cdot \mathbf{h}\mathrm{d}\Gamma - \int_{\Omega_0} \delta\mathbf{E} : \mathbf{S}\mathrm{d}\Omega = \mathbf{0}, \tag{1.79}$$

which is conveniently posed in the weak formulation if we define our trial and test function spaces accordingly as:

$$S_{\mathbf{d}} = \{\mathbf{d}|\mathbf{d}(\cdot,t) \in (H^1(\Omega_t))^{n_{sd}}, d_i = g_i \text{ on } (\Gamma_t)_{\mathrm{g}i}\}, \tag{1.80}$$

$$V_{\mathbf{d}} = \{\mathbf{w}|\mathbf{w}(\cdot) \in (H^1(\Omega_t))^{n_{sd}}, w_i = 0 \text{ on } (\Gamma_t)_{\mathrm{g}i}\}. \tag{1.81}$$

Eq. 1.79 relies on the unknown time-varying scalar, ρ. However, a brief analysis through conservation of mass shows that $\rho = \frac{\rho_0}{J}$. Since ρ_0 is the initial known density field, this allows the computation of ρ at any time t. This is proven below:

$$m_t = \int_{\Omega_t} \rho \mathrm{d}\Omega, \tag{1.82}$$

$$m_0 = \int_{\Omega_0} \rho_0 \mathrm{d}\Omega, \tag{1.83}$$

where m_t is the mass of the system at time t, and m_0 is the initial mass, which we presume is equal to m_t by conservation of mass. We can change the integrand in Eq. 1.82 to the reference domain

$$\begin{aligned} m_t &= \int_{\Omega_t} \rho \mathrm{d}\Omega \\ &= \int_{\Omega_0} \rho J \mathrm{d}\Omega \\ \therefore m_0 &= \int_{\Omega_0} \rho J \mathrm{d}\Omega \end{aligned} \tag{1.84}$$

and therefore deduce that $\rho_0 = \rho J$.

The other term in Eq. 1.79 that requires further analysis is $\delta\mathbf{E}$, which can be rewritten as:

$$\delta\mathbf{E} = \frac{1}{2}\left(\mathbf{F}^T \nabla_{\mathbf{X}} \mathbf{w} + \nabla_{\mathbf{X}} \mathbf{w}^T \mathbf{F}\right), \tag{1.85}$$

where the subscript $\mathbf{X}$ on the differential operator, ∇, implies that the derivatives are taken with respect to the reference configuration. The term $\delta\mathbf{E} : \mathbf{S}$ then reduces to:

$$\delta\mathbf{E} : \mathbf{S} = \nabla_{\mathbf{X}} : (\mathbf{FS}). \tag{1.86}$$

Another change of variables in the integral, this time to the current configuration, yields:

$$\int_{\Omega_0} \nabla_{\mathbf{X}} \mathbf{w} : (\mathbf{FS}) \mathrm{d}\Omega = \int_{\Omega_0} \frac{\partial w_i}{\partial X_J} \frac{\partial x_i}{\partial X_I} S_{IJ} \mathrm{d}\Omega \tag{1.87}$$

$$= \int_{\Omega_t} \frac{\partial w_i}{\partial x_j} \left(\frac{\partial x_i}{\partial X_I} S_{IJ} \frac{\partial x_j}{\partial X_J} J^{-1} \right) \mathrm{d}\Omega \tag{1.88}$$

$$= \int_{\Omega_t} \nabla \mathbf{w} : \left(J^{-1} \mathbf{F} \mathbf{S} \mathbf{F}^T \right) \mathrm{d}\Omega. \tag{1.89}$$

Using the definition of the second Piola-Kirchhoff stress tensor shown in Eq. 1.78, we can finally rewrite the above equation as:

$$\int_{\Omega_t} \nabla \mathbf{w} : \left(J^{-1} \mathbf{F} \mathbf{S} \mathbf{F}^T \right) \mathrm{d}\Omega = \int_{\Omega_t} \epsilon(\mathbf{w}) : \sigma \mathrm{d}\Omega, \tag{1.90}$$

where ϵ is the symmetric differential operator. Finally, we may rewrite Eq. 1.79

$$\int_{\Omega_t} \mathbf{w} \rho \cdot \mathbf{f} \mathrm{d}\Omega - \int_{\Omega_t} \mathbf{w} \rho \cdot \ddot{\mathbf{d}} \mathrm{d}\Omega + \int_{(\Gamma_t)_h} \mathbf{w} \cdot \mathbf{h} \mathrm{d}\Gamma - \int_{\Omega_t} \epsilon(\mathbf{w}) : \sigma \mathrm{d}\Omega = \mathbf{0}, \tag{1.91}$$

which is the weak form of the structural equations.

The Cauchy stress term is modeled as

$$\sigma = \mathbb{C} \epsilon(\mathbf{d}), \tag{1.92}$$

where $\mathbb{C}$ is a fourth-rank tensor of elastic coefficients. This tensor may be designed to represent different material model combinations, such as material anisotropy and hyperelastic materials. For now, we will assume an isotropic, linearly elastic material. Different material models will be presented as they are used. We briefly present the case of linear elasticity, as this is model used to model mesh motion in Sec. 1.4.2.

1.3.1 Linear Elasticity

The classic linearly elastic definition of $\mathbb{C}$ is

$$\mathbb{C}_{ijkl} = \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}) + \lambda \delta_{ij} \delta_{kl}, \tag{1.93}$$

where μ and λ are called the Lamé parameters. These parameters are related to the Young's modulus, E, and the Poisson ration, ν, of the structural material, given by

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \tag{1.94}$$

$$\mu = \frac{E}{2(1+\nu)}. \tag{1.95}$$

These relations, as well as others, are covered in [148]. We can then rewrite the stress term of Eq. 1.91 in a compact way as follows:

$$\int_{\Omega_t} \epsilon(\mathbf{w}) : \sigma \mathrm{d}\Omega = \int_{\Omega_t} \epsilon(\mathbf{w}) \cdot \mathbf{D}\epsilon(\mathbf{d}) \mathrm{d}\Omega, \tag{1.96}$$

where $\mathbf{D} = D_{IJ}$ is a second rank tensor representation of the fourth rank tensor $\mathbb{C}$. In three dimensions, $\mathbf{D}$ is

$$\mathbf{D} = \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix}. \tag{1.97}$$

The symmetric differential operator, $\epsilon(\mathbf{j})$, is then also reduced to a vector from a second rank tensor as shown for an arbitrary vector input, $\mathbf{j}$. In three dimensions, this is given as

$$\epsilon(\mathbf{j}) = \begin{pmatrix} j_{1,1} \\ j_{2,2} \\ j_{3,3} \\ j_{2,3} + j_{3,2} \\ j_{3,1} + j_{1,3} \\ j_{1,2} + j_{2,1} \end{pmatrix}. \tag{1.98}$$

Eq. 1.91 is rewritten with these modifications, and put into the discrete form:

$$\int_{\Omega_t} \mathbf{w}^h \rho \cdot \ddot{\mathbf{d}}^h \mathrm{d}\Omega + \int_{\Omega_t} \epsilon(\mathbf{w}^h) \cdot \mathbf{D}\epsilon(\mathbf{d}^h) \mathrm{d}\Omega - \int_{\Omega_t} \mathbf{w}^h \rho \cdot \mathbf{f}^h \mathrm{d}\Omega - \int_{(\Gamma_t)_h} \mathbf{w}^h \cdot \mathbf{h}^h \mathrm{d}\Gamma = \mathbf{0}. \tag{1.99}$$

1.3.2 Generalized-α

Employing the generalized-α time integration scheme on this discretization is similar to the approach taken for fluid dynamics. The nodal solutions at the intermediate time levels are defined as:

$$\begin{aligned}\mathbf{d}_{n+\alpha_f} &= \mathbf{d}_n + \alpha_f(\mathbf{d}_{n+1} - \mathbf{d}_n) && (1.100)\\ \ddot{\mathbf{d}}_{n+\alpha_m} &= \ddot{\mathbf{d}}_n + \alpha_m(\ddot{\mathbf{d}}_{n+1} - \ddot{\mathbf{d}}_n), && (1.101)\end{aligned}$$

and we can again employ the Newmark formula to establish a relationship between the time derivatives of the unknown vectors. This relation is:

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \Delta t\dot{\mathbf{d}}_n + \frac{\Delta t^2}{2}\left((1-2\beta)\ddot{\mathbf{d}}_n + 2\beta\ddot{\mathbf{d}}_{n+1}\right), \tag{1.102}$$

where β is a user provided value that affects the stability and accuracy of the solution. Chung and Hulbert [42] showed that if β is chosen such that

$$\beta = \frac{1}{4}(1 + \alpha_m - \alpha_f)^2, \tag{1.103}$$

then the results will remain second-order accurate in time. Unconditional stability is also ensured, provided $\alpha_m \geq \alpha_f \geq \frac{1}{2}$, as required previously.

1.3.3 Structural RHS Vector

The structural residual vector is analogous to the formulation of the fluid residual vectors, in that the discrete equations will invariably not identically equal $\mathbf{0}$, and the actual value of the RHS vector is referred to as the residual and is denoted by $\mathbf{R}^{str}$. Using the generalized-α method, the RHS of the structure in Eq. 1.99 takes the form:

$$\begin{aligned}\mathbf{R}^{str}_{A_i} = &\int_{\Omega_t} N_A\hat{\mathbf{n}}_i\rho\cdot\ddot{\mathbf{d}}^h_{n+\alpha_m}\mathrm{d}\Omega + \int_{\Omega_t}\epsilon(N_A\hat{\mathbf{n}}_i)\cdot\mathbf{D}\epsilon(\mathbf{d}^h_{\alpha_f})\mathrm{d}\Omega\\ &- \int_{\Omega_t} N_A\hat{\mathbf{n}}_i\rho\cdot\mathbf{f}^h\mathrm{d}\Omega - \int_{(\Gamma_t)_h} N_A\hat{\mathbf{n}}_i\cdot\mathbf{h}^h\mathrm{d}\Gamma. \end{aligned}\tag{1.104}$$

1.3.4 Structural LHS Matrices

The matrix system for structural mechanics is often written as

$$\mathbf{M}\ddot{\mathbf{d}}_{n+1} + \mathbf{K}\mathbf{d}_{n+1} - \mathbf{F} = \mathbf{0}, \tag{1.105}$$

where $\mathbf{M}$ is called the *mass matrix*, and $\mathbf{K}$ is called the *stiffness matrix*. However, the Newmark formula employed in Eq. 1.102 allows us to solve in terms of only one unknown, $\ddot{\mathbf{d}}_{n+1}$, and we can iteratively solve the following system:

$$\frac{\partial \mathbf{R}^{str}}{\partial \ddot{\mathbf{d}}_{n+1}} \Delta \ddot{\mathbf{d}}_{n+1,i} = -\mathbf{R}^{str}_{i-1}, \tag{1.106}$$

where i is an iterative index. We represent the matrix system $\frac{\partial \mathbf{R}^{str}}{\partial \ddot{\mathbf{d}}_{n+1}}$ as $\mathbf{K}$, and we can rewrite:

$$\mathbf{K}\Delta \ddot{\mathbf{d}}_{n+1,i} = -\mathbf{R}^{str}_{i-1}. \tag{1.107}$$

The construction of the matrix $\mathbf{K}$ is dependent on differentiating the residual vector with respect to the unknown variable, and we perform this derivation on Eq. 1.104 term-by-term, using Eqs. 1.100, 1.101, and 1.102 to replace terms as necessary. The first term reduces to:

$$\begin{aligned} \frac{\partial}{\partial \ddot{\mathbf{d}}_{n+1}} \int_{\Omega_t} N_A \hat{\mathbf{n}}_i \rho \cdot \ddot{\mathbf{d}}^h_{n+\alpha_m} \mathrm{d}\Omega &= \int_{\Omega_t} N_A \rho \alpha_m N_B \hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_j \mathrm{d}\Omega \\ &= \int_{\Omega_t} \alpha_m \rho N_A N_B \delta_{ij} \mathrm{d}\Omega. \end{aligned} \tag{1.108}$$

The second term in Eq. 1.104 includes two terms of the form $\epsilon(N_A \hat{\mathbf{n}}_i)$. We represent this vector in a compact form by using a matrix-vector product, denoted $\mathbf{B}_A \hat{\mathbf{n}}_i$. The matrix $\mathbf{B}_A$ can be computed as:

$$\mathbf{B}_A = \begin{bmatrix} N_{A,1}\ 0\ 0 & & \\ 0 & N_{A,2} & 0 \\ 0 & 0 & N_{A,3} \\ 0 & N_{A,3} & N_{A,2} \\ N_{A,3} & 0 & N_{A,1} \\ N_{A,2} & N_{A,1} & 0 \end{bmatrix}. \tag{1.109}$$

We can then differentiate and rewrite the second term as follows:

$$\frac{\partial}{\partial \ddot{\mathbf{d}}_{n+1}} \int_{\Omega_t} \epsilon(N_A \hat{\mathbf{n}}_i) \cdot \mathbf{D}\epsilon(\mathbf{d}^h_{\alpha_f}) \mathrm{d}\Omega = \int_{\Omega_t} \alpha_f \beta \Delta t^2 \hat{\mathbf{n}}_i \cdot \mathbf{B}_A^T \mathbf{D} \mathbf{B}_B \hat{\mathbf{n}}_j \mathrm{d}\Omega. \tag{1.110}$$

The remaining terms have no dependence on $\ddot{\mathbf{d}}_{n+1}$, and are thus neglected. We can thus write the LHS of the linear structural equations as

$$\mathbf{K}_{ij} = \int_{\Omega_t} \alpha_m \rho N_A N_B \delta_{ij} \mathrm{d}\Omega + \int_{\Omega_t} \alpha_f \beta \Delta t^2 \hat{\mathbf{n}}_i \cdot \mathbf{B}_A^T \mathbf{D} \mathbf{B}_B \hat{\mathbf{n}}_j \mathrm{d}\Omega. \tag{1.111}$$

1.4 Strong Coupling Formulation

The strongly coupled FSI system is posed in the same way as shown in Sections 1.2 and 1.3. Namely, given residual vectors, $\mathbf{R}^{mom}_{A_i}$, $\mathbf{R}^{con}_{A}$, and $\mathbf{R}^{str}_{A_i}$, we can solve the following system simultaneously:

$$\begin{aligned}
\frac{\partial \mathbf{R}^{mom}}{\partial \dot{\mathbf{u}}_{n+1}} \Delta \dot{\mathbf{u}}_{n+1,i} &+ \frac{\partial \mathbf{R}^{mom}}{\partial p_{n+1}} \Delta p_{n+1,i} + \frac{\partial \mathbf{R}^{mom}}{\partial \ddot{\mathbf{d}}_{n+1}} \Delta \ddot{\mathbf{d}}_{n+1,i} &= -\mathbf{R}^{mom}_{i-1} \\
\frac{\partial \mathbf{R}^{cont}}{\partial \dot{\mathbf{u}}_{n+1}} \Delta \dot{\mathbf{u}}_{n+1,i} &+ \frac{\partial \mathbf{R}^{cont}}{\partial p_{n+1}} \Delta p_{n+1,i} + \frac{\partial \mathbf{R}^{cont}}{\partial \ddot{\mathbf{d}}_{n+1}} \Delta \ddot{\mathbf{d}}_{n+1,i} &= -\mathbf{R}^{cont}_{i-1} \\
\frac{\partial \mathbf{R}^{str}}{\partial \dot{\mathbf{u}}_{n+1}} \Delta \dot{\mathbf{u}}_{n+1,i} &+ \frac{\partial \mathbf{R}^{str}}{\partial p_{n+1}} \Delta p_{n+1,i} + \frac{\partial \mathbf{R}^{str}}{\partial \ddot{\mathbf{d}}_{n+1}} \Delta \ddot{\mathbf{d}}_{n+1,i} &= -\mathbf{R}^{str}_{i-1}.
\end{aligned} \tag{1.112}$$

Written in matrix form, this system can be expressed as

$$\begin{pmatrix} \mathbf{K}_{11} & \mathbf{G}_1 & \mathbf{K}_{12} \\ \mathbf{D}_1 & \mathbf{L} & \mathbf{D}_2 \\ \mathbf{K}_{21} & \mathbf{G}_2 & \mathbf{K}_{22} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{u}}_{n+1} \\ p_{n+1} \\ \ddot{\mathbf{d}}_{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{R}^{mom} \\ \mathbf{R}^{cont} \\ \mathbf{R}^{str} \end{pmatrix}, \tag{1.113}$$

where the upper 2×2 block of matrices containing $\mathbf{K}_{11}$, $\mathbf{G}_1$, $\mathbf{D}_1$, and $\mathbf{L}$ correspond exactly to the matrices derived in Section 1.2, and can be explicitly computed using Eqs. 1.69– 1.72 without further modification. Likewise, the matrix denoted $\mathbf{K}_{22}$ corresponds with the structural LHS derivation found in Section 1.3, and can be computed using Eq. 1.111. In this formulation, there are four new matrices which have yet to be derived, namely $\mathbf{K}_{12}$, $\mathbf{K}_{21}$, $\mathbf{D}_2$, and $\mathbf{G}_2$. For some problems, these new blocks may be assumed to be empty. This is a *loosely* coupled or *block iterative* approach. However, this approach fails for more complex FSI problems. In this case, we must populate the off-diagonal matrices and solve the fully coupled system.

1.4.1 Fluid–Structure Interface

To derive the off-diagonal terms, we must first define the surface of the FSI interface, as shown:

$$\Gamma_I = \Gamma_0 \bigcap \Gamma_0^s, \tag{1.114}$$

where Γ_0 is the reference fluid surface, and Γ_0^s is the reference structural surface. The non-interfacial boundaries are then denoted as Γ_E, and Γ_E^s, for the fluid and structure respectively. We further require that $\Omega_t \bigcap \Omega_t^s = \emptyset$, where Ω_t and Ω_t^s are the time dependent domains for the fluid and structure, respectively. This requirement guarantees that the fluid and structure do not occupy the same volume, which would violate the physics of the problem. We enforce two distinct boundary conditions on Γ_I to enforce this requirement:

$$\mathbf{u} = \dot{\mathbf{d}} \quad \text{on} \quad \Gamma_I, \tag{1.115}$$

$$\mathbf{w} = \mathbf{w}^s \quad \text{on} \quad \Gamma_I, \tag{1.116}$$

where $\mathbf{w}$ and $\mathbf{w}^s$ are the test functions of the fluid and structural domain, respectively. Eq. 1.115 is a Dirichlet boundary condition on the fluid component, which ensures that the fluid velocity is equal to the structural velocity at the interface. The introduction of the structural term $\dot{\mathbf{d}}$ on the fluid boundary allows for computation of the blocks $\mathbf{K}_{12}$ and $\mathbf{D}_2$. Eq. 1.116 ensures that the test functions on the interface are the same for both the fluid and structure. We can then examine the resulting coupled set of equations in the weak form. We find $\ddot{\mathbf{d}} \in S_d$, $\mathbf{u} \in S_u$

and $p \in S_p$ such that $\forall \mathbf{w}^s \in V_d$, $\mathbf{w} \in V_u$,, and $q \in V_p$:

$$\begin{aligned}\int_{\Omega_t} \mathbf{w} \cdot (\rho(\dot{\mathbf{u}} + \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{f}) - \nabla \cdot \sigma) \mathrm{d}\Omega \\ + \int_{\Gamma_E} \mathbf{w} \cdot (\sigma \hat{\mathbf{n}} - \mathbf{h}_E) \mathrm{d}\Gamma \\ + \int_{\Omega_t} q \nabla \cdot \mathbf{u} \mathrm{d}\Omega \\ + \int_{\Omega_t^s} \mathbf{w}^s \cdot (\rho^s(\ddot{\mathbf{d}}^s - \mathbf{f}^s) - \nabla \cdot \sigma^s) \mathrm{d}\Omega \\ + \int_{\Gamma_E^s} \mathbf{w}^s \cdot (\sigma^s \hat{\mathbf{n}}^s - \mathbf{h}_E^s) \mathrm{d}\Gamma \\ + \int_{\Gamma_I} (\mathbf{w} \cdot \sigma \hat{\mathbf{n}} + \mathbf{w}^s \cdot \sigma^s \hat{\mathbf{n}}^s) \mathrm{d}\Gamma = 0,\end{aligned} \tag{1.117}$$

where the superscript s denotes structural domain properties. Further details are available in [81, 28]. Equation 1.117 can be verified to reach the expected results on each respective subdomain. That is,

$$\rho(\dot{\mathbf{u}} + \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{f}) - \nabla \cdot \sigma = 0 \text{ on } \Omega_t, \tag{1.118}$$

$$\nabla \cdot \mathbf{u} = 0 \text{ on } \Omega_t, \tag{1.119}$$

$$\sigma \hat{\mathbf{n}} - \mathbf{h}_E = 0 \text{ on } \Gamma_E, \tag{1.120}$$

$$\rho^s(\ddot{\mathbf{d}}^s - \mathbf{f}^s) - \nabla \cdot \sigma^s = 0 \text{ on } \Omega_t^s \sigma^s \hat{\mathbf{n}}^s - \mathbf{h}_E^s = 0 \text{ on } \Gamma_E^s. \tag{1.121}$$

The final term on the boundary Γ_I can thus be written:

$$\int_{\Gamma_I} (\mathbf{w} \cdot \sigma \hat{\mathbf{n}} + \mathbf{w}^s \cdot \sigma^s \hat{\mathbf{n}}^s) \mathrm{d}\Gamma = 0. \tag{1.122}$$

Using the boundary condition in Eq. 1.116, we can replace terms and then verify that:

$$\sigma \hat{\mathbf{n}} + \sigma^s \hat{\mathbf{n}}^s = \mathbf{0} \text{ on } \Gamma_I, \tag{1.123}$$

which ensures the contuity of the traction vector across the fluid–structure interface. This also introduces the fluid terms $\mathbf{u}$ and p into the structural equation through σ, and thus the remaining block terms $\mathbf{K}_{21}$ and $\mathbf{G}_2$ may be computed.

1.4.2 Mesh Motion

We have yet to discuss mesh motion as a part of this coupled system. The mesh motion solution (e.g., $\hat{\mathbf{u}}$) is a critical component of the FSI system, as it contributes terms explicitly to the LHS of the fluid matrix $\mathbf{K}_{11}$ at each iteration. We further require that the mesh motion solution track Γ_I appropriately. This system is solved using linear elastostatics. The equations of linears elastostatics are exactly those derived for linear elasticity, but we enforce in Eq 1.105 that $\mathbf{M} = 0$. This can be accomplished using the same framework derived in Section 1.3 directly if we set $\rho = 0$. To enforce the consistency of Γ_I, we enforce the following boundary condition on mesh motion:

$$\dot{\hat{\mathbf{u}}} = \ddot{d} \;\text{ on }\; \Gamma_I. \tag{1.124}$$

Note that we also ensure that $\mathbf{u} = \hat{\mathbf{u}}$ on Γ_I by Eq. 1.115. Thus, the fluid velocity, structural velocity, and the interfacial velocity are all identical on Γ_I, as desired.

It is possible to couple the equations of mesh motion together with the fluid–structure system, and solve all systems simultaneously. This approach is called *direct coupling*, and is fully explored in the literature [171, 172, 173, 170, 18, 68, 57]. The approach used throughout this dissertation is to solve the mesh motion *block iteratively*, which is also called a *quasi-direct* method [171, 172, 173]. In the quasi-direct case, the mesh motion solution is 'lagged' by one iteration, and updated after each FSI iteration. The resulting system can be written as:

$$\begin{pmatrix} \mathbf{K}_{11} & \mathbf{G}_1 & \mathbf{K}_{12} & \mathbf{0} \\ \mathbf{D}_1 & \mathbf{L} & \mathbf{D}_2 & \mathbf{0} \\ \mathbf{K}_{21} & \mathbf{G}_2 & \mathbf{K}_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{K}_{33} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{u}}_{n+1} \\ p_{n+1} \\ \ddot{\mathbf{d}}_{n+1} \\ \dot{\hat{\mathbf{u}}}_{n+1} \end{pmatrix} = \begin{pmatrix} \mathbf{R}^{mom} \\ \mathbf{R}^{cont} \\ \mathbf{R}^{str} \\ \mathbf{R}^{mesh} \end{pmatrix}, \tag{1.125}$$

where $\mathbf{R}^{mesh}$ and $\mathbf{K}_{33}$ are the mesh motion RHS and LHS described by linear elastostatics, as derived in Section 1.3. In the quasi-direct case, the upper 3×3 block is solved first, and then the updated $n + 1$ solution of the structural displacement is then used to solve $\mathbf{K}_{33}\dot{\hat{\mathbf{u}}}_{n+1} = \mathbf{R}^{mesh}$.

Chapter 2

Fluid Structure Interaction Simulations of Fontan Surgical Connections

As appears in ***International Journal for Numerical Methods in Biomedical Engineering****, DOI: 10.1002/cnm.1485*

Children born with single ventricle heart defects typically undergo a staged surgical procedure culminating in a total cavopulmonary connection (TCPC), or Fontan Surgery. The goal of this work is to perform physiologic, patient-specific hemodynamic simulations of two post-operative TCPC patients using fluid-structure interaction (FSI) simulations. Data from two patients are presented and post-op anatomy is reconstructed from MRI data. Respiration rate, heart rate, and venous pressures are obtained from catheterization data, and inflow rates are obtained from phase contrast MRI data and are used together with a respiratory model. Lumped parameter (RCR) boundary conditions are used at the outlets. We perform FSI simulations using an Arbitrary Lagrangian Eulerian (ALE) finite element framework to account for motion of the blood vessel walls in the TCPC. This study is the first to introduce variable elastic properties for the different areas of the TCPC, including a Gore-Tex conduit. Quantities such as wall shear

stresses and pressures at critical locations are extracted from the simulation and are compared with pressure tracings from clinical data as well as with rigid wall simulations. Hepatic flow distribution and energy efficieny are also calculated and compared for all cases. There is little effect of FSI on pressure tracings, hepatic flow distribution, and time-averaged energy efficiency. However, the effect of FSI on wall shear stress, instantaneous energy efficiency, and wall motion is significant and should be considered in future work, particularly for accurate prediction of thrombus formation.

2.1 Introduction

Congenital heart defects affect nearly 1% of live births, and are one of the most common forms of birth defect. Single ventricle heart defects are a serious subset of congenital heart defects in which a child is born with only a single working ventricle. These cases are uniformly fatal without immediate medical intervention. Patients typically undergo a series of three staged surgical procedures, Norwood, Glenn, and finally Fontan or Total Cavopulmonary Connection (TCPC). This results in an unusual anatomy in which the the vena cavae are disconnected from the heart, and are anastomosed directly to the pulmonary arteries.

In this paper, we perform patient specific modeling on two extra-cardiac Fontan patients. In this procedure, a tube-shaped Gore-Tex conduit is inserted and connected to the pulmonary arteries. While the Fontan procedure has a high initial success rate of about 90%, long-term morbidity and quality of life issues are prevalent, including excercise intolerance, shortness of breath, arteriovenous malformations, and heart failure [95, 110].

While there have been many recent advances in Fontan simulation methods, including multiscale modeling, evaluation of multiple parameters, and optimization [56, 97, 130, 118], there have been relatively few studies incorporating fluid-structure interaction (FSI). The need for FSI evaluation was recently discussed by DeGroff [55], who argues for inclusion of compliant vessel structures, including accurately modeling surgical materials such as a Gore-Tex conduit, and discontinuous elastic properties around anastomosis sites.

FSI was previously introduced to this problem by Masters, et al.[117], and Orlando et. al. [127], who used a significantly simplified geometry of the total cavopulmonary connection. In Bazilevs et al. [19], FSI procedures were employed for a geometrically-complex patient-specific case, which included large portions of the pulmonary circulation. The results of these works demonstrated that vessel wall motion plays an important role in determining wall shear stress, particularly under excercise conditions. Variable wall properties in arterial blood flow simulations have been previously studied [179], but have not been introduced to Fontan simulations. Previous studies were also primarily limited to wall shear stress and energy loss data. In this study, we address previous limitations by including variable elastic and thickness properties and quantifying additional pressure and hepatic flow distribution parameters.

Physiologic boundary conditions at both the inlet and outlet have also been studied, including Windkessel and lumped parameter networks [175, 131]. Energy loss has been examined extensively in previous patient specific modeling studies, and this led to the adoption of an offset geometry to reduce these losses [177, 51, 112]. Optimization techniques to improve several presumed clinically relevant parameters, such as energy efficiency, flow distribution, and wall shear stress have also been introduced [115, 183, 36].

It is well-known that IVC flow (and Fontan flow in general) is strongly affected by respiration. Blood flow through the IVC increases dramatically during inhalation, and drops off precipitously during exhalation, while flow in the SVC is largely unaffected [72, 75]. This has been accounted for in previous Fontan simulations, which were run over multiple respiratory cycles using a polynomial to model the respiratory component [113]. While mean pressure values from catheterization have been used to define appropriate boundary conditions in previous studies, details of pressure tracings have often been ignored. In this work, we make full use of catheterization data to implement patient-specific respiration adjusted inflow profiles at the IVC inlet. This also allows us to directly incorporate the catheterization data into our simulation, thus improving their physiologic realism.

This work expands our previous FSI capabilities by introducing variable elastic properties, and releasing structural boundary conditions at the outlets to allow for more realistic motion. A systematic method for performing this type of simulation is presented, and patient specific catheterization data is utilized. Data from two patients is simulated. Comparisons to a rigid wall simulation and a case with uniform material properties are made. Hepatic flow split calculations for rigid and deformable simulations are compared. Since all patient data is acquired at rest, all simulations are restricted to the rest case in the present study. However, as shown in [19], FSI has a significantly more pronounced effect on the simulation results in the cases when exercise conditions are simulated.

2.2 Methodology

2.2.1 Clinical Data

Catheterization and MRI data were gathered for each patient as part of their routine clinical care with Instituional Review Board (IRB) approval. Each patient was imaged in a 1.5 Tesla MRI scanner (Signa Twin Speed, General Electrics, Milwaukee, WI). Gadolinium-based contrast agents were used, and images were acquired using 3-D fast gradient-recalled echo sequences with the patient in breath-hold. Flow information from the SVC and IVC was gathered using a 2-D phase contrast MRI (PC-MRI) method. In each case, the imaging plane was set to be perpendicular to the vascular centerline in the IVC/SVC, and velocity encoding was placed parallel to the flow. Data was acquired using cardiac gating over several respiratory cycles. Pressure tracings were acquired in the IVC, SVC, and left and right pulmonary arteries for each patient. More details about this data acquisition are in [112]. Patient 1 is a 3 year old male with a BSA of 0.68 m^2, and a mean Fontan pressure of 11 mmHg. Patient 2 is a 6 year old female with a BSA of 0.71 m^2, and a mean Fontan pressure of 9.5 mmHg.

2.2.2 Model Construction

Models of the patient specific vasculature are created using a custom version of the Simvascular software package [143] in the following four steps:

1. Individual blood vessels are identified, and “pathlines” are formed along the vessel centerlines. This creates a tree model of the vasculature.

2. Along each pathline, a series of segmentations are created perpendicular to the pathline to determine the location of the vessel wall lumen.

3. The segmentations are lofted together to create a 3-D solid model using B-Splines.

4. The solid model is meshed into tetrahedral elements using the commercial package MeshSim (Symmetrix, Inc., Troy, NY). A boundary layer mesh is created near the wall for improved resolution.

5. External mesh faces in different material domains are tagged appropriately, and a new file is generated containing this data. This allows for the application of a discontinuous Young’s modulus, Poisson ratio, and density at material boundaries.

Figure 2.1 shows models for the two patients, with the distinct material domains, and the wall thickness used in the FSI simulations. The determination of wall thickness will be discussed in a following section.

2.2.3 Numerical Method

The Arbitrary Lagrangian-Eulerian (ALE) formulation of the Incompressible Navier-Stokes Equations are solved in the fluid domain [104] using a Residual Based Variational Multi-Scale (RBVMS) Finite Element Method[15]. We will couple this with the equations of solid mechanics, and for that, we introduce superscripts to denote the solid or fluid domain. We can write the ALE Incompressible Navier-Stokes Equations in the appropriate weak form as: Find $\{\mathbf{u}, p\} \in V^f$ such

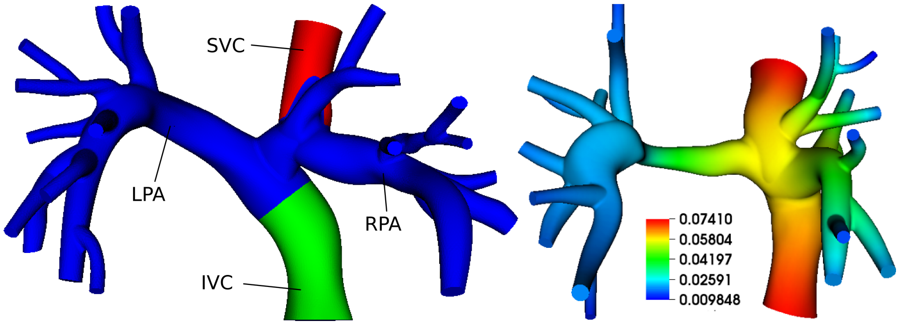


Figure 2.1: Material zones plot for patient 1 (left), including IVC (green), pulmonary arteries (blue), and SVC (red). Wall thickness (cm) plot for patient 2 (right).

that $\forall\{\mathbf{w}^f, q^f\} \in W^f$,

$$\begin{aligned}\int_{\Omega_t^f} \mathbf{w}^f \cdot \rho^f \frac{\partial \mathbf{u}}{\partial t} d\Omega_t^f + \int_{\Omega_t^f} \mathbf{w}^f \cdot \left(\rho^f (\mathbf{u} - \hat{\mathbf{u}}) \cdot \nabla_x \mathbf{u}\right) d\Omega_t^f - \int_{\Omega_t^f} \nabla_x \cdot \mathbf{w}^f p d\Omega_t^f \\ + \int_{\Omega_t^f} \nabla_x^S \mathbf{w}^f \cdot 2\mu^f \nabla_x^S \mathbf{u} d\Omega_t^f + \int_{\Omega_t^f} q^f \nabla_x \cdot \mathbf{u} d\Omega_t^f \\ - \int_{\Omega_t^f} \mathbf{w}^f \cdot \rho^f \mathbf{f}^f d\Omega_t^f - \int_{(\Gamma_t^f)_h} \mathbf{w}^f \cdot \mathbf{h}^f d(\Gamma_t^f)_h = 0 \end{aligned} \quad (2.1)$$

where the superscript f denotes the fluid domain, the subscript x on the differential operator indicates the derivatives are with respect to the current configuration, the symmetric differential operator is defined by $\nabla^S \mathbf{u} = \frac{1}{2}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$, V is the trial solution space, W is the trial weighting space, Ω_t denotes the current configuration of the fluid domain, $(\Gamma_t)_h$ denotes the Neumann boundary, $\mathbf{w}$ is the weighting function for the momentum equation, q is the weighting function for the continuity equation, the derivative $\frac{\partial \mathbf{u}}{\partial t}$ is taken with respect to time in the reference domain, and $\mathbf{h}$ is the boundary traction vector. We also assume $\mathbf{w} = 0$ on the Dirichlet boundary, $(\Gamma_t^f)_g$, and $\mathbf{u} = \mathbf{u}_g$ on the same boundary.

