Proof of Concept of Model-based Cardiovascular Prediction

Call identifier: FP7-ICT-2013-10 - **Grant agreement no**: 611232

Thematic Priority: ICT-2013.5.2: Virtual Physiological Human

Deliverable 4.1

Implementation of mechanics-CFD interface

Due date of delivery: 31st January 2014

Actual submission date: 31st January 2014

Start of the project: 1st October 2013
Ending Date: 30th September 2016

Partner responsible for this deliverable: MUG

Version: 1.2

Dissemination Level: Public

Document Classification

Title	Implementation of mechanics-CFD interface
Deliverable	4.1
Reporting Period	October 2013 – January 2014
Authors	MUG
Work Package	WP4 – Electro-mechanical model of the heart
Security	PU
Nature	Report
Keyword(s)	Software, conversion, finite element

Document History

Name	**Remark**	**Version**	**Date**
Gernot Plank	First draft	0.1	15th January 2014
Paul Groenenboom	Second draft	0.2	17th January 2014

List of Contributors

Name	**Affiliation**
Gernot Plank	MUG
Christoph Augustin	MUG
Anton Prassl	MUG
Paul Groenenboom	ESI

List of reviewers

Name	**Affiliation**
Paul Groenenboom	ESI
Edwin Morley-Fletcher	LYNKEUS

Abbreviations

FE	**Finite Element**
CFD	**Computational Fluid Dynamics**
SPH	**Smoothed Particle Hydrodynamics**
CARP	**Cardiac Arrhythmia Research Package**

Contents

General Description

The main objective of D4.1 is to define a technical standard which allows exchanging data between the electro-mechanical simulation output, as produced by the simulation codes used at the MUG, and the CFD codes used in other work packages. Electro-mechanical simulations are performed using 3D volumetric FE meshes of the heart. The inner surfaces of the heart define the boundaries of the fluid domain, i.e. the blood pool within the heart's cavities. In the considered scenario mechanical deformation and blood flow are considered to be independent (computation of mechanical deformation and blood flow are independent of each other and therefore can be computed sequentially). Thus the inner surfaces of the heart have to be extracted from the 3D volumetric mesh and exported in a format that is a suitable for the CFD simulation tools. The overall process is illustrated in Figure 1. Surfaces relevant for the definition of CFD boundaries are labelled in the volumetric mesh (Step 1), extracted for all time steps computed in the electro-mechanical simulation run (Step 2) and used as input in the CFD simulation (Step 3).

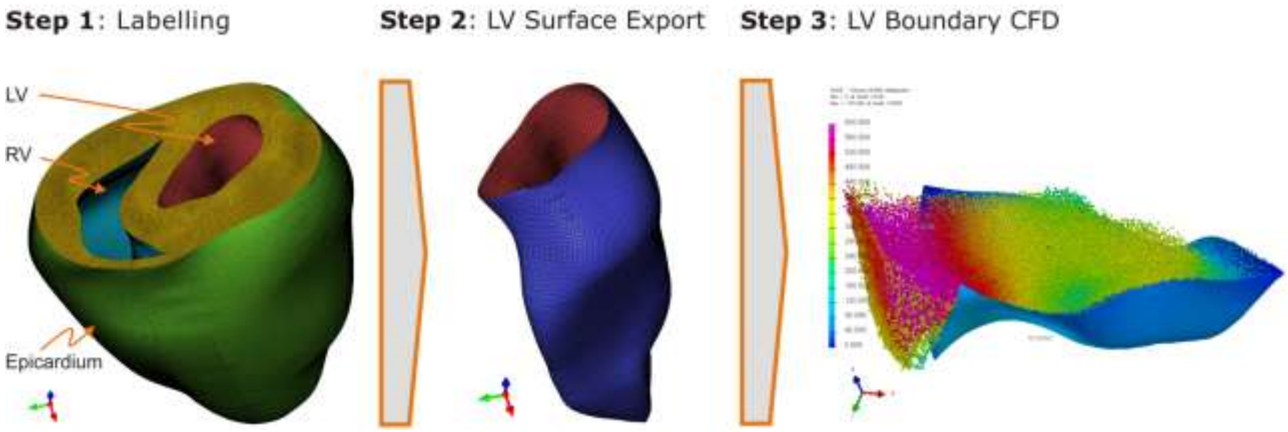

Figure 1 3 step process involved in providing output of electro-mechanical simulation results as boundary conditions to CFD solver tools.

Technical description of CARP output format

Each simulation run produces a 3D volumetric mesh and a time dependent displacement field. Meshes are stored in a simple text format which is briefly described below. Displacement data are stored in the binary igb format.

Conversion tool GlMeshConvert2VTK

A conversion tool has been developed at the MUG. The command line tool is written in standard C99.

Usage

The GlMeshConvert2VTK command is used as follows:

GlMeshConvert2VTK bflag /path2mesh/meshname

Input:

bflag: simple flag to indicate that input data are in ASCII (0) or binary (1) format. Note that this also toggles between ASCII and binary output.

meshname: path to location of mesh and basename of mesh file and displacement file. The code expects four files to be located under /path2meshname

- meshname.pts
- meshname.elem
- meshname.lon
- meshname.dynpt

In the case that a mesh is provided in binary format, the file extensions are bpts, belem, blon and bdynpt.

Output:

For each time slice stored in the dynpt file a separate file is written where each time slice is indexed using ascending integer values. That is for NT time slices in the meshname.dynpt file NT vtk files are written: meshname_0.vtk, meshname_1.vtk, …., meshname_(NT-1).vtk. Depending on the setting of bflag, vtk files are in ASCII or binary format.

