A Re-Engineered Software Interface and Workflow for the Open-Source SimVascular Cardiovascular Modeling Package

Hongzhi Lan
Department of Pediatrics,
Stanford University,
Stanford, CA 94305

Adam Updegrove
Department of Mechanical Engineering,
University of California, Berkeley,
Berkeley, CA 94720

Nathan M. Wilson
Open Source Medical Software Corporation,
Santa Monica, CA 90403

Gabriel D. Maher
ICME,
Stanford University,
Stanford, CA 94305

Shawn C. Shadden
Department of Mechanical Engineering,
University of California, Berkeley,
Berkeley, CA 94720

Alison L. Marsden[1]
Department of Pediatrics,
Stanford University,
Clark Center E100B 318 Campus Drive,
Stanford, CA 94305-5428;
ICME,
Stanford University,
Stanford, CA 94305;
Department of Bioengineering,
Stanford University,
Stanford, CA 94305
e-mail: amarsden@stanford.edu

Patient-specific simulation plays an important role in cardiovascular disease research, diagnosis, surgical planning and medical device design, as well as education in cardiovascular biomechanics. SIMVASCULAR *is an open-source software package encompassing an entire cardiovascular modeling and simulation pipeline from image segmentation, three-dimensional (3D) solid modeling, and mesh generation, to patient-specific simulation and analysis. SimVascular is widely used for cardiovascular basic science and clinical research as well as education, following increased adoption by users and development of a GATEWAY web portal to facilitate educational access. Initial efforts of the project focused on replacing commercial packages with open-source alternatives and adding increased functionality for multiscale modeling, fluid–structure interaction (FSI), and solid modeling operations. In this paper, we introduce a major SimVascular (SV) release that includes a new graphical user interface (GUI) designed to improve user experience. Additional improvements include enhanced data/project management, interactive tools to facilitate user interaction, new boundary condition (BC) functionality, plug-in mechanism to increase modularity, a new 3D segmentation tool, and new computer-aided design (CAD)-based solid modeling capabilities. Here, we focus on major changes to the software platform and outline features added in this new release. We also briefly describe our recent experiences using SimVascular in the classroom for bioengineering education.* [DOI: 10.1115/1.4038751]

Keywords: patient-specific simulation, image-based modeling, CFD, hemodynamics, open-source, finite element method, biomechanics software

1 Introduction

Hemodynamics plays a critical role in the origin, progression, and treatment of cardiovascular disease. Relevant quantities such as blood pressure, velocity, wall shear stress, and vessel wall deformation are used to investigate the influence of mechanical stimuli in cardiovascular disease, and in personalized treatment planning [1–6]. Altered pressure and flow can lead to extensive remodeling of vascular and cardiac morphology and composition. These changes play a key role in the progression of numerous diseases including coronary artery disease, bypass graft failure, cardiomyopathy, pulmonary hypertension, aneurysm growth and rupture, and postoperative remodeling in congenital heart disease [7–12]. Mechanical forces and flow disruption resulting in endothelial injury, flow stagnation, and shear-related platelet activation can play a key role in thrombus formation, elevating the risk of stroke, heart attacks, and embolisms [13–15]. Early studies of hemodynamics played a key role in explaining the localization of atherosclerotic plaques to bifurcation regions, such as in the carotid sinus where flow separation occurs and wall shear stress is relatively low [16]. Since then, the advent of patient-specific image-based modeling together with advances in computational methods has enabled clinical and basic science studies in a wide range of cardiovascular diseases.

Advanced medical imaging techniques such as computed tomography [17], magnetic resonance imaging [18], ultrasonography [19], angiography, and elastography [20] can capture increasingly detailed anatomical structure, and in some cases, blood flow velocity and tissue mechanical properties. With these recent advances, there are increasing opportunities for personalized models to predict postoperative and treatment outcomes, disease progression, and stratify patient risk. Blood flow simulations using medical image-based anatomical models now offer a noninvasive technique to determine accurate flow velocity, wall shear stress, and other clinically relevant quantities [21–29] to augment medical imaging. In addition, they offer a means to quantify mechanical stimuli triggering mechanobiologic responses such as vessel wall remodeling and aneurysm initiation and progression [9,30]. Furthermore, one can efficiently perform multiple simulations while systematically varying geometric and physiologic parameters for surgical planning, medical device design, and optimization [31–37].

The first Food and Drug Administration approved cardiovascular simulation technology, $\mathrm{FFR}_{\mathrm{CT},}$ was recently introduced by HeartFlow, Inc. (Redwood City, CA). $\mathrm{FFR}_{\mathrm{CT}}$ offers a noninvasive means to obtain a measure of fractional flow reserve (FFR), which is the clinical gold standard for evaluating the functional significance of coronary stenosis. Comparisons of $\mathrm{FFR}_{\mathrm{CT}}$ to measurements from cardiac catheterization have been favorable and clinical adoption is on the rise [38]. Reduced-order models and machine learning methods have also shown promise [39–42].

The SimVascular (SV) project originated from an in-house software framework and was publicly released by Charles Taylor's

[1] Corresponding author.

Manuscript received December 5, 2017; final manuscript received December 11, 2017; published online January 25, 2018. Editor: Victor H. Barocas.

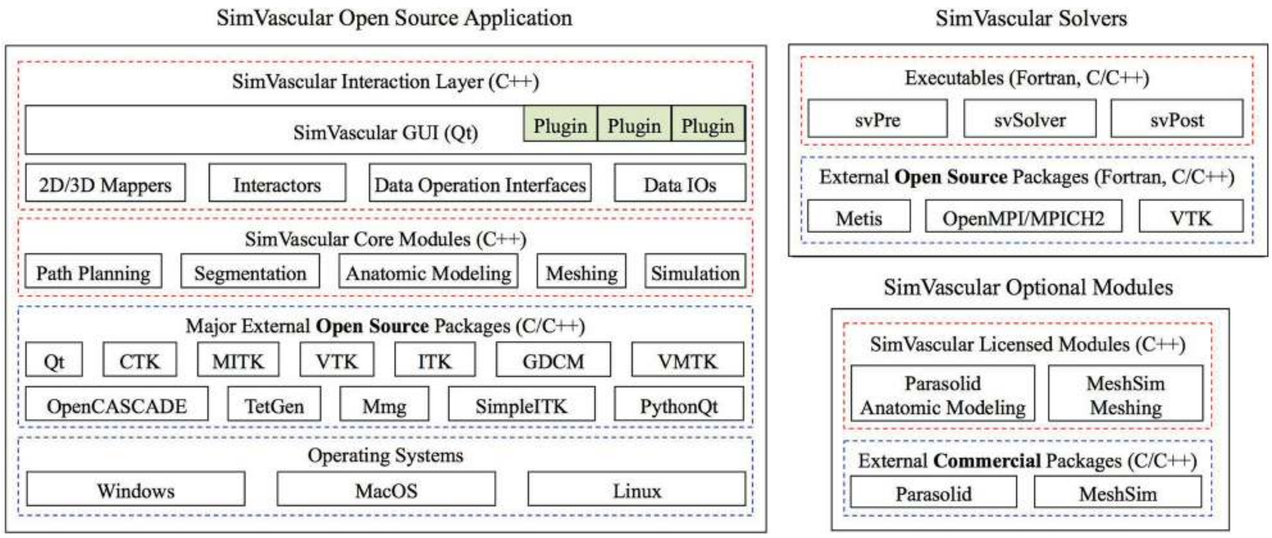

SimVascular Open Source Application
SimVascular Interaction Layer (C++)
SimVascular GUI (Qt)
Plugin
Plugin
Plugin
2D/3D Mappers
Interactors
Data Operation Interfaces
Data IOs
SimVascular Core Modules (C++)
Path Planning
Segmentation
Anatomic Modeling
Meshing
Simulation
Major External Open Source Packages (C/C++)
Qt
CTK
MITK
VTK
ITK
GDCM
VMTK
OpenCASCADE
TetGen
Mmg
SimpleITK
PythonQt
Operating Systems
Windows
MacOS
Linux
SimVascular Solvers
Executables (Fortran, C/C++)
svPre
svSolver
svPost
External Open Source Packages (Fortran, C/C++)
Metis
OpenMPI/MPICH2
VTK
SimVascular Optional Modules
SimVascular Licensed Modules (C++)
Parasolid Anatomic Modeling
MeshSim Meshing
External Commercial Packages (C/C++)
Parasolid
MeshSim

Fig. 1 The multilayered multiple-component architecture of SV comprised of main application, solvers, and optional licensed modules. The main application consists of three layers: interaction layer, core modules, and external packages, supported on three major operating systems.

lab at Stanford in 2007 [43], forming the initial basis for HeartFlow's technology. Nevertheless, at the time it required three commercial components for modeling, meshing, and simulation, and the lack of subsequent development, maintenance and user support after the initial release deterred widespread user adoption. To overcome those obstacles, our team launched the SV revitalization project in 2013, and after implementing open source alternatives for all commercial components, SV was released as a fully open-source software package. In addition, expanded capabilities were added such as discrete modeling, fluid structure interaction (FSI) with variable wall properties, coupled boundary conditions (BCs) with closed-loop lumped parameter networks, cross-platform support, and a unified build system [44]. Related software packages offering alternatives for all or portions of the cardiovascular simulation pipeline include the Vascular Modeling Toolkit Lab (VMTKLab),[2] and the Cardiovascular Integrated Modeling and Simulation (CRIMSON)[3].

Since its initial release, SimVascular has contributed to numerous advances in fundamental disease research, surgical planning, and medical device design [30,45–55]. An accompanying publicly available data repository[4] provides patient-specific image data, models, and simulation results of over 120 clinical cases from different regions of the vasculature, including cerebrovascular, coronary, aortofemoral, pulmonary, and congenital heart disease [33,56,57]. Over the past several years, there has been a sustained effort to overhaul SV and add additional key functionality. In this paper, we describe a new and overhauled software architecture and graphical user interface (GUI), data management system, and new features including three-dimensional (3D) segmentation, graphical assignment of simple boundary conditions, computer-aided design (CAD)-based solid modeling capabilities, and expanded documentation. We focus on changes to features and functionality, referring the reader to a prior overview paper for additional details on underlying methodology [44]. We also describe our recent experience using SV in the classroom for hands-on student projects, and GATEWAY resources to facilitate adoption of these tools for education.