The structural domain of the models is solved simultaneously with the fluid domain, using the following weak formulation: Find $\mathbf{d} \in V^s$ such that $\forall w^s \in W^s$,

$$\int_{\Omega_0^s} \mathbf{w}^s \cdot \rho_0^s \frac{\partial^2 \mathbf{d}}{\partial t^2} d\Omega_0^s + \int_{\Omega_0^s} \nabla_X \mathbf{w}^s : (\boldsymbol{F}\boldsymbol{S}) d\Omega_0^s - \int_{\Omega_0^s} \mathbf{w}^s \cdot \rho_0^s \mathbf{f}^s d\Omega_0^s - \int_{(\Gamma_0^s)_h} \mathbf{w}^s \cdot \mathbf{h}^s d(\Gamma_0^s)_h = 0 \quad (2.2)$$

where the super script s denotes the solid domain, V is the trial solution space,

W is the trial weighting space, $(\Gamma_0)_h$ is the Neumann boundary, $\mathbf{d}$ is displacement, $\mathbf{w}$ is the weighting function, $\boldsymbol{F}$ is the deformation gradient, $\mathbf{f}$ is a body force, $\mathbf{h}$ is the boundary traction vector, the derivative $\frac{\partial^2 \mathbf{d}}{\partial t^2}$ is taken with respect to time in the material domain, ρ_0 is the density in the initial configuration, and $\boldsymbol{S}$ is the second Piola-Kirchhoff stress tensor. The equations are considered in the initial configuration, Ω_0, which is also the material domain. The differential operator subscript, X, indicates the spatial derivatives are taken with regard to the material coordinates. Velocity is assumed to match at the boundary between the blood domain and the elastic wall. The second Piola-Kirchhoff stress tensor is obtained as follows:

$$\boldsymbol{S} = 2\frac{\partial \phi}{\partial \boldsymbol{C}}(\boldsymbol{C}, J) \tag{2.3}$$

where J =det$\boldsymbol{F}$, and $\boldsymbol{C}$ is the Cauchy-Green deformation tensor, defined as $\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}$. The parameter ϕ is a stored elastic energy of the form:

$$\phi(\bar{\boldsymbol{C}}, J) = \frac{1}{2}\mu^S\left(e^{\mathtt{tr}\bar{\boldsymbol{C}}-3} - 1\right) + \frac{1}{2}\kappa^S\left(\frac{1}{2}(J^2-1) - \mathtt{ln}J\right) \tag{2.4}$$

where μ^S is the structure shear modulus and κ^S is the structure bulk modulus. Further we define $\bar{\boldsymbol{C}} = \bar{\boldsymbol{F}}^T\bar{\boldsymbol{F}}$, and $\bar{\boldsymbol{F}} = J^{-1/3}\boldsymbol{F}$. Further details may be found in [19] and [74]. This represents an improvement on our previous work, by introducing an exponential term which will stiffen the material faster with greater deformation, which is more physiologic. The motion of the interior discretized mesh is governed by linear elastostatics, using displacement determined in the arterial wall as the boundary conditions. See e.g., [167, 159, 92, 160]. The time dependent equations are discretized and solved using a second order generalized-α integration scheme, following our previous work [19, 16].

Additionally, a backflow stabilization term was introduced on the outlets according to the presentation in [120]. It is subtracted off the left hand side of the governing equations. The term is of the form:

$$\int_{(\Gamma_t^f)_h} \beta\mathbf{w}^f \cdot \left(\rho^f\frac{\mathbf{u}\cdot\mathbf{n} - |\mathbf{u}\cdot\mathbf{n}|}{2}\mathbf{u}\right) d(\Gamma_t^f)_h \tag{2.5}$$

where β is a positive coefficient between 0 and 1. A value of 0.5 was used during these simulations. This term vansihes when the velocity vector has a component in the outward normal direction.

The simulations were all run on 64 processors, with a time step size of 0.005 seconds. Convergence was reached using 4 non-linear Newton iterations per time step, and the linear algebra was solved using GMRES with 800 Krylov-space iterations.

2.2.4 Boundary Conditions

The outflow boundary conditions are an RCR circuit [175]. The values of resistance at each outlet are determined iteratively to match the mean pressures from the catheterization report. Flow distribution to the pulmonary outlets was determined by grouping outlets into upper, medial, and lower lobes, using vessel area, and incorporating pulmonary morphometry data following our previous work [151, 112].

We include respiratory effects by incorporating both the PC-MRI data and the catheterization pressure tracings. First, the PC-MRI data is analyzed at the IVC and SVC inlets, yielding a flow waveform for one cardiac cycle. Since respiration plays little role in the flow of the SVC, we directly apply the waveform from the PC-MRI data, following our previous work[113].

To generate the IVC inflow waveform, we first take the mean flowrate in the IVC from the PC-MRI data. We then run a sample simulation under steady flow conditions using the patient-specfic mean flowrate, and further tune the resistance values of our RCR outlet BC's until we reach the correct average patient-specific pressure. We then generate two curves using Lagrage interpolation, one for inhalation and one for exhalation. The curve amplitudes are then determined iteratively until the amplitude of the pulmonary artery pressure waveforms in the simulation match the catheterization data. This is done for the FSI case with variable wall properties. All other simulations utilize the exact same boundary conditions. The cardiac component is then superimposed on the curve, keeping the mean flow rate constant. Thus, this curve respects both the mean flowrate and mean pressure (Figure 2.2). Since the area and shape of the inlet will change due to FSI, we use a uniform flow profile at the inlets for these simulations.

In each of three major sections of the vasculature, different material prop-

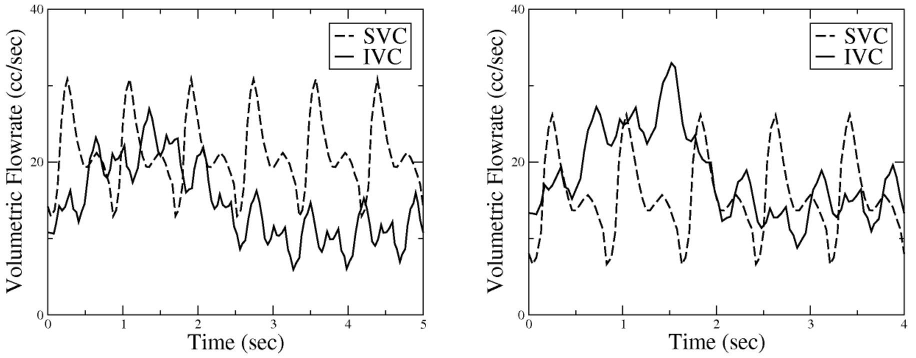


Figure 2.2: Inflow parameters for Patient 1 (left) and Patient 2 (left).

erties are used, namely the Gore-Tex conduit, the pulmonary arteries (PA's), and the SVC (Figure 2.1). The Gore-Tex conduit was modeled with an elastic coefficient of 4×10^8 dynes/cm^2, and a density of 3.3 g/cm^3. The elastic properties are on the same order of magnitude as values reported by W.L. Gore, and are many orders of magnitude stiffer than the pulmonary arteries.

The elastic modulus of the pulmonary arteries was estimated from a study of Greenfield and Griggs, in which direct masurements were made during surgery on eleven patients [91]. Blood pressure and vessel diameter were recorded, however the wall thickness was not reported. This property is necessary to determine the elastic modulus as described by the following equation:

$$E = \frac{\Delta P(R + \Delta r)^3}{\Delta r R t}$$

where ΔP is the pressure jump from diastole to peak systole, Δr is the difference in vessel radius from diastole to peak systole, R is the diastolic radius, and t is the diastolic wall thickness. The diastolic wall thickness was assumed to be 10% of the diastolic vessel radius. Using the above equation, and averaged data from the above study, we obtained an average elastic modulus for the PA's of 2.6×10^6 dynes/cm^2. The value for the SVC in patients was assumed to be 10% more elastic than the pulmonary arteries, or 2.34×10^6 dynes/cm^2, based on observations from our PCMRI data.

Wall thickness is necesary in the simulated vasculature to perform FSI. It is generally accepted that vessel wall thickness is approximately 10% of the hydraulic diameter. However, this assumption breaks down near vessel branches, junctions, and deformations such as aneurysms. We use this assumption to compute the wall thickness at all of the outlets in the model. To approximate the wall thickness in the interior, we solve a Laplace equation, $\nabla^2 T = 0$, where T is the wall thickness, using the inlet and outlet thicknesses (obtained as 10% of their effective radii) as boundary conditions. This technique was proposed in [19]. The issue of variable wall thickness was also explored in [154, 174, 155] for cerebral aneurysm simulations. The elastic deformation in the Gore-Tex region is nearly zero, and small changes in the thickness do not affect the outcome in any meaningful way.

The structural boundary conditions in previous work fixed the inlet and outlet nodes in space. Since the rest of the domain expands and contracts with pressure, the outlets become nozzle-like, which is unrealistic. In this work, we constrain the nodes on all inlet and outlet surfaces to a zero-displacement boundary condition in the normal direction, but allow the nodes to move in-plane as presented in [22]. This allows the surfaces to slide, expand, and contract with changing pressure, as expected.

2.3 Results

A chief goal of this study is to determine the effects of FSI versus rigid wall simulations of the same patient. We also aim to determine the effects of our incorporation of multiple material parameters. Rigid wall simulations were performed using identical boundary conditions as in the FSI simulations, allowing for quantitative comparisons. Additionally, a “control” FSI simulation on each patient was carried out using uniform material properties of 2.6×10^6 dynes/cm^2, and a density of 1 g/cm^3. These control simulations still incorporated variable wall thickness. We compare several hemodynamic parameters that are believed to be clinically relevant.

2.3.1 Pressure

Pressure results in the IVC for patient 1 are shown in Figure 2.3, with simulated pressure overlayed with catheterization data. There is poor agreement between raw numerical data and the catheterization data, owing to the fact that catheterization data is low-pass filtered. After applying a filter to our numerical results with a time constant of 0.3 seconds, we obtain improved agreement (Figure 2.3). Reasonable agreement is obtained for both patients with catheterization data in the IVC. In both cases the SVC is less well-matched. Since the IVC catheterization data was used to scale the inflow profile of our numerical simulation, this is an expected result.

As mentioned, our RCR outlet boundary conditions were tuned for the FSI problem to match the clinical pressure data. When these same boundary conditions are used on the rigid case, the mean pressure is increased significantly, as expected (Figure 2.4). This is in contrast to previous simulations which used a resistance only boundary condition and found no significant difference in mean pressure between FSI and rigid simulations [19]. The mean pressure difference between the rigid and FSI case is 1.3 mmHg in patient 1, or 11.8% and 1.0 mmHg in patient 2, or 10.5%, with a maximum difference at peak flow of about 2.0 mmHg in patient 1, and 1.8 mmHg in patient 2.

Effects of FSI on the amplitude and phase shift of the pressure tracings were examined. Unfiltered pressure results in the IVC from both patients are presented below in Figure 2.5. The tracing from the rigid simulation was shifted down to match the mean of the FSI simulation, for easier comparison. A damping effect is clearly visible, although no discernible phase shift occurs.

2.3.2 Wall Shear Stress (Fluid)

Shear stress in the fluid is thought to be an important clinical factor in thrombus formation, [101, 122] and a comparison of fluid wall shear stress at peak flow in both rigid simulations and FSI simulations is presented in Figure 2.6. Although the pressure differences between FSI and rigid wall simulations are slight, there are more significant differences in both patients when computing peak wall

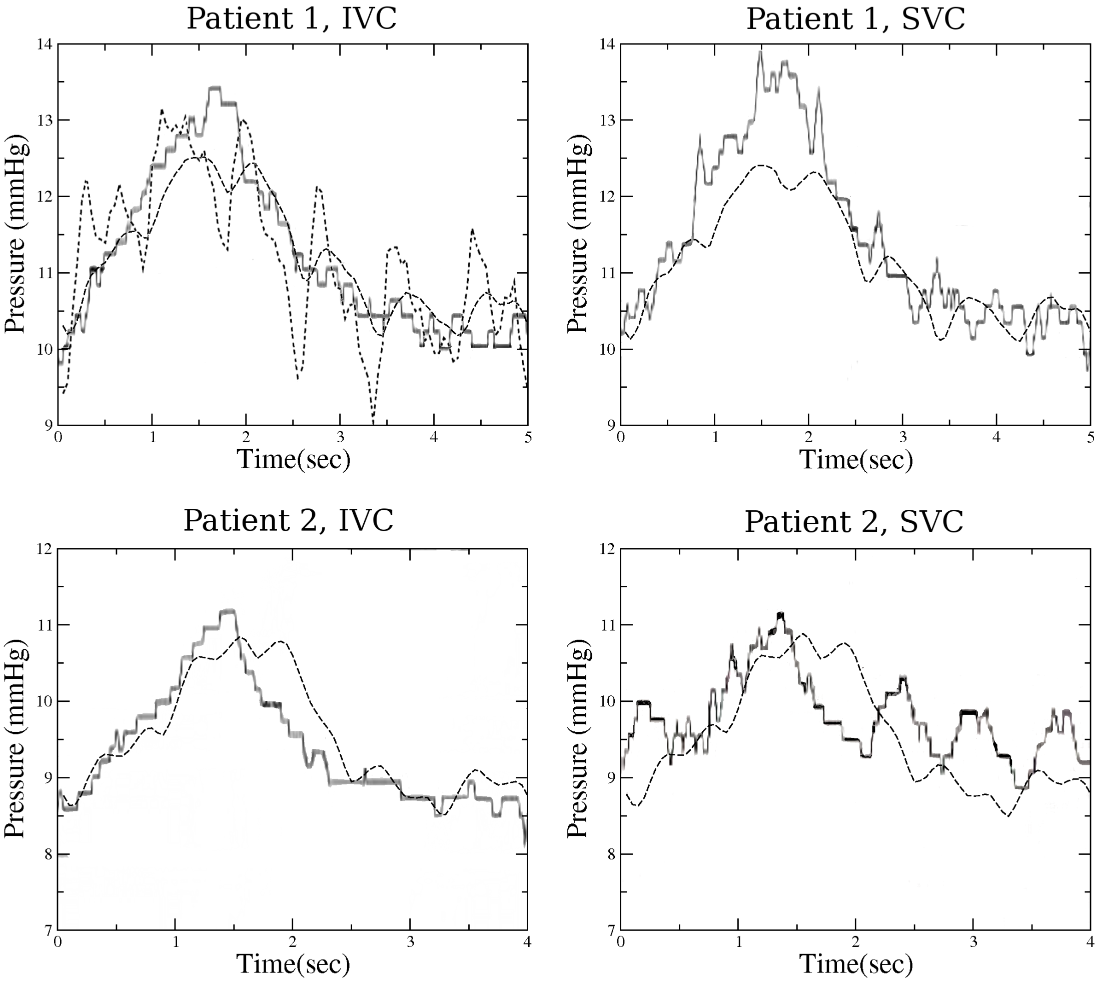


Figure 2.3: Filtered IVC (left) and SVC (right) results (dashed) compared to cath data (solid), with patient 1 in top row and patient 2 on bottom. Patient 1 IVC plot also shows unfiltered numerical data in bold dashed line.

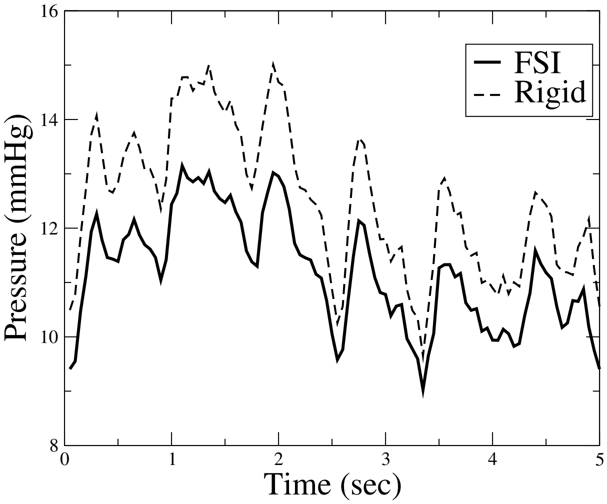


Figure 2.4: Unfiltered pressure in the IVC from FSI and rigid wall simulation.

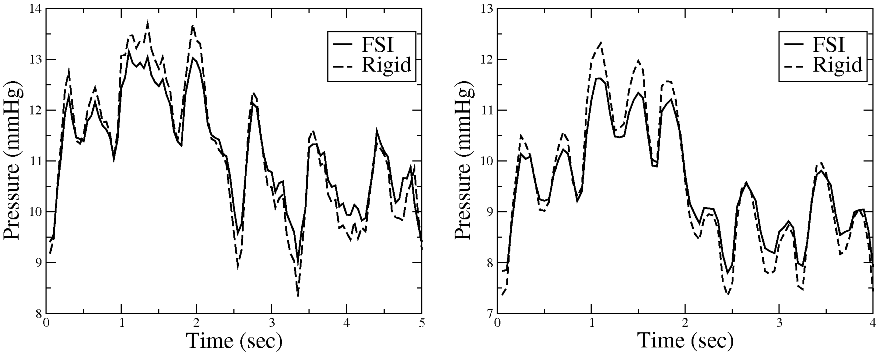


Figure 2.5: Unfiltered pressure in the IVC from FSI simulation, and adjusted Rigid simulation for both patients.

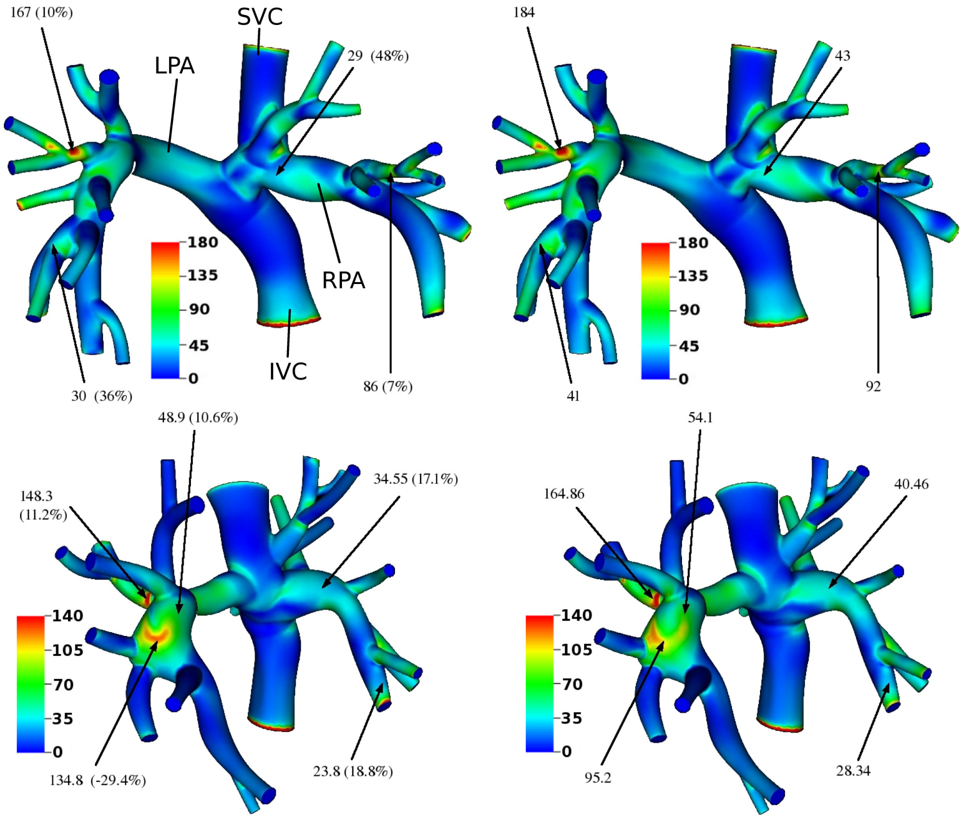


Figure 2.6: Wall shear stress (dynes/cm^2) in both FSI (left) and rigid wall simulations (right) for patient 1 (top) and 2 (bottom).

shear stress. Peak wall shear is determined by computing the wall shear for the time step that corresponds with maximum flowrate for both patients. In patient 1, the distribution of wall shear stress is qualitatively unchanged, but there are large local amplitude differences, as high as 48%. In patient 2, there is a clear qualitative difference in the RPA WSS distribution. This results in isolated spots where FSI over-estimates the wall shear stress by as much as 29% compared to the rigid case.

The control simulations are similar to the regular FSI simulations, with the largest differences coming in the IVC Gore-Tex conduit, as expected (Figure 2.7). Differences are consistently in the 10-20% range. The FSI case with a discontinuous material properties has a large shear stress gradient across the material boundary, while the control case shows a smoother profile. The effects of material property

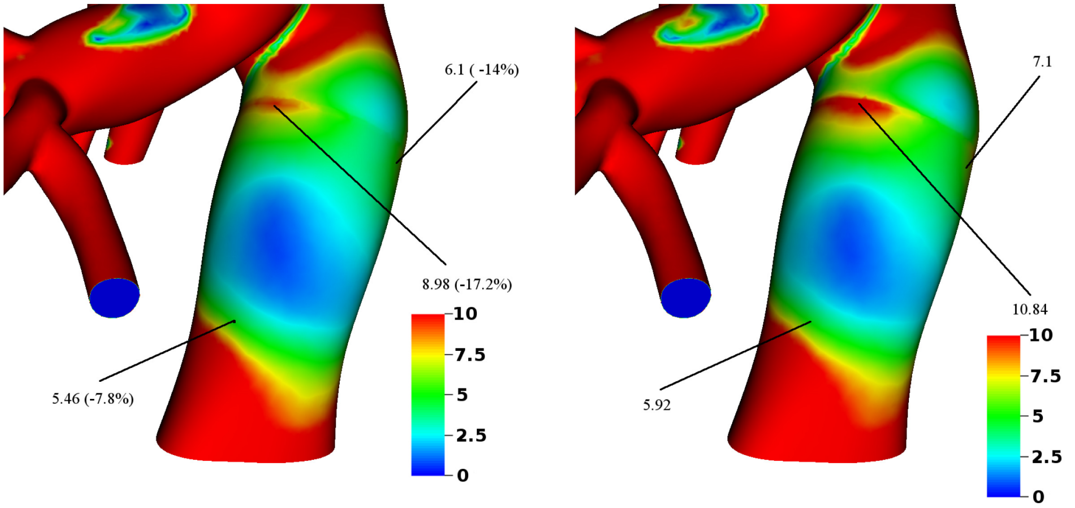


Figure 2.7: Fluid wall shear stress (dynes/cm^2) in the IVC in both the Control case (left) and the FSI case (right) in patient 1.

variance can be seen even in areas which have the same elastic properties. In the PA's, the wall shear stress of the control simulation can exhibit differences up to 5% from the standard FSI simulation (Figure 2.8).

2.3.3 Hepatic Flow Distribution

Hepatic flow distribution is of critical importance for lung development in Fontan patients due to an unknown hepatic factor present in the IVC blood [96, 133]. It has been clinically shown that the IVC flow should be well distributed to prevent the formation of pulmonary arteriovenous malformations (PAVM's), and that PAVM's can be reversed by correcting poor distribution in some patients. We calculate the percentage flow split of IVC flow to the RPA and LPA. Results from both patients, for both rigid and FSI simulations are presented in Table 2.1.

To calculate the percentage flow split, we post-process the simulation results with an advection-diffusion solver that is an implementation of the formulation found in [86]. The IVC flow is seeded with a constant scalar value, and this value is allowed to advect throughout the domain, allowing us to track the concentration of IVC flow in the pulmonary arteries. The quantity of IVC flow at

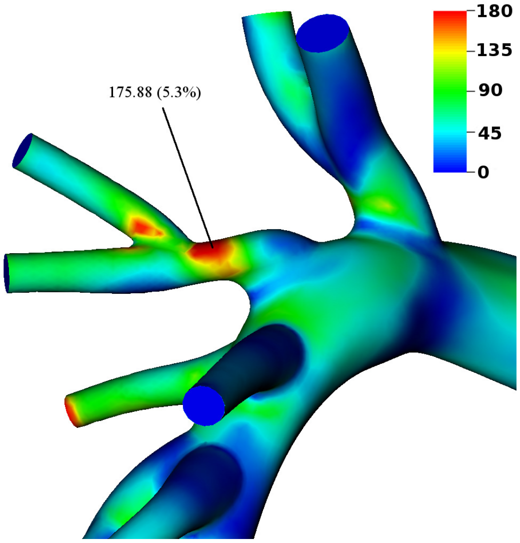


Figure 2.8: Fluid wall shear (dynes/cm^2)in the RPA of the control simulation of patient 1. Percentage difference is from normal FSI case.

Table 2.1: Calculations of IVC hepatic flow split data from both patients.

	FSI		Rigid		Control	
	RPA	LPA	RPA	LPA	RPA	LPA
Patient 1	87.8%	12.2%	87.3%	12.7%	86.5%	13.5%
Patient 2	24.3%	75.6%	24.3%	75.6%	24.9%	75.1%

the outlets and inlets is integrated over the time cycle to calculate the percentage flow split. Differences in hepatic flow distribution between FSI and rigid simulations were minimal, varying by at most 0.5%. Our "control" simulation with constant material properties impacts the solution somewhat more by making the Gore-Tex section much more elastic, but all differences remain less than 2%, which is clinically insignificant.

2.3.4 Structural Effects

The cauchy stress tensor was calculated using the second Piola-Kirchhoff stress and deformation gradient tensor during the non-rigid simulations. Of interest to us is the effect of multiple material parameters on stress and strain (deflection)

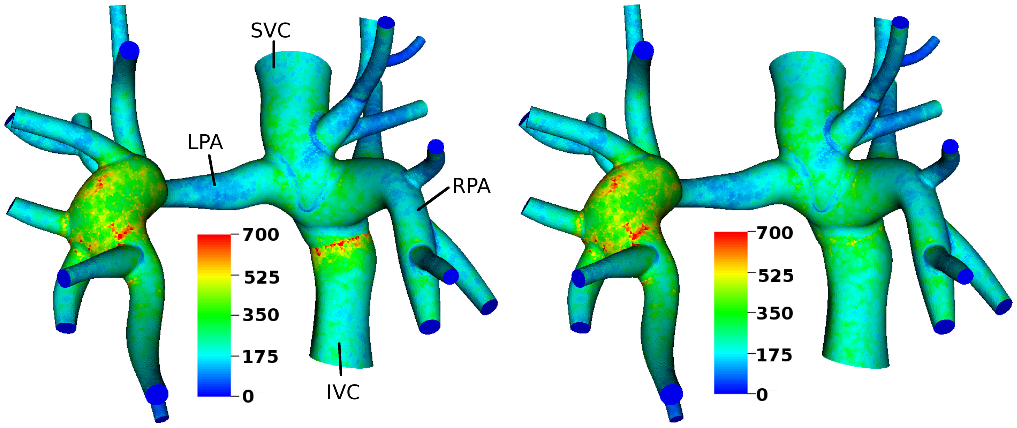


Figure 2.9: Plots of the von Mises stress (dynes/cm^2) for patient 2 on both the variable material property simulation (left) and the Control simulation (right).

in the structure. The von Mises stress is computed from the stress tensor during simulation, and we notice a large difference in IVC stress distribution between the control FSI simulation and the variable material properties FSI simulation (Figure 2.9). Stress distribution in other regions of the vasculature are not as dramatically impacted.

Displacement of the control simulation matches almost exactly with displacement of the variable material properties simulation, except most notably in the IVC conduit, which changes elasticity by an order of magnitude. Additionally, we can clearly see the dramatic impact of having switched to sliding structural BCs (Figure 2.10). Although the pressure, hepatic flow, and energy are largely unaffected, it is very clear that displacement and wall shear are highly subject to structural boundary conditions, and any work incorporating these parameters should account for this.

Other structural effects include actual displacement of the vessel wall. The maximum displacement during inhalation was recorded in each of the separate material domains, and the value of displacement at the same locations was also recorded during exhalation (Table 2.2). The large values of deflection in the Gore-Tex region are due to the sliding outlet BC's we impose on the IVC inlet. The structure is very stiff and the effect is of a wholly translational movement in this region. The variance of deflection in the Gore-Tex region is $+/-0.005$ mm, which

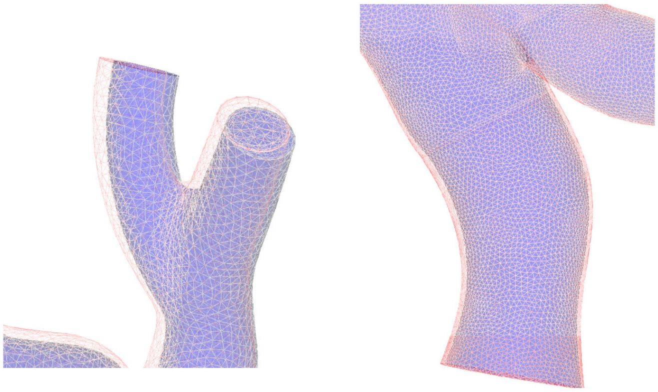


Figure 2.10: Outlets of FSI simulation (red) overlayed on rigid simulation (blue) at peak flow in patient 1 (left), and the IVC of our control simulation (red) overlayed on our normal FSI simulation (blue) at peak flow (right).

Table 2.2: Peak displacement recordings in mm of both patients at inhalation (Inh) and exhalation (Exh).

	SVC		IVC/Gore-Tex		PAs	
	Inh	Exh	Inh	Exh	Inh	Exh
Patient 1	1.37	0.98	1.06	0.8	1.67	1.20
Patient 2	0.79	0.53	0.51	0.4	1.12	0.89

is the true structural deformation in this region.

2.3.5 Energy Loss

Energy efficiency of the Fontan conduit is a widely studied indicator of Fontan performance [63, 177], although its clinical importance is yet undetermined. We calculate energy flux at each time step at each inlet and outlet, and then time-average over a respiratory cycle. Kinetic and potential energy are summed and integrated exactly over the inlet and outlet faces, using the following equation:

$$E_{effic} = \frac{\sum_{i=1}^{N_{out}} \int_{A_i} \left(p + \frac{1}{2}\rho u^2\right) \mathbf{u} \cdot \mathbf{n} dA}{\sum_{i=1}^{N_{in}} \int_{A_i} \left(p + \frac{1}{2}\rho u^2\right) \mathbf{u} \cdot \mathbf{n} dA}$$

where u is velocity, p is pressure, ρ is density, A_i is the i^{th} inlet/outlet area, and N_{in} and N_{out} are the number of inlets and outlest on the model, respectively[112].

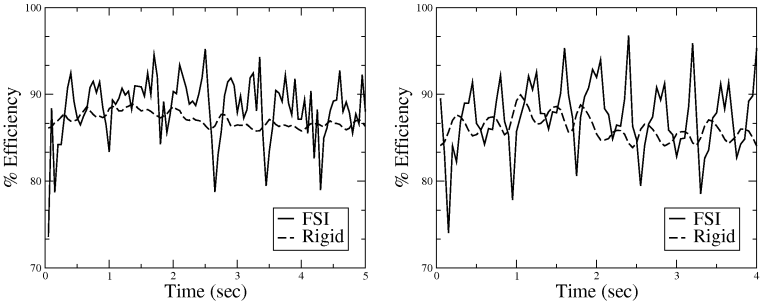


Figure 2.11: Efficiency versus time for both Patient 1 (left) and Patient 2 (right) under Rigid wall and FSI conditions.

Table 2.3: Comparison of energy efficiency for FSI, rigid, and control cases in both patients.

	FSI	Rigid	Control
Patient 1	88.4%	87.1%	88.5%
Patient 2	87.4%	86.1%	86.5%

Without time-averaging the results, we can calculate the energy efficiency for each time step (Figure 2.11). Time-averaged results are tabulated in Table 2.3. We obtain a small increase in energy efficiency after incorporating FSI. This suggests that time-averaged energy efficiency, like hepatic flow distribution, is a fairly robust parameter that is largely insensitive to small fluctuations in geometry and material properties. While the time-averaged values are robust, we note the instantaneous efficiency varies significantly in the FSI simulations, while remaining nearly constant in the rigid cases. When the inflow flowrate increases as in systole, the pressure rises and the vessel wall is displaced in the outward normal direction by the blood flow forces. This increases in lumen volume and leads to lower flowrate through the outlet branches. This, in turn, decreases the energy efficiency of the system at this time instant. On the other hand, when the inlet flowrate decreases, the corresponding decrease in the pressure forces the vascular wall to displace inward and push the blood through the outlet branches at an increased rate. This, in turn, leads to increased energy efficiency at this time instant.