Availability

For now the source code along with a binary and an ASCII test dataset can be downloaded from here: http://carp.medunigraz.at/~augustin/cardioproof/mesh/

CARP Mesh Format

A CARP mesh description consists of three components which are stored in individual files, a points file storing the x,y,z coordinates of mesh points, a connectivity file holding information on finite element type, nodal indices which span the element and material identifiers, and a tissue structure file which stores the Eigen axes (fiber and sheet orientation) within each element.

CARP nodal points file

Extension: pts

NN

x(0) y(0) z(0)

x(1) y(1) z(1)

...

x(NN-1) y(NN-1) z(NN-1)

where NN specifies the number of nodes in the file. Each tuple x,y,z specifies a nodal location in space, measured in µm. Indexing sticks to C conventions, that is, indices run from 0 to NN-1.

CARP connectivity file

Extension: elem

NE

Tt e(0)-n0 e(0)-n1 e(0)-n2 e(0)-n3 materialID (0)

Tt e(1)-n0 e(1)-n1 e(1)-n2 e(1)-n3 materialID (1)

…

Tt e(NE-1)-n0 e(NE-1)-n1 e(NE-1)-n2 e(NE-1)-n3 materialID (NE-1)

where NE is the number of elements in the mesh, Tt (Tetraeder) identifies the element type and e(I)-n(J) identifies the J-the node of element I.

CARP tissue structure file

Extension: lon

fx(0) fy(0) fz(0) sx(0) sy(0) sz(0)

fx(1) fy(1) fz(1) sx(1) sy(1) sz(1)

...

fx(NN-1) fy(NN-1) fz(NN-1) sx(NN-1) sy(NN-1) sz(NN-1)

where **f**=[fx(e) fy(e) fz(e)] and **s**=[sx(e) sy(e) sz(e)] are the unit eigenvectors of element e which are used to define the orthotropic Eigen axes of the material.

Binary IGB Format

Displacement data are stored in a binary igb format. This file consists of a header in ascii format which is 1024 bytes long. The body of the file stores deformation data in binary format where each coordinate is represented by a 4 Byte float value. first the time step (e.g. 0.000000) and then the current position of the nodes as in the points file. Note that the values are not the displacement but the (x,y,z)-coordinates of the points at the specific time step.

Target VTK Format

Mesh and displacement data are converted to a widely used standard format. It has been agreed upon that this target file format will be VTK (www.vtk.org) in either ascii or binary format. A VTK-file consists of five parts:

- a header describing the basic properties of the file
- a list of the (x,y,z) coordinates of all mesh points
- a list of all cells in the grid based on the point indices and a list of cell types
- an optional block of datasets based on the grid points
- an optional block of datasets based on the grid cells

For more detailed information on the structure of VTK files see www.vtk.org/VTK/img/file-formats.pdf.

Modified conversion tool GlMeshConvert2VTK (VPS version)

Introduction

The CFD code to be employed for the flow in the blood pool inside the ventricles and, later on, to limited parts of the aorta is the mesh-less SPH module from the VPS software product from ESI. The boundary conditions for the flow in the left ventricle required by VPS may best be imposed by providing the displacements of the surface mesh provided by the CARP output.

Since the VTK format is not accepted by the pre-processor (VISUAL) of VPS, the conversion tool from MUG had to be adapted. In addition a special code has been created to create an input file that may be read by VISUAL. This has been accomplished by ESI.

Usage/Input

The usage and input is identical to the original version, but is only available for the binary version.

Modifications

In addition to binay files in VTK format, formatted files are created for the (initial) nodal coordinates, the mesh connectivity and the displacements at each time step. These files are suitable to be read by special FORTRAN code genheart1 as described below.

Output

The following, formatted new files are created:

- 'nodes' with the number of nodes followed by the x(0), y(o), z(o) at each subsequent line
- 'mesh', with (three) node numbers for each element
- 'dynpts', with the number of time steps followed by the number of points and the displacements of each node, at each time step

Conversion tool Genheart1

Introduction

The files created by the modified GlMeshConvert2VTK tool are not immediately convenient for the creation of input files in VISUAL since in VPS displacement must be provided as functions of time, separate for x, y and z-direction, to be assigned to the appropriate nodes. In addition, the nodes and element must be provided with a positive number. For this reason a special-purpose FORTRAN code genheart1 has been written.

Input

The following files are required:

- 'fort.7' which is the 'nodes' file stripped from the first line appended by the 'mesh' file
- 'fort.8' which is the 'dynpts' file
- 'genheartest1.inp' a single line containing a dummy integer, the number of nodes, the number of elements, the offset for the node numbers, the offset for the element numbers, the part number and the number of time steps

The offset numbers may be useful in case this mesh has to be combined with other meshes; a positive part number is mandatory in order to assign properties to the elements.

Usage

genheartest1.x < genheartest1.inp

Output

The following, formatted files are created:

- Fort 6 – copy of genheartest1.inp and for selected time steps and a selected node the x-displacement and x(0) just for checking
- Fort 9 – nodal coordinates and shell elements for the entire mesh
- Fort 17 – DISPX directives to functions for all nodes
- Fort 18 – DISPY directives to functions for all nodes
- Fort 19 – DISPZ directives to functions for all nodes
- Fort 21 – Definitions of functions for DISPX

- Fort 22 – Definitions of functions for DISPY
- Fort 23 – Definitions of functions for DISPZ

These files (except fort.6) may be combined into the VPS input file for the mesh and displacements of the model.

Both codes take only a few minutes to run.