An interactive computational pipeline to investigate ventricular hemodynamics with real-time three-dimensional echocardiography and computational fluid dynamics

Jan-Niklas Thiel[1] Daniel Verhülsdonk[1] Ulrich Steinseifer[1]
Katharina Linden[2] Ulrike Herberg[2] Ingeborg Friehs[3] Daniel Diaz-Gil[3]
Michael Neidlin[1]

[1]Department of Cardiovascular Engineering, Institute of Applied Medical Engineering Medical Faculty, RWTH Aachen University, Aachen, Germany

[2]Department of Paediatric Cardiology, Medical Faculty, University Hospital Aachen, Aachen, Germany

[3]Department of Cardiac Surgery, Boston Children's Hospital, Boston, USA

neidlin@ame.rwth-aachen.de

June 25, 2024

Abstract

Blood flow within the ventricle can provide important information on the performance of the heart. The determined blood flow structures are used to extract flow biomarkers to quantify cardiac function. Patient-specific computational fluid dynamics (CFD) models that import segmented ventricular deformations from non-invasive imaging data for an individualized hemodynamical analysis are often used. However, tedious pre-processing of those geometries is often necessary and decisions on the modeling of the valve and the surrounding vessels have to be made on an individual level. This leads to a lack of reproducibility and usability of the existing computational models. In this work we introduce *IP-HEART* - an interactive and open-source computational pipeline to perform geometry processing for CFD models of ventricular blood flow. We showcase its use on real-time three-dimensional echocardiography data of three patient datasets from two different clinical centers. We outline how different modeling assumptions of the mitral valve can be easily implemented and quantify their effect on CFD simulations. The results correspond well with clinical data on transvalvular Doppler ultrasound recordings and distinct flow features such as mitral jet and diastolic vortex formation can be observed. The pipeline is accompanied by an extensive video tutorial and freely available code for further use.

Keywords: Ventricular hemodynamics; real-time three-dimensional echocardiography; moving mesh simulations; geometry processing

1 Introduction

For the heart, it is known that fluid dynamics within the ventricle play an important role in many aspects of physiology. There exists a relationship between fluid dynamical features and cardiac behavior in health and disease [7, 19, 18]. Some examples include the

vortex formation time for the diagnosis and prognosis of left ventricular heart failure [20] or the kinetic energy as a marker for cardiac function in single ventricle circulation [25]. These simulation derived markers have the potential to corroborate clinical decisions and ultimately improve treatment of cardiovascular diseases.

Recent advances in cardiovascular imaging, such as time-resolved magnetic resonance imaging (4D MRI) and real-time three-dimensional echocardiography (RT3DE), gave rise to a vast amount of Computational Fluid Dynamics (CFD) models of ventricular hemodynamics focusing on the intriguing role of the intraventricular flow features and their relation to the cardiac physiology. By integrating anatomical and physiological information from individual patients, simulation models can provide detailed insights into the patient's cardiac function and hemodynamics. Independent of the imaging modality, such personalized models extract the movement from the endocardium and impose it as a boundary condition on the computational mesh. This approach has been applied to investigate blood flows in adult [21, 14, 3], pediatric [13], and fetal [24] subjects to quantify intraventricular hemodynamics during health and disease.

In all studies of ventricular hemodynamics, the modeling approach of the atrioventricular valve is a critical point which influences the intraventricular flow field. At the same time, the anatomy of the valve is extremely complex and some degree of abstraction is always required to keep computational costs in an acceptable range. In previous studies, the left atrioventricular valve, the mitral valve (MV), was completely neglected [22] or modelled as a simple wall [11], as a porous media [5] or as a model with segmented, moving leaflets [1]. Apart from the valve model, the surrounding vascular structures, i.a. the atrium and aorta/pulmonary artery, may also influence the intraventricular flow field [8]. Consequently, it is necessary to make decisions with respect to those anatomies. All these aspects require extensive pre-processing, which is often tedious and error-prone and must be carried out manually. As such, results might be significantly influenced by those choices. Further on, some flow markers can be determined with simplified approaches, while others require more sophisticated modeling techniques [23]. Thus, in order to make the existing models reusable and comparable with each other, open-source tools that enable automated geometry operations and simulation pre-processing are required. Furthermore, these should facilitate the adaptation of existing pipelines and the expansion of models.

In this work, we thus develop a comprehensive CFD pipeline for the heart through an automated geometry pre-processing framework, implementing it in the widely used *Blender* software [4]. Further on, we present how different model choices can be selected with our model, their influence on the simulation results and showcase its use on clinical data from two different centers.

2 Methods

2.1 Clinical data

Three-dimensional real-time echocardiography data of three subjects (P1-P3) was recorded and segmented with the TomTec Image Arena LV software at the two clinical sites and exported in ucd format with 11-17 segmentations per cardiac cycle. In addition, the size of the aortic and mitral valves as well as Doppler ultrasound recordings of the transvalvular velocities were exported by the clinicians. Table 1 summarizes the ventricular volumes, cycle durations, and valve sizes of the patients.

Table 1: Ventricular volumes, cardiac cycle features, and valve sizes. ESV: End-systolic volume, EDV: End-diastolic volume, t_{RR}: RR-duration, t_D: onset of diastole, r_{MVl}: radius of mitral valve long axis, r_{MVs}: radius of mitral valve short axis, r_{AV}: radius of aortic valve.

Variable	Unit	P1	P2	P3
Ventricular volumes:				
ESV	mL	18.4	7.2	2.4
EDV	mL	44.9	17.1	5.5
Cardiac cycle:				
t_{RR}	ms	822	514	528
t_D	ms	370	166	199
MV:				
r_{MVl}	mm	13	1.05	2.8
r_{MVs}	mm	10.5	0.89	2.37
AV:				
r_{AV}	mm	8.5	2.12	2.12

2.2 Pre-processing pipeline

Blender was utilized for the geometrical reconstruction of the ventricle geometries. To aid in this process, a *Blender* user-interface add-on has been developed using *Python*. The add-on is designed to carry out various operations commonly used during geometrical reconstruction, with functions sorted into multiple panels. These functions include both algorithmic operations and parameter setup assistance.

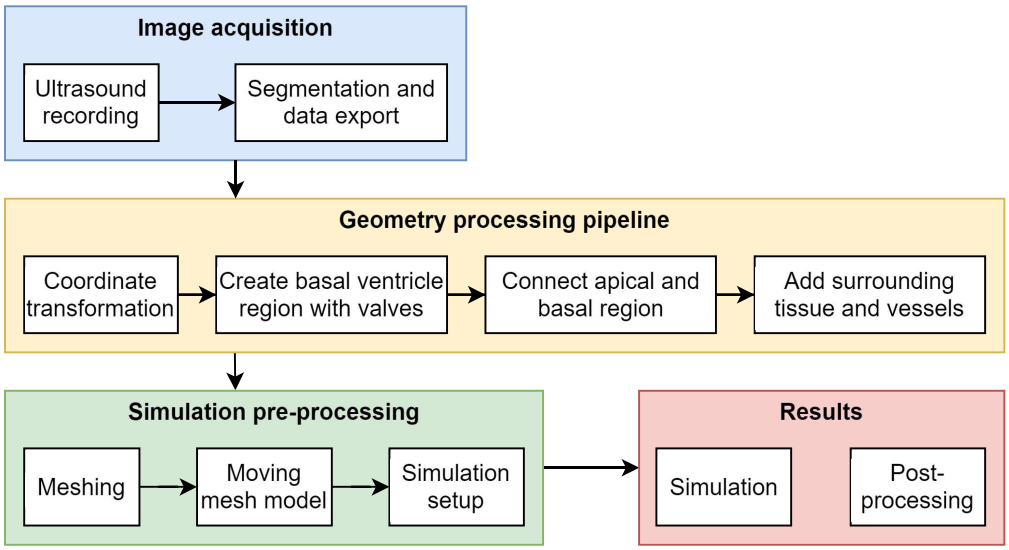


Figure 1: Pipeline for hemodynamic analysis of the left ventricle.

The pipeline steps for creating a reconstructed ventricle geometry are depicted in Figure 1. The input for the pipeline consists of object files in stl-format generated from segmented ventricles using RT3DE data and TomTec Image Arena software. The initial step for all approaches in the pipeline is coordinate transformation since the recorded

ultrasound data is not in the same coordinate system. Afterwards, new basal regions of the ventricle are generated together with aortic and mitral valves. The new basal regions are then connected to the ventricle. The final step involves adding the surrounding tissue and vessels, i.e. aorta and atrium. The resulting objects can be exported from *Blender* in stl-format and are suitable for numerical simulations using any moving mesh technique within a CFD solver. More detailed information on the individual steps of the pipeline and their implementation are presented in the appendix A.

2.3 Simulation setup

For all simulations the moving mesh approach from [9] was used through user defined functions in *Fluent 2021R2* (Ansys Inc., Canonburg, USA). In brief, node coordinates are extracted from the exported STL files and separate surface files with the coordinates of each node are created. Cubic spline interpolation is used to increase the temporal resolution of the STL data. These surface files are used to define mesh motion at each time-step of the CFD simulation. A more detailed description of the model and a video tutorial can be found on `https://github.com/mneidlin/movingmesh`. Different modeling approaches (A1-A4) were investigated for patient P1 to quantify the influence of modeling choices on hemodynamic markers.

- A1 used the raw data exported from the TomTec software and defined the MV and aortic valve (AV) as individually selected faces on the basal surface of the ventricle. The opening and closing of the valve is achieved through the alternating definition of the surfaces as either a wall or a pressure boundary condition, by means of dynamic mesh events.
- A2 used the geometric reconstruction as described in the pre-processing pipeline. The reconstruction included the creation of a rigid basal region, valves defined by diameters and orientation angles (mention which ones, after pre-processing is written) and a transition region between the rigid and the moving part of the domain. The valve dimensions were then fitted to match Doppler ultrasound data on the transmitral velocities.
- A3 considered an additional left atrium and ascending aorta, taken from [8] and scaled to fit the MV and AV sizes. Further on, MV and AV were defined as two-dimensional porous faces, using the approach of Daub et al.[5].
- A4 included a 3D description of the MV as a porous medium allowing to model normal and tangential Darcy losses as introduced in our previous work [23].

The detailed description on the implementation of the different models is explained in the appendix B

Meshing was performed with *ANSYS Meshing* utilizing the patch-conforming method with a uniform linear cell size. A mesh independence study for P1-A4 yielded a core mesh element size of $4e-4$ m. The porous regions were meshed with an element size of $8e-5$ m and five inflation layers at the ventricle wall with an initial height of $2.5e{-}5$ m and a growth rate of 1.1 were set. These settings resulted in mesh sizes of 2-3.2 million elements for all simulations. Transient simulations were performed in *ANSYS Fluent* with a pressure-based solver employing the coupled scheme, least squares cell based discretization and bounded second order implicit transient formulation. Blood was modeled as a Newtonian fluid using the model with a density of 1056.4 kg m-3 and a viscosity of 3.6 mPas. A *hybrid initialisation* method was used. The adaptation of the volume mesh to the movement of the ventricular walls was achieved using a dynamic mesh model with *linear elastic solid smoothing* and a Poisson's ratio of 0.3. The time step size was calculated based on the RR duration, see Table 1, with 1400 time steps per cardiac cycle and a convergence criterion

of 1e-3. The simulations were performed for three cardiac cycles to obtain independent results. For all simulations across different approaches, the same boundary definitions were employed. Figure 2 illustrates the defined boundary regions for reference and shows the approaches A1-A4.