2.4 Discussion

Fontan simulations have primarily been restricted to the use of rigid wall simulations. While FSI had been used in at least one previous study, its effect on multiple material parameters of potential clinical importance had not been sufficiently examined. The lack of proper previous exploration of the effects of FSI, including the known wide range of material property values, called into question the reliability of existing simulations. In this work, a systematic method for simulating patient-specific Fontan conduits with variable property fluid-structure interaction has been presented. FSI simulations were carried out on two patients using patient-specific respiration-corrected inflow profiles reconstructed from catheterization and PC-MRI data. Tetrahedral meshes with a refined boundary layer were used to model the fluid domain, and a hyperelastic structural model with a zero through-thickness stress condition was used to model the solid domain. Variable wall thickness was achieved by assuming thickness at the inlet/outlets, and using a Laplace equation to determine the thickness in the interior.

Variable wall properties were introduced, allowing for more realistic modeling of different materials anastomosed into one model. Structural boundary conditions were relaxed from previous work, and inlets and outlets were allowed to expand and "slide" within the plane. Both of these increased the physiologic realism of the simulations and were introduced to Fontan simulations for the first time.

Pressure data showed an overall agreement with patient-specific catheterization data. This was an expected result, as catheterization data was used to generate the inflow waveforms. Pressure results also showed a clear amplitude damping effect in the FSI simulations, but no discernible phase shift. This is likely due to the small size of the model. Agreement of the simulated pressure tracings with catheterization data is necessary in FSI simulations because it directly affects wall motion and WSS data. Without properly matching the pressure tracings, reliable comparisons of these quantities would not be possible.

FSI and rigid wall simulation comparisons have shown important differences, both in amplitude and qualitative distribution of wall shear stress. Simulating the same patient geometry as [19], but with improved patient-specific inflow parameters and variable material properties yielded a notable variance in wall shear stress distribution between the rigid and FSI cases. The differences in wall shear stress distribution between the two FSI simulations (control and variable wall proerty) show that changes in material modeling are indeed important and should not be neglected in future work. In the case of exercise simulations, we expect these differences to become amplified. This reinforces the need for FSI of variable material properties in Fontan simulations, particularly for prediction of thrombus formation.

Ensuring a proper flow split between left and right pulmonary arteries from the IVC is believed to be important for the prevention of arterio-venous malformations, which can lead to poor outcomes in Fontan patients. The effect of FSI on this flow split was previously unclear, and had not been examined in other FSI studies. In both patients, the effects of the FSI had little to no effect on the hepatic flow distribution. Our reported results also agree well with previously reported values using a different flow solver, and different respiratory inflow conditions on the same two patients. This demonstrates that the flow split computations are robust, and largely insensitive to minor wall fluctuations.

Instantaneous efficiency was considerably affected by FSI. This is expected, as energy is both absorbed by the structure during systole while the vessel walls expand, and elastic energy is converted back into kinematic energy during diastole when the vessel walls contract. As a result, FSI is essential if one wants to capture the time-dependent energy efficiency characteristics. However, if only the cycle-averaged energy efficiency is of interest, both rigid wall and FSI assumptions lead to very similar results. Other relevant information, such as total particle residence time, and integrated shear should also be determined. It remains unclear how hepatic flow distribution and energy efficiency may be affected by FSI during exercise cases. This will be the subject of future work.

Results presented in this study compared rigid and FSI simulations in a range of parameters. The summary of these results indicate that there is little effect of FSI on pressure tracings, hepatic flow distribution, and energy efficiency.

However, the effect of FSI on WSS and wall motion is significant and should be considered in future work, particularly for accurate prediction of thrombus formation. These results help to increase confidence in current simulation methods focusing on energy efficiency and hepatic flow distribution, as these quantities are found to be extremely robust to changes in material properties. However, reported values of WSS in current rigid wall simulations should be treated with care.

2.5 Conclusion

This work demonstrates that FSI and variable wall properties has a clear effect on clinically relevant hemodynamic parameters such as wall shear stress. It also strengthens the argument presented by DeGroff that an accurate modeling of surgical anastomosis point, and surgical materials is important, and that future research should incorporate these aspects. We have also taken into account respiratory effects, a necessary step toward making full use of patient-specific catheterization data. This is an essential step for accurate modeling of wall motion, since wall motion is primarily driven by time-varying pressurization of the vessels in the model. It also strengthens the case for reliability and robustness of previous Fontan simulation work focusing on energy loss and hepatic flow distribution, as these parameters are relatively unaffected by wall motion and wall properties.

There are several enduring limitations to patient-specific Fontan studies. Lack of patient-specific data is the main limiting factor, including exercise data, real-time respiratory flow data, and patient-specific material property data. The extent of external tissue support and translational deflection from respiration and cardiac motion remain unknown. Future work should use a multiscale closed loop lumped parameter network to better incorporate respiratory effects[103, 11]. The material properties of the venous side have considerable uncertainty associated with them, and an uncertainty quantification is in order. Additionally, hepatic flow distributions predicted by simulations should be validated clinically using lung perfusion data, and a Womersley flow profile should be implemented on the inlets.

2.6 Acknowledgements

Funding was provided by a Burroughs Wellcome Fund Career Award at the Scientific Interface, a Leducq Foundation Network of Excellence Grant, and the UCSD collaboratories program. The authors would like to acknopwledge assistance from Mahdi Esmaily Moghadam in developing the advection diffusion solver, and Dr. Frandics Chan for performing the MRI imaging.

Chapter 2, in full, is a reprint of the material as it appears in the article "Fluid–structure interaction simulations of the Fontan procedure using variable wall properties", which was published in May 2012 in Volume 28, Issue 5 of **International Journal for Numerical Methods in Biomedical Engineering**. Authors include: C.C. Long, M.-C. Hsu, J.A. Feinstein, and A.L. Marsden. The dissertation author was the primary investigator and author of this paper.

Chapter 3

Fundamentals of Isogeometric Analysis

In this chapter, we present a brief review of the fundamentals of Isogeometric Analysis (IGA). This chapter is confined to IGA using Non-Uniform Rational B-Splines (NURBS). Other choices, such as T-Splines [145], may also be used. The subject is a fast-growing and active area of research. For a more thorough discussion of mathematical developments, basis function research, geometry modeling, model quality assessment, and early applications, the reader is referred to [48], and the references therein.

3.1 Introduction

The idea of Isogeometric Analysis (IGA) was first developed in [87]. The core concept of IGA is to use models generated via Computer-Aided Design (CAD) programs directly in analysis, as opposed to using a discretization based on that model as FEM requires. The achievement of this goal immediately allows for a tighter integration between engineering design and analysis. Only one model is created for both design and analysis purposes, and analysis can be carried out with no geometric error introduced by discretization.

There are several choices of basis functions in CAD programs, and the most widely used are Non-Uniform Rational B-Splines (NURBS) [67, 132, 135].

NURBS are the industry standard for CAD software, and are well-developed, well-documented, and present a natural starting point for the development of IGA. NURBS have several useful properties for analysis, such as the ability to insert 'knots' at will, which is analogous to traditional h-refinement. The exact nature of the geometrical representation is valid for conic sections, and several efficient and stable algorithms are readily available for generating NURBS objects.

3.2 B-splines

NURBS are built from B-splines, which are in turn defined by a set of control points and a *knot vector*. In one dimension, a *knot vector* is simply a non-decreasing set of $n + p + 1$ real numbers, where n is the number of basis functions used in the B-spline, and p is the polynomial order of the B-spline. It may be written as $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$, where $\xi_i \in \mathbb{R}$ is the i^{th} *knot*, and i is the knot index, $i = 1, 2, \ldots, n + p + 1,$. The knots partition the parameter space into elements, and a given knot value, ξ_i, is a boundary between two elements. In two dimensions, each knot represents a line. In this case, the element boundaries in the physical space are the images of knot lines under the B-spline mapping.

If the knots are equally spaced, the resulting B-Spline is said to be 'uniform', and is otherwise 'nonuniform'. A knot vector's entries are non-decreasing, but entries may be repeated consecutively. If the first and last entries of the knot vector are repeated $p + 1$ times, the knot vector is said to be *open*. In CAD modeling, open knot vectors are considered standard. Repeating the first and last entries of a knot vector corresponds with the parameter space boundaries, and using an open knot vector creates useful interpolatory features at these boundaries. In one dimension, the basis functions at the parameter space boundary, $[\xi_1, \xi_{n+p+1}]$, are linear and thus interpolatory. Linear basis functions are also generated in the corners of higher dimensional NURBS objects which are created using open knot vectors, but are not generally created for interior knots.

More generally, the use of open knot vectors in higher dimensions results in unique boundary properties. We can say that a NURBS object with n_{sd} dimensions has a boundary which is itself a NURBS object with $n_{\mathrm{sd}} - 1$ dimensions. For example, each edge of two-dimensional NURBS surface is itself a one dimensional NURBS curve, and in turn, every two-dimensional surface on a three dimensional NURBS volume is itself a NURBS surface. This allows the separate construction of basis functions for the interior and boundary elements of a computational domain, and allows for imposition of Dirchlet or Neumann boundary conditions in a traditional and straightforward way (see, e.g., [176]).

B-Spline basis functions are initialized as piecewise constant, ($p = 0$) for a given knot vector as shown below:

$$N_{i,0}(\xi) = \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

To define a set of basis functions with a given polynomial order p, we then use *Cox-de Boor recursion formula* (see [50, 54]) as shown:

$$N_{i,p}(\xi) = \frac{\xi - \xi_i}{\xi_{i+p} - \xi_i} N_{i,p-1}(\xi) + \frac{\xi_{i+p+1} - \xi}{\xi_{i+p+1} - \xi_{i+1}} N_{i+1,p-1}(\xi). \tag{3.2}$$

This formula can be used to increase the polynomial order of the basis functions arbitrarily, as each call increases the order by 1.

Using the formulae given by Eqs. (3.1) and (3.2), it can be shown that the B-spline basis functions are nonnegative pointwise, that is, $N_{i,p}(\xi) \geq 0 \; \forall \xi$. They also result in a *partition of unity*, which is convenient for implementation. This property is defined such that $\forall \xi$

$$\sum_{i=1}^{n} N_{i,p}(\xi) = 1. \tag{3.3}$$

A distinguishing feature of IGA is that a p^{th} order basis function has $p - 1$ continuous derivatives across the element boundaries, allowing for 'smooth' solutions across element/knot boundaries. The support structure of B-spline functions is also unique from traditional FEM. Support of a p^{th} order B-Spline function is $p+1$ elements/knot spans, as opposed to the local support in stadard FEM. A higher-order NURBS function will thus have support over a relatively large portion of the domain. It is important to note that given a function of any polynomial order, the number of NURBS basis functions that share support of this function is only $2p+1$. This implies that the "bandwidth" of the assembled matrix is the same for IGA and traditional FEM for a given polynomial order p. Thus the use of IGA produces a linear system of the same sparsity as traditional FEM, but generates no geometric error and is $p-1$ continuous across the domain.

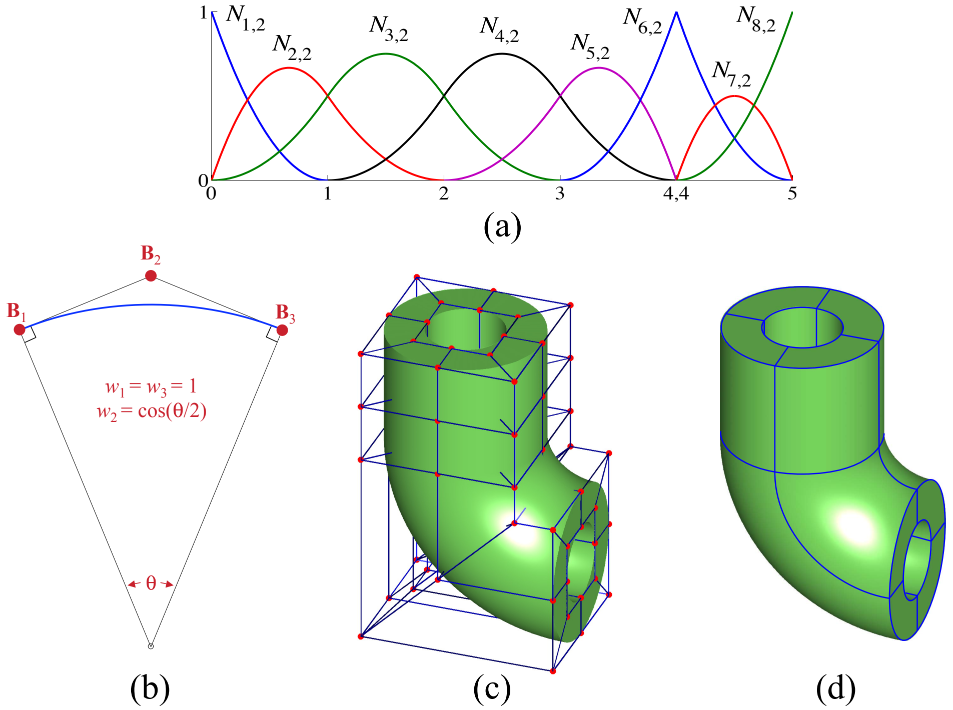


Figure 3.1: (a) Quadratic basis functions for open, nonuniform knot vector $\Xi = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$. (b) Quadratic NURBS description of a circular arc. Control points and weights are given in the figure and the underlying knot vector is $\Xi = \{0, 0, 0, 1, 1, 1\}$. (c) Section of a hollow circular pipe represented as a NURBS solid: control mesh. (d) Section of a hollow circular pipe represented as a NURBS solid: quadratic NURBS mesh.

Figure 3.1a shows the basis functions for the open, nonuniform knot vector $\Xi = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$. There is a repeated knot at $\xi = 4$, where the continuity is only of order C^0. However, this does not affect the interpolatory properties on the interval. Elsewhere, the functions are C^1-continuous, due to basis functions of order p having $p - m_i$ continuous derivatives across knot ξ_i. We use m_i to denote the multiplicity of ξ_i. Since $p = 2$ in this example, we attain C^1-continuity.

3.3 NURBS basis functions, curves, surfaces, and volumes

As previously mentioned, conic sections (such as circles and ellipses) may be represented *exactly* using NURBS. This is achieved through a projective transformation of piecewise quadratic curves. In general, a NURBS object in $\mathbb{R}^{n_{\text{sd}}}$ may be obtained by the projective transformation from an object in $\mathbb{R}^{n_{\text{sd}}+1}$ [67]. The NURBS basis governing these transformations can be written as

$$R_i^p(\xi) = \frac{N_{i,p}(\xi) w_i}{W(\xi)}, \tag{3.4}$$

where $W(\xi)$ is a *weighting function* expressed as

$$W(\xi) = \sum_{i=1}^{n} N_{i,p}(\xi) w_i, \tag{3.5}$$

where $N_{i,p}(\xi)$ is the B-spline basis function, and w_i is a positive real *weight*. Thus, the new basis given by Eqs. (3.4) and (3.5) is no longer piecewise polynomial, but piecewise rational.

A NURBS curve in $\mathbb{R}^{n_{\text{sd}}}$ is constructed by taking a linear combination of NURBS basis functions. *Control points* are analogous to nodal coordinates in traditional FEM, and are the vector coefficients of the basis functions. A piecewise-rational NURBS curve is then given by

$$\mathbf{C}(\xi) = \sum_{i=1}^{n} R_i^p(\xi) \mathbf{B}_i, \tag{3.6}$$

where R_i^p is a set of n basis functions, $w_i \in \mathbb{R}$ is a set of n given weights, and $\mathbf{B}_i$ is a set of n control points, each of which is a vector coordinate. A *control mesh* is defined as piecewise linear interpolation of the given control points. A section of a circular arc is depicted in Figure 3.1b, where a quadratic NURBS curve is used to capture this geometry exactly.

We can generalize NURBS basis functions for two- and three-dimensional shapes, as:

$$R_{i,j}^{p,q}(\xi,\eta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)w_{i,j}}{\sum_{\hat{i}=1}^{n}\sum_{\hat{j}=1}^{m} N_{\hat{i},p}(\xi)M_{\hat{j},q}(\eta)w_{\hat{i},\hat{j}}}, \tag{3.7}$$

$$R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta) = \frac{N_{i,p}(\xi)M_{j,q}(\eta)L_{k,r}(\zeta)w_{i,j,k}}{\sum_{\hat{i}=1}^{n}\sum_{\hat{j}=1}^{m}\sum_{\hat{k}=1}^{l} N_{\hat{i},p}(\xi)M_{\hat{j},q}(\eta)L_{\hat{k},r}(\zeta)w_{\hat{i},\hat{j},\hat{k}}}. \tag{3.8}$$

Here $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$, $\mathcal{H} = \{\eta_1, \eta_2, \ldots, \eta_{m+q+1}\}$, and $\mathcal{Z} = \{\zeta_1, \zeta_2, \ldots, \zeta_{l+r+1}\}$ are the knot vectors, and $w_{\hat{i},\hat{j}}$ and $w_{\hat{i},\hat{j},\hat{k}}$ are the weights for surfaces and volumes, respectively.

By introducing a two-dimensional control mesh, $\{\mathbf{B}_{i,j}\}$, where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$, we can assign distinct polynomoial orders for the separate knot vectors, denoted by p and q. Then, given knot vectors $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$ and $\mathcal{H} = \{\eta_1, \eta_2, \ldots, \eta_{m+q+1}\}$ as above, we can construct a set of NURBS basis functions, $R_{i,j}^{p,q}(\xi,\eta)$, using Eq.(3.7). We can then define a NURBS surface as

$$\mathbf{S}(\xi,\eta) = \sum_{i=1}^{n}\sum_{j=1}^{m} R_{i,j}^{p,q}(\xi,\eta)\mathbf{B}_{i,j}. \tag{3.9}$$

The same methodology may be extended to three dimensions and used to define a NURBS volume. Namely, if we are given a set of knot vectors, $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$, $\mathcal{H} = \{\eta_1, \eta_2, \ldots, \eta_{m+q+1}\}$, and $\mathcal{Z} = \{\zeta_1, \zeta_2, \ldots, \zeta_{l+r+1}\}$ with polynomial orders of p, q, and r, respectively, and a three dimensional control mesh given as $\{\mathbf{B}_{i,j,k}\}$, where $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$, and $k = 1, 2, \ldots, l$, we can construct a set of NURBS basis functions $R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta)$. The NURBS volume is then defined as

$$\mathbf{S}(\xi,\eta,\zeta) = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{l} R_{i,j,k}^{p,q,r}(\xi,\eta,\zeta)\mathbf{B}_{i,j,k}. \tag{3.10}$$

Many of the properties of NURBS curves can be generalized to surfaces and volumes due to the tensor-product construction. In all cases, the basis forms a partition of unity and is pointwise nonnegative. The number of continuous partial derivatives across element boundaries is given by the polynomial order of the associated knot vector in the parametric direction. Furthermore, any NURBS surface or volume will reduce to a standard B-spline definition if all weights are equal. This can be determined by inspection of Eqs. (3.4), (3.7), and (3.8). An example of a NURBS volume is given in Figure 3.1c, where the object is depicted together with its control mesh.

3.4 h-, p-, and k-refinement of NURBS meshes

Standard FEM practices allow for solution refinement in two ways, h- and p-refinement. Both strategies have analogs in NURBS-based IGA which are explained below. A hybrid method of refinement, which has been termed k-refinement, is unique to IGA. This method is also explained and introduced.

3.4.1 h-refinement

The h-refinement strategy in standard FEM is a simple reduction in a characteristic element length scale. This is achieved in IGA through the process of *knot insertion*. Given a particular knot vector, $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{n+p+1}\}$, we desire to insert m knots to refine the vector, without modifying the original curve. We then construct some new knot vector, $\bar{\Xi} = \{\bar{\xi}_1 = \xi_1, \bar{\xi}_2, \ldots, \bar{\xi}_{n+m+p+1} = \xi_{n+p+1}\}$, such that $\Xi \subset \bar{\Xi}$. We can then create a new set of $n+m$ basis function by applying the Cox–de Boor recursion formula (Eqns. 3.1 and 3.2) on $\bar{\Xi}$. A new set of $n+m$ control points and weights must also be formed. A linear combination of the original control points and weights is used, via a linear transformation given in [49, 48]. An inserted knot value which is alread present in the original knot is repeated in $\bar{\Xi}$. The continuity of the overall NURBS curve is not affected, but the continuity of the NURBS basis is reduced for each repeated index. This results in a solution with a reduced order continuity across the repeated knot index.

3.4.2 p-refinement

In standard FEM, p-refinement entails raising the polynomial order of the underlying basis functions. NURBS-based IGA has a direct analog to this approach, which is sometimes called *degree elevation*. The underlying basis functions which are used to define the geometry are polynomials, and degree elevation raises the polynomial order used for this basis by using Eq. 3.2. When this order is increased, the multiplicity of the knots must also be increased if we desire to preserve the continuity of the original curve. That is, for each polynomial order we add, we must repeat each knot value in the respective knot vector by exactly one. No new knot values are added, entries are merely repeated. An example for the knot vector Ξ is shown after refinement from $p = 2$ to $p = 3$:

$$\begin{aligned} \Xi_{p=2} &= \{0,0,0,1,2,3,4,4,5,5,5\} \\ \Xi_{p=3} &= \{0,0,0,0,1,1,2,2,3,3,4,4,4,5,5,5,5\}. \end{aligned} \tag{3.11}$$

This preserves the original $p-m_i$ continuity, and the geometry remains unaffected. Typically, p-refinement is used as a precursor to k-refinement, but may also be used as described here.

3.4.3 k-refinement

In standard FEM, h- and p-refinement demonstrate a commutative property. That is, given a mesh, $\mathbf{M}$, we can perform h-refinement first, and then perform p-refinement, and arrive at a new mesh $\bar{\mathbf{M}}$. If the order of operations is reversed, and we instead perform the p-refinement first, the end result is still uniquely mesh $\bar{\mathbf{M}}$. In IGA, this is not the case.

Consider, for example, a curve of polynomial order p. We may perform a knot insertion (h-refinement) for a single unique knot value, $\bar{\xi}$. After the knot insertion, the number of continuous derivatives in the underlying basis functions at $\bar{\xi}$ is $p-1$. Consider a degree elevation on this same knot to polynomial order q, where $q > p$. As discussed in Sec. 3.4.2, the entry of $\bar{\xi}$ in the degree elevated knot vector has a multplicity of $q-p$. Since each repeated entry reduces continuity in the derivatives by one, we can say that after performing an h- and p-refinement, we can express the continuity of the function at knot $\bar{\xi}$ as

$$\begin{aligned} \bar{\xi}_{cd} &= q - (q - p) - 1 \\ &= p - 1, \end{aligned} \tag{3.12}$$

where $\bar{\xi}_{cd}$ denotes that number of continuous derivatives in the basis functions at knot $\bar{\xi}$.

If the refinement is carried out in reverse order, then after the initial p-refinement, we have a continuity of $p - 1$ at all existing knots. However, the insertion of knot $\bar{\xi}$ *after* the p-refinement results in continuity of $q-1$ at knot $\bar{\xi}$, and $p-1$ for all other pre-existing knots. Thus, the two operations are not commutative in IGA. It should further be noted that we can often perform p-refinement on the coarsest possible B-spline curve or surface, and only then perform knot insertions. This results in all internal knots sharing the higher order $q - 1$ continuity at knot boundaries. This is the procedure known as k-refinement. Using traditional FEM will always result in uniform C^0 continuity, and thus there is no direct analog for this method. Additionally, the traditional p-refinement method discussed in Sec. 3.4.2 results in a proliferation in the number of nodes because continuity is maintained through refinement. However, in k-refinement, this growth is limited. Further discussion may be found in [48].

3.5 Analysis Framework

The previous sections have given a brief background and introdcution to NURBS basis functions and volumes, as well as a brief discussion of refinement procedures. We now provide an introduction to the IGA framework as an extension of tradtional FEM frameworks. In the following section, all B-splines are referred to as "NURBS", for simplicity of notation. In this spirit, the term $N(\boldsymbol{\xi})$ may be used for any basis function, regardless of dimension or rationality.

The following section will include various FEM notations, which are outlined here. A physical domain will be referred to by Ω, and the parametric domain as $\hat{\Omega}$. As is standard, a rectangle and cuboid are used as parametric domains for two and three dimensions, respectively. The term *patch* may refer to either Ω or $\hat{\Omega}$, and in general is defined as a single NURBS volume or surface. While a single patch can be used to model several complex geometries, multiple patches can also be used for complex cases. In this case, the two (or more) patches are solved simultaneously, and the patches are merged with C^0 continuity across the boundary. A bijective mapping between the geometric domain and the parametric domain will be used and written as follows: $\mathbf{x} : \hat{\Omega} \to \Omega$ corresponds to the mapping onto the geometric domain, and $\mathbf{x}^{-1} : \Omega \to \hat{\Omega}$ maps onto the parametric domain. Traditional finite element methods assume that every element in the mesh is an image of the same parametric element, and the parameter space is local to the element. A key distinction in NURBS-based analysis is that the parameter space is local to the patch and each physical element is an image of its own parametric element.

A *knot span* is a region bounded by knots, which define an element domain. As previously discussed, a knot in one dimension is a point along the NURBS curve. The extension to two and three dimensions is straightforward, and results in NURBS curves and surfaces, respectively. As previously established, basis functions are C^{p-m} continuous across knot boundaries, where p is the polynomial order of the basis function, and m is the multiplicity of the knot entry. The element itself is considered to be a NURBS mapping of a knot span. Knot spans are denoted by $\hat{\Omega}^e$, and Ω^e, where $e = 1, \ldots, n_{\text{el}}$ and n_{el} denotes the number of elements. $\hat{\Omega}^e$ and Ω^e refer to the knot span in the parametric and physical spaces, respectively. Thus, we can conclude:

$$\hat{\Omega} = \bigcup_{e=1}^{n_{\text{el}}} \hat{\Omega}^e, \tag{3.13}$$

$$\Omega = \bigcup_{e=1}^{n_{\text{el}}} \Omega^e. \tag{3.14}$$

The indices $A, B, C, \ldots$ and $a, b, c, \ldots$ are used to identify global and local basis functions, respectively. Each alphabetic index has a numeric range of either $1, \ldots, n_{\text{np}}$ or $1, \ldots, n_{\text{en}}$ for global and local basis indices, respectively. The variable

n_{np} is taken to be the total number of basis functions in the NURBS object, and n_{en} is taken as the total number of basis functions supported on a given element. For simplicity of presentation, summation notation is used such that double and triple summations for two and three dimensional objects are represented as a single summation over the global or local index, as appropriate. Given the coordinates of the control points, $\mathbf{x}_A$ or $\mathbf{x}_a$, we can write the geometric mapping for a patch as follows:

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_{A=1}^{n_{\mathrm{np}}} \mathbf{x}_A N_A(\boldsymbol{\xi}). \tag{3.15}$$

This mapping also holds strictly on the element level:

$$\mathbf{x}(\boldsymbol{\xi}) = \sum_{a=1}^{n_{\mathrm{en}}} \mathbf{x}_a N_a(\boldsymbol{\xi}). \tag{3.16}$$

The basis functions $N_A(\boldsymbol{\xi})$ are not required to be interpolatory. The local shape functions, $N_a(\boldsymbol{\xi})$, are computed based on the global shape functions. This is performed using a simple mapping array known as an *IEN array*. The IEN array is identical to the usage of IEN arrays in traditional FEM (see e.g., [81]).

Given a set of control variables on the parametric domain, u_A, we may then define a solution array, $u^h(\boldsymbol{\xi})$, on the parametric domain as follows:

$$u^h(\boldsymbol{\xi}) = \sum_{A=1}^{n_{\mathrm{np}}} u_A N_A(\boldsymbol{\xi}). \tag{3.17}$$

We may then naturally define the solution field in the physical domain as

$$u^h(\mathbf{x}) = \sum_{A=1}^{n_{\mathrm{np}}} u_A N_A(\boldsymbol{\xi}(\mathbf{x})). \tag{3.18}$$

We may then employ an identical *isoparametric construction* of the basis functions in the physical space as in traditional FEM:

$$N_A(\mathbf{x}) = N_A(\boldsymbol{\xi}(\mathbf{x})). \tag{3.19}$$

This states that the discrete geometry and solution are computed in the same function space. While this is unchanged from traditional FEM, the function space of NURBS can capture complex geometries without introducing geometric error.

Using the constructed IEN array, we can restrict the basis functions and the solution field to the local element level:

$$N_a(\mathbf{x}) = N_a(\boldsymbol{\xi}(\mathbf{x})), \tag{3.20}$$

$$u^h(\mathbf{x}) = \sum_{a=1}^{n_{\mathrm{en}}} u_a N_a(\mathbf{x}). \tag{3.21}$$

Given a definition of the basis functions and solution field at the element level, we may finally proceed with an element-by-element assembly routine to construct appropriate left-hand-side matrices and residual vectors. A local, element-based left-hand-side matrix and right-hand side residual vector is constructed for each element, and is then assembled into a global left-hand-side matrix and right-hand side residual vector. The local-to-global assembly is directly analogous to standard FEM, and requires no special treatment. Once assembled, the system can be solved with any appropriate linear solver.

A Note on Quadrature: All work presented within this dissertation uses a Gaussian quadrature rule defined on each individual parametric element, which is identical to quadrature approaches in traditional FEM. This allows for accurate evaluation of left-hand-side and right-hand-side data structures, and presents no issues within an IGA framework. However, this approach does not make use of the underlying continuity of the basis functions across knot spans. Thus, more efficient approaches to quadrature that are unique to NURBS-based IGA warrant exploration. One such proposed approach which uses the continuity of the basis functions in the formulation is outlined in [88].

Chapter 4

Isogeometric Analysis of Lagrangian Hydrodynamics: Axisymmetric Formulation in the *rz*-Cylindrical Coordinates

As submitted to ***Journal of Computational Physics****.*

A recent Isogeometric Analysis (IGA) formulation of Lagrangian shock hydrodynamics [14] is extended to the 3D axisymmetric case. The Euler equations of compressible hydrodynamics are formulated using the rz-cylindrical coordinates, and are discretized in the weak form using NURBS-based IGA. Artificial shock viscosity and internal energy projection are added to stabilize the formulation. The resulting discretization exhibits good accuracy and robustness properties. It also gives exact symmetry preservation on the appropriately constructed meshes. Several benchmark examples are computed to examine the performance of the proposed formulation.

4.1 Introduction

Axisymmetric formulations, if applicable to a given problem class, reduce the complexity of 3D problems to that of 2D, resulting in significant computational cost savings. As a result, it is common practice to employ axisymmetric formulations in Lagrangian gas dynamics computations. However, it is desirable that the numerical schemes for axisymmetric formulations, besides being robust and accurate, are able to exactly preserve spherical, cylindrical, and planar symmetries in the solution. The importance of preserving spherical symmetry in Lagrangian simulations is well recognized. For instance, for inertial confinement fusion simulations, small departures from spherical symmetry coupled with strong compression may be amplified by Rayleigh–Taylor instability, leading to large symmetry errors, and, more importantly, uncertainty as to whether nonsymmetric results are due to numerical errors or the underlying physics.

Several methods have been previously proposed for better symmetry preservation. In [108] the authors developed a curvilinear differencing technique, while in [40] the authors proposed a modified gradient approach, both in the context of finite volume methods. The proposed techniques exactly preserve spherical symmetry, however their spatial accuracy is restricted to second order for the kinematic variables, and first order for the thermodynamic variables. Higher-order methods for Lagrangian shock hydrodynamics, in the context of finite elements, were first proposed in [58]. While higher-order solution accuracy was achieved and symmetry preservation was greatly improved compared to lower-order approaches, exact cylindrical or spherical symmetry preservation was not attained. More recently, in [14], the authors developed a formulation of Lagrangian shock hydrodynamics using Isogeometric Analysis (IGA) [87, 48] based on Non-Uniform Rational B-Spline (NURBS). In this framework, owing to the properties of NURBS functions, both higher-order accuracy and symmetry preservation were attained. Furthermore, the increased smoothness of the NURBS basis functions gave better per-degree-of-freedom-accuracy than standard higher-order finite elements, which are only C^0-continuous.

In this study, we extend the NURBS-based IGA discretization of the Euler equations from [14] to the 2D rz-cylindrical coordinate system. We numerically demonstrate the higher-order accuracy and exact spherical symmetry preservation properties (under exact integration) of the proposed technique for the axisymmetric case. In Section 4.2 we present the Euler equations of gas dynamics in the weak form. Both the continuous and semi-discrete formulations are shown. In Section 4.3, assuming the axisymmetric case, we introduce the kinematics in the rz coordinate system, and give the weak, semi-discrete formulation of the compressible Euler equations in this coordinate system. The resultant formulation is the so-called *volume-weighted* case. We then show how the *area-weighted* case may be obtained from the volume-weighted formulation by a suitable change of the test functions. In Section 4.4 we compute several numerical examples focusing on the accuracy and symmetry preservation characteristics of the proposed formulation. The numerical tests are done using the volume-weighted formulation. In Section 4.5 we draw conclusions and outline future research directions.

