Horizon 2020

Reduced Order Modelling, Simulation and Optimization of Coupled systems

Validation of Fluid-Structure Interaction Simulations in Membrane-Based Blood Pumps

Deliverable number: D5.3

Version 0.1

Funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 765374

Project Acronym: ROMSOC
Project Full Title: Reduced Order Modelling, Simulation and Optimization of Coupled systems
Call: H2020-MSCA-ITN-2017
Topic: Innovative Training Network
Type of Action: European Industrial Doctorates
Grant Number: 765374

Editor:	Andrés Prieto, ITMATI
Deliverable nature:	Report (R)
Dissemination level:	Public (PU)
Contractual Delivery Date:	01/07/2021
Actual Delivery Date	01/07/2021
Number of pages:	9
Keywords:	Benchmark, Fluid-Structure Interaction, Blood Pumps, Model Validation, HQ curves
Authors:	**Marco Martinolli**, MOX, Dipartimento di Matematica, PoliMi **Christian Vergara**, LaBS, Dipartimento di Chimica, Materiali e Ingegneria Chimica, PoliMi **Luc Polverelli**, CorWave Inc.
Peer review:	ITMATI

Abstract

the proposed benchmark can be used for training in the fields of mathematical modeling of a coupled system, model testing and error estimation.

Disclaimer & acknowledgment
This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 765374.

Contents

List of Acronyms

ITMATI	Technological Institute of Industrial Mathematics
FSI	Fluid-Structure Interaction
ITMATI	Technological Institute of Industrial Mathematics
LVAD	Left Ventricular Assist Devices
OP	Operating Point
XFEM	Extended Finite Element Method
WMBP	Wave Membrane Blood Pumps

1. Introduction

Wave Membrane Blood Pumps (WMBP) [1], developed at CorWave SA, may represent the new frontier of Left Ventricular Assist Devices (LVAD). The pumping technnology of WMBP is based on the wave undulations of an immersed elastic membrane that propels the blood against an adverse pressure gradient, from the left ventricle into the ascending aorta. In Figure 1, left, we show the cross sectional view of the pump device.

The intertwined dynamics in WMBP are studied in the framework of Fluid-Structure Interaction (FSI) modeling and solved in the mathematical domain shown in Figure 1, right. Blood is modeled as a viscous incompressible Newtonian fluid with density ρ_f and viscosity μ_f, by means of Navier-Stokes Equations. The wave membrane Ω_1^s is considered a linear material at small deformations, according with previous publication [2], with density ρ_s^1 and Lamé parameters λ_s^1 and μ_s^1. The same holds for the magnet ring Ω_2^s, with parameters ρ_s^2, λ_s^2 and μ_s^2. For sake of simplicity, we combine structure properties in unique space-dependante variables, e.g. $\tilde{\rho}_s(\mathbf{x}) = \rho_s^1$ if $\mathbf{x} \in \Omega_1^s$, and $\tilde{\rho}_s(\mathbf{x}) = \rho_s^2$ if $\mathbf{x} \in \Omega_2^s$. Hence, the formulation of the problem is the following: for each time $t > 0$, find fluid velocity and pressure $(\mathbf{u}(t), p(t))$ in the fluid domain $\Omega^f(t)$ and structures displacement $\widehat{\mathbf{d}}(t)$ in the reference structure domain $\widehat{\Omega}^s = \Omega^s(0) = \Omega_1^s(0) \cup \Omega_2^s(0)$, such that:

$$\rho_f \left(\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u}\right) - \nabla \cdot \mathbf{T}^f(\mathbf{u}, p) = \mathbf{0} \qquad \text{in } \Omega^f(\mathbf{d}), \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega^f(\mathbf{d}), \tag{2}$$

$$\tilde{\rho}_s \partial_{tt} \widehat{\mathbf{d}} - \nabla \cdot \widehat{\mathbf{T}}^s(\widehat{\mathbf{d}}) = \mathbf{0} \qquad \text{in } \widehat{\Omega}^s, \tag{3}$$

$$\mathbf{u} = \partial_t \mathbf{d} \qquad \text{on } \Sigma(\mathbf{d}), \tag{4}$$

$$\mathbf{T}^f(\mathbf{u}, p)\, \mathbf{n} = \mathbf{T}^s(\mathbf{d})\, \mathbf{n} \qquad \text{on } \Sigma(\mathbf{d}). \tag{5}$$

