



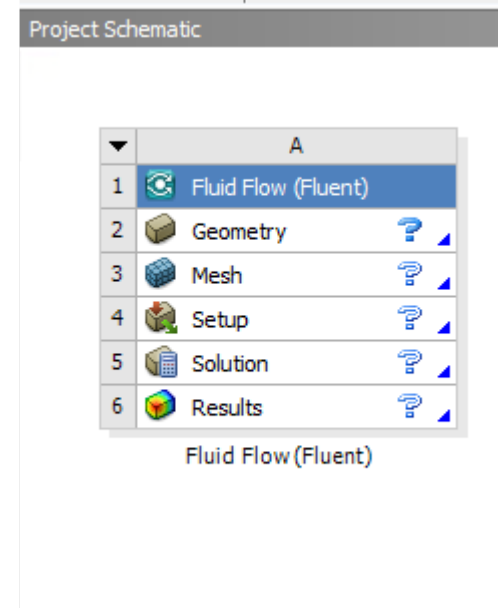
Perform moving mesh simulations based on 3D RTE as explained in Gruenwald et al. 2021

Prerequisites

- Visual Studio (<https://visualstudio.microsoft.com/en/downloads/>) full version with all packages
- Python and any IDE like Spyder or Sublime Text 3
 - Install: pip, numpy wheel, matplotlib, scipy
- UCD files (export of Tomtec ImageArena) for ventricular cavities at all time steps
- Ansys Workbench, SpaceClaim, Meshing, FLUENT, CFD-Post (Version 2021 R2)

Step 0: Workbench

- Unzip example project files from repository
- Load project file into Workbench



Step 1: Prepare STL files

- Execute ucd2stl MATLAB code to convert UCD to STL files
- Copy and rename existing STL files (of all time steps) into the folder STL
 - "ventricle_n" with n starting from 0 (End-systolic volume)
 - Folder STL is in the same location as code ps_detNPts.py (Fluent folder in project files)

Pat3_DS5ja_RV_KL_00.ucd
Pat3_DS5ja_RV_KL_01.ucd
Pat3_DS5ja_RV_KL_02.ucd
Pat3_DS5ja_RV_KL_03.ucd
Pat3_DS5ja_RV_KL_04.ucd
Pat3_DS5ja_RV_KL_05.ucd
Pat3_DS5ja_RV_KL_06.ucd
Pat3_DS5ja_RV_KL_07.ucd
Pat3_DS5ja_RV_KL_08.ucd
Pat3_DS5ja_RV_KL_09.ucd
Pat3_DS5ja_RV_KL_10.ucd
Pat3_DS5ja_RV_KL_11.ucd
Pat3_DS5ja_RV_KL_12.ucd
Pat3_DS5ja_RV_KL_13.ucd
Pat3_DS5ja_RV_KL_14.ucd
Pat3_DS5ja_RV_KL_15.ucd
Pat3_DS5ja_RV_KL_16.ucd
Pat3_DS5ja_RV_KL_17.ucd
Pat3_DS5ja_RV_KL_18.ucd
Pat3_DS5ja_RV_KL_19.ucd
Pat3_DS5ja_RV_KL_header