4.2 Governing equations of Lagrangian hydrodynamics: The general case

We begin with the variational formulation of the Euler equations of gas dynamics in the Lagrangian frame: find $\mathbf{v} \in \mathcal{S}$, such that $\forall \mathbf{w} \in \mathcal{V}$:

$$\int_{\Omega_t} \mathbf{w} \cdot \rho \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} \, \mathrm{d}\Omega + \int_{\Omega_t} \nabla \mathbf{w} : \boldsymbol{\sigma} \, \mathrm{d}\Omega - \int_{(\Gamma_t)_\mathrm{h}} \mathbf{w} \cdot \mathbf{h} \, \mathrm{d}\Gamma = 0, \tag{4.1}$$

where

$$\mathbf{v} = \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}, \tag{4.2}$$

is the flow velocity, $\mathbf{x}$ are the spatial coordinates of the current configuration $\Omega_t \in \mathbb{R}^{n_\mathrm{sd}}$, ρ is the density in the current configuration, $\boldsymbol{\sigma}$ is the Cauchy stress, and $\mathbf{h}$ is the applied traction vector on $(\Gamma_t)_\mathrm{h}$. Also in Eq. (4.1), $\mathbf{w}$ are the test functions, and $\mathcal{S}$ and $\mathcal{V}$ as the appropriate function spaces for $\mathbf{v}$ and $\mathbf{w}$, respectively.

The formulation given by Eq. (4.1) is augmented with additional equations of mass conservation in the Lagrangian description,

$$\rho_0 = \rho J, \tag{4.3}$$

$$J = \det \frac{\partial \mathbf{x}}{\partial \mathbf{X}}, \tag{4.4}$$

energy balance,

$$\rho \frac{\mathrm{d}e}{\mathrm{d}t} = \nabla^s \mathbf{v} : \boldsymbol{\sigma}(\mathbf{x}, e), \tag{4.5}$$

and the constitutive law,

$$\boldsymbol{\sigma}(\mathbf{x}, e) = -p\mathbf{I}, \tag{4.6}$$

$$p = (\gamma - 1)\rho e. \tag{4.7}$$

Here, $\mathbf{X}$ are the coordinates of the reference configuration $\Omega_0 \in \mathbb{R}^{n_{\mathrm{sd}}}$, which is the configuration occupied by the fluid material particles at $t = 0$, p is the pressure, e is the internal energy, and γ, a positive constant, is the adiabatic index. In Eq. (4.5) ∇^s is the symmetric gradient, and

$$\nabla^s \mathbf{v} = \frac{1}{2}(\nabla \mathbf{v} + \nabla \mathbf{v}^T) \tag{4.8}$$

is the strain rate. Equations (4.6)–(4.7) assume the that fluid is an ideal gas. All the time derivatives in the above equations are taken holding the material coordinates $\mathbf{X}$ fixed.

To approximate the Euler equations at the discrete level, it is common to introduce an artificial shock viscosity in the formulation [123, 39] as well as project the internal energy variable onto a suitable lower-dimensional subspace (see, e.g., [60]). In this case, the semi-discrete counterpart of Eq. (4.1) becomes: find $\mathbf{v}^h \in \mathcal{S}^h$, such that $\forall \mathbf{w}^h \in \mathcal{V}^h$:

$$\int_{\Omega_0} \mathbf{w}^h \cdot \rho_0 \frac{\mathrm{d}\mathbf{v}^h}{\mathrm{d}t}\, \mathrm{d}\Omega + \int_{\Omega_t} \nabla \mathbf{w}^h : \boldsymbol{\sigma}^h(\mathbf{x}^h, \mathbf{v}^h, \bar{e})\, \mathrm{d}\Omega - \int_{(\Gamma_t)_\mathrm{h}} \mathbf{w}^h \cdot \mathbf{h}\, \mathrm{d}\Gamma = 0, \tag{4.9}$$

where the stress tensor is now modified to include the shock viscosity terms as

$$\boldsymbol{\sigma}^h(\mathbf{x}^h, \mathbf{v}^h, \bar{e}) = -(\gamma - 1)\rho \bar{e}\mathbf{I} + 2\mu_{sh} \nabla^s \mathbf{v}^h. \tag{4.10}$$

In Eqs. (4.9) and (4.9),

$$\bar{e} = \Pi e, \tag{4.11}$$

where Π is a projection operator. In [60] the authors approximate $\bar{e}$ in a space of discontinuous finite elements that is one order lower than that of the kinematic variables, while in [14], and in this work, we apply a projector to a discrete, integration-point representation of the internal energy. Note that in Eq. (4.9) the inertial terms simplify owing to conservation of mass. In Eq. (4.10), μ_{sh} is the shock viscosity parameter defined in [14].

4.3 Governing equations of Lagrangian hydrodynamics: Axisymmetric formulation

In this section, we present the axisymmetric version of the formulation given by Eq. (4.9). We first briefly recall the kinematics in the rz-cylindrical coordinate system, where we assume axisymmetry. Then we introduce the kinematics of the rz coordinate system directly into Eq. (4.9) to arrive at the semi-discrete formulation.

4.3.1 Kinematics in the *rz* coordinate system

The Cartesian coordinates of the current-configuration position vector $\mathbf{x} = (x, y, z)^T$ may be expressed in terms of their rz-cylindrical-coordinate counterparts $\mathbf{r} = (r, z, \theta)^T$ as

$$x = r\cos\theta, \tag{4.12}$$

$$y = r\sin\theta, \tag{4.13}$$

$$z = z. \tag{4.14}$$

Likewise, the Cartesian coordinates of the reference-configuration position vector $\mathbf{X} = (X, Y, Z)^T$ may be expressed in terms of their rz-cylindrical-coordinate counterparts $\mathbf{R} = (R, Z, \Theta)^T$ as

$$X = R\cos\Theta, \tag{4.15}$$

$$Y = R\sin\Theta, \tag{4.16}$$

$$Z = Z. \tag{4.17}$$

The deformation gradient may be expressed as

$$\frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{R}} \frac{\partial \mathbf{R}}{\partial \mathbf{X}} = \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \frac{\partial \mathbf{r}}{\partial \mathbf{R}} \frac{\partial \mathbf{x}}{\partial \mathbf{R}}^{-1}, \tag{4.18}$$

and its determinant may be computed as

$$J = \det \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \det \frac{\partial \mathbf{x}}{\partial \mathbf{r}} \det \frac{\partial \mathbf{r}}{\partial \mathbf{R}} \det \frac{\partial \mathbf{x}}{\partial \mathbf{R}}^{-1} = r J_{2D} \frac{1}{R} = J_{2D} \frac{r}{R}, \tag{4.19}$$

where

$$J_{2D} = \det \begin{bmatrix} \frac{\partial r}{\partial R} & \frac{\partial r}{\partial Z} \\ \frac{\partial z}{\partial R} & \frac{\partial z}{\partial Z} \end{bmatrix}. \tag{4.20}$$

In transforming from Eq. (4.19) to Eq. (4.20) we assume that $\theta = \Theta$, which is the axisymmetry condition.

The gradient and divergence operators in the rz coordinate system are given as follows. Let $\mathbf{u} = (u_r, u_z, u_\theta)^T$ be the axisymmetric vector (i.e., $u_\theta = 0$). The gradient of $\mathbf{u}$ (with respect to the current configuration coordinates) is given by

$$\nabla \mathbf{u} = \begin{bmatrix} \nabla_{2D}\mathbf{u} & \mathbf{0} \\ \mathbf{0}^T & \frac{u_r}{r} \end{bmatrix}, \tag{4.21}$$

where

$$\nabla_{2D}\mathbf{u} = \begin{bmatrix} \frac{\partial u_r}{\partial r} & \frac{\partial u_r}{\partial z} \\ \frac{\partial u_z}{\partial r} & \frac{\partial u_z}{\partial z} \end{bmatrix}, \tag{4.22}$$

is the Cartesian-like 2D gradient. Its symmetrization is given by

$$\nabla^s_{2D}\mathbf{u} = \frac{1}{2}(\nabla_{2D}\mathbf{u} + \nabla_{2D}\mathbf{u}^T). \tag{4.23}$$

The divergence, which is the trace of the corresponding 3D gradient, becomes

$$\nabla \cdot \mathbf{u} = \frac{\partial u_r}{\partial r} + \frac{\partial u_z}{\partial z} + \frac{u_r}{r}. \tag{4.24}$$

Finally, it can be shown (see, e.g., [59]) that the components of the axisymmetric velocity and acceleration vectors in the rz coordinate system are

$$v_r = \frac{\mathrm{d}r}{\mathrm{d}t}, \tag{4.25}$$

$$v_z = \frac{\mathrm{d}z}{\mathrm{d}t}, \tag{4.26}$$

and

$$a_r = \frac{\mathrm{d}v_r}{\mathrm{d}t}, \tag{4.27}$$

$$a_z = \frac{\mathrm{d}v_z}{\mathrm{d}t}, \tag{4.28}$$

respectively.

4.3.2 The *rz* formulation of shock hydrodynamics

Introducing the rz coordinate system kinematics shown in the previous section into the formulation given by Eq. (4.9), and assuming the solution of the Euler equations is homogeneous in the θ-direction, we obtain the following variational equations for v_r^h and v_z^h, the discrete velocity components in the radial and axial directions, respectively: find $\mathbf{v}^h = (v_r^h, v_z^h)^T$, such that $\forall \mathbf{w}^h = (w_r^h, w_z^h)^T$:

$$\begin{aligned}\int_{A_0} \rho_0 (w_r^h \frac{\mathrm{d}v_r^h}{\mathrm{d}t} + w_z^h \frac{\mathrm{d}v_z^h}{\mathrm{d}t}) \; R\mathrm{d}R\mathrm{d}Z - \int_{A_t} (\frac{\partial w_r^h}{\partial r} + \frac{\partial w_z^h}{\partial z} + \frac{w_r^h}{r}) p \; r\mathrm{d}r\mathrm{d}z \\ + \int_{A_t} (\nabla_{2D}^s \mathbf{w}^h : 2\mu_{sh} \nabla_{2D}^s \mathbf{v}^h + \frac{w_r^h}{r} 2\mu_{sh} \frac{v_r^h}{r}) \; r\mathrm{d}r\mathrm{d}z = \int_{S_t} (w_r^h h_r + w_z^h h_z) \; r\mathrm{d}S.\end{aligned} \tag{4.29}$$

Here, A_0 and A_t are the "meridian cuts" that define the body of revolution in the reference and current configurations, respectively, S_t is the boundary curve of the meridian cut where the traction boundary conditions are prescribed, and h_r and h_z are the components of the prescribed traction vector $\mathbf{h}$ in the rz coordinate system. The resulting formulation given by Eq. (4.29) is similar to that in [33, 59].

The above rz formulation is augmented with the additional equation for energy conservation given by

$$\rho \frac{\mathrm{d}e}{\mathrm{d}t} = -(\frac{\partial v_r^h}{\partial r} + \frac{\partial v_z^h}{\partial z} + \frac{v_r^h}{r}) p + \nabla_{2D}^s \mathbf{v}^h : 2\mu_{sh} \nabla_{2D}^s \mathbf{v}^h + \frac{v_r^h}{r} 2\mu_{sh} \frac{v_r^h}{r}, \tag{4.30}$$

the equation of state given by Eq. (4.7), projection of the internal energy given by Eq. (4.11), and Lagrangian form mass conservation given by Eq. (4.3), where the Jacobian determinant is computed based on Eqs. (4.19) and (4.20).

To develop a fully discrete formulation of the Lagrangian hydrodynamics equations we adopt the methodology in [14]. The linear momentum and energy equations are integrated in time using the total-energy-conserving, second-order Runge–Kutta scheme (see also [58, 60, 14]). Energy conservation for the axisymmetric case is a consequence of the general case presented in [14]. Gaussian quadrature is employed to evaluate the integrals in Eq. (4.29). For a NURBS discretization of order p, $p+1$ Gauss points are typically used in each tensor-product direction, however, higher-order quadrature is used in some cases to improve solution accuracy. The internal energy projection is performed directly on its Gauss-point values in each element as in [14]. The shock viscosity parameter μ_{sh} is defined as in [60, 14], with the only modification being that the measure of compression $|\Delta_s \mathbf{v}^h|$ and the directional length scale h_s make use of $\nabla^s_{2D}\mathbf{v}^h$, the 2D strain rate on the meridian cut. The consistent mass matrix, which is time-independent [29], is computed once at program execution. The system of linear equations is solved using the Conjugate Gradient technique with diagonal scaling (see, e.g., [137]). Due to the academic nature of the problems shown in the next section, and the emphasis on symmetry preservation, the system of linear equations is solved to machine precision.

Remark: The formulation given by Eq. (4.29) is the so-called volume-weighted formulation. Alternatively, considering only the Galerkin terms in Eq. (4.29), and changing variables to the current configuration in the inertial terms, the formulation may be written as: find $\mathbf{v}^h = (v_r^h, v_z^h)^T$, such that $\forall \mathbf{w}^h = (w_r^h, w_z^h)^T$:

$$\begin{aligned}\int_{A_t} \rho(rw_r^h\frac{\mathrm{d}v_r^h}{\mathrm{d}t} + rw_z^h\frac{\mathrm{d}v_z^h}{\mathrm{d}t})\ \mathrm{d}r\mathrm{d}z - \int_{A_t}(\frac{\partial rw_r^h}{\partial r} + \frac{\partial rw_z^h}{\partial z})p\ \mathrm{d}r\mathrm{d}z \\ = \int_{S_t}(rw_r^h h_r + rw_z^h h_z)\ \mathrm{d}S. \qquad (4.31)\end{aligned}$$

Replacing the test functions as $\tilde{w}_r^h \leftarrow rw_r^h$ and $\tilde{w}_z^h \leftarrow rw_z^h$ yields the formulation:

find $\mathbf{v}^h = (v_r^h, v_z^h)^T$, such that $\forall \tilde{\mathbf{w}}^h = (\tilde{w}_r^h, \tilde{w}_z^h)^T$:

$$\int_{A_t} \rho(\tilde{w}_r^h \frac{\mathrm{d}v_r^h}{\mathrm{d}t} + \tilde{w}_z^h \frac{\mathrm{d}v_z^h}{\mathrm{d}t}) \ \mathrm{d}r\mathrm{d}z - \int_{A_t} (\frac{\partial \tilde{w}_r^h}{\partial r} + \frac{\partial \tilde{w}_z^h}{\partial z}) p \ \mathrm{d}r\mathrm{d}z = \int_{S_t} (\tilde{w}_r^h h_r + \tilde{w}_z^h h_z) \ \mathrm{d}S. \tag{4.32}$$

The formulation given by Eq. (4.32) is referred to as the area-weighted formulation (see, e.g., [33]). It may be stabilized by shock viscosity terms, which only depend on the 2D strain rate on the meridian cut. While the area-weighted formulation may appear more attractive from the implementation standpoint (the factor r is no longer present in the integrals), it leads to a mass matrix that needs to be recomputed at each step. Furthermore, building in total energy conservation into the discrete formulation is no longer as straightforward as for the volume-weighted case. (We note, however, that this was recently accomplished in [12] in the context of a finite volume technique.) In the numerical examples presented in this paper only the volume-weighted formulation is considered.

4.4 Numerical results

In this section, we present simulations for several benchmark problems in shock hydrodynamics. The proposed formulation is examined from the standpoint of solution accuracy and symmetry preservation characteristics. In all cases, unless stated otherwise, quadratic NURBS of full continuity (C^1), and three Gaussian quadrature points in each tensor-product direction are employed.

4.4.1 Coggeshall–Meyer-ter-Vehn problem

The Coggeshall–Meyer-ter-Vehn problem is a three-dimensional, asymmetric, large-deformation adiabatic compression problem with no shocks. It has an existing analytical solution given in [47], which allows one to study the convergence properties of newly proposed numerical techniques in this fully nonlinear setting.

The problem setup considered here and taken from [109] is as follows. We begin with a spherical geometry of unit radius and apply a time-dependent pressure boundary condition on the outer surface of the sphere. The analytical solution may be expressed as follows. The position vector is given by

$$r = (1 - t)R, \tag{4.33}$$

$$z = (1 - t)^{1/4} Z, \tag{4.34}$$

which gives the velocity vector

$$v_r = -R = \frac{-(1 - t)R}{(1 - t)} = \frac{-r}{(1 - t)}, \tag{4.35}$$

$$v_z = \frac{-(1 - t)^{-3/4} Z}{4} = \frac{-(1 - t)^{1/4} Z}{4(1 - t)} = \frac{-z}{4(1 - t)}. \tag{4.36}$$

The density and internal energy are given by

$$\rho = (1 - t)^{-9/4}, \tag{4.37}$$

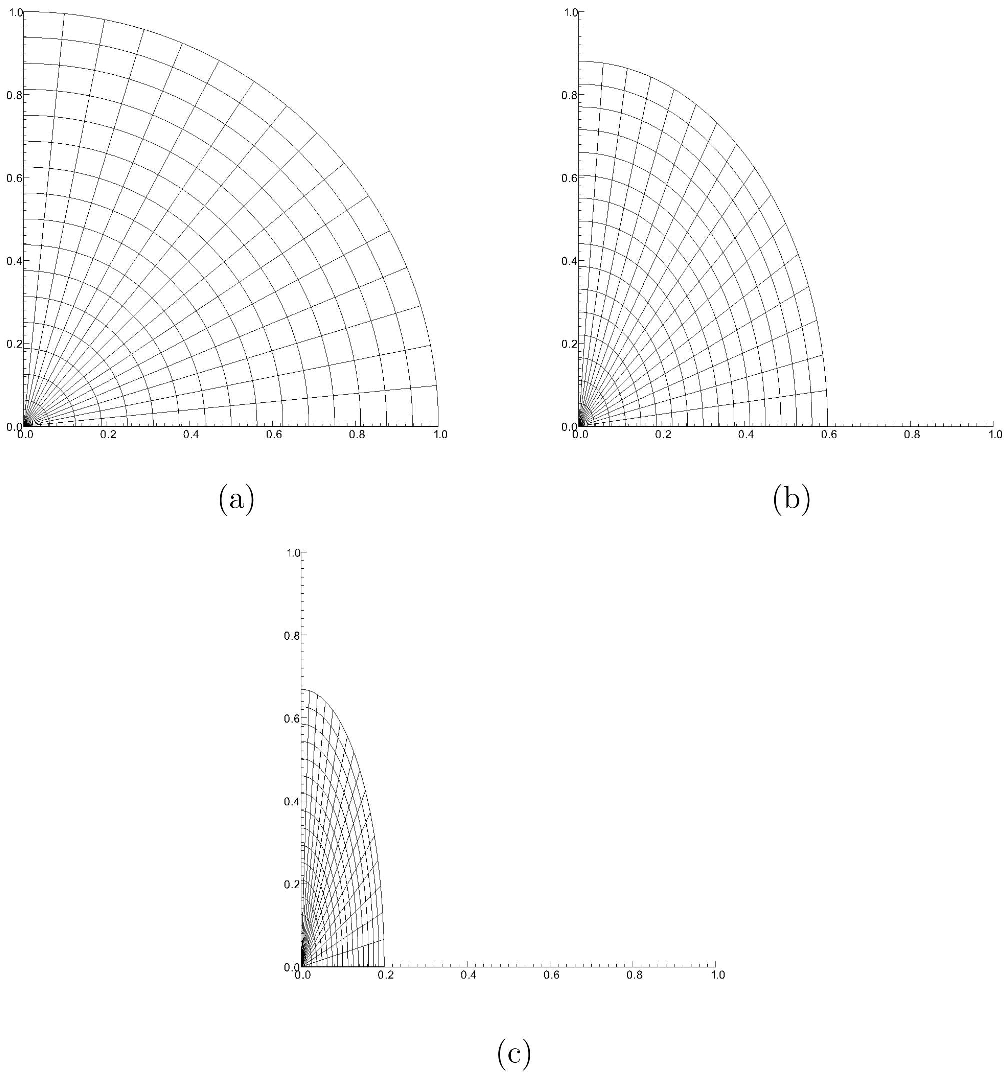
$$e = \left(\frac{3z}{8(1 - t)} \right)^2, \tag{4.38}$$

and the pressure is computed from the ideal gas equation of state. The adiabatic index is set to $\gamma = \frac{5}{3}$.

The kinematics in the Coggeshall–Meyer-ter-Vehn problem are quite simple. At each time t the reference radial position is globally scaled by $(1-t)$ and the axial position by $(1-t)^{1/4}$, which amounts to deforming the initially spherical shape into an ellipse (see Eqs. (4.33) and (4.34)). Because NURBS can represent the reference spherical configuration exactly, and because they are affine-covariant, this motion may be reproduced exactly in the discrete space. It also follows from Eqs. (4.35) and (4.36) that the divergence of the velocity field is a time-dependent global constant, which, together with the spatially-constant density (see Eq. (4.37)), simplifies the internal energy equation to

$$\frac{\mathrm{d}e}{\mathrm{d}t} = c(t)e, \tag{4.39}$$

where $c(t)$ is a time-dependent function. The form of Eq. (4.39) implies that the internal energy may be expressed as a product of the initial spatial profile and a time-dependent function. Provided the initial conditions, the time integration scheme, the numerical quadrature, and the linear equation solver are exact, the



(a)

(b)

(c)

Figure 4.1: Coggeshall–Meyer-ter-Vehn problem computed on the 16×16-element mesh. (a) $t = 0$; (b) $t = 0.4$; (c) $t = 0.8$.

solution of the Coggeshall–Meyer-ter-Vehn problem may be exactly represented using a quadratic NURBS mesh.

We used the 16×16 mesh of NURBS elements to discretize the meridian surface of the problem domain, which is one quarter of a circular disc of unit radius (see Figure 4.1a). The NURBS mesh is generated by creating a 90° circular arc of unit radius using three control points, and coalescing the other three control points at the origin. This gives an exact representation of the computational domain

Table 4.1: Max norm of the error.

	ρ	p	v_r	v_z
3GP, Δt	7.205E-004	8.801E-005	1.676E-006	6.187E-005
4GP, $\Delta t/2$	8.630E-007	2.697E-007	3.616E-009	5.003E-008
5GP, $\Delta t/4$	5.447E-008	3.520E-008	9.118E-010	3.548E-010

that is linear in the radial direction and quadratic in the azimuthal direction. The degree of the polynomial in the radial direction is raised to quadratic through order elevation [87, 48], and then knots are inserted [87, 48] in the parametric domain to achieve a 16×16-element quadratic NURBS discretization that is C^1-continuous at the element boundaries.

We compute the problem using three, four, and five quadrature points in each tensor-product direction, each time dividing the time step by a factor of two. This gives a sequence of solutions with improved accuracy of quadrature and time integration. Traction (i.e., normal pressure) boundary conditions are prescribed weakly, where, in the boundary integral terms, pressure is taken from the analytical solution. We do not use shock viscosity or other enhancements such as energy projection; a pure Galerkin formulation of the Euler equations of gas dynamics is employed in this problem. Table 4.1 shows the max norm of the solution errors. The max norm is obtained by comparing the magnitude of the error at all quadrature points in the mesh and taking the maximum value. The results indicate that the errors may be reduced to arbitrarily low levels given sufficient accuracy of time integration and quadrature. A sequence of mesh snapshots is shown in Figure 4.1, where a smooth mesh deformation into an elliptical shape is observed.

4.4.2 Sedov blast problem

We now compute the Sedov blast problem. We use a square domain with the edge length $L = 1.1$. In this problem, energy is released at the origin, which creates an expanding shock wave. The initial condition of the continuous problem consists of zero velocity everywhere, and the Dirac delta distribution of the internal

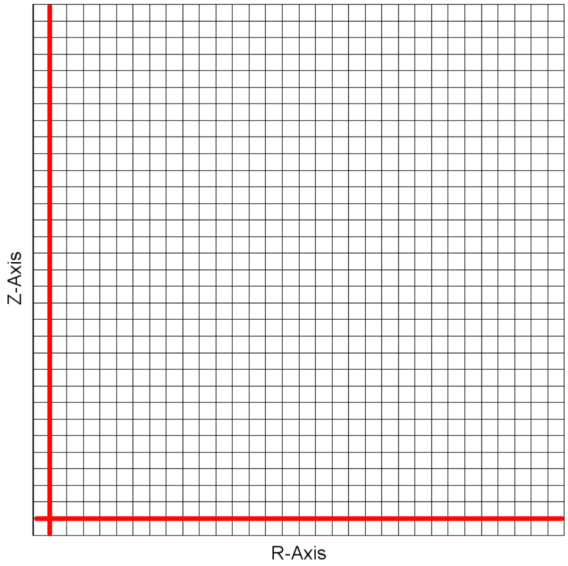


Figure 4.2: Sedov blast problem. NURBS mesh of 32 × 32 square elements. Continuity of the basis functions is reduced along the red lines to the C^0-level to isolate the discontinuity in the initial condition.

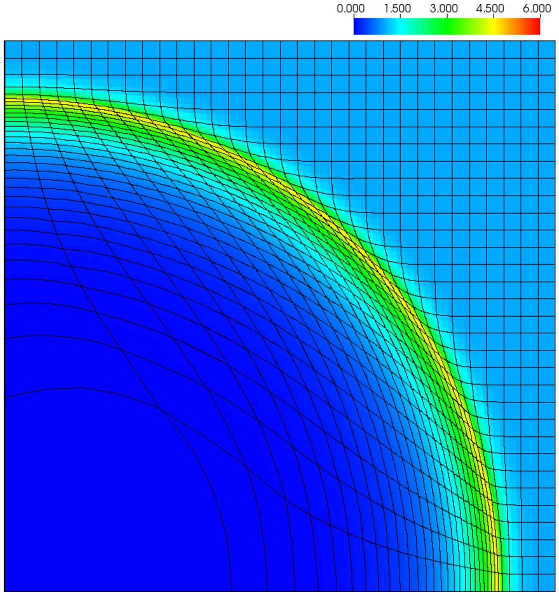


Figure 4.3: Sedov blast problem. Deformed mesh at $t = 1$ and density contours for a 32 × 32 mesh computation.

energy at the origin. The Dirac delta is approximated using a bilinear function, which attains its maximum at the origin, and vanishes on the boundaries of the corner element. The value of the bilinear function at the origin is set such that the total energy is $E_{tot} = 1$. This initial condition, which is assigned directly to the Gauss points, is consistent with that reported in [58, 14]. The adiabatic index and density are set to $\gamma = 1.4$ and $\rho_0 = 1$, respectively. We “isolate” the initial-condition singularity in the corner element using the lines of C^0-continuity, a technique that was found to produce very good results in [14]. The lines of C^0-continuity can be clearly seen in Figure 4.2, which shows one of the meshes employed in the calculations.

The spherical Sedov problem has an exact solution. In particular, at time $t = 1$, the shock is located at $r = 1$ with a peak density of 6. The density results on the 32×32 mesh are shown in Figure 4.3. The deformed mesh in the figure shows curved elements and appears smooth.

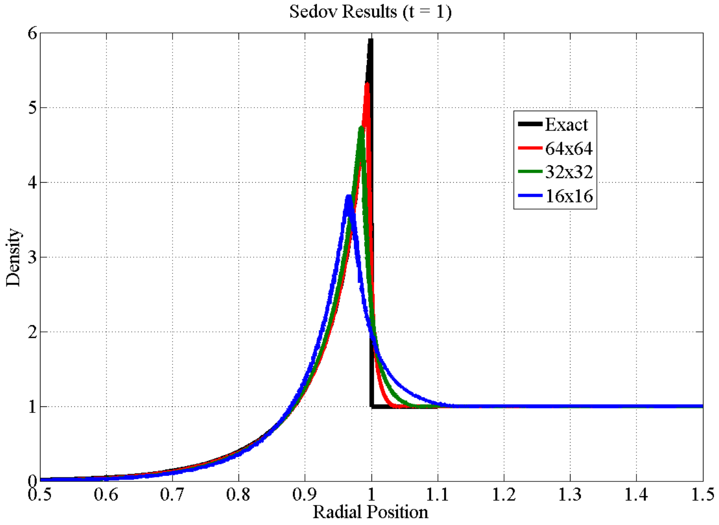


Figure 4.4: Sedov blast problem. Density vs. radius scatter plot at time $t = 1$. Convergence to the analytical solution occurs under mesh refinement.

The Sedov blast problem was also computed using 16×16 and 64×64-element meshes to qualitatively assess convergence of the proposed technique under mesh refinement. The results demonstrate good convergence to the analytical solution, as evidenced in the density scatter plot shown in Figure 4.4. Furthermore, no significant density scatter is present in the figure, which suggests good symmetry preservation characteristics of the proposed technique. (Note that because of the “square” meshes used in this example, exact preservation of spherical symmetry is not expected.)

4.4.3 Noh implosion problem

We next compute the Noh implosion problem [125]. The problem domain is a 90° slice of a circular disc with a unit radius. The initial condition consists of a velocity unit vector pointing in the direction of the origin, with internal energy set to zero, and $\rho_0 = 1.0$ throughout the domain. At the origin, the velocity is set to zero. The adiabatic index is set to $\gamma = \frac{5}{3}$. Symmetry boundary conditions are applied on the lateral boundaries of the problem domain, and a zero-traction boundary conditions are applied on the remaining parts of the boundary.

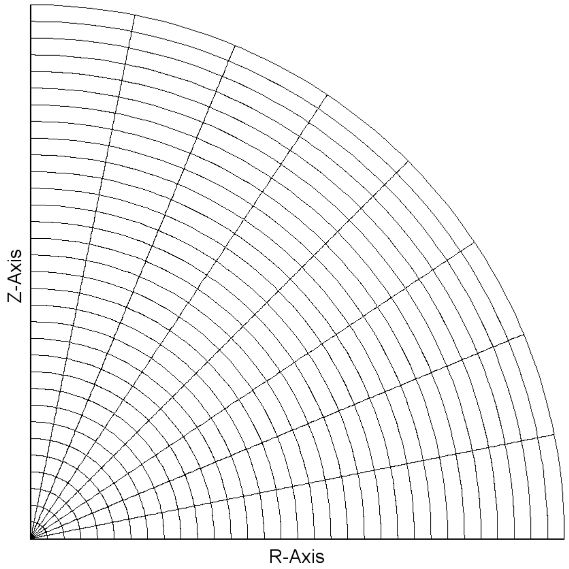
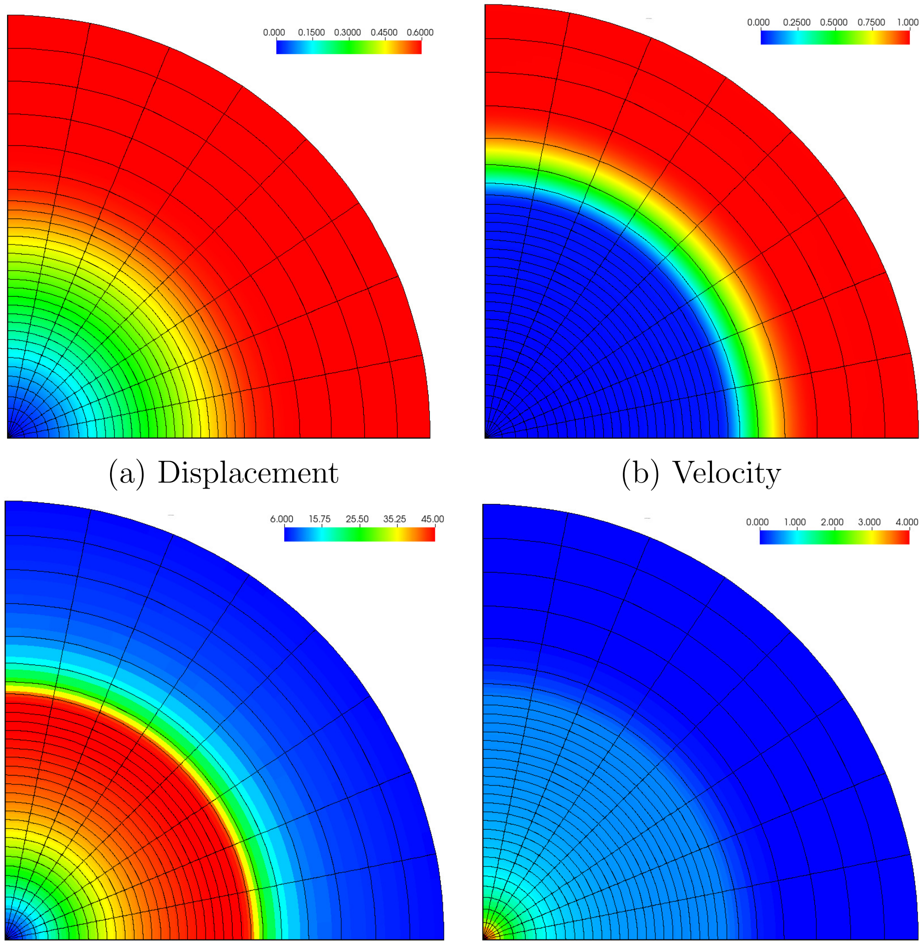


Figure 4.5: Noh implosion problem. The mesh of 8×32 elements employed in one of the computations.

