Computer Methods in Biomechanics and Biomedical Engineering

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/gcmb20

A semi-automated method for patient-specific computational flow modelling of left ventricles

Vinh-Tan Nguyen\textsuperscript{a}, Chong Jia Loon\textsuperscript{b}, Hoang Huy Nguyen\textsuperscript{a}, Zhong Liang\textsuperscript{cd} & Hwa Liang Leo\textsuperscript{b}

\textsuperscript{a} Institute of High Performance Computing, 1 Fusionopolis Way, Connexis Tower, Singapore 138632, Singapore
\textsuperscript{b} Division of Bioengineering, National University of Singapore, Singapore
\textsuperscript{c} Department of Cardiology Research, National Heart Centre of Singapore, Singapore
\textsuperscript{d} Duke-NUS Graduate Medical School Singapore, Singapore

Published online: 16 Aug 2013.

To cite this article: Vinh-Tan Nguyen, Chong Jia Loon, Hoang Huy Nguyen, Zhong Liang & Hwa Liang Leo (2015) A semi-automated method for patient-specific computational flow modelling of left ventricles, Computer Methods in Biomechanics and Biomedical Engineering, 18:4, 401-413, DOI: 10.1080/10255842.2013.803534

To link to this article: http://dx.doi.org/10.1080/10255842.2013.803534

PLEASE SCROLL DOWN FOR ARTICLE

Taylor & Francis makes every effort to ensure the accuracy of all the information (the "Content") contained in the publications on our platform. However, Taylor & Francis, our agents, and our licensors make no representations or warranties whatsoever as to the accuracy, completeness, or suitability for any purpose of the Content. Any opinions and views expressed in this publication are the opinions and views of the authors, and are not the views of or endorsed by Taylor & Francis. The accuracy of the Content should not be relied upon and should be independently verified with primary sources of information. Taylor and Francis shall not be liable for any losses, actions, claims, proceedings, demands, costs, expenses, damages, and other liabilities whatsoever or howsoever caused arising directly or indirectly in connection with, in relation to or arising out of the use of the Content.

This article may be used for research, teaching, and private study purposes. Any substantial or systematic reproduction, redistribution, reselling, loan, sub-licensing, systematic supply, or distribution in any form to anyone is expressly forbidden. Terms & Conditions of access and use can be found at http://www.tandfonline.com/page/terms-and-conditions
A semi-automated method for patient-specific computational flow modelling of left ventricles

Vinh-Tan Nguyen*, Chong Jia Looⁿ, Hoang Huy Nguyen*, Zhong Liangc,d and Hwa Liang Leo

*Institute of High Performance Computing, 1 Fusionopolis Way, Connexis Tower, Singapore 138632, Singapore; bDivision of Bioengineering, National University of Singapore, Singapore; cDepartment of Cardiology Research, National Heart Centre of Singapore, Singapore; dDuke-NUS Graduate Medicate School Singapore, Singapore

(Received 23 May 2012; final version received 6 May 2013)

Patient-specific computational fluid dynamics (CFD) modelling of the left ventricle (LV) is a promising technique for the visualisation of ventricular flow patterns throughout a cardiac cycle. While significant progress has been made in improving the physiological quality of such simulations, the methodologies involved for several key steps remain significantly operator-dependent to this day. This dependency limits both the efficiency of the process as well as the consistency of CFD results due to the labour-intensive nature of current methods as well as operator introduced uncertainties in the modelling process. In order to mitigate this dependency, we propose a semi-automated method for patient-specific computational flow modelling of the LV. Using magnetic resonance imaging derived coarse geometry data of a patient’s LV endocardium shape throughout a cardiac cycle, we then proceed to refine the geometry to eliminate rough edges before reconstructing meshes for all time frames and finally numerically solving for the intra-ventricular flow. Using a sample of patient-specific volunteer data, we demonstrate that our semi-automated, minimal operator involvement approach is capable of yielding CFD results of the LV that are comparable to other clinically validated LV flow models in the literature.

Keywords: computational fluid dynamics; left ventricles; patient-specific; cardiac cycle; geometry reconstruction

1. Introduction

Cardiac or heart failure is a significant problem that affects people from all over the world. Broadly speaking, heart failure can be understood as a condition where the heart can no longer pump enough blood to the rest of the body. In the USA, 550,000 new cases of heart failure are diagnosed yearly and currently more than five million Americans are suffering from it. It is also a leading cause of hospitalisation among Medicare beneficiaries, and in 2007, 33 billion US Dollars were spent in the fight against this disease (Schucken et al. 2008).

The left ventricle (LV) functions as the terminal chamber of the heart for blood before it is pumped to the entire body with the exception of the lungs. LV diastolic dysfunction, which commonly occurs in patients with hypertension and diabetes mellitus, and/or the elderly, carries a substantial risk for the subsequent development of heart failure. One way of improving diastolic heart failure patient survival rates would be through better early diagnosis methods (Gaasch and Little 2007). This is especially so when we consider that some studies have found that diastolic heart failure is often misdiagnosed or under-diagnosed during primary care (Davies et al. 2001). Hence, a better means for ventricular function assessment is necessary in order to improve diastolic heart failure patient outcomes.