where $\mathbf{T}^f(\mathbf{u}, p) = -p\mathbf{I} + 2\mu_f \mathbf{D}(\mathbf{u})$, with $\mathbf{D}(\mathbf{u}) = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T)$, is the fluid Cauchy stress tensor and $\widehat{\mathbf{T}}^s(\widehat{\mathbf{d}}) = \tilde{\lambda}_s \left(\widehat{\nabla} \cdot \widehat{\mathbf{d}}\right) \mathbf{I} + 2\tilde{\mu}_s \widehat{\mathbf{D}}(\widehat{\mathbf{d}})$ is first Piola-Kirchhoff structure tensor. Notice that the fluid domain Ω_f and the interface Σ depend over time on structure displacement $\mathbf{d}$ (geometric coupling). Physical coupling conditions (4) and (5) guarantee the continuity of velocity and of stresses at the fluid-structure interface Σ, respectively.

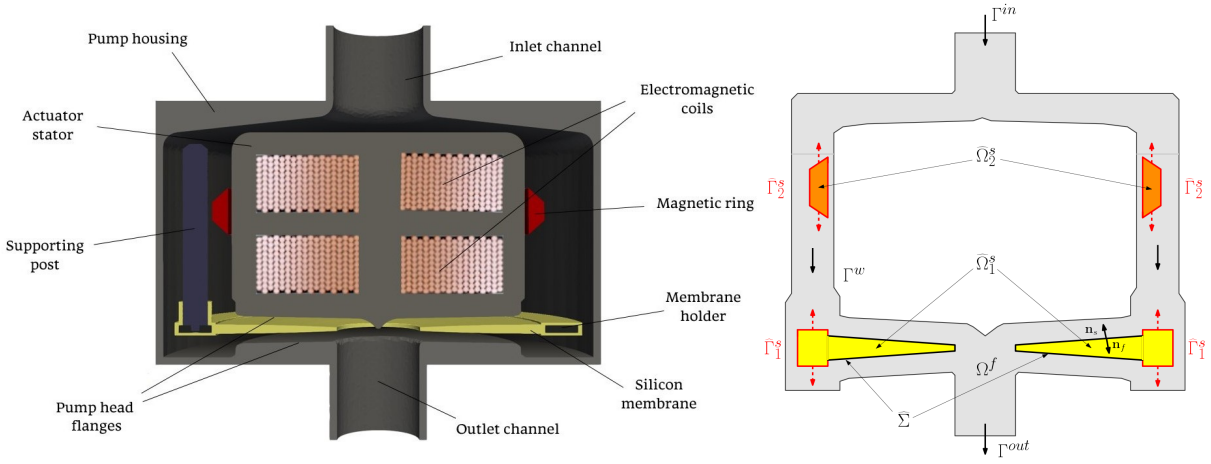


Figure 1: Left: Cross section of wave membrane blood pumps. The blood enters from the inlet channel, it flows down along the sides of the central actuator body, it interacts with the wave silicon membrane and it is finally ejected into the outlet channel. Membrane vibrations are triggered by the oscillations of the magnet ring. Right: Representation of the mathematical domain.

The boundary conditions define the operating point of WMBP. For the fluid sub-problem, the hydraulic resistance in the pump, that is the head pressue H, is prescribed by means of a pair of Neumann conditions at the

inlet Γ^{in} and at the outlet Γ^{out}; while non-slip wall conditions are imposed at boundary Γ^{w}:

$$\mathbf{T}^f(\mathbf{u},p)\,\mathbf{n}^f = \mathbf{0} \qquad \text{on } \Gamma^{in}, \tag{6}$$

$$\mathbf{T}^f(\mathbf{u},p)\,\mathbf{n}^f = H\,\mathbf{n}^f \qquad \text{on } \Gamma^{out}, \tag{7}$$

$$\mathbf{u} = \mathbf{0} \qquad \text{on } \Gamma^{w}, \tag{8}$$

The progressive undulations in the membrane are caused by the oscillations of the magnet ring with boundary Γ^s_2; then, they are transmitted to the external edge of the membrane Γ^s_1. Such oscillations are assumed to be purely sinusoidal with frequency f and amplitude Φ. Thus, the oscillations are prescribed by means of a Dirichlet condition on boundaries Γ^s_i $(i = 1, 2)$:

$$\mathbf{d}\,(t) = \Phi\,\sin\left(2\pi\,f\,t\right)\,\mathbf{e}_z \qquad \text{on } \Gamma^s \tag{9}$$

The benchmark consists in the validation of a numerical method to solve the FSI problem in WMBP against real hydraulic data provided by CorWave SA. Specifically, the goal is to predict the outflow volume rate Q^{sim} given a certain operating point of the pump (H, f, Φ) and compare the numerical result with the measured flow data Q^{data}.