The mesh for our baseline simulation is comprised of 32 evenly-spaced elements in the radial direction, and eight evenly-spaced elements in the azimuthal



(a) Displacement

(b) Velocity

(c) Density

(d) Energy

Figure 4.6: Noh implosion problem. Snapshots of the solutions at $t = 0.6$. The solutions appear to be radially symmetric.

direction (see Figure 4.5). At the origin, all the control points are coalesced at the same spatial location. The weights of the control points at the origin are the same as the weights of the control points on all other azimuthal mesh lines. The mesh exactly represents the geometry of the meridian surface. Radially-symmetric initial conditions on the velocity are imposed using a procedure described in [14]. *This procedure, which is unique to the NURBS discretization, produces a radially-symmetric initial condition at every point in the domain.* The Noh problem is computed until $t = 0.6$. The contours of displacement, velocity, density, and energy at this time instant are shown in Figure 4.6. The solution is free of oscillations and appears to be symmetric. (We will make a quantitative assessment of the solution symmetry in what follows.) Figure 4.7 shows the scatter plot of the density

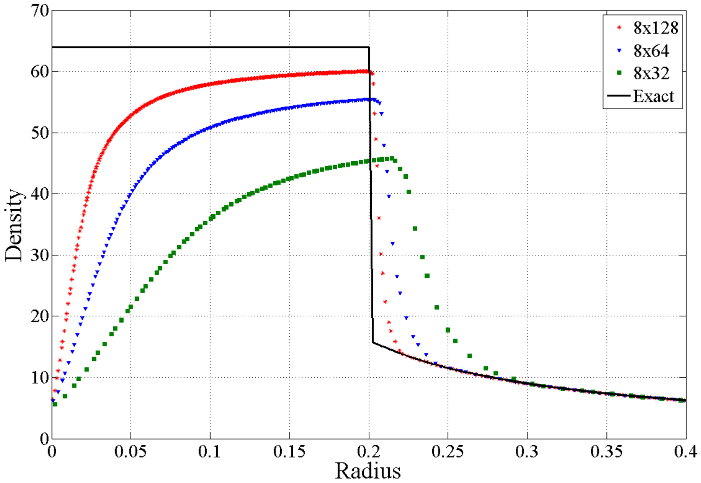


Figure 4.7: Noh implosion problem. Scatter plot of density vs. radial coordinate for several mesh refinements in the radial direction at $t = 0.6$. Convergence to the analytical solution is observed.

vs. radial coordinate. Two mesh refinements in the radial direction are performed. The analytical solution is also plotted for comparison. At $t = 0.6$, the shock location is expected at $r = 0.2$, with a peak post-shock density of $\rho = 64$. The shock location is predicted fairly well. However, on the coarse mesh, the density amplitude in the post-shock region is somewhat lower than the analytical solution. As the mesh is refined, the numerical results approach the analytical solution.

While the density variation in the 8×32 case is small ($\approx 0.001\%$ variation), this error can be further reduced by increasing the order of the Gaussian quadrature rule (see Figure 4.8). The 8×32 case was re-computed using four- and five-point rules, which reduced the density variation over two orders of magnitude for each extra Gauss point used. The density variation is measured as the standard deviation of the solution on the radial layer of Gauss points closest to the origin,

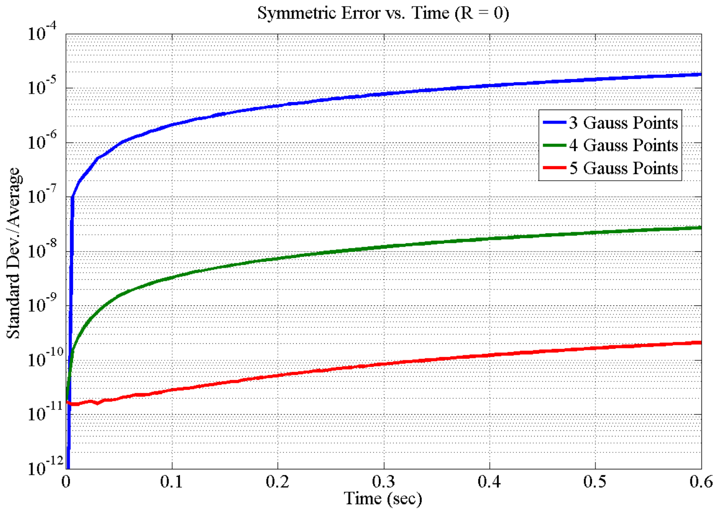


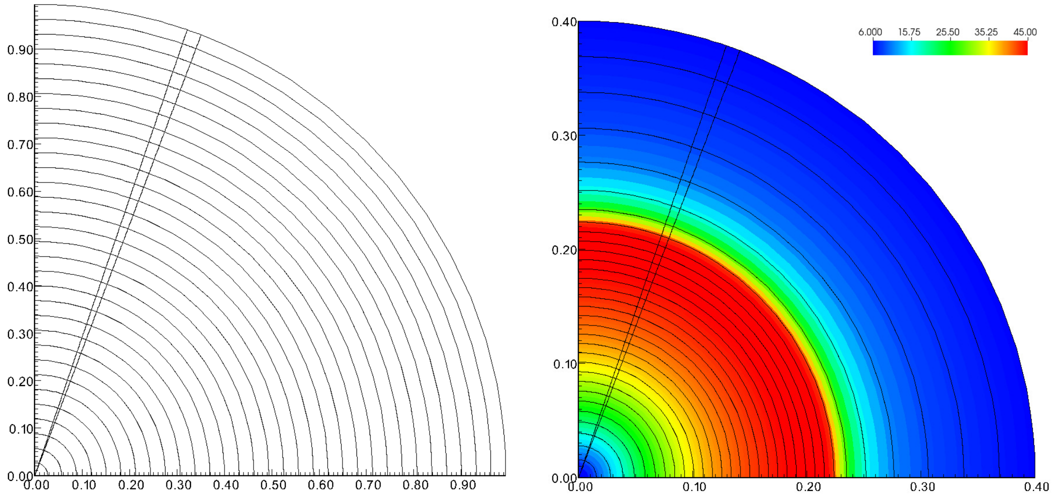
Figure 4.8: Noh implosion problem. Time history of symmetry error in the density for the 8×32-element mesh, measured as the density standard deviation on the radial layer of Gauss points closest to the origin. Increasing the number of Gauss points by one in each direction reduces the symmetry error by two orders of magnitude.

which is then divided by the mean density $\bar{\rho}$ for the same set of points:

$$s = \left(\frac{1}{\bar{\rho}^2 (n-1)} \sum_{i=1}^{n} (\rho_i - \bar{\rho})^2 \right)^{\frac{1}{2}} . \tag{4.40}$$

The Gauss points in the layer are initially equidistant from the origin, and thus the proposed quantity present a good measure of the symmetry error in the density. Furthermore, the origin, due to the degenerated geometric mapping and the factor r present in the weak-form integrals, is often considered to be the most “vulnerable” location in the mesh. It is clear from the figure that rapid convergence to a perfectly symmetric result occurs with the increasing order of Gaussian quadrature. Numerical quadrature is the sole source of the (very small!) symmetry error.

To verify that the solution is independent of the azimuthal discretization, we solved the Noh implosion problem using a 3×32-element, highly stretched and asymmetric mesh shown in Figure 4.9a. The density contours for this computation are shown in Figure 4.9b. The density symmetry error for this case is shown in



(a) (b)

Figure 4.9: Noh implosion problem. (a) Mesh of the initial configuration; (b) Density contours at $t = 0.6$ on the deformed mesh.

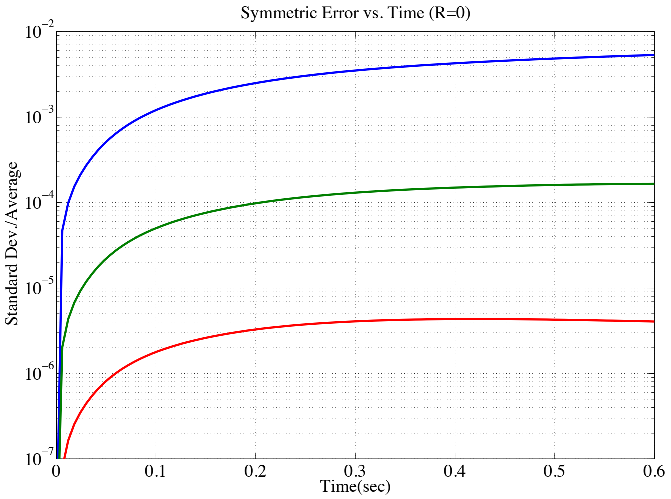


Figure 4.10: Noh implosion problem. Time history of the symmetry error in the density for the 3×32-element highly stretched and asymmetric mesh. Increasing the number of Gauss points by one in each direction reduces the symmetry error by two orders of magnitude.

Figure 4.10. The solution goes to a perfectly symmetric result with increased order

of Gaussian quadrature. The results are indistinguishable from those obtained using the 8×32 mesh.

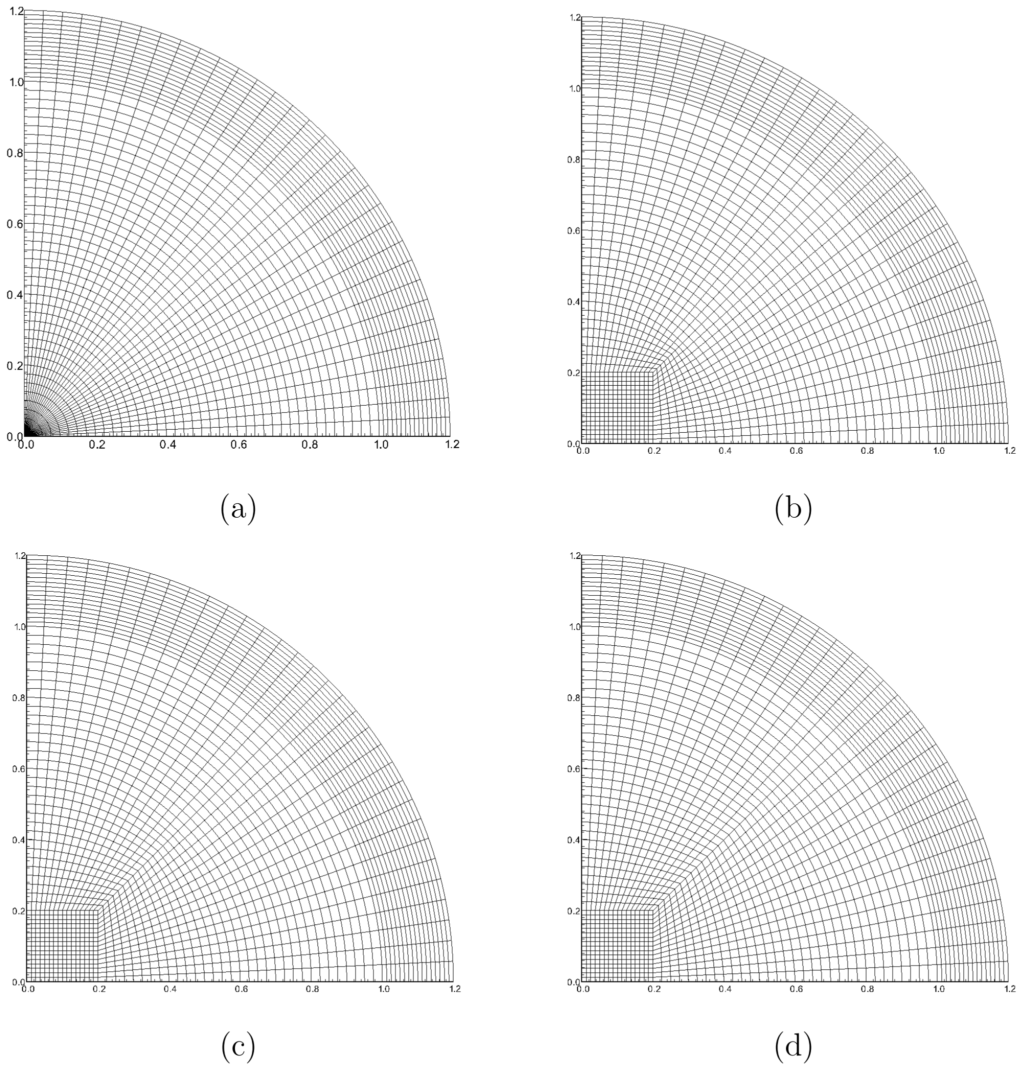
4.4.4 Multi-material implosion problem

This problem was originally posed in a 2D cylindrical domain by [70] to test symmetric errors in shock wave calculations in multi-material media. We consider a spherically symmetric extension of this case, as presented in [59]. The problem consists of a sphere of low-density gas ($\rho = 0.05$) of radius 1. A shell of higher density gas ($\rho = 1$) surrounds the sphere, and has the initial thickness of 0.2. The entire domain has the initial pressure of $p = 0.1$, and both gasses have the same adiabatic constant $\gamma = \frac{5}{3}$. The outer surface of the shell has a prescribed velocity of $v_r = -5$.

A spherical shock wave originates from the outer surface and moves toward the origin. The shock crosses the interface numerous times during the simulation. The density jump at the interface gives an initial Atwood number of $At \approx 0.905$. In this range of Atwood numbers, symmetry errors tend to grow quickly due to Rayleigh–Taylor and Richtmyer–Meshkov-type instabilities.

We compute the problem using quadratic-order NURBS. The computations are performed on the polar and the "butterfly"-type meshes. In the latter, the problem domain is created using three distinct regions: The square inner region near the origin, the axisymmetric outer region, and the transition region in-between. The three regions correspond to the NURBS patches, which are joined with C^0-continuity. The square region near the origin has a fixed size of 0.2×0.2, while the location of the outer boundary of the transition region is varied in the simulations. Three locations are considered corresponding to $R = 0.4$, $R = 0.6$, and $R = 0.8$. The closer the location of the transition-region outer boundary to the origin, the larger the fraction of the domain that can support an exact axisymmetric solution. The initial-configuration meshes are shown in Figure 4.11.

An azimuthally-uniform mesh is comprised of 32×56 elements. The mesh uses 40 evenly spaced elements in the radial direction to discretize the inner gas region, and 16 evenly spaced elements for the outer region. The two material



(a)

(b)

(c)

(d)

Figure 4.11: Multi-material implosion problem. Initial mesh using (a) Polar mesh; and Butterfly meshes with C^0 lines at $r =$ (b) 0.4, (c) 0.6, and (d) 0.8.

regions are separated by a line of C^0-continuity. The butterfly meshes are designed such that the element sizes in each region are similar to the polar mesh case.

Figure 4.12 shows the density contours superposed on the deformed mesh at $t = 0.15$. There are no visible symmetry errors in the figure.

Figure 4.13 shows the time history of the symmetry error in the density at the interface between the two gases. To compute this quantity, we use Eq. (4.40) applied to the layer of quadrature points closest to the interface on the side of the

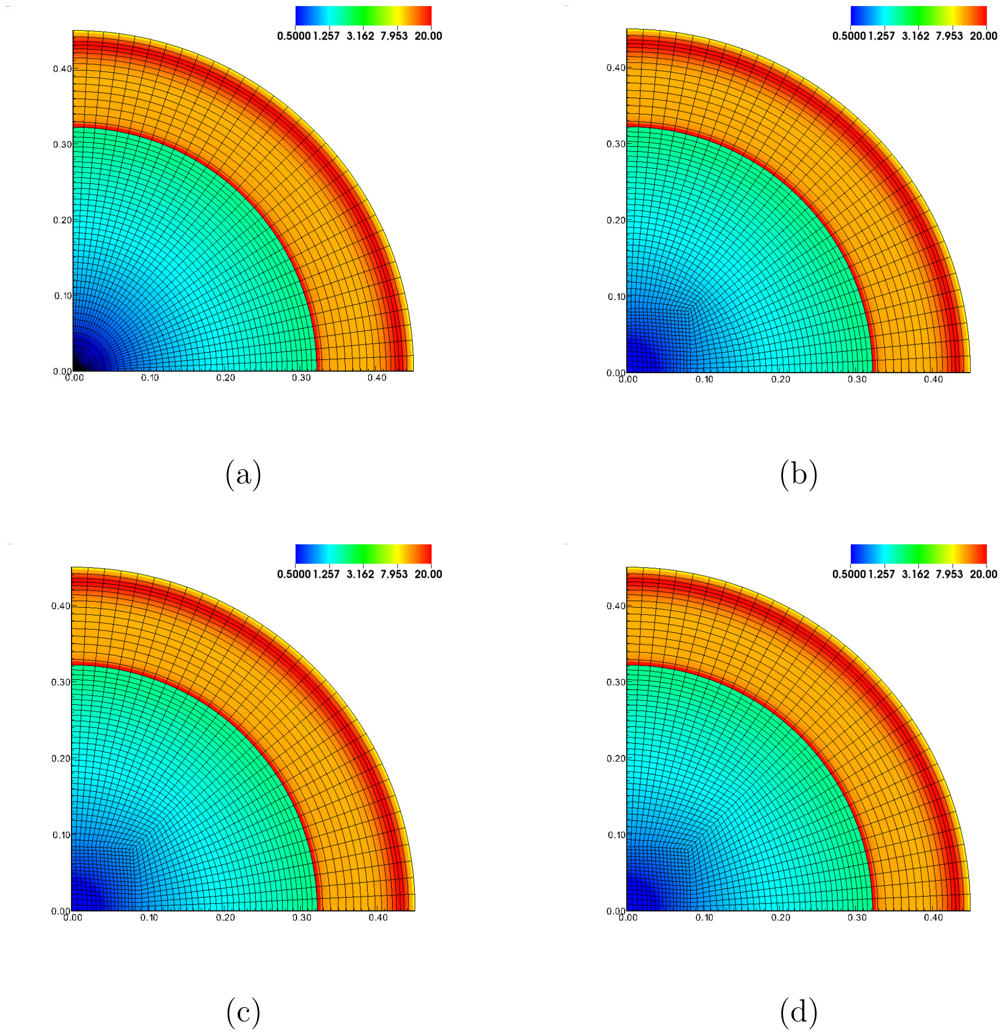


Figure 4.12: Mesh and density at time $t = 0.15$ for (a) Polar mesh, and for butterfly meshes with initial C^0 lines at $r =$ (b) 0.4, (c) 0.6, and (d) 0.8.

lower density gas. In the figure, for a reference value, we provide the symmetry error results at the final configuration for the Q2Q1 computation taken from [59], where a similar butterfly mesh configuration was employed. For the butterfly meshes, the final-configuration symmetry error is independent of the location of the outer boundary of the transition zone, and is on the order of 10^{-4}, which is somewhat lower, but still comparable to the results in [59]. However, the point at which rapid transition to a non-symmetric solution occurs correlates with the

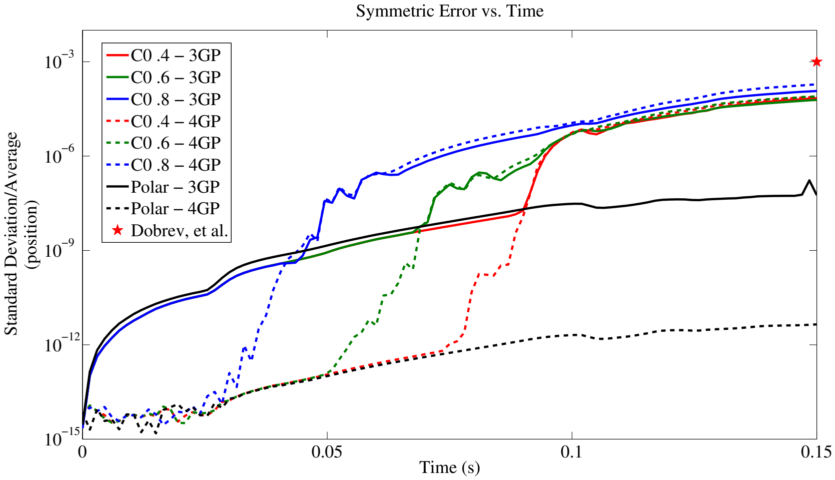


Figure 4.13: Multi-material implosion problem. Time history of the symmetry error in the radial position for the three- and four-Gauss-point computations. The star at time $t = 0.15$ represents the final-time positional symmetry error from [59].

location of the outer boundary of the transition region; moving the outer boundary closer to the origin delays the rapid symmetry error growth at the material interface. The polar mesh does not produce such symmetry errors. The only source of the symmetry error in the polar case is the error due to quadrature. Once the symmetry is broken on a butterfly mesh, quadrature errors play a minor role when it comes to symmetry preservation.

4.5 Conclusions

IGA is applied to solve axisymmetric gas dynamics problems in 3D using rz-cylindrical Lagrangian coordinates. The IGA rz formulation inherits the high-order accuracy and spherical symmetry preservation (under exact integration) properties of the originally proposed IGA formulation for the Cartesian case. Well known axisymmetric benchmark problems were computed in support of these results.

4.6 Acknowledgements

This work was performed under the auspices of the National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396. M.J. Shashkov acknowledges funding by the DOE Office of Science Advanced Scientific Computing Research (ASCR) Program at the Los Alamos National Laboratory (LANL). M.J. Shashkov, D.J. Benson, Y. Bazilevs, and C.C. Long also acknowledge the funding from the Advanced Simulation and Computing (ASC) Program at LANL. This work was partially supported by the UCSD-Los Alamos Educational Collaboration. We also wish to thank Drs. Tz.V. Kolev, V.A. Dobrev, and R.N. Rieben at the Lawrence Livermore National Laboratory for continued fruitful discussions on higher-order methods for Lagrangian hydrodynamics, and for encouraging us to look at the *rz* formulation using IGA.

Chapter 4, in full, is a reprint of the material as it was submitted in the article “Isogeometric Analysis of Lagrangian Hydrodynamics: Axisymmetric Formulation in the *rz*-Cylindrical Coordinates”, submitted to **Journal of Computational Physics**. Authors include: Y. Bazilevs, C.C. Long, I. Akkerman, D.J. Benson, and M.J. Shaskov. The dissertation author was the primary investigator and author of this paper.

Chapter 5

Pediatric Ventricular Assist Devices

*As appears in **Computational Mechanics**, DOI:10.1007/s00466-013-0858-3*

In this paper we present a collection of fluid–structure interaction (FSI) computational techniques that enable realistic simulation of pulsatile Ventricular Assist Devices (VADs). The simulations involve dynamic interaction of air, blood, and a thin membrane separating the two fluids. The computational challenges addressed in this work include large, buckling motions of the membrane, the need for periodic remeshing of the fluid mechanics domains, and the necessity to employ tightly coupled FSI solution strategies due to the very strong added mass effect present in the problem. FSI simulation of a pulsatile VAD at realistic operating conditions is presented for the first time. The FSI methods prove to be robust, and may be employed in the assessment of current, and the development of future, pulsatile VAD designs.

5.1 Introduction

Ventricular Assist Devices (VADs) are devices which provide mechanical circulatory support to a single ventricle of the heart [71, 7]. They are used primarily as a bridge to transplant, extending the life of the patient until a compatible donor can be found. Two device types are available: pulsatile displacement pump designs, and a continuous flow impeller designs. Devices now available to the pediatric community include only pulsatile designs, and the we therefore choose to focus on pulsatile VADs in this work. The pediatric population suffers from increased risk of thromboembolic events (i.e., blood clots) while using VADs, and thrombo-embolic events may occur in up to 40% [9] of cases. This has made these devices too risky for long-term use, and reliable only as a short-term bridge to transplant. However, in children, particularly those with congenital heart defects or cardiomyopathy, it may be of particular importance to develop a long-term reliable device. Pediatric patients in heart failure due to dilated cardiomyopathies have shown recovery of the native heart tissue when a VAD is used in a bridge to recovery scenario. This is an effect that has not been observed in the adult population [41].

A pulsatile VAD provides mechanical support to a single ventricle of the heart. A patient may receive two VADs if support is required for both ventricles, depending on the underlying disease state. The design of the device is as follows: Two domed chambers are separated by a flexible polyurethane membrane. One chamber is an air compartment, which is driven pneumatically. The other is a blood chamber, which delivers blood from the right atrium/left ventricle to the pulmonary arteries/aorta, for a Right/Left heart VAD, respectively [7]. The flow in the air chamber moves the thin membrane, which causes displacement in the blood chamber and drives the blood through the device.

The low survival rates of VADs may be alleviated if one had a better understanding of how specific flow features in the blood chamber are linked to the formation of blood clots. For this, as a first step, one needs the ability to accurately predict the blood flow itself inside the device. The latter is not possible without fluid–structure interaction (FSI) modeling that involves the interaction of air, blood, and a thin membrane separating the two fluids. For the modeling to be realistic, the relatively complex geometry of the VAD, the large, time-dependent motions of the membrane, and the actual pump operating conditions (i.e., flow rates and pressures) must be taken into account. In this work we present computational FSI methods in which these effects are incorporated, and which may be used for high-fidelity VAD design. While there is extensive literature on patient specific modeling for pediatric applications with and without FSI [106, 65, 111], very little has been done to date for numerical simulation of pulsatile VADs. In [62] an idealized VAD design was analyzed with a linearly elastic shell model for the membrane and non-physiological outlet boundary conditions, and in [100] VAD simulations were performed using imposed membrane motion rather than true FSI. We stress that membrane motion in a VAD is expected to be critical to the device's performance, and thus accurate prediction of the time-dependent membrane response is crucial to the simulation and design of the device. To the best of the authors' knowledge this is the first 3D, full-scale, high-fidelity FSI modeling of pulsatile VADs. The paper is outlined as follows. In Section 5.2, we present the computational methods employed to simulate the VAD FSI problem. A rotation-free isogeometric Kirchhoff–Love shell formulation is used to model the thin membrane in combination with a moving-domain ALE-VMS finite element formulation for the blood and air flow. The FSI solution strategy involves strong coupling in the left-hand-side matrix, which is accomplished using a combination of sparse-matrix-based and matrix-free techniques. Strong coupling in the left-hand-side matrix is essential for convergence of the coupled FSI equation system for this application. In Section 5.3 we provide a detailed description of the VAD problem setup and present a numerical simulation of the device. In the course of the simulation we periodically remesh the fluid mechanics domain to maintain good quality of the finite element discretization. The simulation predicts physiologically realistic blood flow features and membrane deformation patterns. In Section 5.4 we draw conclusions and present future research directions.

5.2 Numerical Methods for VAD FSI

In this section we briefly discuss the fluid and structural mechanics formulations used in this work, namely ALE-VMS and Isogeometric Analysis (IGA), respectively. We mostly summarize the main features of these methods and provide references where the reader may find the mathematical details of these techniques. One of the main computational challenges of this work is robust FSI coupling, which we present in some detail in this section

5.2.1 ALE-VMS Fluid Mechanics Formulation

Standard Galerkin methods are not a sufficiently robust technology for advection-dominated flows. For this reason, stabilized methods [37, 85, 169, 82, 168, 161, 89, 170, 79] were designed to circumvent this shortcoming of the Galerkin technique. Stabilized methods, which are essentially residual-based modifications of the Galerkin method, exhibit uniform stability and convergence behavior across the full range of advective and diffusive phenomena.

The basic theory of variational multiscale (VMS) methods was developed in [80], wherein stabilized methods were first identified as multiscale methods. Relationship between stabilized methods and subgrid scale modeling was also identified in [80], and now presents an important research direction [84]. Recently, in [15], the authors proposed a residual-based turbulence modeling and computational framework that is based on the VMS theory, named RBVMS. This technique performs well on both laminar and turbulent flows, for a wide range of Reynolds numbers.

The extension of the RBVMS framework to the moving-domain case, where the motion of the fluid mechanics domain is handled using the Arbitrary Lagrangian–Eulerian (ALE) formulation [83], was named ALE-VMS in [153, 24]. The ALE-VMS formulation discretized with linear tetrahedral FEM is used in this work to compute the fluid mechanics part of the VAD problem.

An important additional feature of the ALE-VMS methodology is weak enforcement of essential boundary conditions. Weakly enforced essential boundary conditions were introduced in [25] in order to improve solution accuracy on meshes with insufficient boundary-layer resolution [27, 26, 13, 77]. Although the weak BCs are now routinely used for wind turbine aerodynamics [21, 76, 78] and ship hydrodynamics [5, 4, 6], we do not use them in this work. However, we feel that they will likely be beneficial in cardiovascular blood flow and FSI computations in that they may further improve boundary-layer accuracy and produce more accurate wall quantities such as wall shear stress or oscillating shear index, which are critically important in numerous cardiovascular applications [140, 146].

5.2.2 Rotation-Free Isogeometric Thin Shell Formulation

The circular membrane separating the blood and air chambers of the device is a very thin structure. The membrane stress-free reference configuration is not flat, but convex. As the membrane undergoes large cyclic deformation, it is almost always in a state of compression, which leads to local buckling and wrinkling. As a result, it is desirable to represent the membrane or thin shell with numerical technology that is efficient and capable of representing the underlying complex structural dynamics without posing significant challenges associated with robustness of the structural mechanics computations and large local deformations of the fluid mechanics domain boundary.

Low-order, bi-linear quadrilateral finite elements, which are widely used and are considered standard shell element technology, exhibit several shortcomings: 1) These elements require the use of displacement and rotation degrees of freedom to describe shell kinematics; 2) One needs a fine mesh to represent shell geometries with high local curvature, and to simultaneously achieve the desired solution accuracy; 3) Ad-hoc element technology is necessary to overcome membrane and shear locking; 4) In the case of implicit time integration employed in this work, the presence of rotational degrees of freedom doubles the size of the solution and right-hand-side residual arrays, quadruples the size of the left-hand-side matrix, and results in an order-of-magnitude increase in linear solver time.

Isogeometric shell analysis was recently proposed in [31] to address the shortcomings of standard shell technology listed above. It was found that higher-order continuity (C^1 and above) of the IGA basis functions significantly improved the per-degree-of-freedom accuracy and robustness of thin shell discretizations as compared to the FEM. Furthermore, the increased continuity of the IGA discretizations enabled the use of shell kinematics without rotational degrees of freedom [99, 30, 32], leading to further computational cost savings. The isogeometric rotation-free Kirchhoff–Love shell formulation for structures composed of multiple structural patches, called the bending strip method, was developed in [98], which enabled the application of the rotation-free IGA technology to real-life structures, such as wind turbine rotors (see [23, 20, 78]). Besides significant savings in computational time, the rotation-free shell discretization makes FSI coupling simpler than the discretization with rotational degrees-of-freedom. Finally, the smooth structural motion computed with IGA gives a smooth fluid mechanics mesh at the fluid–structure boundary, which adds accuracy and robustness to the fluid mechanics computation.

Non-Uniform Rational B-Splines (NURBS) [**?**] are employed in this work to discretize the structural mechanics equations of the membrane separating the blood and air chambers. T-splines [17, 53], a relative newcomer to IGA currently receiving significant attention, are also well suited for the proposed structural modeling approach. For related rotation-free shell formulations the reader is also referred to [44, 43, 45, 129, 128, 124].

5.2.3 FSI Coupling

In order to take advantage of the benefits of IGA for structural mechanics, and to leverage the existing advanced automatic mesh generation tools for the FEM, we choose to couple low-order FEM for the fluid and IGA for structural mechanics. As a result, the FSI coupling assumes a nonmatching fluid–structure interface discretization. Nonmatching interface discretizations in FSI problems necessitate the use of interpolation or projection of kinematic and traction data between the nonmatching surface meshes (see, e.g., [66, 156, 21, 158, 28], where [158] is more comprehensive than [156]). A computational procedure, which can simultaneously handle the data transfer for IGA and FEM discretizations, was proposed in [21]. The procedure also includes a robust approach in identifying "closest points" for arbitrary shaped surfaces. While such interface projections are rather straightforward for loosely-coupled FSI algorithms, they require special techniques (such as developed in [170, 157, 158, 28] as well as this paper) for strongly-coupled methods that are monolithic-like and that are necessary for the present application.

A full discretization of the FSI formulation leads to coupled, nonlinear equation systems that need to be solved at every time step.The equation systems can be written as follows:

$$\mathbf{N}_1 \left(\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3\right) = \mathbf{0}, \tag{5.1}$$

$$\mathbf{N}_2 \left(\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3\right) = \mathbf{0}, \tag{5.2}$$

$$\mathbf{N}_3 \left(\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3\right) = \mathbf{0}. \tag{5.3}$$

Here $\mathbf{N}_1$, $\mathbf{N}_2$, and $\mathbf{N}_3$ are the discrete residual functions, and $\mathbf{d}_1$, $\mathbf{d}_2$, and $\mathbf{d}_3$ are the vectors of nodal (or control-point in the case of IGA) unknowns, corresponding to the fluid mechanics, structural mechanics, and mesh problems,

In the block-iterative coupling [162, 163, 164, 165, 171, 172, 173, 166, 170, 28], the fluid, structure, and mesh systems are treated as separate blocks, and the nonlinear iterations are carried out sequentially. First, the fluid block is solved, then the structure, and then the mesh. In solving a given block of equations the most current values of the other blocks of unknowns are used. The sequence of solves is repeated until all the equation systems are solved to an a priori set tolerance. This strategy is the easiest to implement, and it performs very well in applications where the structure is heavy relative to the surrounding fluid.

