Science and Technology Facilities Council

User Guide – Parallel Partitioned Fluid-Structure Interaction (FSI) Simulation Framework (ParaSiF_CF)

01/10/2021

1 Overview

Parallel Partitioned Fluid-Structure Interaction (FSI) Simulation Framework, ParaSiF_CF, employs Code_Saturne to solve the computational fluid dynamics (CFD), FEniCS to solve the computational structure mechanics (CSM) and MUI for data exchange. It offers a platform where users can carry out fully coupled fluid-structure interaction studies using supercomputers.

The framework uses a partitioned approach. It takes several advantages of the MUI library:

- Good scalability on communications among domains for large simulations;
- Keeping the development of the coupled solvers decoupled. It allows for easier independent testing of each solver and avoids potential incompatibilities between two solvers (i.e if they both use a certain library X each one depends on a different version of it which are not compatible);
- Coupling of multiple solvers which have different programming language interfaces (e.g C, C++, FORTRAN and Python);
- ”Plug and play” strategy. One solver can be replaced by another incrementally and without the need of recompiling if the MUI interface is used as a common adaptor;
- Use of multiple solvers which have two incompatible licenses exploiting the dual licensing of the MUI library (both solvers are never mixed source-wise or binary-wise).

The structure of ParaSiF_CF is as follows:

```
Fluid-Structure Interactions
├── Fluid Solver
│   └── Code_Saturne_V6.0.6
```

```
├── Structure Solver
│   └── FEniCS_V2019.1.0
└── Communicator
    └── MUI_V1.1.2
```

Figure 1 shows the flow chart of the present framework. The left hand side is the CFD solver for the fluid domain and the right hand side is the CSM solver for the structure domain. In the time step of $t = t + \delta t$ and sub-iteration $iter + 1$, the fluid domain solved the flow field and push the fluid forces at each cell of the fluid structure interface to the structural domain through the MUI library. The displacements at each cell of the interface were determined by the MUI coupling utility based on the fetched value from the structural domain. The calculated displacements of each cell at the interface were then applied to the fluid domain as a Dirichlet boundary condition. While the structural domain fetch fluid forces and applied as a Neumann boundary condition of the structure. It further calculated the deformation of the structure and push to the fluid solver. The stress of the structure will then be updated. Both fluid and structure domains were move to the next sub-iteration after the completion of the above actions. Several sub-iterations were needed within each time step until a convergence is reached, i.e. the R_k is smaller enough that below the criteria.

Apart from the loose coupling, the present framework also implemented both the Fixed Relaxation and the Aitken's δ^2 methods with FSI sub-iterations implemented in both CFD and CSM solvers, as shown in Figure 1. The Radial Based Function (RBF) that has been implemented in the MUI library was used to ensure the forces at the interface are conserved between the two domains.

This framework is under active development. Such infrastructure will make it possible to simulate large FSI related problems with high performance computing facilities.

2 Download

The core code of ParaSiF_CF can be obtained from GitHub:
https://github.com/ParaSiF-CF

3 Installation

There is no need to install ParaSiF_CF itself, but the following should be done before ParaSiF_CF can be used:

- FEniCS v2019.0.1 (FFC v2019.1.0.post0: https://bitbucket.org/fenics-project/ffc/src/2019.1.0.post0/ Dolfin v2019.1.0.post0: https://bitbucket.org/fenics-project/dolfin/src/2019.1.0.post0/) should be installed;
- Code_Saturne v6.0.6 (https://github.com/code-saturne/code_saturne/tree/v6.0.6) should be installed;
- MUI v1.1.2 (https://github.com/MxUI) should be obtained and its C and Python wrappers should be compiled;