In this report, we will describe the steps to take to solve the FSI problem with our numerical strategy, based on the Extended Finite Element Method (XFEM) [3, 4]. XFEM is an unfitted technique which has two main advantages compared with other approaches for FSI problems: i) since the fluid mesh is kept fixed on the background, it avoids the remeshing procedure normally occurring in case of element distortion; ii) the accuracy of the solution is maintained at the interface, thanks to the local enrichment of the functional space of the extended finite elements. However, since the structure mesh moves in the foreground cutting the underlying fluid mesh, XFEM requires to compute the mesh intersections at each time istant to identify the fluid elements that are cut in multiple subportions (called *split elements*), leading to a higher computational cost. For more details on the numerical formulation of the XFEM with a DG mortaring at the interface, the reader can find an exaustive explanation in the reference papers [5, 6].

2. Input data

Input data to run the benchmark consist of four different types of files:

1. **meshes**: a folder that contains the fluid mesh, the membrane mesh and the magnet mesh;
2. **dataFile**: it details the list of meshes, physical parameters, stability parameters, operating parameters, time settings and other numerical parameters;
3. **solverFile**: it contains the parameters for the linear solver;
4. **validation data**: a csv datafile that collects the experimental data to be used to validate the numerical results

2.1. Meshes

For this benchmark, we consider the flat membrane pump geometry, studied in [5]. The CAD geometry files have been provided by the partner company CorWave SA. The unfitted meshes, reported in Figure 2, have been created using GMESH. The unfitted meshes selected for this benchmark showed positive convergence results. Lengths are expressed in cm.

2.2. DataFile

The dataFile presents the following information:

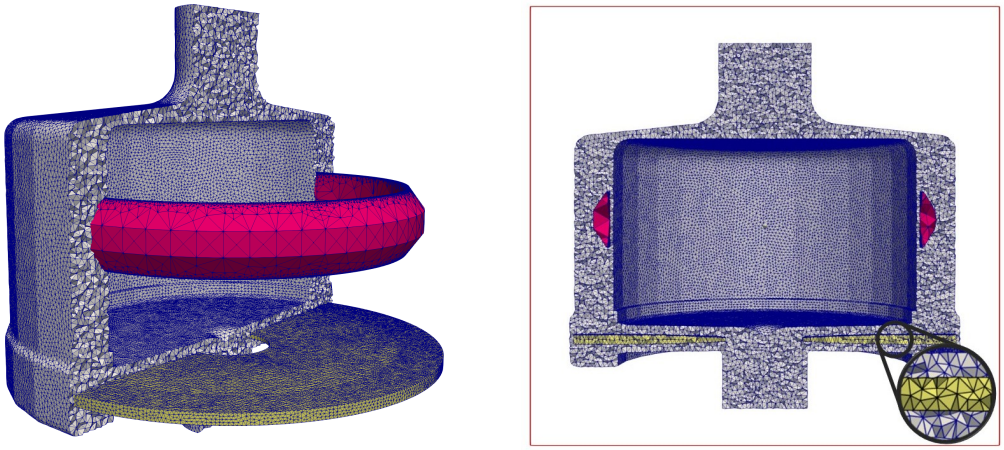


Figure 2: Prospective visualization and section of the fluid mesh (gray), the membrane mesh (yellow) and magnet mesh (red).

- strings of the mesh files:
  - fluid.mesh ($1.2M$ elements),
  - membrane.mesh ($280k$ elements),
  - magnet.mesh ($50k$ elements);
- fluid properties:
  - density $\rho_f = 1\ \frac{g}{cm^3}$,
  - viscosity $\mu_f = 0.035\ \frac{dyne}{cm^2}$;
- membrane properties:
  - density $\rho_s = 1.125\ \frac{g}{cm^3}$,
  - Young modulus $E_s = 1.686 \cdot 10^7\ \frac{dyne}{cm^2}$,
  - Poisson ration $\nu_s = 0.49$;
- magnet ring properties:
  - density $\rho_r = 7.85\ \frac{g}{cm^3}$,
  - Young modulus $E_r = 2.05 \cdot 10^{12}\ \frac{dyne}{cm^2}$,
  - Poisson ratio $\nu_r = 0.28$;
- Operating Point (OP) parameters:
  - head pressure $H = 50\ mmHg$ (default),
  - frequency $f = 120\ Hz$ (fixed),
  - amplitude $\Phi = 0.053\ cm$ (fixed);
- Continuous Interior Penalty penalty parameters for stability:
  - $\gamma_{v1} = 0.05$ (default),
  - $\gamma_{v2} = 0.5$ (default),
  - $\gamma_p = 0.05$ (default);
- time settings:
  - initial time $t_0 = 0\ s$,
  - final time $T = 0.025\ s$,
  - time step $\Delta t = 0.0002\ s$;

- preconditioner settings (fixed);
- string of the solverFile: solverFile.xml (default).