In the present application, the membrane separating the blood and air chambers of the VAD is extremely thin, and its mass is significantly smaller than the mass of the surrounding fluid that is displaced as a result if the membrane motion. Because of the relatively low structural mass, block-iterative FSI is not an appropriate technique for this application. Instead, we employ the quasi-direct coupling technique [171, 172, 173, 170, 28], where the fluid+structure and mesh systems are treated as two separate blocks, and the nonlinear iterations are carried out one block at a time until all the equation systems are solved to an a priori set tolerance. In an iteration step, given the solution at i, the solution $i+1$ is obtained by solving the following two blocks of equations:

$$\left.\frac{\partial \mathbf{N}_1}{\partial \mathbf{d}_1}\right|_i \mathit{\Delta}\mathbf{d}_1^i + \left.\frac{\partial \mathbf{N}_1}{\partial \mathbf{d}_2}\right|_i \mathit{\Delta}\mathbf{d}_2^i = -\mathbf{N}_1\left(\mathbf{d}_1^i,\ \mathbf{d}_2^i,\ \mathbf{d}_3^i\right), \tag{5.4}$$

$$\left.\frac{\partial \mathbf{N}_2}{\partial \mathbf{d}_1}\right|_i \mathit{\Delta}\mathbf{d}_1^i + \left.\frac{\partial \mathbf{N}_2}{\partial \mathbf{d}_2}\right|_i \mathit{\Delta}\mathbf{d}_2^i = -\mathbf{N}_2\left(\mathbf{d}_1^i,\ \mathbf{d}_2^i,\ \mathbf{d}_3^i\right), \tag{5.5}$$

$$\mathbf{d}_1^{i+1} = \mathbf{d}_1^i + \mathit{\Delta}\mathbf{d}_1^i, \tag{5.6}$$

$$\mathbf{d}_2^{i+1} = \mathbf{d}_2^i + \mathit{\Delta}\mathbf{d}_2^i, \tag{5.7}$$

$$\left.\frac{\partial \mathbf{N}_3}{\partial \mathbf{d}_3}\right|_i \mathit{\Delta}\mathbf{d}_3^i = -\mathbf{N}_3\left(\mathbf{d}_1^{i+1},\ \mathbf{d}_2^{i+1},\ \mathbf{d}_3^i\right), \tag{5.8}$$

$$\mathbf{d}_3^{i+1} = \mathbf{d}_3^i + \mathit{\Delta}\mathbf{d}_3^i. \tag{5.9}$$

The above systems of linear equations are solved using a GMRES technique [138], requiring the computation of matrix-vector products. In this work the matrix-vector products involving $\left.\frac{\partial \mathbf{N}_1}{\partial \mathbf{d}_1}\right|_i$, $\left.\frac{\partial \mathbf{N}_2}{\partial \mathbf{d}_2}\right|_i$, and $\left.\frac{\partial \mathbf{N}_3}{\partial \mathbf{d}_3}\right|_i$ are computed using a sparse-matrix-based approach, where the tangent matrices are derived analytically and assembled into a sparse-matrix data structure in a standard fashion. The remaining matrix-vector products involving $\left.\frac{\partial \mathbf{N}_1}{\partial \mathbf{d}_2}\right|_i$ and $\left.\frac{\partial \mathbf{N}_2}{\partial \mathbf{d}_1}\right|_i$ are approximated using a matrix-free approach, namely

$$\left.\frac{\partial \mathbf{N}_1}{\partial \mathbf{d}_2}\right|_i \mathit{\Delta}\mathbf{d}_2^i = \frac{\mathbf{N}_1\left(\mathbf{d}_1^i,\ \mathbf{d}_2^i + \epsilon_1 \mathit{\Delta}\mathbf{d}_2^i,\ \mathbf{d}_3^i\right) - \mathbf{N}_1\left(\mathbf{d}_1^i,\ \mathbf{d}_2^i,\ \mathbf{d}_3^i\right)}{\epsilon_1}, \tag{5.10}$$

$$\left.\frac{\partial \mathbf{N}_2}{\partial \mathbf{d}_1}\right|_i \mathit{\Delta}\mathbf{d}_1^i = \frac{\mathbf{N}_1\left(\mathbf{d}_1^i + \epsilon_2 \mathit{\Delta}\mathbf{d}_1^i,\ \mathbf{d}_2^i,\ \mathbf{d}_3^i\right) - \mathbf{N}_1\left(\mathbf{d}_1^i,\ \mathbf{d}_2^i,\ \mathbf{d}_3^i\right)}{\epsilon_2}, \tag{5.11}$$

where ϵ_1 and ϵ_2 are relatively small real numbers. We feel this approach is well suited for cases that require a relatively larger number of GMRES iterations ($O(100)$) for good overall nonlinear convergence. This is because the sparse matrices are formed once every nonlinear iteration (or, possibly, once every time step to further decrease the computational cost), making the associated computational cost independent of the number of GMRES iterations employed. Although matrix-vector products given by Eqs. (5.10) and (5.11) need to be assembled once per GMRES iteration, the FEM assembly takes place over a narrow band of fluid elements near the fluid–structure interface, which is a lot less expensive than assembling the discrete residuals over the entire fluid mechanics domain. For a comprehensive exposition of matrix-based and matrix-free approaches see [28].

5.3 VAD Simulation

5.3.1 Problem Setup

Geometry For the initial study, we use a generic VAD device as our computational domain. Geometric properties, such as width, height, and angles of the entrance/exit arms are consistent with current designs, and are meant to be a generic representation of current commercially available devices. The chosen design for the initial study has a width of 7.7 cm, and an apex to apex height of 4.5 cm. The incline angle between the arms and the main blood chamber is 30°, with one assigned exclusively as the inlet, and the other as an outlet. The outlet faces are 1.5 cm in diameter. The air chamber has one small inlet/outlet port of diameter 0.8 cm. These are labeled in Figure 5.1. A stroke volume of 73 mL was chosen for this device, which yields an ejection fraction of 68%. A beat frequency of 80 bpm is used, for a pump output of 5.8 L/min. This initial study uses a VAD that is too large to be considered a pediatric model, but all data is within an acceptable physiologic range for adult models.

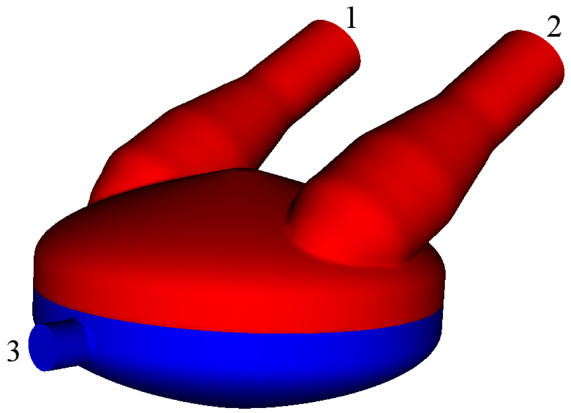


Figure 5.1: The computational domain, with the blood domain in red, and the air domain in blue. The inlet and outlet face of the blood chamber are labeled 1 and 2, respectively. The air-side inlet/outlet face is labeled 3.

Boundary Conditions Each pump cycle may be broken up into two components: the fill stage and the ejection stage. We impose the fill period of 0.45 s, and the ejection period of 0.3 s, and we also enforce that each stage must fill or eject the same volume, 73 mL. For simplicity, the flow is assumed to behave sinusoidally during each stage. We can therefore impose the air chamber inflow flow rate q at a given time t as

$$q = \begin{cases} q_e \sin^{\frac{1}{2}}(\frac{t}{0.3}\pi) & \text{if } t < 0.3 \\ \\ q_f \sin^{\frac{1}{2}}(\frac{t-0.3}{0.45}\pi) & \text{otherwise} \end{cases}, \tag{5.12}$$

where q_e and q_f are constants equal to the peak flow rate of each stage. The constants may be obtained from the equations

$$\int_0^{0.3} q_e \sin^{\frac{1}{2}}(\frac{t}{0.3}\pi)\ \mathrm{d}t = 73, \tag{5.13}$$

and

$$\int_{0.3}^{0.75} q_f \sin^{\frac{1}{2}}(\frac{t-0.3}{0.45}\pi)\ \mathrm{d}t = -73, \tag{5.14}$$

and are equal to $q_e = 319.02$ cc/s and $q_f = -212.68$ cc/s.

On the blood side, we alternate boundary conditions at the inlet/outlet between a Neumann condition and a Dirichlet condition as necessary since we do not directly simulation motion of valves in the simulation. On the outlet, for example, we have two conditions. If we are in the fill stage, then we impose a zero-velocity (i.e., no flow) boundary condition. During the ejection, however, we impose a resistance boundary condition

$$p = C_r q + p_0,$$

where q is the volumetric flow rate on the outlet face, C_r is a prescribed resistance value, p_0 is the distal pressure, and p is the pressure at the outlet face. For the simulation we choose p_0 to be 65 mmHg, which enforces a minimum pressure of 65 mmHg during the expel. The resistance value is set to $C_r = 183$ g/(s cm^4), which gives a maximum systolic pressure of 108 mmHg. The inlet face uses the same boundary conditions, but, obviously, with opposite phase. The structural membrane is simply supported around the circumference.

Remark: Note that, because the incompressible flow assumption is employed for both blood and air, the flow rate into the air chamber must equal to the flow rate out of the blood chamber, and vice versa, the flow rate out of the air chamber must equal to the flow rate into the blood chamber. As a result, by controlling the total volume of the air going in and out of the air chamber, we automatically control the total volume of blood that flows in and out of the blood chamber. The proposed quasi-direct FSI coupling guarantees that at every nonlinear iteration this balance holds. Loosely-coupled FSI approaches, besides being unsuitable for this problem due to the strong added mass effect, cannot guarantee this balance unless special procedures are devised to enforce it (see, e.g., [102]).

Blood, Air, and Membrane Properties Both air and blood are treated as incompressible, Newtonian fluids. The blood density and dynamic viscosity are set to 1 g/cm^3 and 0.04 poise, respectively. The air density and dynamic viscosity are set to $1.205 * 10^{-3}$ g/cm^3 and $2 * 10^{-4}$ poise, respectively.

The membrane is a flexible thin sheet, commonly made of polyurethane. We use membrane material properties consistent with those of the Penn State VAD, the LionHeart [61]. The LionHeart membrane has a thickness of 0.38 mm, density of 1.1 g/cm^3, and a Young's modulus of 550 MPa [61]. In our simulation, we use a thinner membrane of 0.25 mm, which is reflective of the smaller device used for the pediatric population, as was provided in a private communication from the authors of [61]. The membrane initial configuration is obtained by taking a circular disc, which is exactly represented using a single NURBS patch with four corner singularities, and displacing the interior control points in the direction normal to the plain of the disc toward the air chamber. The initial shape of the membrane is assumed to be sinusoidal, and the displacement of the control points is given by the equation

$$d = 1.52 * \cos(\frac{r}{3.85} * \frac{\pi}{2}) \tag{5.15}$$

where r is the radial distance of the control point from the center of the disc, and d is the nodal displacement.

Meshing, Mesh Moving, and Remeshing The blood and air chamber volumes in the reference configuration are meshed using MeshSim(Symmetrix Inc, Clifton Park, NY). The number of elements in the air chamber is 238,322 and in the blood chamber is 497,160. The membrane is discretized using 1,024 C^1-continuous quadratic NURBS elements. The simulations are run for two time cycles of 0.75 s each, with a time step size of 1.0 ms. Generalized-α time integration is used for the coupled FSI equation system (see [42, 90, 16]).

As the computation proceeds, the fluid mechanics mesh is moved using equations of elastostatics with Jacobian-based stiffening [167, 159, 92, 28], which better preserves the mesh quality in the simulations than the no-stiffening approach and and delays the necessity to remesh. However, due to very large motions of the membrane the mesh eventually becomes highly deformed and a remesh is necessary to preserve the quality of the fluid mechanics discretization. The necessity to remesh is quantified in terms of the change in the element volume as measured by the ratio of the Jacobian determinants of the elements in the current step and the step immediately after the previous remesh. For this simulation, remeshing is performed once the ratio of 72% for compression or 170% for expansion is achieved.

During the remesh the surface meshes of the blood and air chamber, including those at the fluid–structure interface, are preserved, and a new tetrahedral mesh is generated on the interior of both subdomains. The solution data at the current step, which includes fluid velocity, acceleration, and pressure, as well as mesh velocity and displacement, is transferred to the new mesh by means of a nodal interpolation procedure that involves the computation of the inverse mapping. To efficiently locate the element in the old mesh containing the nodal point of interest, we use a “point in a polygon” method [126]. Once the data is transferred to the new mesh, the FSI computation continues. No special procedures for transferring the pressure data (e.g., pressure clipping [93, 170, 156]) are employed.

5.3.2 Simulation Results

The VAD simulation was carried out using 4 compute nodes on Trestles, a 100 Tflop/s Appro cluster at the San Diego Supercomputing Center [144]. The

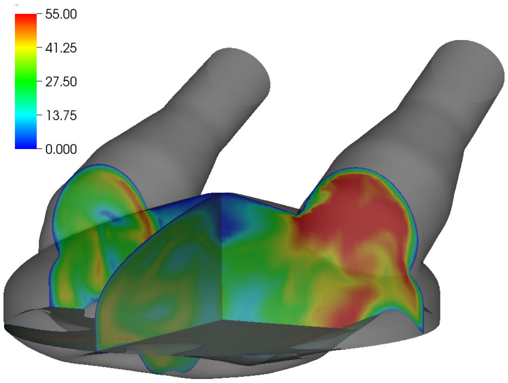


Figure 5.2: Flow speed (cm/s) in the deformed blood chamber configuration at $t = 0.15$ s.

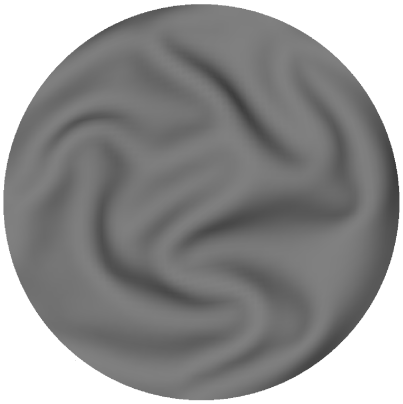
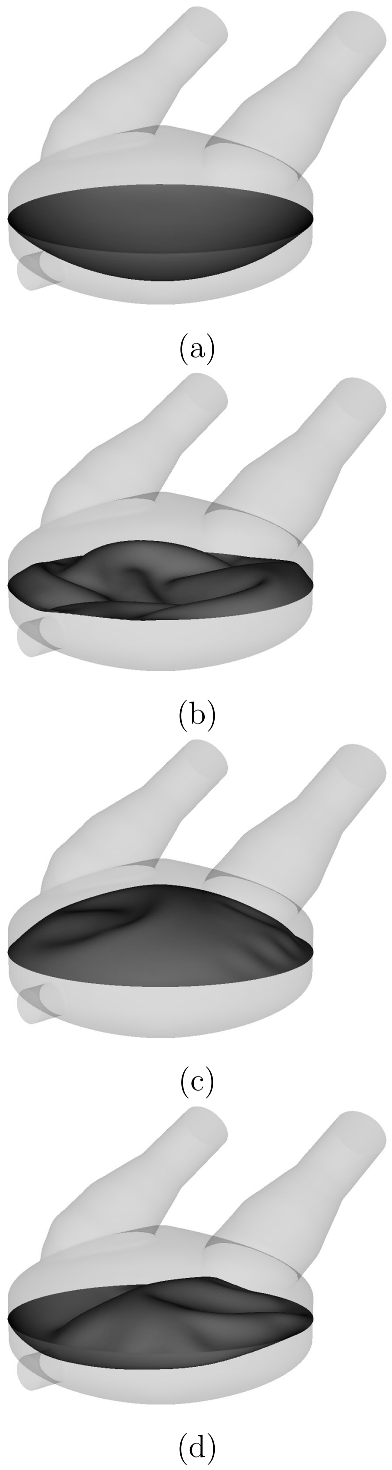


Figure 5.3: Top view of the membrane deformed configuration at $t = 0.15$ s. Despite the complex deformation pattern, the wrinkles on the membrane surface are smooth.

cluster contains 324 compute nodes. Each compute node consists of 4 distinct 8-core 2.4 GHz AMD Magny-Cours processors, for a total of 32 cores per node. The simulation was run for two time cycles. All the data presented is gathered from the second time cycle. The time $t = 0$ refers to the beginning of the second cycle.

Figures 5.2-5.4 show snapshots of the computed blood flow speed and membrane deformation. The simulation captures a very complex membrane motion, with many folds, clearly seen in Figures 5.3 and 5.4. The deformed membrane surface is notably smooth, with no sharp kinks on the mesh edges, which is due



(a)

(b)

(c)

(d)

Figure 5.4: The membrane deformed configuration at time (a) $t = 0$ s, (b) $t = 0.15$ s, (c) $t = 0.3$ s, and (d) $t = 0.525$ s.

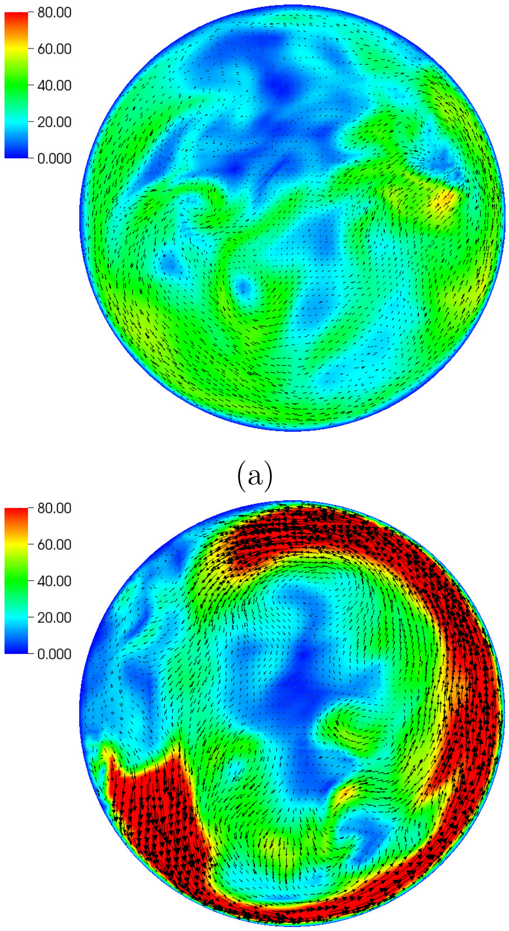
to the underlying smoothness of the NURBS discretization. This buckling motion is smoother than is typically attained using more traditional methods. Since the structural kinematics is used to drive the fluid mechanics mesh deformation, the smoother buckling motion ensures that the fluid mechanics mesh at the fluid–structure interface remains smooth.

During the fill stage, the inlet jet impinges on the chamber wall, and flows along the wall creating a strong vortex. The vortex is destroyed early in the eject phase, as seen in Figure 5.5. This strong vortex is a chief source of the wall shear stress and flow stagnation in the center of the device, and may play an important role in thrombus formation. Strong rotating flow during filling was also observed experimentally in [136] and will be of interest in the future validation efforts.

Figures 5.6 and 5.7 show the time history of volume-averaged pressure and flow speed. The pressure drop across the membrane is small relative to the mean pressure, which is not surprising as it takes little effort to move the membrane. The peak average flow speed in the blood chamber during the fill stage is nearly 20% greater than during the eject stage. Although the eject stage imposes a flow rate 50% higher than the fill stage, the corresponding peak average flow speed is lower. This is in large part due to the rotational flow seen in Figure 5.5, which is present only during the fill stage.

5.4 Conclusions and Future Work

This paper addressed several computational challenges involved in the FSI modeling of pulsatile VADs. These include large, buckling motions of the membrane, the need for periodic remeshing of the fluid mechanics domains, and the necessity to employ tightly coupled FSI solution strategies due to the very strong added mass effect present in the problem. Structural modeling of the membrane makes use of IGA, which has several accuracy and robustness benefits associated with the smoothness of the underlying discretization. The strong FSI coupling is efficiently implemented using a combination of matrix-free and sparse-matrix-based techniques. The simulations captured the essential blood flow features and



(a)

(b)

Figure 5.5: Blood flow speed (cm/s) at 0.5 cm above the plane separating the blood and air chambers. In-plane vectors shown during (a) expel stage ($t = 0.14$ s) and (b) fill stage ($t = 0.665$ s).

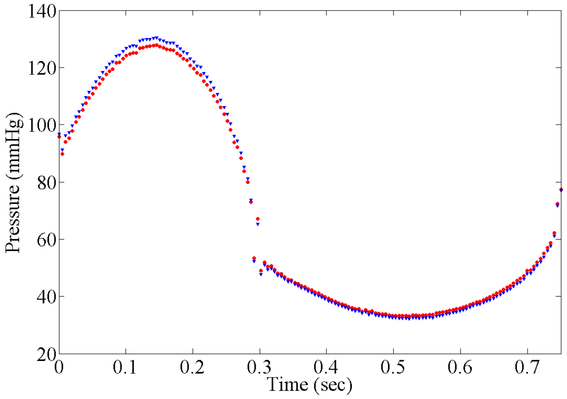


Figure 5.6: Time history of the volume-averaged pressure in the blood and air chambers.

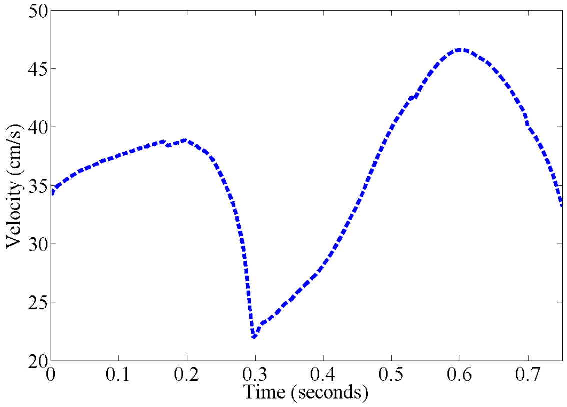


Figure 5.7: Time history of the volume-averaged flow speed in the blood chamber.

structural deformations observed clinically and experimentally in pulsatile VADs. This is the first 3D, full-scale, high-fidelity FSI modeling of pulsatile VADs.

The computational FSI tools developed here provide a foundation for the study of the fluid and structural mechanics inside pulsatile VADs, with clinically relevant implications. We intend to use such simulations in the future to investigate design improvements that will mitigate risk of thrombosis, especially for pediatric populations. Methodically exploring a parameterized design space using computational FSI combined with modern optimization techniques and uncertainty quantification [115, 116, 139, 142, 183] may lead to novel designs that will improve patient outcomes. Thrombosis formation involves a complex interplay between hemodynamics and blood chemistry, presenting significant modeling challenges [180, 181]. Future work could incorporate reduced order models of blood chemistry to capture the key features of this process.

A strong validation effort is also planned. Particle Image Velocimetry data is available for the Penn State device [136], which we intend to simulate in the future.

ACKNOWLEDGEMENTS

The support of the AFOSR Award No. FA9550-12-1-0005 and a Burroughs Wellcome Fund Career Award at the Scientific Interface is gratefully acknowledged. We also thank the San Diego Supercomputing Center (SDSC) at the University of California, San Diego for providing HPC resources that have contributed to the research results reported in this paper.

Chapter 5, in full, is a reprint of the material as it appears in the article "Fluid–Structure Intercation Simulation of Pulsatile Ventricular Assist Devices," published online April 2013 in **Computational Mechanics**. Authors include: C.C. Long, A.L. Marsden, and Y. Bazilevs. The dissertation author was the primary investigator and author of this paper.

Chapter 6

Toward Optimization of a PVAD Device

6.1 Introduction

Over 670,000 diagnoses of heart failure (HF) are predicted this year in the US [105], with a median survival of only 2 to 3 years after initial diagnosis [73]. Cardiac transplantation remains a limited treatment option due to a lack of available donors, and rigorous selection criteria. The demand for alternative treatment options has been partially met with the use of VADs, with as many as $250,000$-$300,000$ potential recipients in the United States [119]. VADs have notably poorer performance in the pediatric population, yet the number of available donor hearts for children is currently fixed at $\approx$ 500 each year. As the need for reliable mechanical circulatory support in this population is growing [46], VAD usage has also increased in this population.

As discussed in Chapter 5, development of VAD technology has taken place almost exclusively for the adult population. The Berlin EXCOR device remains the only FDA approved device for children, and thromboembolic events in these devices occur at an alarming rate. Design modifications may be helpful to reduce the rate of these events, and computational simulations may be helpful in exploring the design space. Simulations are a method of cheaply and quickly testing several device designs simultaneously, identifying the most promising designs and expediting the overall design process. Numerical simulations are already a critical part of the design process in several industries, such as aerospace [121], but have been slow to be adopted by the medical device industry. This is in part due to the challenges and complexities associated with these types of simulations.

The successful physiologic simulation of a PVAD is an important first step towards computer-aided design of these devices. However, successful shape optimization relies on a reasonable cost function, computational expediency, and an automated parameterization approach to model contruction. Approaches to each of these challenges and initial results are presented in this chapter, as well as discussion on the future directions of this work.

6.2 Cost Function Definition

Simulations provide a great wealth of data on flow characteristics inside VAD devices, which can be leveraged for design purposes. For example, the simulation of an airfoil can provide data on the lift and drag characteristics of a particular design. Ideally, a cost function for VAD design should aim to maximize output and efficiency while minimizing the risk of thrombosis and hemolysis. However, it is challenging to asses the effects of VAD design changes on thrombus formation, as the relationship between hemodynamics and thrombus formation is not yet fully understood. Generally, an optimizatoin procedure takes the following basic form:

$$\begin{aligned} &\text{minimize} \quad J(\mathbf{x}) \\ &\text{subject to} \quad \mathbf{x} \in \Omega, \end{aligned} \tag{6.1}$$

where J is a *cost function*, which returns a single scalar value correlated to the desired outcome based on a set of variable design parameters $\mathbf{x}$. We define Ω as the design space of the parameters, which may include upper or lower bounds on individual parameters in $\mathbf{x}$. Further discussion of optimization is discussed in Section 6.3.

Thrombus formation is the result of a complex sequence of chemical reactions in the bloodstream, resulting in activated platelets aggregating and fibrin networks forming around these aggregations. A natural starting place for assessing thrombotic risk is then to model these chemical reactions as faithfully as possible. Then, a simple value such as the total number of platelets activated may suffice as an appropriate cost function. However, a discrete approach (e.g., modeling each platelet and molecular compound directly) is inherently costly, and is limited to very small time scales, whereas clots may form over several hours or days in the clinic [69]. Computational fluid dynamics simulations of VADs model blood as a continuous medium, and therefore it is more natural to model the appropriate chemical compounds and platelet concentrations as continuous variables within that medium. A set of differential equations may then be written to describe the chemical cascade, with the concentrations of chemicals being advected throughout the flow. A great deal of research has been dedicated to the proper development of this approach [152, 149, 150, 3, 107, 52], however most rely on a chemical trigger (von Willebrand factor) which initiates the reactions. Physiologically, the von Willebrand factor is secreted by an injured vessel wall, which is absent in the interior of a VAD. The initiation of this chemical cascade due purely to hemodynamics and exposure to artificial material is not well-studied, and an initial attempt at a coagulation implementation resulted in time scales much longer than were computationally feasible. Additionally, the heavy usage of anti-coagulation therapy in VAD patients affects blood chemistry and complicates analysis.

Due to the difficulty in determining a cost-effective and reasonably physiologic blood coagulation model, we explore other surrogates for thrombotic risk that may be directly computed from simulation data. Long residence times and areas of recirculation or stagnation may lead to increased risk of thrombosis [34, 134], and we formulate methods to extract this information from simulations in an automated way in Sections 6.2.3 and 6.2.4. Results for these initial analyses are also provided.

6.2.1 Pediatric and Adult Device Simulations

Our choice of cost function parameters are subject to validation. It is well known that adult models of pulsatile VADs perform better in regard to thrombogenesis than existing pediatric models. Thus, simulations of both an existing adult model and pediatric model should provide a reasonable basis to verify that our chosen cost function variables reflect the increased risk associated with pediatric devices. The simulation carried out in Chapter 5 was based on an adult-sized Berlin Heart EXCOR device, and serves as the adult model for this analysis. An additional simulation of a 10 mL stroke volume Berlin Heart EXCOR device is carried out to serve as the pediatric model for comparison purposes.

The 10 mL simulation was carried out using 12 compute nodes on the Kraken cluster at Oak Ridge National Laboratory. Kraken is a Cray XT5 cluster, with 9408 nodes, each equipped with two 2.6 GHz six-core AMD Opteron processors, for a total of 12 processors per node. The simulation was run for a total of three time cycles. All data presented is from the third time cycle, and we denote $t = 0$ as the beginning of the third cycle. To match typical clinical values, a beat frequency of 100 bpm is used for the 10 mL device, which yields an average pump flowrate of 1 L/m. The flowrate at a given point in time is defined similarly to the adult model, and Equation 5.12 is modified for the pediatric device as follows:

$$q = \begin{cases} q_e \sin^{\frac{1}{2}}(\frac{t}{0.24}\pi) & \text{if } t < 0.24 \\ \\ q_f \sin^{\frac{1}{2}}(\frac{t-0.24}{0.36}\pi) & \text{otherwise} \end{cases}, \tag{6.2}$$

where q_e and q_f are the peak flowrate of each stage, solved in the same manner as the adult model. The values used are $q_e = 54.63$ cc/s and $q_f = -36.42$ cc/s.

The number of elements in each domain fluctuates with remeshing, but is initialized with 623570 elements in the blood chamber and 302102 elements in the air chamber. Each remesh respects the initial length scale of the elements. The membrane is again represented with 1024 C^1-continuous quadratic NURBS elements. The time cycle period is 0.6 s, and a time step size of 0.8 ms is used.

6.2.2 Advection-Diffusion Solver

A finite element based advection-diffusion solver is used to post-process the simulation results. This process has been previously used successfully to compute particle residence time in fixed domain FEM fluid simulations, and a full discussion is available in [64]. The advection diffusion solver requires an advective velocity, denoted $\mathbf{u}_a$, which is computed using the mesh nodal locations, connectivity, mesh velocity and fluid velocity from the completed simulations. The advection problem is shown below:

$$\frac{\partial \tau}{\partial t}|_{\hat{\mathbf{x}}} + \mathbf{u}_a \cdot \nabla \tau - \nabla \cdot \kappa \nabla \tau - H = 0, \tag{6.3}$$

where τ is a nodal scalar quantity which is advected through the flow, H is a source term, and κ is the diffusivity. The subscript $\hat{\mathbf{x}}$ on the time derivative term indicates the time derivative is with respect to the referential domain. The advective velocity at a given nodal point is computed as

$$\mathbf{u}_a = \mathbf{u} - \hat{\mathbf{u}}, \tag{6.4}$$

where $\mathbf{u}$ and $\hat{\mathbf{u}}$ are the nodal values of fluid velocity and mesh velocity determined from the computational results. A finite element approach is used to solve Eq. 6.3, and we reformulate to a stabilized weak form as shown in [86, 64]. Find $\tau \in S^h$ such that for all $q \in V^h$, the following holds:

$$\int_{\Omega_t} q \cdot \left(\frac{\partial \tau}{\partial t} + \mathbf{u}_a \cdot \nabla \tau \right) \mathrm{d}\Omega + \int_{\Omega_t} \nabla q \cdot \bar{\kappa} \nabla \tau \mathrm{d}\Omega + \sum_e \int_{\Omega_e} \nabla q \cdot \mathbf{u}_a \tau_m \mathbf{R}^{adv} \mathrm{d}\Omega = 0, \tag{6.5}$$

where $\mathbf{R}^{adv}$ is the RHS residual, computed such that:

$$\mathbf{R}^{adv} = \frac{\partial \tau}{\partial t} + \mathbf{u}_a \cdot \nabla \tau - \nabla \cdot \kappa \nabla \tau - H, \tag{6.6}$$

and τ_m and $\bar{\kappa}$ are stability terms defined as:

$$\tau_m(\mathbf{x}, t) = \frac{1}{\sqrt{\frac{4}{\Delta t^2} + \mathbf{u}_a^T \xi \mathbf{u}_a + 3\bar{\kappa}^2 \xi : \xi}}, \tag{6.7}$$

$$\bar{\kappa}(\mathbf{x}, t) = \kappa + \frac{|\mathbf{R}^{adv}|}{2\sqrt{\nabla \tau^T \xi \nabla \tau}}. \tag{6.8}$$

In the moving domain case, $\mathbf{x}$ refers to the current configuration, which is gathered at each time step from our previously computed data.

The partial differential equation in Eq. 6.6 is then solved using a generalized-α time integration scheme, and is run for any arbitrary number of cycles. For each simulation, we use a single time cycle, and loop the results as necessary into the advection-diffusion solver. That is, at time $t = T + \Delta t$, where T is the time period of the cycle, and Δt is the time step size, the results from time $t = \Delta t$ are used to continue the analysis. This is advantageous as the FSI simulation is cost prohibitive to run for more than 2-3 cycles, but the advection-diffusion analysis may continue as long as necessary.

The remeshing capability of the FSI simulation poses a unique problem to the post-processing analysis, as the internal mesh is not consistent. Each time a remesh occurs during the FSI simulation, a flag is placed in the output data to note it is the final data set on a given mesh. The final mesh is written out separately, and we denote it Ω_1^h. After the remesh, a new mesh at the exact same point in time is created, denoted Ω_2^h. Since the surface mesh is unmodified, and $t_1 = t_2$, the facts that $\Omega_1 = \Omega_2$ and $\Gamma_1^h = \Gamma_2^h$ are guaranteed, although the internal discretizations, Ω_1^h and Ω_2^h, vary. When the advection-diffusion solver encounters this flag, the post-processor stops and writes out the the nodal values of τ^h on Ω_1^h. A point-in-a-polygon method is then used to interpolate the solution field of τ onto Ω_2^h in an efficient way [126]. The post-processing can then continue unimpeded.

Remark: Consider the domains Ω_0^h and Ω_T^h, which we define as the initial discretized mesh at time $t = 0$, and the final discretized mesh at $t = T$. It should be noted that due to remeshing, the domain $\Omega_T^h \neq \Omega_0^h$, and we must perform an interpolation of the solution field τ at the beginning of each subsequent cycle as well. However, the point-in-a-polygon method requires that $\Gamma_0^h = \Gamma_T^h$, which is NOT guaranteed in this case. The simulation is periodic, but the membrane structure does not settle in exactly the same configuration at the end of each cycle, and we can only guarantee that $\Gamma_0^h \approx \Gamma_T^h$. The differences in Γ_0^h and Γ_T^h are quite small

(≤ 1 mm for a given node), and we define a vector field $\Phi(\mathbf{x})$ on Γ_0^h such that

$$\mathbf{x} = \tilde{\mathbf{x}} + \Phi(\tilde{\mathbf{x}}), \tag{6.9}$$

where $\mathbf{x}$ denotes the positions of the nodes on Γ_T^h, and $\tilde{\mathbf{x}}$ denotes the nodal coordinates on Γ_0^h. The perturbed initial mesh is denoted $\bar{\Gamma}_0^h$, and the definition of Φ enforces that $\bar{\Gamma}_0^h = \Gamma_T^h$. Given this condition, the point-in-a-polygon method is used to transfer data, and the simulation can continue.