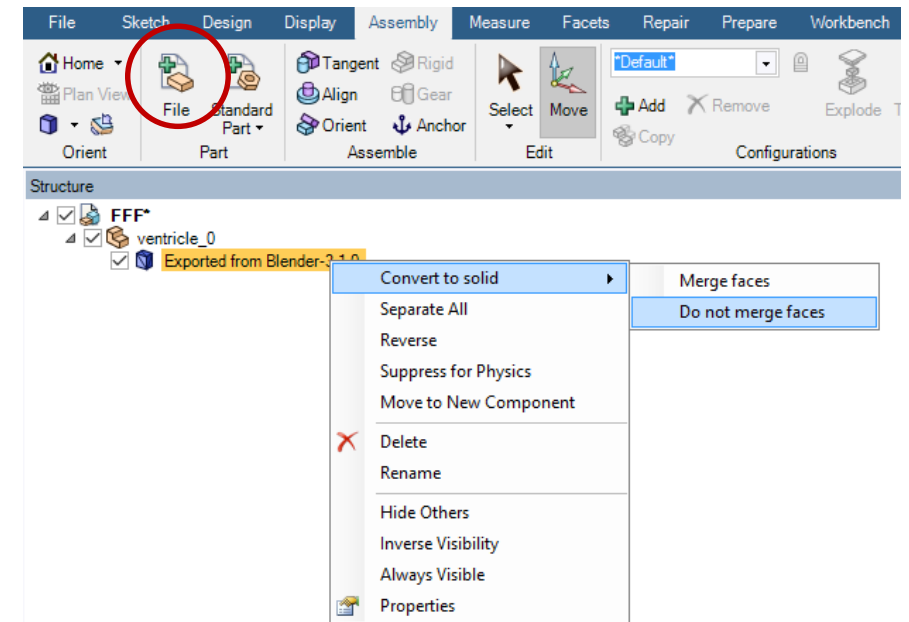
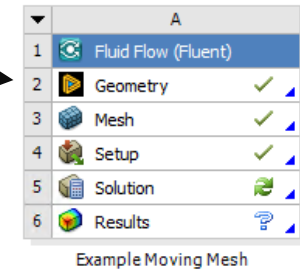


ventricle_0
ventricle_1
ventricle_2
ventricle_3
ventricle_4
ventricle_5
ventricle_6
ventricle_7
ventricle_8
ventricle_9
ventricle_10
ventricle_11
ventricle_12
ventricle_13
ventricle_14
ventricle_15
ventricle_16
ventricle_17
ventricle_18
ventricle_19

Step 2: SpaceClaim

- Open SpaceClaim through Workbench
- Import files:
 - Assembly → File → Choose only initial timeframe ('ventricle_0.stl')
 - Options → File Options → STL → Set Import units (not automatic)
- Convert to solid → Do not merge faces
- Close SpaceClaim

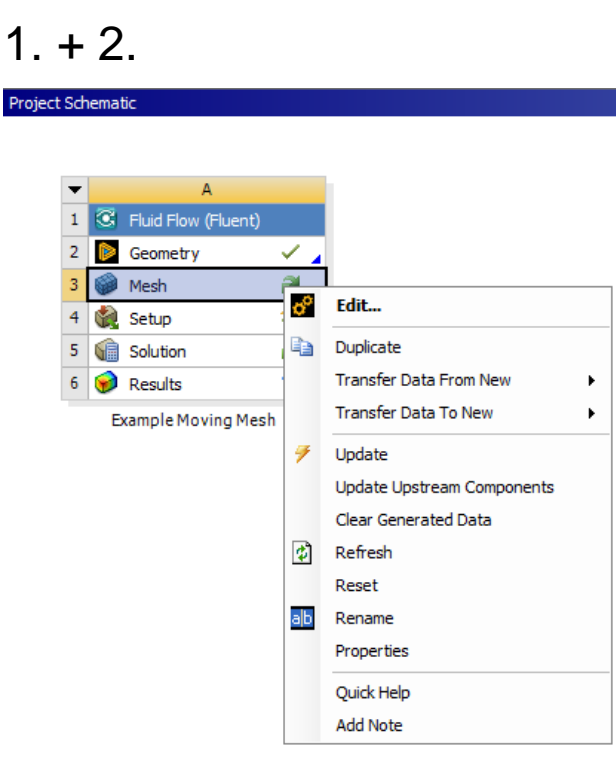
Project Schematic



Step 3: Meshing I

1. Update upstream Components
2. Edit Mesh
3. Select Body for meshing method
Customize settings of meshing method
4. Assign named selections
 1. Ventricle (Whole ventricle body)
 2. Inlet/MV (Faces of MV)
 3. Outlet/AV (Faces of AV)
 4. Prism layers (All faces except MV/AV)
5. Generate Mesh (Check mesh afterwards)
6. Close meshing
7. Update upstream components of Setup

1. + 2.

