PETSc installation notes

P. Balty, P. Billuart and G. Winckelmans

March 24, 2021

Introduction

PETSc is the world's most widely used parallel numerical software library to solve sparse linear systems. The solving can be performed parallely (with the Message Passing Interface (MPI) standards) or on a single CPU. For the MECA2660 project, you will use the PETSc routines in order to solve, on a single CPU, the following Poisson equation, second step of the reprojection temporal scheme :

$$\nabla^2 \Phi = \frac{\nabla \cdot \mathbf{v}^*}{\Delta t} \tag{1}$$

The boundary condition to impose will be an homogeneous Neumann boundary condition :

$$\frac{\partial \Phi}{\partial \mathbf{n}} = 0 \tag{2}$$

Imposing such a constraint is equivalent to impose an equality between the velocity predictor along the normal direction $\mathbf{v}^* \cdot \mathbf{n}$ and the velocity solution at the next step along the normal direction too : $\mathbf{v}^{n+1} \cdot \mathbf{n}$. Indeed, rewritting the correction step at the boundary and projecting it on the normal direction provides :

$$\left(\frac{\mathbf{v}^{n+1} - \mathbf{v}^*}{\Delta t} \right) \cdot \mathbf{n} = -\left(\nabla \Phi \right) \cdot \mathbf{n} = -\frac{\partial \Phi}{\partial \mathbf{n}} = 0 \tag{3}$$

By imposing a Neumann boundary condition for Φ on the boundaries, one also equivalently imposes a zero mass flow for $\mathbf{v}^*$ at the computational domain boundary. Indeed, if one integrates the Poisson equation on the computational domain :

$$\int_{\Omega} \nabla^2 \Phi \, dS = \int_{\Omega} \frac{\nabla \cdot \mathbf{v}^*}{\Delta t} \, dS \tag{4}$$

and one applies to this last expression the divergence theorem, one obtains :

$$\int_{\partial\Omega} \nabla \Phi \cdot \mathbf{n} \, dl = \int_{\partial\Omega} \frac{\partial \Phi}{\partial \mathbf{n}} \, dl = \int_{\partial\Omega} \frac{\mathbf{v}^*}{\Delta t} \cdot \mathbf{n} \, dl \tag{5}$$

The normal derivative of Φ being set to zero $\left(\frac{\partial \Phi}{\partial \mathbf{n}} = 0 \right)$, one then obtains :

$$\int_{\partial\Omega} \mathbf{v}^* \cdot \mathbf{n} \, dl = 0 \tag{6}$$

Such constraint must be satisfied at each time step before solving the Poisson equation in order to guarantee the convergence of the PETSc solver. In this project, we enforce the inflow velocity to the Log-Law profile and we compute the velocity at the outflow via a natural ouflow boundary condition. The mass flow leaving the computational box on the right boundary is therefore not necessary equal to the mass flow entering by the left boundary. To satisfy the divergence free constraint of the flow, you should then slighty correct (an uniform correction is already perfect) the velocity computed by the natural outflow boundary condition in such a way to enforce a strict equality between the two mass flows. This correction must be done before computing the RHS of the Poisson equation.

After discretization on the mesh grid, the Poisson equation will take the form of a linear equation at each mesh point. Gathering the equations from all mesh points leads to a linear system that can be written in a matricial form as :

$$\underline{\underline{A}} \cdot \underline{\Phi} = \underline{b} \tag{7}$$

In the file *poisson.c*, you will find a set of functions partially implemented which solve the Poisson equation. To complete the implementation, two tasks are basically demanded to you :

- to fill the sparse matrix $\underline{\underline{A}}$ with the correct factors.
- to fill the rhs of the system (vector $\underline{b}$) at each time step.

Exact instructions for modification of function are given above each prototype in file *poisson.c*. The current implementation of function *initialize_poisson_ solver* is solving the linear system in a direct way, i.e. by computing the inverse of $\underline{\underline{A}}$ once (at the beginning of the simulation) and performing at each time step the product of this inverse with the vector $\underline{b}$. You should be able to run the empty project by first compiling it with the makefile (juste tape *make*) and next launching it by taping *./project*, where *project* is your default executable file. Don't forget to get a look at the header file *poisson.h* too. This last should be included in files using the functions defined in *poisson.c*.

