



**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

## Step 0: Use ucd2stl MATLAB code to convert UCD to STL 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

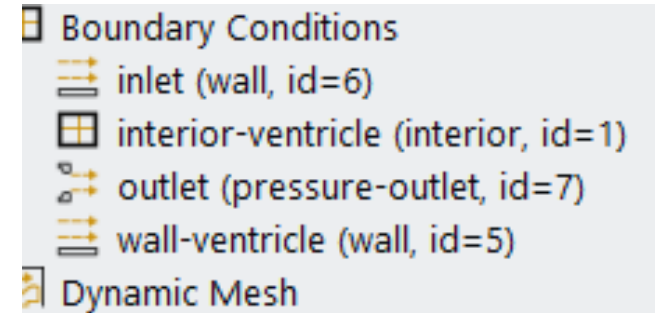
## Step 1: Load one timestep and mesh in ANSYS

---

- Can be done through conversion of STL to stp (with Creo or Solidworks)
- Other approaches (Spaceclaim) also possible
- No specific requirements for meshing

## Step 2: Use UDF udf\_extPts.c through ANSYS Fluent

- Open FLUENT: Node Memory is used -> user defined -> memory -> turn on node Memory and adjust to number of used cores
- Check and adjust the zone IDs of the outer walls (here: Inlet, outlet and wall-ventricle) in the UDF (if necessary)



```
/* Define zones of ventricle, inlet and outlet*/  
#define ventricle 5  
#define inlet 6  
#define outlet 7
```

## Step 2: Use UDF udf\_extPts.c through ANSYS Fluent

---

- Use of UDF in ANSYS Fluent
  - User defined -> functions -> compiled -> choose udf\_extPts
  - User defined -> execute on demand -> First\_AssignID
  - 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)

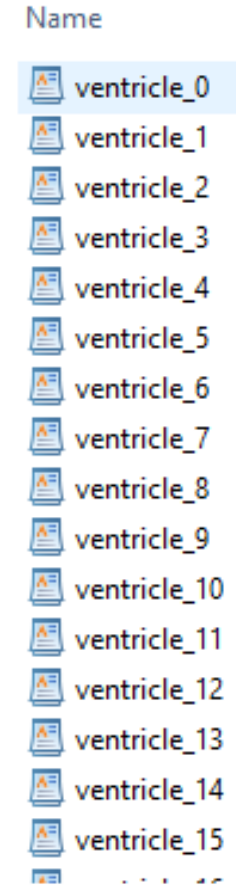
**surface file**

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 3: Create STL folder

---

- Copy and rename existing STL files (of all time steps) into the folder STL
  - "ventricle\_n" with n starting from 0
  - Folder STL is in the same location as code ps\_detNPs.py



## Step 4: Execute ps\_detNPts.py → STL to PTS

---

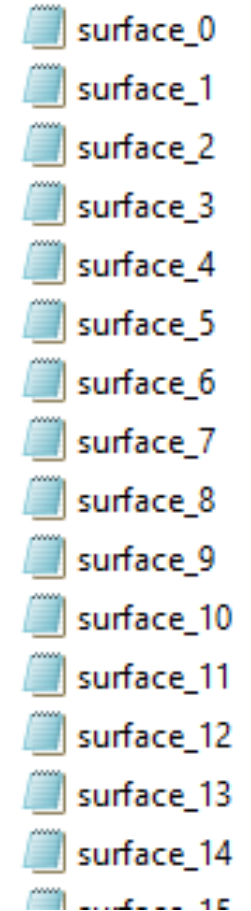
- With python e.g. in Sublime Text 3 or Spyder
- If first execution: Answer Do you need to generate correlation file? Yes: 1. No: 0. with “1” beantwortet
- The reference frame ID is the number of the STL-file in the folder STL, that has been loaded and meshed with ANSYS (see step 1).
- Output:
  - Number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



## Step 4: Execute ps\_detNPts.py → STL to PTS

---

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



## Step 5: Execute ps\_intNpts.py → PTS to UDFPTS

---

- Execute the ps\_intNpts code in (same location as PTS folder)
- Set number of stl files per cycle as „Total number of frames in a cycle”
- Set number of interpolation steps (timesteps from simulation) „Number of intermediate frames“
- 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 6: Adjust udf\_exeDynMesh.c

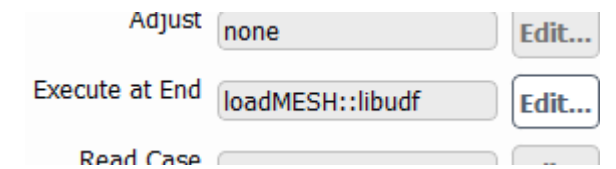
---

- Before execution several parameters in „dynMesh“ UDF have to be set:
  - 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 (see slide 6)

## Step 6: Execute udf\_exeDynMesh.c

---

- User defined -> functions -> compiled -> choose dynMesh
- User defined -> functions -> function hooks -> execute at end -> add loadMesh
- Put folder UDFPTS into folder „Fluent“ of the simulation (same location of the „surface“ file, slide 6)
- User defined -> functions -> execute on demand -> First\_load Mesh