The experimental data for the validation are provided only for oscillation frequency of 120 Hz and amplitude Φ of 0.053 cm. Hence they should not be changed. Also notice that pressure conditions can largely affect the pump dynamics and consequently the stability requirements. For this reason, we suggest the user to set the head pressure parameter H and the CIP penalty parameters according to Table 1. All parameters are expressed in the unit system cgs.

	$H = 50$ mmHg	$H = 55$ mmHg	$H = 60$ mmHg
γ_{v1}	0.05	0.05	0.5
γ_{v2}	0.5	0.5	5
γ_p	0.05	0.1	0.1

Table 1: Continuous Interior Penalty parameters for different head pressure conditions.

2.3. SolverFile

The solverFile, reported in Appendix A.1, collects some numerical settings for the solver of the linear system. In general, such parameters should be kept fixed for parallel runs with more than 3 processors (10 cores are suggested to reduce computational cost). In case of serial runs or less than 3 processors, the user has to reduce the dimension of Krylov space (kspace) to 200 and of the number of maximum iterations (max_iter) to 500. Hence, in this case use file *SolverFile_serial.xml*.

2.4. Validation data

The experimental data used for the benchmark are *HQ curves* (H head pressure, Q flow rate), which are standard representations of pump hydraulic performance. They are obtained during the *in vitro* testings in a pump characterization bench, measuring the pressure difference H between the outlet and the inlet (called head pressure), and the corresponding flow rate Q generated at the outlet. Figure 3 shows the HQ curve of the pump system when the frequency of oscillation of the membrane is fixed to 120 Hz and the amplitude of oscillation to 0.053 cm. Raw data are provided in MATLAB file *HQ_data.mat*, consisting of two separate data columns with the measurements of head pressure in mmHg (1 mmHg = 1333.22 g /cm s^2) and of the outflow volume rate in liters per minute, or lpm (1 lpm = 0.06 cm^3/s).

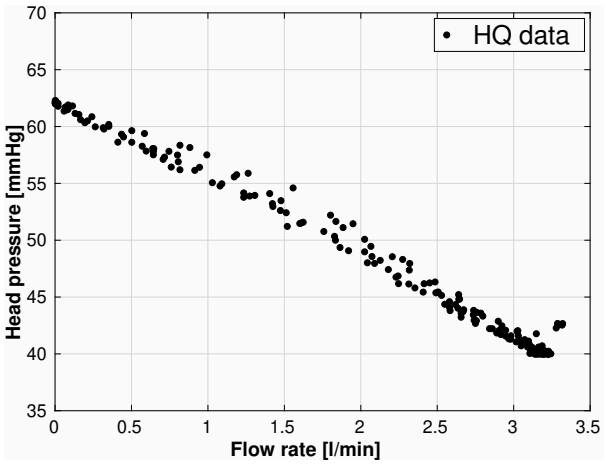


Figure 3: HQ experimental curve at oscillation frequency $f = 120$ Hz and amplitude $\Phi = 0.053$ cm.

3. Output data

The output of interest for this benchmark is the flow volume rate at the pump outlet Γ^{out}. Indeed, the goal is to compare, for a certain input head pressure H, the corresponding estimated outflow rate Q^{sim} with the experimental data Q^{data} extracted from the HQ curves (see Figure 3). At each time iteration, the software prints out the outflow computed at that time istant. Since this quantity oscillates in time with frequency f, we need to compute the average in time of the outflow results at regime, i.e. during the last period of oscillation τ. Thus:

$$Q^{sim} = \int_{T-\tau}^{T} \int_{\Gamma^{out}} \mathbf{u}(\mathbf{x}, t) \cdot \mathbf{n} \, d\mathbf{x} \, dt \tag{10}$$

Finally, a MATLAB code can be used to plot the HQ curve and the obtained numerical results Q^{sim} to check their proximity and compute absolute and relative errors. The user has just to type in the simulated head pressure H and the corresponding predicted mean flow Q^{sim}. In Table 2 we show the results expected for this benchmark for certain head pressure conditions, as reported in [5].

