**Install and Build PETSc and Crunch**

Find all of the packages here (you will need Base Kit and HPC, since HPC includes Fortran)

[https://www.intel.com/content/www/us/en/developer/tools/oneapi/toolkits.html#hpc-kit](https://www.intel.com/content/www/us/en/developer/tools/oneapi/toolkits.html%23hpc-kit)

**Intel oneAPI BaseKit Install**

Find all of the packages here (you will need Base Kit and HPC, since HPC includes Fortran)

[https://www.intel.com/content/www/us/en/developer/tools/oneapi/toolkits.html#hpc-kit](https://www.intel.com/content/www/us/en/developer/tools/oneapi/toolkits.html%23hpc-kit)

Command line on Linux machine:

wget https://registrationcenter-download.intel.com/akdlm/irc\_nas/19079/l\_BaseKit\_p\_2023.0.0.25537.sh

Install as root (requires root password)

sudo sh ./l\_BaseKit\_p\_2023.0.0.25537.sh

or as the current user

sh ./l\_BaseKit\_p\_2023.0.0.25537.sh

**Intel oneAPI hpckit Install**

sudo apt install intel-hpckit

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Then, install PETSC

[**https://petsc.org/release/install/**](https://petsc.org/release/install/)

**git clone -b release https://gitlab.com/petsc/petsc.git petsc**

**git pull # obtain new release fixes (since a prior clone or pull)**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Debug Version using oneAPI (using MKL library distributed with oneAPI)**

Source the file:

/opt/intel/oneapi/setvars.sh

then

./configure --with-cc=mpiicc --with-cxx=mpiicpc --with-fc=mpiifort COPTFLAGS=" -g -O0" FOPTFLAGS=" -g -O0" CXXOPTFLAGS=" -g -O0" --with-debugging=1 --with-blaslapack-dir=/opt/intel/oneapi/mkl PETSC\_ARCH=mpi-oneAPI-dbg

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Optimized Version using oneAPI (using MKL library distributed with oneAPI)**

Source the file:

/opt/intel/oneapi/setvars.sh

then

./configure --with-cc=mpiicc --with-cxx=mpiicpc --with-fc=mpiifort COPTFLAGS=" -g -O3" FOPTFLAGS=" -g -O3" CXXOPTFLAGS=" -g -O3" --with-debugging=0 --with-blaslapack-dir=/opt/intel/oneapi/mkl PETSC\_ARCH=mpi-oneAPI-opt

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**To download and build CrunchTope from GitHub**

Create directory where you want to install the Github version, typically:

mkdir Git-CrunchTope

Change directory (cd) to this directory and type

git clone <https://github.com/CISteefel/CrunchTope>

This will install all of the Fortran files and the up to date Makefile here:

~/Git-CrunchTope/CrunchTope

where the tilde is the relative path (depends on your directory structure, typically with your own username embedded there)

To update the repository (do not repeat the “clone” command), move to the directory and enter:

git pull

This should update your repo to the latest of what is in the Master branch on GitHub in

<https://github.com/CISteefel/CrunchTope>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Compiling Crunchtope

Source the file:

/opt/intel/oneapi/setvars.sh

To compile CrunchTope, you set the PETSC\_ARCH to the desired build. Normally you will have likely built PETSc with

linux-intel-dbg

linux-intel-opt

so for any available and desired, you can set in the CrunchTope repo:

export PETSC\_ARCH=linux-intel-opt

and the settings will be inherited from the PETSc build. If you are running a true Debug version of CrunchTope, perhaps to trap subscript overruns, you can add to the Makefile:

For Intel:

FFLAGS  = -w -traceback -check bounds -ffpe-trap=invalid,overflow,zero

For gfortran:

FFLAGS  = -w -fcheck=bounds -fbacktrace - -ffpe-trap=invalid,overflow,zero

For production or optimized runs, these checks and trackbacks will slow the execution down, so remove them for optimized runs.

Then to avoid having to copy the CrunchTope executable into every directory, you can add the address in your .bashrc file (which then must be “source” to have the commands take effect)

CrunchTope\_Dir=/home/cisteefel/Git-CrunchTope/CrunchTope

PETSC\_DIR=/home/cisteefel/Git-petsc/petsc

PETSC\_ARCH=linux-intel-opt

export PETSC\_DIR PETSC\_ARCH CrunchTope\_Dir

Now when you do “which CrunchTope” from ANY directory, it should echo the destination you have given in the .bashrc file (or you could do this at the command line for any terminal you have opened).

So at the command line in the terminal, you could do:

export CrunchTope\_Dir=/home/cisteefel/Git-CrunchTope/CrunchTope

which CrunchTope

/home/cisteefel/Git-CrunchTope/CrunchTope