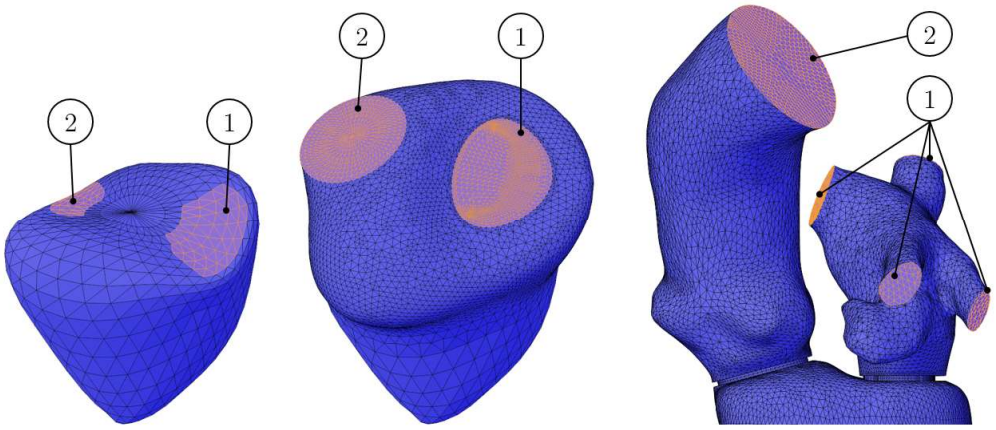


Figure 2: Boundary regions for A1 (left), A2 (center), A3-4 (right). The inlet (1) and outlet (2) are colored in orange and the walls are colored in blue.

In the simulations, closed valves were represented as walls, while open valves were characterized by inlet or outlet pressure profiles. These profiles were obtained from clinical patient data [9, 12]. The transition between boundary conditions during diastole and systole was handled using dynamic mesh events within the *ANSYS Fluent* software. The mesh and cycle independence studies can be found in the appendix C.

2.4 Investigated scenarios

At first, the reconstruction quality of the pre-processing pipeline was quantified by looking at the normalized volume-time curves of the raw and the reconstructed data. In addition, the Hausdorff distance metric according to equations 1 was determined for the EDV and ESV case. This metric is defined as the maximum of the minimum Eucledian distance between two points x,y given the point sets X, Y:

$$HD(X,Y) = \max_{x \epsilon X} \min_{y \epsilon Y} \|x - y\|^2 \tag{1}$$

In the next step, four different MV modeling approaches were compared with each other for P1 regarding their influence on kinetic energy and area-averaged wall shear stresses. One approach was selected and translated to all patients P1-P3. At last, the simulation results were compared to Doppler ultrasound measurements of the transvalvular velocities at the MV and AV and the intraventricular flow structures throughout the cardiac cycle were analyzed.

3 Results

3.1 Reconstruction quality

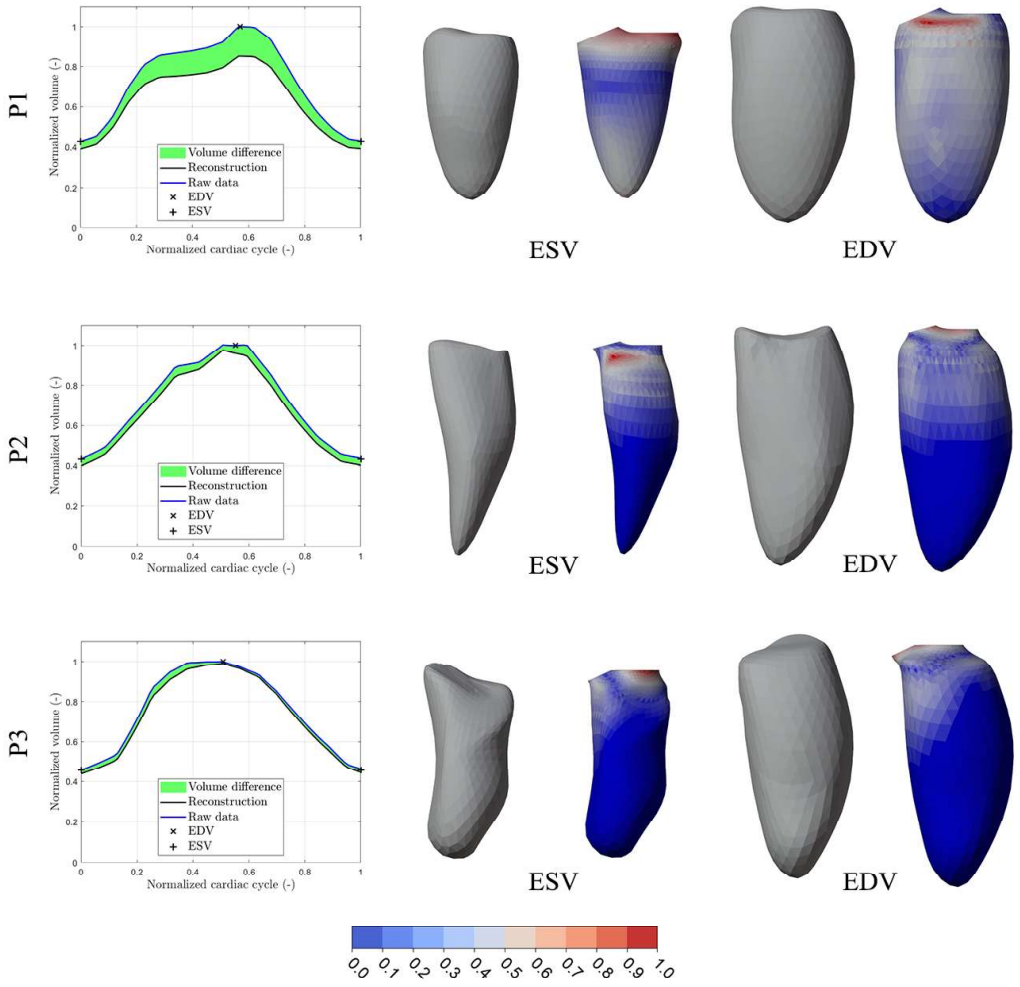


Figure 3: Analysis of reconstruction quality. Left: Normalized volume-time curves of raw data and reconstructed ventricles. Center and Right: Raw data and reconstructed ventricles during end-systole and end-diastole color coded by the normalized Hausdorff distance.

The reconstruction quality of the pipeline is shown in Figure 3 . On the left side of the figure, the normalized volume over the normalized cardiac cycle for both, the raw data (blue line) and the reconstructed data (black line) is shown. Reconstructed data includes all pre-processing steps and describes the stl-files that are exported for downstream CFD simulations. The normalized volume difference is shown in green. P2 and P3 have very small changes between raw and reconstructed data, while P1 exhibits differences of up to 15% at the end-diastole. In the center and on the right side the respective raw data for the ESV and EDV and the reconstructions color-coded with the Hausdorff distance as presented in equation 1 are shown. The values are normalized per patient towards the

maximum distance. It can be observed that for P2 and P3 the differences are seen only in the upper (basal) part of the ventricle. P1 has a biggest offset with Hausdorff distance of 0.5, especially during end-diastole. In all cases, the largest differences are at the valve annuli and surrounding reconstructed tissue.

3.2 Comparison of modeling approaches

Plots of the EDV-normalized kinetic energy ω_{kin} and the area-averaged wall shear stress $\bar{\tau}$ during the diastole are shown for P1 and the four different MV modeling approaches A1-A4 in Figure 4. In early diastole, all approaches show a similar pattern with respect to $\bar{\tau}$, characterised by a small initial peak, followed by a relatively constant phase, and then a steep increase. However, A4 consistently shows higher values for $\bar{\tau}$ during early diastole and an earlier occurrence of the peak compared to other approaches. During end-diastole, the values of $\bar{\tau}$ gradually align, following a similar curve pattern across all approaches. The model without any valve (A1) exhibits a delay for ω_{kin} and $\bar{\tau}$ in comparison to the remaining approaches.

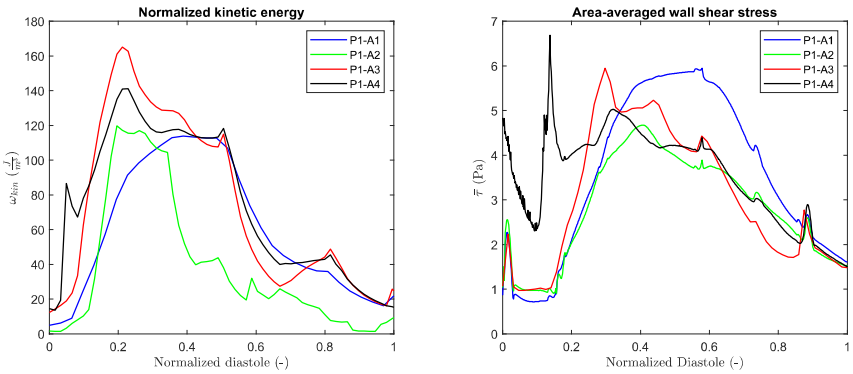


Figure 4: Normalized kinetic energy and area-averaged wall shear stress for P1 and modeling approaches A1-A4.

The porous medium approach A4 further exhibits high values in the beginning of the diastole with strong changes indicating numerical instabilities in this approach. Thus, the simpler approach A3 has been used for further simulations of P1-P3 and the investigation of the intraventricular flow structures.

3.3 Comparison to Doppler ultrasound

The comparison of the transvalvular velocities of CFD simulations and Doppler ultrasound measurements are shown in Figure 5 for the P1-P3, with numerical values shown with solid lines and experimental values shown in dashed lines. The experimental data was linearly interpolated onto the temporal resolution of the CFD data and a moving average window filter was applied to the numerical results to remove low amplitude oscillations of the velocity values caused by the incompressibility of the fluid and the rigid walls of the simulation.

Overall, there is a very good agreement in both, the shape and amplitude of the velocities indicating the correct implementation of the moving mesh setup, the appropriate measurement of the valve dimensions and the possibility for the simplified representation of the mitral valve. There are two noticeable differences between simulation and experiment. For P2, distinct E -and A waves are seen in the computed velocities across the MV

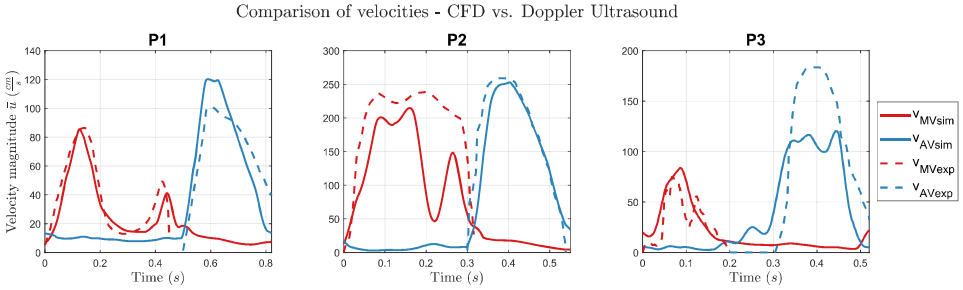


Figure 5: Comparison between CFD and Doppler ultrasound data for MV and AV flow of patients P1-P3. Solid lines: CFD data. Dashed lines: Experimental data

with an E/A ratio of approximately 1.25. This is not observable experimentally and it seems that the A wave is not present at all. For P3, the aortic flow during peak systole is about 60% higher in the experiment when compared to the numerical data. However, the duration and shape of the systole is very similar.

3.4 Application

To further showcase the use of our CFD pipeline, in Figure 6 we present intraventricular blood flow velocities in a cut-plane through the ventricles of P1-P3 for 10%, 25%, 50% and 75% of the diastole and 75% of the total cardiac cycle, respectively. As the ventricles have very different sizes (EDVs ranging from 5.5 - 45 mL), they are not presented on the same scale. For P1, a homogeneous mitral jet is forming during the beginning of the diastole. During mid- and end-diastole, a stable vortex is formed in the central part of the ventricle, that is ejected during diastole. For P2, a much larger mitral jet is formed, as also indicated by the transmitral velocities in Figure 5. Vortex formation is delayed in comparison to P1 with one small vortex core at the apex of the ventricle and a second one during late diastole in the center of the ventricle. P3 does not exhibit any detectable vortex formation and rather irregular flow patterns are seen during the diastole. The scaling of the ascending aorta and the atrium to the respective valve sizes can also be seen for all patients.