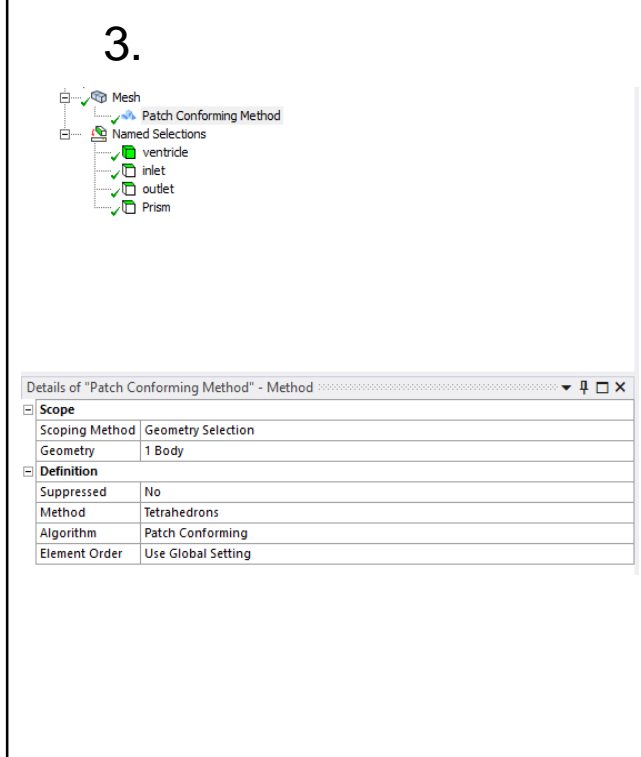


Project Schematic

1. Fluid Flow (Fluent)
2. Geometry
3. Mesh
4. Setup
5. Solution
6. Results

Example Moving Mesh

3.



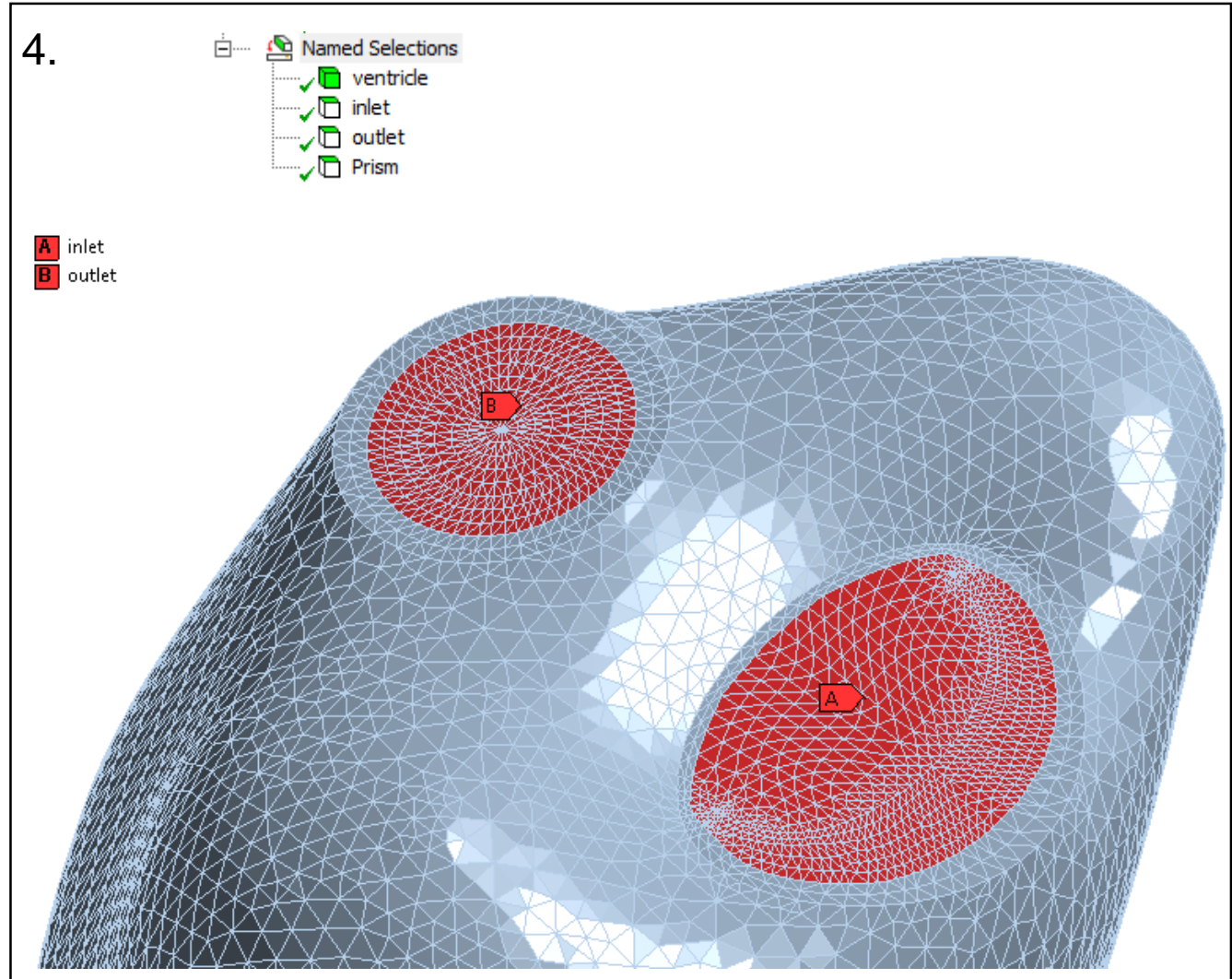
Mesh
Patch Conforming Method
Named Selections
ventricle
inlet
outlet
Prism