2 Software Architecture

The re-engineered SV platform has a multilayered multiple-component architecture (Fig. 1) composed of three parts: main application, solvers, and optional modules. The main application is fully open source and leverages multiple existing open source packages. The licensed modules are nonessential, and provide legacy support for commercial meshing and solid modeling alternatives.

The main application has multiple layers and runs on all three major operating systems (Windows, Mac OS X, and Linux). Its architecture has two distinct layers for user interaction and data processing. The interaction layer is object-oriented and avoids unnecessary coupling with the data processing layer. The data processing layer is comprised of core modules and is mainly used for defining internal data types and data processing operations.

The interaction layer provides a GUI, developed using the open source package Qt.[5] This layer handles the interaction between the user interface and core functions, enabled by the Medical Imaging Interaction Toolkit (MITK)[6] and employs the Common Toolkit (CTK)[7] plugin framework. With a consistent plugin interface, developers and users can create their own custom plugins. The interaction layer loads the SV and MITK plugins dynamically upon launch. The plugins contain tools for the entire modeling pipeline, including image visualization and processing, two-dimensional (2D)/3D segmentation, modeling, meshing, and simulation. Python scripting to access Qt-based functions is available through PythonQt.[8]

To interface between the GUI and data processing layer, custom classes have been developed for data visualization, input/output (IO), and user interaction. Specialized 2D/3D mappers are incorporated for various data types to enhance visualization in 2D and 3D views. Interactors use extensible markup language (XML)-based scripts to define and control user interaction with data through the GUI. By implementing data operation interfaces, data changes can now be tracked unlike in prior SV versions, enabling a new undo/redo feature to restore data to previous states. The data IOs provide the flexibility to read and write data of various types in multiple formats, such as XML, images, CAD models, and unstructured meshes. SV also supports importing/exporting data in legacy SV file formats to facilitate user migration to the latest version.

Multiple data types, including images, paths, segmentations, geometric models, unstructured meshes, and simulation results are involved in the cardiovascular modeling and simulation pipeline. Furthermore, a number of options and parameters are used for

[2] vmtklab.orobix.com
[3] www.crimson.software
[4] www.vascularmodel.org
[5] www.qt.io
[6] www.mitk.org
[7] www.commontk.org
[8] pythonqt.sourceforge.net

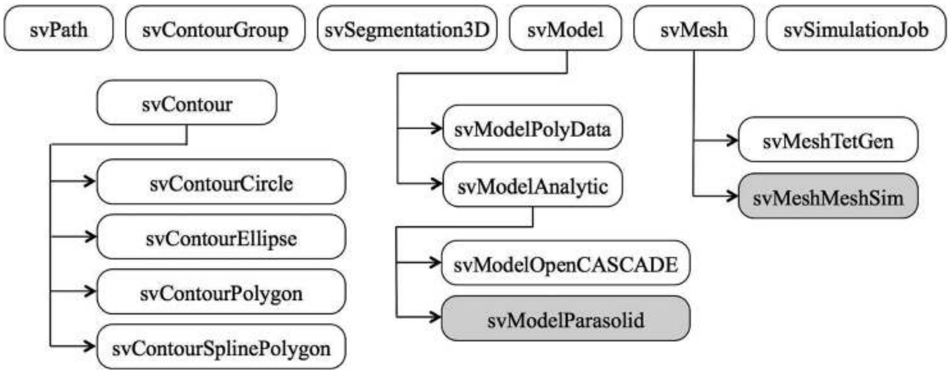


Fig. 2 Data types and their inheritance relationships in SV. The classes in white are used in open source modules, while the classes in gray are for licensed modules.

data creation and manipulation. To deal with data complexity, the core modules define corresponding data types to store different types of data and parameters using six main data classes: svPath, svContourGroup, svSegmentation3D, svModel, svMesh, and svSimulationJob (Fig. 2). The data structure design is critical for modularity and extensibility during development and deployment. The core modules use external open-source packages including: the Visualization Toolkit (VTK)[9] and Vascular Modeling Toolkit (VMTK)[10] for visualization and modeling, Insight Segmentation and Registration Toolkit (ITK)[11] and Grassroots DICOM (GDCM)[12] for image IO and processing, OpenCASCADE[13] for CAD models, Tetrahedral Mesh Generator (TetGen)[14] and Mmg (Mesh Modification and Generation)[15] for surface and volume meshing.

The SV solver uses the finite element method to solve the incompressible, Newtonian form of the Navier–Stokes equations, with FSI based on the coupled momentum method [58–61]. Detailed descriptions of the numerical methodology used in the solver can be found elsewhere [44,62,63]. The solver module includes three independent executable programs: svPre, svSolver, and svPost, which are called as external programs to prepare and run blood flow simulations and postprocess results for analysis. The solvers incorporate several open-source packages including Metis for mesh partition, OpenMPI/MPICH2 for parallel computing, and the SimVascular linear solver (svLS), which uses a bipartition method and custom resistance-based preconditioner [64,65].

As mentioned previously, older versions of SV required multiple commercial libraries for core functionality. The SV licensed modules provide legacy support to optional commercial packages Parasolid (Siemens PLM Software, Plano, TX) and MeshSim (Simmetrix, Inc., Clifton Park, NY).

The SV source code is hosted in two public repositories for the front-end GUI[16] and the solver,[17] providing access for modification and contributions by other developers. Binary installer releases are available at SimTK,[18] together with examples and clinical cases including medical images and data files for user training. Comprehensive documentation for users and developers is available at the website.[19] Users can postquestions and receive technical support through an active user forum hosted by the website.[20]

2.1 Redesigned Graphical User Interface. The new SV GUI provides a data manager, a multiple-view display window, and plugins for data visualization and operations, accessible from the tool bar (see Fig. 3). The data manager uses a data-oriented strategy to organize all the data for a simulation case in a tree-structured project. It creates data nodes for individual objects such as images, paths, contour groups, 3D segmentations, models, meshes, and simulation jobs. Data objects of the same type are placed in the same tree node. As the user manipulates data from the GUI, SV handles most underlying file operations without user intervention. All data for a project can optionally be stored in a directory without dependence on outside files, making the project portable. Upon opening a project, all data are automatically loaded (except the mesh data, which can be large) and listed in the data manager. Through the data manager, users can conduct operations, such as creating new data objects, renaming or removing objects, changing visibility, opacity and color, duplicating objects, and importing and exporting data. As the user works on a data object, SV activates the corresponding tool and updates its interface. This feature lends itself to a data-driven workflow.

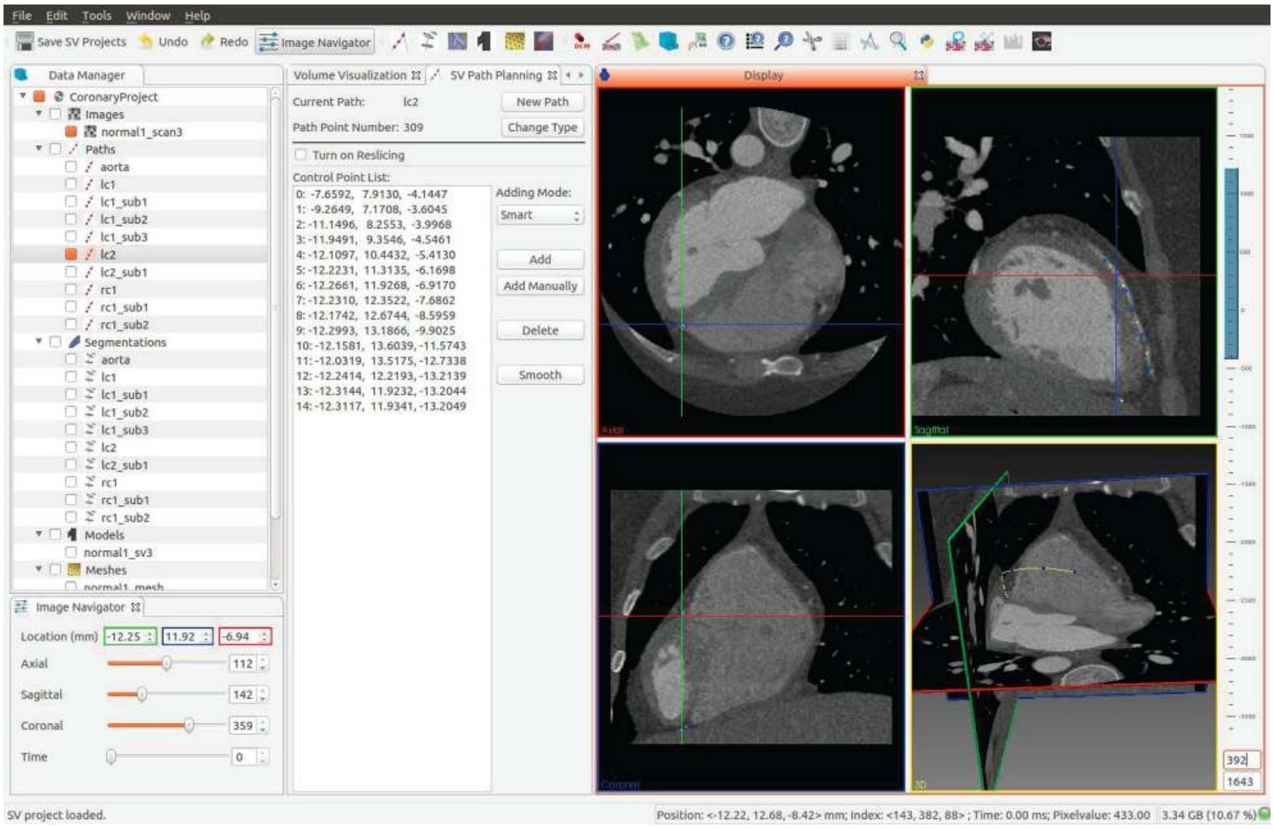
Standard MITK tools have been incorporated, including a display window with three orthographic 2D views and one 3D view in a flexible layout, which facilitates data visualization and manipulation by users. 2D view planes can be in any arbitrary orientation, making it possible for the user to visualize desired parts of the vasculature. With the image navigator tool, users can visualize time-dependent image data. In addition, the GUI provides custom tools for modeling and simulation as described in Sec. 3. Implemented with Qt multithreading, multitasking is available so that users can launch a time-consuming operation in the background and continue to work on data.