# If compiling doesn't work

1. Compute locally(Desktop)

2. Compiling of UDFs

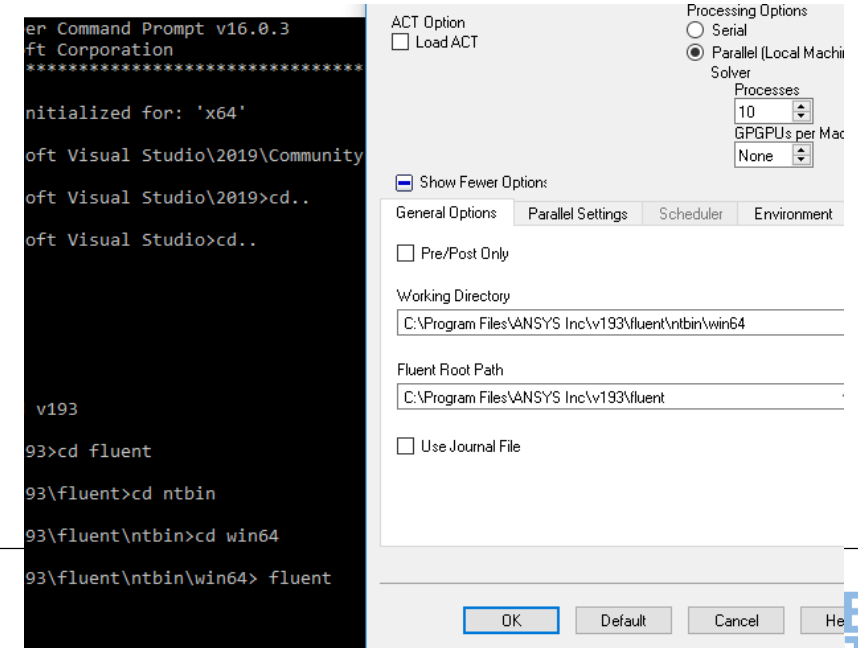
compiling with Native visual command prompt

Go to path and open fluent: enter fluent + ENTER

In launcher: set core numbers

Working directory where simulation has been saved

```
x64 Native Tools Command Prompt for VS 2019
** Visual Studio 2019 Developer Command Prompt v16.0.3
** Copyright (c) 2019 Microsoft Corporation
*****
[vcvarsall.bat] Environment initialized for: 'x64'
C:\Program Files (x86)\Microsoft Visual Studio\2019\Community>cd..
C:\Program Files (x86)\Microsoft Visual Studio\2019>cd..
C:\Program Files (x86)\Microsoft Visual Studio>cd..
C:\Program Files (x86)>cd..
C:\> cd program files
C:\Program Files>cd ansys inc
C:\Program Files\ANSYS Inc>cd v193
C:\Program Files\ANSYS Inc\v193>cd fluent
C:\Program Files\ANSYS Inc\v193\fluent>cd ntbin
C:\Program Files\ANSYS Inc\v193\fluent\ntbin>cd win64
C:\Program Files\ANSYS Inc\v193\fluent\ntbin\win64>
```



## Use udf\_exeDynMesh.c

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

The screenshot shows the 'Dynamic Mesh Zones' dialog box 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 'Mesh Motion UDF' dropdown is 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

# Dynamic Mesh Settings

☒ Dynamic Mesh

Mesh Methods	Options
<input checked="" type="checkbox"/> Smoothing	<input type="checkbox"/> In-Cylinder
<input checked="" type="checkbox"/> Layering	<input type="checkbox"/> Six DOF
<input type="checkbox"/> Remeshing	<input type="checkbox"/> Implicit Update
<button>Settings...</button>	<input type="checkbox"/> Contact Detection
	<button>Settings...</button>

Smoothing Layering Remeshing

**Method**

☐ Spring/Laplace/Boundary Layer  
☒ Diffusion  
☐ Linearly Elastic Solid

**Parameters**

Spring Constant Factor   
Convergence Tolerance   
Number of Iterations

**Elements**

☒ Tet in Tet Zones  
☐ Tet in Mixed Zones  
☐ All

Laplace Node Relaxation   
Diffusion Function   
Diffusion Parameter   
Poisson's Ratio

Smoothing Layering Remeshing

**Options**

☐ Height Based  
☒ Ratio Based

Split Factor   
Collapse Factor

# Prohibit repartitioning

---

- Parallel -> Partition/Load Balance ->

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 has a checkbox for 'Physical Models' (unchecked), a 'Threshold (%)' input field with the value '10', and an 'Interval' spinner field with the value '10'. The second row has a checkbox for 'Dynamic Mesh' (unchecked), a checked 'Auto' checkbox, a 'Threshold (%)' input field with the value '20', and an 'Interval' spinner field with the value '10'. The third row has a checkbox for 'Mesh Adaption' (unchecked) and a 'Threshold (%)' input field with the value '5'. There are vertical scroll bars on the right side of the panel.

	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	



## Mesh motion

---

- Before simulation start → preview mesh motion
  - Preview Mesh Motion
- If mesh movement doesn't work (e.g. negative cell volumes):
  - Fluent in serial model → Neustart Fluent → Fluent Launcher **serial**
  - Recompute mesh nodes (steps 2 -5)
  - Settings: Dynamic Mesh → Mesh Methods → Remeshing with default settings

## Valve opening/closing

---

- In „Dynamic Mesh“ → „Events“ set three events
  - Inlet to wall
  - Outlet to pressure-outlet
  - Profile to pressure-outlet
- Initialise simulation, then firstloadmesh, then simulation starten
- At end of cycle NO initialisation, but
  - Inlet to pressure-inlet, Outlet to wall
  - Event – time adjusted to MV closing of next cycle
- start w/o initialisation

## Contact

---

Michael Neidlin

[neidlin@ame.rwth-aachen.de](mailto:neidlin@ame.rwth-aachen.de) or

[michael.neidlin@gmail.com](mailto:michael.neidlin@gmail.com)