Building PETSc from Sources

November 28, 2023

In the context of the project, PETSc `http://www.mcs.anl.gov/petsc/` provides a collection of solvers for the pressure equation. PETSc allows for greater flexibility by allowing to select the solver to use.

This document should guide you on the process of compiling the source of the libraries and using them for the project.

Here, it is assumed that you are using Ubuntu. However, these instructions are applicable to many Linux distributions, besides the process of obtaining the building tools.

More details can be obtained from `http://www.mcs.anl.gov/petsc/documentation/installation.html`

Installing the building tools

The required building tools for Ubuntu can be obtained with the following command:

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

Python 2 is also required for the configuration, but it is included in Ubuntu.

Obtaining the sources

To begin with, retrieve the source code for PETSc. Right now, the program uses the interface of the version 3.3, so we'll get that file. The package can be located in `http://ftp.mcs.anl.gov/pub/petsc/release-snapshots/petsc-3.3-p6.tar.gz`. Here, `p6` denotes the revision. You can select a more recent one and probably not encounter a problem, however, you will have to replace the version when entering the commands.

You can obtain and extract the package by entering:

```
$ wget http://ftp.mcs.anl.gov/pub/petsc/release-snapshots/petsc-3.3-p6.tar.gz
$ tar -xf petsc-3.3-p6.tar.gz
```

Enter the source directory by typing

```
$ cd petsc-3.3-p6
```

Configure and build

You can configure by entering

```
$ ./configure --with-shared-libraries=1 --with-cc=gcc --with-fc=gfortran
--download-f-blas-lapack --download-mpich --prefix=INSTALLATION_DIRECTORY
```

This will download MPI and BLAS-LAPACK. If you already have these libraries, you can remove the corresponding option. If you decide not to download MPI and instead provide the path to the libraries already in your computer, you should also remove the options that specify the compiler to use (`--with-cc` and `--with-fc`). INSTALLATION_DIRECTORY corresponds to the directory where you want to install Petsc. If a default directory should be chosen by Petsc, you can also skip this option.

If you have problems running the configuration script, you can try explicitly calling Python:

```
$ python ./configure <options>
```

After the configuration is complete, the configuration scripts ends and tells you the next steps. These are:

```
$ make PETSC_DIR=CURRENT_DIRECTORY PETSC_ARCH=SOMEARCHITECTURE all
```

In the default installation, the architecture type will contain the keyword *debug* which states that your Petsc installation supports debugging options. This may be helpful during debugging of your program, however, it will reduce computational efficiency. In order to have a release installation, you may step through this installation once again adding the option "–with-debugging=no" during the configuration step and using another build directory INSTALL_DIRECTORY for this build.

After this command completed, you are again told what to do by the final output: just run the command

```
$ make PETSC_DIR=CURRENT_DIRECTORY PETSC_ARCH=SOMEARCHITECTURE install
```

and your installation should be finished.

You can subsequently test your installation by running the command:

```
$ make PETSC_DIR=INSTALLATION_DIRECTORY test
```

In order to use the installation, you need to specify the location of your Petsc installation and so make it visible to your system. Therefore, change into your home directory by typing “cd ” and open the file .bashrc by any editor of your choice. Add a line

```
export PETSC_DIR=INSTALLATION_DIRECTORY
```

to the file, save and close it. Each time you open a terminal from now, INSTALLATION_DIRECTORY will point to your Petsc installation. You should now be able to successfully compile NS-EOF on your system.