3 Cardiovascular Modeling and Simulation Pipeline

Figure 4 displays the typical steps of the image-based modeling pipeline in SimVascular. The pipeline starts with image visualization, after which, paths along vessel centerlines are defined. The typical model creation method involves creating groups of 2D contours of the vessels' lumen at perpendicular cross sections, lofting a surface from the 2D contours for each vessel, and joining individual vessels together to form a complete solid model. In addition, methods to create a solid model using direct 3D segmentation or by importing an existing 3D geometry (e.g., from an STL file) have been added to the pipeline. From the model, an unstructured mesh is generated, initial and boundary conditions are defined, and a blood flow simulation is performed. Updates to the software for each of these steps in the pipeline are described in Secs. 3.1–3.6.

3.1 Image Visualization. Medical image data are typically defined in a 3D structured grid with intensity values for individual voxels, and different organs or tissues within the image have varying intensity values. Incorporating MITK and ITK libraries, the

[9] www.vtk.org
[10] www.vmtk.org
[11] www.itk.org
[12] gdcm.sourceforge.net
[13] www.opencascade.com
[14] wias-berlin.de/software/tetgen
[15] www.mmgtools.org
[16] www.github.com/SimVascular
[17] https://github.com/SimVascular/svSolver
[18] www.simtk.org/projects/simvascular
[19] www.simvascular.org
[20] simtk.org



- CoronaryProject
  - Images
    - normal1_scan3
  - Paths
    - aorta
    - lc1
    - lc1_sub1
    - lc1_sub2
    - lc1_sub3
    - lc2
    - lc2_sub1
    - rc1
    - rc1_sub1
    - rc1_sub2
  - Segmentations
    - aorta
    - lc1
    - lc1_sub1
    - lc1_sub2
    - lc1_sub3
    - lc2
    - lc2_sub1
    - rc1
    - rc1_sub1
    - rc1_sub2
  - Models
    - normal1_sv3
  - Meshes
    - normal1_mesh

Current Path: lc2

Path Point Number: 309

☐ Turn on Reslicing

Control Point List:

```
0: -7.6592,  7.9130, -4.1447
1: -9.2649,  7.1708, -3.6045
2: -11.1496,  8.2553, -3.9968
3: -11.9491,  9.3546, -4.5461
4: -12.1097, 10.4432, -5.4130
5: -12.2231, 11.3135, -6.1698
6: -12.2661, 11.9268, -6.9170
7: -12.2310, 12.3522, -7.6862
8: -12.1742, 12.6744, -8.5959
9: -12.2993, 13.1866, -9.9025
10: -12.1581, 13.6039, -11.5743
11: -12.0319, 13.5175, -12.7338
12: -12.2414, 12.2193, -13.2139
13: -12.3144, 11.9232, -13.2044
14: -12.3117, 11.9341, -13.2049
```

Adding Mode: Smart

Image Navigator

Location (mm) -12.25 11.92 -6.94

Axial 112

Sagittal 142

Coronal 359

Time 0

SV project loaded.

Position: <-12.22, 12.68, -8.42> mm; Index: <143, 382, 88> ; Time: 0.00 ms; Pixelvalue: 433.00 3.34 GB (10.67 %)

Fig. 3 The redesigned SV GUI, containing a data manager, a multiview display window, and multiple plugins for data processing. In this example, a plugin for path planning is activated, providing options for a user to create and modify a path.

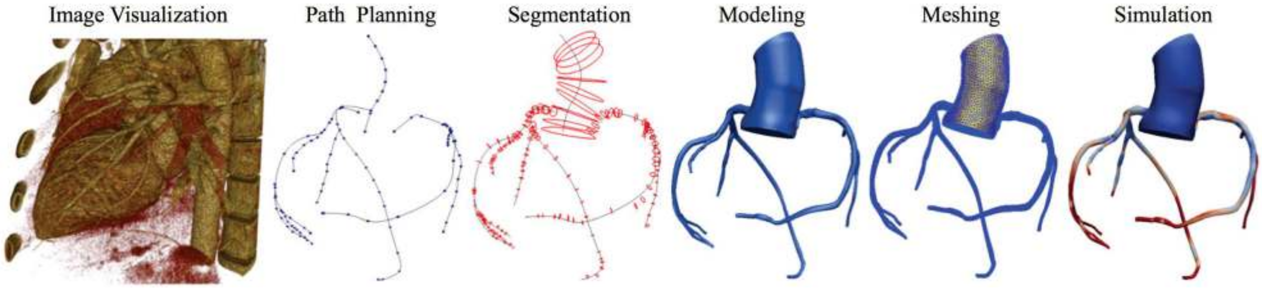


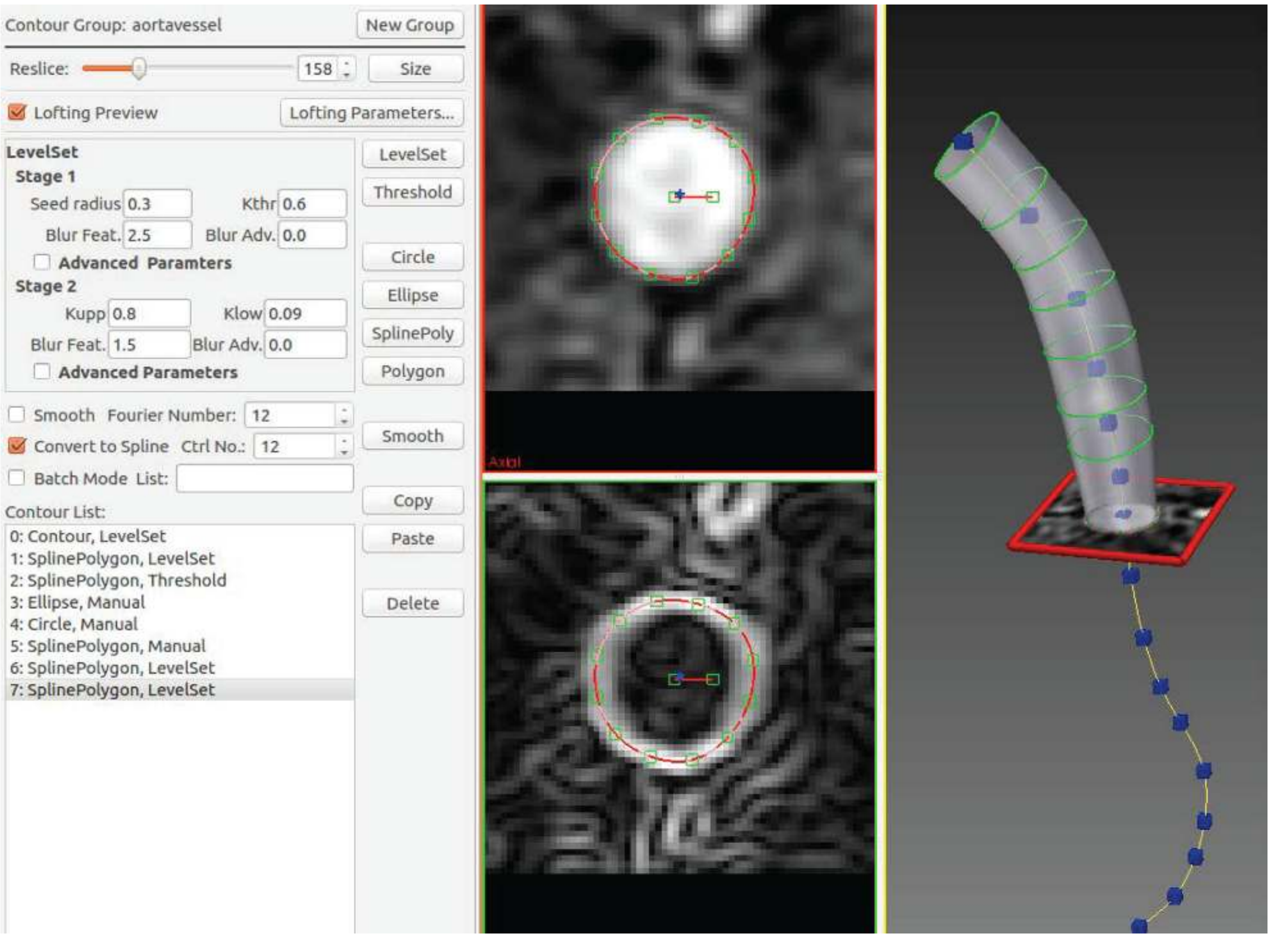
Fig. 4 The SV pipeline encompasses the complete process for cardiovascular modeling and simulation, from image visualization, to anatomic model construction (including path planning and image segmentation) to meshing, BC assignment, blood flow simulations, and analysis

new interface provides improved support for medical images such as magnetic resonance imaging and computed tomography scans. Images in different formats can be loaded, including VTK, MetaImage, raw, NRRD, NIFTI, and DICOM. Basic image processing tools, such as commonly used filters and arithmetic operations, are available. Additional options include image volume rendering, including central processing unit or graphics processing unit-based techniques and common transfer function presets.

3.2 Path Planning. In the typical modeling pipeline (Fig. 4), the first step after image visualization is to create paths along the vessel centerlines. A corresponding data type (svPath, Fig. 2) is defined for paths. It uses a small number of control points to calculate an interpolating spline and resamples to a larger number of spline points (called path points) to describe the path. When a user works on a path, the Path Planning tool (Fig. 3) is activated, providing several options for the user to create and modify a path.

To create a new path for a vessel, the user can take advantage of the three 2D view windows simultaneously to locate center points for the vessel lumen and add them to the path as control points. There are five modes available for adding a control point to the path. In the new "smart" mode, SV calculates the distance and orientation of the new control point relative to existing control points and determines the correct insertion order along the path. However, in some cases, such as highly tortuous vessels, users may also decide the appropriate point ordering. Based on the control points, there are three ways to calculate path points, with different methods to resample path points from the spline: (1) constant spacing, such as the minimum image spacing, (2) user-specified values for the total number of path points, or (3) user-specified values for the number of path points between two adjacent control points.

