

Simulation tools for mitral valve repair

Viktoriya Antonova

Supervisors: Dr. ir. Matthieu De Beule, Prof. dr. ir. Benedict Verhegge
Counsellors: Tim Dezutter, dhr. Philippe Bertrand

Master's dissertation submitted in order to obtain the academic degree of
International Master of Science in Biomedical Engineering

Department of Electronics and Information Systems
Chairman: Prof. dr. ir. Jan Van Campenhout
Faculty of Engineering and Architecture
Academic year 2013-2014



Abstract

As the second most common heart disease, the mitral valve regurgitation has been subject of many researches through the years. With the improvement of the technology resources more and more computer simulations that regard the dynamics and effect of different repair techniques of the mitral valve are done. In order to produce a patient-specific mitral valve model there are many steps that should be performed. Every step requires time and knowledge about different software products. There is a need of a software tool that combines big part of the steps and makes the process more efficient.

In this study two demonstration tools that save time and resources are constructed. A web-based application is developed which, when installed on a server, can be reached by every computer with internet. In the application a patients-specific data can be loaded as well as various mitral valve parameters can be chosen. The application is designed to keep records of all uploaded data and to generate a file that later can be used in the second produced application.

The second application that was created, gives an opportunity to the user through its graphic user interface to load patient-specific data files as well as files with mitral valve parameters. The application generates a preview of the mitral valve finite element analysis model and makes an input file for Abaqus. With the help of the graphic user interface the user saves time preparing different simulations and eliminates the need for having previous knowledge of the used software.

Acknowledgements

I would like to express my gratitude to those who made this thesis possible. I am very thankful to my promoters, Dr. Matthieu De Beule and Prof. Benedict Verhegge for sharing their vast expertise and guiding me in completion of this thesis work, as well as for the opportunities they have provided me with. I would also like to thank PhD student Tim Dezutter and Evangelos Mantadakis for their collaboration while developing the software tools. Last, but, certainly, not least, I also owe gratitude to Dr. Phillippe Bertrand who shared his clinical knowledge and expertise.

Table of Contents

Abstract.....	1
Acknowledgements	2
Table of Contents.....	3
List of Figures	5
List of Tables	8
1. Introduction	9
1.1. Anatomy and physiology of the heart.....	9
1.2. Anatomy and physiology of the mitral valve	11
1.3. Pathology of the mitral valve	16
1.4. Mitral valve replacement	17
1.5. Mitral valve repair	18
1.6. Computational techniques used in the clinical practice	20
2. Literature review	23
2.1. Geometry of the mitral valve	23
2.2. Material models of the mitral valve.....	31
2.3. Boundary conditions	32
2.4. Computational analysis of mitral valve repair	33
2.5. Current research of mitral valve modeling in a spin-off of Ghent University	33
2.6. Aim of the thesis	34
3. Overview of the used software technologies.....	36
3.1. Programming languages.....	36
3.2. pyFormex.....	38
3.3. Django	39

4. Software development process.....	40
4.1. Software requirements.....	41
4.2. Software design.....	53
4.3. Implementation.....	58
4.4. Verification.....	74
4.5. Maintenance.....	78
5. Conclusions, limitations and future perspectives.....	80
References.....	82
Appendices.....	87
Appendix A: pyFormex application user manual.....	87
General information.....	87
System summary.....	88
Getting started.....	88
Using the system.....	93
Appendix B:Django application user manual.....	100
General information.....	100
System summary.....	101
Getting started.....	101
Using the system.....	105

List of Figures

Figure 1 Blood circulation in human body [1]	9
Figure 2 Heart valves [2]	10
Figure 3 Wiggers diagram [3].....	11
Figure 4 Mitral valve a. mitral valve in open position b. mitral valve in close position[4].....	12
Figure 5 Side view of the healthy mitral valve[5]	13
Figure 6 Carpenter's nomenclature[8]	13
Figure 7 The ventricular side of a porcine mitral[9]	14
Figure 8 Histologic sections of normal mitral valve. (A) Annulus.....	15
Figure 9 Cross section of chordae tendinae	16
Figure 10 A) Normal MV B)Mitral valve regurgitation [11]	17
Figure 11 Open commissurotomy [13]	18
Figure 12 Annuloplasty [13].....	19
Figure 13 Edge-to-edge repair[14].....	20
Figure 14 Mitral valve FEA models in research papers by: A. Kunzelman et al[15]., B.Kunzelman et al.[18], C Salgo et al.[22], D Maisano et al.[21], E Votta et al.[27], F Prot et al.[29], G Conti et al.[34], H Burlina et al[39], I Rim et al.[41]	30
Figure 15 Framework for mitral valve FEA modeling	34
Figure 16 Waterfall model	40
Figure 17 System architecture.....	53
Figure 18 Activity diagram pyFormex: Load a JSON file	55
Figure 19 Activity diagram pyFormex: Generate a JSON file.....	55
Figure 20 Activity diagram pyFormex: Preview of mitral valve FEA model.....	56
Figure 21 Activity diagram pyFormex: Generate a input file for Abaqus	56
Figure 22 Activity diagram web application: Generate a JSON file	57
Figure 23 GUI mitral valve pyFormex script	59
Figure 24 Structure of the GUI in pyFormex.....	59
Figure 25 GUI pyFormex : Mitral valve parameters tab	60
Figure 26 GUI pyFormex : Posterior leaflet material properties tab.....	61

Figure 27 GUI pyFormex : Visualization parameters tab.....	62
Figure 28 pyFormex: Structure of the button functions.....	62
Figure 29 Standard Django architecture[55]	66
Figure 30 File tree of Django project	67
Figure 31 Database diagram Django.....	69
Figure 32 Django login template	70
Figure 33 Django patient data list template	71
Figure 34 Django submit patient data template	71
Figure 35 Django loading data tab.....	72
Figure 36 Django mitral valve parameters tab	72
Figure 37 Django material properties tab	73
Figure 38 Django simulation tab.....	73
Figure 39 Tabs of MitralValvepyF	89
Figure 40 MitralValvepyF : Loading data tab.....	89
Figure 41 MitralValvepyF : Mitral valve parameters tab.....	90
Figure 42 MitralValvepyF: Material properties tab/ Anterior leaflet tab.....	91
Figure 43 MitralValvepyF : Material properties tab/ Posterior leaflet tab	91
Figure 44 MitralValvepyF : Visualization parameters tab	92
Figure 45 MitralValvepyF input dialog.....	92
Figure 46 MitralValvepyF :Loading TRC file pop-up window.....	93
Figure 47 MitralValvepyF :Visualization parameters color palette pop-up	95
Figure 48 MitralValvepyF: Export all parameters button pop-up	96
Figure 49 MitralValvepy: Export to Abaqus. inp file pop-up window	96
Figure 50 MitralValvepy: Successful generation of INPUT file message	97
Figure 51 MitralValvepyF: Preview of FEA mitral valve model pop-up.....	98
Figure 52 MitralValvepyF: Example of generated 3D FEA mitral valve model.....	98
Figure 53 MitralValvepyF: Import parameters button pop-up window.....	99
Figure 54 MitralValvepy: Import parameters button message window	100
Figure 55 MitralValveDj: Run server message.....	102

Figure 56 MitralValveDj: Login screen	102
Figure 57 MitralValveDj: Tab menu	103
Figure 58 MitralValveDj: Loading data tab	103
Figure 59 MitralValveDj: Mmitral valve parameters tab	104
Figure 60 MitralValveDj: Mmaterial properties tab	105
Figure 61 MitralValveDj: Simulation parameters tab	105
Figure 62 MitralValveDj: Welcome screen	106
Figure 63 MitralValveDj: Submit patient data screen	106
Figure 64 MitralValveDj: Verification of the file extensions.....	107
Figure 65 MitralValveDj: Linear elastic material model	108
Figure 66 MitralValveDj: Hyperelastic material model	108
Figure 67 MitralValveDj: List of submitted patient data entries	109

List of Tables

Table 1 Summarized information of mitral valve FEA literature review	29
Table 2 Stakeholder analysis.....	42
Table 3 Traceability matrix.....	78

1. Introduction

This chapter consists of a description of the heart anatomy and general function of it, followed by the anatomy and physiology of the mitral valve. The chapter continues with the pathology of the mitral valve and the current treatment methods. At the end of the chapter a brief overview of the computational techniques used in the clinical practice is made.

1.1. Anatomy and physiology of the heart

The heart is placed in the medial of the thorax. With the size of a fist and a mass up to 350 grams it is an essential organ without which our body can't live. The main function of the heart is to pump blood through the vessels to all parts of the body. This is possible thanks to the four chambers and the four valves in the heart. The left and right atriums are receiving chambers and left and right ventricles are discharging chambers. The blood circulates in the body through the pulmonary and systemic circuit (Figure 1.). The right side of the heart pumps deoxidized blood to the lungs, where the blood gains oxygen and goes back to the left side of the heart. From the left side of the heart the blood is pumped through the systemic circuit to all of the body parts, where it loses oxygen and gains carbon dioxide.

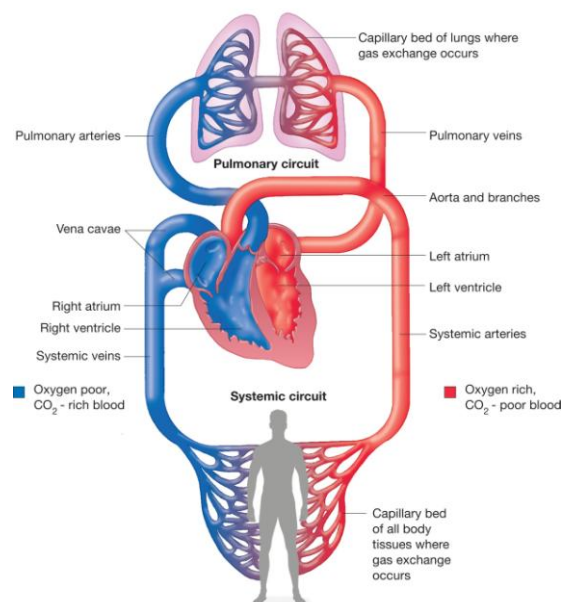


Figure 1 Blood circulation in human body [1]

The heart has four valves In order to regulate the blood flow: mitral, aortic, tricuspid, and pulmonary valve (Figure 2). If working properly the valves act like one-way valves which allow the blood to flow from one chamber to another or to flow out of the heart in one direction.

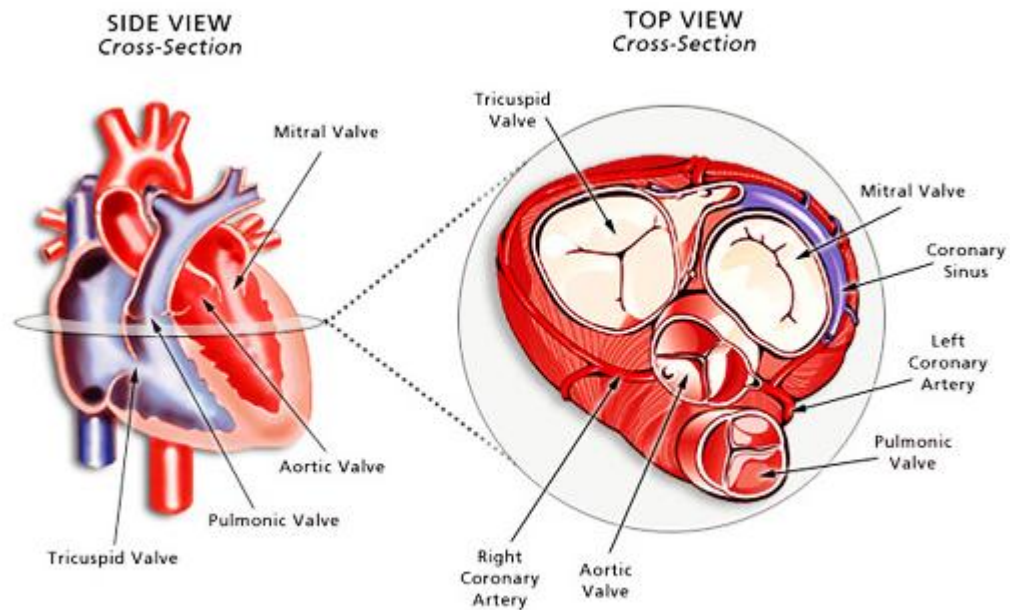


Figure 2 Heart valves [2]

The bicuspid or, also known as, mitral valve consists of two cusps and is located between the left atrium and left ventricle.

The tricuspid valve has three leaflets and is located between the right atrium and right ventricle.

The aortic valve also has three leaflets and is located between the left ventricle and the aorta.

The pulmonary valve consists of three leaflets and is located between the right ventricle and the pulmonary artery.

The valves open and close to allow blood flow. Their dynamics depend on the activity of the heart muscle i.e. when the ventricles are relaxed the aortic and pulmonary valves close and the mitral and tricuspid valves open in order to allow blood flow from the atria to the ventricles. In this period the atria contract, allowing income of blood to the ventricles (diastole)

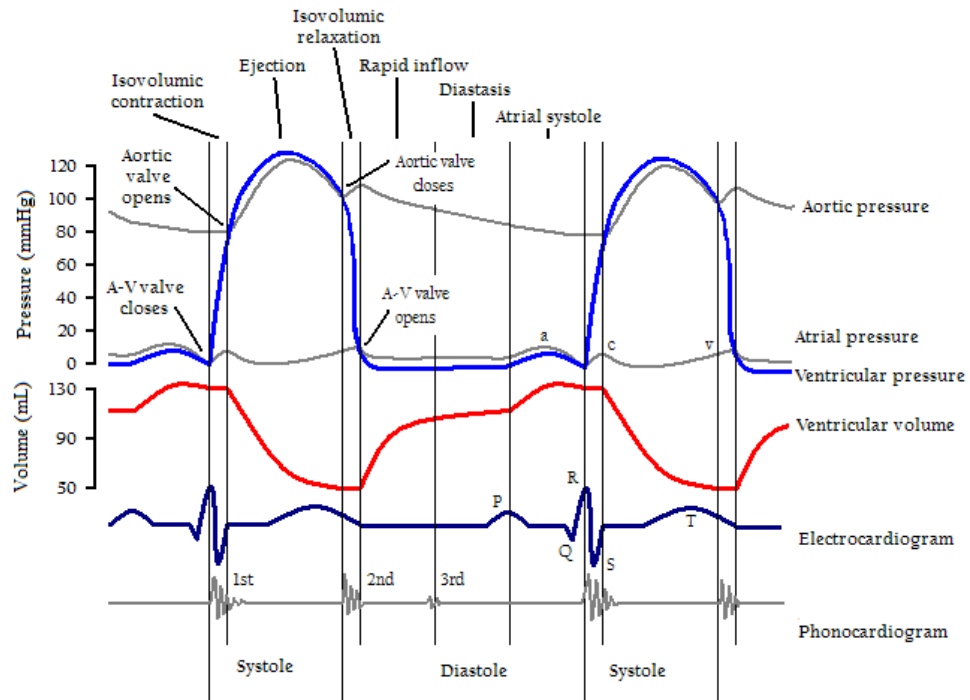


Figure 3 Wiggers diagram [3]

When the ventricles contract (systole) the mitral and tricuspid valves are closed. The aortic and pulmonary valves open to allow blood flow through the blood vessels

1.2. Anatomy and physiology of the mitral valve

The mitral valve (MV) or bicuspid valve is the only bi-leaflet valve in the heart. It is situated in the left side of the heart between the left atrium and the left ventricle. During diastole the blood fills the left atrium and raises the pressure there. Once the pressure is higher than the pressure in the ventricle the mitral valve opens and the blood is directed to the ventricle. At the end of the diastole atrial contraction occurs and the blood is transferred in the left ventricle. Then the mitral valve closes which prevents a reversal of the blood flow (Figure 4).

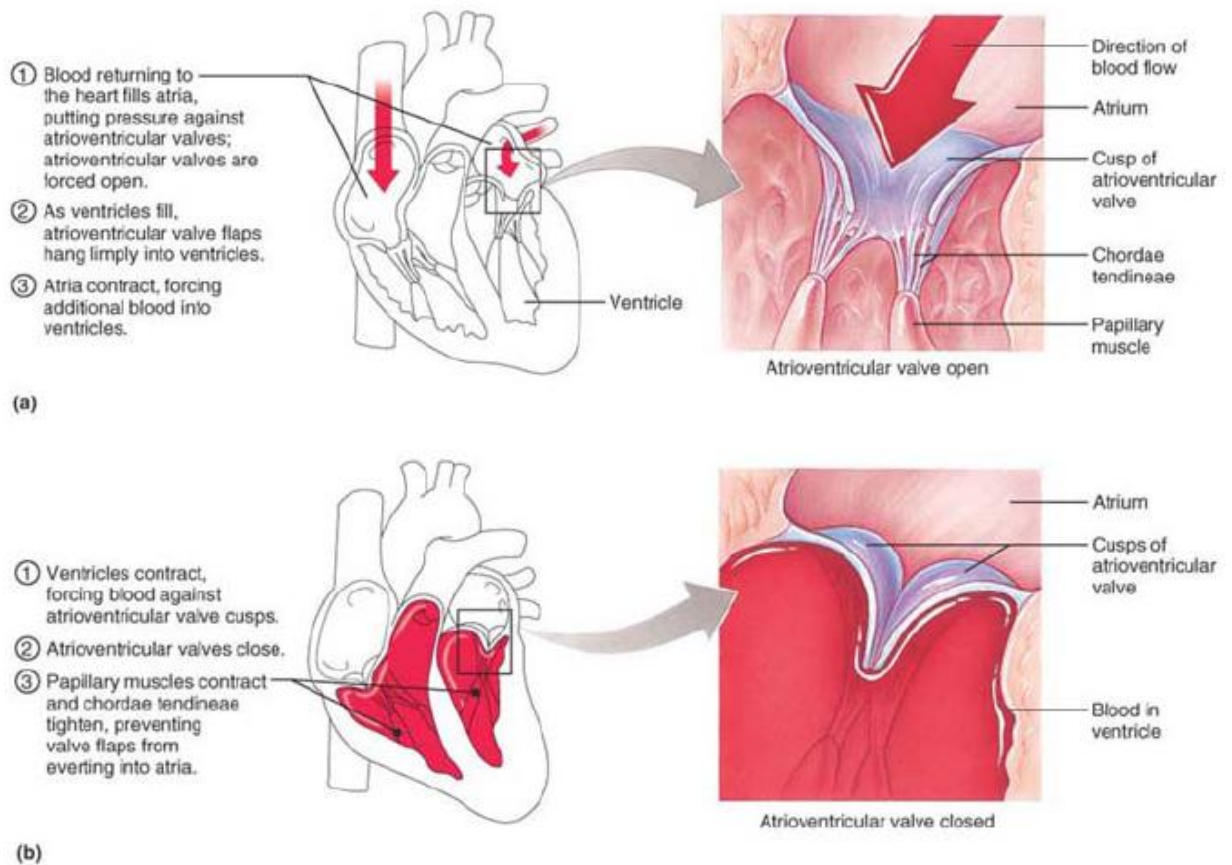


Figure 4 Mitral valve a. mitral valve in open position b. mitral valve in close position[4]

The mitral valve has a complex structure and the normal function of the mitral valve requires proper coordination between all of its components. On Figure 5 the elements of a healthy mitral valve are depicted. The MV has two thin, asymmetrical leaflets – anterior and posterior leaflets. They are attached to the left ventricle at the annulus (fibrous ring). The anterior leaflet is larger than the posterior but the latter having longer attachment to the annulus. The chordae tendinae are string-like structures that connect the free edge of the leaflets with the two papillary muscle heads.

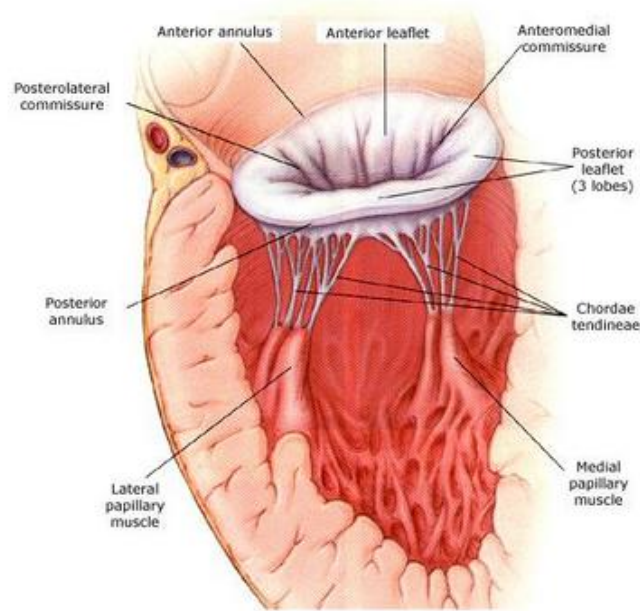


Figure 5 Side view of the healthy mitral valve[5]

1.2.1. Macroscopic analysis

The normal mitral valve consists of two leaflets, anterior (aortic) and posterior (mural). The posterior leaflet has two indentations that generally form three segments. There are two nomenclatures: Duran[6] and Carpenter[7] which divide the leaflets in different zones. Nowadays, the most widely used one is the Carpenter nomenclature in which every leaflet is divided by 3 zones as you can see on Figure 6. The zones at the anterior leaflets are A1 for the most lateral segment near to the anterior commissure, A2 for the central one and A3 for the medial segment near to the posterior commissure.

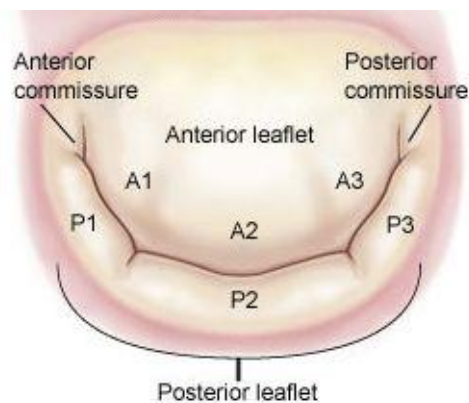


Figure 6 Carpenter's nomenclature[8]

Important role of the function of the mitral valve plays the branching chordae tendinae network which extends from the mitral leaflets to papillary muscles.

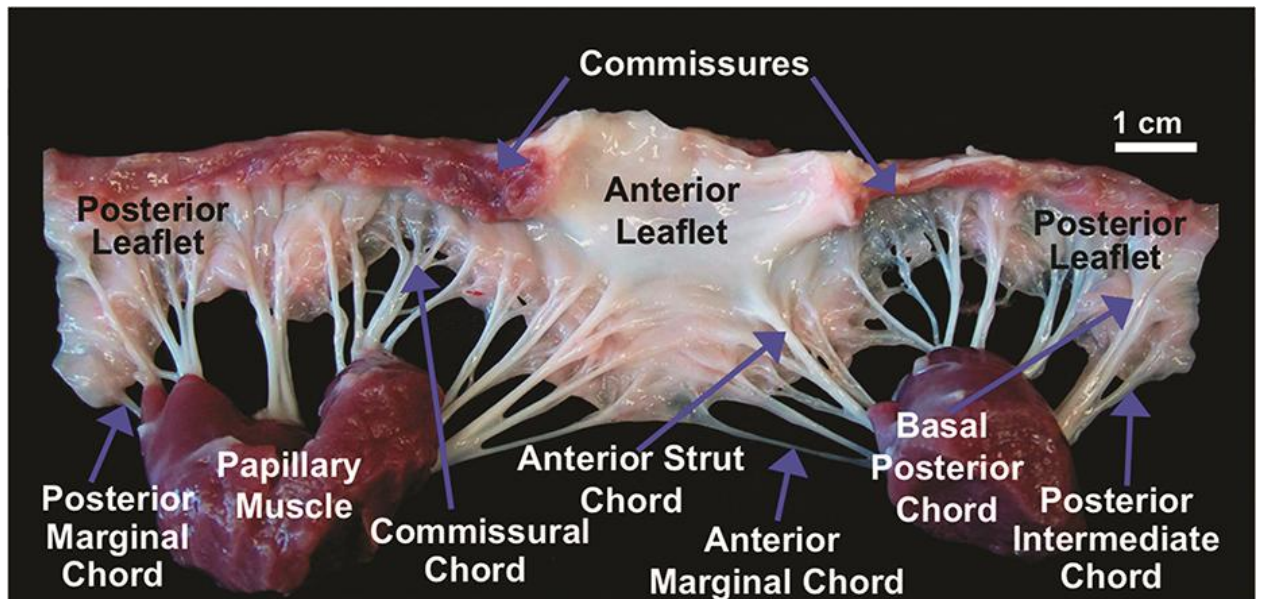


Figure 7 The ventricular side of a porcine mitral[9]

In general three types of chordae tendinae can be distinguished: marginal (primary) chordae which are connected to the free edge, strut (secondary) chordae which are inserted in layers to the rough zone of the anterior leaflets and basal (tertiary) chordae that extend from the papillary muscles and attach to the base of the posterior leaflet. Anatomically usually in the healthy mitral valve, the marginal chordae are thinner than the basal chordae, while the strut are the thickest. The main role of the marginal chordae is to maintain leaflet flexibility and governing proper valve closure, while the main role of the basal chordae is to provide a constant connection between the annulus and papillary muscles[9].

