Project: Smart three-sphere swimmer near a wall

3. The computational framework

Luca Berti, L.G., C.P.

18 October 2021

What will be used

Programming language/libraries

- Python (and related libraries)
- `Feel++` fluid toolbox (to launch fluid mechanics simulations)

Software

- VSCode (integrated development environment)
- Paraview (to visualise simulations results)

Cluster

- CEMOSIS

Python and related libraries

In this project we will work with Python. We will need its libraries:

- `Pytorch` (`https://pytorch.org`) for the ML part
- `Feel++` for the fluid simulation part

Both are already available in CEMOSIS.

Python and related libraries

In this project we will work with Python. We will need its libraries:

- `Pytorch` (`https://pytorch.org`) for the ML part
- `Feel++` for the fluid simulation part

Both are already available in CEMOSIS.

A tutorial on the use of `Pytorch` for deep Q-learning is available at `https://pytorch.org/tutorials/intermediate/reinforcement_q_learning.html`.

Python and related libraries

In this project we will work with Python. We will need its libraries:

- `Pytorch` (`https://pytorch.org`) for the ML part
- `Feel++` for the fluid simulation part

Both are already available in CEMOSIS.

A tutorial on the use of `Pytorch` for deep Q-learning is available at `https://pytorch.org/tutorials/intermediate/reinforcement_q_learning.html`.

It shows how to implement the fundamentals of a deep Q-learning (Q-network) algorithm, i.e.

- the neural network
- the training and test processes

Python and related libraries

Example with Pytorch.

Feel++ fluid toolbox

We will use the Feel++ library to solve the fluid problem:

- Finite element library in C++

Feel++ fluid toolbox

We will use the Feel++ library to solve the fluid problem:

- Finite element library in C++
- Supports parallel computations

Feel++ fluid toolbox

We will use the Feel++ library to solve the fluid problem:

- Finite element library in C++
- Supports parallel computations
- Available on GitHub https://github.com/feelpp/feelpp

Feel++ fluid toolbox

We will use the Feel++ library to solve the fluid problem:

- Finite element library in C++
- Supports parallel computations
- Available on GitHub https://github.com/feelpp/feelpp
- Code for the fluid toolbox https://github.com/feelpp/feelpp/tree/develop/toolboxes/feel/feelmodels/fluid

The toolbox is configured via three files:

- .geo file, which specifies the geometry of the problem

Feel++ fluid toolbox

We will use the Feel++ library to solve the fluid problem:

- Finite element library in C++
- Supports parallel computations
- Available on GitHub https://github.com/feelpp/feelpp
- Code for the fluid toolbox https://github.com/feelpp/feelpp/tree/develop/toolboxes/feel/feelmodels/fluid

The toolbox is configured via three files:

- .geo file, which specifies the geometry of the problem
- .json file, which specifies the fluid characteristics, the boundary conditions and the fields and measures to export for visualisation

Feel++ fluid toolbox

We will use the Feel++ library to solve the fluid problem:

- Finite element library in C++
- Supports parallel computations
- Available on GitHub https://github.com/feelpp/feelpp
- Code for the fluid toolbox https://github.com/feelpp/feelpp/tree/develop/toolboxes/feel/feelmodels/fluid

The toolbox is configured via three files:

- .geo file, which specifies the geometry of the problem
- .json file, which specifies the fluid characteristics, the boundary conditions and the fields and measures to export for visualisation
- .cfg file, which collects the information concerning time-stepping, algebraic solution

The three-sphere swimmer configuration files

Configuration files for the three-sphere swimmer.

VSCode

IDE that can be used to connect to CEMOSIS, copy-paste files easily and use an integrated terminal.

- Download it here `https://code.visualstudio.com`
- Download the extensions `Remote - SSH; Python`
- Add your CEMOSIS account to the ssh targets
- Connect to CEMOSIS using your credentials

The environment is ready for you to launch your simulations.

Paraview

You can now download the results to your computer via the `rsync` command and visualise them using `Paraview` `https://www.paraview.org`.
To be more specific, you open a terminal where you want to save your results and type

```
rsync -avz YOURNAME@cemosis.feelpp.org:/home/YOURNAME/PATH .
--exclude '*log*' --exclude '*expr*' --exclude '*h5'
```

Paraview

You can now download the results to your computer via the `rsync` command and visualise them using `Paraview` `https://www.paraview.org`.
To be more specific, you open a terminal where you want to save your results and type

```
rsync -avz YOURNAME@cemosis.feelpp.org:/home/YOURNAME/PATH .
--exclude '*log*' --exclude '*expr*' --exclude '*h5'
```

In Paraview:

- You can visualise the results by opening the `fluid.exports/Export.case` file to see the fluid velocity and pressure
- `fluid.measures.csv` to see the trajectory of the body and some of the body's properties.