After a path is created, the user can modify the path by manipulating the control points, moving and deleting them interactively in 2D/3D view windows, or applying Fourier smoothing with a user-defined subsampling interval. Users can reslice the image



Contour Group: aortavessel

Reslice: 158

☑ Lofting Preview

LevelSet

Stage 1

Seed radius 0.3 Kthr 0.6

Blur Feat. 2.5 Blur Adv. 0.0

□ Advanced Paramters

Stage 2

Kupp 0.8 Klow 0.09

Blur Feat. 1.5 Blur Adv. 0.0

□ Advanced Parameters

□ Smooth Fourier Number: 12

☑ Convert to Spline Ctrl No.: 12

□ Batch Mode List:

Contour List:

0: Contour, LevelSet
1: SplinePolygon, LevelSet
2: SplinePolygon, Threshold
3: Ellipse, Manual
4: Circle, Manual
5: SplinePolygon, Manual
6: SplinePolygon, LevelSet
7: SplinePolygon, LevelSet

Fig. 5 SV provides several methods to create and modify contours, loft solid models of vessels, and interactively preview the model during construction as new segmentations are added

perpendicular to the current path, enabling them to check that the path remains in the center of the vessel. A quality path is important for subsequent 2D segmentation along the path.

3.3 Image Segmentation. To create an anatomic model based on medical image data, the geometry of a region of interest is reconstructed through image segmentation. Extensive studies have been carried out in the field of medical image segmentation [66,67] and SimVascular exploits established techniques to conduct 2D and 3D image segmentation of the vasculature of interest.

3.3.1 2D Segmentation. The data type svContourGroup is defined to store and manage all the contours for a vessel, with five data types defined for different contour types, including svContour, svContourCircle, svContourEllipse, svContourPolygon, svContourSplinePolygon (Fig. 2). When a user works on a contour group, the SV 2D Segmentation tool (Fig. 5) is activated and provides several methods and options to create and modify 2D contours and preview the lofted surface for a vessel. Users can now modify the corresponding path after a contour group has been created, offering increased editing flexibility compared to prior versions. This is done by automatically fetching the original path points of existing contours and inserting them into the modified path in the correct order, based on relative distance and orientation, to create a new temporary path for image reslicing.

During 2D segmentation, the display changes automatically to three windows. The two left 2D-view reslice windows show the image intensity and magnitude of the gradient on a plane perpendicular to the vessel path. The right 3D-view window shows the path, the contour groups, and an image reslice probe that travels along the path. In either reslice window, a user can create a 2D contour using (1) automated segmentation based on standard level set and thresholding methods [44], (2) analytical methods to create a circle, ellipse, or spline polygon interactively, or (3) by manually drawing the outline of the vessel lumen. The new version enables the user to see the contour change interactively as the user varies the threshold value. Similar to a path, a contour has multiple control points available for the user to translate or scale as needed. All contours in the group are listed in the GUI with relevant information such as construction method, centroid, area, and perimeter. After creation, a contour can be converted to another type, Fourier smoothed, or copied to another location along the path.

During 2D segmentation, the user can now preview the lofted surface as additional contours are added along the path(s). The surface represents the vessel lumen and is updated automatically as the group or individual contours are modified. There are two methods for lofting the contours together, spline or nonuniform rational basis spline (NURBS) based interpolation, each with several tunable parameters. Batch segmentation using the level set or thresholding method is implemented, enabling the user to create a series of contours in along a path automatically.

3.3.2 3D Segmentation. SimVascular now incorporates a new 3D segmentation tool based on the colliding fronts algorithms [68] originally developed for VMTK. Compared to 2D segmentation, 3D segmentation may be better suited to capture local features such as aneurysms and vessel junctions, and can be used to rapidly segment entire vessel branches.

For a given domain, Ω, with boundary $\delta\Omega$, the colliding fronts algorithm solves an Eikonal equation [69]

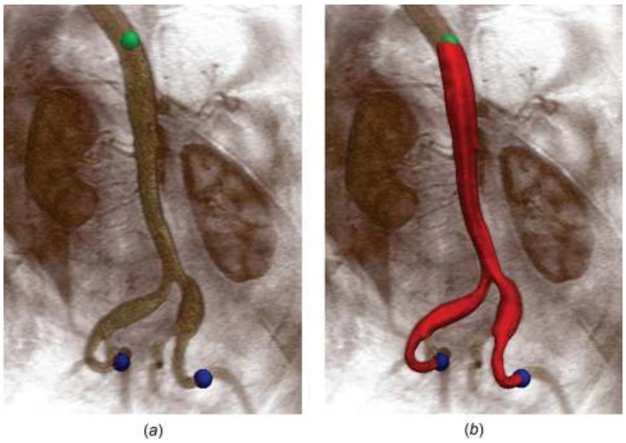


Fig. 6 3D segmentation of an MRI scan of the aorta and iliac arteries using the colliding fronts algorithm by (*a*) placement of colliding fronts seed points and (*b*) final smoothed model of the aorta and iliac arteries

$$\begin{aligned} |\nabla T| &= I^{-1},\ (x,y,z) \in \Omega \\ T &= 0,\ (x,y,z) \in \delta\Omega \end{aligned} \qquad (1)$$

where I is the image intensity, $|\cdot|$ is the Euclidean norm, and $T = T(x,y,z)$ is the arrival time function indicating the time taken to travel from an initial point to the location (x, y, z) in the speed field $I(x,y,z)$. The boundary, $\delta\Omega$, is typically set to be a point and indicates the starting point of the front.

As specified in Eq. (1), the speed with which the front(s) propagate decreases with lower image intensity. As a result, when initialized inside the vessel lumen, in contrast-enhanced medical images, fronts will propagate slowly once they reach vessel boundaries. Additionally, the gradient of the arrival time, ∇T, will be aligned along the vessel centerline. Using this method, a single vessel can be segmented by initializing two fronts, T_1 and T_2, at the start and end points of a vessel, solving the Eikonal equation and selecting all points where $\nabla T_1 \cdot \nabla T_2 \leq 0$. For multivessel models, multiple pairs of seed points can be placed interactively.

SimVascular provides a user interface to launch the colliding fronts algorithm, with which users can interactively position starting and ending seed points by manipulating axial, coronal, and sagittal planes in a 3D view. Additionally, lower and upper image intensity threshold bounds can be specified. The threshold option can improve segmentation in regions with low intensity contrast.

Figure 6 shows an example of colliding fronts segmentation in the SV GUI. In this example, the external iliac arteries are segmented simultaneously in an MR medical image volume. Figure 6(*a*) shows placement of the start and end seed points in the 3D view, noting that it is possible to segment multiple branches at once by placing multiple seed points. After placing the seed points, the 3D segmentation is generated and smoothed, producing the final model shown in Fig. 6(*b*). As illustrated, the colliding fronts segmentation follows the desired vessel.

3.4 Solid Modeling. In SimVascular, models are generated by joining 2D and/or 3D segmentations of individual vessels [70], including vessel walls, inlets, and outlets. Two types of models are supported: discrete, which uses a triangulated surface to define a PolyData model, and analytic, which includes CAD models created in OpenCASCADE or Parasolid. The SV Modeling tool is used to create and modify models (Fig. 7). It provides a face list with relevant information where the user can also change face type, color, visibility, and opacity. As a user works on different models, the interface changes depending on model type, offering corresponding options for model manipulation. The capability to interactively assign appropriate face types can facilitate subsequent steps of meshing and prescribing boundary conditions. SimVascular can also import discrete or CAD models created by other software packages, such as vtkPolyData (.vtp), stereolithography (.stl), and boundary representation (.brep).

There are several options available to process or edit PolyData models. Face operations are face-specific manipulations, such as extracting faces from a model based on differences in cell normals, combining multiple faces to a single face, filling holes, remeshing faces using VMTK, and deleting faces. Global operations that act on the entire model include remeshing with Mmg, decimation, Laplacian smoothing, butterfly subdivision, windowed sync smoothing, and densification. Local operations act on user-selected regions, which could be faces, vessel junction regions, a region within a sphere, or cells on the surface mesh. The available operations for local regions are decimation, Laplacian smoothing, constrained smoothing (which avoids shrinkage compared to regular smoothing in junction regions), linear subdivision, and loop subdivision. Trimming operations provide a plane or box for the user to cut the model at an arbitrary location or specific position perpendicular to a path.