To install the PETSc library on your computer, please refer to the corresponding instructions below in function of your operating system. If during the installation you meet a problem and, after a half day of research, you still didn't manage to solve it, please contact as soon as possible Philippe Billuart by email (philippe.billuart@uclouvain.be) to set an appointment.

MacOS

Check first if you have howebrew installed; paste the following line in a macOS terminal :

```
$ brew --version
```

If installed, the Howebrew version should appear. If it's not, the brew command is not found. Paste then the following line in a your macOS Terminal to install it :

```
$ /usr/bin/ruby -e "$(curl␣-fsSL␣https://raw.githubusercontent.com/
    ↪ Homebrew/install/master/install)"
```

Once Howebrew installed, install PETSc. By default, all the dependencies needed by PETSC and not yet installed or not up-to-date will be installed or updated at the same time. To do that, paste the following command line :

```
$ brew install petsc
```

Once PETSc installed, find the path to the folder where it was installed. This path was written in comments during the installation and should be */usr/local/opt/petsc*. If this last folder doesn’t exist, find it using the command line :

```
$ find /usr/ -name ’petsc’
```

and write the paths to the library and include folders into the Makefile, and also the shortcut of the dynamical library (before the *.dylib*) :

```
INC_DIR := -I/usr/local/opt/petsc/include
LIB_DIR := -L/usr/local/opt/petsc/lib
LIB := -lpetsc.3.10
```

Linux

First install or update the gcc compiler :

```
$ sudo apt-get install gcc
```

Do the same for the gfortran compiler :

```
$ sudo apt-get install gfortran
```

Do the same for the g++ compiler :

```
$ sudo apt-get install g++
```

Do the same for python compiler :

```
$ sudo apt-get install python
```

Install also make :

```
$ sudo apt-get install make
```

Afterwards, download the archive of the full distribution of PETSc (i.e. NOT the ’lite’ version), on the link *https://www.mcs.anl.gov/petsc/download/index.html* and untar it :

```
$ tar -zxvf petsc.3.10.4.tar.gz
```

Next go to the folder untared :

```
$ cd petsc-3.10.4
```

and configure the installation :

```
$ ./configure --prefix=/home/alain/Bureau/Projet_2660/lib_petsc --with-cc
   ↪ =gcc --with-cxx=g++ --with-fc=gfortran --download-mpich --download
   ↪ -fblaslapack
```

Next build the PETSc library by followning the instruction given by PETSc at the end of the configuration. The command line should seem like :

```
$ make PETSC_DIR=/home/alain/Bureau/petsc-3.10.4 PETSC_ARCH=arch-linux2-c
   ↪ -debug all
```

Install the PETSc library by followning the instruction given by PETSc at the end of the configuration. The command line should seem like :

```
$ make PETSC_DIR=/home/alain/Bureau/petsc-3.10.4 PETSC_ARCH=arch-linux2-c
   ↪ -debug install
```

And test the correct working of the library by following again the instruction given at the end of the configuration. The command line should seem like :

```
$ make PETSC_DIR=/home/alain/Bureau/Projet_2660/lib_petsc PETSC_ARCH=""
   ↪ test
```

Write next the paths to the library and include folders in the Makefile, and also the shortcut of the dynamical library (before the *.so*) :

```
  INC_DIR := -I/home/alain/Bureau/Projet_2660/lib_petsc/include
  LIB_DIR := -L/home/alain/Bureau/Projet_2660/lib_petsc/lib -Wl,-rpath=/
     ↪ home/alain/Bureau/Projet_2660/lib_petsc/lib
  LIB := -lpetsc
```

Windows

If your OS is Window 10, you can directly install a *Windows Subsystem for Linux (WSL)* feature from *Microsoft store*. To do the installation, follow the instructions at the following link : `https://support.ceci-hpc.be/doc/_contents/QuickStart/ConnectingToTheClusters/WSL.html`. Install next the PETSc library as you were on Linux and implement the project on the WSL exclusively.

If your OS is Windows 7 or 8, consider to upgrade to Windows 10 cause it is still free, next install the *Windows Subsystem for Linux (WSL)* feature from *Microsoft store* (see Windows 10) and finally install the PETSC library as you were on *Linux*.

If your OS is yet older or your don't want to upgrade to Windows 10, the best solution would be to install a *VM-Virtual Machine* with *Linux* or buidling a *dual boot*. If you are in this situation and not confident with yourself with any of the two solutions proposed, take an appointment as soon as possible with Philippe Billuart (philippe.billuart@uclouvain.be).