	$\Delta P = 50$ mmHg	$\Delta P = 55$ mmHg	$\Delta P = 60$ mmHg
Q^{data}	$1.834 \frac{\text{l}}{\text{min}}$	$1.091 \frac{\text{l}}{\text{min}}$	$0.352 \frac{\text{l}}{\text{min}}$
Q^{sim}	$1.792 \frac{\text{l}}{\text{min}}$	$1.039 \frac{\text{l}}{\text{min}}$	$0.400 \frac{\text{l}}{\text{min}}$
$\lvert Q^{data} - Q^{sim} \rvert$	$0.042 \frac{\text{l}}{\text{min}}$	$0.052 \frac{\text{l}}{\text{min}}$	$0.048 \frac{\text{l}}{\text{min}}$

Table 2: Experimental and simulation data for the model validation against experimental mesures.

4. Procedure

The aim of the proposed benchmark is to validate the FSI model via comparison of the numerical results against experimental HQ curves in wave membrane blood pumps. In this section, we explain the steps to take to install the software, set the data, run the FSI simulations and post-process the computed numerical results.

4.1. Step 0: Installation

The source files to solve this benchmark can be found on Bitbucket (`https://bitbucket.org/Martin592/`). Hence, a Bitbucket account is required. The user can clone the repository using file `clone.sh`, provided that they *xxx* with their bitbucket username. Notice that an additional repository from zstefano () is required, presenting a differnt version of Tetgen library. Then type:

```
./clone.sh
```

The LifeV environment to solve this benchmark, with all libraries installed, is available on Docker hub. In order to access to the Docker image, a free registration to `https://hub.docker.com/` is required. Therefore, it is sufficient to type:

```
docker pull martin592/validation –wmbp
```

Create a docker container using file `runLife.sh`:

```
./runLifeV.sh
```

Now you should have created a LifeV container on another terminal, with lifev-src, lifev-build (empty) and tetgen folders.

To configurate the system and build `lifev-build`, it is finally sufficient to use provided files `config.sh` and `build.sh`:

```
./config.sh
./build.sh
```

This final step may require some time and produces executable `WMBP.exe`. All the instructions for the installation are also reported in `INSTALL` file.

4.2. Step 1: Setting input data

The input data needed to run the simulations include i) the meshes/ folder, ii) the dataFile, and iii) the solverFile. Be sure that all these documents are present in the working directory.

Most of the input settings need to be maintaiend fixed for this benchmark. However, the user should set the head pressure H (in mmHg) in the pump by changing variable `[OP/pressure]` in the `dataFile` (default value is 50 mmHg). We suggest also to change the penalty parameters at lines `[penalty]` of the `dataFile`, according to what indicated in Table 1 to ensure stability. Finally, in case of serial runs or for parallel runs with less than 4 processors, change the variable `[prec/paramListFile] to solverFile_serial.txt`.

4.3. Step 2: Run simulations

To run the FSI simulations in wave membrane blood pumps, it is sufficient to execute the program with the following command from command line:

```
./WMBP.exe -f dataFile > benchmark_output.txt
```

in case of serial runs, or

```
mpirun -np 7 WMBP.exe -f dataFile > benchmark_output.txt
```

in case of parallel runs wiht number of processors equal to 7.

Notice that these simulations can last for several days and require a minimum of 40 GB of RAM for serial runs. In addition, simulation may stop due to the failure of external library tetgen, trying to solve particularly complex geometric sub-problems, inherent to XFEM-based strategy. In this case, the user should restart the simulation from the last saved time step, following the instructions detailed in the Appendix A.2.

4.4. Step 3: Post-processing

The numerical results of interest for this benchmark consist of the flow rate at the outlet. At each time istant, the software prints the computed flow rate in the output file `benchmark_output.txt`. Post-processing is structured in two steps:

1. extract the flow data from the output file `benchmark_output.txt` using Python script `extract_flowResults.py` as follows:

   ```
   python extract_flowResults.py benchmark_output.txt .
   ```

2. run the MATLAB code `validation.m` to compute the mean flow rate Q^{sim} as in Equation 10, plot it with the HQ curve, and compute the numerical error. For this step, be sure to have the experimental data in the same working directory and set the simulated head pressure H in the MATLAB script.

References

[1] C. N. Botterbusch, S. Lucquin, P. Monticone, J. B. Drevet, A. Guignabert, and P. Meneroud, "Implantable pump system having an undulating membrane," May 15 2018, uS Patent 9,968,720.