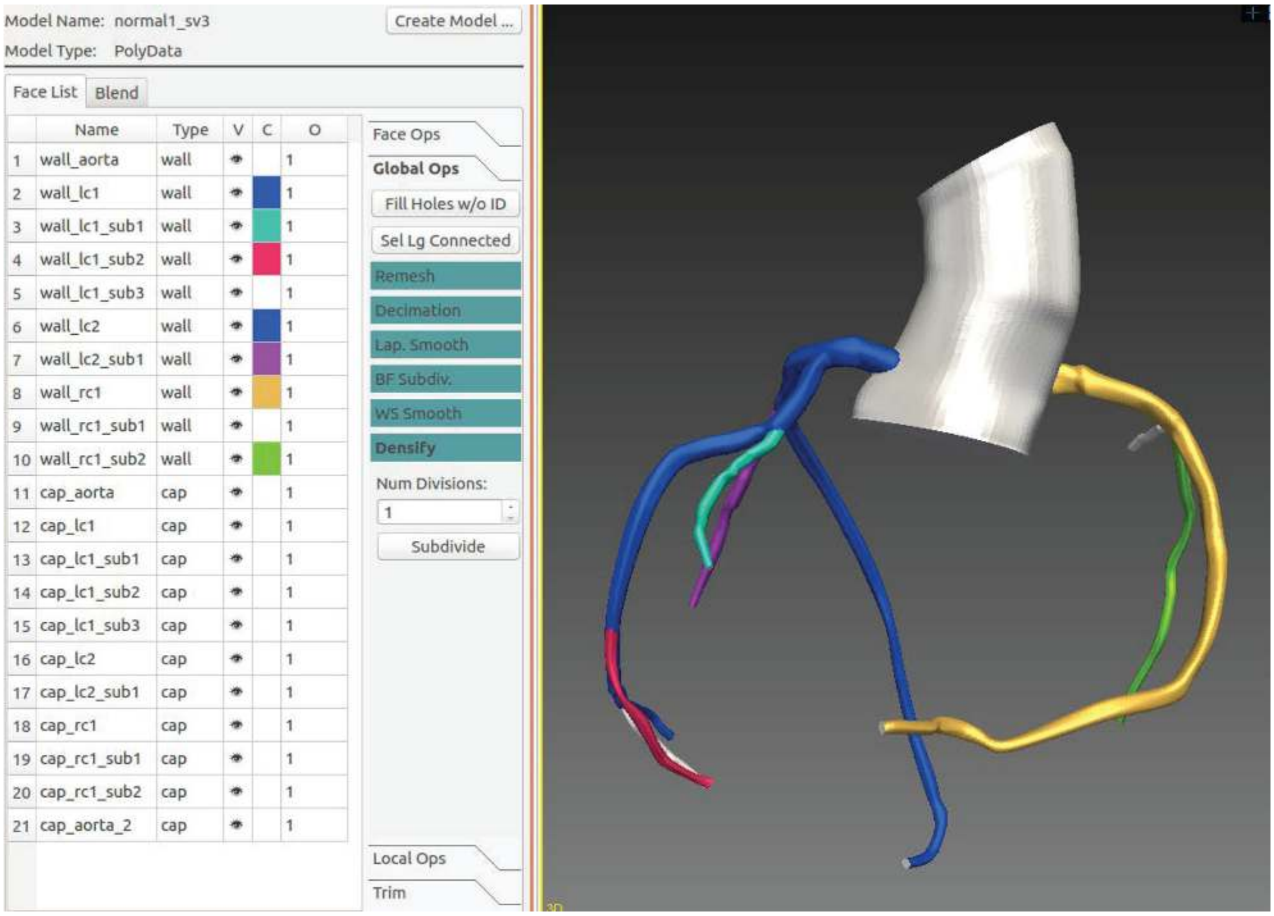
For analytic (CAD) models, blending operations are available to create smooth transitions between adjacent walls for vessel junctions. Walls are identified and paired for the user to specify the transition radius. Analytical models can be converted to PolyData models with a prescribed user-specified surface mesh size.

3.4.1 Improved B-Spline-Based Lofting. The original lofting operations from OpenCASCADE often caused undesirable artifacts in vascular models, particularly at the ends of vessels. Therefore, an improved B-spline-based method was developed to significantly reduce artifacts, compared to prior methods (Fig. 8, [71]). Specifically, each 2D segmentation is resampled to the same user-defined number of fit points, which lie exactly on the B-spline surface $S(u, v)$. The points around the circumference of the 2D segmentations ($n+1$ points) become the u parametric direction, and the points along the length of the vessel ($m+1$ points) become the v parametric direction. The equation for the improved surface is based on the full NURBS equation and reduced using a uniform weighting to

$$S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) P_{i,j} \qquad (2)$$

where $N_{i,p}$ and $N_{j,q}$ are the basis functions in the u and v parametric directions, of the desired surface degree p and q, respectively. These functions are the B-spline polynomials, found for each fit point with a corresponding parametric coordinate (u, v), which vary between 0 and 1. These functions are evaluated at every fit point to give $(n+1)(m+1)$ systems for the control points, and define a formula for the entire NURBS surface. The method also provides the flexibility to specify the degree, knot span type, and parametric span type of the result.

3.5 Meshing. To numerically solve the Navier–Stokes equations for fluid flow, the anatomic model interior must be volumetrically meshed. For meshing, SimVascular generates unstructured tetrahedral elements using TetGen (or optionally MeshSim) in the meshing module. The meshing interface contains several sections to guide the user, including *global meshing*, *advanced options*, *local size*, *regional refinement*, and *advanced flags*. With these options, in addition to global meshing and boundary layer meshing, the user can interactively select specific faces or multiple spherical regions for local refinement. With TetGen meshing, Mmg can be used for fast global surface meshing without considering the types of individual model faces. In cases where the surface mesh is undesirable, VMTK is used for individual face remeshing to improve surface mesh quality. All the data for the generated mesh are saved, including the surface mesh, the volume mesh, and the user-defined options and parameters used in mesh


Model Name: normal1_sv3

Model Type: PolyData

	Name	Type	V	C	O
1	wall_aorta	wall	👁		1
2	wall_lc1	wall	👁	🟦	1
3	wall_lc1_sub1	wall	👁	🟦	1
4	wall_lc1_sub2	wall	👁	🟥	1
5	wall_lc1_sub3	wall	👁		1
6	wall_lc2	wall	👁	🟦	1
7	wall_lc2_sub1	wall	👁	🟪	1
8	wall_rc1	wall	👁	🟨	1
9	wall_rc1_sub1	wall	👁		1
10	wall_rc1_sub2	wall	👁	🟩	1
11	cap_aorta	cap	👁		1
12	cap_lc1	cap	👁		1
13	cap_lc1_sub1	cap	👁		1
14	cap_lc1_sub2	cap	👁		1
15	cap_lc1_sub3	cap	👁		1
16	cap_lc2	cap	👁		1
17	cap_lc2_sub1	cap	👁		1
18	cap_rc1	cap	👁		1
19	cap_rc1_sub1	cap	👁		1
20	cap_rc1_sub2	cap	👁		1
21	cap_aorta_2	cap	👁		1

Num Divisions: 1

Fig. 7 A PolyData model created by SV using 2D contour groups. Faces are automatically extracted with specific types (wall or cap). In the interface, different colors can be assigned to faces for easy recognition.

creation. Thus, the user can reload a mesh and continue to modify parameters to create a revised mesh. Adaptive meshing is implemented for TegGen to change local mesh resolution using a posteriori error estimation based on blood flow speed (velocity magnitude) from simulation results [72]. Specifically, SV calculates the time-averaged flow and its second directional derivative at user-specified time steps to estimate the level of spatial change, and generates an adapted mesh capable of providing more accurate results with subsequent simulations.

3.6 Simulation. Following generation of an unstructured mesh, svPre is called to create prerequisite data files for the flow solver. Blood flow simulations are performed using svSolver, and simulation results are converted to VTK files through svPost to calculate relevant hemodynamic quantities such as wall shear stress. The simulation interface is divided to six sections to guide the user through the required simulation procedures. Many parameters are preset with default values to minimize user specification.

Fluid density and viscosity, and initial flow pressure and velocity, are specified in *basic parameters*. The user must first set up boundary conditions for individual inlets and outlets of a model in *inlet and outlet* BCs. For an inlet, a prescribed velocity profile is applied, which can be plug, parabolic, or Womersley. For an outlet, three types of boundary conditions are available in the interface to represent the downstream vasculature: Resistance, Windkessel resistance-capacitance-resistance, and Coronary lumped parameter boundary conditions [61,73]. Based on vessel type, the user assigns a suitable BC type along with appropriate parameter values.

SimVascular now offers an interactive way to split resistance or capacitance values among a set of outlets, given a total resistance or capacitance value. For example, the resistance for each outlet is calculated with the equation [74]

$$R_i = \frac{\sum \sqrt{A_i}^k}{\sqrt{A_i}^k} R_{\text{tot}} \tag{3}$$

where R_i is the resistance of the individual outlet, A_i represents the outlet area, k is Murray's law coefficient, and R_{tot} represents the total resistance of all the outlets. The capacitance for each outlet is derived from the equation

$$C_i = \frac{A_i}{\sum A_i} C_{\text{tot}} \tag{4}$$

where C_i is the capacitance of the individual outlet, and C_{tot} represents the total capacitance of all the outlets.

Wall properties provides three methods to specify vessel wall type: rigid, deformable-constant, deformable-variable. For all types, no-penetration, no-slip boundary conditions are applied. For the Rigid type, there is no wall deformation. For the deformable-constant type, the vessel wall is modeled as a linear elastic material, and assigned constant thickness, elastic modulus, Poisson's ratio, and shear modulus. In this type, vessel walls are deformable and FSI is incorporated into the flow solver using the coupled momentum method [60]. Alternatively, the deformable-variable type requires the user to specify thickness or elastic modulus at specific locations or regions, often at vessel inlets or outlets. svPre uses these input values as Dirichlet boundary conditions to solve Laplace's equation, producing a smoothly varying spatial distribution of thickness and elastic modulus on the vessel wall.