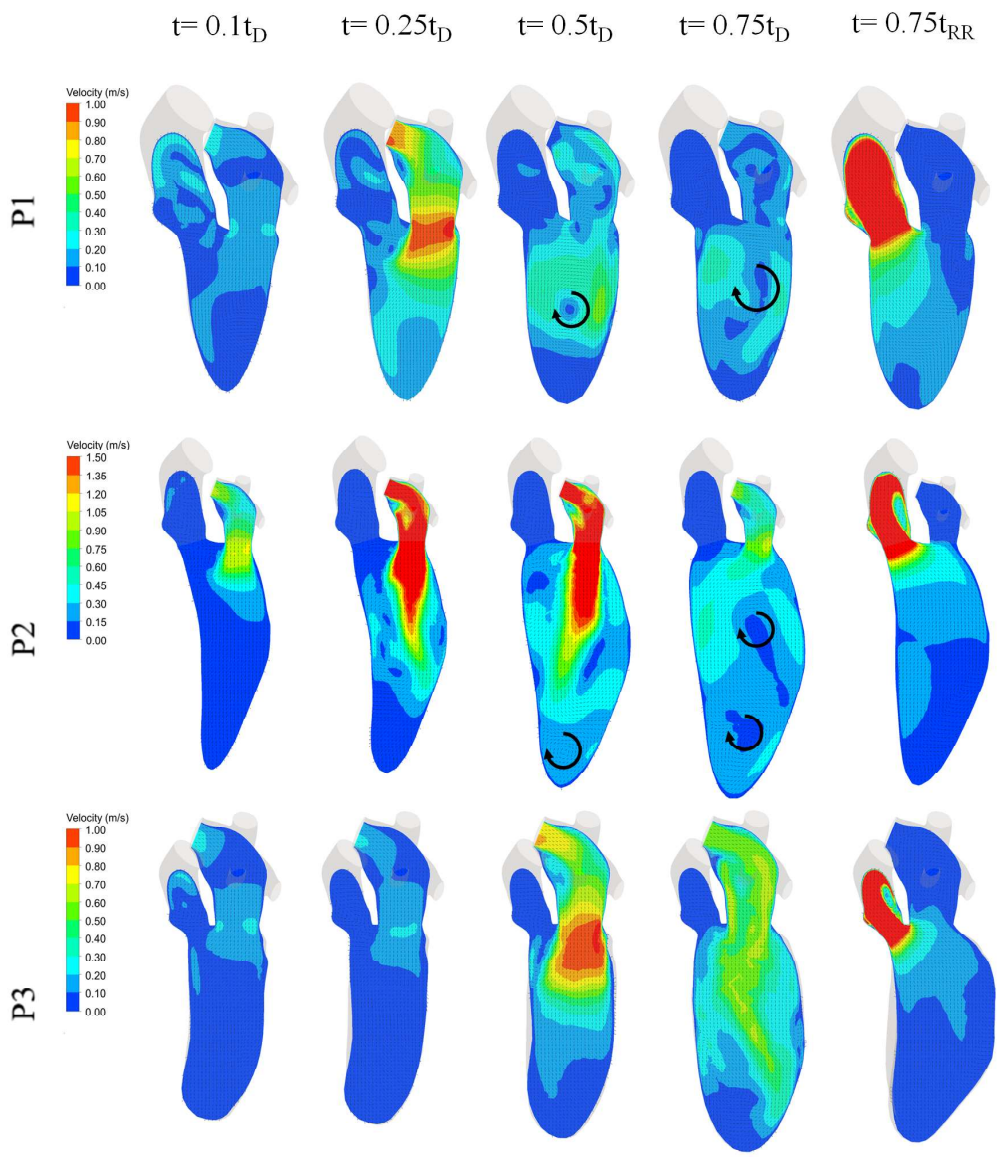


Figure 6: Blood flow velocities for patients P1-P3 throughout the cardiac cycle. Vortex formation is emphasized by black arrows. t_D = diastolic duration, t_{RR} = RR duration.

4 Discussion

The aim of this work was to develop an interactive pipeline to perform image-based simulations of ventricular blood flow. We have used the open-source software *Blender* and created a *Python* add-on which uses segmented ventricular geometries from real-time three-dimensional echocardiography data to execute pre-processing steps such as coordinate transformation, geometry smoothing as well as the incorporation of valves and surrounding vascular structures. Further on, different modeling approaches for the atrioventricular valve can be included for further downstream comparison in subsequent CFD simulations.

The reconstruction pipeline is capable of generating a high-quality representation of the ventricle, including well-defined interfaces for the valves. Temporal interpolation between the individual recordings enables to add valve geometries that can vary over a short time, as it happens during the opening and closing phases. The resulting volumes demonstrate a good correspondence with the raw data. Offsets exist in the case of larger changes of the basal area of the ventricle and the insertion of the valves (see P1 in Figure 3). This is an inevitable consequence of the imaging modality. The segmentation of the RT3DE data focusses on the ventricular movement and both the annuli and valve locations are poorly represented as can be observed in all of the raw datasets.

In the current approach, the geometries of the atrium and aorta were scaled based on valve diameters. However, in the case of congenital pathologies such as the hypoplastic left heart syndrome, the patients typically have an enlarged atrium and a small aorta. Thus, different scaling laws may be implemented in the future, or the geomtries may be adjusted directly to patient-specific data. At last, the valvular plane is fixed and there is no annular movement included. This simplification was necessary to further streamline the model, as such information could not be properly extracted from the clinical data. However, movement of the annuli and/or deformation of the surrounding structures is something that can be implemented in future work.

In summary, four different MV modeling approaches have been compared to each other regarding their influence on kinetic energy and wall shear stresses. We have shown that the MV modeling approach has a large influence on flow derived quantities and should be evaluated on an individual basis, i.e. depending on the specific question at hand. For instance, the simple MV modeling A1 (no valve at all) yields time delays in the kinetic energy and wall shear stress waveforms of 30% of the diastolic duration. We have made similar observations in our previous work on mitral valve hemodynamics in a generic framework [23]. Then, one approach - A3, has been adapted and translated to three datasets from two clinical centers. The valves were sized according to individual measurements performed by the clinicians and yielded a good agreement to aortic and mitral valve Doppler ultrasound velocities. The higher velocities during the diastasis of P2 and peak systole of P3 can be partially attributed to the small geometries investigated. The EDV of P2 was 17 mL and the EDV of P3 was 5.5 mL. As such, measurements of the valves and Doppler ultrasound measurement are connected to high experimental uncertainties. The propagation of these uncertainties and their influence on the simulation output represents an both interesting and urgent topic for further research. Automated geometry processing and simulation preparation is required for such tasks and can be facilitated with the presented pipeline.

It has to be noted, that without regurgitation and under knowledge of the volume-time-curves of the ventricles, the valves can be sized appropriately through simple mathematical operations and directly implemented into the pipeline. We decided to choose

the simpler 2D porous zone approach (A3) and not the 3D porous medium setup for the MV, as unrealistically high values were observed during the cardiac cycle (see Figure 4). We explain this with isovolumetric relaxation and contraction phases, which can cause instabilities and could also be omitted in the simulations.

At last, we have shown intraventricular velocities during the cardiac cycle with flow structures that have also been observed in other studies such as mitral jet formation and intraventricular vortex formation [19, 20, 6] . We have focused on the blood flow velocities, although other important flow quantities such as wall shear stresses or pressure gradients can be computed as well. Other important scalar quantities such as kinetic energy and energy loss [6] can also be determined. As the investigated patients had different pathologies, we decided not to draw further relations between the fluid dynamic results and the clinical observations. The results presented herein are thus intended to showcase the general capabilities. Our pipeline has been developed using ventricle deformations extracted from real-time three-dimensional echocardiography scans. Other imaging modalities such as computed tomography (CT) or magnetic resonance imaging (MRI) can also be included, as our pipeline uses only stl-files as an input. Differences in ventricle simulations between RT3DE and CT/MRI would be an interesting question to explore. In addition, further dynamics such as the movement of the valve annuli or the investigation of right ventricle hemodynamics would be possible future work.

Taken together, we have presented an interactive framework to streamline CFD simulations of ventricular blood flow using patient-specific imaging data. Most of the operations can be performed with minimal user input, supporting reproducibility of computational studies. Common model assumptions, particularly concerning the valve representations, can be easily selected and its effect on the simulation output quantified. The tool has been implemented in a widely used and open-source geometry processing software, which allows it to be freely used and extended by the scientific community. We further supplement its use by detailed video tutorials and full accessibility of the code.

Data availability

The code and tutorials to perform moving mesh simulations is located on `https://github.com/mneidlin/movingmesh`. The *Python* code of the *Blender* add-on is located on `https://github.com/DanielVerhuls/Ventricle-reconstruction-pipeline` with videos on how to use the *IP-Heart* pipeline.

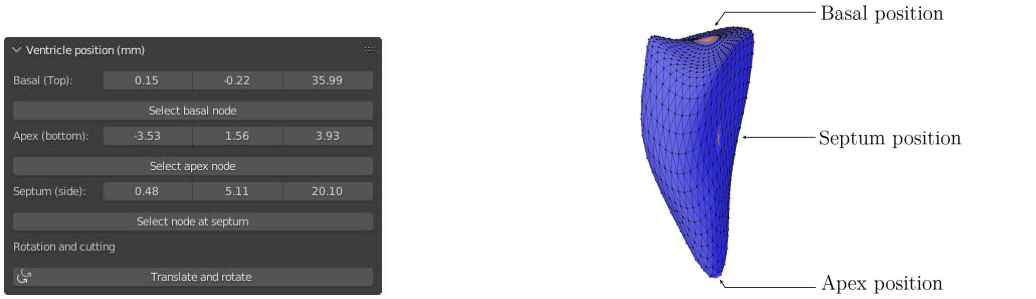
Appendix

A Detailed description of pre-processing pipeline

A detailed description of the individual pre-processing steps is shown below.

A.1 Coordinate transformation

In order to standardize the algorithm, the objects imported from stl-files, which are generated from ultrasound data, are required to be transformed from their local coordinate system to a global and uniform coordinate system. This transformation is achieved by selecting three nodes of the ventricle, which can be accomplished either through the use of buttons located in the *Ventricle position* panel while in edit mode in *Blender* or by entering the coordinates manually into text edits found in the same panel. Figure S1 shows the



(a) Panel for the ventricle position.

(b) Node placements on the surfaces mesh.

Fig. S1: Setup for the coordinate transformation using three distinct nodes on the ventricle surface.

setup used to perform the coordinate transformation. The panel displayed in figure S1a of the *Blender* user-interface contains buttons and text edits for selecting the three vertices, as well as a button for executing the coordinate transformation, as detailed in [16]. The selected nodes are used to determine the alignment of the object after the translation and rotation. They are shown in figure S1b. The apex node of the ventricle is employed to move the coordinate system to the origin (0,0,0) of the global coordinate system. The remaining two selected nodes are used for rotating the object. The top position of the ventricle which always lies central at the basal side of the ventricle geometry and will be rotated such that it has the coordinates $(0,0,z_b)$ With z_b being the distance between the apex and basal position. Finally the septum position is used for the last rotation around the z-axis to align it with the global y-axis.

A.2 Create basal ventricle region with valves

The next stage in the pipeline involves generating the basal region, which is situated in the upper region of the left ventricle and is directly linked to both heart valves.

A.2.1 Choose reference ventricle

To create the basal region a reference geometry is chosen from a list of selected ventricle objects. Two approaches have been tried to choose this reference ventricle. One is to choose the object with a volume closest to the mean of all object volumes and the other is to choose the object with the largest volume.

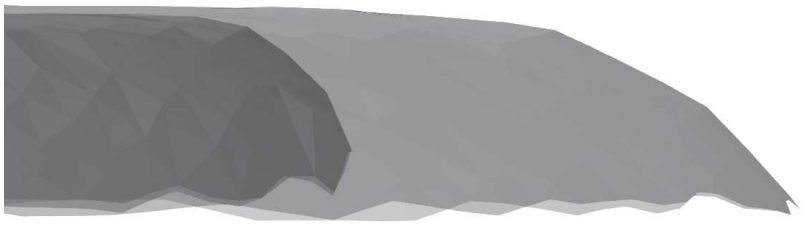


Fig. S2: Edge of the basal region created for reference using mean volume (left) and maximum volume (right).