6.2.3 Residence Time Computations

To compute the residence time in a particular region, of the time-varying domain $\Omega_\tau(t)$, the source term $H(\mathbf{x})$ in Eq. 6.3 is defined as

$$H(\mathbf{x}, t) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_\tau(t) \\ \\ 0 & \text{otherwise} \end{cases}, \tag{6.10}$$

where $\mathbf{x}$ denotes the current coordinates of a nodal point. This definition enforces that τ is incremented in time whenever a particle enters Ω_τ, and Eq. 6.3 may be rewritten in the Lagrangian frame as:

$$\frac{\mathrm{d}\tau}{\mathrm{d}t} = H(\mathbf{x}, t). \tag{6.11}$$

Thus, for a particle in $\Omega_\tau(t)$, we note that $\frac{\mathrm{d}\tau}{\mathrm{d}t} = 1$. For all computations presented, we define $\Omega_\tau(t) = \Omega_t$, but Ω_τ in general may be any time-varying subdomain of particular interest.

The advection diffusion equations are solved while looping over the periodic cycles, and computation is ceased when the solution field τ becomes periodic. The resulting solution field is the point-wise particle residence time. In [64], Esmaily-Moghadam explores two primary methods of processing the resulting solution field to indicate a single scalar residence time for a cycle, denoted RT_1 and RT_2, assuming a fixed domain. These parameters have moving-domain analogs which must be derived. The value RT_1 is defined as a time- and volume-averaged value of τ. For a fixed domain, RT_1 may be computed as

$$RT_1 = \frac{1}{V_\tau T} \int_{(N-1)T}^{NT} \int_{\Omega_\tau} \tau(\mathbf{x}, t) \mathrm{d}\Omega \mathrm{d}t, \tag{6.12}$$

where V_τ is the volume of Ω_τ, and T is the time period of the cycle. This is redefined for the moving domain as

$$RT_1 = \frac{1}{T}\int_{(N-1)T}^{NT} \frac{1}{V_t}\int_{\Omega_\tau(t)} \tau(\mathbf{x}, t)\mathrm{d}\Omega\mathrm{d}t, \tag{6.13}$$

where V_t is the volume of Ω_τ at time t, defined as

$$V_t = \int_{\Omega_\tau(t)} \mathrm{d}\Omega. \tag{6.14}$$

The second parameter, RT_2, is found by time-averaging the boundary integral as shown

$$RT_2 = \frac{1}{T\bar{Q}}\int_{(N-1)T}^{NT}\int_{\Gamma_\tau} (\tau\mathbf{u}_a - \kappa\nabla\tau)\cdot\mathbf{n}\mathrm{d}\Gamma\mathrm{d}t, \tag{6.15}$$

where $\bar{Q}$ is the average flux through Ω_τ, Γ_τ is the time-varying boundary of Ω_τ, and $\mathbf{n}$ is the unit outward normal on Γ_τ. $\bar{Q}$ may be computed explicitly:

$$\bar{Q} = \frac{1}{T}\int_{(N-1)T}^{NT}\int_{\Gamma_{\tau_o}} \mathbf{u}_a\cdot\mathbf{n}\mathrm{d}\Gamma\mathrm{d}t, \tag{6.16}$$

where Γ_{τ_o} is used to denote the outlet boundary at a given time t. It may be formally defined as the set of coordinates that exist on the boundary, Γ, such that $\mathbf{u}_a\cdot\mathbf{n} \geq 0$ at time t.

Esmaily-Moghadam [64] found that Eq. 6.15 can be further reduced to eliminate all time-dependent integrals. For a fixed domain, $RT_2 = \frac{V_\tau}{\bar{Q}}$. The derivation on a moving domain yields a similar result. Integration over Ω_τ on Eq. 6.3 yields

$$\begin{aligned}\int_{(N-1)T}^{NT}\int_{\Omega_\tau} \frac{\partial\tau}{\partial t}|_{\hat{\mathbf{x}}}\mathrm{d}\Omega\mathrm{d}t \\ + \int_{(N-1)T}^{NT}\int_{\Omega_\tau} \left(\mathbf{u}_a\cdot\nabla\tau - \nabla\cdot\kappa\nabla\tau - H\right)\mathrm{d}\Omega\mathrm{d}t = 0.\end{aligned} \tag{6.17}$$

We split the integration into two terms for convenience, and will work on them separately. Using the product rule, we note that:

$$\mathbf{u}_a\cdot\nabla\tau = \nabla\cdot\tau\mathbf{u}_a - \tau\nabla\cdot\mathbf{u}_a. \tag{6.18}$$

Substitution of Eq. 6.18 and an application of the divergence theorum to the second integral yields:

$$\int_{(N-1)T}^{NT}\int_{\Gamma_\tau}(\tau\mathbf{u}_a-\kappa\nabla\tau)\cdot\mathbf{n}\mathrm{d}\Gamma\mathrm{d}t-\int_{(N-1)T}^{NT}\int_{\Omega_\tau}\tau\nabla\cdot\mathbf{u}_a+H\mathrm{d}\Omega\mathrm{d}t. \tag{6.19}$$

By definition, $H = 1$ on Ω_τ, and so we can rewrite:

$$\int_{(N-1)T}^{NT}\int_{\Gamma_\tau}(\tau\mathbf{u}_a-\kappa\nabla\tau)\cdot\mathbf{n}\mathrm{d}\Gamma\mathrm{d}t-\int_{(N-1)T}^{NT}\int_{\Omega_\tau}\tau\nabla\cdot(\mathbf{u}-\hat{\mathbf{u}})\mathrm{d}\Omega\mathrm{d}t-\bar{V}T, \tag{6.20}$$

where $\bar{V}=\frac{1}{T}\int_{(N-1)T}^{NT}V_t\mathrm{d}t$. By continuity, $\nabla\cdot\mathbf{u}=0$, and we can rewrite again:

$$\int_{(N-1)T}^{NT}\int_{\Gamma_\tau}(\tau\mathbf{u}_a-\kappa\nabla\tau)\cdot\mathbf{n}\mathrm{d}\Gamma\mathrm{d}t+\int_{(N-1)T}^{NT}\int_{\Omega_\tau}\tau\nabla\cdot\hat{\mathbf{u}}\mathrm{d}\Omega\mathrm{d}t-\bar{V}T. \tag{6.21}$$

We will now move the final term, $\tau\nabla\cdot\hat{\mathbf{u}}$, into the first integral term of Eq. 6.17, as shown:

$$\int_{(N-1)T}^{NT}\int_{\Omega_\tau}\frac{\partial\tau}{\partial t}|_{\hat{\mathbf{x}}}+\tau\nabla\cdot\hat{\mathbf{u}}\mathrm{d}\Omega\mathrm{d}t+\int_{(N-1)T}^{NT}\int_{\Gamma_\tau}(\tau\mathbf{u}_a-\kappa\nabla\tau)\cdot\mathbf{n}\mathrm{d}\Gamma\mathrm{d}t=\bar{V}T. \tag{6.22}$$

An identity from structural mechanics [178] which states

$$\frac{\partial\hat{J}}{\partial t}|_{\hat{\mathbf{x}}}=\hat{J}\nabla\cdot\hat{\mathbf{u}}, \tag{6.23}$$

where $\hat{J}=\det\hat{\mathbf{F}}$ and $\hat{\mathbf{F}}$ is the standard deformation gradient, may be employed on the term $\tau\nabla\cdot\hat{\mathbf{u}}$, yielding:

$$\begin{aligned}\int_{(N-1)T}^{NT}\int_{\Omega_\tau}&\frac{\partial\tau}{\partial t}|_{\hat{\mathbf{x}}}+\tau\nabla\cdot\hat{\mathbf{u}}\mathrm{d}\Omega\mathrm{d}t\\ &=\int_{(N-1)T}^{NT}\int_{\Omega_\tau}\frac{\partial\tau}{\partial t}|_{\hat{\mathbf{x}}}+\frac{\tau}{\hat{J}}\frac{\partial\hat{J}}{\partial t}|_{\hat{\mathbf{x}}}\mathrm{d}\Omega\mathrm{d}t\\ &=\int_{(N-1)T}^{NT}\int_{\Omega_\tau}\frac{1}{\hat{J}}\frac{\partial\hat{J}\tau}{\partial t}|_{\hat{\mathbf{x}}}\mathrm{d}\Omega\mathrm{d}t.\end{aligned} \tag{6.24}$$

Using a spatial transform defined as $\hat{\tau}=\hat{J}\tau$, we change variables in this integral term from $\Omega_t\rightarrow\Omega_0$, as shown:

$$\int_{(N-1)T}^{NT}\int_{\Omega_0}\frac{\partial\hat{J}\tau}{\partial t}|_{\hat{\mathbf{x}}}\mathrm{d}\Omega\mathrm{d}t. \tag{6.25}$$

Since the integral is now in terms of the reference domain Ω_0, the order of integration is now arbitrary. Swapping the order yields:

$$\begin{aligned}\int_{\Omega_0}\int_{(N-1)T}^{NT} \frac{\partial \hat{J}\tau}{\partial t}|_{\hat{\mathbf{x}}}\mathrm{d}t\mathrm{d}\Omega \\ = \int_{\Omega_0} \hat{J}(NT,\hat{\mathbf{x}})\tau(NT,\hat{\mathbf{x}}) - \hat{J}((N-1)T,\hat{\mathbf{x}})\tau((N-1)T,\hat{\mathbf{x}})\mathrm{d}\Omega. \quad (6.26)\end{aligned}$$

Since the solution field and mesh motion are both cyclic, we can eliminate this integral entirely, such that Eq. 6.26= 0. Equation 6.22 then reduces to:

$$\int_{(N-1)T}^{NT}\int_{\Gamma_\tau} (\tau\mathbf{u}_a - \kappa\nabla\tau)\cdot\mathbf{n}\mathrm{d}\Gamma\mathrm{d}t = \bar{V}T. \quad (6.27)$$

Finally, combining Eqs. 6.27 and 6.15 yields

$$RT_2 = \frac{\bar{V}}{\bar{Q}}, \quad (6.28)$$

which can be computed a priori if $\Omega_\tau = \Omega_t$.

A new parameter, denoted $\bar{\tau}(t)$, is also computed. We define $\bar{\tau}$ as the volume averaged value of τ at a given time t, or

$$\bar{\tau} = \frac{1}{V_t(t)}\int_{\Omega_\tau(t)} \tau(\mathbf{x},t)\mathrm{d}\Omega. \quad (6.29)$$

A plot of $\bar{\tau}$ with time for both devices can be seen in Figure 6.1. The two curves in this plot have similar shapes, but with differing slopes. Both devices have a uniformly increasing value of $\bar{\tau}$ during the ejection phase. This is expected, as no new material is entering Ω_τ, and the average is expected to rise with time. If Ω_τ is well-mixed, a linear increase is expected, and is seen on the 73 mL curve. However, an inhomogeneous mixture of τ values in Ω_τ will result in a non-linearities during this phase, as 'chunks' of fluid with variable τ values leave the system. This is the behavior seen in the 10 mL curve. The injection phase shows a brief and rapid decrease in $\bar{\tau}$, as new material with $\tau = 0$ enters Ω_τ. The trend reverses and $\bar{\tau}$ begins to increase again as the influx of new material slows and the volume V_t grows.

Values of RT_1 and RT_2 are computed for both devices, and are shown in Table 6.1. The maximum global value of τ and $\bar{\tau}$ are also reported. The value

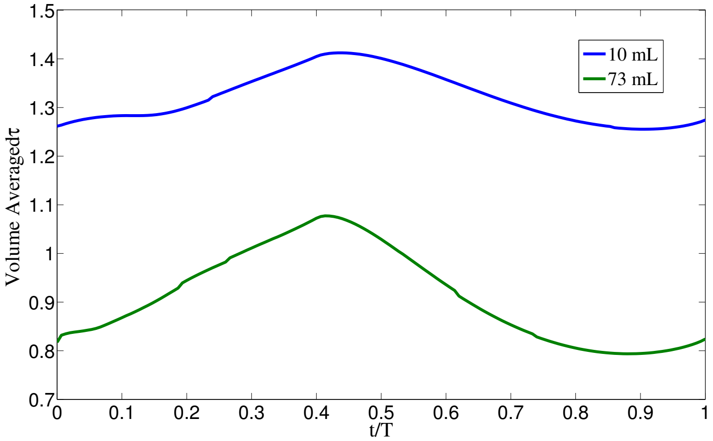


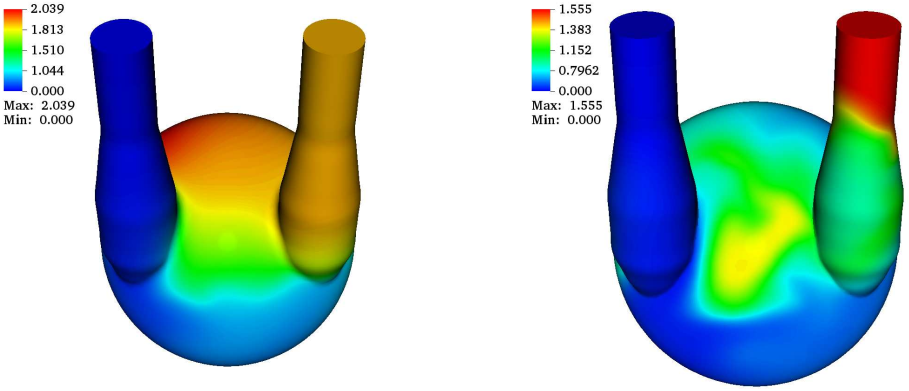
Figure 6.1: Plot of $\bar{\tau}$ with non-dimensional time, t/T. The fill stage is defined as $t/T \in [0, 0.4]$, and the eject phase as $t/T \in (0.4, 1]$

Table 6.1: Residence time results for both devices. RT_1 and RT_2 are computed as described above. The value max(τ) is the maximum nodal value of τ achieved at any point in the simulation, and max($\bar{\tau}$) is the maximum volume-averaged value of τ at any point in the simulation.

	$\bar{V}$ (cm^3)	$\bar{Q}$ ($\frac{\text{mL}}{\text{s}}$)	RT_1 (s)	RT_2 (s)	max(τ) (s)	max($\bar{\tau}$) (s)
10 mL	27.26	16.67	1.320	1.635	2.04	1.413
73 mL	100.49	97.50	0.913	1.031	1.55	1.077

max(τ) corresponds to the maximum residence time for a given particle in the device, and max($\bar{\tau}$) corresponds with the maximum average residence time for the device. All values consistently show that the adult device is able to transport fluid more effectively, which may contribute to its superior clinical performance.

Further information can be ascertained by visualizing the solution field, τ. In Fig. 6.2, τ is shown in both devices at the final time step of the final cycle. The maximum values are different, but the distribution of τ also differs in significant ways. In the 73 mL device, the bulk of the 'oldest' material resides directly in the outlet branch. These images are taken at the very beginning of the ejection stage, and this device is set to immediately eject its oldest material. The 10 mL device, however, does not efficiently eject the oldest material. A stagnation zone



(a)

(b)

Figure 6.2: Visualization of τ at the final time step in (a) the 10 mL device, and (b) the 73 mL device.

forms beneath the inlet arm, where τ grows, and $\max(\tau)$ is observed. Thus, the oldest material in this device is fairly distal to the outlet arm, which immediately suggests this device may be problematic. Further, the distribution of residence time in the 10 mL device is generally more heterogeneous than in the adult model, with the bulk of the 'old' material residing in the 'back' half-circle of the device, while the 'front' half-circle is filled with new material.

6.2.4 Dye Injection

In this second approach, the source term is set as $H = 0$, and a Dirichlet condition on τ is imposed at the inlet such that:

$$\tau = \begin{cases} 1 & \text{if } N = 1 \\ \\ 0 & \text{otherwise} \end{cases}, \tag{6.30}$$

Table 6.2: Percentage of dye remaining at the end of each cycle, subsequent to the initial 'fill' cycle. The time required to remove ≈ 95% of dye is reported in the final column.

Cycle No.	1	2	3	4	5	6	7	T_{95}
10 mL	78.92	48.89	29.65	19.08	12.22	7.80	4.98	3.83
73 mL	48.13	20.44	8.71	3.72	-	-	-	2.47

where N is the cycle number. This has the effect of 'injecting' a dye for one cycle, which can be visualized as it moves through the domain. The volume of dye in the domain at any given time, V_{dye}, can be written as follows:

$$V_{dye} = \int_{\Omega_t} \tau \mathrm{d}\Omega. \tag{6.31}$$

At time $t = 0$, V_{dye} has a non-zero value due to the Dirichlet condition at the inlet, but this effect adds a trivial amount of 'dye' to the system. After the completion of the first time cycle, $V_{dye} \approx$ S.V., where S.V. is the stroke volume of the device simulated. The dye is then ejected over subsequent cycles, and we compute the volume of dye remaining at the end of each cycle. The cycles are repeated until at least 95% of the dye is removed from the system. These results can be seen in Table 6.2. The table shows that the adult device clearly ejects more fluid each cycle than the pediatric device. After accounting for the shorter time period of the pediatric device, the adult device still outperforms the pediatric model in total time until 95% ejection is reached. This time is denoted T_{95}, and is also reported in Table 6.2. The percentage of dye remaining vs. time is shown for both devices in Figure 6.3.

6.3 Optimization Framework

The fundamental goal of an optimization framework is to solve Eq. 6.1 for a given cost function and set of design constraints. The design space Ω may be as multi-dimensional as necessary for a given problem. In the case of the PVAD problem, there are many obvious design parameters to choose from. As a simple

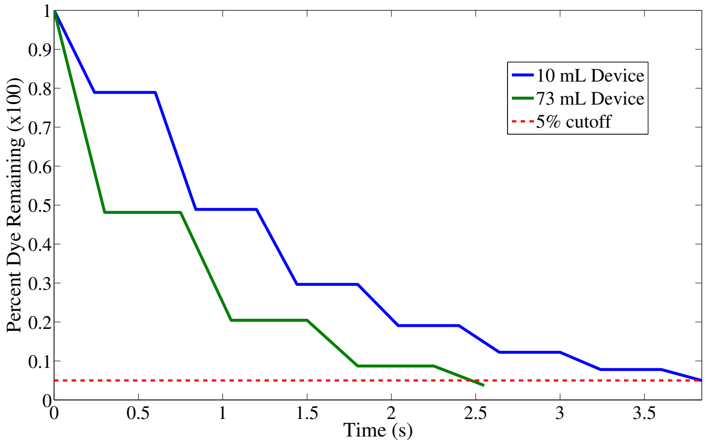


Figure 6.3: Plot of percentage of dye remaining vs. time for both devices

1D example, consider the parameter α, which we define as the angle of the inlet arm coming off the main blood chamber, with respect to the base axis. Constraints are then placed on the parameter, and in this example we enforce: $10^\circ \leq \alpha \leq 85^\circ$. The optimization scheme will evaluate J at several values of α, and will cease when it has found a minimum. The ending value of α is thus the optimal angle to use for a branch arm within our limited design space. A higher-dimensional design space operates similarly, where all parameters in the design space are optimized simultaneously.

Each evaluation of J requires a full simulation of a PVAD design that was constructed with parameter values $\mathbf{x}$, where $\mathbf{x}$ is a list of n parameter values used in model construction under the constraint $\mathbf{x} \in \Omega$. After simulation, the results must be post-processed to evaluate J. A simple choice for J is shown as follows:

$$J = \frac{RT_1 + RT_2 + \max(\tau)}{3} + T_{95}. \tag{6.32}$$

This choice for J yields $J_{10} = 5.495$, and $J_{73} = 3.635$ for the 10 and 73 mL devices, respectively. Alternative combinations of weights or norms can of course be introduced to emphasize or deemphasize various parameters.

The full evaluation of a three-dimensional FSI simulation for each evaluation of J makes this type of shape optimization computationally expensive. Previous efforts at optimization in this realm have been mostly limited to 2-D or steady flow [2, 1, 8, 38].

A current optimization method that has been successfully adopted for use in cardiovascular simulations uses the surrogate management framework (SMF) [35, 147, 10]. If we consider J as a function of n parameters, then we can construct an n-dimensional interpolating function that intersects all known evaluations of J. This function is referred to as the *surrogate.* With each new evaluation of J, the surrogate function is updated. Kriging has been the most common method of choice surrogate construction, though other approaches can be used, including reduced order simulation models.

As the surrogate becomes well-seeded with data, it approaches a more faithful representation of the true function. Analyzing the surrogate function is relatively easy compared with function evaluations, and we can use the surrogate to locate a minimum in Ω. This is used as an educated 'guess' for our next choice of function evaluation. This is the key advantage to the SMF method, as it can efficiently approach reasonable design parameters, saving costly function evaluations.

We discretize the n-dimensional design space, where each nodal mesh point is a set of potential design parameters. After initially 'seeding' the surrogate function, the algorithm has two basic steps:

1. `SEARCH` step. Use the Kriging surrogate function to find a likely set of parameters, $\mathbf{x}$, to minimize J.

2. `POLL` step. At the currently best known nodal position in the function space mesh, a positive basis set in n-space is generated. This set forms a 'poll set' of neighboring points in the design space to evaluate.

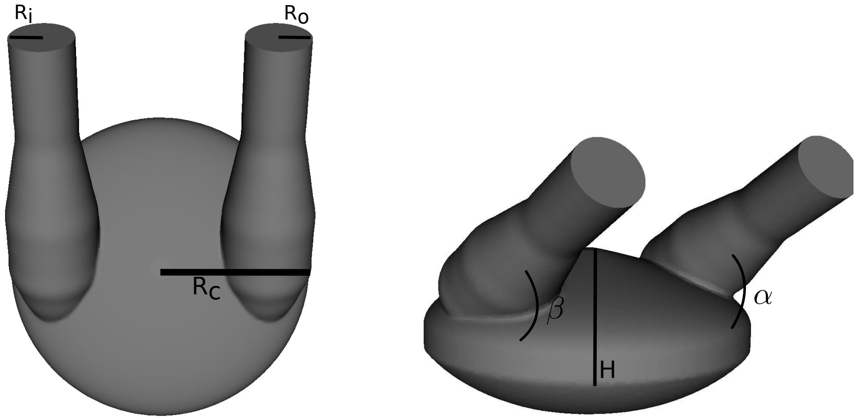
The traditional approach after initially seeding the surrogate is to perform a `SEARCH` step and perform a cost function evaluation. Then, use the evaluation to update the surrogate. Another `SEARCH` step is performed, as the updated surrogate may have a new minimum. This process is repeated as improved points are identified by the surrogate. Then, the algorithm proceeds to a `POLL` step. The `POLL` step explores a minimal set of adjacent points in the mesh in a systematic way, in an attempt to find a new minimum. If the `POLL` step is successful, another `POLL` step is performed at the newly minimum. If the `POLL` step is unsuccessful, the function space discretization is refined, the surrogate is updated, and a `SEARCH` step is performed. The algorithm is exited after an unsuccessful `POLL` step, if a user-defined mesh discretization tolerance has been reached. While several cost function analyses may be required during a `POLL` step, these evaluations may be performed in parallel given adequate computing resources.

The SMF framework has been previously been successfully coupled to 3-D Navier-Stokes simulations. It has been used to minimize noise-emanating trailing-edge vortices and broadband turbulence on airfoils [114, 116], and more recently has been successfully applied to cardiovascular problems [115]. SMF was used to perform shape optimization of a Y-graft conduit used in the Fontan surgical bypass procedure [183, 182]. It has also been integrated successfully with uncertainty quantification, and the optimal anastomosis angle in a cornary bypass graft was shown to shift when taking surgical uncertainties into account [139, 141].

Several variables suggest themselves for use in the design space, Ω. Geometric properties of the device are the most obvious, and six values have already been parameterized in the model generation software. These are: R_c, R_i, R_o, α, β, and H, as seen in Figure 6.4. These correspond with the radius of the main pumping chamber, the radii of the inlet and outlet branch arms, the angle of the inlet and outlet branches with respect to the membrane plane, and the height of the pumping chamber, respectively.

Other geometric values should also be considered. For example, the attachment coordinates of each branch arm on the main chamber are an interesting option. An offset in the arm attachment points may produce distinct changes in the haeodynamics of the device, and could significantly alter the pump characteristics. Two new angles, γ and ϕ, could also be used to 'swivel' the branch arms in the membrane plane. This could lead to a design in which the branch arms attach in opposite directions. This would potentially improve the performance of the 10 mL device, as the stagnation zone in the upper region of Fig. 6.2(a) would be preferentially ejected if the outlet branch were rotated 180°.

Other non-geometric parameters may also prove useful to investigate. Oper-



(a) (b)

Figure 6.4: Parameters currently used in PVAD model generation.

ational parameters, such as beat frequency, stroke volume, and the relative length of fill and eject phases may significantly alter device function. An initial optimization involving only these parameters may be useful to determine the ideal operating conditions for devices already in use on the market, without requiring a lengthy and costly FDA approval process for a physical design change.

Membrane properties can also be optimized. Some basic membrane properties are obvious candidates, such as: the Elastic modulus, E, and the thickness, h. However, we can also introduce material anisotropy, or experiment with the initial stress-free shape of the membrane. Both may give the membrane preferred buckling modes, which could be useful in disrupting stagnation zones and strong vortices. Anisotropy is fairly simple to implement into the optimization with a few additional parameters. Complex stress-free membrane configurations may be more difficult to implement, but a parameterization of the membrane geometry itself will make this possible.

We have discussed more than 15 easily parameterized design variables which may have a profound effect on device performance. However, using higher numbers of design parameters also requires performing more computations to seed the surrogate and find a minimum. To alleviate this, an initial 2D sensitivity study may be prudent to identify the most impactful parameters.

6.4 Conclusions and Future Work

Initial efforts to develop a cost function capable of predicting thrombotic risk are promising. Candidate components of a cost function include the previously studied residence time variables, RT_1 and RT_2, which were expressly introduced to quantify residence time and thrombotic risk in cardiovascular simulations [64]. Several other candidates are presented as possible components to a cost function, such as dye ejection rates and $\max(\tau)$ values. All studied values predict a higher risk in pediatric devices, corresponding with observed clinical performance.

The dye injection/ejection computations in particular are a useful development, as they have a twofold purpose. This computation is promising for cost function evaluations, as discussed, but also has a clear physical analog that can be performed experimentally. A PVAD may be operated under controlled conditions, with dye injected for several time cycles, and then cut off. This would provide a viable means of validating our PVAD simulations, which has not yet been feasible.

The simulations themselves are remarkably robust. The process of simulation and post-processing is fully automated; the initial mesh decomposition, the parallel execution of the FSI code, the intermittent remeshing sequences, and the post-processing described in Secs. 6.2.3 and 6.2.4 require no input or maintenance after the process has begun. This is a critical feature to ensure smooth integration with an optimization scheme. This fact, combined with the early success of our cost function analysis, suggests PVADs are excellent candidates for computational shape optimization. Future work will entail fully integrating these computational functionalities within a formal optimization framework as described in Sec. 6.3.

Integration with an optimization scheme requires a formal definition of the parameterization space. While a multitude of possible design parameters have been discussed, a sensitivity analysis of these parameters may identify a few key parameters with the most potential impact on performance. Additionally, a methodology for automatic model generation is required, and not yet implemented. While the simulation and post-processing are fully automated, model generation itself remains a somewhat tedious and manual task.

Continued efforts to develop the cost function are warranted. Despite the initial success of the residence time and dye injection computations, a broader range of variables that predict thrombotic risk are desirable. For example, an analysis of wall shear stress is an obvious candidate for the cost function evaluation, and is not yet implemented. While it is currently possible to compute wall shear stress, the remeshing sequence disrupts the boundary layer mesh. Thus the computation of wall shear as currently implemented is potentially underresolved. A method for circumventing this problem has been developed, but is not yet fully implemented. Instead of a single discretized volume, Ω^h, two distinct volume meshes are created. One contains the interior volume, Ω^h_{int}, and the other is comprised solely of the boundary layer, Ω^h_{BL}. A matching interface allows for seamless computation. The remeshing sequence then generates a new interior mesh, while retaining the original boundary layer throughout the simulation. Fully implemented, this approach will allow accurate wall shear computations to be included in the cost function. Other efforts should continue to be pursued as well. Coupling simulation results with a reduced order model of blood chemistry has enormous potential to improve the cost function analysis. It is worthwhile to explore this possibility, although it remains a long-term goal.

Bibliography

[1] F. Abraham, M. Behr, and M. Heinkenschloss. Shape optimization in steady blood flow: A numerical study of non-newtonian effects. *Computer Methods in Biomechanics and Biomedical Engineering*, 8:127–137, 2005.

[2] F. Abraham, M. Behr, and M. Heinkenschloss. Shape optimization in unsteady blood flow: A numerical study of non-newtonian effects. *Computer Methods in Biomechanics and Biomedical Engineering*, 8:201–212, 2005.

[3] K. Affeld, J. Reininger, J. Gadischke, K. Grunert, S. Schmidt, and F. Thiele. Fluid mechanics of the stagnation point flow chamber and its platelet deposition. *Artificial Organs*, 19(7):597–602, 1995.

[4] I. Akkerman, Y. Bazilevs, D. J. Benson, M. W. Farthing, and C. E. Kees. Free-surface flow and fluid–object interaction modeling with emphasis on ship hydrodynamics. *Journal of Applied Mechanics*, 79:010905, 2012.

[5] I. Akkerman, Y. Bazilevs, C. E. Kees, and M. W. Farthing. Isogeometric analysis of free-surface flow. *Journal of Computational Physics*, 230:4137–4152, 2011.

[6] I. Akkerman, J. Dunaway, J. Kvandal, J. Spinks, and Y. Bazilevs. Toward free-surface modeling of planing vessels: simulation of the Fridsma hull using ALE-VMS. *Computational Mechanics*, 50:719–727, 2012.

[7] A. Amodeo, G. Brancaccio, G. Michielon, S. Filippelli, Z. Ricci, S. Morelli, M.G. Gagliardi, R. Iacobelli, G. Pongiglione, and R.M. Di Donato. Pneumatic pulsatile ventricular assist device as a bridge to heart transplantation in pediatric patients. *Artificial Organs*, 34(11):1017–1022, November 2010.

[8] J. F. Antaki, O. Ghattas, G. W. Burgreen, and B. He. Computational flow optimization of rotary blood pump components. *Artificial Organs*, 19:608–615, 1995.

[9] F.A. Arabia, P.H. Tsau, R.G. Smith, P.E. Nolan, V. Paramesh, R.K. Bose, D.S. Woolley, G.K. Sethi, B.E. Rhenman, and J. Copeland. Pediatric bridge

to heart transplantation: Application of the berlin heart, medos, and the thoratec ventricular assist devices. *Journal of Heart and Lung Transplantation*, 25(1):16–21, January 2006.

[10] C. Audet and J.E. Dennis Jr. Mesh adaptive direct search algorithms for constrained optimization. *SIAM Journal on Optimization*, 17(1):2–11, 2006.

[11] R. Balossino, G Pennati, F. Migliavacca, L. Formaggia, A. Veneziani, M. Tuveri, and G. Dubini. Computational models to predict stenosis growth in carotid arteries: Which is the role of boundary conditions? *Computational Methods in Biomechanics and Biomedical Engineering*, 12(1):113–123, 2009.

[12] A. Barlow, D. Burton, and M. Shashkov. Compatible, energy and symmetry preserving 2D Lagrangian hydrodynamics in rz-cylindrical coordinates. *Procedia Computer Science*, 1:1893–1901, 2012.

[13] Y. Bazilevs and I. Akkerman. Large eddy simulation of turbulent Taylor–Couette flow using isogeometric analysis and the residual–based variational multiscale method. *Journal of Computational Physics*, 229:3402–3414, 2010.

[14] Y. Bazilevs, I. Akkerman, D.J. Benson, G. Scovazzi, and M.J. Shashkov. Isogeometric analysis of Lagrangian hydrodynamics. *Journal of Computational Physics*, 2013. Published online. DOI: http://dx.doi.org/10.1016/j.jcp.2013.02.021.

[15] Y. Bazilevs, V. M. Calo, J. A. Cottrel, T. J. R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197:173–201, 2007.

[16] Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and Y. Zhang. Isogeometric fluid–structure interaction: theory, algorithms, and computations. *Computational Mechanics*, 43:3–37, 2008.

[17] Y. Bazilevs, V.M. Calo, J.A. Cottrell, J.A. Evans, T.J.R. Hughes, S. Lipton, M.A. Scott, and T.W. Sederberg. Isogeometric analysis using T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199:229–263, 2010.

[18] Y. Bazilevs, V.M. Calo, Y. Zhang, and T.J.R. Hughes. Isogeometric fluid-structure interaction analysis with applications to arterial blobczh06,od flow. *Computational Mechanics*, 38:310–322, 2006.

[19] Y. Bazilevs, M.-C. Hsu, D. Benson, S. Sankaran, and A.-L. Marsden. Computational fluid-structure interaction: Methods and application to a total cavopulmonary connection. *Computational Mechanics*, 45:77–89, 2009.

[20] Y. Bazilevs, M.-C. Hsu, J. Kiendl, and D. J. Benson. A computational procedure for prebending of wind turbine blades. 89:323–336, 2012.

[21] Y. Bazilevs, M.-C. Hsu, and M. A. Scott. Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines. 2012. doi:10.1016/j.cma.2012.03.028.

[22] Y. Bazilevs, M.-C. Hsu, Y. Zhang, W. Wang, T. Kvamsdal, S. Hentschel, and J. Isaksen. Computational vascular fluid-structure interaction: Methodology and application to cerebral aneurysms. *Biomechanics and Modeling in Mechanobiology*, 9:481–498, 2010.

[23] Y. Bazilevs, Ming-Chen Hsu, J. Kiendl, R. Wüchner, and Kai-Uwe Bletzinger. 3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades. *International Journal for Numerical Methods in Fluids*, 65:236–253, 2011.