1.2.2. Microscopic analysis

Normal mitral leaflets have four layers that can be seen in Figure 8. The layer marked with A is the Auricularis, S is Spongiosa, F stands for Fibrosa and V comes from Ventricularis[10].

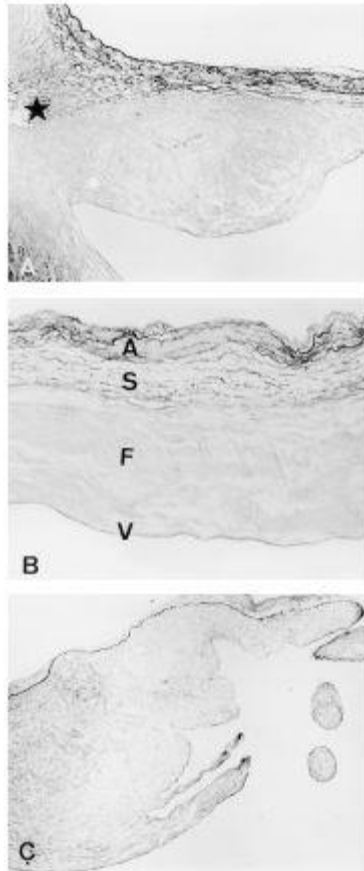


Figure 8 Histologic sections of normal mitral valve. (A) Annulus and basal third of the leaflet. (B) Middle third of the leaflet. (C) Distal third of the leaflet.

The auricularis is a thin of loose connective tissue containing elastic fibers and collagen which is the main component of the annulus. Spongiosa is a loose connective tissue with abundant proteoglycans, few elastic fibers and few collagen fibers. It extends along the entire length of the leaflet from the annulus to the free edge and is the main component of the free edge. The fibrous consists of bundle of collagen fibers which extend from the annulus. The ventricularis consists of loose connective tissue that is rich in collagen fibers which extend to the chordae tendinae. The distal third of the leaflets contains only fibrous and spongiosa (Figure 8 (C)).

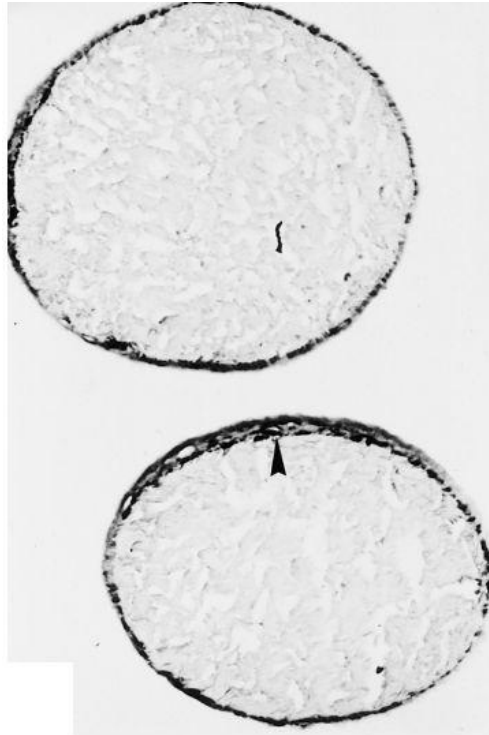


Figure 9 Cross section of chordae tendinae

At Figure 9 two cross sections of chordae tendinae are shown. The center cores of the chordae are composed of longitudinally oriented collagen bundles. The core is surrounded by elastic fiber (see the arrow at Figure 9).

1.3. Pathology of the mitral valve

Conditions like rheumatic fever, calcium deposits forming around the mitral valve, radiation treatment to the chest, mitral valve prolapse, infective endocarditis, congenital heart abnormality, etc. can lead to abnormal function of the mitral valve. The well-known classification of the mitral valve disease names the different conditions mitral stenosis (MS) and mitral regurgitation (MR). Mitral stenosis means that the MV is narrowed and blood's passage from the left atrium to the left ventricle is obstructed. If MR is present there is leakage of blood back to the left atrium when the ventricle contracts. That ultimately leads to overworking of the heart since it impedes the pumping of enough oxygenated blood.

Nowadays MS is not that common disease but MR is the second most common valvular disease in Europe. In Figure.10 you can see the difference between a normal MV and a MV with regurgitation. Enlargement of the left ventricle can lead to dilation of the annulus, displacement of the papillary muscles, rupture of the chordae tendinae. All of these abnormalities of the MV can lead to decreased coaptation between the leaflets and leakage of blood.

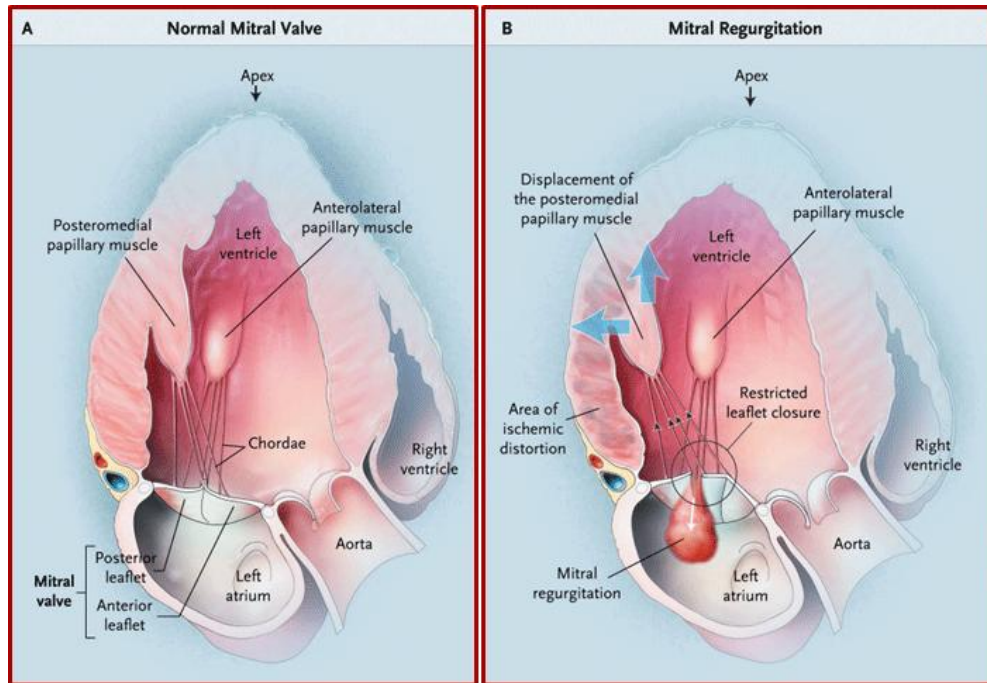


Figure 10 A) Normal MV B) Mitral valve regurgitation [11]

The treatment of MR depends on the patient-specific problem, their symptoms, causes of the MR and also presence of other medical conditions. People with mild or moderate MR may not require any specific treatment. Once the MR is severe and chronic, surgical repair or replacement can be performed.

1.4. Mitral valve replacement

When the MV has end-stage disease and MV replacement is needed, the native valve is surgically replaced by a mechanical or bioprosthetic valve. The replacement valve is most

effective modality and gold standard treatment for mitral regurgitation. Although it is very durable, the replacement valve is also a foreign material for the body and a patient with mechanical valve should take anticoagulation medicines throughout all their life. The biological valves can be from animal origin, usually from porcine heart. The biological valves have better haemodynamical performance than the mechanical ones, although the blood flow is not completely physiological. The disadvantage of the biological valves is their comparatively low durability with an average of 10-15 years. Because of this drawback they are mostly used among elderly people.

1.5. Mitral valve repair

Studies in recent years shown that mitral valve repair is more successful than replacement with a better survival rates[12]. In the last decade many different techniques for MV repair have been developed depending on the patient specific problem. The surgeon may need to shorten the leaflets length, replace or relocate some of the chordae tendinae, fix the annulus shape, reposition the papillary muscles or just fix the hole in the leaflet. At the next figures you can see the most commonly used mitral valve repair techniques.

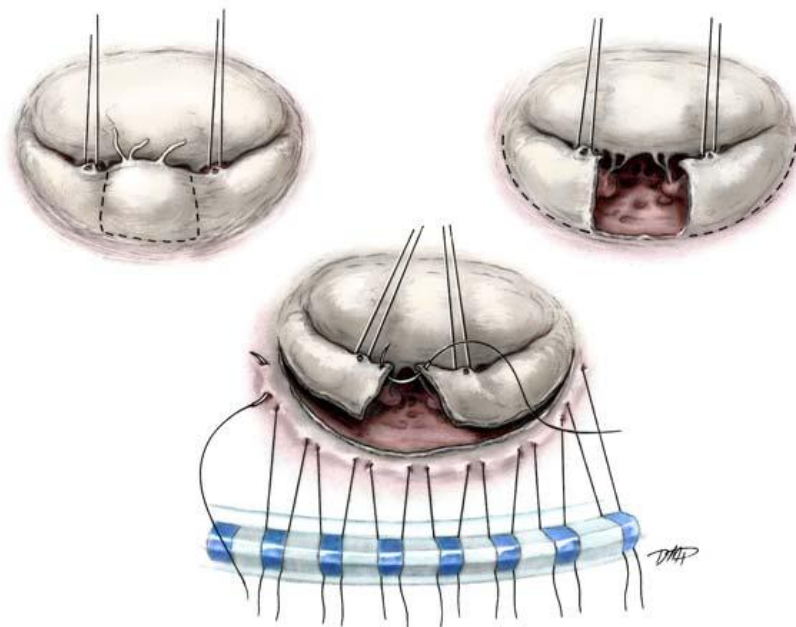


Figure 11 Open commissurotomy [13]

Figure 11 shows the open commissurotomy procedure. The mitral commissurotomy is aimed at opening up the corners of the valve, for a proper whole size and adequate coaptation of the valve leaflets.

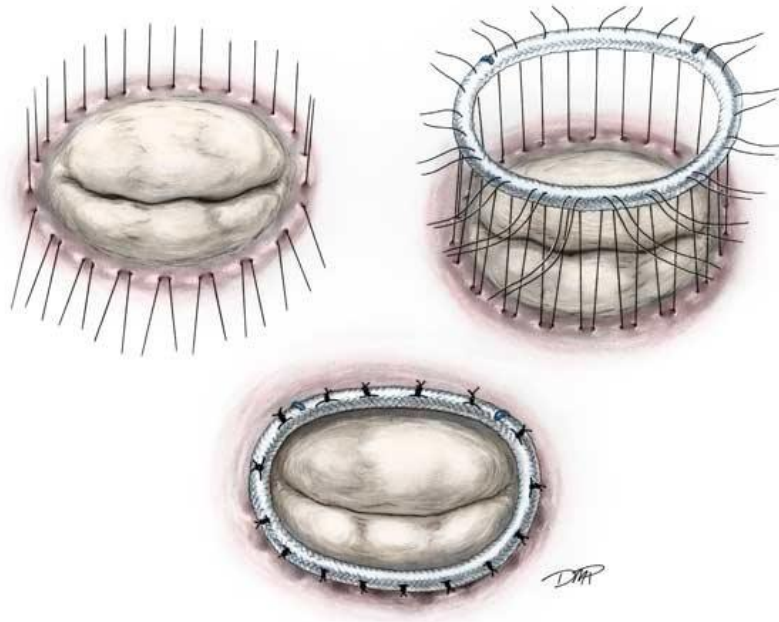


Figure 12 Annuloplasty [13]

At Figure 12 the annuloplasty repair technique is depicted. A surrounding device called annuloplasty ring is implanted in order to pull the leaflets together and to facilitate coaptation and re-establish proper mitral valve function.

Edge-to-edge repair technique is visualized at Figure.13. Suturing of the anterior and posterior leaflets creates a double-orifice mitral valve. The suturing can be performed with sutures or implanted clip.

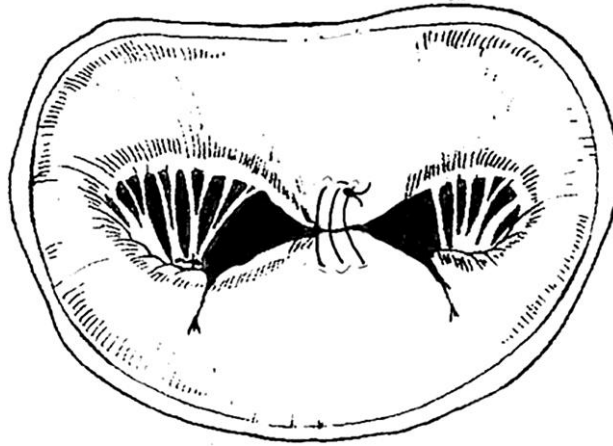


Figure 13 Edge-to-edge repair[14]

1.6. Computational techniques used in the clinical practice

1.6.1. Added value

The pressure, stresses, strains and shear rates are important when considering damage or rupture of the tissue and also are involved in various biological processes. Fluid dynamics and the information about the shear stresses along the leaflets of the mitral valve can give useful information about the hemodynamics of the mitral valve. This information is hard to obtain in vivo or in vitro because of the complex 3D shape of the mitral valve and also the high resolution that is needed. Therefore computer simulations are a useful tool that can help to understand the patient-specific problem. With the help of imaging and the data that can be extracted from it, the computer simulations could be more realistic and will aid doctors and researchers in their decision making.

1.6.2. Finite element analysis (FEA)

The finite element analysis method (FEA) is designed for use on computers and allows solving physical problems associated with complicated geometries on differential equations. It is used to design and improve the mechanical behavior of products and industrial applications as well as to aid the simulation of complex physiological and biological systems. The variety of problems that it can be applied to has grown enormously.

The element in the name of finite element analysis means that the structure usually is divided into elements, defined by points that are called nodes. Then the governing equations are calculated for every element and when one looks at the elements as a whole, there is an approximation of the solution for the simulated problem. The key aspects while modeling a FEA are the geometry, the material properties and the boundary conditions.

1.6.3. Computational fluid dynamics (CFD)

The computational fluid dynamics (CFD) is a branch of fluid mechanics that uses numerical methods and algorithms to solve and analyze problems on the flow of substances. Computers are used to perform millions of calculations required to simulate the interaction of the liquid and the gases with complex surfaces.

The method involves definition of the geometry as well as the volume occupied by the fluid is divided into discrete cells (the mesh). Boundary conditions and conservation equations are also defined. After solving the equations the result is visualized.

1.6.4. Fluid-structure interaction (FSI)

Fluid-structure interaction (FSI) is the interaction between a fluid flow and a moving or deforming structure. The fluid flow applies forces on the adjacent structure, which results in deformation of the structure. This deformation affects the fluid flow as well. Governing equations are usually defined and solved in order to see the result of the simulation.

2. Literature review

While speaking of the FEA modeling in respect to the mitral valve, one can distinguish different methodologies applied to the geometry and the used material models. In the early days of FEA modeling, researchers used simplified and idealized geometries, while in the past decade more and more patient-specific and complex geometry models are being applied. Also, in the construction of the material models utilized in the past, only linear elastic properties were used, while today a much greater variety of FEA models exists which describe the behavior of the biological materials by making use of more realistic non-linear constitutive model.

2.1. Geometry of the mitral valve

A realistic and accurate reconstruction of the mitral valve geometry is hard to obtain even nowadays. The complex morphology of the MV and the difficulty in measuring and examining it make the modeling of the geometry very hard. During the years different approaches of obtaining the geometry were used - from an idealized, general shape to more a patient-specific one using the available technologies like computer tomography and echocardiography. In general, in the first published paper on the matter in 1993 (Kunzelman et al), the obtained data is reconstructed to 3D model of the mitral valve, including the annulus, leaflets, chordae tendinae and the papillary muscles. There the geometry of the mitral valve is obtained from porcine heart in open position. The leaflets are with given uniform thickness, the chordae tendinae are defined as straight, single lines and the papillary muscle heads are represented as a single point to which the chordae tendinae are attached[15]. The FEA model of the mitral valve from this research is used for several studies by the same authors[16][17]. They utilize it to investigate the annular dilatation increase of the stress in the mitral valve and delays of the coaptation[18], as well as to compare the flexible versus rigid ring annuloplasty for mitral valve annular dialation[19]. In these studies they assume the symmetry of the valve in order to

reduce the computational time. Also, a non-branching chordae network is assumed. In 1997 in the study by Kunzelman et al the FEA model of the mitral valve is no longer symmetric and basal chordae tendinae are taken into account. [18]

A second research group that reconstructs mitral valve in 1999 in Italy was the group of Maisano et al. They investigate the hemodynamics of the normal MV and examine the effect of repair for mitral valve regurgitation[20]. For this study a simplified model of the MV in open and static position without representing the chordae tendinae is used. The shape of the annulus is also simplified to a circular one – similar to an annuloplasty ring. In 2002 the same Italian research group designs a new, more accurate model of the mitral valve with which they examine the stresses during systole and diastole in order to obtain their distributions on the leaflets after edge-to-edge repair of the mitral valve regurgitation[21]. In this study they also represent the chordae tendinae as straight lines connected from the free margin to the papillary muscles.

Also in 2002 the American research group of Salgo et al. studied the effect of the annular shape on leaflet curvature. They produce a model of the closed mitral valve without taking into account the chordae tendinae[22].

In 2003 the research group of Einstein et al. uses the geometry model of the mitral valve previously done by Kunzelman to implement the nonlinear, anisotropic hyperelastic material model[23] and later in 2005 to produce the first non-linear fluid-coupled computational model of the mitral valve[24].

In 2005 the Italian research group of Maisano et al. produces a new mitral valve geometry model. They use the literature data to define the geometry with idealized leaflets and free margin defined by a sinusoidal function [25]. Thus, they modeled three cases: the first model has a circular annulus, the second model - a D-shaped annulus representing the implantation of a typical annulus ring and the third model modified-shape prosthesis. Along with that, they also model forty chordae tendinae with a constant section and fixed papillary muscles head. Later in 2007 the same group incorporates the different thickness of the leaflets at the area adjacent to

their free edge. It is important to note that they model the FEA simulation with 52 chordae but still the geometry of the mitral valve is symmetric as in all of their previous research.

In 2005 Lim et al. use 12 transceiver crystals placed in the mitral valve in order to better capture the boundary conditions and to obtain a more realistic geometry of the mitral valve. Their mitral valve model is asymmetrical with a twisted, saddle-shape mitral annulus geometry[26]. They model only marginal chordae with uniform thickness and also represent the leaflets like a single surface without distinguishing the anterior and posterior leaflets.

In 2008 Votta et al. use a new approach to define the mitral valve geometry. They apply ultrasound data to extract the annulus and papillary muscle motion. The end-diastole valve configuration, when open is chosen because the pressure there at that moment is minimal. Transthoracic real-time three-dimensional echocardiography has given them the opportunity to manually select points on the annulus and then interpolate the data with a sixth-order Fourier function in order to get the continuous annular profile[27]. The geometry of the leaflets is modeled as in their previous research [14,18,[28]. They implement a different thickness of the anterior and posterior leaflets and model 58 chordae, all of which are marginal with the exception of two which are strut. Basal chordae are not included.

In 2009 the Norwegian research group of Prot et al. models the geometry of the mitral valve using three-dimensional echographic measurements from a porcine heart. The shape of the annulus is a symmetric non-planar ellipse. They also perform an autopsy measurement from which the length of the annulus and distance from the annulus to the anterior and posterior leaflets are being extracted. Innovative in their research is that they model a branched chordae network with branches starting from the middle point of the marginal chordae. A total of twenty marginal and two basal chordae are modeled[29]. In 2010 the same research group publishes a paper in which the mitral valve in healthy and pathological state is being examined. Based on their own measurements for the deceased mitral valve they change the thickness of the anterior and posterior leaflets and implement different values for the cross-section of the chordae tendinae[30]. In another research of the same authors they show that the leaflets have

3 layers, each with different material properties[31]. In this study the effects of different material properties across the thickness of the mitral valve are investigated.

In 2010 Wenk et al. construct a FEA model of the left ventricle that including the mitral valve too. With the help of magnetic resonance imaging (MRI), they extract the geometry of the leaflets and thus, different thickness for the anterior and posterior leaflets are implemented. Since the chordae tendinae can't be captured by MRI, they are approximated from anatomic images. Twenty four marginal and eight basal ones, with different material properties are modeled. The points where the chordae tendinae connect to the papillary muscles are modeled to distribute the load, instead of concentrating them in single points[32].

The geometry of the mitral valve is successfully extracted from 3D Transesophageal Echocardiography (3D-TTE) in 2010 by Burlina et al. They obtain it in open position and predict the configuration at end systole[33].

Conti et al.(2010) investigate the mitral valve modeling in ischemic patients where they obtain the mitral valve geometry from cardiac magnetic resonance data and thus build three models – one for a healthy mitral valve and two representing the mitral valve regurgitation associated with the ischemic disease. They reconstruct the mitral valve model by segmentation of the imaging data[34].

In 2011, the research group of Inge et al. models the geometry of the mitral valve by obtaining the data from 3D-microCT (CT stands for Computer Tomography). They place 20 onyx glass bead markers in ovine heart. In this way, after segmentation the leaflets and the chordae tendinae are being constructed[35].

The same year, Jassar et al. publish a paper about mitral valve modeling using real-time three-dimensional echocardiography where they make segmentation at midsystole using different axis and rotational template with eighteen long-axis. In each one of the eighteen axes they identify two annular intersection points. The anterior and posterior commissures are identified also. The coaptation zone between the anterior and posterior leaflets is captured across the entire extend of the mitral valve[36].

Quaini et al.(2012) produce a three –dimensional computational fluid dynamics model (CFD) of the mitral valve flow with the aim of validation against in vitro standards and 3D color Doppler methods. They model the chambers of the heart as idealized cylinders with a circular and rectangular orifice. A 2D spectral Doppler transducer and a 3D-color Doppler transducer are used to access trans-orifice flow[37].

Wang and Sun published a paper in 2012 about the modeling of mitral valve dynamic deformation using patient-specific multi-slice CT scans. There they segment the scans from middle diastole (after mitral valve opening and before closure) and middle systole (after mitral valve closure and before opening). From the CT scans they obtain the geometry of the left ventricle, mitral valve, chordae tendinae, papillary muscles and chordae intersection points. Their model has 21 chordae tendinae[38].

Burlina et al. use 3D echocardiography to predict a patient-specific mitral valve closure in 2013. After the automatic segmentation, the authors refine the geometry into a 3D model of the open mitral valve during diastole which is used as an initial configuration for the prediction of the valve in closed state. The length of chordae tendinae is automatically set. The model includes the basal and marginal chordae, fourteen in total. The papillary muscle heads are manually specified[39].

In 2013 Pouch et al publish a research about semi-automated mitral valve morphometry where the authors study the occurring stresses. They construct the geometry of the mitral valve from ultrasound data. Real-time 3D transesophageal echocardiography (3D TEE) and volumetric images of the mitral valve are segmented at mid-diastole for two patients. The first patient is healthy and the second one has severe ischemic mitral regurgitation. They model a total of sixty four chordae, while the papillary muscle heads are represented by points[40].

The research group of Rim et al. studies the effect of patient-specific annular motion on dynamic simulation of the mitral valve in 2013. The geometry of the mitral valve is obtained from 3D TEE for two subjects – a healthy one and another which has mitral regurgitation. The leaflets and the annulus are segmented from the ultrasound data. The PM tips are modeled as continuously deforming. The chordae tendinae are represented as line elements from the

papillary muscle heads to the edge nodes of the mitral valve leaflets. From the 3D TEE data, the authors extract the annulus geometry at peak systole and end diastole. In order to obtain dynamic annular motion they apply time-varying nonlinear nodal displacement of the nodes across the annulus[41].

In Table 1 summarized information for most of the reviewed papers is shown.

