**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>

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**To download PETSc from GitHub**

Follow the instructions on the PETSc web site:

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

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

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

**To build PETSc from GitHub**

**+++++++++++++++++++++> Macintosh Version ++++++++++++++++++++++++++++++**

This approach worked in the past, and was recently tested on a Mac with the M1 chip. Note that the M1 is not an Intel chip, so Intel oneAPI described below for Linux and Windows does not work. So, you have to use *gcc* and *gfortran*…

If your Mac already has HomeBrew, then

brew install gcc

and you should get both gcc and gfortran. If not, install HomeBrew from the site:

<https://brew.sh/>

From your terminal window (Unix environment), check that it is found with:

which gcc

which gfortran

and you should see the path for these compilers echoed. You may have to change to the BASH shell from Z-shell, so if needed, give the command:

chsh -s /bin/bash

Then set your Environment Variables within the BASH shell:

PETSC\_DIR=/Users/ YourName/petsc

PETSC\_ARCH=mac-gnu-debug or =mac-gnu-opt

The PETSC\_ARCH Environment Variable is typically overwritten with the Python “configure” script that PETSc uses to build, but this may not be carried over into the build for CrunchTope, so best to set it.

Then

export PETSC\_DIR PETSC\_ARCH

Change directories (cd) to /Users/YourName/petsc and run the “configure” script.

**Debug Version**

./configure --with-cc=gcc --with-cxx=g++ --with-fc=gfortran --download-fblaslapack --with-mpi=0 --with-debugging=1 PETSC\_ARCH=mac-gnu-debug

**Optimized Version**

./configure --with-cc=gcc --with-cxx=g++ --with-fc=gfortran --download-fblaslapack --with-mpi=0 --with-debugging=0 PETSC\_ARCH=mac-gnu-opt

If the configure script is successful, you will send a “Make” command that can be copied and pasted to build PETSc and then another one to check it.

**+++++++++++++++++++++> Linux Version ++++++++++++++++++++++++++++++**

You can use gcc and gfortran if you really want, in which case you should follow the steps above for Mac, including HomeBrew if you do not have current versions of gcc and gfortran.

Otherwise, the recommended path is to use the free Intel oneAPI compilers, which also are typically optimized for the Intel chips that are likely used in your Linux workstation.

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>

*Intel oneAPI BaseKit Install*

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

**Building PETSc with Intel oneAPI**

Source the file:

/opt/intel/oneapi/setvars.sh

Check that your compilers are there:

which icc

which ifort

then run the Python configure script provided by the PETSc team from within your “petsc” directory (typically “/home/YourName/petsc”. This command line will link the code the Math Kernel Library (MKL) from Intel, which is optimized for the Intel chip sets.

For Intel oneAPI, one can use either “icc” and “ifort”, or “mpiicc” and “mpif90”. The “mpi” versions seem to produce faster executable, and can be used even with the single processor option (--with-mpi=0) below

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

This will build the single processor optimized version of Crunch using the oneAPI-provided MKL library.

./configure --with-cc=mpiicc --with-cxx=mpiicpc --with-fc=mpif90 COPTFLAGS=" -g -O3" FOPTFLAGS=" -g -O3" CXXOPTFLAGS=" -g -O3" --with-debugging=0 --with-mpi=0 --with-blaslapack-dir=/opt/intel/oneapi/mkl PETSC\_ARCH=linux-oneAPI-opt

**Optimized Version using oneAPI (with icc and ifort instead of mpiicc and mpif90—tends tp be slower)**

This will build the single processor optimized version of Crunch using the oneAPI-provided MKL library, but with icc and ifort instead of the mpi versions (tends to be slower).

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

**Optimized Version using oneAPI (download “fblaslapack”)**

This will build the single processor optimized version of Crunch with a download of “blaslapack”.

./configure --with-cc=mpiicc --with-cxx=mpiicpc --with-fc=mpif90 COPTFLAGS=" -g -O3" FOPTFLAGS=" -g -O3" CXXOPTFLAGS=" -g -O3" --with-debugging=0 --with-mpi=0 --download-fblaslapack PETSC\_ARCH=linux-oneAPI-opt

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

This will build the single processor optimized version of Crunch.

./configure --with-cc=mpiicc --with-cxx=mpiicpc --with-fc=mpif90 COPTFLAGS=" -g -O3" --with-debugging=1 --with-mpi=0 --with-blaslapack-dir=/opt/intel/oneapi/mkl PETSC\_ARCH=linux-oneAPI-debug