Software versions of FEniCS and FEniCSx. -

The Legacy FEniCs library, including Dolfin, is no longer actively developed. The latest version of legacy FEniCs (2019.1.0) latest release was in April 2019. The FEniCS project has moved to FEniCSx which is an improved software compared to FEniCS, DOLFINx is the new version of DOLFIN. The latest version of DOLFINx is version 09.0 released on October 2024, both version are problem-solving interfaces of FEniCS and FEniCSx respectively.

The differences between DOFIN and DOFINx are listed in the following table:

Table (1-1) Feature differences between DOLFIN and DOLFINx

Feature	DOLFIN(Legacy)	DOLFINx
Language	C++, Python, and Python2	C++, Python3 interface
Mesh backend	CGAL	Basix + GMSH + MeshIO
Linear algebra backend	PETSc	PETSc
Symbolic engine	UFL	UFL
Parallelism	MPI-based	MPI + better scalability
Status	Deprecated/Legacy	Actively developed

User's guide in DOLFIN and DOLFINx. –

The user's guide of DOLFIN has been updated to DOLFINx. The tutorial has been removed from the fenics.org webpage but there are some introductory demos that can be useful to understand how the software works. In order to solve the Navier Stokes equations a method using mixed elements is introduced, and how iterative linear solvers are employed to get more efficient results.

The Application Programming Interface Reference, documents on how to use the functions, classes, methods, and objects provided by the software library and framework. This list of functions is given with a short description of the arguments that need to be introduced as well as the output they produce.

There is a tutorial that can be downloaded for free at https://link.springer.com/book/10.1007/978-3-319-52462-7 that contains a book written by Hans Peter Langtangen and Anders Logg with the title: "Solving PDE's in Python. The FEniCS Tutorial 1" that is available at Springer Open.

Installation. -

1.- The following commands have to be written in a PC computer that has Windows Subsystem Linux installed in order to install FEniCs on Ubuntu:

- sudo apt-get install software-properties-common

- sudo add-apt-repository ppa: fenics-packages/fenics

- sudo apt-get-update

- sudo apt-get install fenics

In order to install DOLFIN or DOLFINx using CONDA we have to implement the following steps:

1) Load de dependencies. - module load python

2) Create a New CONDA environment. - conda create –n fenicx-env –c conda –forge fenics –dolfinx

conda activate fenicsx-env

3) Test install. - python –c import dolfinx; print (dolfinx. __version__)

The advantages of using this kind of installation is that no root access is needed, and it is easy to update.

Installing the Perfusion model.-

The perfusion model can be installed in MPI, and also in a local machine. The benefits of installing the code in a local machine are that the debugging, development, visualization can be done without having to use MPI/HPC environment. Due to the mesh size of the brain: 1042301 Tetrahedron elements in order to run full simulations in a time frame between 5 to 15 minutes it is worth to consider running in HPC.

Installation in a local machine can be accomplished by:

a) Installing FEniCS or FEniCSx using CONDA

b) Install the Python dependencies

c) Run the model locally

For the installation on HPC the following steps have to be executed:

a) Load dependencies via modules

b) Install FEniCS or FEniCSx using a custom environment with MPI support or load a precompiled FEniCs module on the cluster.

c) Run using mpirun or srun

MPI is used inside the code(MPI.comm.world) to split mesh and parallelize.

Running the Perfusion model. –

Before running the permeability initializer, we have to extract the brain meshes from the directory in which they are located or extract them from the archive of brain_meshes.tar. xz

The permeability initializer is executed: permeability_initialiser.py

The output of this code gives the following pe.xdmf, pe.h5, main_direction.xdmf, main_direction.h5, K1_form.xdmf, K1_form.h5, e_loc.xdmf, e_loc.h5.

Among this files pe_xdmf does not work as expected. The other files give correct results compared to FEniCS-Legacy.

Running the Basic_Flow_solver.-

To compute the pressure and velocity field we can use basic_flow_solver.py. For parallel execution we need to use mpirun –n indicating the number of processors python3 complex_geom_solver.py. Using 4 cores and first order finite elements, the execution takes around 2 minutes and 56 seconds.

The output of this code are the following files: vel1.xdmf, vel1.h5, vel2.xdmf, vel2.h5, volume_integrals.csv, surface_integrals.csv, press1.xdmf, press1.h5, press2.xdmf, press2.h5, press3.xdmf, press3.h5, perfusion. xdmf, perfusion.h5, beta12.xdmf, beta12.h5, beta23.xdmf, beta23.h5, K1.xdmf, K1.h5, K2.xdmf, K2.h5, K3.xdmf, K3.h5.

a)

b)

c)

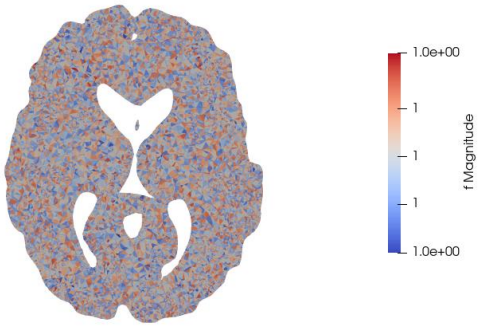


Fig (1-1) a) f magnitud middle slice b) f magnitud vertical slice c) f magnitud horizontal slice

Functionality & Correctness. –

For the permeability_initialiser code, it could be verified that it runs successfully on the University HPC cluster and takes 1 minute and 37 seconds to deliver the results.