Computational fluid dynamics (CFD) is a branch of fluid mechanics which utilises numerical methods and algorithms to solve and analyse fluid flow problems. The calculations required for simulating the interactions between fluid flow and surfaces defined by certain boundary conditions are performed by a computer. Due to increasing recognition of correlations between haemodynamic parameters such as flow patterns and the state of health of the human circulatory system (Barber 2007), manifold numerical models have been created that are able to generate detailed flow fields, given physiological geometries and boundary conditions (Long et al. 2008).

The numerical models of the heart can roughly be divided into two types: first, the models which take into account the interaction between the fluid flow and the motion driving it: the fluid–structure interaction (FSI) models, including fictitious domain models and coupled FSI models and second, those that take the movement of the inner ventricular wall as a boundary condition: the prescribed geometry models. Both models cannot be considered as alternatives to each other, but as approaches in their own right (Schenkel et al. 2009).

While a coupled FSI approach is promising for future work, the numerical schemes necessary to simulate the coupled systems are not yet fully developed and tested. Furthermore, in order to be truly realistic, total heart function modelling needs to integrate not only FSI but also cardiac anatomy, electrical activation and metabolism as well, which is to date a highly complex task (Khalafvand et al. 2011). In light of this, for a clinical status quo analysis...
and evaluation of intraventricular flow, a prescribed geometry model is better suited especially if we assume that the effect of forces exerted by the heart muscle onto the fluid is much larger than the effect of wall forces exerted by the fluid back onto the muscle tissue. Based on the results of several studies (Saber et al. 2003; Schenkel et al. 2009), there is no evidence that this assumption is not true for the systole, and comparison with FSI simulations (Krittian et al. 2010) show similar flow patterns also for diastole.

Existing prescribed-geometry patient-specific LV CFD simulation research has focused primarily on qualitative improvements of said CFD models in order to yield results that approximate better to physiological data. To illustrate, past works have established the sensitivity of LV flow patterns to parameters such as atria geometry (Schenkel et al. 2009), LV shape (Iwase et al. 2003; Ge and Ratcliffe 2009), boundary condition type (Long et al. 2003; Merrifield et al. 2004), as well as the opening mode of the mitral valve orifice (Nakamura et al. 2006). While this represents steady progress made towards making patient-specific LV CFD simulations clinically relevant, a significant hurdle that remains to be overcome is the often labour-intensive nature of conducting CFD simulations which makes it difficult for conducting large-scale patient-specific studies. For example, Doenst et al. (2009) acknowledged that while it is conceivable for the future to generate patient-specific CFD models for ventricular reconstruction patients in order for surgeons to know what shape would be optimal for the patient before surgery is performed, the cumbersome and time-consuming nature of current fluid model generation processes remain a significant hurdle to be overcome.

In addition, the process from image acquisition to simulation result is often not integrated and relies heavily on several operator-dependent steps such as image-segmentation, mesh reconstruction and CFD simulation. Besides being cumbersome and non-trivial, operator-dependent tasks also constitute potential sources of error of which influence is difficult to mitigate and quantify (Antiga et al. 2008). For example, Long et al. (2008) noted that uncertainties caused by surface smoothing and interpolation (3D time) are among the key issues for error control in their LV geometry reconstruction methodology. Unless such errors can be effectively managed, the ability of patient-specific LV CFD methodologies to generate consistent results especially in large scale studies may potentially be significantly compromised. Without efficacious large-scale patient-specific studies, the usefulness of CFD simulations will then remain inconclusive while its results will continue to have difficulties fulfilling the criteria of evidence-based medicine.

To date, several other CFD simulation models have reported conducting the mesh smoothing process either manually (Saber et al. 2003; Doenst et al. 2009) or with the help of interpolation schemes such as polar Catmull–Rohm splines (Long et al. 2008). A potential issue associated with this would be the inconsistent quality of the final geometry due to the different and sometimes operator-dependent methods involved in the smoothing process. Meanwhile, although semi-automatic grid generation techniques have also been reported in those earlier works, the need for grid and time-step independency tests which even involves grid interpolation in order to generate sufficient time-frame geometries within a cardiac cycle. Sufficient time-frame geometries are needed so that the simulation time step complies with the Courant criterion necessary for time-step independency (Saber et al. 2003).

In this context, we would hence like to propose a semi-automated and integrated method for conducting geometry reconstruction, LV mesh smoothing and CFD simulation. Although it is important to reconstruct geometries from medical images, the segmentation process is not in the scope of this work. Assuming that a reasonably accurate extraction of geometries from medical images is possible, in this proposed approach, the patient-specific LV geometries obtained from segmentation of medical images are reconstructed using subdivision surfaces. The reconstructed triangular surfaces then serve as input fronts to the Delaunay triangulation algorithm to generate unstructured tetrahedral meshes for CFD simulations. The motion of LV walls captured by magnetic resonance imaging (MRI) scans are translated into dynamic motions of mesh boundaries using fast radial basis function (RBF) interpolation, thus enabling unsteady simulations of blood flows over a cardiac cycle taking into account the pumping motion of LV walls. In this work, the blood is modelled as Newtonian incompressible flows and a second order finite volume method on unstructured grids is employed to solve for incompressible Navier–Stokes (NS) flows. It is worth mentioning that our proposed approach is capable of streamlining the execution of the aforementioned steps through automated smoothing and grid generation codes before conducting the CFD simulation automatically. In addition, our automated smoothing method using subdivision surfaces enables us to accept low-resolution surface mesh data and then proceed to convert it into a smooth and high-resolution geometry. Using a sample patient-specific data of a healthy volunteer, we demonstrate how the proposed automated approach reconstructs the geometry throughout a cardiac cycle before numerically modelling for blood flows over the prescribed motion of LV walls. Comparisons were then made between the present CFD intra-ventricular flow pattern results and the literature data.

