The DTU HPC system

and how to use TopOpt in PETSc on a HPC system, visualize and 3D print results.

Niels Aage
Department of Mechanical Engineering
Technical University of Denmark

Email: naage@mek.dtu.dk

Group homepages:
www.topopt.dtu.dk
www.camm.elektro.dtu.dk

March 3, 2016

Getting started

This note describes how to 1) set up your profile to use PETSc on the DTU cluster, 2) where to download and how to compile the TopOpt in PETSc [1] framework and 3) how to run jobs on the cluster. A separate exercise sheet provides a number of programming and numerical examination exercises.

On a side note we advise you to have an installation of PETSc on your local machine, e.g. your desktop or laptop, such that you can continue your work after the course ends. Compilation of PETSc [2] and the TopOpt framework is most easily done using a Linux machine (see `http://www.mcs.anl.gov/petsc/` and `www.topopt.dtu.dk/PETSc`, respectively). Note however, that both Windows and Mac installations are possible but that we do not have experience with this and therefore can not provide you with support.

Running jobs on the DTU cluster

Before you get started with the PETSc exercises on the HPC system you must complete the following setup tasks to ensure that you 1) can access the HPC system, 2) have access to an optimized PETSc build and 3) that you can compile the TopOpt in PETSc framework.

2.1 Accessing the DTU cluster and the PETSc module

You will need to connect to the cluster through a secure shell (ssh) and submit your jobs to the queueing system (see www.cc.dtu.dk for more on system, hardware, etc.). Below we describe a simple way to do this on the DTU MCAD databar using thinlinc and some alternative methods.

1 If you are already familiar with ssh you, use a terminal (on Mac or Linux) to access the HPC system at `login.hpc.dtu.dk`.

2a Alternatively, you can use a Thinlinc Client and connect it to the following address: `thinlinc.gbar.dtu.dk` using your DTU username and password (Thinlinc is freely available and installed on the Windows machines in the MCAD databar in 421).

2b Use the open a terminal (Application Menu → DTU → xterm) and connect to the cluster by typing `ssh login.hpc.dtu.dk`.

Once you can connect to the cluster you need to ensure that you have access to the PETSc library and MPI compilers. That is, you should load the correct modules to get fortran, c and c++ compilers and MPI as well as access to the pre-compiled PETSc library.

1a Create a file in your home directory called `.gbarrc` (the dot is important) and add the following line to it:
`MODULES=petsc/3.6.3-march-2016`

This gives you access to a set of compilers and an optimized PETSc installation including several additional packages.

1b Disconnect by pressing (ctrl+d). The next time you connect to the cluster you will have access to the compilers and PETSc.

2 Alternatively, you can load the module each time you login by typing:

```
module load petsc/3.6.3-march-2016
```

You are now ready to compile PETSc programs on the HPC system.

2.2 Batch jobs on the HPC cluster

The TopOpt cluster has a total of about 1300 cores divided into two main set of nodes (64 and 42) with 8 and 20 cores respectively. You are given access to *a total of 64 processes* to use for your studies. In order to start a programme on the HPC system you will need to submit a job script to the scheduler. An example of such a script is show below which you can use as a basis for your studies.

```
#!/bin/sh
#PBS -N JobName
#PBS -l nodes=2:ppn=8
# -- estimated wall clock time (execution time): hh:mm:ss --
#PBS -l walltime=24:00:00
#PBS -k oe
# -- choose the queue
#PBS -q hpc # 'hpc' has ppn=8 or 20 (topopt has ppn=12)

# -- load the right PETSc install
module load petsc/3.6.3-march-2016

echo $PBS_O_WORKDIR
cd $PBS_O_WORKDIR

# -- Due to patch of arcchtectures, compile before running !
# COMPILE COMMAND

mpirun -np 16 -mca btl openib,self ./myPetscProgramme
```

You should copy the above lines into a file called e.g. `job_script.sh`. This script tells the scheduler the name of the job (-N), how many nodes and cores are desired (-l where nodes and ppn corresponds to the number of nodes and cores respectively), that you wish both output and error messages (-k) and that you want to run in the topopt queue (-q). The following lines ensures that you use the right environment variables a The final lines are the instructions for executing the code and should be followed by a list of input parameters for PETSc to set at run time. The list of commands between `mpirun -np 16` and the programme name, i.e. `-mca btl openib,self` tells the HPC system how to organize your job on each node (optimize memory usage, etc.)