Table 1 Summarized information of mitral valve FEA literature review

Research group	Year	Geometry	Material models	Loading	Tasks
Kunzelman et al.	1993, 1995, 1996, 1997, 1998	*asymmetric leaflets *different thickness on the posterior and interior leaflet *porcine heart	*linear orthotropic *collagen fibres	pressure curve from literature	*effect of chordal replacement with sutures on valve stress *the effects of annular dilatation on coaptation *compare two types of annuloplasty ring prostheses *support a study on altered leaflet collagen in response to increased leaflet stress
Votta et al.	2002, 2007, 2008, 2009, 2011	*asymmetric leaflets *no basal chordae *no branched chordae * Ultrasound and Transthoracic real-time three-dimensional echocardiography *human	*leaflets - hyperelastic anisotropic *chordae - hyperelastic isotropic	pressure curve from literature	*analysis of the stress on mitral valve leaflets, edge-to-edge repair *compare different mitral annuloplasty prostheses
Prot et al.	2007, 2008, 2009, 2010	*symmetric *branched chordae *three- dimensional echographic measurements *basal and marginal chordae	*leaflets - hyperelastic anisotropic *chordae - hyperelastic isotropic	pressure curve from literature	*analysis of healthy and pathological human mitral valves *comparison between human and porcine data *determine the effects of using different material properties across the thickness of the valve *annulus shape effect and chordal force distribution
Einstein et al.	2003, 2007, 2011	*symmetric *branched chordae *in vivo data	*leaflets - hyperelastic isotropic *chordae- tension-only cables	atrial and ventricular pressure based on in vivo porcine measurements	*fluid structure interaction investigating the behaviour of the normal and pathological mitral valve
Wenk et al	2010, 2012	*non-symmetric *basal and marginal chordae *no branching *magnetic resonance images *sheep	* leaflets - hyperelastic, transversely isotropic *chordae-cable element formulation	measured in vivo end-diastolic and end-systolic LV pressures	*The Effect of Mitral Annuloplasty Shape in Ischemic Mitral Regurgitation *Insights into Ischemic Mitral Regurgitation *Model of the Left Ventricle With Mitral Valve

At the Figure 14 different FEA models of the mitral valve used in various research papers is depicted.

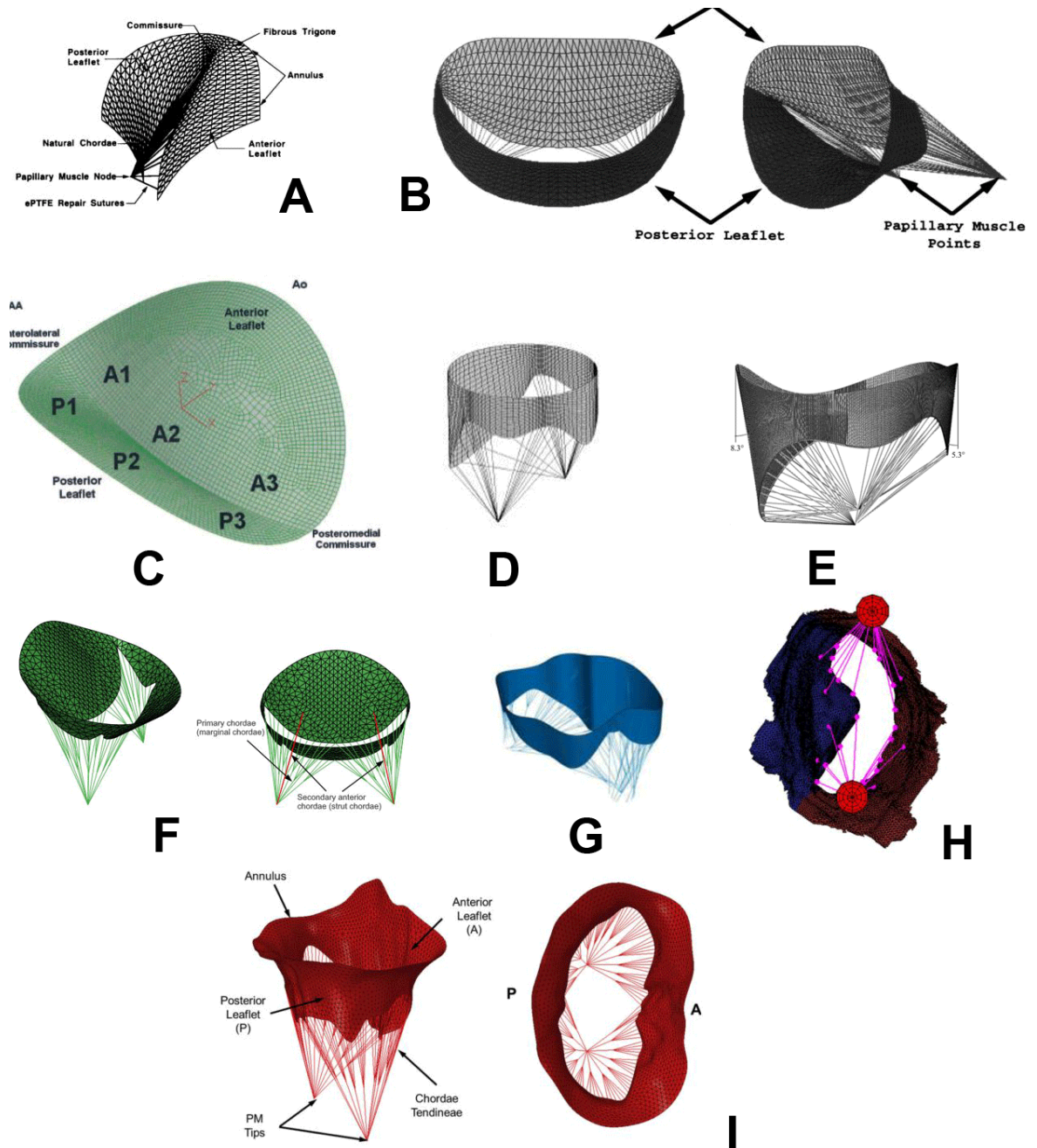


Figure 14 Mitral valve FEA models in research papers by: A. Kunzelman et al.[15], B. Kunzelman et al.[18], C. Salgo et al.[22], D. Maisano et al.[21], E. Votta et al.[27], F. Prot et al.[29], G. Conti et al.[34], H. Burlina et al.[39], I. Rim et al.[41]

2.2. Material models of the mitral valve

The microstructure of the mitral valve shows that the leaflets and the chordae tendinae tissue are constructed from cells and fibrous tissue. Because of various orientations and types of the fibers, the behavior of the tissue is nonlinear, transversely isotropic and elastic. The leaflets of the mitral valve have different material properties along the radial and circumferential direction while in the third direction they exhibit certain similarities. All that leads to a transversely isotropic behavior of the mitral valve leaflets.

In the past years a lot of improvement has been made considering material models for the mitral valve despite the comparatively scarce availability of sources of material data, mathematical models and also easy implementation the different material models.

In the first FEA model of the mitral valve (1993), Kunzelman et al. implement linear elastic model for the leaflet tissue. In order to find the elastic moduli in the radial and circumferential direction they perform a uni-axial tensile test[15]. The same methodology is used during the years for many other studies[18],[25],[28].

In 1995 May-Newman et al. make a biaxial test of excised porcine mitral valve leaflets in order to get more accurate stress-strain curves. They perform the test of leaflet specimens in both principle directions. Results from the test show that both anterior and posterior leaflets behave anisotropically, the obtained stress-strain relations are highly non-linear. They also find that, in general, in both principle directions the curves follow the same relation[42]. Based on the results of this study later in 1998 the authors define a constitutive law describing the mitral valve material properties[43].

After the publishing of the constitutive law by May-Newman, the first successful implementation for FEA analysis is performed by Einstein et al. in 2003[23]. Additionally, one year later, Einstein et al. implement the first non-linear material properties for mitral valve[44],[45].

Conti et al also utilize the constitutive law by May –Newman to describe the material behavior of the leaflets[34].

In 2005 the research group of Holzapfel publishes a paper in which they determine another non-linear constitutive mode which, in turn, is implemented in 2007 by Prot et al. in their FEA model of the mitral valve[46]. This material model is again used in later publications[30], [31] of theirs. Later in 2008 the Holzapfel model is, as well, implemented in the FEA model of Votta et al[27].

2.3. Boundary conditions

Depending on the goal of study, applied loads and boundary conditions are different. There are three main analyses, as mentioned above: finite element analysis, computational fluid dynamics analysis and fluid structure interaction. Currently, most of the published studies examine the structure of the mitral valve. In FEA analysis the blood flow through the mitral valve is neglected while in CFD analysis normally the deformation of the leaflets of the mitral valve is neglected and the focus is on the blood flow through the valve. In contrast, in FSI simulations both the structural deformation and the blood flow are captured.

In the FEA analysis the blood flow is simulated by applying pressure load curves which are usually based on medical data. The authors mainly use the pressure difference between the left ventricle and the left atrium derived for the whole cardiac cycle[15]. Some research groups like that of Maisano use a linear pressure curves[21], [25], [27], [28]. The groups of Prot and Kunzleman use more accurate transvalvular pressure profiles[17],[43], [47].

The boundary conditions used in the modeling of the mitral valve are really similar and reflect the physiological aspects of the valve. Usually the boundary conditions are applied at the annulus and the papillary muscles. A typical assumption made in many studies is that the papillary muscles head are fixed. Some more recent published papers investigate the dynamics of the annulus and the papillary muscles while applying more complex boundary conditions combined with loads[25], [27], [41],[47].

2.4. Computational analysis of mitral valve repair

Because of improvements made in past years regarding the modeling of the geometry of the mitral valve and the applied material properties, more and more researchers study the effects of the mitral valve repair techniques. Investigation about the mitral annuloplasty ring size based on the patient-specific data were made by Votta et al[28] and Stevanella et al[48]. In 2009 Schievano et al. compare the single balloon technique versus the double balloon one for percutaneous mitral valve dilatation[49]. The edge-to-edge repair procedure was also simulated by Votta et al[21] and Avanzini et al.[50]. In 2011 Avanzini et al. conclude that the stresses and transvalvular pressure gradient are similar to those after the surgical edge-to-edge procedure[51]. Lau et al. also study the edge-to-edge technique with the help of FSI model on the idealized geometry of the mitral valve in 2011[52].

2.5. Current research of mitral valve modeling in a spin-off of Ghent University

Currently FEops, a spin-off from IBiTech-bioMMeda a research group of Ghent University works in the framework of finite element modeling of the mitral valve. The study is based on in-vivo 3D-TEE images from which the mitral valve geometry and annular and papillary muscle motions are extracted semi-manually (Figure 15).

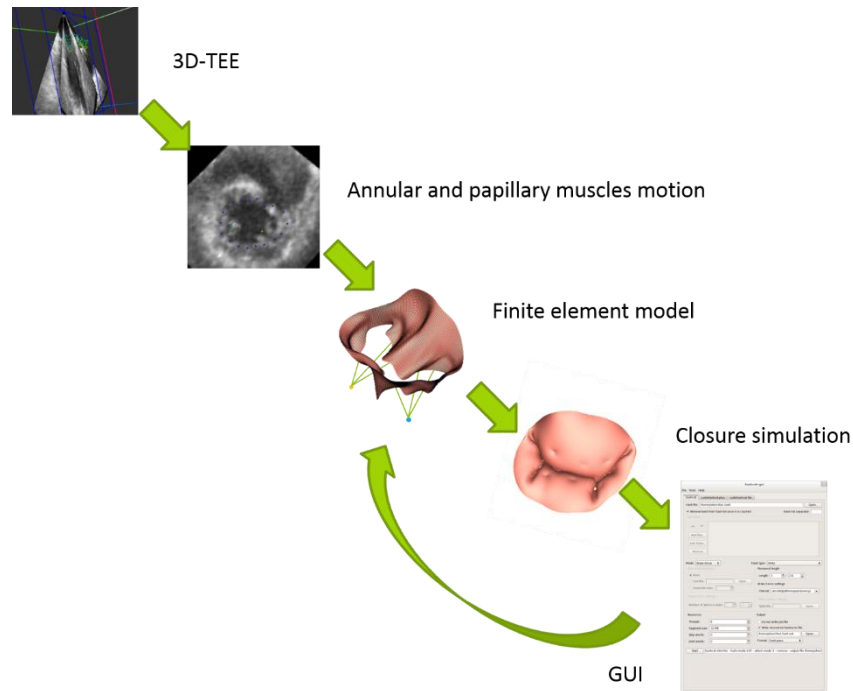


Figure 15 Framework for mitral valve FEA modeling

Once the data is obtained, it's further post-processed in pyFormex. With the help of pyFormex, a patient-specific FEA model is generated. There is also a possibility to generate the input file for ABAQUS in which the patient's specific annular and papillary muscle motion is taken into an account. Also a patient-specific measured transvalvular pressure is integrated into the input file. Once all of the data is implemented in the input file for ABAQUS, closure simulation can be done. The research group is also working on the graphic user interface in pyFormex and a web based application in order to help people without programming skills to interact with the available resources.

2.6. Aim of the thesis

The treatment of MR is a patient-specific problem, based on symptoms, causes of the MR and also presence of other medical conditions. Because of this, medical doctors are sometimes

impeded to decide which method to use for the patient-specific problem. Currently there is no software tool available by which the doctors can simulate the disease of a given person, experiment and thus, find out which treatment will give the best results. Also for the researchers that are interested in improving and developing new technologies there is a need for a user friendly and easy-to-use tool that can be utilized from anywhere in order to help them to better understand the hemodynamics of the mitral valve. Such a system can improve the work of the medical doctors and researchers. The future ability to simulate any problem and also simulate any treatment will save time and might also permit process impact.

The aim of the present thesis is to make a demo for a graphic user interface (GUI) for the existing mitral valve modeling script in pyFormex. Another goal is to make web based application using the Django framework. With the help of a GUI in pyFormex each interested user will have the opportunity to load different patient-specific data, to change chosen parameters considering the mitral valve geometry and also to implement different material properties for the leaflets and chordae tendinae, without any programming knowledge. With the help of Django a web based basic version will be made and every user will have access to the framework remotely.

3. Overview of the used software technologies

3.1. Programming languages

3.1.1. Python

Python is an interpreted programming language which emphasizes a very clean syntax and encourages readable code.

It is a multi-paradigm programming language, since it supports object-oriented, imperative programming and to a lesser extent, functional programming. It is an interpreted language, uses dynamic typing and is multiplatform.

It is managed by the Python Software Foundation. It has an open source license, called Python Software Foundation License, which is compatible with the GNU General Public License from version 2.1.1.

3.1.2. HTML

HTML stands for HyperText Markup Language and refers to the markup language for developing web pages. A standard, serving as a reference for the development of web pages in different versions, it defines a basic structure and a code (called HTML) for defining the contents of a web page, such as text, images, etc. It is a standard by the W3C, an organization dedicated to the standardization of almost all technologies related to the web, especially with regard to writing and interpretation. It is the language which web pages are defined on.

The HTML language development is based on referencing. To add an external element to the page (image, video, script, etc.), this is not embedded directly in the page code, but a reference of the location of that item is done using text blocks. Thus, the web page contains only text,

while the web browser (code interpreter) has the task of uniting all the elements and, consequently, displays the final page. As a standard, HTML seeks to be a language that allows for any web page written in a particular version to be interpreted in the same way (standard) for any updated browser.

3.1.3. CSS

Cascading Style Sheets (Cascading Style Sheets) is a style sheet language used to describe the look and formatting of a document written in a markup language, including several HTML and XML-based languages such as XHTML or SVG.

Style information can be attached as a separate document or in the same HTML document. In the latter case, general styles could be defined in the header of the document or in a particular label by a "<style>" attribute.

3.1.4. JavaScript

JavaScript (commonly abbreviated " JS ") is an interpreted programming language, a dialect of the ECMAScript standard. It is defined as object-oriented prototype-based, imperative, weakly typed and dynamic.

It is mainly used in form client side, implemented as part of a web browser enabling improvements in the user interface and dynamic web pages but there is a way-side JavaScript (Server-side JavaScript or SSJS). Use in external to the web, applications such as PDF documents, desktop applications (mostly widgets) is also significant.

JavaScript was designed with syntax similar to C, although names and adopts conventions from the Java programming language. It is important to note that Java and JavaScript are not related in terms of semantics and purposes.

All modern browsers interpret JavaScript code embedded in Web pages. To interact with a web page, the JavaScript language provides an implementation of the Document Object Model (DOM).

3.1.5. jQuery

jQuery is a JavaScript library, which simplifies the way to interact with HTML documents , manipulate the DOM, handling events , developing animations, and add AJAX interaction with art websites.

jQuery is free, open source software, has a dual-licensed under the MIT License and the GNU General Public License v2, allowing their use in free software projects. jQuery, like other libraries, offers a number of features based in JavaScript that would otherwise require a lot more code. With the functions of this library great results are achieved in less time and space.

3.2. pyFormex

pyFormex is a Python based program which allows the user to generate and manipulate large and complex geometric models of 3D structures. It has a single and consistent environment which provides many features that usually can be done by other CAD systems but none of them offer all of these features in single software. The core idea of pyFormex is that the 3D geometry of the models can be obtained from mathematical description through interactive generation of its subparts and their consequent assemblage. Commonly pyFormex is used to create 3D models from medical scan images. Predefined operations help for the pre- and post-processing of the finite elements analysis models[53]. A possibility of incorporating a graphic user interface makes the use of pyFormex easier. PyFormex is a free program under GNU license.

3.3. Django

Django is an open source web framework open source, which is written in Python and follows the paradigm known as Model Template View. It was originally developed to manage several news-oriented sites of the World Company of Lawrence, Kansas, and was released to the public under a BSD license in July 2005.

Django's primary goal is to facilitate the creation of complex websites. Django emphasizes re-use, connectivity and extensibility of components, rapid development and the principle of not repeating oneself (DRY, Do not Repeat Yourself). Python is used in all parts of the framework, even in settings, files, and data models.

4. Software development process

Two demonstrator of software tools are developed during the work on this project. The first one is developed in pyFormex and the second one is a web based application developed using of Django. Their functionalities allow the user to load patient-specific data and to enter values for input parameters that are essential for the generation of mitral valve finite element model. Modules of the demonstrators will generate 3D patient-specific mitral valve models, generate an input file for Abaqus and also will allow the user to save or load files with information about all of the input parameters. The development of the demonstrators is following the so called “Waterfall model”. In this model the software development is divided in five stages: Software requirements, Software design, Implementation, Testing, Maintenance (Figure 16).

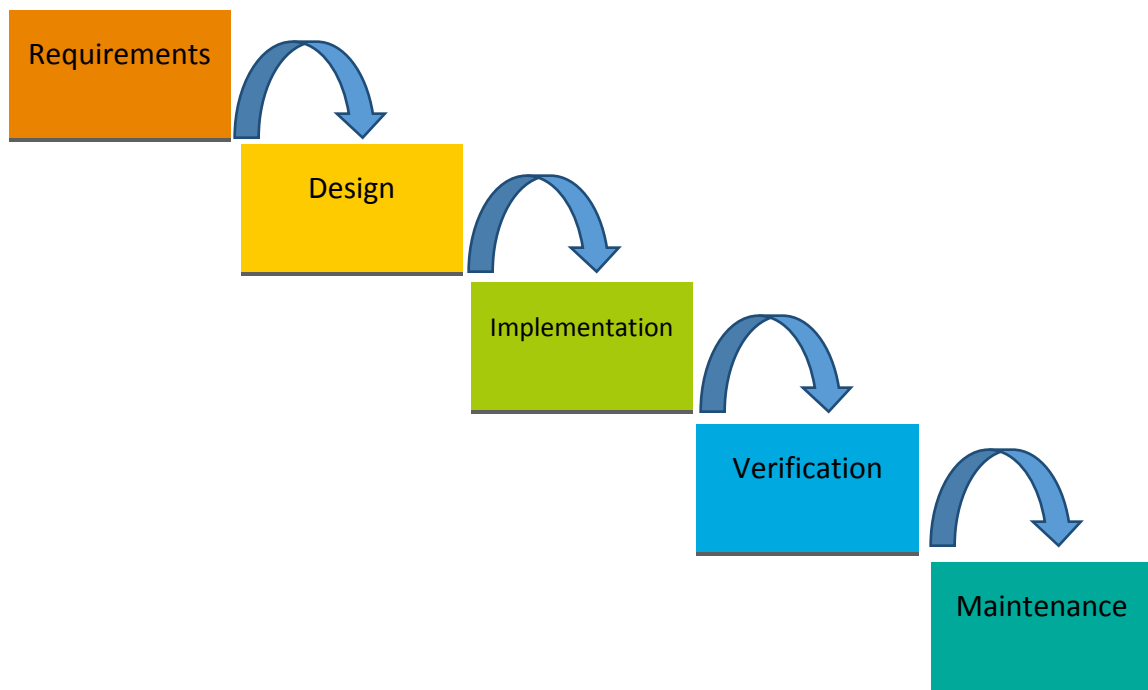


Figure 16 Waterfall model

The last stage of the model, the so called “Maintenance” is out of the scope of this project because it refers to the stage when the software is fully ready and already in use from the users.

All of the stages of the waterfall model are performed for both software demonstrators – the web based application and the pyFormex graphic user interface. Further in the text when not explicitly specified which demonstrator is referred, it is implied that both are being referred.

4.1. Software requirements

Software requirements help the developer in designing quality software that meets the stakeholder's needs. The steps performed are:

- Step1: Stakeholder analysis
- Step2: Requirements specification(functional and non-functional)
- Step3: Use case specification
-

4.1.1. Stakeholders

The first step is to identify the stakeholders. All persons that are involved in the given problem must be considered. In the Table a separate scheme of the stakeholders, their relation to the project, their problems and their expectations is shown.

Table 2 Stakeholder analysis

Stakeholder	Characteristics	Expectations	Potentials and deficiencies	Implications and conclusions for the project
Medical doctors/ Surgeons	Strives for the best treatment of his patient, Interested in user friendly and easy software tools that will help them to plan their surgery more efficient	Submission of patient-specific data, visualization of 3D FEA model, saving and loading of input parameters, request a input file for simulations	Expert in surgery, lacks technical knowledge	Could be conservative, so refusing new technologies, wide range of patient-specific problems makes it difficult to find a common solution
Researchers/ Students	Interested in user friendly and easy software tools that will help them understanding better the mitral valve apparatus	Submission of patient-specific data, visualization of 3D FEA model, saving and loading of input parameters, request a input file for simulations	Different background knowledge	Wide range of research topics makes it difficult to develop a common solution
Developers	Involved in the development of the applications	High quality programming code and good documentation for it	Differences in working processes	Essential for developing and maintains of the applications
Industry	Interested in innovative products at low development costs	Money	Commercialising the product	Knowledge of market potential

The main identified stakeholders are the medical doctors/surgeons, researchers/students, developers and industry. The expectations of the first two stakeholders are very similar in terms of their implications and conclusion for the project. The deficiencies are also similar for the first

three stakeholders. The industry is also present in the table since it should it should always be considered when working on the project, even in the early stages.

4.1.2. Requirements specification

Requirements specification consists of functional, non-functional and environmental requirements. The functional requirement defines the functions of the system or, in other words, what the system is supposed to do. The non-functional requirements define how the system is supposed to be. The environmental requirements - which software should be installed in order the demonstrators to work.

4.1.2.1. Functional requirements

FUN-001: The application shall allow the user to upload patient specific data which further will be used in modeling the FEA mitral valve.

FUN-002: The application shall allow the user to change the chosen parameters of the FEA mitral valve model.

FUN-003: The application shall allow the user to view the preview of the mitral valve FEA model (pyFormex).

FUN-004: The application shall allow the user to view all their requests in a list (Django).

FUN-005: The application shall allow the user to save all of the input parameters that are used to model the FEA mitral valve.

FUN-006: The application shall allow the user to request the input file for further use in Abaqus (pyFormex).

FUN-007: The application shall allow the user to load the input parameters from a file.

4.1.2.2. Non-functional requirements

- Usability requirements

USA-001: The application shall have a user manual.

- Other requirements

OTH-001: The application shall not accept or store any kind of patient information.

4.1.2.3. Environmental requirements

Environmental requirement during deployment of the software:

- System side requirements
 - Operating system : Debian jessy release
 - Web application: python-django 1.6.1
 - Other software: pyFormex 0.9.1
 - Database: SQLite 2.8.17
- Client side requirements
 - Operating system : Linux-debian, Windows XP/7/8
 - Web browser : Mozilla Firefox +24, Google Chrome +30

4.1.3. Use cases

A use case is a description of the steps or activities to be undertaken to carry out a process. The characters or entities that participate in a use case are called actors. In the context of software engineering, a use case is a sequence of interactions that take place between a system and its actors in response to an event that initiates an actor's principal on the system itself. The use case diagrams are used to specify the communication and behavior of a system by interacting with users and/or other systems. In the use cases described below

under Operator the author means that the actor can be any of the listed: medical doctor/surgeon, researcher/student or developer testing the system.

4.1.3.1. Use cases Django application

Use Case ID:	1
Use Case Name:	Log in to the application
Actors:	Operator, System
Description:	To login to the application and to be authenticated
Preconditions:	Installed and working web browser
Postconditions:	
Normal Flow:	<ul style="list-style-type: none"> • The user types his username and his password • The system checks if the username and password is correct : <ul style="list-style-type: none"> ○ If correct, logs the user into the system. ○ If wrong, the user is notified that he typed the wrong username/password.