2. Problem statement

In this study, we are concerned with simulations of blood flows in the left ventricle of patients whose geometrical
information of their specific LV is retrieved from medical imaging process. The blood flows are modelled as incompressible viscous fluids and governed by incompressible NS equations.

Considering incompressible viscous flows over a domain $\Omega$, the governing incompressible NS equations are written in conservative, non-dimensional form as follows:

$$\frac{d}{dt} \int_{\Omega} \mathbf{U} \, d\Omega + \int_{\partial\Omega_t} (\mathbf{F}_j - \hat{\mathbf{F}}_j) n_j \, dS - \int_{\partial\Omega} \mathbf{G}_j n_j \, dS = 0,$$

where $n_j$ is the component, in direction $x_j$, of the outward unit normal vector to $\partial\Omega_t$, $\mathbf{U} = (p, u_1, u_2, u_3)^T$ is the non-dimensional unknown variable, $I = \text{diag}(0, 1, 1, 1)$ is the modified unit matrix. In the above description, the inviscid fluxes $\mathbf{F}_j$, the arbitrary Lagrangian–Eulerian (ALE) fluxes $\hat{\mathbf{F}}_j$ and the viscous fluxes $\mathbf{G}_j$ are defined by

$$\mathbf{F}_j = \begin{bmatrix} u_j \\ u_1 u_j + p \delta_{1j} \\ u_2 u_j + p \delta_{2j} \\ u_3 u_j + p \delta_{3j} \end{bmatrix}, \quad \hat{\mathbf{F}}_j = \begin{bmatrix} 0 \\ u_1 \tilde{u}_j \\ u_2 \tilde{u}_j \\ u_3 \tilde{u}_j \end{bmatrix}, \quad \mathbf{G}_j = \begin{bmatrix} 0 \\ \tau_{1j} \\ \tau_{2j} \\ \tau_{3j} \end{bmatrix},$$

respectively, where $u_j$ is the velocity components of the fluid, $\delta_{ij}$ is the Kronecker delta and

$$\tau_{ij} = \frac{1}{Re} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right).$$

3. Geometry reconstructions

MRI scanning was performed in a healthy volunteer (1.5T Siemens scanner, Avanto, Siemens Medical Solutions, Erlangen, Germany). Ventricular 2-chamber, 4-chamber and short-axis planes with 12–14 equidistant slices covering both ventricle, atrium and aorta were acquired. The field of view was typically 320 mm with in-plane spatial resolution of $\leq 1.5$ mm. Images were acquired in a single breath hold, with 25 temporal phases per heart cycle. The short- and long-axis views from the MRI were used to reconstruct 3D LV geometry in conjunction with dynamics LA and ascending aorta using CMRtools suits (CVIS, Imperial College, UK) as well as customised algorithm and used in other studies (Long et al. 2008). This method allows the representation of the dynamic morphology of anatomical features in the LV, LA and aorta. The smooth surfaces were generated from a set of arbitrarily control points with an interpolation scheme by using polar Catmull–Rohm spines. Temporal spline interpolation was used to generate a complete time series of models. The processing methods of MRI data for LV were developed and partly used in our previous studies (Zhong et al. 2009, 2011).

As a product from patient-specific MRI scans and segmentation, a discrete surface of the LV geometry is constructed at different time during one cardiac cycle. In the current work, raw data of LV discrete surfaces were obtained from a normal object during a typical cardiac cycle of $T = 847$ ms. The data comprise a series of 25 frames recorded during the cardiac cycle and each frame consists of a relatively coarse resolution of 3328 vertices representing the LV. Figure 1 shows snapshots of LV discrete surfaces at different instances. In order to perform CFD simulations of intra-ventricular flows, this raw data of LV discrete surfaces need to be smoothed and computational grids are generated to conform with the motion of LV’s wall. We proposed a process of smoothing surfaces using subdivision surface together with an unstructured grid generation algorithm for producing computational grids and an effective mesh moving scheme to conform the computational grids to motions of LV wall. The process of smoothing surfaces and volume mesh generation is only applied to an arbitrarily chosen frame, considered as a seed frame. The information on LV wall motion will then be interpolated and translated into other frames subsequently. In the subsequent subsections, these processes will be elaborated further from...
smoothing surfaces using subdivision to volume mesh generation, and finally interpolation of MRI captured surface motions into computational mesh movement.

3.1 Subdivision surface

Given the discrete surfaces obtained from MRI scans and segmentation, the surface mesh resolution is not sufficient to carry out CFD simulations of intra-ventricular flows. Particularly, viscous flows require much higher mesh resolution near the wall areas for capturing boundary layers. It is thus desirable to refine the raw surfaces to a required resolution for CFD calculations. Convergence and quality of viscous computational also require the reconstructed surfaces to be $C^1$ continuous, or differentiate functions whose first derivatives are continuous. For discrete surfaces of unstructured grids, it is difficult to ensure $C^1$ continuity in the conventional sense of strict slope continuity across elements due to lack of geometrical information. There are a number of alternatives being developed in the literatures for addressing the difficulties in the $C^1$ continuity requirement. In particular, subdivision surfaces (Peters and Reif 2008) have been widely employed for generating smooth surfaces from an arbitrary collection of nodal set points. In this work, as the initial raw surfaces are triangular meshes, the Loop scheme (Loop 1987) is employed to construct the smooth surfaces (Figure 2).