Details of "Patch Conforming Method" - Method

Scoping Method	Geometry Selection
Geometry	1 Body
Definition	
Suppressed	No
Method	Tetrahedrons
Algorithm	Patch Conforming
Element Order	Use Global Setting

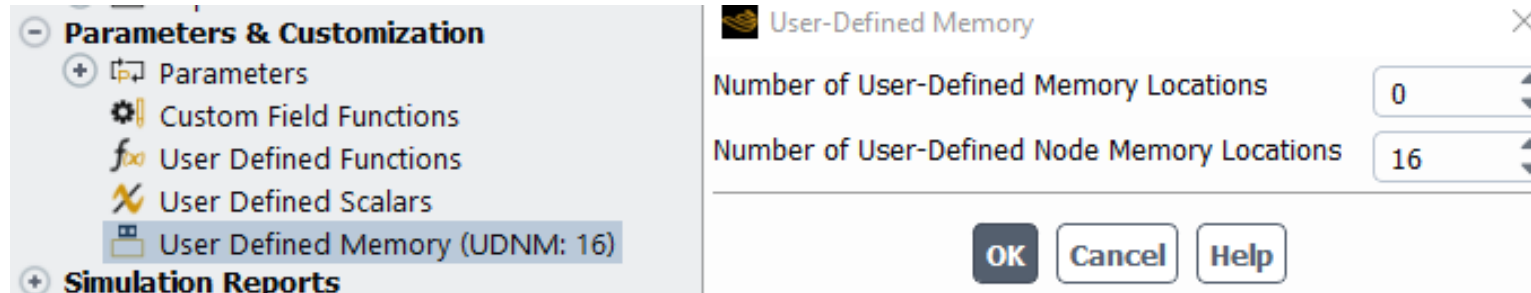
Step 3: Meshing II

1. Update upstream Components
2. Edit Mesh
3. Select Body for meshing method
Customize settings of meshing method
4. Assign named selections
 1. Ventricle (Whole ventricle body)
 2. Inlet/MV (Faces of MV)
 3. Outlet/AV (Faces of AV)
 4. Prism layers (All faces except MV/AV)
5. Generate Mesh (Check mesh afterwards)
6. Close meshing
7. Update upstream components of Setup