Use Case ID:	2
Use Case Name:	Log out from the in to the application
Actors:	Operator, System
Description:	To log out from the application
Preconditions:	User is logged into the system
Postconditions:	
Normal Flow:	<ul style="list-style-type: none"> • The user selects to log out from the application • The system logs the user out from the application

Use Case ID:	3
Use Case Name:	Load patient-specific data
Actors:	Operator, System
Description:	To load a patient-specific data from .trc, .csv, .pgf files
Preconditions:	User is logged into the system, User has the files in advance
Postconditions:	
Normal Flow:	<ul style="list-style-type: none"> • The user selects a .trc file containing the geometry of the mitral valve • The user selects a .csv file containing hemodynamic data • The user selects a .pgf file containing generated mitral valve leaflet geometry (mesh's nodes and connectivity) • The system validates that the user selected all the files. • The system stores the patient-specific data • The system assigns an id for the submitted patient-specific data.

Use Case ID:	4
Use Case Name:	Saving all of the parameters in .json file
Actors:	Operator, System
Description:	To request a mitral valve simulation by submitting input parameters needed for the generation of the finite element model of the mitral valve simulation and saving them in .json file which could be used later in pyFormex script.
Preconditions:	User is logged into the system User selected a patient specific data
Postconditions:	
Normal Flow:	<ul style="list-style-type: none"> • The user selects a patient specific data entry. • The user defines the geometry of the mitral valve: <ul style="list-style-type: none"> ○ Thickness of the anterior and posterior leaflets ○ Number of basal and marginal chordae tendinae ○ Relative height of the basal chordae tendinae ○ Cross section of the marginal and basal chordae tendinae ○ Type of the basal and marginal chordae tendinae – truss or connector • The user defines the material properties of the anterior leaflet <ul style="list-style-type: none"> ○ The user can choose to use linear elastic model ○ The user can choose to use hyperelastic model (Holzapfel) • The user defines the material properties of the posterior leaflet <ul style="list-style-type: none"> ○ The user can choose to use linear elastic model ○ The user can choose to use hyperelastic model (Holzapfel) • The user defines the simulation parameters <ul style="list-style-type: none"> ○ Friction coefficient ○ Time step • The system validates that the user typed the correct kind of data • The system saves the parameters in a .json file • The system assigns an id for the submitted request of a mitral valve simulation model • The system collects the following data : <ul style="list-style-type: none"> ○ the .json file ○ patient specific data (.trc, .csv, .pgf)

4.1.3.2. Use cases pyFormex script

Use Case ID:	1
Use Case Name:	Open the application
Actors:	Operator
Description:	Open and run the application
Preconditions:	Installed and working pyFormex 0.9.1 User has the pyFormex script in advance
Postconditions:	
Normal Flow:	<ul style="list-style-type: none"> • The user opens the program – pyFormex • The user selects the script code • The user runs the script

Use Case ID:	2
Use Case Name:	Close the pyFormex script
Actors:	Operator
Description:	To close the pyFormex script and program
Preconditions:	User opened the pyFormex script
Postconditions:	
Normal Flow:	<ul style="list-style-type: none"> • The user selects to close the graphic user interface • The user selects to close the pyFormex program

Use Case ID:	3
Use Case Name:	Load patient-specific data
Actors:	Operator, System
Description:	To load a patient-specific data from .trc, .csv, .pgf files
Preconditions:	User opened and ran the pyFormex script User has the files in advance
Postconditions:	
Normal Flow:	<ul style="list-style-type: none"> • The user selects a .trc file containing the geometry of the mitral valve • The user selects a .csv file containing hemodynamic data • The user selects a .pgf file containing generated mitral valve leaflet geometry(mesh's nodes and connectivity)

Use Case ID:	4
Use Case Name:	Loading of the mitral valve parameters from .json file
Actors:	Operator, System
Description:	Loading the mitral valve parameters from .json file which was previously generated by Django application
Preconditions:	User opened and ran pyFormex script
Postconditions:	
Normal Flow:	<ul style="list-style-type: none"> • The user selects to load the mitral valve parameters • The system allows the user to select the .json file • The system reads the .json file • The system loads the parameters in the graphic user interface in pyFormex • The system puts predefined values for the mitral valve parameters

	<p>if they are not present in the .json file</p> <ul style="list-style-type: none"> • The system warns the user that he needs to choose some of the radio buttons in the material properties tab
--	---

Use Case ID:	5
Use Case Name:	Entering of mitral valve parameters
Actors:	Operator
Description:	To enter all of the mitral valve chosen parameters
Preconditions:	<p>User opened and ran the pyFormex script</p> <p>User loaded patient specific data</p>
Postconditions:	
Normal Flow:	<ul style="list-style-type: none"> • The user defines the geometry of the mitral valve: <ul style="list-style-type: none"> ○ Thickness of the anterior and posterior leaflets ○ Number of basal and marginal chordae tendinae ○ Relative height of the basal chordae tendinae ○ Cross section of the marginal and basal chordae tendinae ○ Type of the basal and marginal chordae tendinae – truss or connector • The user defines the material properties of the anterior leaflet <ul style="list-style-type: none"> ○ The user can choose to use linear elastic model ○ The user can choose to use hyperelastic model (Holzapfel) • The user defines the material properties of the posterior leaflet <ul style="list-style-type: none"> ○ The user can choose to use linear elastic model ○ The user can choose to use hyperelastic model (Holzapfel) • The user defines the simulation parameters <ul style="list-style-type: none"> ○ Friction coefficient ○ Time step • The user defines how many computers will calculate the Abaqus simulation on the cluster • The user chooses with what colors the system should show him the preview of the generated mitral valve FEA model

Use Case ID:	6
Use Case Name:	Saving the mitral valve parameters
Actors:	Operator, System
Description:	Saving the mitral valve parameters
Preconditions:	User opened and ran pyFormex script User defined the mitral valve previously
Postconditions:	
Normal Flow:	<ul style="list-style-type: none"> • The user selects to save the mitral valve parameters • The system give chance to the user to choose where and with what name to save the mitral valve parameters • The system saves the mitral valve parameters • The system loads the parameters in the graphic user interface in pyFormex • The system puts predefined values for the mitral valve parameters if they are not present in the .json file
Alternative Flows:	

Use Case ID:	7
Use Case Name:	Generating the input file for Abaqus simulations
Actors:	Operator, System
Description:	Generating the .inp file which can be further used in order to make simulations in Abaqus
Preconditions:	User opened and ran pyFormex script User defined the parameters of the mitral valve User loaded the patients specific data

Postconditions:	
Normal Flow:	<ul style="list-style-type: none"> • The user selects to generate the .inp file • The system give chance to the user to select where and with what name to save the .inp file • The system warns the user that .request file will be also generated • The system generates and saves the .inp file • The system generates and saves .request file

Use Case ID:	8
Use Case Name:	Preview of the mitral valve generated FEA model
Actors:	Operator, System
Description:	Preview of the mitral valve FEA model
Preconditions:	<p>User opened and ran pyFormex script</p> <p>User defined the parameters of the mitral valve</p> <p>User loaded the patient specific data</p>
Postconditions:	
Normal Flow:	<ul style="list-style-type: none"> • The user selects to view the mitral valve FEA model • The system generates the FEA mitral valve model • The system shows the user a 3D dynamic model of the mitral valve in predefined colors
Alternative Flows:	

4.2. Software design

Software design is the last activity to be performed before the development of the software. In the software engineering today there is a standard way to visualize the design of the system using Unified Modeling Language (UML). The UML offers a way to visualize system's blueprints in a diagram. Appropriate use of UML notation is essential part in creating a complete and meaningful model. For more information about the basic notations in UML please follow the link: http://www.tutorialspoint.com/uml/uml_basic_notations.htm. In this section the system architecture and activity diagrams will be defined with the help of UML.

Software design is based on previously developed pyFormex script for generation of input file for Abaqus and generation of mitral valve finite element analysis 3D model. The chosen mitral valve parameters to be implemented will be mentioned.

4.2.1. System architecture

The system architecture is a conceptual model describing the structure and the behavior of the system. In the Figure 17. a UML diagram of the system architecture is shown.

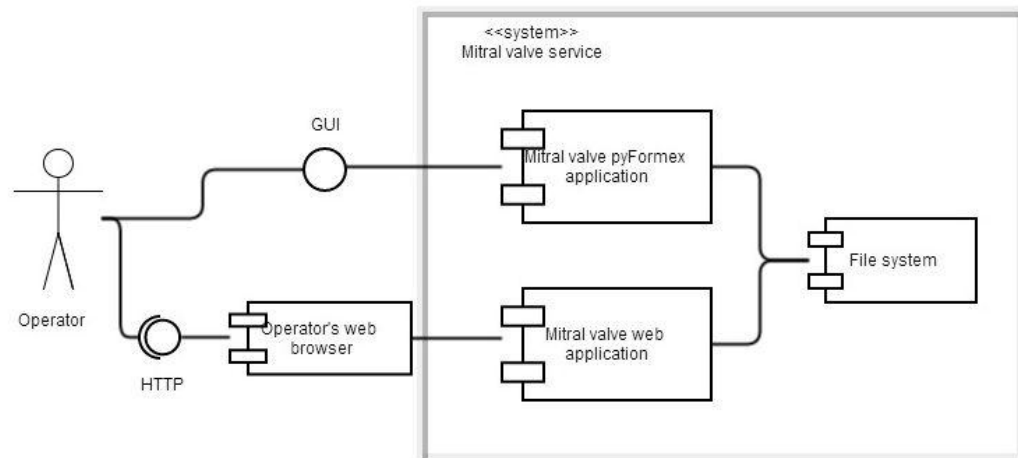


Figure 17 System architecture

The system shall have three components: Mitral valve pyFormex script, Mitral valve web application and File system. The developed components shall be demonstrators of what is possible to be done, for easy differentiating of the components they will be called Mitral valve pyFormex script, Mitral valve web/Django application. Mitral valve pyFormex script shall be responsible for generation of the 3D finite element analysis (FEA) model of the mitral valve as well as for generation of input file for Abaqus. Other functionality of this application shall be saving and loading patient input data. In the mitral valve web application, the operator shall have possibility to enter and save all of the input parameters necessary for the generation of the mitral valve model as well as the patient-specific data. In the file system the Mitral valve web application shall keep the three files (TRC, PGF,CSV) that the operator has previously uploaded and also a JSON file with all of the input parameters of the mitral valve. After submitting a request from the operator for input file for Abaqus or saving the input parameters, the Mitral valve pyFormex script shall save INP, REQUEST and JSON file in the file system. The REQUEST file is necessary in case the operator wants to run a simulation on the cluster. An operator in such UML diagram could be any of the predefined actors: medical doctor/surgeon, researcher/student or developer which testing the system. The operator shall interact with the Mitral valve pyFormex via graphic user interface (GUI) and with Mitral valve web application via web browser and HTTP protocol.

4.2.2. Activity diagrams

Activity diagrams represent activities that occur during the process flows in the system. Explanation about the notations used in this section can be found at <http://sourcemaking.com/uml/modeling-business-systems/external-view/activity-diagrams>.

The diagrams are divided by activity diagrams for Mitral valve pyFormex script and activity diagrams for Mitral valve web application. In the diagrams below, by Operator the author means any of the predefined actors (medical doctor/surgeon, researcher/student, developer).

System is a summarizing term (depending in which section is used can mean Mitral valve pyFormex script or Mitral valve web application).

4.2.2.1. Activity diagrams Mitral valve pyFormex script

- Load a JSON file

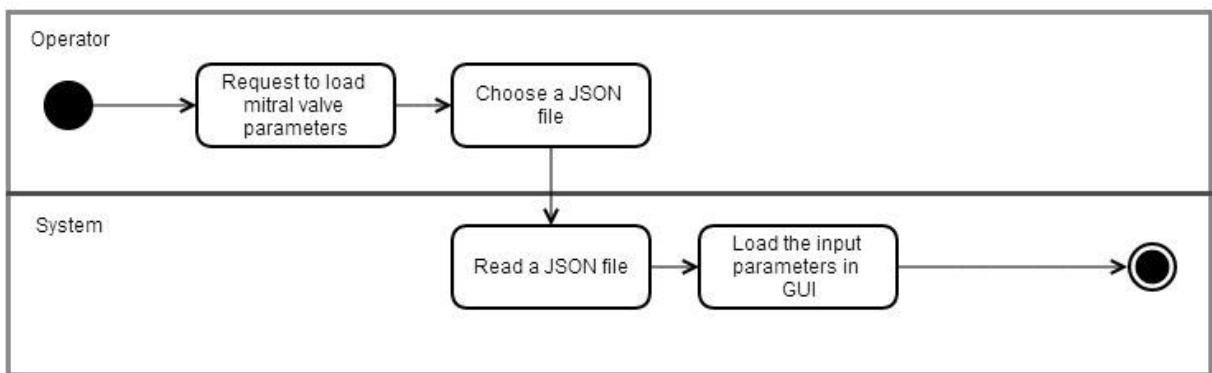


Figure 18 Activity diagram pyFormex: Load a JSON file

In order to load the JSON file, first the operator shall submit request for it and then choose a JSON file from the file system. Then the system shall read the JSON file and load the mitral valve parameters in the graphic user interface dialog.

- Generate a JSON file

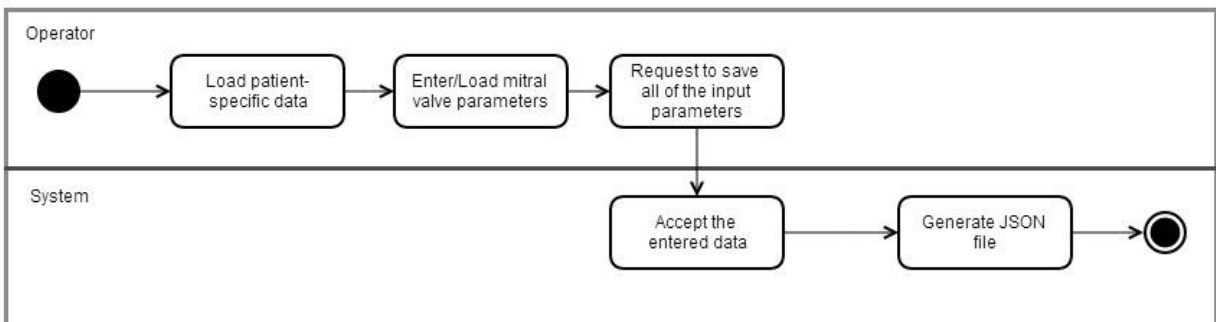


Figure 19 Activity diagram pyFormex: Generate a JSON file

If the operator loads patient-specific data and mitral valve parameters and request to save them, the system shall accept the data and generate and store it in a JSON file.

- Preview of mitral valve FEA model

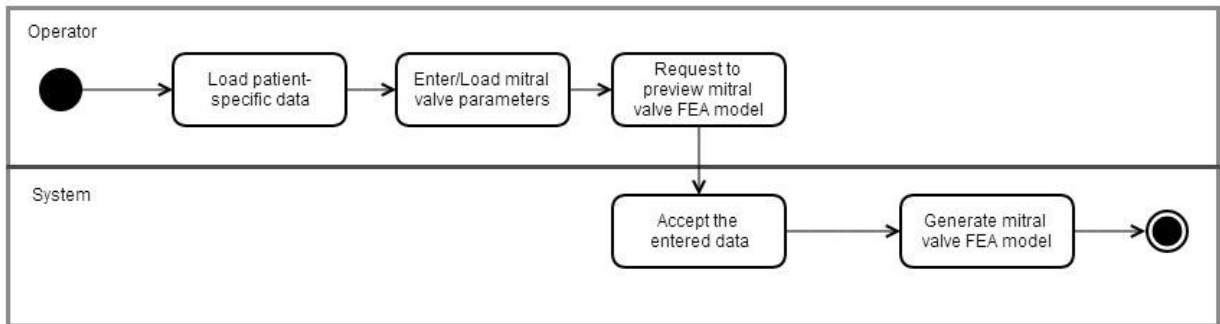


Figure 20 Activity diagram pyFormex: Preview of mitral valve FEA model

If the user loads the patient specific data files and mitral valve parameters and then submits a request for a preview of the mitral valve FEA model, the system shall accept the data and generate a 3D FEA model. Mitral valve parameters here refer not only to the parameters involved in the generation of the geometry of the mitral valve but also to the colors in which the operator wants the model to be.

- Generate the input file for Abaqus

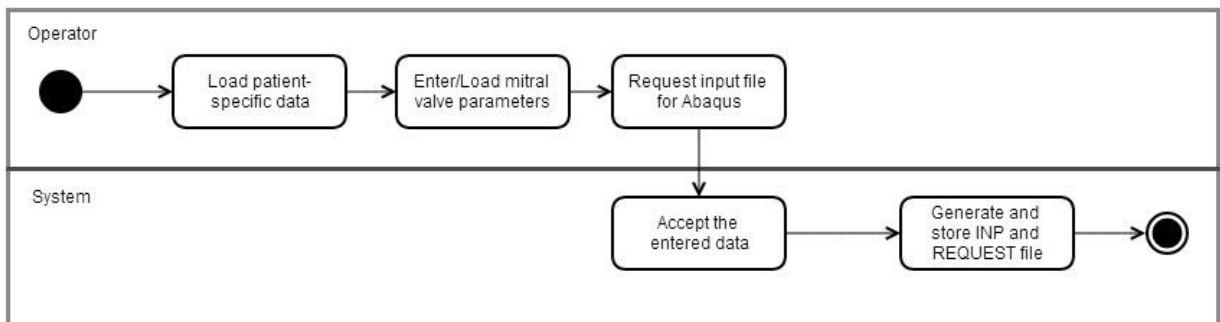


Figure 21 Activity diagram pyFormex: Generate a input file for Abaqus

Once the user submits the patient specific data files and the mitral valve parameters and request input file for Abaqus, the system shall accept the data. Moreover the system shall generate and store INP and REQUEST files. The REQUEST file shall be automatically generated in order for the operator to have the possibility to submit the simulation on the cluster immediately.

4.2.2.2. Activity diagram for mitral valve web application

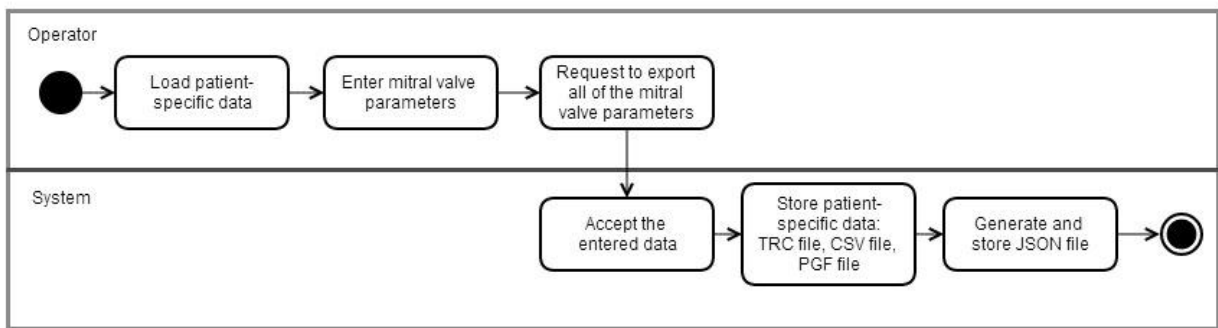


Figure 22 Activity diagram web application: Generate a JSON file

In mitral valve web application the operator shall have possibility to load patients-specific data files as well as to enter mitral valve parameters. Moreover the operator shall be able to request export of all of the mitral valve parameters. Then the system shall accept the entered data, store the patient-specific data files (TRC, CSV and PGF files) and generate and store the JSON file.

4.2.3. Mitral valve parameters

The design of the software is based on the previously developed pyFormex script which has functionalities to generate an input file for Abaqus as well as to generate 3D finite element model of the mitral valve. Taking into account the developed software and the literature review, some essential parameters of the mitral valve were chosen to be implemented. The chosen

parameters are: thickness of the anterior and posterior leaflets, number of basal and marginal chordae tendinae, relative height of the basal chordae, cross-section of the basal and marginal chordae, type of the chordae tendinae and friction coefficient. With the help of these parameters, various pathology states as well as reconstruction techniques can be simulated.

4.3. Implementation

After defining the software requirements and design, the implementation phase was completed. The demonstrators were done on a personal computer under Linux Jessie operating system. A pyFormex 0.9.1, Django 1.6.1 and SQLite 2.8.17 were used as well as web browsers Google Chrome 30 and Mozilla Firefox 24.

4.3.1. Mitral valve pyFormex script

Mitral valve pyFormex demonstrator was developed under pyFormex program version 0.9.1. The mitral valve pyFormex script is recorded in a Python file. A previously developed, by other student, code was used as well. Extension of the code was done by implementing a graphic user interface (GUI). New material models for the anterior and posterior leaflets were included. The full documented script can be found on the CD that comes with this paper work or at the web address: <https://www.dropbox.com/s/1w44v6fla6oa77n/MitralValvepyF.py>.

Considering the software requirements and the software design, a tabbed GUI is developed. Figure 23 a screenshot from developed GUI can be seen.

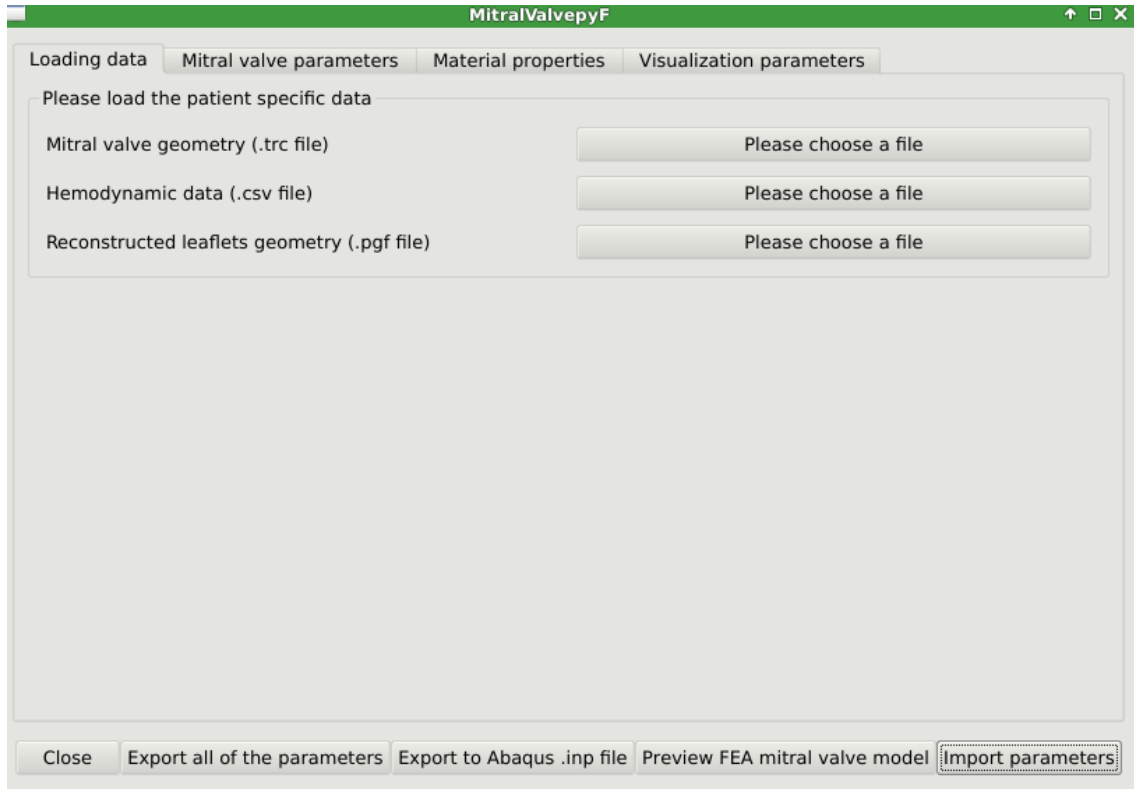


Figure 23 GUI mitral valve pyFormex script

The structure of the graphic user interface can be seen in the figure below.