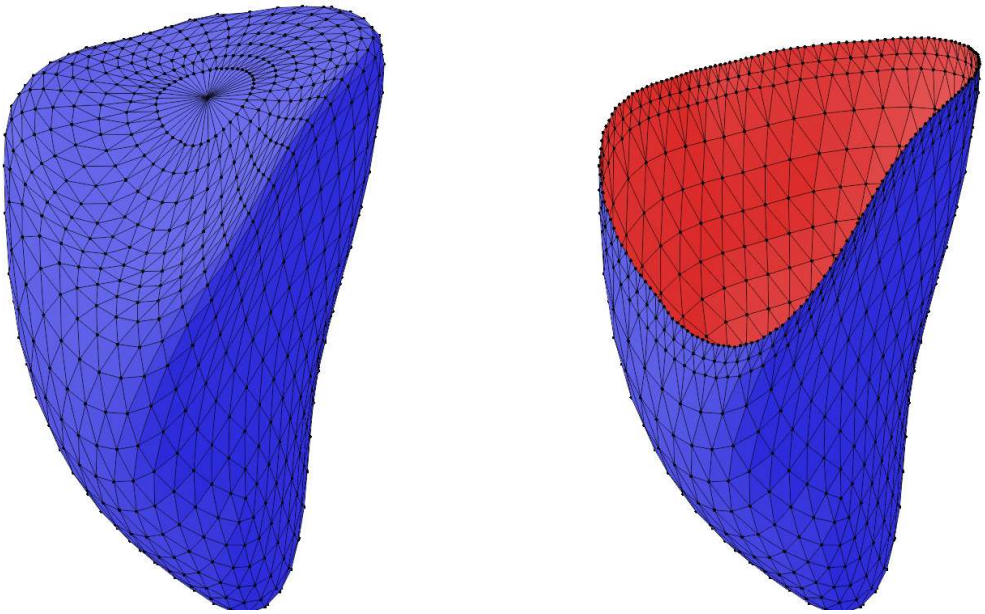
The comparison of the resulting basal regions of the two approaches is depicted in figure S2. The mean volume approach creates the basal region for a smaller object resulting in a smaller basal region where the lower faces are more curved towards the apex when compared to the other approach. Experience shows, that a more open basal region as created in the maximum volume approach creates a better reconstruction of the connection between the apical and basal region for most cases. Using the maximum volume results not just in a larger basal region in general, but also to an offset between apical and basal region for small apical regions. This offset can be smoothed out, which removes the downside of this approach.

A.2.2 Removal of basal nodes

Once the reference object is selected, a duplicate of it is made to perform additional operations. The first step is to remove vertices in the basal region of the reference object to create space for the valve and support structure nodes. This removal process is shown in figure S3 comparing the raw ventricle geometry before and after removing the basal region in figures S3a and S3b respectively. The removal algorithm is also used for the connection between the apical and basal regions, which will be described in detail in chapter A.3.1.

A.2.3 Insert valve interface nodes

The next step involves adding nodes of the heart valves into the geometry. This is done by accessing the valve geometry objects for the AV and MV, which are stored in the *Blender* project. These objects are then scaled, rotated, and translated as defined in the valves panel before being added into the geometry by copying them into the previously created copy of the reference object.
The valve options panel, as shown in Figure S4, allows for individual selection of the position, angles, and radii of the AV and MV. The long mitral radius is the intercommissural radius r_{MVic} and the small mitral radius is the anterior-posterior radius r_{MVap}. The annular height scaling is determined as the middle between the commissural and anterior-posterior scaling. For the AV only one radius r_{AV} needs to be selected since it is simplified as a circular shape. In contrast, the MV has a semi-elliptic shape and can be modified in both directions for size and shape. The *long mitral radius* is the distance from the valve-center to the annulus at the commissures and the *small mitral radius* is the distance from the valve center to the outer annulus ring in the direction of the leaflets. The panel also includes a checkbox for selecting A4 for the MV. The button *Add valve interface nodes* adds the nodes at the interface between valves and the ventricle into the current object. *Build support structure around valves* adds an upscaled and downscaled version of the valve nodes and places them at the same position and angle as their respective valve interface nodes. These up- and downscaled valve nodes are used to improve the quality



(a) Before removal (b) After removal

Fig. S3: Removal of the basal region of a raw data ventricle geometry.

of the Poisson surface reconstruction algorithm by smoothing the created surface around the valve nodes because it reduces the point normal gradient on the interface nodes.

A.2.4 Surface reconstruction

The surface reconstruction creates a closed surface object for the ventricle with a given point cloud object consisting of the apical region, valve and support structure nodes.

Figure S5 illustrates the progress during the surface reconstruction. For each point in the point cloud a vector is computed oriented in normal direction to a virtual surface tangent to the locally surrounding points. The point coordinates and normalized orientation data, along with the maximum octree depth D, are used as inputs for the Poisson surface reconstruction, resulting in a surface object of the ventricle, as shown in figure S5a. The quality of the created object is sensitive to the placement of the valve nodes. Large empty spaces between points, such as valve points far away from ventricle points, or large gradients between point normals, such as valves with sharp angles, can lead to open surfaces. When the object is not closed by the surface reconstruction the input parameters need to be changed such that it creates lower point normal gradients in the basal region.

To more evenly space the face sizes and reduce the number of skewed cells with sharp angles the object is remeshed. For this, the *Blender* voxel remesh modifier is used. Once the object is remeshed, the *Blender* merge operator is applied to remove nodes that are too close to each other, using a threshold value. This reduces the number of surface nodes on the ventricle and improves the performance. Figure S5c shows the resulting object after the merge operator has been applied.

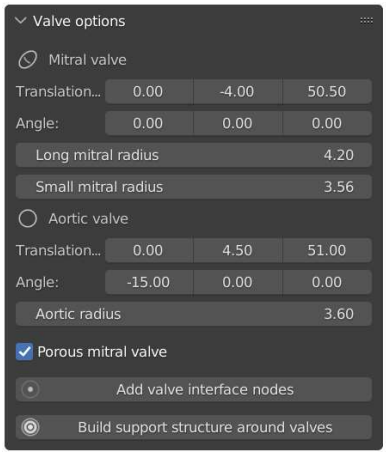
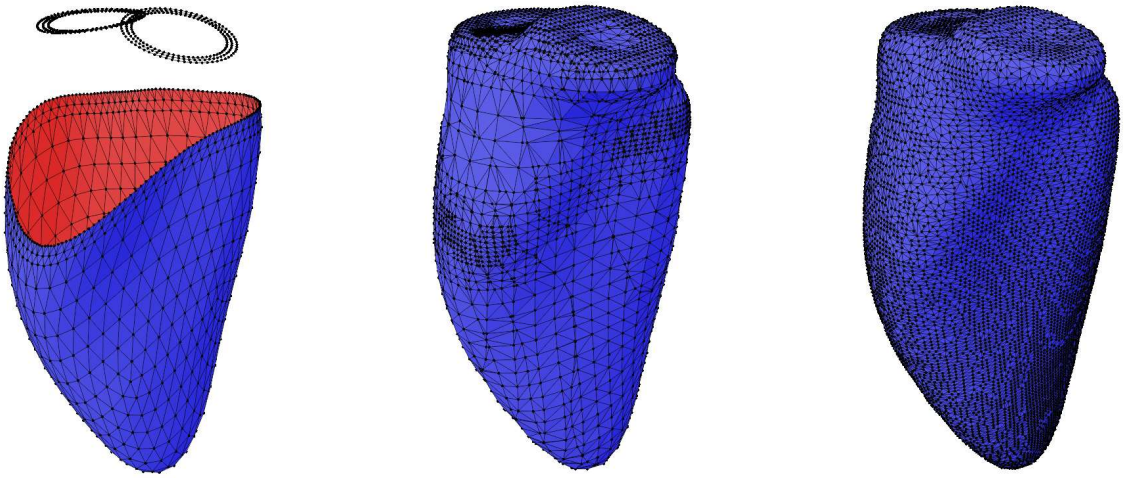


Fig. S4: Panel for the selection of valve parameters and the manual creation of valves.



(a) Before Poisson algorithm

(b) After Poisson algorithm

(c) Remesh and refinement

Fig. S5: Step-by-step visualization of the surface reconstruction.

A.2.5 Removal of apical region

After refining the surface mesh, the next step involves removing the apical region created during the surface reconstruction, which will be replaced with the original apical region nodes in a later step. This is achieved by removing all vertices with a z-coordinate below a threshold h_{th}, which lies between the valves and the apical regions.
Figure S6 shows the interface nodes of both valves above apical regions for all ventricles with the AV interface nodes on the left in an angle and the MV interface nodes in a horizontal line on the right. The threshold lies between the highest z-coordinate of nodes of all apical regions h_{max} and below the lowest vertex z-coordinate of any interface valve node h_{min}. The threshold value was chosen as follows:

$$h_{th} = \frac{b * h_{min} + h_{max}}{b + 1} \tag{2}$$

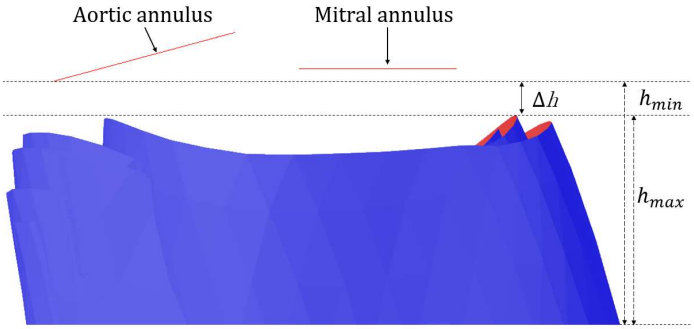


Fig. S6: Side view of all objects before Poisson surface reconstruction including the mitral and aortic annulus.

with b being a scaling factor to shift the threshold closer to the basal region, while it remains between the two points. Multiple values for b have been tested. Larger values create more obtuse angles in the basal region, but getting too close to the valve interface nodes reduces the quality of the connection between the apical and basal regions. If apical nodes lay above the valve interface nodes, the user will be informed and the execution is prohibited. There will also be a warning displayed when these values are very close to one another.
When deleting nodes coinciding faces will also be deleted. For nodes barely below the threshold parts of the deleted face lay above the threshold. When these deleted nodes are neighbouring ones just above the threshold, the deletion process creates spikes at the last edge loop. So after the removal of the apical nodes the lowest edge loop of the basal region will be smoothed. Without doing this, the connection between apical and basal region would become corrugated.

A.2.6 Insertion of valves

The final step in the creation of the basal region is the insertion of the MV and the AV. The algorithm for inserting valves is visualized in figure S7 starting with the basal region created in the previous chapter in figure S7a. The first column shows the removal of nodes in the valve region to create an orifice for valve insertion. To do this, the nodes in the surrounding region of the valves are selected and removed. Each valve has its own geometry with a low surface normal gradient encapsulating all valve interface nodes. This geometry is placed in the same position where the valve is to be inserted as seen in figure S7b, and all basal region nodes inside it are selected and removed to create an orifice edge loop shown in figure S7c. The surface normal gradient is kept low to detect if a node lies in the given region. After creating the orifice in the first column, the valve can be inserted into the basal region, as shown in figure S7d. For A5, multiple versions of the basal region are created and filled with the given timeframes of the MV geometry. The remaining gap between them is then closed by connecting the orifice edge loop with the valve interface nodes as seen in figure S7e. Smoothing the orifice edge loop before connecting improves the quality of this connection.

A.3 Connection between apical and basal region

This part of the algorithm exchanges the basal regions from the raw data with ones created in the previous chapter. First the old basal regions must be removed leaving only the apical regions. Each apical region is assigned a basal region leaving a gap in between. This gap is then closed by adding edges between the two regions connecting both orifice edge loops.