Reproducibility. –

When applying the same input consistently, the same output is produced.

Scalability Testing. –

When running the simulations with different problem sizes it works as expected.

No excessive memory leaks or excessive resource consumption.

Error Handling & Robustness. –

There are no error messages for incorrect input

Integration & Dependency Testing. –

Compatibility with required libraries and external tools has been confirmed

Documentation & Usability. -

There are some websites from which instructions for the setup in a local computer can be found.

The installation of FEniCS and FEniCSx has already been done in Crescent2 which is the HPC used by university students.

Version Control & Continuos Integration.-

The latest versions of FEniCs (2019.1.0) and FEniCSx (09.0) are the ones that are avaliable to use in Crescent2 HPC.

Automated tests have been run for the permeability_initialiser.py by intoducing random values to the config_permeability_initialiser_yaml file.

Installing the Oxygen model.-

The oxygen model can be installed in a local computer, and also in MPI. Using CONDA the following steps need to be followed for installing in a local computer:

a) Create an environment

b) Install dependencies

c) Install Legacy FEniCS (dolfin)

d) Run using the following command: python3 oxygen_solver.py

To install the oxygen model in HPC the following steps are needed:

a) Load Environment Modules

b) Create virtual Environment

c) Install FEniCs(from source or module)

d) Run in parallel using the following command: mpirun -n 6 python3 oxygen_solver.py

Running the Oxygen model.-

The oxygen model reproduces the oxygen transport across three compartments: Arteria, Capillary, and Tissue. As an input, the code takes YMAL files with simulation parameters, also supports optional results directory.

The subprograms are IO function and Finite Element Solver that accomplish the following tasks:

IO function: Read the input mesh

Finite Element Solver: All the calculations are realized in the Finite Element Solver.

The program solves oxygen equations, specifically coupled partial differential equations, and return concentrations in all the compartments.

Saves the concentration fields to C1, C2, C3 that correspond to:

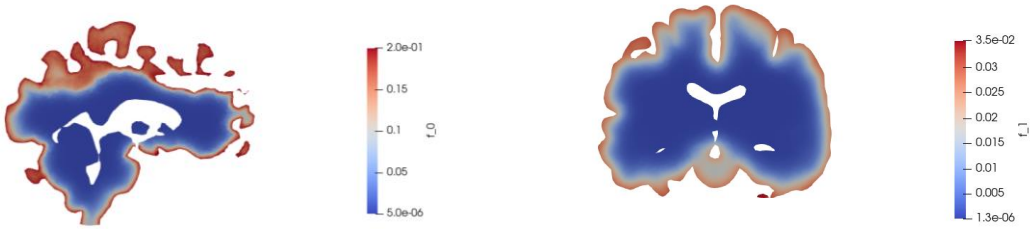
C1 = C_a (arterial oxygen)

C2 = C_c (capillary oxygen)

C3 = C_t (tissue_oxygen)

The results are saved in ,xdmf and .h5 extensions that are suitable to open in ParaView.

a) b)



c)

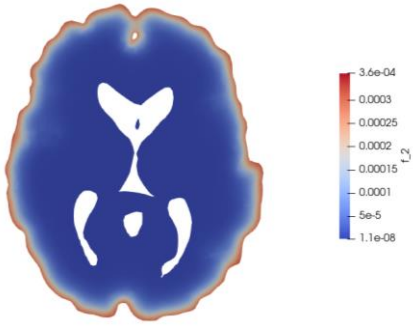


Fig (1-1) a) f_0: arterial b) f_1: capillary c) f_2: tissue oxygen distributions

Functionality & Correctness. –

The code runs successfully on the University HPC cluster at takes less than 1 minute to deliver the results.

Reproducibility. –

When applying the same input consistently, the same output is produced.

Scalability Testing. –

When running the simulations with different problem sizes it works as expected.

No excessive memory leaks or excessive resource consumption.

Error Handling & Robustness. –

There are no error messages for incorrect input

Integration & Dependency Testing. –

Compatibility with required libraries and external tools has been confirmed

Documentation & Usability. -

There are some websites from which instructions for the setup in a local computer can be found.

The installation of FEniCS and FEniCSx has already been done in Crescent2 which is the HPC used by university students.

Version Control & Continuos Integration.-

The latest versions of FEniCs (2019.1.0) and FEniCSx (09.0) are the ones that are avaliable to use in Crescent2 HPC.