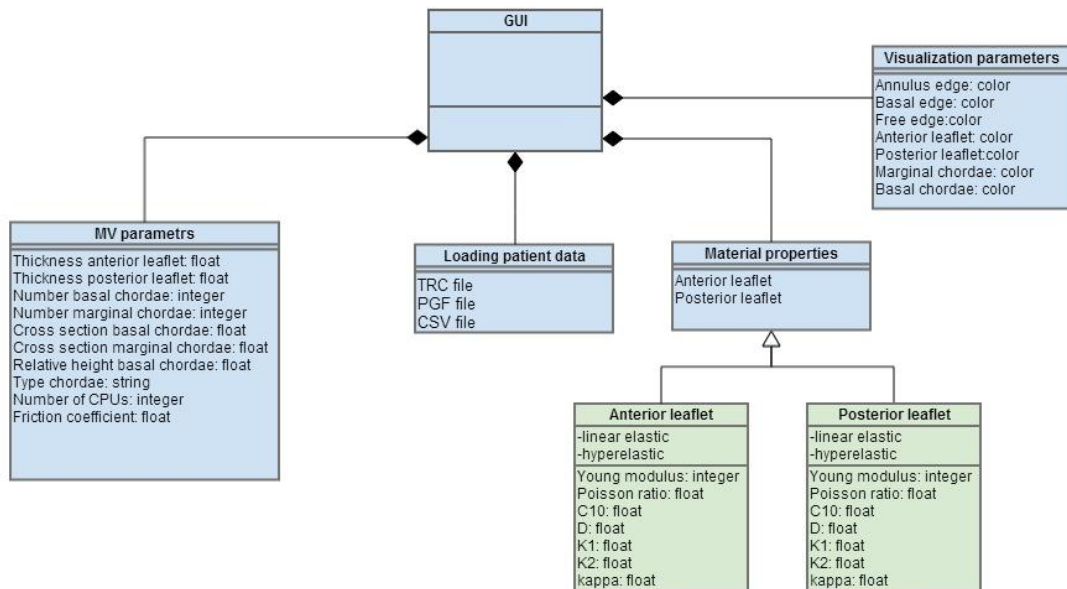


Figure 24 Structure of the GUI in pyFormex

The shown scheme represents the structure of the input dialog (note that it does not incorporate the standards of the UML language). The four tabs are shown in blue and the two sub-tabs - in green. Every tab group contains group elements in which different types of parameters are defined.

The first tab group is related to the loading of the patient-specific data files. Three button-like file fields are defined. Every button opens pop-up window in which the user can choose a file. The script shows to the user only the files with the same extension as the file that should be loaded. For example, if the user chooses to load a TRC file, the script will show them only the TRC files in the directory and will not give them possibility to change the extension of the shown files.

The second tab group contains different types of fields of all of the mitral valve parameters that can be changed by the user. They are structured in three groups: leaflets parameters, chordae tendinae parameters and simulation parameters (Figure 25).

The screenshot shows a GUI window with three main sections, each with a title bar in a light blue tab:

- Leaflets:** Contains two text input fields. The first is labeled "Thickness of the anterior leaflets (mm)" with a value of 0.25. The second is labeled "Thickness of the posterior leaflets (mm)" with a value of 0.25.
- Chordae tendineae:** Contains several controls:
 - "Number of the basal chordae": A text input field with the value 12 and a corresponding slider control.
 - "Number of the marginal chordae": A text input field with the value 28 and a corresponding slider control.
 - "Relative height of basal chordae": A text input field with the value 0.5.
 - "Cross section marginal chordae(mm)": A text input field with the value 0.4.
 - "Cross section basal chordae(mm)": A text input field with the value 0.23.
 - "Type of the chordae tendinae": A radio button group with "truss" selected (indicated by a black dot) and "connectors" unselected (indicated by a grey dot).
- Simulation parameters:** Contains two controls:
 - "Number of CPUs to be used by the cluster": A radio button group with "4" selected (black dot), "6" unselected (grey dot), and "8" unselected (grey dot).
 - "Friction coefficient": A text input field with the value 0.05. A yellow callout box points to this field with the text "Like 'select', this allows sel".

Figure 25 GUI pyFormex : Mitral valve parameters tab

Numerical fields are used for most of the values. The numbers of basal and marginal chordae tendinae are fields of type slider in order to ensure that the user will enter integers. The chordae type and the number of CPUs that will be used in the simulation on the cluster are radio buttons with predefined values.

The third tab group contains input information about the material properties of the leaflets. The group has two sub-tab groups about the anterior and posterior leaflets. Every sub-tab group contains two check fields: one for the linear elastic and one for the hyperelastic Holzapfel material model. Depending on the check field, various numerical parameters can be entered (Figure 26).

The image shows a software interface for configuring material properties. It has two tabs: 'Anterior leaflet' and 'Posterior leaflet'. The 'Anterior leaflet' tab is active. There are two checkboxes: 'Linear elastic' (unchecked) and 'Hyperelastic anisotropic' (checked). Under 'Linear elastic', there is a 'Parameters' section with two input fields: 'Young modulus(GPa)' with the value 117 and 'Poisson ratio' with the value 0.355. Under 'Hyperelastic anisotropic', there is a 'Holzapfel parameters' section with five input fields: 'C10' (0.0502), 'D' (0.003), 'K1' (3.0207), 'K2' (144.4848), and 'kappa' (0.0534).

Figure 26 GUI pyFormex : Posterior leaflet material properties tab

The fourth tab group contains button-like fields with values corresponding to the colors of different parameters that are used for preview of the mitral valve finite element model (Figure 27). Once clicked on the field, a palette window pops-up.

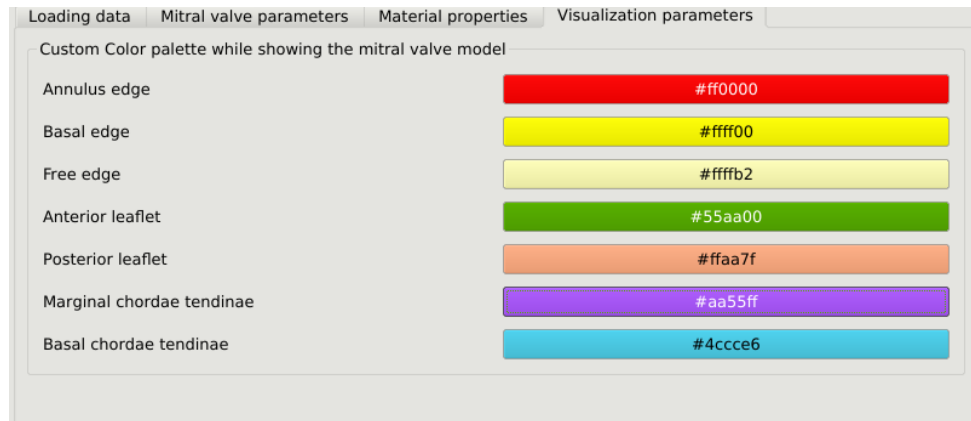


Figure 27 GUI pyFormex : Visualization parameters tab

Predefined values for each parameter are integrated in the graphic user interface. For the hyperelastic Holzapfel model, the values are taken from the paper of Wang et al [38].

The main function that creates the graphic user interface and its functionalities is createDialog(). In Figure 28 a diagram of the functionality of the script provided by the buttons is shown.

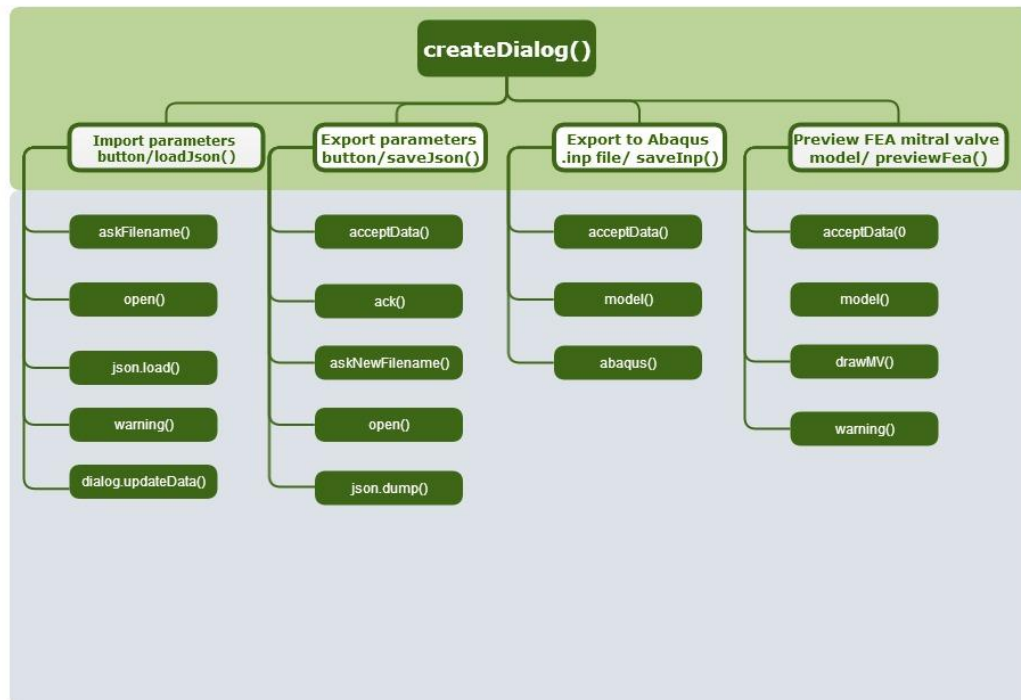


Figure 28 pyFormex: Structure of the button functions

In the `createDialog()` function, four buttons are defined. Every button is represented by a function.

The first button is “Import parameters” and the function that correspond to it in the script is `loadJson()`. Once the user clicks on this button, `askFilename()` function is called and thus they can choose a JSON file which contains mitral valve parameters. The user needs to have the file in advance. Such a file can be generated by the `pyFormex` script, Django application or simply written by the users themselves. Once the file is chosen, then the script opens it and loads the data. In order for the data from the JSON file to load in the GUI, a `dialog.updateData()` function is used. Also a warning message is created here to remind the user that he needs to activate one of the checkboxes in the Material properties tab.

The second button in the GUI is “Export parameters” which is responsible for saving all of the entered mitral valve parameters in JSON file. This is possible, because of the function `saveJson()`. This function contains a function for accepting the entered data – `acceptData()`, a function which makes dialog window in which the user is asked whether to save export the parameters – `ack()`. Moreover there is a function `askNewFilename()` which asks where and with what name the file should be saved. The next step is to open the file and write the mitral valve parameters in JSON format.

The third button is “Export to Abaqus .inp file” and the function that is behind it is `saveInp()`. In this function there are three main functions: `acceptData()`, `model()`, `abaqus()`. The first reads and accepts the input data from the dialog and makes it available for the other functions. The functionality of the functions `model()` and `abaqus()` were previously developed by other student and they are integrated in this script. The function is responsible for generating the finite element model. It takes the PGF file and loads the valve geometry. The PGF file contains information about the mesh’s nodes and connectivity of the anterior and posterior leaflets. Then the function uses the CSV file to load the patient-specific transvalvular pressure and the TRC file to reconstruct the whole mitral valve geometry as well as the annular and papillary muscle motion. Then `model()` generates the basal and marginal chordae tendinae taking into account the entered parameters from the user. The last function used in `saveInp()` function is

abaqus() which is responsible for the generation and save of the input file for Abaqus. The function generates in a correct format the geometry of the mitral valve, elements and nodes set, the material properties of the leaflets and chordae tendinae, then makes sections and assigns them. The function incorporates the pressure loads, annulus and papillary muscles movement and defines the ODB output and the steps. Changes in the functionality of the function are made in the initial, previously written script. An option is added in the functionality of the aboqus() function which allows the user to choose from linear elastic and hyperelastic Holzapfel material model for the leaflets of the mitral valve. With the help of if-else statements and manual operations the right format of the Holzapfel model was generated in the input file. PyFormex is under development and still doesn't support automatically implementation of Holzapfel material model. If the user chooses hyperelastic material properties for some of the leaflets, the script opens the INP file after it is generated and searches for the place in it where the information for the material properties is written. After finding the proper lines, the script adds new lines with the parameters necessary for the Holzapfel model. The material properties of the chordae tendinae are hard-coded in the script. The material data is taken from uniaxial tensile tests performed by Kunzelman[54]

The next button in GUI is "Preview FEA mitral valve model". The function behind it is previewFea() and it's responsible for visualization of the 3D finite element model of the mitral valve. The first step that is performed by the function is to accept the input data, then to execute the function model() which was explained previously. Once the function model() generates the FEA model, a drawMV() function is called and the patient-specific mitral valve model is drawn in the pyFormex window. The colors utilized are the one that the user picked in the visualization tab. A warning message appears that the model can be seen behind the GUI.

These were the main functions used in the pyFormex script. There are more that were written but their functionality is just supportive and they are not explained here.

4.3.2. Django

The python web framework Django is designed to make common web-development tasks fast and easy to complete. The Django application is developed under Linux Debian Jessie release. Web browsers Mozilla Firefox 17.0.9 and Google Chrome 35 are used for visualization of the application. The used database is SQLite 3.8.4. The final version of Python was also installed (2.7.5). After choosing and installing the database and the web browsers, a Django 1.6.1 was installed. The full documented application can be found on the CD that comes with this document or at the web address:

<https://www.dropbox.com/s/m1w3geeu197xyix/MitralValveDj.zip>

Once all of the software is up and running a Django project is made. In Django, the difference between project and application is that a project is a collection of applications and configurations for a given Web site, while an application is the Web application that executes certain actions and interacts with the users. Once the project was made, an application was created. The Django framework automatically generates the initial files that are needed to have a working application.

The application follows the standard Django architecture which can be seen in Figure 29.

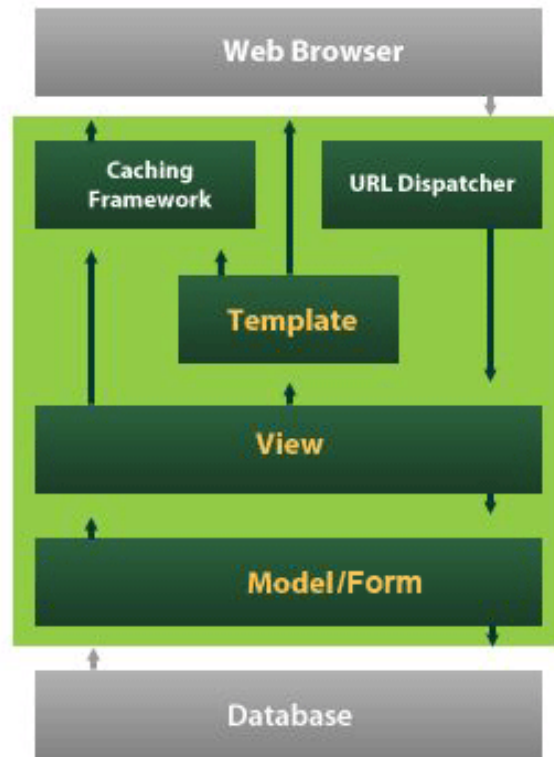


Figure 29 Standard Django architecture[55]

The URL dispatcher (urls.py) maps the requested URL to view a function and consequently calls it. The view function (views.py) performs the requested action. The model (models.py) defines the data and interacts with it. The data is stored in database. After performing the requested task, the view function takes the result and returns it through the template to the web browser.

Following these architecture and requirements, a Django application was developed. The complete file tree of the Django project can be seen at the next figure.

```
MV/
manage.py
MV_april/
  __init__.py
  settings.py
  urls.py
  wsgi.py
mitral_valve/
  static/
    mitral_valve/
      css/
        basic_style.css
      js/
        jquery-1.9.1.js
        tab-strip.js
  templates/
    admin/
      base_site.html
    mitral_valve/
      base.html
      login.html
      pt_data_list.html
      pt_data_submit.html

  __init__.py
  admin.py
  forms.py
  models.py
  tests.py
  urls.py
  views.py
media/
  1/
  2/
  ...
```

Figure 30 File tree of Django project

The outer MV/ root directory is just a container for the project. The python file manage.py is a command-line utility that lets the user to interact with the Django project. The inner directory MV_april/ contains the main python files necessary for the project. The MV_april/__init__.py is an empty file that tells Python that this folder should be considered as a Python package. The MV_april/settings.py file consists of the settings of the Django project. MV_april/urls.py contains the URL declarations for the Django project while MV_april/wsgi.py contains an entry-point for WSGI-compatible web servers to serve the project

The directory mitral_valve/ is the main directory of the application and consists of files and two other folders. In the directory mitral_valve/templates, files regarding the template of the

Django application are stored. In the folder `mitral_valve/static` the CSS and JavaScript files are stored. The directory `media` consists of the patient-specific entries that the user submits, which are arranged by dependence on the number of such submissions. Each folder of these consists of the submitted TRC, CSV and PGF files and the generated JSON file. As shown above, the file system of mitral valve Django application is wide and in this section only essential files and some of their content will be discussed.

In order to have an application consistent with the requirements and the used technology a few steps were performed:

- A model was defined
- A form was created
- URLs scheme was made
- Views were written
- A template was designed

Model:

A model was defined in `mitral_valve/model.py` and it contains the fields in which the user can load the patient specific data: TRC, PGF and CSV files as well as a function which deletes user entries of patient-specific data. Once the model was done, synchronization with the database was performed and a database table was automatically created. The database diagram at Figure 31 shows the organization of the information in the project. There is a user which has id, username and password that are kept in the database. The database keeps record also of the patient data that is submitted by the user.

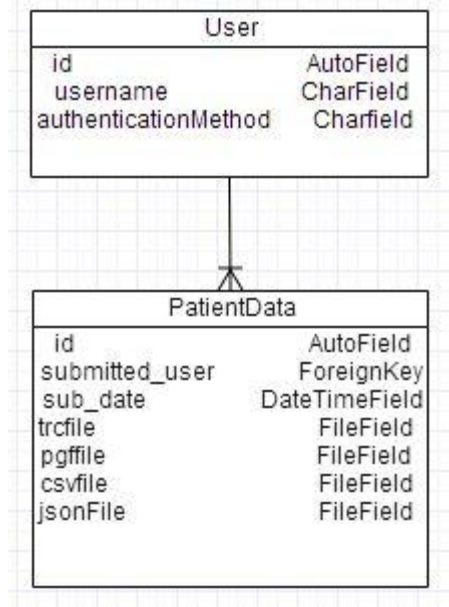


Figure 31 Database diagram Django

Form:

An HTML form was created in `mitral_valve/templates/mitral_valve/ pt_data_submit.html`. In `mitral_valve/forms.py` a class `PtDataForm` was made in which verification of the input parameters for the mitral valve were implemented. In the `ptDataForm()` also a validation for the extensions of the submitted files is done by the help of the functions `clean_trcfile()`, `clean_pgffile()` and `cleancsvfile()`. This class is also responsible for storing the TRC, PGF and CSV files in the `mitral_valve/media` directory. A generation of the JSON file, which contains all of the input parameters, is also done there.

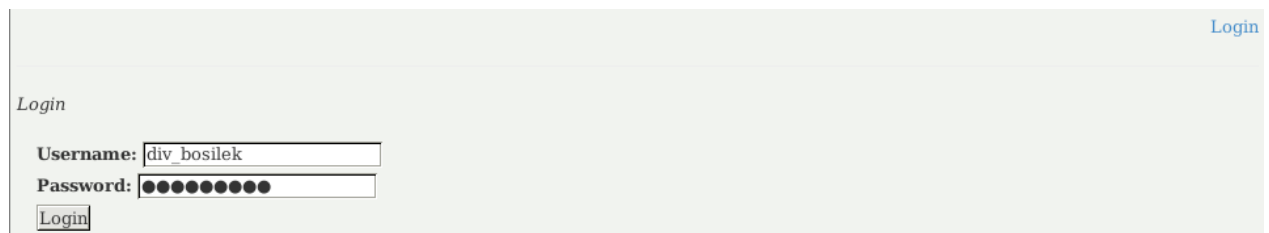
URLs:

Scheme of the URLs was made in the `mitral_valve/urls.py`. A pattern of URLs was defined and once the user requests a page, Django goes through each pattern in order to find the first one that matches the requested URL. If there is a match, Django calls the given view that is defined in `mitral_valve/views.py`. There are defined views for the login and logout of the user, view for the submission of the mitral valve parameters and also a view which shows to the user a list with all of the submitted patient data entries sorted by id.

Template:

In the directory `mitral_valve/templates/mitral_valve/` four different HTML files were made. All of the files contain HTML code and also a Django template language. The `base.html` file consists of the base “skeleton” template in which all of the common elements of the web site are written. The child templates can override the `base.html`. There are three child templates that were made for the submission of the patient data, for the list with the submitted patient data and also template for the login of the user. Styles of the pages were imported from Bootstrap[56] . In the `base.html`, JavaScript files are also imported from the same source.

The first from the three templates is `login.html` which is responsible for the template of the page with the login form (Figure 32).



The image shows a web browser window displaying a login page. In the top right corner, there is a blue link labeled "Login". Below this, the word "Login" is written in a smaller, italicized font. The main form contains two input fields: "Username:" with the text "div_bosilek" entered, and "Password:" with ten black dots representing a masked password. Below these fields is a button labeled "Login".

Figure 32 Django login template

The second one `pt_data_list.html` contains a HTML code representing a simple table in which the user can see list of the submitted patient data by id. Except the id the user can see the date in which he submitted the data and can open any of the patient-specific data files as well as the generated JSON file. The user can also choose to delete some of the entries in the table (Figure 33).

Patient data

[Add new patient data](#)

Table

ID	Submitted date	Patient specific files	Mitral valve parameters	Actions
1	June 4, 2014, 5:23 p.m.	- TRC file - PGF file - CSV file	- JSON file	delete

Figure 33 Django patient data list template

The third child template file is `pt_data_submit.html` which is responsible for the submission of the mitral valve parameters. A POST method is used in the HTML form. The template is designed to be user-friendly and has four tabs which contain different data (Figure 34). The JavaScript code was implemented in order provide interactions between the tabs.

Patient data > [Submit new patient data](#)

Loading data

[Mitral valve parameters](#)[Material properties](#)[Simulation parameters](#)

Patient specific data

Mitral valve geometry (.trc file) No file chosen**Hemodynamic data (.csv file)** No file chosen**Reconstructed leaflets geometry (.pgf file)** No file chosen

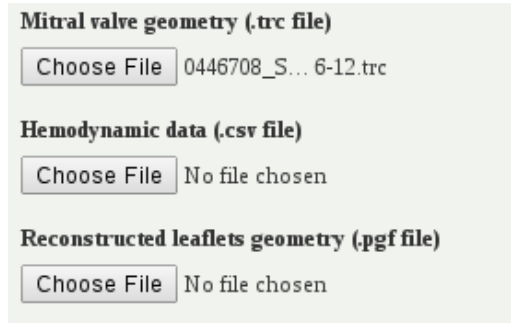
Figure 34 Django submit patient data template

The defined tabs are:

- Loading data tab – enables the user to choose the patients specific files. The user has possibility to load TRC file from which, later, the pyFormex application can extract the geometry of the mitral valve. A PGF file that contains information for the mesh's nodes and connectivity of the anterior and posterior leaflets. The CSV file contains the patient specific transvalvular pressure. The fields are defined as type: file with a

specific name and id (Figure 35). If the user enters an invalid file a message will appear.

-



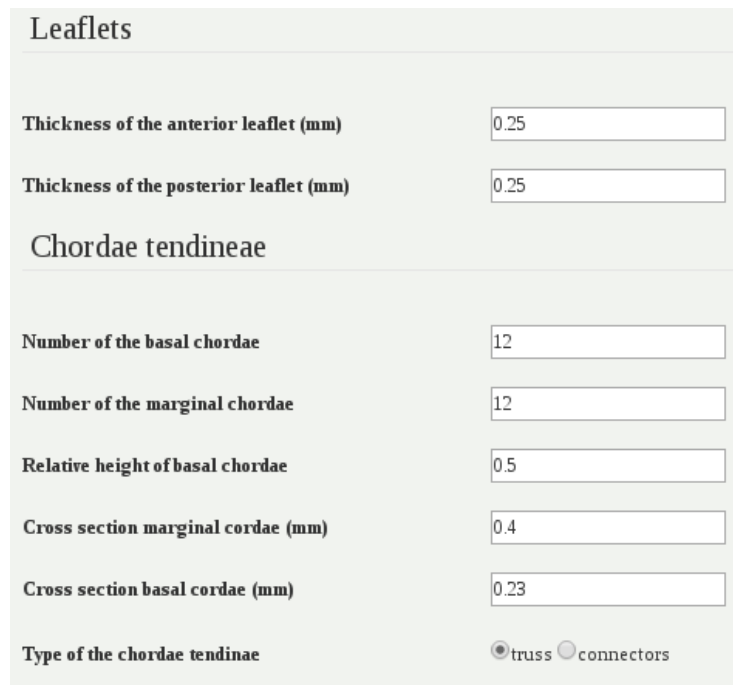
Mitral valve geometry (.trc file)
Choose File 0446708_S... 6-12.trc

Hemodynamic data (.csv file)
Choose File No file chosen

Reconstructed leaflets geometry (.pgf file)
Choose File No file chosen

Figure 35 Django loading data tab

- Mitral valve parameters – text fields are defined in order to collect data for the thickness of the anterior and posterior leaflets, number of basal and marginal chordae, relative height of the basal chordae tendinae, cross section of the basal and marginal chordae and also the type of the chordae tendinae for which a radio button has been implemented (Figure 36).