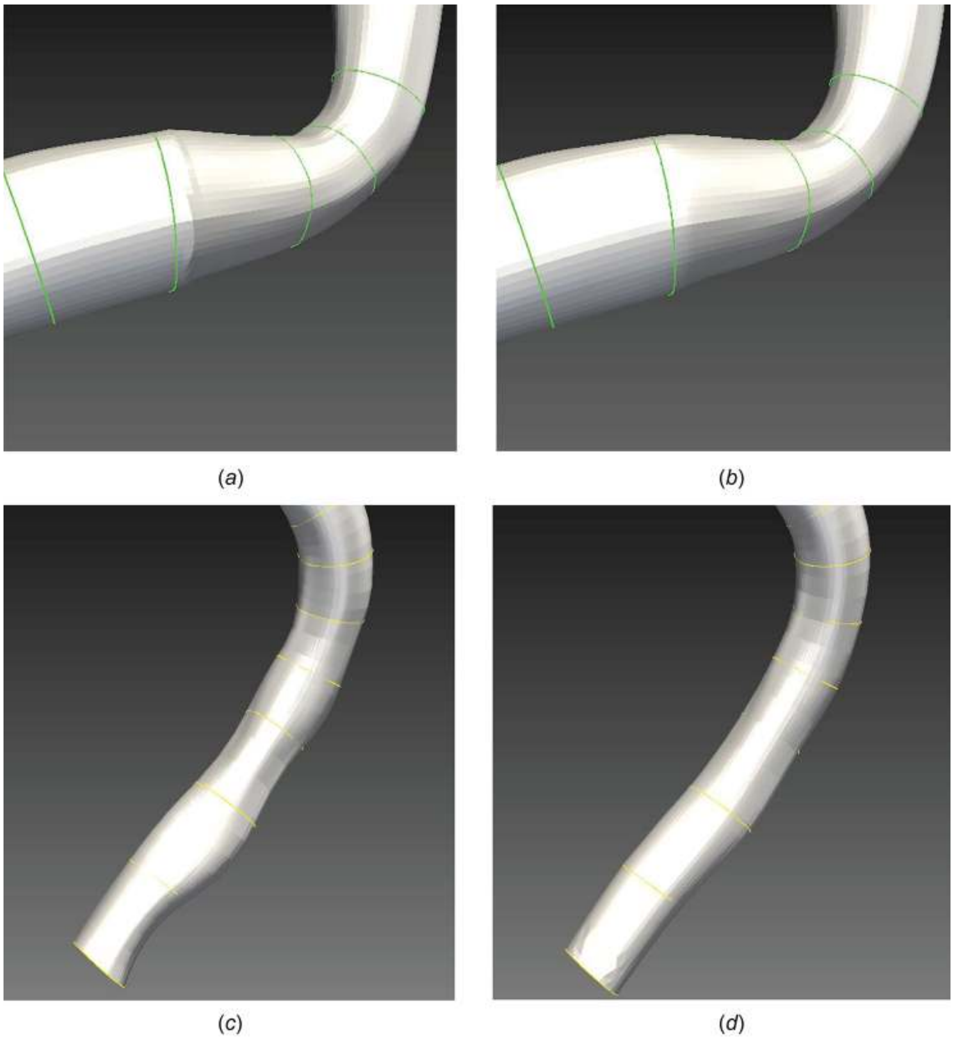


Fig. 8 Comparison of lofted surfaces between the various lofting methods: (*a*) PolyData model using a Kochanek–Bartels spline lofting, (*b*) PolyData model using the custom b-spline based lofting method, (*c*) OpenCASCADE model using the default lofting method in OpenCASCADE, and (*d*) OpenCASCADE model using the custom b-spline-based lofting method [44,71]

Solver parameters lists all required and optional parameters for the flow solver, including total number of time steps, time-step length, tolerances for the solvers, and interval for writing out simulation data to files. In *create files and run simulation*, the user-specified mesh is used to create data files containing the mesh data, initial and boundary conditions, and material properties, for launching a simulation. Simulations can be run in serial or in parallel, and the interface shows simulation progress and residual values to inform the user if the simulation is converging. Furthermore, SV enables the user to stop an ongoing simulation any time before it completes, and to subsequently restart the simulation. Many users may also wish to transfer their simulation input files to a cluster machine to run parallel jobs in a high-performance computing (HPC) environment. Following completion, *convert results* is used to convert simulation results to VTP and VTU files at user-specified time steps, and calculate relevant hemodynamic quantities such as time averaged wall shear stress and oscillatory shear index on the walls, and pressures and flow rates at the inlets and outlets. The open source package ParaView[21] can be used in conjunction with SV for data visualization and further analysis of simulation results.

4 SimVascular in the Classroom

SimVascular has been deployed for use in hands-on class projects in recent courses on computational modeling in the cardiovascular system at several institutions. Students were each given a set of medical image data, and asked to build an anatomic model, mesh the model, assign boundary conditions, run simulations, and analyze the results. Each student's image data was from the Vascular Model Repository, and corresponded to a different cardiovascular anatomic region and disease. Students reported (in oral presentations and written reports) on their chosen disease topic and its clinical relevance, methods they used for simulations, including mesh convergence and boundary condition choices, and analysis of relevant flow quantities. This provided students with hands-on, interactive experiences in a way that was accessible to students from a range of backgrounds from engineering to medicine. An educational survey-based study was conducted, with institutional review board approval, to assess progress on learning objectives [75], and results demonstrated significant gains from start to end of the course.

To increase access for students without local access to HPC resources, a web portal has been created as part of the Science GATEWAYS project[22] to provide high performance computing resources to other institutions for research and educational

[21] www.paraview.org

[22] https://sciencegateways.org/

purposes. The GATEWAY at the website[23] is currently in use by several institutions to provide students access to precompiled versions of svSolver on XSEDE machines [76,77]. The GATEWAY enables students to create their anatomic models and meshes locally on a personal computer and then upload the files for simulation to an HPC environment and download the results for processing. Students can participate in user forums and follow documentation as well as written and video tutorials on the SimVascular website.

5 Conclusions

We have reported and summarized recent developments and improvements to the SimVascular open source project, which aims to provide a community resource for cardiovascular basic science and clinical research and bioengineering education. These developments include new data handling, interactive graphical user interface, and expanded functionality. As multiple data types are involved in the cardiovascular modeling and simulation pipeline, a data manager is now used to organize data into a project format and automatically handle file IO. This facilitates creation of complicated projects and storage of multiple pieces of corresponding data. The new graphical user interface, which is built with a modernized software architecture, now provides multiple "intelligent" tools spanning the entire cardiovascular modeling and simulation pipeline, with an interface that automatically updates as the user proceeds through the steps and corresponding data types. The interface enables users to directly and interactively manipulate data in new a 2D/3D-view windows format. The 3D segmentation tools have been improved to better complement prior 2D segmentation tools. With this tool, users can better capture the geometry of junctions and complex local anatomic features such as aneurysms. In addition, SimVascular has been used for hands-on learning in the classroom setting, and a web portal has been deployed to expand access to student users.

SimVascular is an actively maintained open source project, with additional enhancements and new features in preparation. For image segmentation, deep learning-based 2D segmentation has been explored with the goal to speed up the image-based anatomic modeling process and make it possible for large scale application of cardiovascular simulation on clinical studies [78,79]. These tools will be added to the GUI in future releases. For solid modeling, new algorithms, to convert a discrete model to a globally valid NURBS model (the CAD standard) through a polycube structure, are under development [80]. This will enable users to manipulate solid models more easily and facilitating shape optimization for treatment planning. An integrated one-dimensional solver will be released, enabling fast lower-order simulations and wave propagation studies. Automated tuning methods will be integrated for parameter estimation in lumped parameter networks to match clinical data. In the near future, we plan to release an arbitrary Lagrangian–Eulerian-based solver to handle large deformation, wall motions, and ventricular hemodynamics. Python access to SimVascular core functions will soon be supported, which will enable the user to write custom scripts to carry out custom or batch operations. This will facilitate use of SimVascular for optimization, uncertainty quantification, and studies with large patient cohorts.

Funding Data

- Directorate for Computer and Information Science and Engineering (Grant Nos. 1339824 and 1663671).
- National Institute of Biomedical Imaging and Bioengineering (Grant No. 1R01EB018302-01A1).

References

[1] Nichols, W., O'Rourke, M., and Vlachopoulos, C., 2011, *McDonald's Blood Flow in Arteries: Theoretical, Experimental and Clinical Principles*, 6th, ed., CRC Press, Boca Raton, FL.

[2] Davies, P. F., Civelek, M., Fang, Y., and Fleming, I., 2013, "The Atherosusceptible Endothelium: Endothelial Phenotypes in Complex Haemodynamic Shear Stress Regions In Vivo," Cardiovasc. Res., **99**(2), pp. 315–327.

[3] Kwak, B. R., Back, M., Bochaton-Piallat, M. L., Caligiuri, G., Daemen, M. J., Davies, P. F., Hoefer, I. E., Holvoet, P., Jo, H., Krams, R., Lehoux, S., Monaco, C., Steffens, S., Virmani, R., Weber, C., Wentzel, J. J., and Evans, P. C., 2014, "Biomechanical Factors in Atherosclerosis: Mechanisms and Clinical Implications," Eur. Heart J., **35**(43), pp. 3013–3020.

[4] Malek, A. M., Alper, S. L., and Izumo, S., 1999, "Hemodynamic Shear Stress and Its Role in Atherosclerosis," J. Am. Med. Assoc., **282**(21), pp. 2035–2042.

[5] Morbiducci, U., Kok, A. M., Kwak, B. R., Stone, P. H., Steinman, D. A., and Wentzel, J. J., 2016, "Atherosclerosis at Arterial Bifurcations: Evidence for the Role of Haemodynamics and Geometry," Thromb. Haemostasis, **115**(3), pp. 484–492.

[6] Wentzel, J. J., Janssen, E., Vos, J., Schuurbiers, J. C., Krams, R., Serruys, P. W., Feyter, P. J., and Slager, C. J., 2003, "Extension of Increased Atherosclerotic Wall Thickness Into High Shear Stress Regions Is Associated With Loss of Compensatory Remodeling," Circulation, **108**(1), pp. 17–23.

[7] Chatzizisis, Y. S., Coskun, A. U., Jonas, M., Edelman, E. R., Feldman, C. L., and Stone, P. H., 2007, "Role of Endothelial Shear Stress in the Natural History of Coronary Atherosclerosis and Vascular Remodeling: Molecular, Cellular, and Vascular Behavior," J. Am. Coll. Cardiol., **49**(25), pp. 2379–2393.

[8] Dobrin, P. B., 1995, "Mechanical Factors Associated With the Development of Intimal and Medial Thickening in Vein Grafts Subjected to Arterial Pressure. A Model of Arteries Exposed to Hypertension," Hypertension, **26**(1), pp. 38–43.

[9] Valentin, A., Cardamone, L., Baek, S., and Humphrey, J., 2009, "Complementary Vasoactivity and Matrix Remodelling in Arterial Adaptations to Altered Flow and Pressure," J. R. Soc. Interface, **6**(32), pp. 38–43.

[10] Giaid, A., and Saleh, D., 1995, "Reduced Expression of Endothelial Nitric Oxide Synthase in the Lungs of Patients With Pulmonary Hypertension," N. Engl. J. Med., **333**(4), pp. 214–221.

[11] Tada, Y., Wada, K., Shimada, K., Makino, H., Liang, E. I., Murakami, S., Kudo, M., Kitazato, K. T., Nagahiro, S., and Hashimoto, T., 2014, "Roles of Hypertension in the Rupture of Intracranial Aneurysms," Stroke, **45**(2), pp. 579–586.

[12] Arzani, A., Suh, G. Y., Dalman, R. L., and Shadden, S. C., 2014, "A Longitudinal Comparison of Hemodynamics and Intraluminal Thrombus Deposition in Abdominal Aortic Aneurysms," Am. J. Physiol., **307**(12), pp. H1786–H1795.