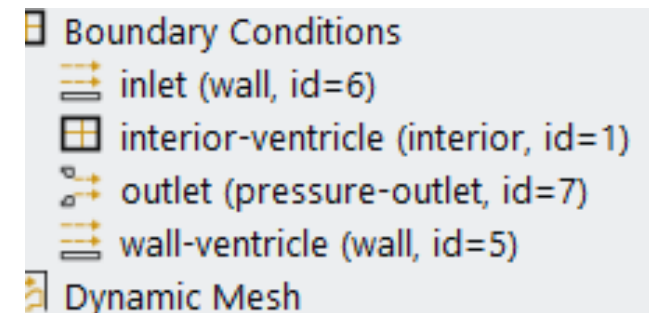
Step 4: FLUENT Setup solver nodes

- Start FLUENT with double precision and desired Solver processes (#Nodes)
- Change Solver processes (Parameters & Customization → User-defined memory)



- Check and adjust the zone IDs of the outer walls (here: Inlet, outlet and wall-ventricle) in the UDF (udf_exPts.c)

```
/* define zoneIDs of ventricle, inlet and outlet */  
#define nFZones 3  
#define ventricle 5  
#define inlet 6  
#define outlet 7
```



Step 5: FLUENT Setup up initial Boundaries

- Check Boundaries in Fluent
 - Initial state (End-systole)
 - Inlet abs. pressure (0 pa)
 - Outlet wall (closed)

Zone Name: inlet

Momentum Thermal Radiation Species DPM Multiphase Potential Structure UDS

Reference Frame: Absolute

Gauge Total Pressure [Pa]: 0

Supersonic/Initial Gauge Pressure [Pa]: 0

Direction Specification Method: Normal to Boundary

☐ Prevent Reverse Flow

Turbulence

Specification Method: Intensity and Viscosity Ratio

Turbulent Intensity [%]: 5

Turbulent Viscosity Ratio: 10

Apply Close Help

Boundary Conditions

- Inlet
 - inlet (pressure-inlet, id=6)
- Internal
 - interior-ventricle (interior, id=1)
- Wall
 - outlet (wall, id=7)
 - wall-ventricle (wall, id=5)

Zone Name: outlet

Adjacent Cell Zone: ventricle

Momentum Thermal Radiation Species DPM Multiphase UDS Potential Structure Ablation

Wall Motion

☒ Stationary Wall
☐ Moving Wall

Motion

☒ Relative to Adjacent Cell Zone

Shear Condition

☒ No Slip
☐ Specified Shear
☐ Specularity Coefficient
☐ Marangoni Stress

Wall Roughness

Roughness Models

☒ Standard
☐ High Roughness (Icing)