In a nutshell, subdivision schemes construct smooth surfaces through a recursive limiting procedure of refinement from the initial mesh, also known as control mesh of the surface. Typically, the subdivision process consists of a refinement step and an updating step. In the refinement step, triangular elements are refined by quadri-section. The nodal positions of the refined mesh are then computed as weighted averages of the nodal positions of the previous mesh.

For nodes on the boundary of the mesh, one needs to pay extra care to update the new nodal positions. It can be considered as a 1D curve and subdivision rules in 1D can be applied for vertices on the boundary. Subdivision surfaces obtained by the above Loop scheme are proved to be $C^2$ globally, except at a number of isolated points where only $C^1$ continuity is guaranteed.

3.2 Unstructured grid generation

Once the surface mesh of LV is sufficiently refined to a desirable resolution, the Delaunay triangulation is employed to generate the volume unstructured grid. A Delaunay mesh construction approach is a concept which consists of two tasks. One addresses the mesh topography which is defined through the placement of mesh points by a variety of techniques. The Delaunay triangulation is performed afterwards on the complete set of points to create the 3D mesh topology. There are several reported algorithms for constructing the Delaunay triangulation. Given that it is required to build unstructured grids of millions of points from an arbitrary discrete surface, we adopted the Delaunay triangulation approach by Weatherill and Hassan (1994). The adopted fast and efficient Delaunay triangulation is based on the Bowyer algorithm by sequentially adding points into an existing Delaunay-satisfied structure until filling up the whole domain. Nodes provided for Delaunay triangulation are created using automatic point creation scheme driven by the boundary.

Figure 1. Snapshots of LV discrete surfaces obtained from MRI scan and segmentation at different time in one cardiac cycle.

Figure 2. From (a) the coarse surface mesh (pre-processed raw data) which is on the left, the refined surface mesh (b) is generated using subdivision surface.
frames, the nodal displacement is calculated as

\[ \mathbf{d}_n^k = \mathbf{x}_n^k - \mathbf{x}_n^{k-1}, \]

where the superscript \( n \) denotes the time frame number and \( \mathbf{x}_n \) is the coordinate of raw surface nodes. Radial basis function interpolation is employed to spread the nodal displacements from raw surfaces to smooth surfaces. The interpolation function, \( \sigma \), describing the displacement of nodes on refined meshes can be approximated by a sum of basis functions

\[ \mathbf{s}(\mathbf{x}) = \sum_{j=1}^{N_s} \lambda_j \phi(\|\mathbf{x} - \mathbf{x}_{ij}\|) + \mathbf{p}(\mathbf{x}), \]  

where \( \mathbf{x}_{ij} = [x_{ij}, y_{ij}, z_{ij}] \) are the nodal positions on raw surfaces, \( N_s \) is number of nodes on the raw surfaces, \( p \) is the polynomial of total degree at most \( k \) and \( \phi \) is a given basis function. The interpolating coefficients \( \lambda_j \) and polynomial coefficients \( c_i \) are determined by the condition that interpolation function must return the exact displacement at the nodes on the raw surface, \( \mathbf{d}_{ij} \).

\[ \mathbf{s}(\mathbf{x}_{ij}) = \mathbf{d}_{ij}. \]

Further conditions are imposed on coefficient vector \( \mathbf{\lambda} \) that

\[ \sum_{j=1}^{N_s} \lambda_j q(\mathbf{x}_{ij}) = 0 \]

for all polynomials \( q \) of degree less than or equal to that of polynomial \( p \). The minimal degree of polynomial \( p \) depends on the choice of the basis function \( \phi \). In this work, a linear polynomial is used \( p(\mathbf{x}) = c_0 + c_1 x_j + c_2 y_j + c_3 z_j \) for the conditionally positive definite basis functions of order \( m = 2 \) (de Boer et al. 2007). The interpolation coefficients are calculated from solving the corresponding system of equations

\[ \begin{pmatrix} \mathbf{A} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{O} \end{pmatrix} \begin{pmatrix} \mathbf{\lambda} \\ \mathbf{c} \end{pmatrix} = \begin{pmatrix} \mathbf{d}_i \\ 0 \end{pmatrix}, \]

in which \( \mathbf{A} \) is an \( N_s \times N_s \) matrix containing the evaluation of the basis functions at data nodes \( A_{ij} = \phi(\mathbf{x}_{ij}, \mathbf{x}_{ij}) \) and \( \mathbf{P} \) is an \( N_s \times 4 \) matrix with rows of polynomial evaluated at source nodes \( P_{ij} = p_j(\mathbf{x}_{ij}) \). In this work, various iterative techniques including least square, conjugate gradient and the generalised minimal residual (GMRES) scheme had been tested and GMRES is the most effective and robust for solving the system of equations (8). The displacement of nodes on the smooth surfaces (Figure 4) can be derived from evaluating the interpolation function (5) at these nodes as

\[ \mathbf{d}_{ij} = \mathbf{d}(\mathbf{x}_{ij}) = \mathbf{s}(\mathbf{x}_{ij}). \]