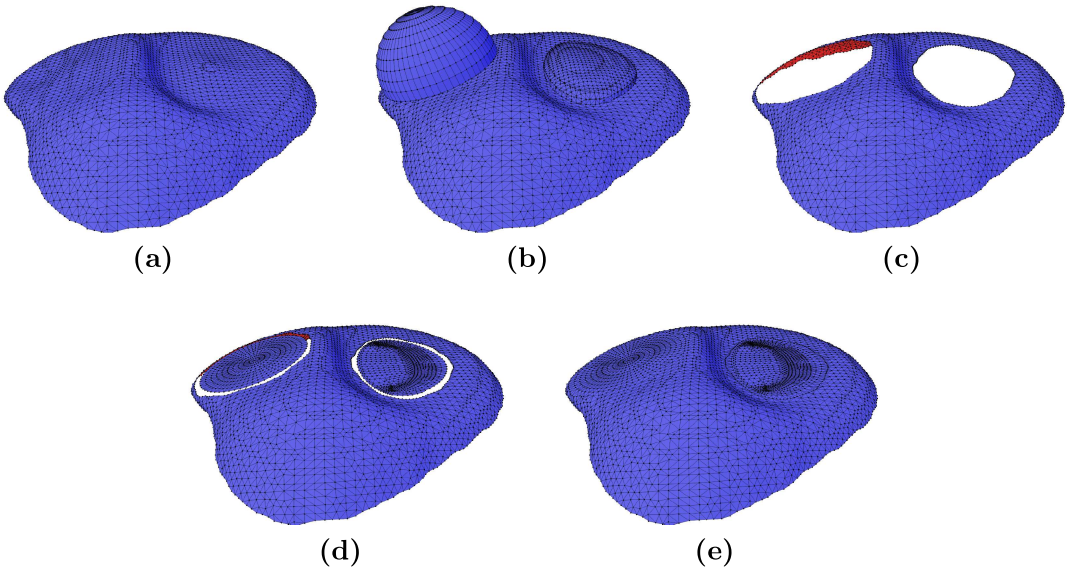
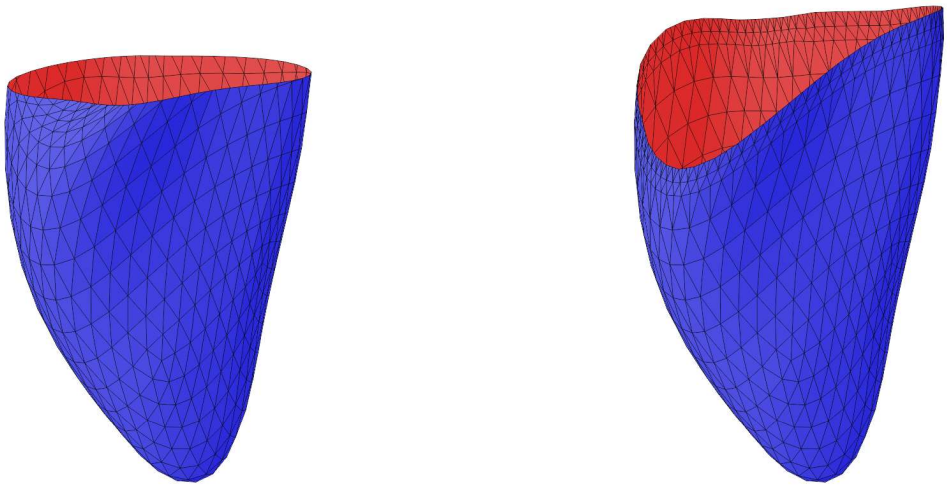


Fig. S7: Step-by-step visualization of the valve insertion algorithm: **(a)** Initial basal region **(b)** objects to choose nodes in valve region added **(c)** remove valve region nodes **(d)** add valve surface meshes and **(e)** connect basal region and valves.

To ensure consistency of this connection, the connection algorithm is performed on a copied reference ventricle. The connecting edges are then copied to all timeframes. This is done to avoid the tools used in the creation of the connection from connecting the nodes differently for different timeframes of the ventricle. Lastly the surfaces of the connection will be refined and triangulated using the same idea of copying the node connectivity of a reference ventricle.

A.3.1 Remove basal nodes

As a preparation for the connection algorithm the old basal regions of each ventricle must be removed to make room for the basal regions created in the previous step. Different methods were attempted to remove the basal region, as illustrated in figure S8. The first iteration of the algorithm, as depicted in figure S8a, utilized the *Blender* boolean modifier in difference mode with a cube located in the basal region. The lower surface of the cube acted as a threshold for the removal of nodes, but also all edges from the original object intersecting with the cube create a new node at that position resulting in a flat edge loop on top. The unpredictability of the creation of these edge loop leads to problems when applying it to time-dependent geometries. The node connectivity can not be assured using this approach and it also creates nodes that may be very close to one another, leading to skewed faces around these points. An approach keeping the node connectivity consistent between timeframes is shown in figure S8b. Using the basal position node as chosen in chapter A.1 nodes of the surrounding edge loops are deleted creating the surface shown in figure S8b. After that, the next edge loop which now lays on the created orifice is refined in two steps to create a smooth transition from the coarse mesh in the apical region to the finer mesh in the connection and basal region.

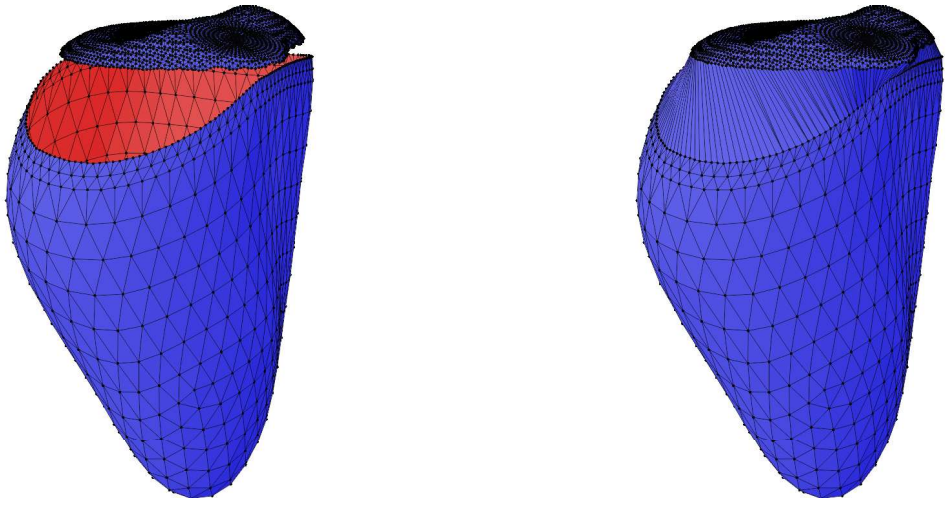


(a) Remove nodes below horizontal plane (b) Remove edge loops around basal position

Fig. S8: Approaches for the removal of the basal region.

A.3.2 Create reference connection

To ensure a uniform connection between the basal and apical regions across different timeframes, a reference connection is created for a single time frame. This reference connection can then be replicated for the other timeframes to create a consistent mesh The algorithm for creating a reference connection involves copying the apical region and a



(a) Before connection (b) After connection

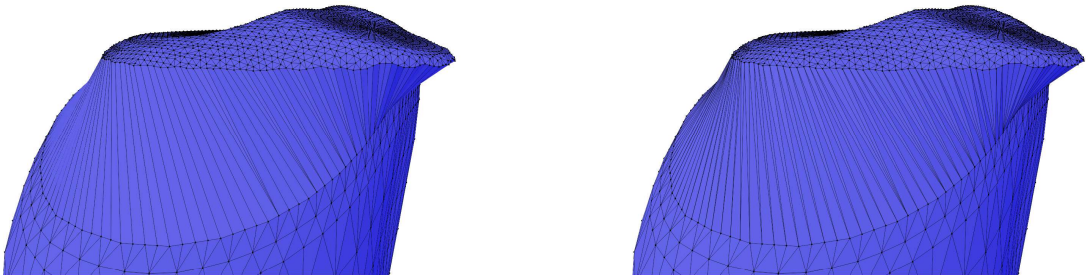
Fig. S9: Apical and basal region before and after connection.

basal region and combining them into a single object. This is shown in figure S9a. Next, the lower edge loop of the basal region and the upper edge loop of the apical region are connected using the bridge algorithm from the Looptools *Blender* add-on. This creates squared surfaces for edge loops with the same number of nodes, and triangular faces when the number of nodes between the edge loops differs. The resulting surface is shown in

figure S9b. Squared connection faces are preferred because they create fewer new elements and lead to less skewed cells when triangulating the surface. This reference connection can then be copied and used for later timeframes to ensure a uniform connection between the basal and apical regions.

A.3.3 Refine reference connection

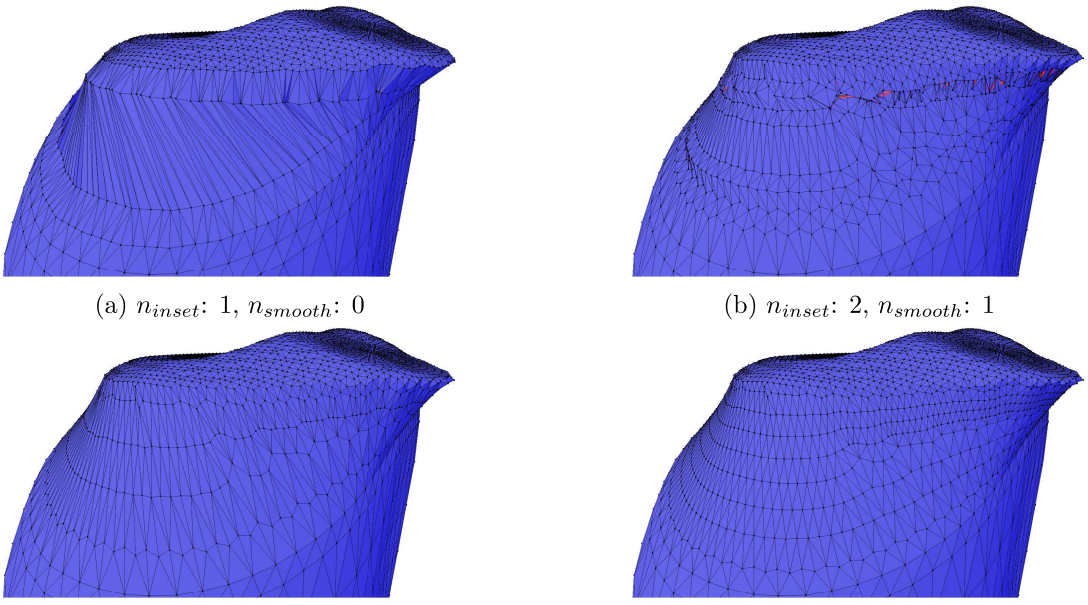
The surfaces generated by the connection algorithm span a gap that is much larger than the distance between the nodes of the edge loops, resulting in faces with edges along the connection that are much longer than those along the edge loops.



(a) No refinement (b) No refinement triangulated

Fig. S10: Connection between apical and basal region without refinement.