Sand-Grain Roughness

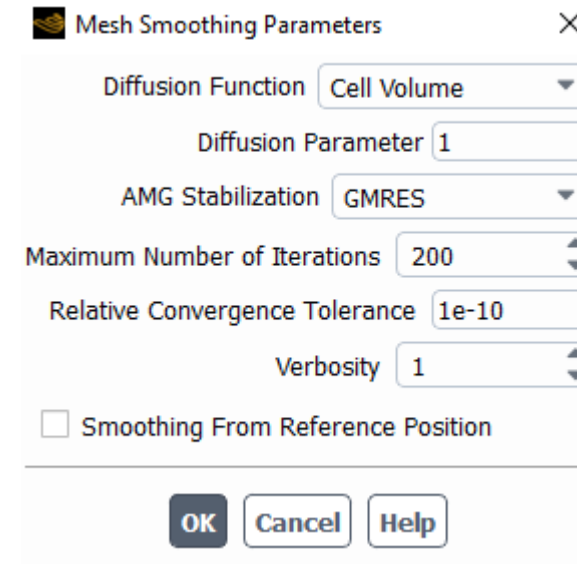
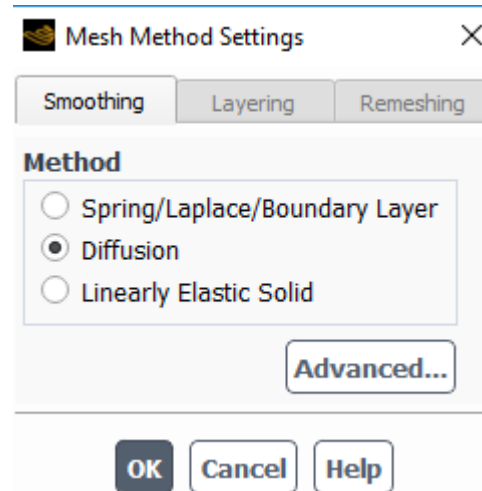
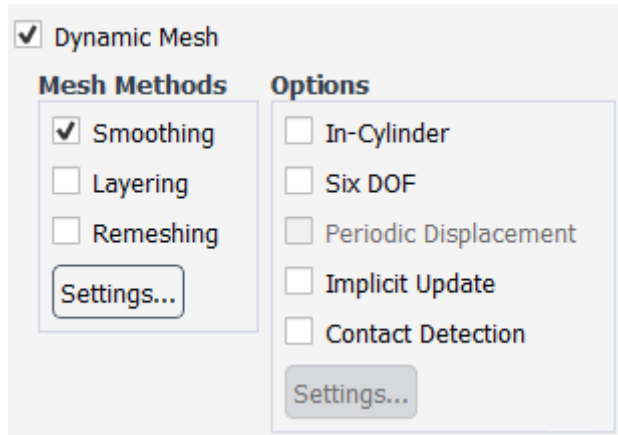
Roughness Height [m]: 0

Roughness Constant: 0.5

Apply Close Help

Step 6: FLUENT Setup up dynamic mesh settings

- Setup → Dynamic Mesh → Settings



Step 7: FLUENT Setup up dynamic Boundaries

- Setup dynamic mesh event settings
 - Setup → Dynamic Mesh → Events
 - 3 events to switch between diastole and systole (here diastole to systole)

- Inlet/Outlet to wall
- Outlet/Inlet to pressure
- Setup pressure profile

Command: define boundary-conditions pressure-outlet outlet y y n outpres_ao_3 pres_ao n y n n y 5 10 y n n n

- At end of cycle boundaries are not resetted to initial state
 - Inlet to pressure-inlet, Outlet to wall
 - Event – time adjusted to MV closing of next cycle