3.3.1 Boundary interpolation

Given the nodal positions of raw surfaces mesh at different frames, the nodal displacement is calculated as

\[ \mathbf{d}_n^k(\mathbf{x}_n^k) = \mathbf{x}_n^k - \mathbf{x}_n^{k-1}, \]

3.3.2 Internal mesh movement

As the boundary surfaces motion is computed from the above interpolation procedure, it is required to update...
internal mesh nodal positions to maintain mesh quality. One way of updating the mesh is to remesh the entire computational domain every time the geometry moves. However, regeneration of the grids usually requires more computational resources and thus should be avoided if possible. Moreover, certain desirable features of computations such as geometric conservation are difficult to ensure if the mesh structure changes between time steps. It is, therefore, a good practice to instead strive to retain the connectivity of the previous mesh. In this work, a grid smoothing scheme is employed to update the grid and minimise the distortion. In principle, the grid movement is constructed for the whole domain such that the grid quality is assured and the displacement at the boundaries matches the given boundary displacement. Several mesh movement approaches have been investigated by other researchers. One approach uses a stress/strain analogy where the computational domain is thought of as an elastic material subjected to a deformation. This approach involves solving a Laplace-type system (see Fahrat et al. 1998; Nguyen et al. 2010). Alternatively, a similar approach is adopted in this work where it is assuming that the edges of the mesh are behaving like linear springs connecting the nodes together (Nguyen et al. 2010). The force acting on an edge is proportional to the difference between the current length and the new length due to motions of the boundary nodes. Once the motion of the nodes on the boundary has been determined, the motion of the remaining inner nodes is calculated by solving a force equilibrium at the nodes as

Figure 4. Node displacement on the smooth surfaces interpolated from raw data using RBF approach.
The force at each node \( I \) at time step \( n \), denoted as \( \mathbf{F}_I(x_I) \), is the summation of the force along each edge connected to the node \( I \). The force acting on an edge is proportional to the difference between the current length and the new length due to motions of the boundary nodes.

\[
\mathbf{F}_I(x_I) = \sum_{J \in \Lambda_I} k_{IJ} (l_{IJ}^n - l_{IJ}^{prev}) = 0, \quad \forall I \in \mathcal{T}_h, 
\]

where \( l_{IJ}^{prev} \) and \( l_{IJ} \) are the lengths of the edge between nodes \( I \) and \( J \) before and after the movement, respectively, and \( k_{IJ} \) is the spring coefficient, which is usually taken as the inverse length of the edge. Therefore, the node displacement can be found by casting the force Equation (11) as a pseudo-time equation system. The coefficient \( b \) is the spring coefficient, which is usually taken as the maximum of 50–100 iterations are proved to be sufficient for smoothing the displacement. As shown in Nguyen et al. (2010), this approach is robust, fast and typically a maximum of 50–100 iterations are used to update mesh motions; hence, a fix number of interactions are used to update mesh motions.

\[
\Delta \mathbf{x}_I : \mathbf{F}_I(x_I) = \sum_{J \in \Lambda_I} k_{IJ} (\Delta \mathbf{x}_I - \Delta \mathbf{x}_J) = 0, \quad \forall I \in \mathcal{T}_h, \quad \text{s.t.} \quad \Delta \mathbf{x}_J = \mathbf{d}_{b_{ij}}, \quad \forall J \in \Gamma. 
\]

In the above expression, \( \Delta \mathbf{x}_I = x_I - x_I^{prev} \) is the displacement of node \( I \) in the discretised domain \( \mathcal{T}_h \) and \( \mathbf{d}_{b_{ij}} \) is the displacement of nodes on the boundary \( \Gamma \) obtained from the previous interpolation step. In practical implementation, the node displacement can be iteratively updated as

\[
\Delta \mathbf{x}_I^{(m+1)} = \frac{\sum_{J \in \Lambda_I} k_{IJ} \Delta \mathbf{x}_J}{\sum_{J \in \Lambda_I} k_{IJ}^2}, \quad \forall I, \quad m = 1, \quad \mathcal{M}_s, \quad (12)
\]

where \( \mathcal{M}_s \) is the number of smoothing iterations to update the internal node displacement. In the current implementation, a fix number of interactions are used to update mesh motions; typically a maximum of 50–100 iterations are proved to be sufficient for smoothing the displacement. As shown in Nguyen et al. (2010), this approach is robust, fast and parallelisable, usually requiring minimal amount of time comparing to the total CPU-time required to solve the system time accurately. The mesh motion solver is integrated in the solver to make the computation seamless and automatic. Figure 5 shows volume meshes at different times during cardiac cycle obtained from using the proposed moving mesh scheme.