Figure S10 depicts a comparison between the original connection created in figure S10a and the automatically generated triangulation when exporting objects from *Blender* in stl-format, shown in figure S10b. As can be seen, the triangulation results in triangles with unevenly sized edges, highlighting the need for a refinement of the connection. The



(a) n_{inset}: 1, n_{smooth}: 0 (b) n_{inset}: 2, n_{smooth}: 1

(c) n_{inset}: 1, n_{smooth}: 1 (d) n_{inset}: 2, n_{smooth}: 2

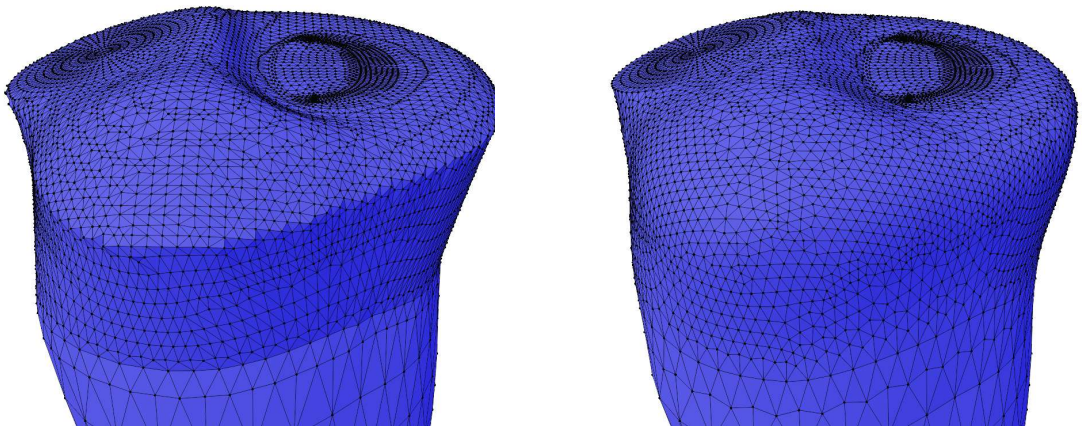
Fig. S11: Refinement of the connection between apical and basal region.

refinement algorithm is an iterative process that involves inseting and smoothing faces

from the connection region, using the number of inset iterations n_{inset} and the number of smoothing iterations n_{inset}. The resulting faces are then triangulated using the same algorithm used in the stl-export by *Blender*, with the newly created edges being saved for later use in maintaining node connectivity across timeframes. In figure S11, a partially executed refinement algorithm is shown, with new faces being inset with a shift from existing faces at a distance defined in figure S6. The shift distance is reduced by a factor of 2 for each continuous iteration to ensure that the new faces remain within bounds between the apical and basal regions and between the new faces created in the previous iteration. Inseting faces may result in an object with unclosed surfaces, as shown in figure S11b, where red faces have negative surface normals compared to blue faces. However, applying a smoothing algorithm to these nodes resolves this issue and results in an object with all surface normals facing in the correct direction, as shown in figure S11d.

A.3.4 Copy connection and triangulation from reference

The connection and it's refinement created in the previous two chapters are used to project them to all ventricle geometries. The timeframes for placing each basal regions in A5 are computed using t_{RR} and t_D. After that, each apical region is connected with a copy of the affiliated basal region. The edges saved during the creation of the connection in the reference ventricle are then used to connect the apical and basal region for each ventricle. The connection is then refined using the algorithm presented in the previous chapter. Deriving the edges created during the triangulation of the reference connection, a triangulation is applied to every ventricle. The connection and the basal region of each ventricle are smoothed in a final step to remove spikes, especially around the valve regions. Figure S12 shows the ventricle before and after smoothing. Before smoothing a hard edge



(a) Before smoothing (b) After smoothing

Fig. S12: Smoothing of the basal and connection region.

lies at the top and bottom of the connection region in figure S12a. The faces on the upper edge switch between facing inward and outward leading to spikes in the geometry. These can cause problems during the simulation with dynamic meshing creating negative cell volumes. Smoothing the nodes in the connection and basal region removes the hard edges on both sides as shown in figure S12b. The algorithm smoothes nodes iteratively, where in each iteration one more edge loop on the lower side of the connection is selected, starting with all connection and basal region nodes except the valve nodes. The smoothing is done iteratively with lenient smoothing parameters to avoid over-smoothing the basal region while leaving the apical region unchanged. Smoothing the geometry with strict smoothing

parameters in a single iteration does not remove the hard edge on the lower side of the connection.

A.4 Surrounding tissue and vessels

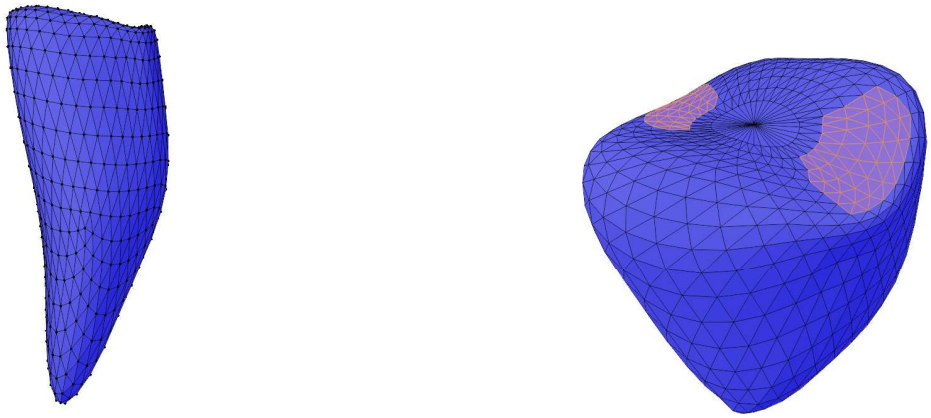
The last step of the *Blender* pipeline is the addition of surrounding tissue of the left ventricle. This includes the MV and AV as well as the aorta and atrium. Each of these has reference objects present in the provided *Blender* project and are shown in figures S15b and S15c. They are copied similarily to the valves using the valve positions, angles and radii to places, rotate and scale the objects.

B Detailed description of modeling choices

A detailed description of how the modeling choices A1-A4 have been implemented is shown below.

Raw data (model A1)

The initial approach, abbreviated with A1, for modeling the MV and AV involves selecting certain faces in the basal region of the ventricle geometry as inlet and outlet boundaries for the valves. These selected faces alternate between being a solid wall or an open surface during the simulation, depending on the phase of the heart cycle. During diastole, the MV is modeled as a pressure inlet and the AV as a wall, and during systole, this is switched to a wall and pressure outlet boundary. This approach provides a simplified representation of the opening and closing of the valves.



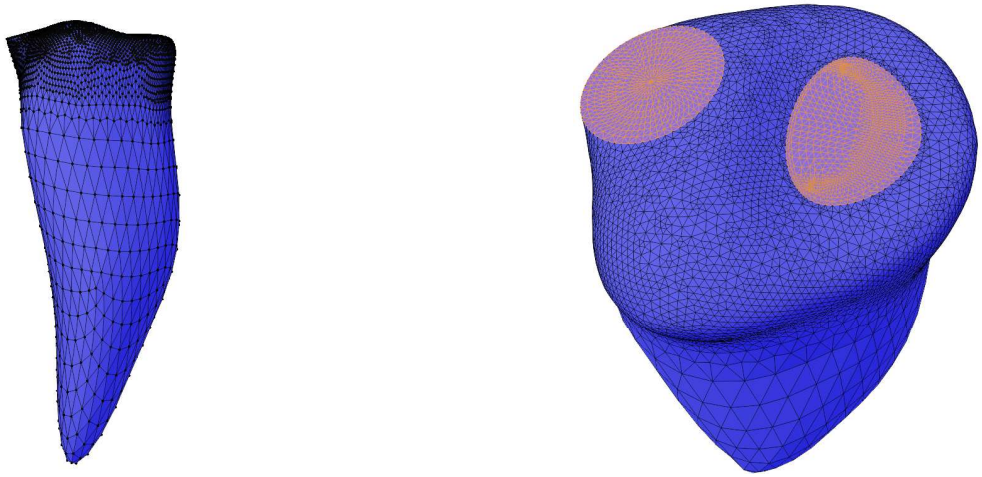
(a) Frontal view (b) View of the basal region

Fig. S13: Surface mesh of A1. Wall surfaces are colored blue and valve surfaces in orange.

Figure S13 shows an example of a geometry used in A1. On the left figure S13a the frontal view of the ventricle is shown and the right figure S13b shows the selected faces used as valves colored in orange.

Geometric reconstruction (model A2)

In the second approach, abbreviated with A2, the geometry of the ventricle is modified to more accurately represent the shape of the valves in the basal region, which is lost when the geometrical data is recorded with ultrasound. The geometric reconstruction was performed using the Poisson surface reconstruction algorithm. The valve surfaces were modeled similarly to the annuli of the respective heart valve as shown above. Similar to the previous approach, the valve regions are switched between wall or open surfaces depending on the cycle state. Figure S14 presents a geometrically reconstructed ventricle viewed from the front and from above. The valve inlet and outlet faces are colored in orange in figure S14b. This approach provides a better representation of the aortic and mitral annulus contour, making it both more realistic and easier to select inlet and outlet boundaries when preparing the simulation setup.



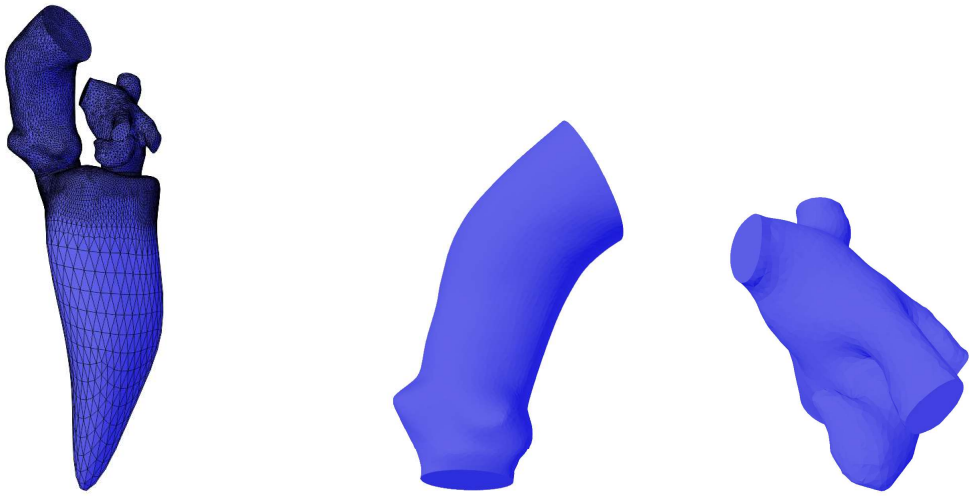
(a) Frontal view (b) View of the basal region

Fig. S14: Surface mesh of A2. Wall surfaces are colored blue and valve surfaces in orange.

Aorta, atrium, and 2D porous jump (model A3)

The third approach, abbreviated with A3, is an extension of the second approach, where the surrounding tissue and vessels are added to the geometry. This includes the atrium, aorta, and the valves. The valve region is represented by disks in the form of the valve interface as seen in figure S14b, which are extruded in the normal direction and are assumed to be hollow for this approach. The simplified opening and closing of the valves is achieved through the instantaneous change in interface conditions applied to the bottom side of the disks connecting them to the ventricle. The interface is set up as a porous jump type and is either permeable or impermeable [5].

Figure S15 illustrates the ventricle with the connected surrounding tissue and vessels, including the atrium, aorta, and valves. The aorta and atrium used in this approach were derived from the supplementary material of Goubergrits et al. [8], with the aortic arch removed as it is assumed to have a low impact on the transmitral and intraventricular flow. The computational cost is further lowered by reducing the surface node resolution.



(a) Frontal view (b) Aorta (c) Atrium

Fig. S15: Surface mesh of A3 in assembled state compared with the separate aorta and atrium.

Aorta, atrium, and 3D porous medium (model A4)

The fourth approach, abbreviated with A4, uses the same geometries as the previous one. Valves are not simplified as walls or open boundaries, but as porous zones where the permeability changes depending on the cycle state and region of the disk, allowing a regulated transmitral flow. The verification of this model considering the pressure drop along the valves was done by Thiel et al. [23] comparing it with the analytical solution and a fluid-structure-interaction approach.