Below a simple introduction on submitting, monitoring and deleting jobs, is given

- Submit the job by typing `qsub job_script.sh`

- Monitor the job by typing `qstat -a`.
- Kill the job by typing `qdel JOBID`, where `JOBID` is the unique ID of the job which can be obtained by the `qstat` command.

You can also monitor the usual output to screen by looking in the file called `JobName.o(JOBID)`. Use e.g. `tail` or `cat` for this purpose.

The DTU HPC cluster is a patchwork of different architectures and I recommend you to use the 20 core nodes. This can be ensured by changing the queue parameter to `#PBS -q hpc` and the resources to `#PBS -l nodes=X:ppn=20`) where $X = 1, 2$ or 3.

Compiling the TopOpt in PETSc framework

You should now be ready to download, compile and test the default example contained in the TopOpt in PETSc framework.

- The first step is to transfer the code to the HPC cluster. The easiest way is to type
  `git clone https://github.com/topopt/TopOpt_in_PETSc.git`
  in a terminal, or download it as a zip archive by typing
  `(wget https://github.com/topopt/TopOpt_in_PETSc/archive/master.zip`.
- Unpack the zip file by typing `unzip master.zip` (if `git clone` was used - skip this step).
- Make sure you have to set the right environment variables! See the previous section for details - otherwise you cannot compile anything with PETSc
- Enter the TopOpt in PETSc directory and copy the reference makefile, that is type
  `cp makefile_ref makefile`
- Edit the `makefile` as follows (use an editor of your choice e.g. `vim`, `nano`).
  Modify `PETSC_DIR` such that it points to the optimize PETSc installation on the HPC system - e.g.
  `PETSC_DIR=/appl/petsc/3.6.3-march-2016/${CPUTYPEV}`
  Comment out the line with `PETSC_ARCH` using suffix # (or delete the line completely).
- You are now ready to compile and test the code. Compilation is conducted by `make topopt -j` and run the default example by typing
  `mpirun -np 4 topopt`

- Postprocess the output by typing `python bin2vtu.py #`, where # refers to the dataset (or iteration) that you wish to visualise. This will extract the data from `output.dat` and convert it to a file using the unstructured VTK format (called `output_#####.vtu`) which can be opened in e.g. ParaView.

You are now ready to transfer your results (the file ending with `.vtu`) to your local machine for visualization in ParaView.

3.1 Transfer of results and visualization

Since you only have a access to the cluster for a limited time, you should make sure to transfer your results, modified code, etc. back to your Windows (or Linux) machine.

- Windows: Use FileZilla FTP to connect to the cluster (`transfer.gbar.dtu.dk`) at port 22.
- Linux: Use `scp` to copy data from the cluster to your private machine.

Once the data is on your local machine (or the PC's in the MCAD databar) you can visualise the results in ParaView.

3.2 Export from ParaView: Design and 3D prints

In ParaView you can use the built-in filters to produce a good looking design representations and prepare the design for 3D printing. Below the necessary steps are presented. All filters can be found in the top menu under alphabetical, while some a available directly from the tool bar.

- Open your TopOpt result in ParaView and press apply.
- Calculator: Create a magnitude field of the displacements by typing the following in expression box `ux*iHat+uy*jHat+uz*kHat`. By default the field is named 'Result'.
- Clean-to-grid: Removes double nodes
- Cell-data-to-point-data: As the name says
- Iso-Volume: Use `xPhys` as input scalar. Set the threshold to $\mathrm{min} = 0.5$ and $\mathrm{max} = 1$ and press apply.

- Extract-Surface: As the name says
- Triangulate: As the name says
- Decimate: reduce the number of surface triangles. IMPORTANT: check the ’Preserve topology’ box.
- Save as STL under File.

You can now 3D print the result on your favorite printer. Note that for pure visualization purposes you may (and should) stop after creating the IsoVolume.

Bibliography

[1] Niels Aage, Erik Andreassen, and Boyan Stefanov Lazarov. Topology optimization using PETSc: An easy-to-use, fully parallel, open source topology optimization framework. *Struct Multidisc Optim*, 51(51), 2015.

[2] Satish Balay, Shrirang Abhyankar, Mark˜F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Victor Eijkhout, William˜D. Gropp, Dinesh Kaushik, Matthew˜G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry˜F. Smith, and Hong Zhang. PETSc Users Manual. Technical Report ANL-95/11 - Revision 3.5, Argonne National Laboratory, 2014.