4. Numerical discretisation of governing equations

In the artificial compressibility approach, the instantaneous unsteady solution is achieved by integrating the artificial compressibility form of the governing equations to steady state in pseudo-time. In the pseudo-time form, Equation (1) is modified as

\[
\int_{\Omega} \mathbf{P}^{-1} \frac{dU}{d\tau} d\Omega + \int_{\Omega(t)} \mathbf{U} d\Omega + \int_{\Omega(t)} (\mathbf{F}_j - \mathbf{F}_j - \mathbf{G}_j) \eta_j dS = 0, \quad (13)
\]

where the diagonal matrix \( \mathbf{P} \), defined as \( \mathbf{P} = \text{diag}(\beta^2, 1, 1, 1) \), is employed as a pre-conditioner for the pseudo-time solution system. The coefficient \( \beta \) can be viewed as a relaxation parameter for the pseudo-time solution procedure, with its value selected so as to optimise the convergence of the solution procedure. Based upon numerical experiments, the value of \( \beta \) is defined as

\[
\beta^2 = \max (\beta^2_{\min}, C_\beta|\mathbf{u}|) \quad (14)
\]

with \( \beta^2_{\min} = 1 \) and \( C_\beta = 2.5 \).

4.1 Finite volume methods

The discretisation of Equation (13) is accomplished using a cell vertex finite volume procedure. The computational domain \( \Omega \) is subdivided into a set of non-overlapping tetrahedral elements using a Delaunay mesh generation process with automatic point creation. To enable the implementation of a cell vertex finite volume solution approach, a median dual mesh is constructed by connecting edge mid-points, element centroids and face centroids such that only one node is present in each control volume. The FV discretisation transforms surface and volume integrals into a sum of face and control volume integrals and approximates them to second order accuracy.

The contribution of the inviscid flux over the control volume surface for node \( I \) is then computed as

\[
\int_{\Gamma_I} \mathbf{F}_j \eta_j \, d\mathbf{x} = \sum_{J \in \Lambda_I} C^I_{\eta_j} \left( F^I_{\eta_j} + F^J_{\eta_j} \right) + \sum_{J \in \Lambda_I^b} D^I_{\eta_j} F^I_{\eta_j}, \quad (15)
\]

where \( \Lambda_I \) denotes the set of nodes connected to node \( I \) by an edge and \( \Lambda_I^b \) denotes the set of nodes connected to node \( I \) by an edge on the computational boundary. In the above formulation, \( C^I_{\eta_j} \) and \( D^I_{\eta_j} \) are the values of edge coefficients for internal and boundary edges, respectively. These coefficients are calculated for each edge using the dual mesh segment associated with the edge as follows:

\[
C^I_{\eta_j} = n^I_{\eta_j} = \sum_{K \in \Gamma^I_k} A_{\eta_j} n^I_{\eta_j} \quad (16)
\]

\[
D^I_{\eta_j} = \sum_{K \in \Gamma^I_k} A_{\eta_j} n^I_{\eta_j}. \quad (17)
\]

In Equation (16), \( A_{\eta_j} \) is the area of facet \( \Gamma^I_k \) and \( n^I_{\eta_j} \) is the outward unit normal vector of the facet from the viewpoint of node \( I \). \( \Gamma^I_{\eta_j} \) is the set of dual mesh facets on the computational boundary touching the edge between nodes \( j \) and \( J \).

Similarly, the viscous terms are approximated by

\[
\int_{\partial \Omega} \mathbf{G}_{ij} \eta_i \, d\mathbf{x} = \sum_{J \in \Lambda_i} C^I_{\eta_j} G^I_{ij} + \sum_{J \in \Lambda_i^b} D^I_{\eta_j} G^I_{ij}, \quad (18)
\]
where

$$G_{ij}^{1} = \begin{bmatrix} 0 \\ \tau_{ij}^{1} \\ \tau_{ij}^{2} \\ u_{ij}^{1} \tau_{ij}^{1} - q_{ij}^{1} \end{bmatrix}; \quad G_{ij}^{2} = \begin{bmatrix} 0 \\ \tau_{ij}^{2} \\ \tau_{ij}^{3} \\ u_{ij}^{2} \tau_{ij}^{2} - q_{ij}^{2} \end{bmatrix}. \quad (19)$$

Here, the discrete form of the deviatoric stress tensor at an edge midpoint is given by

$$\tau_{ij}^{1} = \frac{\mu_{ij} + \mu_{ji}}{2} \left[ -\frac{2}{3} \left( \tilde{\alpha}_{i}^{h} u_{i}^{h} \bigg|_{n}^{n} - \tilde{\alpha}_{j}^{h} u_{j}^{h} \bigg|_{n}^{n} \right) \delta_{ij} + \left( \tilde{\alpha}_{i}^{h} u_{i}^{h} \bigg|_{n}^{n} + \tilde{\alpha}_{j}^{h} u_{j}^{h} \bigg|_{n}^{n} \right) \right],$$

and the energy dissipation term is

$$u_{k}^{n} \tau_{ij}^{n} = \frac{u_{k}^{n} + u_{k}^{n}}{2} \tau_{ij}^{n}. \quad (20)$$

### 4.2 Time discretisation

At a general node $I$, the physical time derivative term in Equation (13) is approximated, using a three-level second order accurate finite difference scheme, as

$$\frac{d}{dt} \int_{\Omega_{I}} U d\Omega \bigg|_{t}^{n} \approx \frac{1}{\Delta t} \left( \frac{3}{2} V_{I}^{n} U^{n} - 2 V_{I}^{n} U^{n-1} + \frac{1}{2} V_{I}^{n-2} U^{n-2} \right). \quad (21)$$

Here, the superscript $n$ refers to an evaluation at time level $t = t_{n}$, the time levels are taken to be equally spaced with a
time step $\Delta t$ and $V_I$ is the volume of $\Omega_I$. To stabilise computations, biharmonic form of artificial viscosity term constructed as in the Jameson-Schmidt-Turkel fashion were added to the discretisation.