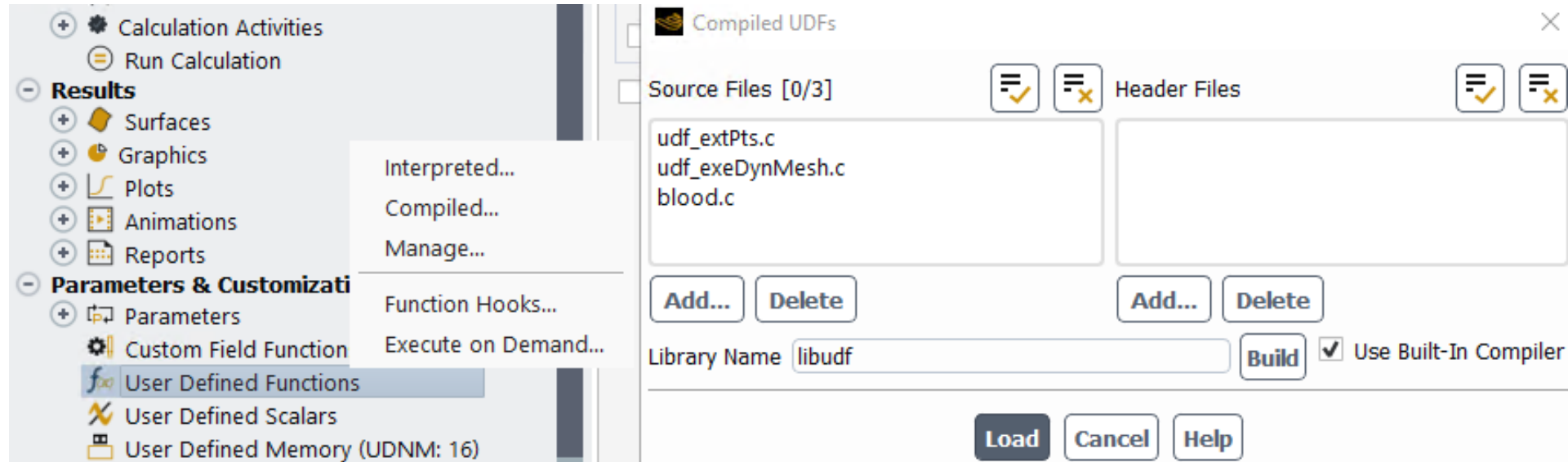
Number of Events 15

On	Name	At Time [s]	
<input checked="" type="checkbox"/>	inlet-to-wall	0.31892	Define...
<input checked="" type="checkbox"/>	wall-to-pressure	0.31892	Define...
<input checked="" type="checkbox"/>	pressure-outl	0.31892	Define...
<input checked="" type="checkbox"/>	outlet-to-wall	0.557	Define...
<input checked="" type="checkbox"/>	wall-to-pressure	0.557	Define...
<input checked="" type="checkbox"/>	pressure-intl	0.557	Define...

In this example: Time diastole: 0,31892s RR-duration: 0,557s

Step 8: FLUENT Compile UDF library

- Parameters & Customization → User Defined Functions → Compiled
- Add... (blood.c; udf_exeDynMesh.c; edf_exPts.c)
- Use Built-In Compiler
- Build
- Load



Step 9: FLUENT Execute udf_extPts.c

- Parameters & Customization → User Defined Functions → Execute on Demand...
 - First_AssignID::libudf
 - File „surface“ is created in the simulation folder
 - Number of nodes for each core are shown in the console (Will be necessary for udf_exeDynMesh)

surface file

1	0	6.485890	16.681522	28.130762
2	1	6.862917	16.265652	28.036011
3	2	6.569711	15.943861	27.625748
4	3	7.079221	15.724083	27.819656
5	4	7.156017	16.587324	28.446129
6	5	7.513730	15.881123	28.214535
7	6	7.481522	16.395054	28.535452
8	7	7.545926	15.367387	27.893742
9	8	7.887307	16.074398	28.607815
10	9	7.446888	16.688888	28.888888

Step 10: FLUENT Execute ps_detNpts.py → STL to PTS I

- With python e.g. in Sublime Text 3 or Spyder
- If first execution: Answer Do you need to generate correlation file? Yes: 1. No: 0. with “1”
- The reference frame ID is the number of the STL-file in the folder STL, that has been loaded and meshed with ANSYS (see step 1).
- Output:
 - Number of Faces
 - allPtsNum

Step 11: FLUENT Execute ps_detNPts.py → STL to PTS II

- First timepoint of cycle as Start FrameID (commonly 0)
- Last used timepoint as End FrameID
- Code creates "surface_" files in a folder PTS

surface_0
surface_1
surface_2
surface_3
surface_4
surface_5
surface_6
surface_7
surface_8
surface_9
surface_10
surface_11
surface_12
surface_13
surface_14
surface_15

Step 12: FLUENT Execute ps_intNpts.py → PTS to UDFPTS

- Execute the ps_intNpts code in (same location as PTS folder / fluent folder)
- Set number of stl files per cycle as „Total number of frames in a cycle”
- Set number of interpolation steps between each stl „Number of intermediate frames“
→ This will result in the number of timesteps each cycle
(Number of intermediate frames x number of stl files)
- Choose a temporal interpolation method between the frames
- Output of Total Number of Frames after interpolation: # of timesteps for one cycle
- Code creates udfsurface.asc files in folder UDFPTS → necessary for mesh movement