Leaflets

Thickness of the anterior leaflet (mm)

Thickness of the posterior leaflet (mm)

Chordae tendineae

Number of the basal chordae

Number of the marginal chordae

Relative height of basal chordae

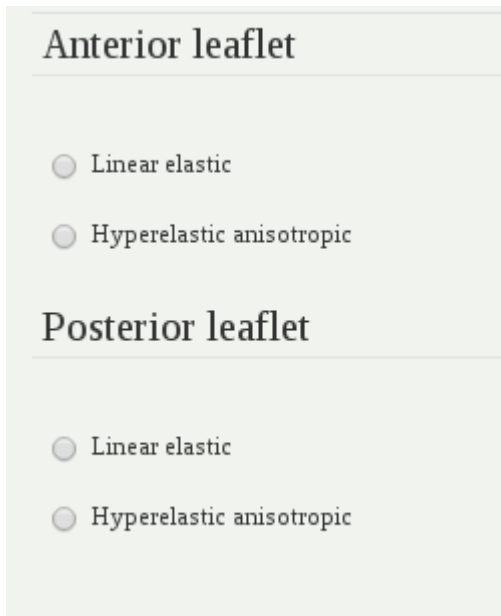
Cross section marginal cordae (mm)

Cross section basal cordae (mm)

Type of the chordae tendinae truss connectors

Figure 36 Django mitral valve parameters tab

- Material properties – a fieldset (an HTML tool) for the anterior and posterior leaflets were written. Every fieldset contains a radio button in order for the user to be able to choose from linear elastic and hyperelastic material models. Text fields were made for the parameters depending on the different material models (Figure 37). A JavaScript code was implemented in order to show and hide the text fields.



The image shows a web form with two sections. The first section is titled "Anterior leaflet" and contains two radio buttons: "Linear elastic" and "Hyperelastic anisotropic". The second section is titled "Posterior leaflet" and also contains two radio buttons: "Linear elastic" and "Hyperelastic anisotropic".

Figure 37 Django material properties tab

- Simulation parameters – a text field for the friction coefficient and radio button for the number of used CPUs were made (Figure 38).



The image shows a web form with two rows. The first row has a label "Friction coefficient" and a text input field containing the value "0.05". The second row has a label "Number of CPUs to be used in the cluster" and three radio buttons labeled "4", "6", and "8".

Figure 38 Django simulation tab

Buttons also were defined by type submission in order to export all of the submitted parameters.

4.4. Verification

In this phase a testing is performed as well as traceability matrix is done.

4.4.1. Testing

The testing is performed for both of the applications. Below tables with all of the performed tests can be seen.

Test Case ID:	TS0001
Test Case Name:	Load patient specific-data
Preconditions:	Operator has run the script
Steps:	Operator loads and runs the script in pyFormex Operator goes to "Loading data" tab Operator chooses TRC, CSV and PGF files
Expected result:	A pop-up window with possibility of choosing a files shows There is option to choose specific file with proper extension The system shows the name of the chosen file in the dialog window
Actual result:	A pop-up window with possibility of choosing files shows There is option to choose specific file with proper extension The system shows the name of the chosen file in the dialog window
Test result:	Pass

Test Case ID:	TS0002
Test Case Name:	Preview of mitral valve FEA model
Preconditions:	Operator has run the script
Steps:	Operator chooses TRC, CSV and PGF files Operator enters a mitral valve parameters Operator chooses colors for the preview Operator clicks on the "Preview FEA mitral valve model" button
Expected result:	Generated 3D dynamic model of the mitral valve appears in the view window in pyFormex Warning message pops-up The model is in the right colors The model corresponds to the input parameters
Actual result:	Generated 3D dynamic model of the mitral valve appears in the view window in pyFormex Warning message pops-up The model is in the right colors The model corresponds to the input parameters
Test result:	Pass

Test Case ID:	TS0003
Test Case Name:	Export all of the parameters
Preconditions:	Operator has run the script
Steps:	Operator chooses TRC, CSV and PGF files Operator enters mitral valve parameters Operator clicks on the "Export all of the parameters" button
Expected result:	Authorization message pops-up File system window appears JSON file with predefined set name is generated
Actual result:	Authorization message pops-up File system window appears JSON file with predefined set name is generated

Test result:	Pass
---------------------	-------------

Test Case ID:	TS0004
Test Case Name:	Import parameters
Preconditions:	Operator has run the script Operator has in advance the JSON file
Steps:	Operator clicks on the “Import parameters” button Operator chooses the JSON file
Expected result:	File system window appears Warning message pops-up The JSON file is loaded and all of the parameters from it appears in GUI
Actual result:	File system window appears Warning message pops-up The JSON file is loaded and all of the parameters from it appears in GUI
Test result:	Pass

Test Case ID:	TS0005
Test Case Name:	Export to Abaqus .inp file
Preconditions:	Operator has run the script
Steps:	Operator chooses TRC, CSV and PGF files Operator enters a mitral valve parameters Operator clicks on the “Export to Abaqus .inp file” button Operator chooses the name for the INP file
Expected result:	File system window appears Warning message pops-up INP file is generated REQUEST file is generated
Actual result:	File system window appears Warning message pops-up INP file is generated REQUEST file is generated
Test result:	Pass

Test Case ID:	TS0006
Test Case Name:	Submit patient-specific data
Preconditions:	Operator's account exists and is active
Steps:	Operator logs into the mitral valve web application Operator clicks on the "Add new patient data" button Operator enters values for the input parameters and uploads patient-specific data files Operator clicks on the "Export all of the parameters" button
Expected result:	JSON file is generated If some of the values is not in required range, the system shows a notification If some of the uploaded files is not with correct extension, the system shows a notification
Actual result:	JSON file is generated If some of the values is not in required range, the system shows a notification If some of the uploaded file is not with correct extension, the system shows a notification
Test result:	Pass

Test Case ID:	TS0007
Test Case Name:	View submitted data
Preconditions:	Operator's account exists and is active Operator submits at least one patient-specific entry
Steps:	Operator logs into the mitral valve web application Operator sees their entries Operator opens the uploaded files Operator opens the generated JSON file Operator deletes entry
Expected result:	List with all of the submitted entries sorted by ID is shown System opens in the browser any file from the list if clicked on it. System deletes the entry if this option is selected

Actual result:	List with all of the submitted entries sorted by ID is shown System opens in the browser any file from the list if clicked on it. System deletes the entry if this option is selected
Test result:	Pass

4.4.2. Traceability matrix

The traceability matrix links the functional requirements of the system throughout the validation process. The purpose of the matrix is to ensure that all of the requirements defined for the system are tested.

Table 3 Traceability matrix

	Functional Requirements	FUN-001	FUN-002	FUN-003	FUN-004	FUN-005	FUN-006	FUN-007
Tests								
TS0001		X						
TS0002			X	X				
TS0003						X		
TS0004								X
TS0005							X	
TS0006		X	X			X		
TS0007					X			

4.5. Maintenance

The maintenance phase refers to the stage when the software is complete, fully integrated and in use. The developed software in this thesis represents a start for the establishment of

framework for modeling of mitral valve FEA models and is not complete and therefore, not yet in commercial use. Therefore the maintenance phase is out of the scope of this project.

5. Conclusions, limitations and future perspectives

The objectives of the study were to develop a graphic user interface of existing code under pyFormex and to make a web based application with the Django framework. The main goals were to create user friendly demo applications which allow every user to generate a patient—specific 3D finite element analysis model of a mitral valve and to give easy way of making an input Abaqus file. A web based application under Django framework was developed which allows the user to save patient-specific data files and generate a file with all of the input parameters of the mitral valve. The second application is developed under pyFormex and allows the user to load input parameters from a file previously generated by the one of the two applications. This application is also responsible for the generation of the 3D finite element analysis model of the mitral valve as well as for the generation of input file for Abaqus.

One of the first limitations of the study to be mentioned is that the two applications work independently. The web based application is useful because can be reached from every computer that has internet on it, but doesn't have the functionality that the other application provides. The pyFormex application needs to be run under Linux and the user cannot work on it via internet easily. In the future the two applications can be run on a server and automatically coupled. This will allow the user to upload all of the patient-specific data files, enter the input parameters for the mitral valve and after submitting them to get the generated input file for Abaqus and images from the generated 3D model in the same application. All that can be done through the web application.

Verification of the input parameters and extension of the file is made only in the Django application. This is a limitation of the pyFormex graphic user interface that could be improved in the future. If both of the applications were coupled together, there would be no need for this because the pyFormex would use the valid parameters entered in the Django application.

In the future, with improvements in the extracting methods for patient-specific data, a functionality that generates the TRC, PGF and CSV files can be added.

There is a future possibility for the application to be used also for investigation of the effect of different repair techniques.

References

- [1] "Tooloop." [Online]. Available: <http://www.tooloop.com/picture-of-blood-circulation-in-the-human-body/picture-of-blood-circulation-in-the-human-body-2/>.
- [2] "Cardiothoratic surgery." [Online]. Available: <http://www.cts.usc.edu/hpg-valvesoftheheart.html>.
- [3] "Wiggers diagram." [Online]. Available: http://en.wikipedia.org/wiki/Wiggers_diagram.
- [4] Marieb, *Human Anatomy and Physiology, 7th Edition*. .
- [5] "Heart valve surgery." [Online]. Available: <http://www.heart-valve-surgery.com/heart-surgery-blog/2008/09/02/mitral-valve-annulus-definition-diagrams-prolapse-calcification-treatment/>.
- [6] N. Kumar, M. Kumar, and C. M. Duran, "A revised terminology for recording surgical findings of the mitral valve.," *J. Heart Valve Dis.*, vol. 4, pp. 70–75; discussion 76–77, 1995.
- [7] A. F. Carpentier, A. Lessana, J. Y. M. Relland, E. Belli, S. Mihaileanu, A. J. Berrebi, E. Palsky, and D. F. Loulmet, "The 'Physio-Ring': An advanced concept in mitral valve annuloplasty," in *Annals of Thoracic Surgery*, 1995, vol. 60, pp. 1177–1186.
- [8] "Mitral valve leaflets." .
- [9] J.-P. M. Rabbah, N. Saikrishnan, A. W. Siefert, A. Santhanakrishnan, and A. P. Yoganathan, "Mechanics of healthy and functionally diseased mitral valves: a critical review.," *J. Biomech. Eng.*, vol. 135, no. 2, p. 021007, Feb. 2013.
- [10] P. Fornes, D. Heudes, J.-F. Fuzellier, D. Tixier, P. Bruneval, and A. Carpentier, "Correlation Between Clinical and Histologic Patterns of Degenerative Mitral Valve Insufficiency," *Cardiovasc. Pathol.*, vol. 8, no. 2, pp. 81–92, Mar. 1999.
- [11] "Libreta de notas." [Online]. Available: <http://libretadenotas93.blogspot.be>.
- [12] M. HOWARD C. HERRMANN, "Transcatheter Mitral Valve Implantation," 2009.
- [13] "CTSNet."
- [14] "Ede-to-edge repair." [Online]. Available: <http://circ.ahajournals.org/content/117/7/963/F7.expansion.html>.
- [15] K. S. Kunzelman, R. P. Cochran, C. Chuong, W. S. Ring, E. D. Verrier, and R. D. Eberhart, "Finite element analysis of the mitral valve.," *J. Heart Valve Dis.*, vol. 2, pp. 326–340, 1993.
- [16] K. S. Kunzelman, D. W. Quick, and R. P. Cochran, "Altered collagen concentration in mitral valve leaflets: biochemical and finite element analysis.," *Ann. Thorac. Surg.*, vol. 66, no. 6 Suppl, pp. S198–205, Dec. 1998.

- [17] K. Kunzelman, M. S. Reimink, E. D. Verrier, and R. P. Cochran, "Replacement of Mitral Valve Posterior Chordae Tendineae with Expanded Polytetrafluoroethylene Suture: A Finite Element Study," *J. Card. Surg.*, vol. 11, no. 2, pp. 136–145, 1996.
- [18] K. S. Kunzelman, M. S. Reimink, and R. P. Cochran, "Annular dilatation increases stress in the mitral valve and delays coaptation: a finite element computer model.," *Cardiovasc. Surg.*, vol. 5, no. 4, pp. 427–34, Aug. 1997.
- [19] K. S. Kunzelman, M. S. Reimink, and R. P. Cochran, "Flexible versus rigid ring annuloplasty for mitral valve annular dilatation: a finite element model.," *J. Heart Valve Dis.*, vol. 7, pp. 108–116, 1998.
- [20] F. Maisano, A. Redaelli, G. Pennati, R. Fumero, L. Torracca, and O. Alfieri, "The hemodynamic effects of double-orifice valve repair for mitral regurgitation: a 3D computational model.," *Eur. J. Cardiothorac. Surg.*, vol. 15, no. 4, pp. 419–25, Apr. 1999.
- [21] E. Votta, F. Maisano, M. Soncini, A. Redaelli, F. M. Montecvecchi, and O. Alfieri, "3-D computational analysis of the stress distribution on the leaflets after edge-to-edge repair of mitral regurgitation.," *J. Heart Valve Dis.*, vol. 11, pp. 810–822, 2002.
- [22] I. S. Salgo, J. H. Gorman, R. C. Gorman, B. M. Jackson, F. W. Bowen, T. Plappert, M. G. St John Sutton, and L. H. Edmunds, "Effect of annular shape on leaflet curvature in reducing mitral leaflet stress.," *Circulation*, vol. 106, pp. 711–717, 2002.
- [23] D. R. Einstein, P. Reinhall, M. Nicosia, R. P. Cochran, and K. Kunzelman, "Dynamic finite element implementation of nonlinear, anisotropic hyperelastic biological membranes.," *Comput. Methods Biomech. Biomed. Engin.*, vol. 6, pp. 33–44, 2003.
- [24] D. R. Einstein, K. S. Kunzelman, P. G. Reinhall, M. A. Nicosia, and R. P. Cochran, "Non-linear fluid-coupled computational model of the mitral valve.," *J. Heart Valve Dis.*, vol. 14, pp. 376–385, 2005.
- [25] F. Maisano, A. Redaelli, M. Soncini, E. Votta, L. Arcobasso, and O. Alfieri, "An annular prosthesis for the treatment of functional mitral regurgitation: Finite element model analysis of a dog bone-shaped ring prosthesis," *Ann. Thorac. Surg.*, vol. 79, pp. 1268–1275, 2005.
- [26] K. H. Lim, J. H. Yeo, and C. M. G. Duran, "Three-dimensional asymmetrical modeling of the mitral valve: a finite element study with dynamic boundaries.," *J. Heart Valve Dis.*, vol. 14, pp. 386–392, 2005.
- [27] E. Votta, E. Caiani, F. Veronesi, M. Soncini, F. M. Montecvecchi, and A. Redaelli, "Mitral valve finite-element modelling from ultrasound data: a pilot study for a new approach to understand mitral function and clinical scenarios.," *Philos. Trans. A. Math. Phys. Eng. Sci.*, vol. 366, no. 1879, pp. 3411–34, Sep. 2008.
- [28] E. Votta, F. Maisano, S. F. Bolling, O. Alfieri, F. M. Montecvecchi, and A. Redaelli, "The Geoform Disease-Specific Annuloplasty System: A Finite Element Study," *Ann. Thorac. Surg.*, vol. 84, pp. 92–101, 2007.

- [29] V. Prot, R. Haaverstad, and B. Skallerud, "Finite element analysis of the mitral apparatus: annulus shape effect and chordal force distribution.," *Biomech. Model. Mechanobiol.*, vol. 8, no. 1, pp. 43–55, Feb. 2009.
- [30] V. Prot, B. Skallerud, G. Sommer, and G. a Holzapfel, "On modelling and analysis of healthy and pathological human mitral valves: two case studies.," *J. Mech. Behav. Biomed. Mater.*, vol. 3, no. 2, pp. 167–77, Feb. 2010.
- [31] V. Prot and B. Skallerud, "Nonlinear solid finite element analysis of mitral valves with heterogeneous leaflet layers," *Computational Mechanics*, vol. 43. pp. 353–368, 2009.
- [32] J. F. Wenk, Z. Zhang, G. Cheng, D. Malhotra, G. Acevedo-Bolton, M. Burger, T. Suzuki, D. a Saloner, A. W. Wallace, J. M. Guccione, and M. B. Ratcliffe, "First finite element model of the left ventricle with mitral valve: insights into ischemic mitral regurgitation.," *Ann. Thorac. Surg.*, vol. 89, no. 5, pp. 1546–53, May 2010.
- [33] P. Burlina, C. Sprouse, D. Dementhon, A. Jorstad, R. Juang, F. Contijoch, T. Abraham, D. Yuh, and E. Mcveigh, "Patient-Specific Modeling and Analysis of the Mitral Valve Using 3D-TEE," pp. 135–146, 2010.
- [34] C. A. Conti, M. Stevanella, F. Maffessanti, S. Trunfio, E. Votta, A. Roghi, O. Parodi, E. G. Caiani, A. Redaelli, P. Milano, N. Ca, and G. Hospital, "Mitral Valve Modelling in Ischemic Patients : Finite Element Analysis from Cardiac Magnetic Resonance Imaging CNR Institute of Clinical Physiology , Pisa , Italy," pp. 1059–1062, 2010.
- [35] I. A. E. W. Van Loosdregt, I. A. E. W. Van Loosdregt, B. Sc, R. Amini, D. Ph, and M. M. Sacks, "Toward developing a finite element model of the mitral valve Towards developing a finite element model of the mitral valve," no. July, pp. 1–10, 2011.
- [36] A. S. Jassar, C. J. Brinster, M. Vergnat, J. D. Robb, T. J. Eperjesi, A. M. Pouch, A. T. Cheung, S. J. Weiss, M. a Acker, J. H. Gorman, R. C. Gorman, and B. M. Jackson, "Quantitative mitral valve modeling using real-time three-dimensional echocardiography: technique and repeatability.," *Ann. Thorac. Surg.*, vol. 91, no. 1, pp. 165–71, Jan. 2011.
- [37] A. Quaini, S. Canic, G. Guidoboni, R. Glowinski, S. R. Igo, C. J. Hartley, W. A. Zoghbi, and S. H. Little, "A Three-Dimensional Computational Fluid Dynamics Model of Cardiovasc Eng Technol Regurgitant Mitral Valve Flow: Validation Against in vitro Standards and 3D Color Doppler Methods," vol. 2, no. 2, pp. 77–89, 2012.
- [38] Q. Wang and W. Sun, "Finite element modeling of mitral valve dynamic deformation using patient-specific multi-slices computed tomography scans.," *Ann. Biomed. Eng.*, vol. 41, no. 1, pp. 142–53, Jan. 2013.
- [39] P. Burlina, C. Sprouse, R. Mukherjee, D. DeMenthon, and T. Abraham, "Patient-specific mitral valve closure prediction using 3D echocardiography.," *Ultrasound Med. Biol.*, vol. 39, no. 5, pp. 769–83, May 2013.

- [40] A. M. Pouch, C. Xu, P. A. Yushkevich, A. S. Jassar, M. Vergnat, J. H. G. Iii, R. C. Gorman, C. M. Sehgal, and B. M. Jackson, "Semi-Automated Mitral Valve Morphometry and Computational Stress Analysis Using 3D Ultrasound," vol. 45, no. 5, pp. 903–907, 2013.
- [41] Y. Rim, D. D. McPherson, K. B. Chandran, and H. Kim, "The effect of patient-specific annular motion on dynamic simulation of mitral valve function.," *J. Biomech.*, vol. 46, no. 6, pp. 1104–12, Apr. 2013.
- [42] K. May-Newman and F. C. Yin, "Biaxial mechanical behavior of excised porcine mitral valve leaflets.," *Am. J. Physiol.*, vol. 269, pp. H1319–H1327, 1995.
- [43] K. May-Newman and F. C. Yin, "A constitutive law for mitral valve tissue.," *J. Biomech. Eng.*, vol. 120, pp. 38–47, 1998.
- [44] D. R. Einstein, K. S. Kunzelman, P. G. Reinhall, R. P. Cochran, and M. A. Nicosia, "Haemodynamic determinants of the mitral valve closure sound: a finite element study.," *Med. Biol. Eng. Comput.*, vol. 42, pp. 832–846, 2004.
- [45] D. R. Einstein, K. S. Kunzelman, P. G. Reinhall, M. A. Nicosia, and R. P. Cochran, "The relationship of normal and abnormal microstructural proliferation to the mitral valve closure sound.," *J. Biomech. Eng.*, vol. 127, pp. 134–147, 2005.
- [46] V. Prot, B. Skallerud, and G. A. Holzapfel, "Transversely isotropic membrane shells with application to mitral valve mechanics . Constitutive modelling and finite element implementation," *Int. J. Numer. Methods Eng.*, vol. 71, pp. 987–1008, 2007.
- [47] K. D. Lau, V. Diaz, P. Scambler, and G. Burriesci, "Mitral valve dynamics in structural and fluid-structure interaction models.," *Med. Eng. Phys.*, vol. 32, no. 9, pp. 1057–64, Nov. 2010.
- [48] M. Stevanella, F. Maffessanti, C. A. Conti, E. Votta, A. Arnoldi, M. Lombardi, O. Parodi, E. G. Caiani, and A. Redaelli, "Mitral Valve Patient-Specific Finite Element Modeling from Cardiac MRI: Application to an Annuloplasty Procedure," *Cardiovascular Engineering and Technology*, vol. 2, pp. 66–76, 2011.
- [49] S. Schievano, K. Kunzelman, M. A. Nicosia, R. P. Cochran, and R. Daniel, "Percutaneous Mitral Valve Dilatation : Single Balloon ver- sus Double Balloon . A Finite Element Study," 2009.
- [50] A. Avanzini, "A computational procedure for prediction of structural effects of edge-to-edge repair on mitral valve.," *J. Biomech. Eng.*, vol. 130, p. 031015, 2008.
- [51] A. Avanzini, G. Donzella, and L. Libretti, "Functional and structural effects of percutaneous edge-to-edge double-orifice repair under cardiac cycle in comparison with suture repair," *Eng. Med.*, vol. 225, pp. 959–971, 2011.
- [52] K. D. Lau, V. D??az-Zuccarini, P. Scambler, and G. Burriesci, "Fluid-structure interaction study of the edge-to-edge repair technique on the mitral valve," *J. Biomech.*, vol. 44, pp. 2409–2417, 2011.

- [53] B. Verhegghe, "pyFormex." .
- [54] C. R. Kunzelman KS1, "Mechanical properties of basal and marginal mitral valve chordae tendineae," 1990.
- [55] "Django architecture." [Online]. Available: <http://mytardis.readthedocs.org/en/2.5/architecture.html#functional-architecture>.
- [56] "Bootstrap." [Online]. Available: <http://getbootstrap.com/>.

Appendices

Appendix A: pyFormex application user manual

General information

General Information section explains in general terms the system and the purpose for which it is intended.

System overview

MitralValvepyF is an application, which allows the generation of a patient specific mitral valve input file for further use in Abaqus software. The application provides possibility for loading patients' specific data from files and changing the essential parameters of the mitral valve apparatus. The application generates and saves input files for Abaqus software and also saves all of the entered parameters in format that can be reused later on. It enables a preview of the 3D finite element analysis model of the mitral valve.

Organization of the manual

The user's manual consists of four sections: General Information, System Summary, Getting Started and Using The System.

General Information section explains in general terms the system and the purpose for which it is intended.

System Summary section provides a general overview of the system. It outlines the uses of the system's hardware and software requirements, system's configuration, user access levels and system's behavior in case of any contingencies.

Getting Started section explains how to get MitralValvepyF. The section presents briefly the system menu.

Using The System section provides a detailed description of system functions.

System summary

System Summary section provides a general overview of the system. The summary outlines the uses of the system's hardware and software requirements, system's configuration, user access levels and system's behavior in case of any contingencies.

System configuration

MitralValvepyF operates on computer platforms with Linux operating system. It is compatible with Ubuntu 10 or higher, Fedora 13 or higher, Debian 7 or higher releases. The application requires installed software: pyFormex 0.9.1 or higher version. After having pyFormex up and running the application doesn't require any further configuration.

User access levels

Everyone can use the application and registration is not needed.

Contingencies

In case of power outage, data are not saved in internal memory of the operating device.

Getting started

Getting Started section explains how to get MitralValvepyF. The section presents briefly system menu.

Installation

The source code of the application can be downloaded from <http://tinyurl.com/MitralValvepyF>. It is a Python file that can be loaded in the pyFormex program. The only thing that is necessary to be done after loading the script in pyFormex is to run it.