When the terms in Equation (13) are approximated in this fashion, the final form of the discrete equation at a node $I$ may then be written as

$$V_I \frac{dU^n_I}{dt} = R^n_I,$$  \hspace{1cm} (22)

where $R^n_I$ represents the discretisation at time $t = t_n$ of the time derivative, the inviscid fluxes, the ALE fluxes, the viscous fluxes and the biharmonic artificial dissipation.

At each physical time level $t = t_n$, the value of $U^n$ is obtained by integrating Equation (22) in pseudo-time to steady state. Using the superscript $(m)$ to denote the evaluation at pseudo-time level $\tau = \tau_m$, this integration is accomplished by using the explicit three-stage scheme

$$U_I^{t(0)} = U_I^{t(m)},$$

$$U_I^{t(k)} = U_I^{t(k-1)} + \alpha_k \left( \frac{\Delta \tau}{V_I} \right) R_I^{t(k-1)}, \hspace{1cm} k = 1, 2, 3, \hspace{1cm} (23)$$

$$U_I^{t(m+1)} = U_I^{t(3)}.$$  

The stability parameter, $\alpha$, is set equal to 1.0 and the coefficients $\alpha_1 = 0.6, \alpha_2 = 0.6, \alpha_3 = 1.0$ are employed; $\Delta$,is the local value for the pseudo-time step.

### 4.3 Boundary conditions

Among the most important considerations for generation of an efficacious CFD model of the LV would be the application of boundary conditions. At the LV wall, the wall boundary condition is applied while inflow and outflow conditions are applied at the mitral and aortic inlets.

#### 4.3.1 Wall boundary condition

For a viscous flow, the boundary conditions on the wall are the no-slip and no-penetration conditions, namely

$$((u - \omega) \cdot t)_{wall} = 0 \hspace{1cm} \text{(no-slip)}, \hspace{1cm} (24)$$

$$((u - \omega) \cdot n)_{wall} = 0 \hspace{1cm} \text{(no-penetration)}, \hspace{1cm} (25)$$

where $w$ is the velocity of the wall, $t$ and $n$ are the tangential and the normal unit vectors to the wall, respectively.

#### 4.3.2 Inflow outflow boundary condition

Both outflow and especially inflow into the LV have been found to have significant impact on intra-ventricular flow patterns (Davies et al. 2001). For example, vortices have been known to form at different locations in the ventricle when slightly different boundary conditions are applied. The challenge in applying this type of boundary condition has inspired many works in the literature, including the type of baffle boundary condition (Schenkel et al. 2009) and hybrid boundary condition (Long et al. 2008). Despite unique strengths of the aforementioned boundary conditions, it is to date unclear which boundary condition type is best suited for simulating patient-specific intraventricular flow. In light of this, we hence propose a relatively simplified boundary condition for our method namely an image-based definition of inflow and outflow conditions. This method is similar to the one used by Iwase et al. (2003) and essentially allows us to estimate the flow rate into and out of the ventricle using a formula which only requires ventricular volume information. A key advantage of this method is that it does not require patient-specific ventricular pressure data, of which acquisition is likely invasive and hence undesirable. Given the volume data obtained from MRI data, the volume flow rate can be derived as the change of volume in time. Assuming that the total mass is conserved during one cardiac cycle, with the information on the area of mitral and aortic inlets, the flow velocity can be calculated. Figure 6 shows the change of volume and the derived velocity profile from a healthy subject during one typical cardiac cycle.

For the inflow boundary condition, velocity is prescribed and pressure is computed from the solution. The inflow boundary condition can be written as

$$u = u_{in}; \hspace{1cm} (u \cdot n)_{inflow} < 0,$$  \hspace{1cm} (26)

where $u_{in}$ denotes the prescribed inflow velocity.

### 5. CFD simulations of blood flows in LVs

From the reconstructed meshes of the patient-specific LV and volume information extracted from scanned images, it is sufficient to carry out numerical simulations of flows in order for us to gain insight into the flow patterns present within the ventricle throughout a cardiac cycle. Using the proposed approach, simulations of flows over LV during a cardiac cycle can be realised to investigate flow characteristics. The flow was assumed to be incompressible Newtonian fluid with a kinematic viscosity of $\nu = 3.85 \times 10^{-6}$ m$^2$ s$^{-1}$. The simulation started from the first frame corresponding to the beginning of systole and was performed for a number of cardiac cycles in order to obtain a full development of the flow. In our experience, it is found that three to four cycles usually produce representative results. As we are interested to assess the ability of our solver to produce results of reasonable accuracy throughout a cardiac cycle, results of flow patterns were evaluated and compared against validated LV CFD flow patterns from the literature (Schenkel et al. 2009).
Figure 7 shows the results of velocity contour of flow during one cardiac cycle. It can be observed that flow structures vary according to the change in the velocity input and motion of the LV walls. In our current approach, the prescribed motion of the LV walls captured from MRI images was translated into the motion of the mesh boundary, thus providing momentum to the internal flows. Combination of wall motion and velocity profile provided at mitral and aortic inlets resulted in different vortex structures during the cardiac cycle.