Step 13: FLUENT Adjust and execute udf_exeDynMesh.c

- Unload libudf
- Delete libudf-folder
- Adjust udf exeDynMesh
 - nTimeSteps: Number of timesteps= maximum index of udf_surface (output of ps_intNPTs.py)
 - allPtsNum: Number of all nodes, output of ps_detNPTs.py
 - PtsNum[]: number of points per core (saved as array); Console output after execution of UDF
First_Assign_ID
- Compile UDF (see slide 12)
- Parameters & Customization → User Defined Functions → Execute on Demand...
 - loadMesh::libudf

Step 14: Setup use of udf_exeDynMesh.c

- Dynamic Mesh → Dynamic Mesh zones → create/edit → user defined function for inlet, outlet, wall-ventricle erstellen

The screenshot shows the 'Dynamic Mesh Zones' panel in ANSYS Fluent. The 'Zone Names' dropdown is set to 'inlet'. The 'Type' section has 'User-Defined' selected. The 'Dynamic Mesh Zones' list on the right contains 'inlet', 'outlet', and 'wall-ventricle'. The 'Motion Attributes' tab is active, showing 'Mesh Motion UDF' set to 'Ven_move::libudf'. The 'Motion Options' section has 'Relative Motion' unchecked and 'Relative Zone' set to a default value. The 'Exclude Mesh Motion in Boundary Conditions' checkbox is also unchecked. At the bottom, there are buttons for 'Create', 'Draw', 'Delete All', 'Delete', 'Close', and 'Help'.

Zone Names
inlet

Type
☐ Stationary
☐ Rigid Body
☐ Deforming
☒ User-Defined
☐ System Coupling

Dynamic Mesh Zones
inlet
outlet
wall-ventricle

Motion Attributes | Geometry Definition | Meshing Options | Solver Options

Mesh Motion UDF
Ven_move::libudf

Motion Options
☐ Relative Motion
Relative Zone
[Dropdown]

☐ Exclude Mesh Motion in Boundary Conditions

Create Draw Delete All Delete Close Help

Step 15: Prohibit repartitioning

- Parallel → Partition/Load Balance → Check off Dynamic Mesh

The screenshot shows a software interface with four tabs: 'Options', 'Optimization', 'Weighting', and 'Dynamic Load Balancing'. The 'Dynamic Load Balancing' tab is selected. Below the tabs, there are three rows of settings. The first row is for 'Physical Models' with a checkbox, a 'Threshold (%)' field set to 10, and an 'Interval' dropdown set to 10. The second row is for 'Dynamic Mesh' with a checkbox, a checked 'Auto' checkbox, a 'Threshold (%)' field set to 20, and an 'Interval' dropdown set to 10. The third row is for 'Mesh Adaption' with a checkbox and a 'Threshold (%)' field set to 5. A vertical scrollbar is visible on the right side of the settings area.

	Threshold (%)	Interval
<input type="checkbox"/> Physical Models	10	10
<input type="checkbox"/> Dynamic Mesh <input checked="" type="checkbox"/> Auto	20	10
<input type="checkbox"/> Mesh Adaption	5	

Step 16: FLUENT Initialize

- Initialization → Initialize with Hybrid initialization (standard initialization also possible)
- If not the first timestep was put out in console after loadMesh::libudf

```
UDFPTS/udfsurface_1.asc  
  
0  
Finish Loading
```

- Repeat execution of loadMesh::libudf (see slide 18)
- Initialization → Initialize

Step 17: Run Simulation

- Optimally before simulation start → preview mesh motion
 - Preview Mesh Motion
- If mesh movement doesn't work (e.g. negative cell volumes):
 - Check if mesh node-connectivity is consistent
 - Recompute mesh nodes (steps 2-12)
 - (Settings: Dynamic Mesh → Mesh Methods → Remeshing with default settings)
- Run simulation (Solution → Run Calculation → Calculate)

Contact

Michael Neidlin

neidlin@ame.rwth-aachen.de or

michael.neidlin@gmail.com