System menu

MitralValvepyF is a tabbed application, which consists of four tabs (Figure 39). There is tab for loading the patient specific data files and a tab in which essential parameters of the mitral valve can be entered. The third tab contains the material models for the anterior and posterior leaflets of the mitral valve. In the last tab user preferences about the colors in which the 3D finite element analysis model are available.

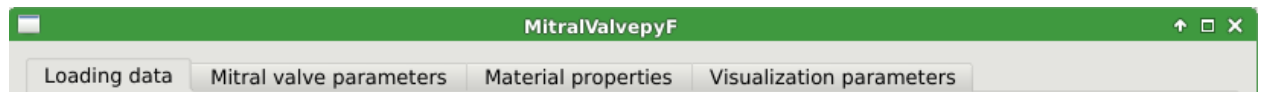


Figure 39 Tabs of MitralValvepyF

- Loading data tab

The Loading data tab (Figure 40) consists of three fields in which files can be chosen. In the first file field a TRC file can be loaded, in the second field - CSV file and the third one - PGF file. All of the files are required in order for the application to work correctly.

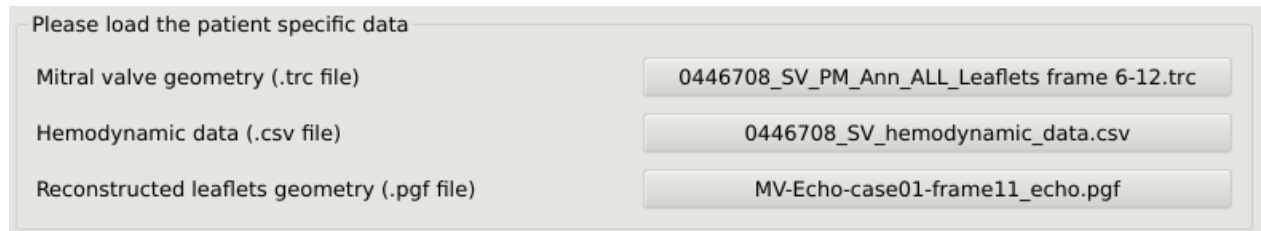


Figure 40 MitralValvepyF : Loading data tab

- Mitral valve parameters tab

Mitral valve tab consists of 10 input fields containing the information about essential parameters necessary for the generation of the finite analysis model of the patient specific data.

The screenshot shows a software interface for defining mitral valve parameters. It is organized into three main sections:

- Leaflets:** Contains two input fields: "Thickness of the anterior leaflets (mm)" with a value of 0.25, and "Thickness of the posterior leaflets (mm)" with a value of 0.25.
- Chordae tendineae:** Contains several input fields and a type selector:
 - "Number of the basal chordae" with a value of 12 and a corresponding slider.
 - "Number of the marginal chordae" with a value of 28 and a corresponding slider.
 - "Relative height of basal chordae" with a value of 0.5.
 - "Cross section marginal chordae(mm)" with a value of 0.4.
 - "Cross section basal chordae(mm)" with a value of 0.23.
 - "Type of the chordae tendinae" with radio buttons for "truss" (selected) and "connectors".
- Simulation parameters:** Contains two input fields:
 - "Number of CPUs to be used by the cluster" with radio buttons for 4 (selected), 6, and 8.
 - "Friction coefficient" with a value of 0.05.

Figure 41 MitralValvepyF : Mitral valve parameters tab

- Material properties tab

The material properties tab contains information about the material models of the anterior and posterior leaflets. It has two child tabs: one for the material properties of the anterior leaflet (Figure 42) and one tab for the material properties of the posterior leaflet (Figure 43). Every tab contains check buttons, which in the beginning are not active. Once the check button is active depending on the chosen material model, parameters can be changed. Every tab has seven input fields, two for the linear elastic material model and five for the hyperelastic Holzapfel material model.

Anterior leaflet Posterior leaflet

Linear elastic

Parameters

Young modulus(GPa) 117

Poisson radio 0.355

Hyperelastic anisotropic

Holzapfel parameters:

C10 0.1245

D 0.0005

K1 11.0069

K2 84.8478

kappa 0.08

Figure 42 MitralValvepyF: Material properties tab/ Anterior leaflet tab

Anterior leaflet Posterior leaflet

Linear elastic

Parameters

Young modulus(GPa) 117

Poisson radio 0.355

Hyperelastic anisotropic

Holzapfel parameters:

C10 0.0502

D 0.003

K1 3.0207

K2 144.4848

kappa 0.0534

Figure 43 MitralValvepyF : Material properties tab/ Posterior leaflet tab

- Visualization parameters tab

The visualization parameters tab contains seven input color fields (Figure 44).

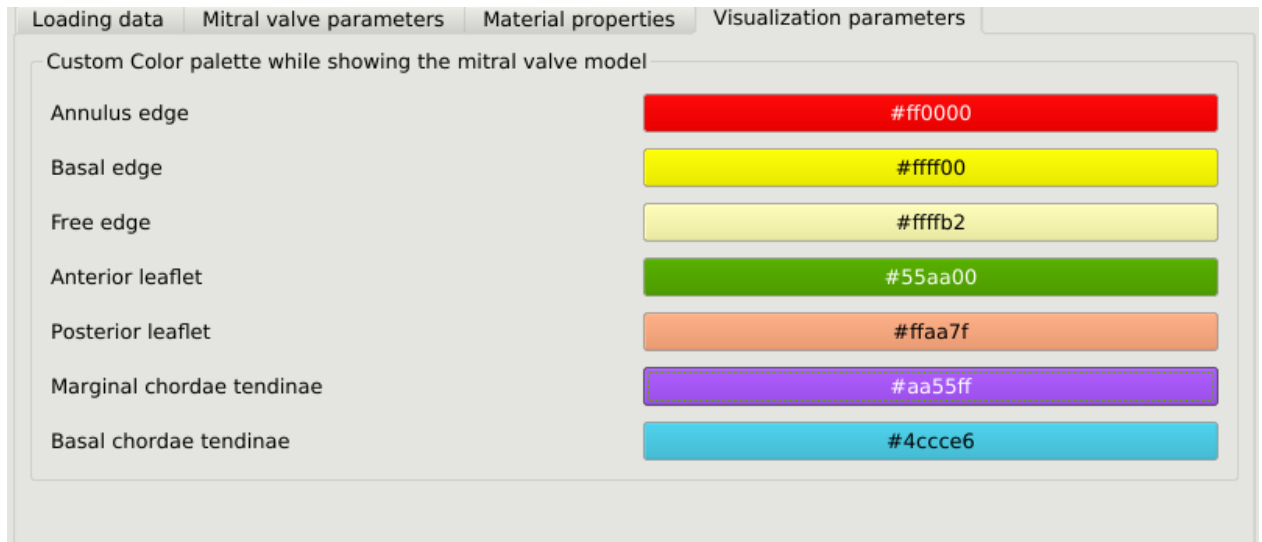


Figure 44 MitralValvepyF : Visualization parameters tab

Exit system

The “x” button on the right upper screen and the close button at the left down corner are the tools that can be used in order to exit the application (Figure 45).

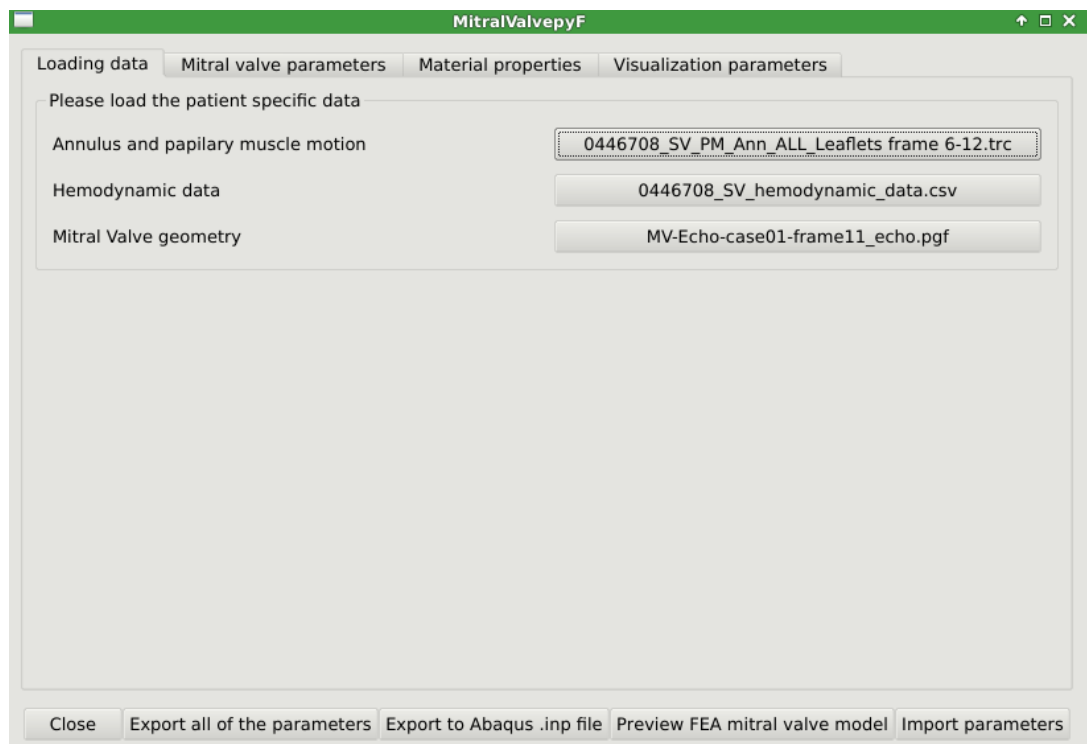


Figure 45 MitralValvepyF input dialog

Using the system

This section provides a detailed description of system functions.

- Loading data tab

Loading data tab consist of three file fields. The first file field is about loading a TRC file which contains information about the geometry of the mitral valve and the annular and papillary muscles. The second file field loads a PGF file which contains the mesh's nodes and connectivity of the anterior and posterior leaflets. The third file field loads CSV file which contains the patient specific transvalvular pressure. The three file fields react like a button and one is clicked on some of them a pop-up window appears. In this window only a file with the chosen format can be seen. At Figure 46. An example of a pop-up after clicking on the TRC file field is showed.

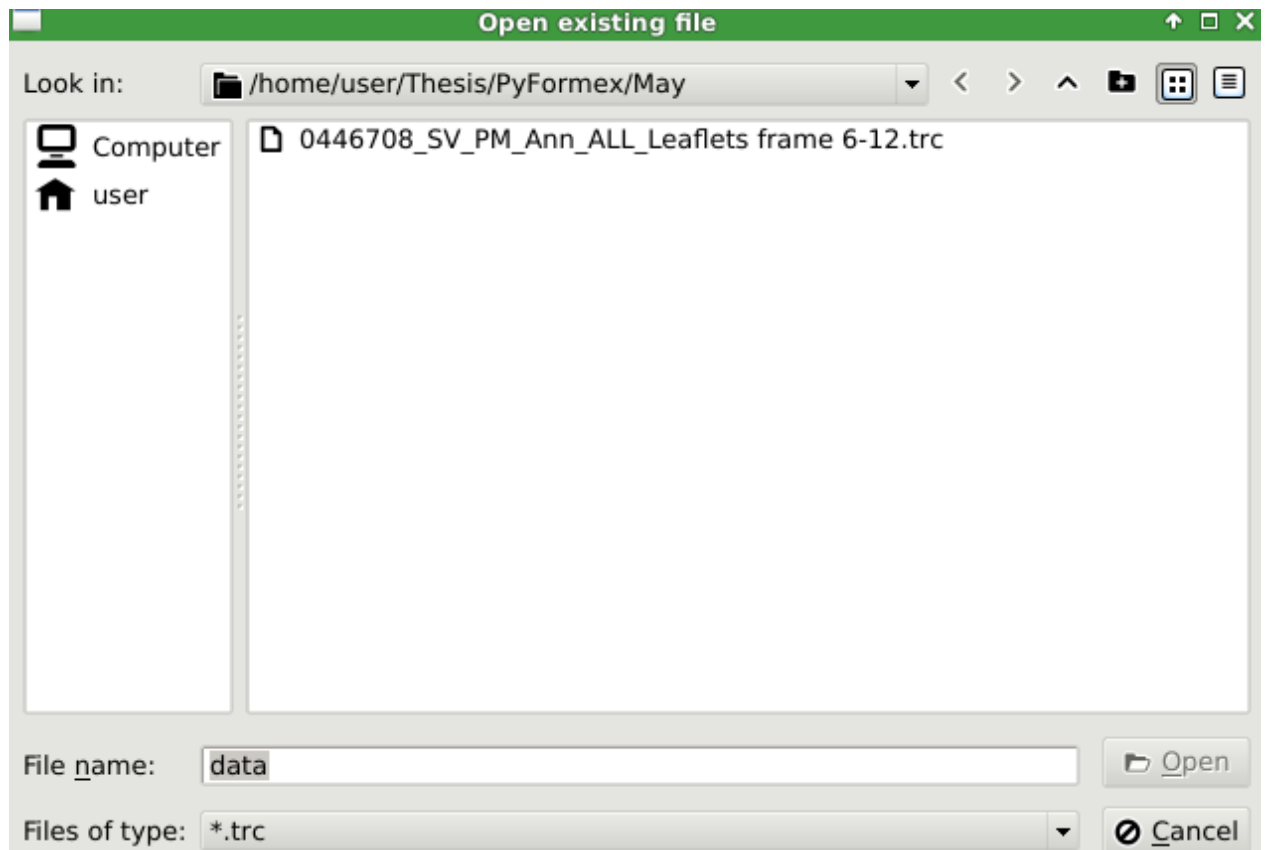


Figure 46 MitralValvepyF :Loading TRC file pop-up window

- Mitral valve parameters tab

Mitral valve parameters tab consist of ten input fields. Thickness of the anterior and posterior leaflets, the relative height of the basal chordae tendinae, the cross section of the basal and marginal chordae and the friction coefficient are numerical fields. The number of the marginal and basal chordae tendinae are slider fields in which only an integer value can be entered. The number of CPUs to be used by the cluster field can be defined from a radio button which offers one of the three values to chosen from.

- Material properties tab

Material properties tab consists of two tabs referring to different material models for the anterior and posterior leaflets respectively. In every tab there are checkboxes for the linear elastic and hyperelastic Holzapfel models. Note that both of the checkboxes can be active in every tab, but one should choose only one of them in order for the application to work correctly. After a checkbox is clicked on, the input fields of the checkbox become active. In the case when the linear elastic checkbox is active, two numerical fields become available, while in the case when the checkbox for hyperelastic material model is activated five numerical fields about the parameters of the Holzapfel model become editable.

- Visualization parameters tab

The visualization parameters tab consists of seven color fields. The parameters according to which the colors can be changed are: annulus edge, basal edge, free edge, anterior leaflet, posterior leaflet, marginal chordae and basal chordae. Every color can be changed by clicking on in. A color palette pops out and the color can be selected from it (Figure 47).

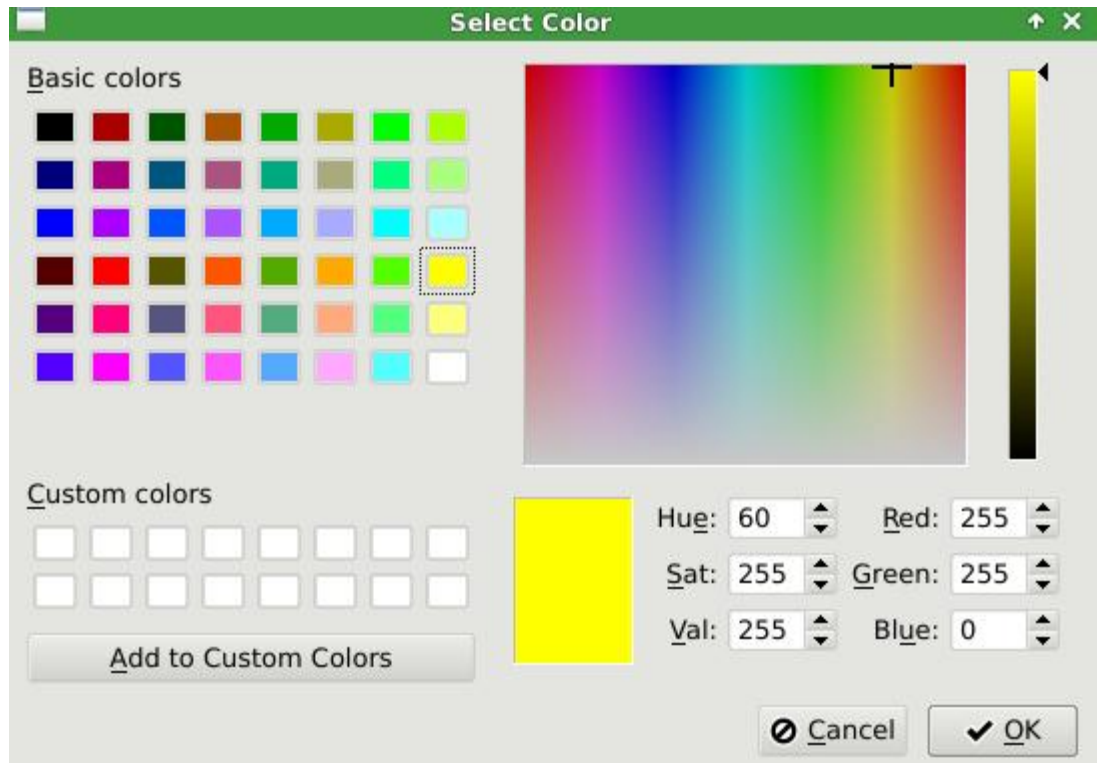


Figure 47 MitralValvepyF :Visualization parameters color palette pop-up

The MitralValvepyF application has five buttons with different functionalities: exporting all of the input parameters, import of mitral valve parameters, generation of the 3D finite element analysis model and generation of input Abaqus file . The buttons are situated at the left bottom corner (Figure 45).

- Export all parameters button

The Export all parameters button makes possible to save all of the input parameters in JSON file. By clicking on the button a pop-up window appears. In it a place where the file will be saved as well as a specific name can be chosen (Figure 48).

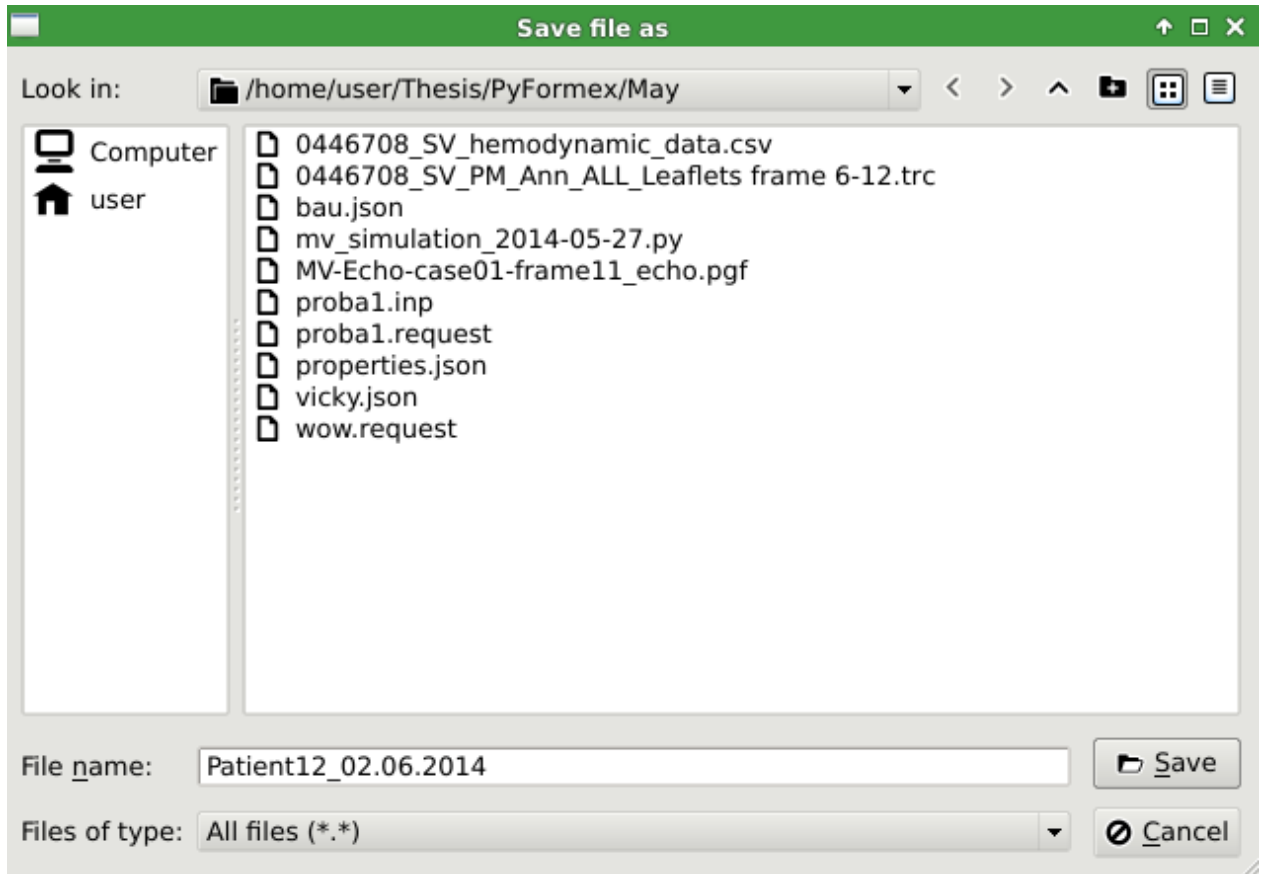


Figure 48 MitralValvepyF: Export all parameters button pop-up

- Export to Abaqus .inp file button

The Export to Abaqus .inp file button takes the loaded patient-specific data file as well as all of the input parameters and generates an input Abaqus file. The application also generates a REQUEST file automatically. The REQUEST file is used for simulations on the bumper cluster of UGent. Once the button is clicked on, a pop-up window appears asking for confirmation (Figure 49).

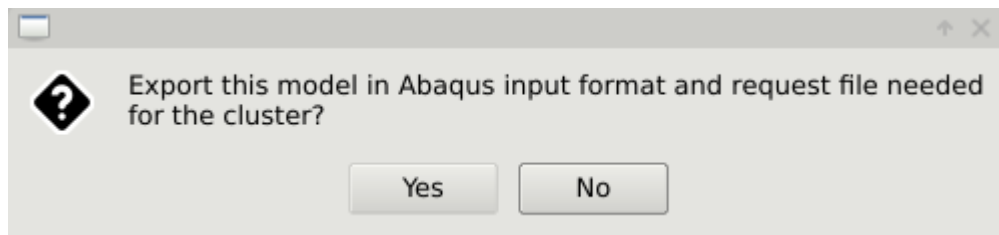


Figure 49 MitralValvepy: Export to Abaqus. inp file pop-up window

After approval another pop-up window comes out asking where and with what name to save the INP file. The pop-up window is similar to the one at Figure 48.

Once the INP and REQUEST files are made a confirmation message shows up (Figure 50).

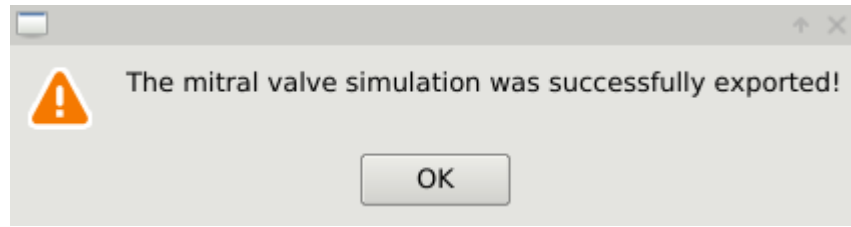


Figure 50 MitralValvepy: Successful generation of INPUT file message

- Preview FEA mitral valve model button

The Preview FEA mitral valve button takes all of the loaded patient-specific data files as well as the input parameters and generates a 3D finite analysis model of the mitral valve. Once the button is clicked on and the 3D model is generated a pop-up message appears (Figure 51).