A comparison analysis was conducted between the literature data (Schenkel et al. 2009) and our own generated LV CFD velocity data depicted in Figure 8. During early diastole, the literature data suggested that the resulting jet flow developed into a ring-like vortex. For our simulation shown in Figure 8(a), a similar vortex formation was observed near the mitral inlet orifice of our model. In the next comparison time frame (Figure 8(b)), the literature reported an initially symmetrical ring-like vortex that was observed to grow asymmetrically throughout diastole to fill the shape of the LV. A similar trend of asymmetrical vortex growth was observed in our simulation. Eventually in the next comparison time frame, the vortex in the literature was tilted to fill the elongated shape of the ventricle. A similar
space filling vortex was observed in our simulation here in Figure 8(c) as well. Finally, during systole the literature results suggested partial dissipation of vortical structures where most of the rotational energy seems to help flush the blood out of the LV. Here our simulation results (Figure 8(d)) once again showed good agreement with the literature results. Overall, our simulation flow pattern results agreed well with the comparison literature data simulation results.

Figure 9 shows pressure contours in the central cut-plane of the LV at various stages during cardiac cycle. It is observed that in the beginning of isovolumetric contraction (Figure 9(a)), the pressure in the lower half of the ventricle is higher than that in the upper half. During systole (Figure 9(b)), high pressure is observed throughout the ventricle except at the outflow orifice region of the ventricle. Figure 9(c) depicts pressure at mid-diastole.

Figure 8. Velocity vectors of our CFD simulation results. (a) Early diastole, (b) mid-diastole, (c) late diastole and (d) early systole. Numbering corresponds to frame number for our simulation result data.

Figure 9. Pressure magnitude distributions of our CFD simulation results. (a) Isovolumetric contraction, (b) systole and (c) mid-diastole. Numbering corresponds to frame number of our simulation result data.
stage where higher pressure distribution at the bottom half of the ventricle as well as low pressure pockets near the top region of the ventricle between the inflow and outflow orifice is observed from our simulation. Similar pressure profiles were reported in the literature (Wantanabe et al. 2004) for flow in LV with FSI. In this study, the incorporation of dynamic wall motions into simulations plays an important role in modelling the pumping function of the LV, thus regulating the flows during a cardiac cycle.

Although our simulation flow pattern results show reasonably good agreement with the comparison literature data simulation results, they are not identical. This is as expected as some minor discrepancies between the two are unavoidable, given that both simulations were based on different patient-specific LV geometries, boundary condition assignment methods, as well as geometry reconstruction methods. It is of note, however, that the main features of LV vortex evolution present in the comparison literature results were successfully captured by our simulation results as well.

Despite such encouraging results, our simulation method does have several limitations which may need to be further worked on in order to improve the overall efficacy of future simulations. Among the limitations of this study would be the lack of integration of patient-specific valve opening profiles as a function of time. Specifically, our current method which assumes an almost instantaneous opening or closing of the valve orifices may not accurately model time-dependent valve orifice sizes. This may hence cause our flow pattern results to deviate from patient-specific physiological patterns. Figure 10 shows pressure history obtained from our simulation at various locations. The results seem to not mimic physiological conditions accurately and in fact seem to suggest that the simulation flow is strongly affected by the trend of the velocity profile prescribed at both the inlet and outlet orifices in Figure 6. Generally, we can see that outflow from the ventricular chamber corresponds to a drop in pressure while inflow corresponds to pressure increase except for the pressure at the location of the aortic valve which was observed to deviate significantly from the other locations throughout most of systole. Although the pressure results indicate a significant deviation from physiological pressure profiles, where the pressure of blood in the LV is expected, for example, to increase rather than decrease during systole, it is likely that one of the main reasons for this is the absence of efficacious valve structures. An efficacious valve system such as the one which involves FSI such as that which was demonstrated by Washio and Hisada (2007) shows great potential for yielding simulation pressure profiles that approximate closely to physiological conditions. Meanwhile, pressure-based boundary conditions have also been considered in place of velocity-based boundary conditions, but they are generally not preferred as the acquisition of patient-specific LV pressure data to date cannot be done non-invasively. Alternatively, pressure data obtained from a simple model of the circulatory system, for example the Windkessel model (Westerhof et al. 2009) could serve as more suitable pressure boundary conditions. Future work will hence likely proceed in the direction of incorporating efficacious heart valve models as well as simple circulatory models into our current simulation framework.

6. Conclusions
We have shown a 3D numerical model of the human LV flow that has been generated using the proposed semi-automated approach. Given segmented surfaces from medical images, using the proposed approach, it is able to generate smoother surfaces as well as volume meshes, taking into account the dynamic motion of LV walls for CFD simulations. The process of conducting CFD simulations from MRI scan can be made automated from geometry reconstruction, mesh generation, moving volume mesh to solving for the flows. We observe generally good agreement between our results and that of the literature. However, there is still room for improvement and hence future work to improve on this model would likely be in the domain of efficacious boundary condition specifications. Looking ahead, in vivo validation of our flow pattern results will likely represent an important step towards making our computational flow modelling results clinically relevant.

Acknowledgements
This work is partially supported by the National Research Foundation, Singapore under its Cooperative Basic Research Grant and administered by the Singapore Ministry of Health’s National Medical Research Council (NMRC/EDG/1037/2011).

Figure 10. Pressure values at various points of the simulation geometry volume in a cardiac cycle.
References