[24] Y. Bazilevs, Ming-Chen Hsu, K. Takizawa, and T. E. Tezduyar. ALE-VMS and ST-VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction. *Mathematical Models and Methods in Applied Sciences*, 22(supp02):1230002, 2012.

[25] Y. Bazilevs and T. J. R. Hughes. Weak imposition of Dirichlet boundary conditions in fluid mechanics. *Computers and Fluids*, 36:12–26, 2007.

[26] Y. Bazilevs, C. Michler, V. M. Calo, and T. J. R. Hughes. Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes. *Computer Methods in Applied Mechanics and Engineering*, 199:780–790, 2010.

[27] Y. Bazilevs, C. Michler, V.M. Calo, and T.J.R. Hughes. Weak Dirichlet boundary conditions for wall-bounded turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 196:4853–4862, 2007.

[28] Y. Bazilevs, K. Takizawa, and T.E. Tezduyar. *Computational Fluid–Structure Interaction: Methods and Applications*. Wiley, Chichester, 2013.

[29] T. Belytschko, W. K. Liu, and B. Moran. *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2000.

[30] D. J. Benson, Y. Bazilevs, E. De Luycker, M.-C. Hsu, M. Scott, T. J. R. Hughes, and T. Belytschko. A generalized finite element formulation for arbitrary basis functions: from isogeometric analysis to XFEM. 83:765–785, 2010.

[31] D. J. Benson, Y. Bazilevs, M.-C. Hsu, and T. J. R. Hughes. Isogeometric shell analysis: The Reissner–Mindlin shell. 199:276–289, 2010.

[32] D. J. Benson, Y. Bazilevs, M.-C. Hsu, and T. J. R. Hughes. A large deformation, rotation-free, isogeometric shell. 200:1367–1378, 2011.

[33] D.J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering*, 99:235–394, 1992.

[34] D. Bluestein, L. Niu, R. Schoephoerster, and M. Dewanjee. Fluid mechanics of arterial stenosis: Relationship to the development of mural thrombus. *Annals of Biomedical Engineering*, 25:344–356, 1997.

[35] A. J. Booker, J. E. Dennis, Jr., P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset. A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization*, 17(1):1–13, 1999.

[36] E. L. Bove, M. R. de Leval, F. Migliavacca, R. Balossino, and G. Dubini. Toward optimal hemodynamics: computer modeling of the Fontan circuit. *Pediatr. Cardiol.*, 28:477–481, 2007.

[37] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32:199–259, 1982.

[38] G. W. Burgreen, J. F. Antaki, Z. J. Wu, and A. J. Holmes. Computational fluid dynamics as a development tool for rotary blood pumps. *Artificial Organs*, 25:336–340, 2001.

[39] E.J. Caramana, M.J. Shashkov, and P.P. Whalen. Formulations of artificial viscosity for multi-dimensional shock wave computations. *Journal of Computational Physics*, 144:70–97, 1998.

[40] E.J. Caramana and P.P. Whalen. Numerical preservation of symmetry properties of continuum problems. *Journal of Computational Physics*, 141:174–198, 1998.

[41] J.L. Cavanaugh, S.D. Miyamoto, E. da Cruz, B.A. Pietra, D.N Campbell, M.B. Mitchell, C.E. Peyton, J. Gruenwald, and J.R. Darst. Predicting recovery: Successful explant of a ventricular assist device in a child with dilated cardiomyopathy. *Journal of Heart and Lung Transplantation*, 29(1):105–108, January 2010.

[42] J. Chung and G. M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-α method. *Journal of Applied Mechanics*, 60:371–75, 1993.

[43] F. Cirak and M. Ortiz. Fully C^1-conforming subdivision elements for finite deformation thin shell analysis. *International Journal of Numerical Methods in Engineering*, 51:813–833, 2001.

[44] F. Cirak, M. Ortiz, and P. Schröder. Subdivision surfaces: a new paradigm for thin shell analysis. *International Journal of Numerical Methods in Engineering*, 47:2039–2072, 2000.

[45] F. Cirak, M. J. Scott, E. K. Antonsson, M. Ortiz, and P. Schröder. Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision. *Computer-Aided Design*, 34:137–148, 2002.

[46] JB Clark, LB Pauliks, JL Myers, and A Undar. Mechanical circulatory support for end-stage heart failure in repaired and palliated congenital heart disease. *Curr Cardiol Rev.*, 7(2):102–9, 2011.

[47] S.V. Coggeshall and J. Meyer-ter-Vehn. Group-invariant solutions and optimal systems for multidimensional hydrodynamics. *Journal of Mathematical Physics*, 33:3585–3601, 1992.

[48] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, Chichester, 2009.

[49] J.A. Cottrell, T.J.R. Hughes, and A. Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer Methods in Applied Mechanics and Engineering*, 196:4160–4183, 2007.

[50] M.G. Cox. The numerical evaluation of B-splines. Technical report, National Physics Laboratory DNAC 4, 1971.

[51] L.P. Dasi, K. Pekkan, H.D. Katajima, and A.P. Yoganathan. Functional analysis of Fontan energy dissipation. *Journal of Biomechanics*, 41(10):2246–2252, 2008.

[52] T. David, S. Thomas, and P.G. Walker. Platelet seposition in stagnation point flow: An analytical and computational simulation. *Medical Engineering and Physics*, 23(5):299–312, 2001.

[53] M.R. Dörfel, B. Jüttler, and B. Simeon. Adaptive isogeometric analysis by local h-refinement with T-splines. *Computer Methods in Applied Mechanics and Engineering*, 199:264–275, 2010.

[54] C. de Boor. On calculation with B-splines. *Journal of Approximation Theory*, 6:50–62, 1972.

[55] C.G. DeGroff. Modeling the Fontan circulation: Where we are and where we need to go. *Pediatric Cardiology*, 29(1):3–12, January 2008.

[56] C.G. DeGroff, B. Birnbaum, R. Shandas, W. Orlando, and J. Hertzberg. Computational simulations of the total cavo-pulmonary connection: Insights in optimizing numerical solutions. *Medical Engineering and Physics*, 27(2):135–146, March 2005.

[57] W.G. Dettmer and D. Peric. On the coupling between fluid flow and mesh motion in the modelling of fluid–structure interaction. *Computational Mechanics*, 195:5754–5779, 2008.

[58] V.A. Dobrev, T.E. Ellis, T.V. Kolev, and R.N. Rieben. Curvilinear finite elements for Lagrangian hydrodynamics. *International Journal for Numerical Methods in Fluids*, 65:1295–1310, 2011.

[59] V.A. Dobrev, T.E. Ellis, Tz.V. Kolev, and R.N. Rieben. High-order curvilinear finite elements for axisymmetric Lagrangian hydrodynamics. *Computer and Fluids*, 197, 2012. Published online. DOI: http://dx.doi.org/10.1016/j.compfluid.2012.06.004.

[60] V.A. Dobrev, T.V. Kolev, and R.N. Rieben. High-order curvilinear finite element methods for Lagrangian hydrodynamics. *SIAM Journal on Scientific Computing*, 34:B606–B641, 2012.

[61] T.L. Haut Donahue, W. Dehlin, J. Gillespie, W.J. Weiss, and G. Rosenberg. Finite element analysis of stresses developed in the blood sac of a left ventricular assist device. *Medical Engineering and Physics*, 31:454–460, 2009.

[62] M.G. Doyle, J.-B. Vergniaud, S. Tavoularis, and Y. Bourgault. Numerical simulations of blood flow in artificial and natural hearts with fluid-structure interaction. *Artificial Organs*, 32(11):870–879, November 2008.

[63] G. Dubini, M.R. de Leval, R. Pietrabissa, F.M. Montevecchi, and R. Fumero. A numerical fluid mechanical study of repaired congenital heart defects: application to the total cavopulmonary connection. *J. Biomech.*, 29:111–121, 1996.

[64] M. Esmaily-Moghadam, T.-Y. Hsia, and A.L. Marsden. A non-discrete method for computation of residence time in fluid mechanics simulations. *Physics of Fluids*, (In Press).

[65] M. Esmaily-Moghadam, F. Migliavacca, I.E. Vignon-Clementel, T.Y. Hsia, and A.L. Marsden. Optimization of shunt placement for the norwood surgery using multi-domain modeling. *Journal of Biomedical Engineering*, 134(5), 2012.

[66] C. Farhat, M. Lesoinne, and P. Le Tallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete

interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. 157:95–114, 1998.

[67] G.E. Farin. *NURBS Curves and Surfaces: From Projective Geometry to Practical Use.* A. K. Peters, Ltd., Natick, MA, 1995.

[68] M.A. Fernandez and M. Moubachir. A Newton method using exact jacobians for solving fluid-structure coupling. *Computers and Structures*, 83:127–142, 2005.

[69] N. Filipovic, M. Kojic, and A. Tsuda. Modeling thrombosis using dissipative particle dynamics method. *Philosophical Transactions of the Royal Society A*, 366(1879):3265–3279, 2008.

[70] S. Galera, P.-H. Maire, and J. Breil. A two–dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction. *Journal of Computational Physics*, 229:5755–5787, 2010.

[71] R Hetzer, E.V. Potapov, B. Stiller, et al. Improvement in survival after mechanical circulatory support with pneumatic ventricula assist devices in pediatric patients. *Annals of Thoracic Surgery*, (82):917–925, 2006.

[72] V.E. Hjortdal, K. Emmertsen, E. Stenbøg, T. Fründ, M. Rahbek Schmidt, O. Kromann, K. Sørensen, and E.M. Pedersen. Effects of exercise and respiration on blood flow in total cavopulmonary connection: A real timemagnetic resonance flow study. *Circulation*, 108:1227, 2003.

[73] KK Ho, KM Anderson, WB Kannel, W Grossman, and D Levy. Survival after the onset of congestive heart failure in framingham heart study subjects. *Circulation*, 88:107115, 1993.

[74] G.A. Holzapfel, T.C. Gasser, and R.W. Ogden. A new constitutive framework for arterial wall mechanics and a comparative study of material models. *Journal of Elasticity*, 61:1–48, 2000.

[75] T.-Y. Hsia, S. Khambadkone, A.N. Redington, F. Migliavacca, J.E. Deanfield, and M.R. de Leval. Effects of respiration and gravity on infradiaphragmatic venous flow in normal and Fontan patients. *Circulation*, 102, 2000.

[76] M.-C. Hsu, I. Akkerman, and Y. Bazilevs. Finite element simulation of wind turbine aerodynamics: Validation study using NREL Phase VI experiment. *Wind Energy*, 2012. In review.

[77] M.-C. Hsu, I. Akkerman, and Y. Bazilevs. Wind turbine aerodynamics using ALE–VMS: Validation and the role of weakly enforced boundary conditions. 2012. doi:10.1007/s00466-012-0686-x.

[78] M.-C. Hsu and Y. Bazilevs. Fluid–structure interaction modeling of wind turbines: simulating the full machine. *Computational Mechanics*, 50:821–833, 2012.

[79] M.-C. Hsu, Y. Bazilevs, V. M. Calo, T. E. Tezduyar, and T. J. R. Hughes. Improving stability of stabilized and multiscale formulations in flow simulations at small time steps. 199:828–840, 2010.

[80] T. J. R. Hughes. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles, and the origins of stabilized methods. *Computer Methods in Applied Mechanics and Engineering*, 127:387–401, 1995.

[81] T. J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, Mineola, NY, 2000.

[82] T. J. R. Hughes, L. P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška–Brezzi condition: A stable Petrov–Galerkin formulation of the Stokes problem accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering*, 59:85–99, 1986.

[83] T. J. R. Hughes, W. K. Liu, and T. K. Zimmermann. Lagrangian–Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29:329–349, 1981.

[84] T. J. R. Hughes and G. Sangalli. Variational multiscale analysis: the fine-scale Green's function, projection, optimization, localization, and stabilized methods. *SIAM Journal of Numerical Analysis*, 45:539–557, 2007.

[85] T. J. R. Hughes and T. E. Tezduyar. Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. *Computer Methods in Applied Mechanics and Engineering*, 45:217–284, 1984.

[86] T.J. Hughes, M. Mallet, and A. Mizukami. A new finite element formulation for computational fluid dynamics: Beyond supg. *CMAME*, 54(3):341–355, March 1986.

[87] T.J.R. Hughes, J.A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry, and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194:4135–4195, 2005.

[88] T.J.R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199:301–313, 2010.

[89] T.J.R. Hughes, G. Scovazzi, and L.P. Franca. Multiscale and stabilized methods. In E. Stein, R. de Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics, Vol. 3, Computational Fluid Dynamics*, chapter 2. Wiley, 2004.

[90] K. E. Jansen, C. H. Whiting, and G. M. Hulbert. A generalized-α method for integrating the filtered Navier-Stokes equations with a stabilized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190:305–319, 1999.

[91] J.C. Greenfield Jr. and D.M. Griggs Jr. Relation between pressure and diameter in main pulmonary artery of man. *J. Appl. Physiol.*, 18(3):557–559, 1963.

[92] A. A. Johnson and T. E. Tezduyar. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *CMAME*, 119:73–94, 1994.

[93] A. A. Johnson and T. E. Tezduyar. Simulation of multiple spheres falling in a liquid-filled tube. *Computer Methods in Applied Mechanics and Engineering*, 134:351–373, 1996.

[94] C. Johnson and A. Szepessy. On the convergence of a finite element method for a nonlinear hyperbolic conservation law. *Mathematics of Computation*, 49(180):427–444, October 1987.

[95] V.M. Joshi, A. Carey, P. Simpson, and S.M. Paridon. Exercise performance following repair of hypoplastic left heart syndrome: Comparison with other types of Fontan patients. *Pediatric Cardiology*, 18:357–360, 1997.

[96] H. Justino, L.N. Benson, and R. M. Freedom. Development of unilateral pulmonary arteriovenous malformations due to unequal distribution of hepatic venous flow. *Circulation*, 103:E39–E40, 2001.

[97] Y. Khunatorn, S. Mahalingam, C.G. DeGroff, and R. Shandas. Influence of connection geometry and SVC-IVC flow rate ratio on flow structures within the total cavopulmonary connection: A numerical study. *J. Biomech. Eng.*, 124(4):364–378, August 2002.

[98] J. Kiendl, Y. Bazilevs, Ming-Chen Hsu, R. Wüchner, and Kai-Uwe Bletzinger. The bending strip method for isogeometric analysis of Kirchhoff–Love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering*, 199:2403–2416, 2010.

[99] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with Kirchhoff–Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198:3902–3914, 2009.

[100] C.S. Konig, C. Clark, and M.R. Mokhtarzadeh-Dehghan. Investigation of unsteady flow in a model of a ventricular assist device by numerical modeling and comparison with experiment. *Mechanical Engineering and Physics*, 21(1):53–64, February 1999.

[101] M. H. Kroll, J.D. Hellums, L.V. McIntire, A.I. Schafer, and J.L. Moake. Platelets and shear stress. *Blood*, 88:1525–1541, 1996.

[102] U. Kuttler, C. Forster, and W. A. Wall. A solution for the incompressibility dilemma in partitioned fluid–structure interaction with pure Dirichlet fluid domains. *Computational Mechanics*, 38:417–429, 2006.

[103] K. Lagana, G. Dubini, F. Migliavacca, R. Pietrabissa, G. Pennati, A. Veneziani, and A. Quarteroni. Multiscale modeling as a tool to prescribe realistic boundary conditions for the study of surgical procedures. *Biorheology*, 39(3-4):359–364, 2002.

[104] M. Lesoinne and C. Farhat. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *CMAME*, 134(1-2):71–90, July 1996.

[105] D Lloyd-Jones, RJ Adams, TM Brown, M Carnethon, S Dai, G De Simone, TB Ferguson, E Ford, K Furie, C Gillespie, A Go, K Greenlund, N Haase, S Hailpern, PM Ho, V Howard, B Kissela, S Kittner, D Lackland, L Lisabeth, A Marelli, MM McDermott, J Meigs, D Mozaffarian, M Mussolino, G Nichol, VL Roger, W Rosamond, R Sacco, P Sorlie, R Stafford, T Thom, S Wasserthiel-Smoller, ND Wong, and J. Wylie-Rosett. American heart association statistics committee and stroke statistics subcommittee. executive summary: heart disease and stroke statistics2010 update: a report from the american heart association. *Circulation*, 121:948954, 2010.

[106] C. C. Long, M.-C. Hsu, Y. Bazilevs, J. A. Feinstein, and A. L. Marsden. Fluid–structure interaction simulations of the Fontan procedure using variable wall properties. *International Journal for Numerical Methods in Biomedical Engineering*, 28:512–527, 2012.

[107] P.W. Longest and C. Kleinstreuer. Particle-hemodynamics modeling of the distal end-to-side femoral bypass: Effects of graft caliber and graft-end cut. *Medical Engineering & Physics*, 25:843–858, 2003.

[108] L. Margolin and M. Shashkov. Using a curvilinear grid to construct symmetry-preserving discretizations for Lagrangian gas dynamics. *Journal of Computational Physics*, 149:389–417, 1999.

[109] L.G. Margoloin, M.J. Shashkov, and M. A. Taylor. Symmetry-preserving discretizations for Lagrangian gas dynamics. In P. Neittaanmaki, Y. Tiihonen,

and P. Tarvainen, editors, *Proceedings of the 3rd European Conference on Numerical Mathematics and Advanced Applications, ENUMATH 99*, pages 725–732. Word Scientific, 2000.

[110] B.S. Marino. Outcomes after the Fontan procedure. *Curr. Opin. Pediatr.*, 14:620–626, 2002.

[111] A. L. Marsden, A. D. Bernstein, V. M. Reddy, S. Shadden, R. Spilker, F. P. Chan, C. A. Taylor, and J. A. Feinstein. Evaluation of a novel Y-shaped extracardiac fontan baffle using computational fluid dynamics. *Journal of Thoracic and Cardiovascular Surgery*, 2008. To appear.

[112] A.-L. Marsden, V.M. Reddy, F.P. Chan, C.A. Taylor, and J.A. Feinstein. A new multi-parameter approach to computational simulation for Fontan assessment and redesign. *Congenital Heart Disease*, 5(2):104–117, 2010.

[113] A.-L. Marsden, I.-E. Vignon-Clementel, F.P. Chan, J.A. Feinstein, and C.A. Taylor. Effects of exercise and respiration on hemodynamic efficiency in CFD simulations of the total cavopulmonary connection. *Annals of Biomedical Engineering*, 35(2):250–263, 2007.

[114] A. L. Marsden, M. Wang, J. E. Dennis, Jr., and P. Moin. Optimal aeroacoustic shape design using the surrogate management framework. *Optimization and Engineering*, 5(2):235–262, 2004. Special Issue: Surrogate Optimization.

[115] A.L. Marsden, J.A. Feinstein, and C.A. Taylor. A computational framework for derivative-free optimization of cardiovascular geometries. *Computer Methods in Applied Mechanics and Engineering*, 197:1890–1905, 2008.

[116] A.L. Marsden, M. Wang, J.E. Dennis Jr., and P. Moin. Trailing-edge noise reduction using derivative-free optimization and large-eddy simulation. *Journal of Fluid Mechanics*, 572:13–36, 2007.

[117] J.C. Masters, M. Ketner, M.S. Bleiweis, M. Mill, A. Yoganathan, and C.L. Lewis. The effect of incorporating vessel compliance in a computational model of blood flow in a total cavopulmonary connection (tcpc) with caval centerline offset. *Journal of Biomechanical Engineering*, 126:709–713, 2004.

[118] F. Migliavacca, M. R. de Leval, G. Dubini, R. Pietrabissa, and R. Fumero. Computational fluid dynamic simulations of cavopulmonary connections with an extracardiac lateral conduit. *Med. Eng. Phys.*, 21:187–193, 1999.

[119] LW Miller. Left ventricular assist devices are underutilized. *Circulation*, 123:15528, 2011.

[120] M.E. Moghadam, Y. Bazilevs, T.-Y. Hsia, I.E. Vignon-Clementel, and A.-L. Marsden. A comparison of outlet boundary treatments for prevention of backflow divergence with relevance to blood flow simulations. *Computational Mechanics*, 48(3):277–291, 2011.

[121] DL Morales, KS Gunter, and CD Fraser. Thirty years of development and application of CFD at Boeing Commercial Airplanes, Seattle. *COMPUTERS & FLUIDS*, 34(10):1115–1151, 2005.

[122] W.S. Nesbitt, E. Westein, F.J. Tovar-Lopez, et al. A shear gradient-dependent platelet aggregation mechanism drives thrombus formation. *Nature Medicine*, 15, 2009.

[123] J. Von Neumann and R.D. Richtmyer. A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics*, 21:232–237, 1950.

[124] N. Nguyen-Thanh, J. Kiendl, H. Nguyen-Xuan, R. Wüchner, K.U. Bletzinger, Y. Bazilevs, and T. Rabczuk. Rotation-free isogeometric thin shell analysis using PHT-splines. 200:3410–3424, 2011.

[125] W. F. Noh. Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux. *Journal of Computational Physics*, 72:78–120, 1987.

[126] S. Nordbeck and B. Rystedt. Computer cartography point-in-polygon programs. *BIT Numerical Mathematics*, 7:39–64, 1967.

[127] W. Orlando, R. Shandas, and C.G. DeGroff. Efficiency differences in computational simulations of the total cavo-pulmonary circulation with and without compliant vessel walls. *Computer Methods and Programs in Biomedicine*, 81(3):220–227, March 2006.

[128] E. Oñate and F. G. Flores. Advances in the formulation of the rotation-free basic shell triangle. 194:2406–2443, 2005.

[129] E. Oñate and F. Zarate. Rotation-free triangular plate and shell elements. *International Journal of Numerical Methods in Engineering*, 47:557–603, 2000.

[130] K. Pekkan, B. Whited, K. Kanter, S. Sharma, D. de Zelicourt, K. Sundareswaran, D. Frakes, J. Rossignac, and A.P. Yoganathan. Patient-specific surgical planning and hemodynamic computational fluid dynamics optimization through free-form haptic anatomy editing tool (surgem). *Medical and Biological Engineering and Computing*, 46(11):1139–1152, 2008.

[131] G. Pennati, C. Corsini, D. Cosentino, T.-Y. Hsia, V.S. Luisi, G. Dubini, and F. Migliavacca. Boundary conditions of patient-specific fluid dynamics modelling of cavopulmonary connections: Possible adaptation of pulmonary resistances results in a critical issue for a virtual surgical planning. *Interface Focus*, 1:297–307, 2011.

[132] L. Piegl and W. Tiller. *The NURBS Book (Monographs in Visual Communication), 2nd ed.* Springer-Verlag, New York, 1997.

[133] N. A. Pike, L. A. Vricella, J. A. Feinstein, M. D. Black, and B. A. Reitz. Regression of severe pulmonary arteriovenous malformations after Fontan revision and hepatic factor rerouting. *Ann. Thorac. Surg.*, 78:697–9, 2004.

[134] A. Reininger, C. Reininger, U. Heinzmann, and L. Wurzinger. Residence time in niches of stagnant flow determines fibrin clot formation in arterial branching model–detailed flow analysis and experimental results. *Thrombosis and Haemostasis*, 74:916–922, 1995.

[135] D.F. Rogers. *An Introduction to NURBS With Historical Perspective.* Academic Press, San Diego, CA, 2001.

[136] B.N. Roszelle, S. Deutsch, W.J. Weiss, and K.B. Manning. Flow visualization of a pediatric ventricular assist device during stroke volume reductions related to weaning. *Journal of Biomechanical Engineering*, 39(7):2046–2058, 2011.

[137] Y. Saad. *Iterative Methods for Sparse Linear Systems.* PWS Pub. Co., Albany, NY, 1996.

[138] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing*, 7:856–869, 1986.

[139] S. Sankaran, C. Audet, and A.L. Marsden. A method for stochastic constrained optimization using derivative-free surrogate pattern search and collocation. *Journal of Computational Physics*, 229(12):4664–4682, 2010.

[140] S. Sankaran, M. Esmaily-Moghadam, A.M. Kahn, J. Guccione, E. Tseng, and A.L. Marsden. Patient-specific multiscale modeling of blood flow for coronary artery bypass graft surgery. *Annals of Biomedical Engineering*, 40(1):2228–2242, 2012.

[141] S. Sankaran and A.L. Marsden. The impact of uncertainty on shape optimization of idealized bypass graft models in unsteady flow. *Physics of Fluids*, 22(12):121902, 2010.

[142] S. Sankaran and A.L. Marsden. A stochastic collocation method for uncertainty quantification in cardiovascular simulations. *Journal of Biomechanical Engineering*, 133:031001, 2011.

[143] Jeanette P. Schmidt, Scott L. Delp, Michael A. Sherman, Charles A. Taylor, Vijay S. Pande, and Russ B. Altman. The simbios national center: Systems-biology in motion. *Proceedings of the IEEE, special issue on Computational System Biology*, 96(8):1266–1280, 2008.

[144] San Diego Supercomputing Center (SDSC). http://www.sdsc.edu/. 1999.

[145] T.W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri. T-splines and T-NURCCS. *ACM Transactions on Graphics*, 22(3):477–484, 2003.

[146] D. Sengupta, A.M. Kahn, J.C. Burns, S. Sankaran, S. Shadden, and A.L. Marsden. Image-based modeling of hemodynamics and coronary artery aneurysms caused by kawasaki disease. *Biomechanics and Modeling in Mechanobiology*, 11(6):915–932, 2012.

[147] D. B. Serafini. *A Framework for Managing Models in Nonlinear Optimization of Computationally Expensive Functions*. PhD thesis, Rice University, Houston, TX, 1998.

[148] I.S. Sokolnikoff. *Mathematical Theory of Elasticity*. New York: McGraw-Hill, 1956.

[149] E.N. Sorensen, G.W. Burgreen, W.R. Wagner, and J.F. Antaki. Computational simulation of platelet deposition and activation: I. Model development and properties. *Annals of Biomedical Engineering*, 27:436–448, 1999.

[150] E.N. Sorensen, G.W. Burgreen, W.R. Wagner, and J.F. Antaki. Computational simulation of platelet deposition and activation: II. Results for Poiseuille flow over collagen. *Annals of Biomedical Engineering*, 27:449–458, 1999.

[151] R.L. Spilker, J.A. Feinstein, D.W. Parker, V.M. Reddy, and C.-A. Taylor. Morphometry-based impedance boundary conditions for patient-specific modeling of blood flow in pulmonary arteries. *Annals of Biomedical Engineering*, 35:546–549, 2007.

[152] A.B. Strong, G.D. Stubley, G. Chang, and D.R. Absolom. Theoretical and experimental analysis of cellular adhesion to polymer surfaces. *Journal of Biomedical Materials Research*, 21:1039–1055, 1987.

[153] K. Takizawa, Y. Bazilevs, and T. E. Tezduyar. Space–time and ALE-VMS techniques for patient-specific cardiovascular fluid–structure interaction modeling. *Archives of Computational Methods in Engineering*, 19:171–225, 2012.

[154] K. Takizawa, J. Christopher, T. E. Tezduyar, and S. Sathe. Space–time finite element computation of arterial fluid–structure interactions with patient-specific data. *International Journal for Numerical Methods in Biomedical Engineering*, 26:101–116, 2010.

[155] K. Takizawa, C. Moorman, S. Wright, J. Christopher, and T. E. Tezduyar. Wall shear stress calculations in space–time finite element computation of arterial fluid–structure interactions. *Computational Mechanics*, 46:31–41, 2010.

[156] K. Takizawa and T. E. Tezduyar. Multiscale space–time fluid–structure interaction techniques. *Computational Mechanics*, 48:247–267, 2011.

[157] K. Takizawa and T. E. Tezduyar. Computational methods for parachute fluid–structure interactions. *Archives of Computational Methods in Engineering*, 19:125–169, 2012.

[158] K. Takizawa and T. E. Tezduyar. Space–time fluid–structure interaction methods. *Mathematical Models and Methods in Applied Sciences*, 2012. To appear, DOI: 10.1142/S0218202512300013.

[159] T. Tezduyar, S. Aliabadi, M. Behr, A. Johnson, and S. Mittal. Parallel finite-element computation of 3D flows. *Computer*, 26(10):27–36, 1993.

[160] T. E. Tezduyar. Finite element methods for flow problems with moving boundaries and interfaces. *Archives of Computational Methods in Engineering*, 8:83–130, 2001.

[161] T. E. Tezduyar. Computation of moving boundaries and interfaces and stabilization parameters. *International Journal for Numerical Methods in Fluids*, 43:555–575, 2003.

[162] T. E. Tezduyar. Stabilized finite element methods for computation of flows with moving boundaries and interfaces. In *Lecture Notes on Finite Element Simulation of Flow Problems (Basic - Advanced Course)*, Tokyo, Japan, 2003. Japan Society of Computational Engineering and Sciences.

[163] T. E. Tezduyar. Stabilized finite element methods for flows with moving boundaries and interfaces. *HERMIS: The International Journal of Computer Mathematics and its Applications*, 4:63–88, 2003.

[164] T. E. Tezduyar. Finite element methods for fluid dynamics with moving boundaries and interfaces. In E. Stein, R. De Borst, and T. J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*, Volume 3: Fluids, chapter 17. John Wiley & Sons, 2004.

[165] T. E. Tezduyar. Moving boundaries and interfaces. In L. P. Franca, T. E. Tezduyar, and A. Masud, editors, *Finite Element Methods: 1970's and Beyond*, pages 205–220. CIMNE, Barcelona, Spain, 2004.

[166] T. E. Tezduyar. Finite elements in fluids: Special methods and enhanced solution techniques. *Computers & Fluids*, 36:207–223, 2007.

[167] T. E. Tezduyar, M. Behr, S. Mittal, and A. A. Johnson. Computation of unsteady incompressible flows with the finite element methods – space–time formulations, iterative strategies and massively parallel implementations. In *New Methods in Transient Analysis*, PVP-Vol.246/AMD-Vol.143, pages 7–24, New York, 1992. ASME.

[168] T. E. Tezduyar and Y. Osawa. Finite element stabilization parameters computed from element matrices and vectors. *Computer Methods in Applied Mechanics and Engineering*, 190:411–430, 2000.

[169] T. E. Tezduyar and Y. J. Park. Discontinuity capturing finite element formulations for nonlinear convection-diffusion-reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 59:307–325, 1986.

[170] T. E. Tezduyar and S. Sathe. Modeling of fluid–structure interactions with the space–time finite elements: Solution techniques. *International Journal for Numerical Methods in Fluids*, 54:855–900, 2007.

[171] T. E. Tezduyar, S. Sathe, R. Keedy, and K. Stein. Space–time techniques for finite element computation of flows with moving boundaries and interfaces. In S. Gallegos, I. Herrera, S. Botello, F. Zarate, and G. Ayala, editors, *Proceedings of the III International Congress on Numerical Methods in Engineering and Applied Science*. CD-ROM, Monterrey, Mexico, 2004.

[172] T. E. Tezduyar, S. Sathe, R. Keedy, and K. Stein. Space–time finite element techniques for computation of fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 195:2002–2027, 2006.

[173] T. E. Tezduyar, S. Sathe, and K. Stein. Solution techniques for the fully-discretized equations in computation of fluid–structure interactions with the space–time formulations. *Computer Methods in Applied Mechanics and Engineering*, 195:5743–5753, 2006.

[174] T. E. Tezduyar, K. Takizawa, and J. Christopher. Multiscale Sequentially-Coupled Arterial Fluid–Structure Interaction (SCAFSI) technique. In S. Hartmann, A. Meister, M. Schaefer, and S. Turek, editors, *International Workshop on Fluid–Structure Interaction — Theory, Numerics and Applications*, pages 231–252. Kassel University Press, 2009.

[175] I.-E. Vignon-Clementel, C.A. Figueroa, K.E. Jansen, and C.A. Taylor. Outflow boundary conditions for three-dimensional simulations of non-periodic blood flow and pressure fields in deformable arteries. *Computer Methods in Biomechanics and Biomedical Engineering*, 13(5):625–640, 2010.

[176] D. Wang and J. Xuan. An improved NURBS-based isogeometric analysis with enhanced treatment of essential boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 199:2425–2436, 2010.

[177] K. K. Whitehead, K. Pekkan, H. D. Kitahima, S. M. Paridon, A. P. Yoganathan, and M. A. Fogel. Nonlinear power loss during exercise in single-ventricle patients after the Fontan: insights from computational fluid dynamics. *Circulation*, 116:I–165–I–171, 2007.

[178] P. Wriggers. *Nonlinear Finite Element Methods*. Springer, 2008.

[179] G. Xiong, C.A. Figueroa, N. Xiao, and C.A. Taylor. Simulation of blood flow in deformable vessels using subject-specific geometry and spatially varying wall properties. *International Journal for Numerical Methods in Biomedical Engineering*, 27:1000–1016, 2011.

[180] Zhiliang Xu, Nan Chen, Malgorzata M Kamocka, Elliot D. Rosen, and Mark Alber. A multiscale model of thrombus development. *J. R. Soc. Interface*, 5:705–722, 2008.

[181] Zhiliang Xu, Nan Chen, Shawn C. Shadden, Jerrold E. Marsden, Malgorzata M Kamocka, Elliot D. Rosen, and Mark Alber. Study of blood flow impact on growth of thrombi using a multiscale model. *Soft Matter*, 5(4):769–779, 2009.

[182] W. Yang, J. A. Feinstein, S.C. Shadden, I.E. Vignon-Clementel, and A. L. Marsden. Optimizatin of a y-graft design for improved hepatic flow distribution in the fontan circulation. *J. Biomech. Eng.*, 135(1), 2013.

[183] W. Yang, J.A. Feinstein, and A.L. Marsden. Constrained optimization of an idealized Y-shaped baffle for the Fontan surgery at rest and exercise. *Computer Methods in Applied Mechanics and Engineering*, 199:2135–2149, 2010.