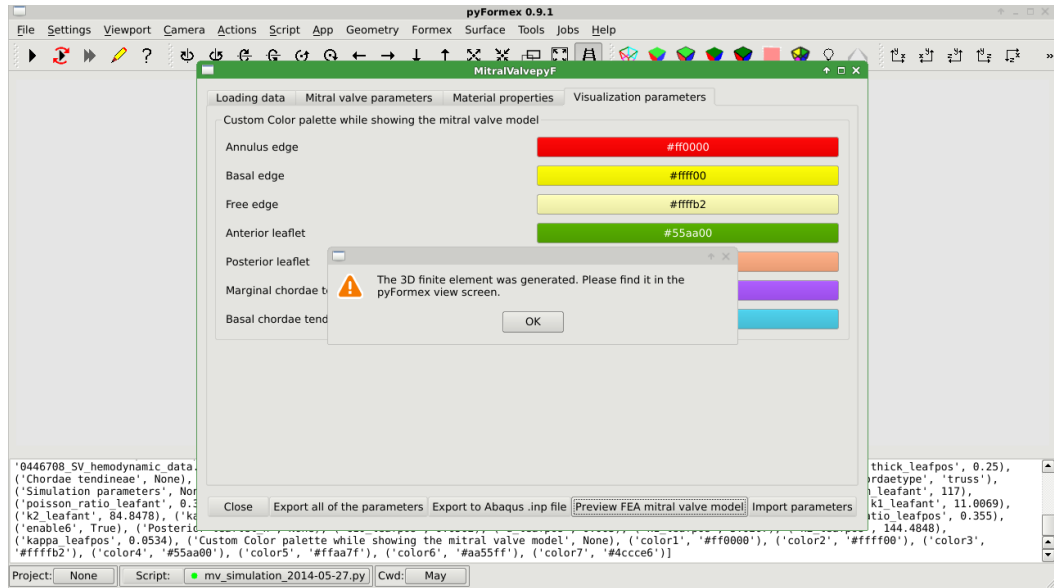


Figure 51 MitralValvepyF: Preview of FEA mitral valve model pop-up

Once the message appears, a 3D finite element (FEA) mitral valve model is generated in the view window of pyFormex which is behind the application. In order to see the FEA model, one has to move the application screen or close it. Example of generated 3D FEA model is shown at Figure 52.

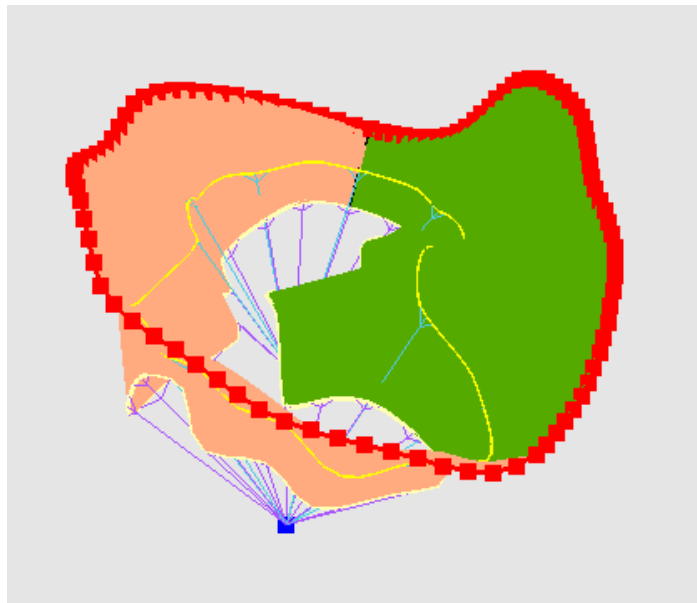


Figure 52 MitralValvepyF: Example of generated 3D FEA mitral valve model

The FEA model can be rotated and seen from different angles by using the mouse buttons.

- Import parameters

Import parameters button makes possible to choose existing JSON file with mitral valve parameters and loading the information from it in the graphic user interface of the application automatically. After clicking on the button, a pop-up window appears. From this window a specific JSON file can be loaded (Figure 53).

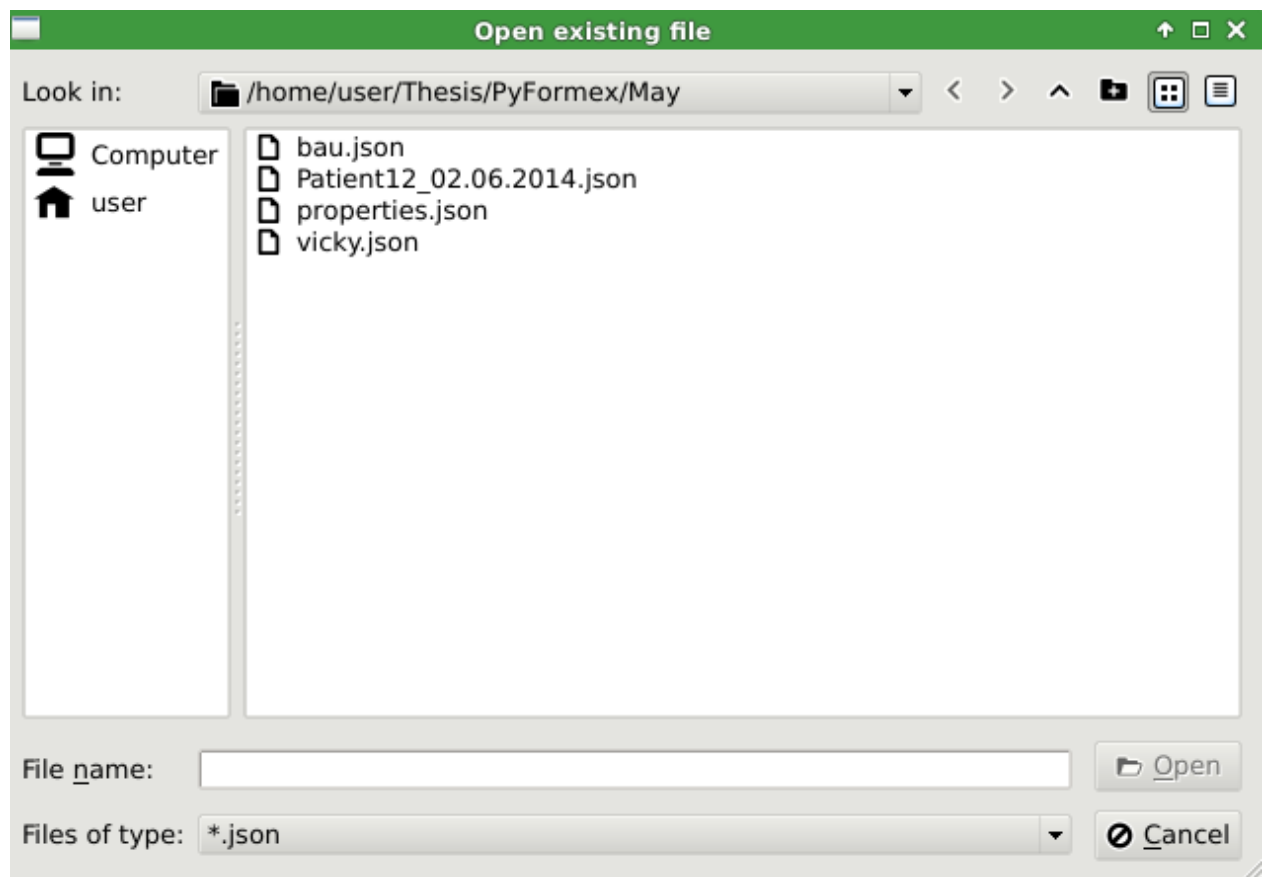


Figure 53 MitralValvepyF: Import parameters button pop-up window

If the import of the mitral valve parameters is successful a confirmation message will appear (Figure 54). Note that if the JSON file is imported correctly, you still need to activate the

checkboxes for the desired material properties of the anterior and posterior leaflets in the Material properties tab.

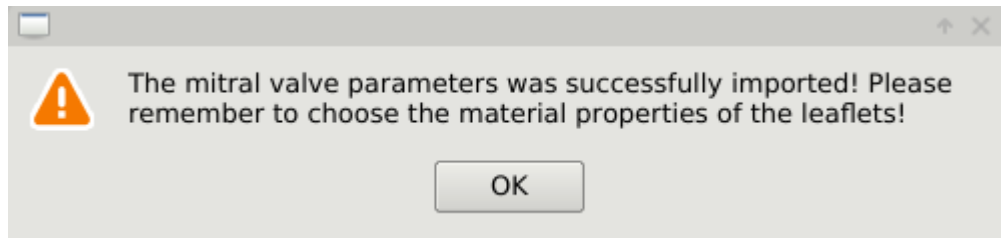


Figure 54 MitralValvepy: Import parameters button message window

Appendix B:Django application user manual

General information

General Information section explains in general terms the system and the purpose for which it is intended.

System overview

MitralValveDj is web based application, which allows collecting patient-specific data for the mitral valve. The application provides electronic version of form in collecting mitral valve parameters. The application saves the collected data to a database. MitralValaveDJ operates on personal computers with any operating system.

Organization of the manual

The user's manual consists of four sections: General Information, System Summary, Getting Started and Using The System.

General Information section explains in general terms the system and the purpose for which it is intended.

System Summary section provides a general overview of the system. The summary outlines the uses of the system's hardware and software requirements, system's configuration, user access levels and system's behavior in case of any contingencies.

Getting Started section explains how to get MitralValveDJ and install it. The section presents briefly the system menu.

Using The System section provides a detailed description of system functions.

[System summary](#)

System Summary section provides a general overview of the system. The summary outlines the uses of the system's hardware and software requirements, system's configuration, user access levels and system's behavior in case of any contingencies.

[System configuration](#)

MitralValveDj operates on computer platforms with any operating system. The application requires installed Django framework 1.6.1+, installed browser: Mozilla Firefox 24+, Google Chrome 30+ and installed database SQLite 2.8.17+. After installing all of the required software and running the server, the MitralValveDj can be used immediately without any further configuration.

[User access levels](#)

Only registered users can use the application.

[Contingencies](#)

In case of power outage data are not saved in internal memory of the computer system.

[Getting started](#)

Getting Started section explains how to get MitralValveDj and install it on the device. The section presents briefly system menu.

Installation and logging in

The newest version of MitralValveDj can be downloaded from <http://tinyurl.com/MitralValveDj> and it is in ZIP file, which should be unzipped on the personal computer where the application will be used. After unzipping the archive you should go to the directory MitralValveDj/MV/ via the terminal and write the following code in order to run the server:

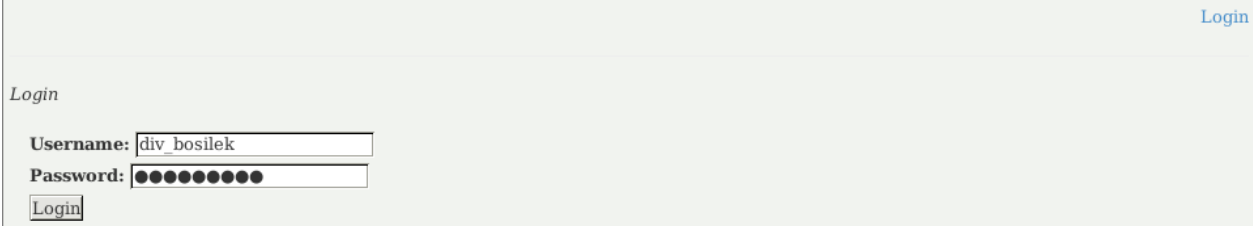
```
$ python manage.py runserver
```

If the server runs correctly the message as shown at Figure 55 should appear.

```
Validating models...  
  
0 errors found  
June 04, 2014 - 14:12:46  
Django version 1.6.2, using settings 'MV_april.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CONTROL-C.
```

Figure 55 MitralValveDj: Run server message

Once the server is working, web browser should be open and an address 127.0.0.1:8000 should be entered in the address bar. The MitralValveDj login screen appears (Figure 56).



Username:

Password:

Figure 56 MitralValveDj: Login screen

For a username and password please contact Viktoriya Antonova at Viktoriya.Antonova@ugent.be.

System menu

MitralValveDj is a tabbed application which consists of four tabs (Figure 57). In the first tab patients-specific data files can be loaded, in the second tab essential parameters of the mitral valve can be edited. The third tab has user preferences about the material models of the anterior and posterior leaflets. In the last tab, parameters related to the simulation can be changed.

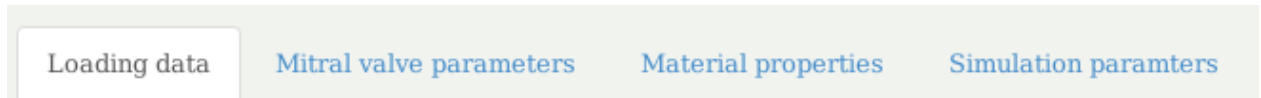


Figure 57 MitralValveDj: Tab menu

- Loading data tab

Loading data tab consist of three file fields (Figure 58). The first file field is about loading a TRC file which contains information about the geometry of the mitral valve and the annular and papillary muscles. The second file field loads a CSV file which contains the patient specific transvalvular pressure. The third one loads a PGF file containing the mesh's nodes and connectivity of the anterior and posterior leaflets. If a file is chosen the name appears next to the field, if not a message "No file chosen" will show. Note that depending on the browser the visualization of the file fields can be different.

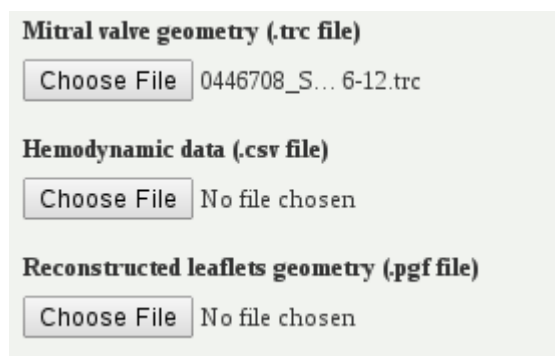


Figure 58 MitralValveDj: Loading data tab

- Mitral valve parameters tab

Mitral valve parameters tab consists of eight fields. Two fields regard the leaflets and their thickness and the other fields are essential parameters of the chordae tendinae (Figure 59).

Leaflets	
Thickness of the anterior leaflet (mm)	<input type="text" value="0.25"/>
Thickness of the posterior leaflet (mm)	<input type="text" value="0.25"/>
Chordae tendineae	
Number of the basal chordae	<input type="text" value="12"/>
Number of the marginal chordae	<input type="text" value="12"/>
Relative height of basal chordae	<input type="text" value="0.5"/>
Cross section marginal cordae (mm)	<input type="text" value="0.4"/>
Cross section basal cordae (mm)	<input type="text" value="0.23"/>
Type of the chordae tendinae	<input checked="" type="radio"/> truss <input type="radio"/> connectors

Figure 59 MitralValveDj: Mmitral valve parameters tab

- Material properties tab

Material properties tab consist of two radio buttons: one for the anterior leaflet and one for the posterior (Figure 60). Each of the radio buttons consists of two options which are linear elastic or hyperelastic Holzapfel material models. Depending on the material properties, two parameters can be entered for the linear elastic and five parameters can be submitted for the hyperelastic model.

Anterior leaflet

Linear elastic

Hyperelastic anisotropic

Posterior leaflet

Linear elastic

Hyperelastic anisotropic

Figure 60 MitralValveDj: Mmaterial properties tab

- Simulation parameters

The Simulation parameters tab consists of two fields as shown in Figure 61.

Friction coefficient

Number of CPUs to be used in the cluster 4 6 8

Figure 61 MitralValveDj: Simulation parameters tab

[Exit system](#)

To exit the system, logout from it.

[Using the system](#)

After login in the system, a welcome screen shows. In this window a list of submitted patient-specific data and parameters can be found. There is a button “Add new patient data” which leads to the window in which patient data can be submitted. In Figure 62 only a welcome screen of newly registered user is shown, because there are no submitted patient data yet.

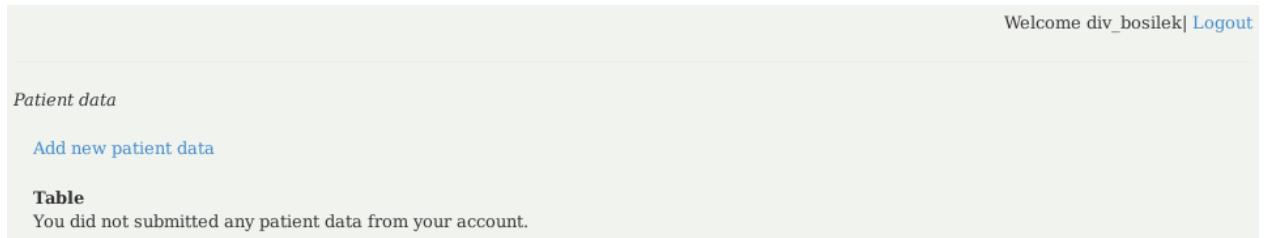


Figure 62 MitralValveDj: Welcome screen

Once the “Add new patient data” is clicked, “Submit patient data” screen appears (Figure 63).

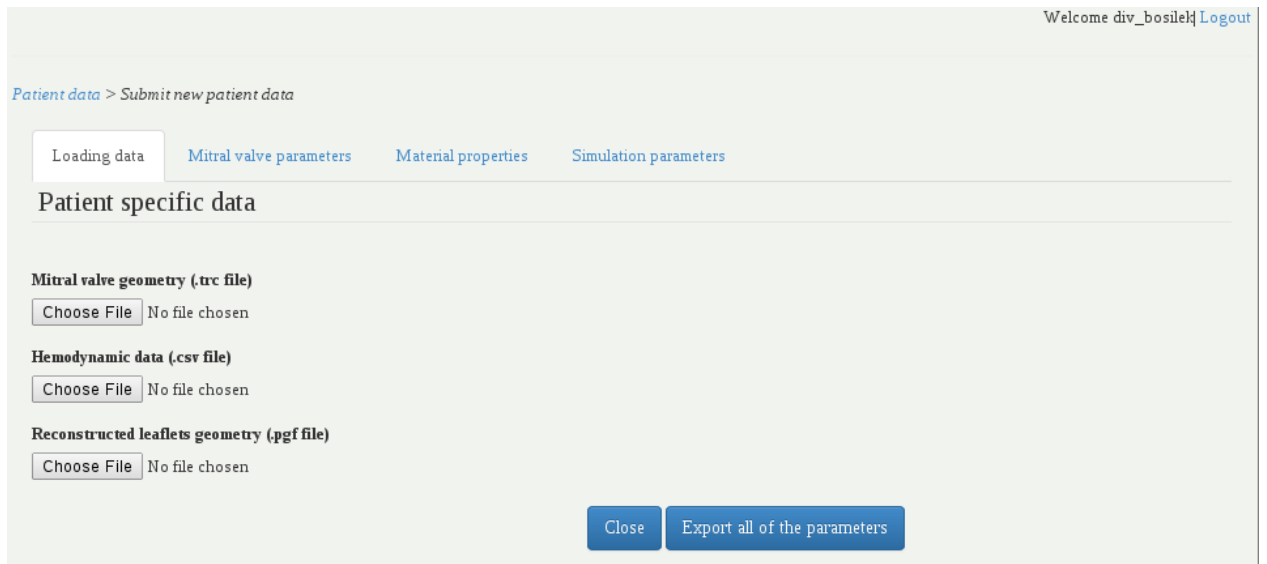


Figure 63 MitralValveDj: Submit patient data screen

- Loading data tab

Loading data tab consists of three file fields in which TRC, CSV and PGF files can be load. After clicking on the Choose File button, a pop-up window appears and the files can be chosen. If a file is not selected a message “No file chosen” next to the file field will appear. If a file is loaded, its name will appear. The MitralValveDj application makes verification for the file extension and if not a right file is selected, an error message is shown (Figure 64) and will lead to an empty “Submit new patient data” form Note that all of the file fields are required.

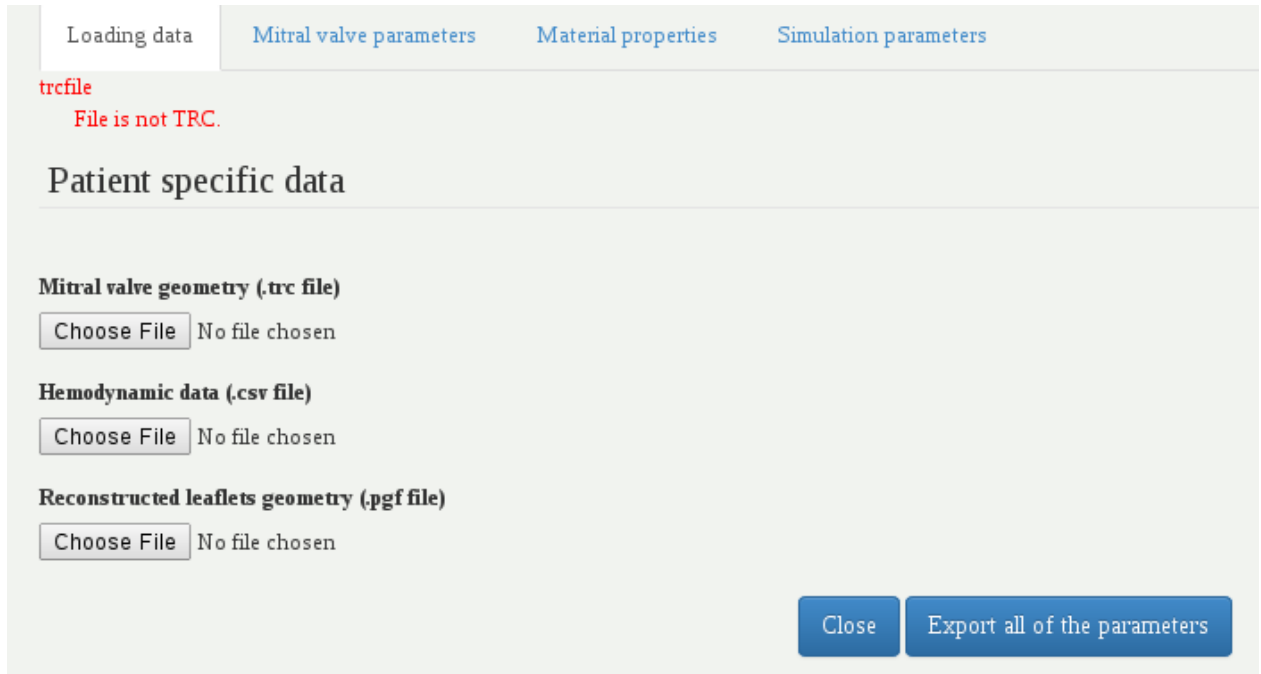


Figure 64 MitralValveDj: Verification of the file extensions

- Mitral valve parameters tab

Mitral valve parameters tab consists of eight fields. The first two fields are float fields concerning the thickness of the anterior and posterior leaflets. The next two fields are integer fields corresponding to the number of basal and marginal chordae tendinae. The next three fields are float fields about the relative height of the basal chordae, cross section of marginal and basal chordae. The last field is a radio button field with two choices for each type of chordae tendinae. Note that all of the fields are required and if any of them is missing enter data, an error message will appear.

- Material properties tab

The material properties tab consists of two radio buttons - one for the material model of the anterior leaflet and one for the posterior. Every radio button consists of two options: one for the linear elastic properties and one for the hyperelastic Holzapfel model. When checking the linear elastic option, two numerical fields are shown: one for the Young modulus and one for the Poisson ratio (Figure 65).

Anterior leaflet

Linear elastic

Parameters

Young modulus (GPa)

Poisson ratio

Figure 65 MitralValveDj: Linear elastic material model

If a hyperelastic option is active a table with five numerical fields shows up. The fields represent the parameters necessary for the Holzapfel model (Figure 66).

Posterior leaflet

Linear elastic

Hyperelastic anisotropic

Holzapfel

Parameters

C10	D	k1	k2	kappa
0.0502	0.003	3.0207	144.4848	0.0534

Figure 66 MitralValveDj: Hyperelastic material model

- Simulation parameters tab

The simulation tab consists of two fields. The first one is a float field concerning the friction coefficient and the second one is a radio button with three options regarding the number of computers that will be used in order to solve the simulation.

- Export all parameters button

At the right bottom corner of the “Submit new patient data” window, there is a button “Export all of the parameters”. Once it is clicked on, the system will show whether there is some error and if not, will bring up the welcome screen. If the system brings up the welcome screen, it means that the three files (TRC, PGF, CSV) were saved in the database and JSON file with all of the input parameters has been generated. Note that the JSON file can be loaded in MitralValvepyF.

- List of the submitted patient data entries

As mentioned above, in the welcome screen there is a table with all of the submitted patient data entries sorted by ID (Figure 67).

Welcome div_bosilek| [Logout](#)

Patient data

[Add new patient data](#)

Table

ID	Submitted date	Patient specific files	Mitral valve parameters	Actions
1	June 4, 2014, 5:23 p.m.	- TRC file - PGF file - CSV file	- JSON file	delete

Figure 67 MitralValveDj: List of submitted patient data entries

In the table, there is a possibility to open and see every file previously loaded for the patient-specific data as well as the JSON file. There is a “delete” button which will delete the entry.