How to install FEniCS

On your laptop

Instructions on how to install FEniCS are available https://fenicsproject.org/download/. You also need a working `python3` installation with several additional packages.

Except for Ubuntu distributions (see below), we suggest at first to try to use Anaconda, which has an advanced package manager that can provide you without effort most of the software that you need. Install the Anaconda version based on *python3*.

On ubuntu (preferred installation)

Install basic Python 3 packages

```
sudo apt update
sudo apt upgrade
sudo apt install black cython3 jupyter-core jupyter-client jupyter-
notebook pylint python3-autopep8 python3-numpy python3-flake8 python3-h5py
python3-jupyter-console python3-matplotlib python3-mpi4py python3-notebook
python3-petsc4py python3-pytest python3-requests python3-scipy python3-
slepc4py python3-skimage python3-sphinx python3-sympy spyder pelican
jupyter-nbconvert jupyter-qtconsole python3-qtconsole
```

Then install the FEniCS specific packages

```
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:fenics-packages/fenics
sudo apt-get update
sudo apt-get install --no-install-recommends fenics
```

On windows

FEniCS is not distributed for Windows boxes. For Windows 10, the preferred option is the Windows subsystem for linux (WSL): install the Ubuntu distribution, then refer to the section above.

If you cannot install the WSL, you will need to create Ubuntu virtual machine. Get in touch with your instructors in that case. Then in side the ubuntu virtual machine

```
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:fenics-packages/fenics
sudo apt-get update && upgrade
sudo apt-get install fenics
sudo apt install python3 python3-pip ipython3 jupyter-core jupyter
sudo pip3 install jupyterlab
```

To launch a notebook:

```
jupyter lab --no-browser
```

And then open the http address indicated in the ubuntu terminal in the browser of your windows system.

Cloud-based Google colab installations

You can run python programs jupyter notebooks and FEniCS on online servers. The basic service is free and can be a solution if all other installation systems fail.

- You need a google account
- You can use this example to start https://colab.research.google.com/drive/1K0tQ4_RGuY-5YaHLUHTWqJEWGvrte9n0?usp=sharing
- You can save the notebooks and your working environment on your google drive

We suggest to use this only as emergency solution

- *advantages:* You do not need to install anything on your machine and you do note use the resources of your machine

- *disadvantages:* It can be slow (especially Azure, colab seems better). You can use only jupyter notebooks, you share your date with google or microsoft, you do not a full control of the system, you need to be online with a good network connection.

Using docker virtual machines (suggested method if others fail) :

This works on windows, linux, macos

1. First download docker https://www.docker.com/get-started
2. On the terminal (this will share the current directory with the directory `/home/fenics/shared` in the virtual machine):

   ```
   docker run --name notebook -w /home/fenics -v
   $(pwd):/home/fenics/shared -d -p 127.0.0.1:8888:8888 cmaurini/fenics-
   stable-20.04 'jupyter-notebook --ip=0.0.0.0 --allow-root'
   ```

3. At the terminal type `docker logs notebook` and open the http address indicated in the output in your browser. This will give you a jupyter notebook working in the virtual machine.

See also https://fenicsproject.org/download/ on the docker section and https://fenics-containers.readthedocs.io/en/latest/index.html for further issue

- If you have an error like

```
docker: Error response from daemon: Conflict. The container name
"/notebook" is already in use by container
"db4b0b976ae09df4156680b9b667291e39237f8c457eee735d5f7de7fff20459".
You have to remove (or rename) that container to be able to reuse that
name.
```

execute at the terminal `docker rm -f notebook` and try again.

- If you want a terminal inside the virtual machine without the notebook interface:

```
docker run -ti  -v $(pwd):/home/fenics/shared -w /home/fenics/shared
cmaurini/fenics-stable-20.04
```

Anaconda install (Mac and Linux)

This procedure could work on Mac (but it is not guaranteed). We do not suggest to follows this on linux.

First install Anaconda, following the instruction here `https://docs.continuum.io/anaconda/install/`.

Then run following commands in your terminal (the file `fenicsproject.yml` must be in your directory):

```
conda config --add channels conda-forge
conda config --set channel_priority strict
conda create -n fenicsproject
source activate fenicsproject
conda install fenics mshr
```

The above commands will create a virtual environment called `fenicsproject`, in which the FEniCS library and other useful packages will be installed. This avoids conflicts between various packages. This means that **you need to activate the `fenicsproject` environment** when you install additional packages or perform FEniCS computations. If the environment is activated, your terminal prompt will display `(fenicsproject)`, like so:

```
(fenicsproject) maurini@maurinis-MacBook-Pro:~$
```

We suggest also to install other useful packages

```
conda install gmsh pip jupyter spyder matplotlib scipy vtkplotter meshio
```

If you need to install further packages, just type `conda install packagename` (preferred) or `pip3 install packagename`.

How to test the installation

Run following commands in your terminal:

```
wget
https://fenicsproject.org/docs/dolfin/latest/python/_downloads/598330b504d
63e359baad030e1010987/demo_poisson.py
python3 demo_poisson.py
```

or copy and paste this code in the notebook and exectute the cell

```
import dolfin
import mshr
import ufl
print("You are using FEniCS version {:s}".format(dolfin.__version__))
mesh = dolfin.UnitSquareMesh(64, 64)
V = dolfin.FunctionSpace(mesh, "Lagrange", 1)
# Define Dirichlet boundary (x = 0 or x = 1)
def boundary(x):
    return x[0] < dolfin.DOLFIN_EPS or x[0] > 1.0 - dolfin.DOLFIN_EPS
# Define boundary condition
u0 = dolfin.Constant(0.0)
bc = dolfin.DirichletBC(V, u0, boundary)
# Define variational problem
u = ufl.TrialFunction(V)
v = ufl.TestFunction(V)
f = dolfin.Expression("10 * exp(-(pow(x[0] - 0.5, 2) + pow(x[1] - 0.5, 2))
/ 0.02)", degree=2)
g = dolfin.Expression("sin(5*x[0])", degree=2)
a = ufl.inner(ufl.grad(u), ufl.grad(v)) * ufl.dx
L = f * v * ufl.dx + g * v * ufl.ds
# Compute solution
u = dolfin.Function(V)
A, b = dolfin.assemble_system(a,L,[bc])
dolfin.solve(A,u.vector(),b)
dolfin.plot(u)
```