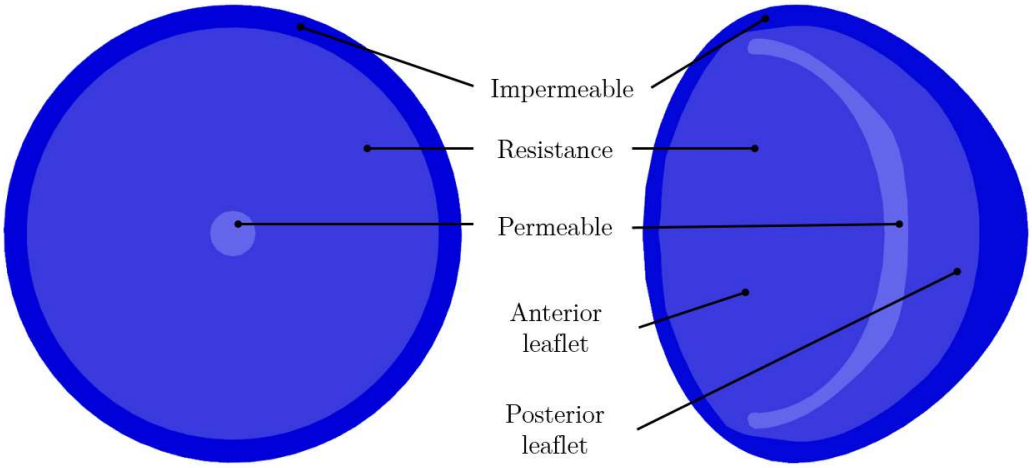


Fig. S16: Zone distribution of porous aortic (left) and mitral (right) valve with each separated into a permeable, resistance and impermeable region.

Figure S16 shows the disks inserted between the ventricle, atrium, and aorta. The AV is simplified as a circular shape and the MV is a 2.5D disk with the outer shape as

defined in A2. The MV is simplified in a way that its asymmetry remains. The disks are separated into three regions. An outer impermeable, intermediate resistance, and inner permeable zone. The outer zone is impermeable acting as a solid wall in the simulation. The two inner zones of each valve change their permeability depending on the stage of the heart cycle. The permeable zone allows transmitral flow for a longer duration than the resistance zone because it opens earlier and closes later.
The heart cycle is separated into diastole and systole, which are individually separated into four phases, an isovolumetric phase, a valve opening phase, a phase where the valve is fully open, and a valve closing phase. The duration of the isovolumetric phases is determined in relation to the RR duration of a heart cycle t_{RR}, based on the typical duration of these phases. The opening and closing times of the AV [10, 17] and the MV [17, 15] were analogously determined in relation to t_{RR}. Together with the duration of either the systole t_S or the duration of the diastole t_D, all phases can be determined.

Darcy's law was employed to model the porosity of the cell zones. This law is commonly used for laminar flows with low Reynolds numbers and connects the pressure drop across a porous medium to the fluid's velocity [23]. To accommodate the dynamic head at high velocities, a correction for inertial losses called the Forchheimer term was incorporated. The equation for the pressure drop Δp in direction i reads as follows:

$$\Delta p_i = \Delta p_{i_{in}} + \Delta p_{i_{visc}}, \tag{3a}$$

$$\Delta p_{i_{in}} = C_{i_{in}} \frac{\rho}{2} \Delta m_i u_i^2 \Delta x_i, \tag{3b}$$

$$\Delta p_{i_{visc}} = C_{i_{visc}} \Delta m_i \mu u_i \Delta x_i, \tag{3c}$$

with $\Delta p_{i_{in}}$ and $\Delta p_{i_{visc}}$ being the inertial and viscous pressure drop due to viscous resistance and inertial losses respectively. $C_{i_{visc}}$ is the viscous resistance, $C_{i_{in}}$ the inertial resistance along axial direction, Δx_i the length of the chanel and u_i the velocity in the given direction. ρ denotes the density of the liquid and μ the dynamic viscosity. The resistance was chosen, such that they only change for a given direction and stay high at a value of k_{visc} and k_{in} respectively for the other two directions. This way a unidirected flow is created along the valves. The direction of flow is normal to the aortic and the mitral annulus plane, and for the MV it is angled 6° towards the ventricle wall to account for the asymmetric length between the anterior and posterior leaflet. To mimic the opening and closing of the valve the values for $C_{i_{in}}$ and $C_{i_{visc}}$ are time-dependent using the following functions

$$C_{i_{visc}}(t) = \begin{cases} k_{visc} \times e^{(-k_{fast}(t-t_{ivr}))} & \text{for } t_{ivr} \leq t < (t_{ivr} + t_{mvo}), \\ 1 & \text{for } (t_{ivr} + t_{mvo}) \leq t < (t_D - t_{mvc}), \\ k_{visc} \times e^{(k_{slow}(t-t_D))} & \text{for } (t_D - t_{mvc}) \leq t < t_D, \\ k_{visc} & \text{else.} \end{cases} \tag{4}$$

$$C_{i_{in}}(t) = \begin{cases} k_{in} \times e^{(-k_{fast}(t-t_{ivr})} & \text{for } t_{ivr} \leq t < (t_{ivr} + t_{mvo}), \\ 0 & \text{for } (t_{ivr} + t_{mvo}) \leq t < (t_D - t_{mvc}), \\ k_{in} \times e^{(k_{slow}(t-t_D))} & \text{for } (t_D - t_{mvc}) \leq t < t_d, \\ k_{in} & \text{else.} \end{cases} \tag{5}$$

The rows in equations 4 and 5 represent the opening, open, closing, and closed state of the valves. The open state of the MV was simplified as a static open valve neglecting the effects during diastasis and atrial contraction. The constant values for $k_{visc} = 2.11 \cdot 10^{11} \frac{1}{m^2}$ and $k_{in} = 2.11 \cdot 10^{10} \frac{1}{m}$ were chosen such that the mass flows through the valves is hindered by a large momentum sink almost nullifying all mass flow. The equations shown are for the permeable MV region during diastole. For the AV, these times would change to fit the systole. The constant parameters $k_{fast} = 900 \frac{1}{s}$ and $k_{slow} = 700 \frac{1}{s}$ were chosen such

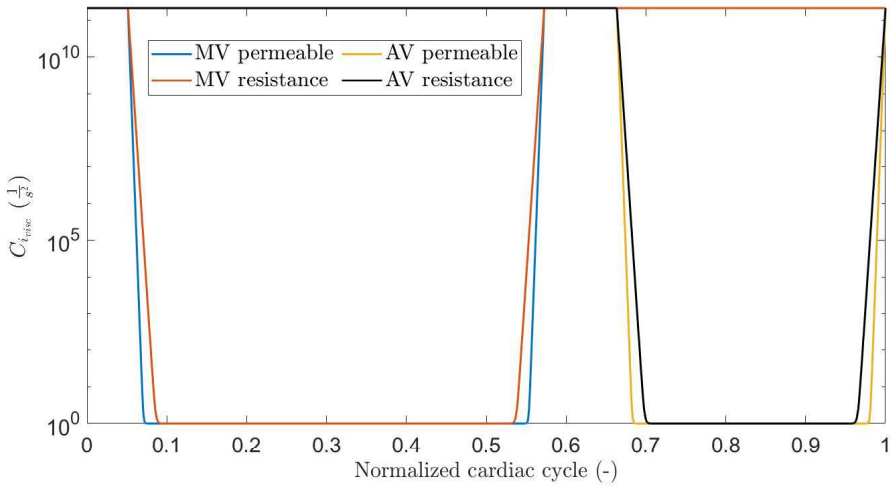


Fig. S17: Visocous resistance $C_{i_{visc}}$ for the permeable and resistance region of the MV and AV.

that $C_{i_{visc}}(t)$ approximately reaches 1.05, a value 5 % above its minimum, after 50 % and 90 % of the openeing or closing time respectively. This allows for a faster opening of the permeable region than the resistance region. The permeable regions open faster and close slower than the resistance region, so between the permeable and resistance region these constants are switched. Figure S17 shows the resulting function for $C_{i_{visc}}$ for a MV permeable and resistance region. The value stays at k_{visc} for the isovolumetric relaxation. After that, the permeable part quickly drops followed by the resistance function. When the valve is fully open the value stays at one and when closing the resistance part closes quicker than the permeable region. After the diastole, the value stays at k_{visc} to emulate a closed valve.

Phase	Definition	
Diastole	$t_D = t_{RR} - t_S$	(6)
Isovolumetric relaxation	$t_{ivr} = \frac{t_{RR}}{20}$	(7)
MV opening	$t_{mvo} = \frac{t_{RR}}{14}$	(8)
Open MV	$t_{mv} = t_D - t_{ivr} - t_{mvo} - t_{mvc}$	(9)
MV closing	$t_{mvc} = \frac{t_{RR}}{11}$	(10)
Systole	$t_S = t_{RR} - t_D$	(11)
Isovolumetric contraction	$t_{ivc} = \frac{t_{RR}}{11}$	(12)
AV opening	$t_{avo} = \frac{t_{RR}}{10}$	(13)
AV open	$t_{av} = t_S - t_{ivc} - t_{av} - t_{avc}$	(14)
AV closing	$t_{avc} = \frac{t_{RR}}{14}$	(15)

Table S1: Definition of the duration of the heart cycle phases used for simulations.

C Mesh and cycle independence

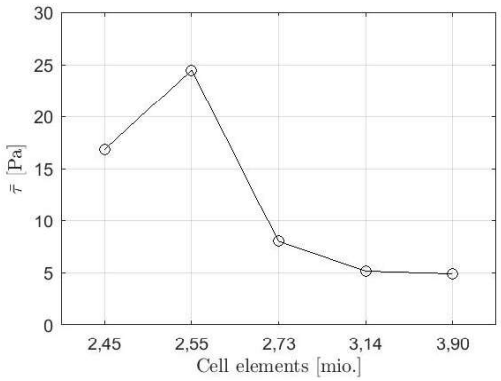
Mesh independence

A mesh-independence study was conducted using the method presented in the work of Celik et al. [2] to estimate the discretization error for a selection of variables that will be important in the subsequent simulation studies. These variables include the mean volume-average kinetic energy $\bar{E}_{kin}$, mean area-average ventricle wall shear stress $\bar{\tau}$ and the axial velocity $|\bar{u_z}|$ along a horizontal plane below the MV. The mesh independence study focused on A4, as it represents the most complex simulation setup capable of producing realistic results. Five refinement levels were analyzed, where the core cell element size was iteratively reduced by a factor of approximately 1.3 for each level, starting with an initial value of 1e-03 m.

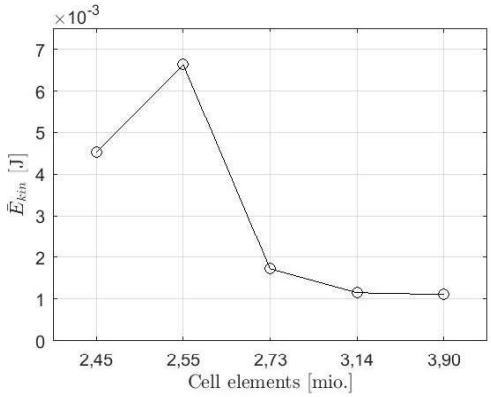
Table S2: Mesh independence study settings for each refinement level.

Refinement level	Core mesh element size (m)	Nodes (mio.)	Elements (mio.)
0	1 e-03	0.526	2.447
1	7.5 e-04	0.548	2.554
2	5.5 e-04	0.598	2.729
3	4 e-04	0.718	3.139
4	3 e-04	0.938	3.903

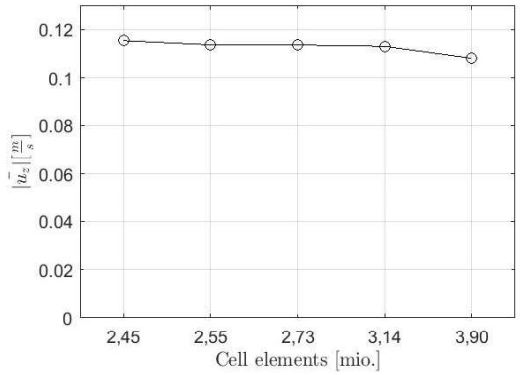
The resulting number of nodes and cell elements for each refinement level is presented in table S2. For the coarse refinement levels, the number of mesh elements is dominated by the inflation layers and the refinement region around the valve disks, while the later refinement levels significantly increase the cell element number of the core mesh. Based on the results depicted in figure S18, refinement level three was selected for further simulations due to the constant behavior observed for $|\bar{u_z}|$ across all refinement levels. Furthermore, the results for $\bar{E}_{kin}$ and $\bar{\tau}$ exhibited only minor deviations between refinement levels three and four, supporting the decision to choose refinement level three as the optimal compromise between accuracy and computational efficiency.