[13] Shadden, S. C., and Hendabadi, S., 2013, "Potential Fluid Mechanic Pathways of Platelet Activation," Biomech. Model. Mechanobiol., **12**(3), pp. 467–474.

[14] Mukherjee, D., Jani, N. D., Selvaganesan, K., Weng, C. L., and Shadden, S. C., 2016, "Computational Assessment of the Relation Between Embolism Source and Embolus Distribution to the Circle of Willis for Improved Understanding of Stroke Etiology," ASME J. Biomech. Eng., **138**(8), p. 081008.

[15] Mukherjee, D., and Shadden, S. C., 2017, "Inertial Particle Dynamics in Large Artery Flows—Implications for Modeling Arterial Embolisms," J. Biomech., **52**(8), pp. 155–164.

[16] Zarins, C. K., Giddens, D. P., Bharadvaj, B. K., Sottiurai, V. S., Mabon, R. F., and Glagov, S., 1983, "Carotid Bifurcation Atherosclerosis. Quantitative Correlation of Plaque Localization With Flow Velocity Profiles and Wall Shear Stress," Circ. Res., **53**(4), pp. 502–514.

[17] Nieman, K., Oudkerk, M., Rensing, B. J., Ooijen, P., Munne, A., Geuns, R., and Feyter, P. J., 2001, "Coronary Angiography With Multi-Slice Computed Tomography," Lancet, **357**(9256), pp. 599–603.

[18] Elkins, C. J., and Alley, M. T., 2007, "Magnetic Resonance Velocimetry: Applications of Magnetic Resonance Imaging in the Measurement of Fluid Motion," Exp. Fluids, **43**(6), pp. 823–858.

[19] Rosch, T., Braig, C., Gain, T., Feuerbach, S., Siewert, J. R., Schusdziarra, V., and Classen, M., 1992, "Staging of Pancreatic and Ampullary Carcinoma by Endoscopic Ultrasonography: Comparison With Conventional Sonography, Computed Tomography, and Angiography," Gastroenterology, **102**(1), pp. 823–858.

[20] Sarvazyan, A., Hall, T. J., Urban, M. W., Fatemi, M., Aglyamov, S. R., and Garra, B. S., 2011, "An Overview of Elastography—An Emerging Branch of Medical Imaging," Curr. Med. Imaging Rev., **7**(4), pp. 255–282.

[21] Taylor, C. A., Hughes, T. J. R., and Zarins, C. K., 1996, "Computational Investigations in Vascular Disease," Comput. Phys., **10**(3), pp. 224–232.

[22] Taylor, C. A., Draney, M. T., Ku, J. P., Parker, D., Steele, B. N., Wang, K., and Zarins, C. K., 1999, "Predictive Medicine: Computational Techniques in Therapeutic Decision-Making," Comput. Aided Surg., **4**(5), pp. 231–247.

[23] Krams, R., Wentzel, J. J., Oomen, J. A. F., Vinke, R., Schuurbiers, J. C. H., Feyter, P. J. D., Serruys, P. W., and Slager, C. J., 1997, "Evaluation of Endothelial Shear Stress and 3D Geometry as Factors Determining the Development of Atherosclerosis and Remodeling in Human Coronary Arteries In Vivo Combining 3D Reconstruction From Angiography and IVUS (ANGUS) With Computational Fluid Dynamics," Arterioscler., Thromb., Vasc. Biol., **17**(10), pp. 2061–2065.

[24] Milner, J. S., Moore, J. A., Rutt, B. K., and Steinman, D. A., 1998, "Hemodynamics of Human Carotid Artery Bifurcations: Computational Studies With Models Reconstructed From Magnetic Resonance Imaging of Normal Subjects," J. Vasc. Surg., **28**(1), pp. 143–156.

[25] Perktold, K., Hofer, M., Karner, G., Trubel, W., and Schima, H., 1998, "Computer Simulation of Vascular Fluid Dynamics and Mass Transport: Optimal Design of Arterial Bypass Anastomoses," Fourth European Computational Fluid Dynamics Conference, Athens, Greece, Sept. 7–11, pp. 484–489.

[26] Moore, J. A., Steinman, D. A., Holdsworth, D. W., and Ethier, C. R., 1999, "Accuracy of Computational Hemodynamics in Complex Arterial Geometries

[23] https://gateway.simvascular.org/

Reconstructed From Magnetic Resonance Imaging," Ann. Biomed. Eng., **27**(1), pp. 32–41.

[27] Shadden, S. C., and Taylor, C. A., 2008, "Characterization of Coherent Structures in the Cardiovascular System," Ann. Biomed. Eng., **36**(7), pp. 1152–1162.

[28] Hendabadi, S., Bermejo, J., Benito, Y., Yotti, R. L., Fernández-Avilés, F., del Álamo, J. C., and Shadden, S. C., 2013, "Topology of Blood Transport in the Human Left Ventricle by Novel Processing of Doppler Echocardiography," Ann. Biomed. Eng., **41**(12), pp. 2603–2616.

[29] Arzani, A., Gambaruto, A. M., Chen, G., and Shadden, S. C., 2017, "Wall Shear Stress Exposure Time: A Lagrangian Measure of Near-Wall Stagnation and Concentration in Cardiovascular Flows," Biomech. Model. Mechanobiol., **16**(3), pp. 787–803.

[30] Ramachandra, A. B., Kahn, A. M., and Marsden, A. L., 2016, "Patient Specific Simulations Reveal Significant Differences in Mechanical Stimuli in Venous and Arterial Coronary Grafts," J. Cardiovasc. Transl. Res., **9**(4), pp. 279–290.

[31] Schiavazzi, D., Kung, E. O., Marsden, A. L., Baker, C., Pennati, G., Hsia, T., and Dorfman, A., 2015, "Hemodynamic Effects of Left Pulmonary Artery Stenosis Following Superior Cavopulmonary Connection: A Patient-Specific Multiscale Modeling Study," J. Thorac. Cardiovasc. Surg., **149**(3), pp. 689–696.

[32] Marsden, A. L., 2014, "Optimization in Cardiovascular Modeling," Annu. Rev. Fluid Mech., **46**, pp. 519–546.

[33] Esmaily-Moghadam, M., Hsia, T.-Y., and Marsden, A. L., 2015, "The Assisted Bidirectional Glenn: A Novel Surgical Approach for First Stage Single Ventricle Heart Palliation," J. Thorac. Cardiovasc. Surg., **149**(3), pp. 699–705.

[34] Masoumi Khalil Abad, E., Pasini, D., and Cecere, R., 2012, "Shape Optimization of Stress Concentration-Free Lattice for Self-Expandable Nitinol Stent-Grafts," J. Biomech., **45**(6), pp. 1028–1035.

[35] Burgreen, G. W., Antaki, J. F., Wu, Z. J., and Holmes, A. J., 2001, "Computational Fluid Dynamics as a Development Tool for Rotary Blood Pumps," Artif. Organs, **25**(5), pp. 336–340.

[36] Marsden, A. L., 2013, "Simulation Based Planning of Surgical Interventions in Pediatric Cardiology," Phys. Fluids, **25**(10), p. 101303.

[37] Yang, W., Vignon-Clementel, I. E., Troianowski, G., Reddy, V. M., Feinstein, J. A., and Marsden, A. L., 2012, "Hepatic Blood Flow Distribution and Performance in Conventional and Novel Y-Graft Fontan Geometries: A Case Series Computational Fluid Dynamics Study," J. Thorac. Cardiovasc. Surg., **143**(5), pp. 1086–1097.

[38] Douglas, P. S., Pontone, G., Hlatky, M. A., Patel, M. R., Norgaard, B. L., Byrne, R. A., Curzen, N., Purcell, I., Gutberlet, M., Rioufol, G., Hink, U., Schuchlenz, H. W., Feuchtner, G., Gilard, M., Andreini, D., Jensen, J. M., Hadamitzky, M., Chiswell, K., Cyr, D., Wilk, A., Wang, F., Rogers, C., and De Bruyne, B., 2015, "Clinical Outcomes of Fractional Flow Reserve by Computed Tomographic Angiography-Guided Diagnostic Strategies Vs. Usual Care in Patients With Suspected Coronary Artery Disease: The Prospective Longitudinal Trial of FFRCT: Outcome and Resource Impacts Study," Eur. Heart J., **36**(47), pp. 3359–3367.

[39] Min, J. K., Leipsic, J., and Pencina, M. J., 2012, "Diagnostic Accuracy of Fractional Flow Reserve From Anatomic CT Angiography," J. Am. Med. Assoc., **308**(12), pp. 1237–1245.

[40] Coenen, A., Lubbers, M. M., Kono, A., Dedic, A., Chelu, R. G., Dijkshoorn, M. L., Gijsen, F. J., Ouhlous, M., Geuns, R. J., and Nieman, K., 2015, "Fractional Flow Reserve Computed From Noninvasive CT Angiography Data: Diagnostic Performance of an On-Site Clinician-Operated Computational Fluid Dynamics Algorithm," Radiology, **274**(3), pp. 674–683.

[41] Ko, B. S., Cameron, J. D., Munnur, R. K., Wong, D. T. L., Fujisawa, Y., and Sakaguchi, T., 2017, "Noninvasive CT-Derived FFR Based on Structural and Fluid Analysis: A Comparison With Invasive FFR for Detection of Functionally Significant Stenosis," J. Am. Coll. Cardiol.: Cardiovasc. Imaging, **10**(6), pp. 663–673.

[42] Itu, L., Rapaka, S., Passerini, T., Georgescu, B., Schwemmer, C., Schoebinger, M., Flohr, T., Sharma, P., and Comaniciu, D., 2016, "A Machine-Learning Approach for Computation of Fractional Flow Reserve From Coronary Computed Tomography," J. Appl. Physiol., **121**(1), pp. 42–52.

[43] Schmidt, J. P., Delp, M. A., Sherman, C. A., Pande, V. S., and Altman, R. B., 2008, "The Simbios National Center: Systems Biology in Motion," Proc. IEEE. Inst. Electr. Electron. Eng., **96**(8), pp. 1266–1280.