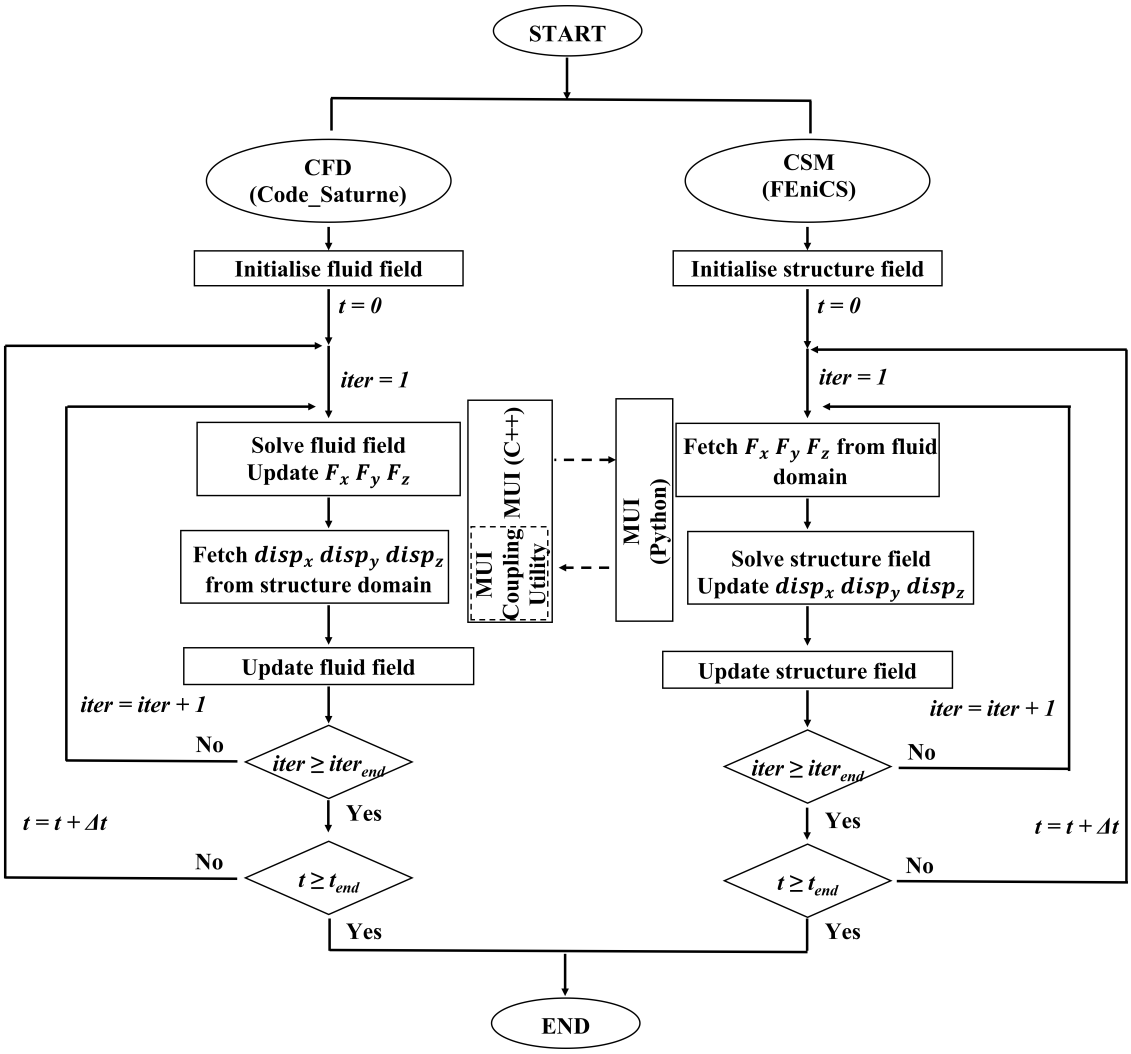

Figure 1: Flow chart of the simulation framework.

- MUI_Utilities (https://github.com/MUI-Utilities) should be obtained;
- Compile the C and Python wrappers of the FSI Coupling Lab of the MUI Utility;
- Follow the Code_Saturne_MUI_Coupling library of the MUI_Utilities to establish the Code_Saturne - MUI coupling.

4 Folder Structure

```
ParaSiF_CF
├── demo
│   └── flexBeam3D [Demo case on 3D FSI simulations]
├── docs
│   └── UserGuide
└── src
    └── CFD
        ├── Code_Saturne
        │   └── V6.0 [the subroutines that should be copied to the
        │       src subfolder of Code_Saturne run folder to enable
        │       the coupling] .3 CSM
        └── FEniCS
            └── V2019.1.0
                └── structureFSISolver [the FEniCS structure solver]
└── LICENSE-GPL-v3 .1 README.md
```

5 Demo Cases

The Framework provided a demo case on 3D FSI simulations `FSIBeam`. The folder structure for a simulation case is as follows:

```
flexBeam3D
└── runFolder [contains the original settings of the simulation
    for both fluid and structure solvers]
    ├── DATA [contains the setup data of Code_Saturne]
    ├── SRC [contains the user-defined functions of Code_Saturne]
    ├── FEniCS [contains the original settings of the simulation
    │   for the structure solver]
    │   ├── dataInput [contains the mesh, RBF matrices and checkpoint
    │   │   data of the FEniCS solver]
    │   ├── structureFSISetup [contains the setup file of the FEniCS
    │   │   solver, such as boundary conditions, input parameters
    │   │   and sub-domains]
    │   └── structureDomainRun.py [high-level script for the FEniCS
    │       solver]
    └── SCRIPTS [contains the running and compiling scripts for
        ParaSiF_CF run]
```

```
    RESU [contains the computation results for ParaSiF_CF
    run]
  MESH [contains the mesh file for the fluid domain]
```

After running, the RESU folder will contain the simulation results:

```
RESU
    fluidDomain [contains the results from the fluid solver.
    It equivalent to the RESU/YYYYMMDD-hhmm directory.  For
    more details, please find:  https://www.code-saturne.org/cms/web/
    structureDomain [contains the results from the structure
    solver]
        dataInput [contains the mesh for the structure solver]
        structureFSISetup [contains the setup for the structure
        solver]
        structureResults [contains the results from the structure
        solver]
        structureDomainRun.py [high-level script for the structure
        solver]
```

To run the demo case:

• Go to the demo folder and extract the mesh file

```
cd MESH && tar -xf Flex_Beam.tar.gz && cd ../
```

• Go to the run folder

```
cd runFolder/
```

• Source the dolfin.conf and Code_Saturne files. Add the path to ParaSiF_CF/src/CSM/FEniCS/V2019.1.0 to the Python path.

• Compile sorce code of Code_Saturne run case and copy FEniCS run case to the RESU sub-folder.

```
./SCRIPTS/compileCS
```

• Run the ParaSiF_CF framework.

```
./SCRIPTS/runcase
```