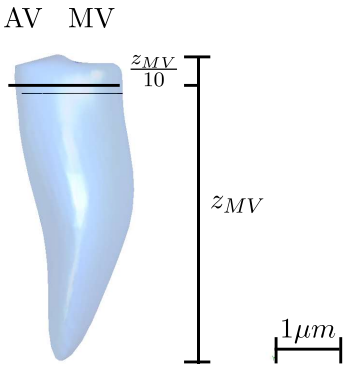
(a) Mean area-average ventricle wall shear stress $\bar{\tau}$ for all refinement levels.



(b) Mean volume-average kinetic energy $\bar{E}_{kin}$ for all refinement levels.



(c) Axial velocity along a horizontal plane below the MV $|\bar{u_z}|$ fore each refinement level.



(d) Horizontal plane used to compute $|\bar{u_z}|$ placed at 90 % the height of the MV z_{MV}.

Fig. S18: Results of the mesh-independence study for each refinement level.

Cycle independence

Each simulation was performed for three complete cardiac cycles, starting from the end-systolic state, to minimize initialization errors and obtain cycle-independent results. Three variables were analyzed for each cycle to determine the most suitable choice. These variables included the absolute average axial velocity $|u_z|$ along a plane positioned just below the basal region as shown in figure S18d, the area-averaged wall shear stress τ on the ventricle wall, and the volume-averaged kinetic energy E_{kin}. For these variables, both the temporal maximum and mean values were calculated. The mean value was considered a better estimate as it is less affected by outliers, whereas the maximum value was used to show outliers. The values were then compared across the three cycles, as presented in table S3. The cycle with the smallest deviations was selected as the cycle of interest for further analysis. For P1, the third cycle was chosen for all approaches since most values for the third cycle lie between the corresponding values of the first and second cycles. P2 exhibited relatively consistent values across all cycles, but cycle three was selected due to its smaller maximum values for τ and E_{kin}. In the case of P3, there was an outlier at the beginning of the simulation, but the values stabilized during the second and third cycles. While either cycle two or three could have been chosen, cycle three was selected for the sake of consistency with the other patients.

Table S3: Results of the cycle independence for each patient.

Variable	Cycle	P1-A1	P1-A2	P1-A3	P1-A4	P2-A2	P3-A2
$\lvert\bar{u_z}\rvert$ $(\frac{m}{s})$	1	0.0737	0.1074	0.0562	0.1072	0.1577	0.0891
	2	0.0747	0.1086	0.0557	0.1083	0.1619	0.0567
	3	0.0719	0.1053	0.0562	0.1067	0.1585	0.0568
$\lvert u_z\rvert_{max}$ $(\frac{m}{s})$	1	0.1536	0.1983	0.2097	0.6795	0.5254	44.28
	2	0.1761	0.2142	0.2027	1.1466	0.5104	0.1794
	3	0.1704	0.2117	0.2212	0.9244	0.5176	0.1794
$\bar{\tau}$ (Pa)	1	2.6527	2.3346	2.4272	5.1529	26.64	12.9172
	2	2.6605	2.3651	2.4409	5.067	26.911	1.8475
	3	2.6531	2.3474	2.4762	5.1629	28.4089	1.8537
τ_{max} (Pa)	1	5.9848	4.7116	5.8335	47.3088	151.63	8971.5
	2	5.9646	4.7362	5.8128	39.2	168.79	6.103
	3	5.9478	4.6742	5.9492	43.5688	150.2	6.0767
$\bar{E}_{kin}$ (mJ)	1	0.6025	0.6606	0.7216	1.2	0.99	0.33
	2	0.6118	0.66395	0.7329	1.1	0.99	0.15
	3	0.6097	0.6601	0.7208	1.1	1.01	0.15
E_{kinmax} (mJ)	1	1.8	1.9	2.6	13.3	8.95	3872.3
	2	1.8	1.9	2.7	11	9.18	0.86
	3	1.8	1.9	2.7	12.6	8.64	0.86
Chosen cycle		3	3	3	3	3	3

References

[1] L. Bennati, C. Vergara, V. Giambruno, I. Fumagalli, A. F. Corno, A. M. Quarteroni, G. Puppini, and G. B. Luciani. An image-based computational fluid dynamics study of mitral regurgitation in presence of prolapse. *Cardiovascular Engineering and Technology*, 14:457 – 475, 2023. doi:10.1007/s13239-023-00665-3.

[2] Celik I. Procedure for estimation and reporting of uncertainty due to discretization in cfd applications. *Journal of Fluids Engineering*, 130(7), 2008. doi:10.1115/1.2960953.

[3] C. Chnafa, C. Chnafa, S. Mendez, and F. Nicoud. Image-based simulations show important flow fluctuations in a normal left ventricle: What could be the implications? *Annals of Biomedical Engineering*, 44:3346–3358, 2016. doi:10.1007/s10439-016-1614-6.

[4] B. O. Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. URL: http://www.blender.org.

[5] A. Daub, J. Kriegseis, and B. Frohnapfel. Replication of left ventricular haemodynamics with a simple planar mitral valve model. *Biomedical Engineering / Biomedizinische Technik*, 65(5):595–603, 2020. doi:10.1515/bmt-2019-0175.

[6] M. S. M. Elbaz, R. J. van der Geest, E. E. Calkoen, A. de Roos, B. P. F. Lelieveldt, A. A. W. Roest, and J. J. M. Westenberg. Assessment of viscous energy loss and the association with three-dimensional vortex ring formation in left ventricular inflow: In vivo evaluation using four-dimensional flow mri. *Magnetic resonance in medicine*, 77(2):794–805, 2017. doi:10.1002/mrm.26129.

[7] J. Eriksson, J. Zajac, U. Alehagen, A. F. Bolger, T. Ebbers, and C. J. Carlhäll. Left ventricular hemodynamic forces as a marker of mechanical dyssynchrony in heart failure patients with left bundle branch block. *Scientific Reports*, 7, 2017. doi:10.1038/s41598-017-03089-x.

[8] L. Goubergrits, K. Vellguth, L. Obermeier, A. Schlief, L. Tautz, J. Bruening, H. Lamecker, A. Szengel, O. Nemchyna, C. Knosalla, T. Kuehne, and N. Solowjowa. Ct-based analysis of left ventricular hemodynamics using statistical shape modeling and computational fluid dynamics. *Frontiers in Cardiovascular Medicine*, 9, 2022. doi:10.3389/fcvm.2022.901902.

[9] A. Grünwald, J. Korte, N. Wilmanns, C. Winkler, K. Linden, U. Herberg, S. Groß-Hardt, U. Steinseifer, and M. Neidlin. Intraventricular flow simulations in singular right ventricles reveal deteriorated washout and low vortex formation. *Cardiovascular engineering and technology*, 13(3):495–503, 2022. doi:10.1007/s13239-021-00598-9.

[10] R. G. Leyh, C. Schmidtke, H. H. Sievers, and M. H. Yacoub. Opening and closing characteristics of the aortic valve after different types of valve-preserving surgery. *Circulation*, 100(21):2153–2160, 1999. doi:10.1161/01.cir.100.21.2153.

[11] S. Liao, B. Simpson, M. Neidlin, T. A. S. Kaufmann, Z. Li, M. A. Woodruff, and S. D. Gregory. Numerical prediction of thrombus risk in an anatomically dilated left ventricle: the effect of inflow cannula designs. *BioMedical Engineering OnLine*, 15, 2016. doi:10.1186/s12938-016-0262-2.

[12] K. Linden, C. Winkler, J. Breuer, and U. Herberg. Assessment of pressure-volume relations in univentricular hearts: Comparison of obtainment by real-time 3d echocardiography and mini pressure-wire with conductance technology. *PloS one*, 16(2), 2021. doi:10.1371/journal.pone.0246031.

[13] Y.-H. Loke, F. Capuano, E. Balaras, and L. J. Olivieri. Computational modeling of right ventricular motion and intracardiac flow in repaired tetralogy of fallot. *Cardiovascular Engineering and Technology*, 13:41 – 54, 2021. doi:10.1007/s13239-021-00558-3.

[14] J. O. Mangual, F. Domenichini, and G. Pedrizzetti. Three dimensional numerical assessment of the right ventricular flow using 4d echocardiography boundary data. *European Journal of Mechanics B-fluids*, 35:25–30, 2012. doi:10.1016/j.euromechflu.2012.01.022.

[15] W. Mao, A. Caballero, R. McKay, C. Primiano, and W. Sun. Fully-coupled fluid-structure interaction simulation of the aortic and mitral valves in a realistic 3d left ventricle model. *PloS one*, 12(9), 2017. doi:10.1371/journal.pone.0184729.

[16] D. P. Mukherjee. *Fundamentals of computer graphics and multimedia*. 2006.

[17] L. Obermeier, K. Vellguth, A. Schlief, L. Tautz, J. Bruening, C. Knosalla, T. Kuehne, N. Solowjowa, and L. Goubergrits. Ct-based simulation of left ventricular hemodynamics: A pilot study in mitral regurgitation and left ventricle aneurysm patients. *Frontiers in Cardiovascular Medicine*, 9:828556, 2022. doi:10.3389/fcvm.2022.828556.

[18] G. Pedrizzetti, G. L. Canna, O. R. Alfieri, and G. Tonti. The vortex—an early predictor of cardiovascular outcome? *Nature Reviews Cardiology*, 11:545–553, 2014. doi:10.1038/nrcardio.2014.75.

[19] G. Pedrizzetti, A. R. Martiniello, V. Bianchi, A. D'Onofrio, P. Caso, and G. Tonti. Changes in electrical activation modify the orientation of left ventricular flow momentum: novel observations using echocardiographic particle image velocimetry. *European heart journal cardiovascular Imaging*, 17 2:203–9, 2016. doi:10.1093/ehjci/jev137.

[20] K. K. Poh, L. C. Lee, L. Shen, E. Chong, Y. L. Tan, P. Chai, T. C. Yeo, and M. J. Wood. Left ventricular fluid dynamics in heart failure: echocardiographic measurement and utilities of vortex formation time. *European heart journal cardiovascular Imaging*, 13 5:385–93, 2012. doi:10.1093/ejechocard/jer288.

[21] T. Schenkel, M. Malvè, M. Reik, M. Markl, B. Jung, and H. Oertel. Mri-based cfd analysis of flow in a human left ventricle: Methodology and application to a healthy heart. *Annals of Biomedical Engineering*, 37:503–515, 2009. doi:10.1007/s10439-008-9627-4.

[22] J. H. Seo and R. Mittal. Effect of diastolic flow patterns on the function of the left ventricle. *Physics of Fluids*, 25:110801, 2013. doi:10.1063/1.4819067.

[23] J.-N. Thiel, U. Steinseifer, and M. Neidlin. Generic framework for quantifying the influence of the mitral valve on ventricular blood flow. *International journal for numerical methods in biomedical engineering*, 2023. doi:10.1002/cnm.3684.

[24] H. S. Wong, B. Li, A. Tulzer, G. Tulzer, and C. H. Yap. Fluid mechanical effects of fetal aortic valvuloplasty for cases of critical aortic stenosis with evolving hypoplastic left heart syndrome. *Annals of Biomedical Engineering*, 51:1485 – 1498, 2023. doi:10.1007/s10439-023-03152-x.

[25] J. Wong, R. Chabiniok, S. M. Tibby, K. Pushparajah, E. Sammut, D. Celermajer, D. Giese, T. Hussain, G. F. Greil, T. Schaeffter, and R. Razavi. Exploring kinetic energy as a new marker of cardiac function in the single ventricle circulation. *Journal of applied physiology*, 125 3:889–900, 2018. doi:10.1152/japplphysiol.00580.2017.