[44] Updegrove, A., Wilson, N. M., Merkow, J., Lan, H., Marsden, A. L., and Shadden, S. C., 2017, "SimVascular: An Open Source Pipeline for Cardiovascular Simulation," Ann. Biomed. Eng., **45**(3), pp. 525–541.

[45] Lonyai, A., Dubin, A. M., Feinstein, J. A., Taylor, C. A., and Shadden, S. C., 2010, "New Insights Into Pacemaker Lead-Induced Venous Occlusion: Simulation-Based Investigation of Alterations in Venous Biomechanics," Cardiovasc. Eng., **10**(2), pp. 84–90.

[46] Marsden, A. L., Reddy, V. M., Shadden, S. C., Chan, F. P., Taylor, C. A., and Feinstein, J. A., 2010, "A New Multiparameter Approach to Computational Simulation for Fontan Assessment and Redesign," Congenital Heart Dis., **5**(2), pp. 104–117.

[47] Arzani, A., Dyverfeldt, P., Ebbers, T., and Shadden, S. C., 2012, "In Vivo Validation of Numerical Prediction for Turbulence Intensity in an Aortic Coarctation," Ann. Biomed. Eng., **40**(4), pp. 860–870.

[48] Bockman, M. D., Kansagra, A. P., Shadden, S. C., Wong, E. C., and Marsden, A. L., 2012, "Fluid Mechanics of Mixing in the Vertebrobasilar System: Comparison of Simulation and MRI," Cardiovasc. Eng. Technol., **3**(4), pp. 450–461.

[49] Carr, I. A., Nemoto, N., Schwartz, R. S., and Shadden, S. C., 2013, "Size-Dependent Predilections of Cardiogenic Embolic Transport," Am. J. Physiol., **305**(5), pp. H732–H739.

[50] Hansen, A., Arzani, K. B., and Shadden, S. C., 2015, "Mechanical Platelet Activation Potential in Abdominal Aortic Aneurysms," ASME J. Biomech. Eng., **137**(4), p. 041005.

[51] Mukherjee, D., Padilla, J., and Shadden, S. C., 2016, "Numerical Investigation of Fluid-Particle Interactions for Embolic Stroke," Theor. Comput. Fluid Dyn., **30**(1), pp. 23–39.

[52] Sengupta, D., Kahn, A. M., Burns, J. C., Sankaran, S., Shadden, S. C., and Marsden, A. L., 2012, "Image-Based Modeling of Hemodynamics in Coronary Artery Aneurysms Caused by Kawasaki Disease," Biomech. Model. Mechanobiol., **11**(6), pp. 915–932.

[53] Sengupta, D., Kahn, A. M., Kung, E., Esmaily-Moghadam, M., Shirinsky, O., Lyskina, G. A., Burns, J. C., and Marsden, A. L., 2012, "Thrombotic Risk Stratification Using Computational Modeling in Patients With Coronary Artery Aneurysms Following Kawasaki Disease," Biomech. Model. Mechanobiol., **13**(6), pp. 1261–1276.

[54] Yang, W., Chan, F. P., Reddy, V. M., Marsden, A. L., and Feinstein, J. A., 2015, "Flow Simulations and Validation for the First Cohort of Patients Undergoing the Y-Graft Fontan Procedure," J. Thorac. Cardiovasc. Surg., **149**(1), pp. 247–255.

[55] Tran, J. S., Schiavazzi, D. E., Ramachandra, A. B., Kahn, A. M., and Marsden, A. L., 2017, "Automated Tuning for Parameter Identification and Uncertainty Quantification in Multi-Scale Coronary Simulations," Comput. Fluids, **142**, pp. 128–138.

[56] Wilson, N. M., Ortiz, A. K., and Johnson, A. B., 2013, "The Vascular Model Repository: A Public Resource of Medical Imaging Data and Blood Flow Simulation Results," ASME J. Med. Devices, **7**(4), p. 040923.

[57] Sankaran, S., Esmaily-Moghadam, M., Kahn, A. M., Guccione, J., Tseng, E., and Marsden, A. L., 2012, "Patient-Specific Multiscale Modeling of Blood Flow for Coronary Artery Bypass Graft Surgery," Ann. Biomed. Eng., **40**(1), pp. 2228–2242.

[58] Marsden, A. L., and Esmaily-Moghadam, M., 2015, "Multiscale Modeling of Cardiovascular Flows for Clinical Decision Support," ASME Appl. Mech. Rev., **67**(3), p. 030804.

[59] Taylor, C. A., Hughes, T. J. R., and Zarins, C. K., 1998, "Finite Element Modeling of Blood Flow in Arteries," Comput. Methods Appl. Mech. Eng., **158**(1), pp. 155–196.

[60] Figueroa, C. A., Vignon-Clementel, I. E., Jansen, K. E., Hughes, T. J. R., and Taylor, C. A., 2006, "A Coupled Momentum Method for Modeling Blood Flow in Three-Dimensional Deformable Arteries," Comput. Methods Appl. Mech. Eng., **195**(41–43), pp. 5685–5706.

[61] Vignon-Clementel, I. E., Figueroa, C. A., Jansen, K. E., and Taylor, C. A., 2010, "Outflow Boundary Conditions for Three-Dimensional Simulations of Non-Periodic Blood Flow and Pressure Fields in Deformable Arteries," Comput. Methods Biomech. Biomed. Eng., **13**(5), pp. 625–640.

[62] Whiting, C. H., and Jansen, K. E., 2001, "A Stabilized Finite Element Method for the Incompressible Navier-Stokes Equations Using a Hierarchical Basis," Int. J. Numer. Methods Fluids, **35**(1), pp. 93–116.

[63] Franca, L. P., and Frey, S. L., 1992, "Stabilized Finite Element Methods: II. The Incompressible Navier-Stokes Equations," Comput. Methods Appl. Mech. Eng., **99**(2–3), pp. 209–233.

[64] Esmaily-Moghadam, M., Bazilevs, Y., and Marsden, A. L., 2015, "A Bi-Partitioned Iterative Algorithm for Solving Linear Systems Obtained From Incompressible Flow Problems," Comput. Methods Appl. Mech. Eng., **286**, pp. 40–62.

[65] Esmaily-Moghadam, M., Bazilevs, Y., and Marsden, A. L., 2013, "A New Preconditioning Technique for Implicitly Coupled Multidomain Simulations With Applications to Hemodynamics," Comput. Mech., **52**(5), pp. 1141–1152.

[66] Sharma, N., and Aggarwal, L. M., 2010, "Automated Medical Image Segmentation Techniques," J. Med. Phys., **35**(1), pp. 3–14.

[67] Norouzi, A., Rahim, M. S. M., Altameem, A., Saba, T., Ehsani, A., Rehman, A., and Uddin, M., 2014, "Medical Image Segmentation Methods, Algorithms, and Applications," IETE Tech. Rev., **31**(3), pp. 199–213.

[68] Antiga, L., Piccinelli, M., Botti, L., Ene-Iordache, B., Remuzzi, A., and Steinman, D. A., 2008, "An Image-Based Modeling Framework for Patient-Specific Computational Hemodynamics," Med. Biol. Eng. Comput., **46**(11), pp. 1097–1112.

[69] Sethian, J. A., 1999, "Fast Marching Methods," SIAM Rev., **41**(2), pp. 199–235.

[70] Updegrove, A., Wilson, N. M., and Shadden, S. C., 2016, "Boolean and Smoothing of Discrete Polygonal Surfaces," Adv. Eng. Software, **95**, pp. 16–27.

[71] Updegrove, A. R., Wilson, N. M., and Shadden, S. C., 2016, "Integration of OpenCASCADE and Improved Global Surface Interpolation Algorithms Into SimVascular 2.0," Summer Biomechanics, Bioengineering and Biotransport Conference (SB3C), June 29–July 2, National Harbor, MD, pp. 929–930.

[72] Sahni, O., Müller, J., Jansen, K. E., Shephard, M. S., and Taylor, C. A., 2006, "Efficient Anisotropic Adaptive Discretization of the Cardiovascular System," Comput. Methods Appl. Mech. Eng., **195**(41–43), pp. 5634–5655.

[73] Kim, H. J., Vignon-Clementel, I. E., Coogan, J. S., Figueroa, C. A., Jansen, K. E., and Taylor, C. A., 2010, "Patient-Specific Modeling of Blood Flow and Pressure in Human Coronary Arteries," Ann. Biomed. Eng., **38**(10), pp. 3195–3209.

[74] Olufsen, M. S., 1999, "Structured Tree Outflow Condition for Blood Flow in Larger Systemic Arteries," Am. J. Physiol., **276**(1), pp. H257–H268.

[75] Goergen, C. C., Shadden, S. C., and Marsden, A. L., 2017, "Simvascular as an Instructional Tool in the Classroom," IEEE Frontiers in Education Conference (FIE), Indianapolis, IN, Oct. 18–21, pp. 1–4.

[76] Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazelwood, V., Lathrop, S., Lifka, D., Peterson, G. P., Roskies, R., Scott, J. R., and Wilkins-Diehr, N., 2014, "XSEDE: Accelerating Scientific Discovery," Comput. Sci. Eng., **16**(5), pp. 62–74.

[77] Lawrence, K. A., Zentner, M., Wilkins-Diehr, N., Wernert, J. A., Pierce, M., Marru, S., and Michael, S., 2015, "Science Gateways Today and Tomorrow: Positive Perspectives of Nearly 5000 Members of the Research Community," Concurrency Comput.: Pract. Exp., **27**(16), pp. 4252–4268.

[78] Merkow, J., Tu, Z., Kriegman, D., and Marsden, A. L., 2015, "Structural Edge Detection for Cardiovascular Modeling," International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany, Oct. 5–9, pp. 735–742.

[79] Merkow, J., Kriegman, D., Marsden, A., and Tu, Z., 2016, "Dense Volume-to-Volume Vascular Boundary Detection," preprint arXiv:1605.08401.

[80] Updegrove, A., Wilson, N. M., and Shadden, S. C., 2017, "Constructing Analysis Suitable Nurbs From Discrete Image-Based Models," Summer Biomechanics, Bioengineering and Biotransport Conference (SB3C), Tucson, AZ, June 21–24, pp. 852–853.