[2] M. Perschall, J. B. Drevet, T. Schenkel, and H. Oertel, "The progressive wave pump: numerical multiphysics investigation of a novel pump concept with potential to ventricular assist device application," *Artificial organs*, vol. 36, no. 9, 2012.

[3] E. Burman and M. A. Fernández, "An unfitted Nitsche method for incompressible fluid–structure interaction using overlapping meshes," *Computer Methods in Applied Mechanics and Engineering*, vol. 279, pp. 497–514, 2014.

[4] A. Hansbo and P. Hansbo, "An unfitted finite element method, based on Nitsche's method, for elliptic interface problems," *Computer methods in applied mechanics and engineering*, vol. 191, no. 47-48, pp. 5537–5552, 2002.

[5] M. Martinolli, J. Biasetti, S. Zonca, L. Polverelli, and C. Vergara, "Extended finite element method for fluid-structure interaction in wave membrane blood pump," *International Journal for Numerical Methods in Biomedical Engineering*, vol. n/a, no. n/a, p. e3467. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.3467

[6] S. Zonca, C. Vergara, and L. Formaggia, "An unfitted formulation for the interaction of an incompressible fluid with a thick structure via an XFEM/DG approach," *SIAM Journal on Scientific Computing*, vol. 40, no. 1, pp. B59–B84, 2018.

A. Appendix

A.1. SolverFile.xml

Listing 1: solverFile.xml

```
<ParameterList>
<!-- LinearSolver parameters -->
<Parameter name="Reuse Preconditioner" type="bool" value="false"/>
<Parameter name="Max Iterations For Reuse" type="int" value="80"/>
<Parameter name="Quit On Failure" type="bool" value="false"/>
<Parameter name="Silent" type="bool" value="false"/>
<Parameter name="Solver Type" type="string" value="AztecOO"/>

<!-- Operator specific parameters (AztecOO) -->
<ParameterList name="Solver: Operator List">

<!-- Trilinos parameters -->
<ParameterList name="Trilinos: AztecOO List">
<Parameter name="solver" type="string" value="gmres"/>
<Parameter name="conv" type="string" value="rhs"/>
<Parameter name="scaling" type="string" value="none"/>
<Parameter name="output" type="string" value="all"/>
<Parameter name="tol" type="double" value="1.e-6"/>
<Parameter name="max_iter" type="int" value="4000"/>
<Parameter name="kspace" type="int" value="2000"/>
<!-- az_aztec_defs.h -->
<!-- #define AZ_classic 0 /* Does double classic */ -->
<Parameter name="orthog" type="int" value="0"/>
<!-- az_aztec_defs.h -->
<!-- #define AZ_resid 0 -->
<Parameter name="aux_vec" type="int" value="0"/>
</ParameterList>
</ParameterList>
</ParameterList>
```

A.2. Simulation - Restart

In case of running error, a restart procedure has to be carried over to continue the simulation from the last saved timestep. At each time instant, the software exports the fluid and solid solutions in .h5 and .xmf format, that can be used to restart the job. For instance, at time iteration 39, the exported restart files are: `fsiRestartF_039`, `fsiRestartS1_039` (membrane) and `fsiRestartS2_039` (magnet).

The restart procedure consists of two steps:

1. Define the settings of `dataFile_restart` as in `dataFile`. In addition, the user has to specify: i) the restart time iteration [importer/initTimeIter] (e.g. to 39), and ii) the corresponding initial time [time_discretization/initialtime] (in s).
2. restart the simulation by typing on the command line:

   ```
   mpirun -np 7 WMBP.exe -f dataFile_restart > benchmark_output_restart.txt
   ```

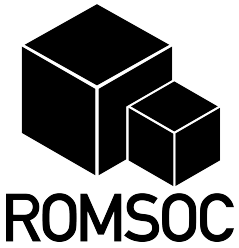
   This can be run either in serial or in parallel, as in Section 4.3. Check that the restart files are present in the working directory for both the restart time iteration and the previous one, e.g. `fsiRestart*_039`

and `fsiRestart*_038`.

Notice that for the post-processing of a simulation that required a restart, the results need to be extracted also from output file `benchmark_output_restart.txt`. Hence, the user has to run:

```
python extract_flowResults.py benchmark_output.txt .
```

and join the results together in a unique file `flowResults.txt`.



The ROMSOC project

May 17, 2021

ROMSOC-D5.3-0.1

Horizon 2020