# Compile from source (Ubuntu)

<https://dafoam.github.io/mydoc_installation_source.html>

**Note:** This section assumes you want to compile the latest DAFoam optimization package from the source on a Linux system. If you use the Docker image, there is no need to compile anything and you can skip this section. For DAFoam older versions, refer to [v2.2.10-](https://dafoam.github.io/mydoc_installation_source_2210.html), [v2.2.0-](https://dafoam.github.io/mydoc_installation_source_220.html), and [v1.0.0](https://dafoam.github.io/mydoc_installation_source_100.html).

The DAFoam package can be compiled with various dependency versions. Here we elaborate on how to compile it on **Ubuntu 20.04** using the dependencies shown in the following table.

| Ubuntu | Compiler | OpenMPI | mpi4py | PETSc | petsc4py | CGNS | Python | Numpy | Scipy | Cython |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 20.04.5 | gcc/9.4 | 3.1.6 | 3.1.3 | 3.14.6 | 3.14.1 | 4.1.2 | 3.8 | 1.21.2 | 1.7.1 | 0.29.21 |

To compile, you can just copy the code blocks in the following steps and run them on the terminal. If a code block contains multiple lines, copy all the lines and run them on the terminal. Make sure each step run successfully before going to the next one. The entire compilation may take a few hours, the most time-consuming part is to compile OpenFOAM.

## Prerequisites

Run this on terminal to install prerequisites:

sudo apt-get update && \

sudo apt-get install -y build-essential flex bison cmake zlib1g-dev libboost-system-dev libboost-thread-dev libreadline-dev libncurses-dev libxt-dev freeglut3-dev texinfo libscotch-dev libcgal-dev gfortran swig wget git vim cmake-curses-gui libfl-dev apt-utils libibverbs-dev ca-certificates pkg-config liblapack-dev libmetis-dev --no-install-recommends

## Root folder

First create a “dafoam” folder in your home directory. Then create a “loadDAFoam.sh” bash script and set up the root path $DAFOAM\_ROOT\_PATH. Finally, we will create the “packages”, “OpenFOAM”, and “repos” folders. We will compile and install everything in $DAFOAM\_ROOT\_PATH.

mkdir -p $HOME/dafoam && \

mkdir -p $HOME/dafoam/packages $HOME/dafoam/OpenFOAM $HOME/dafoam/OpenFOAM/sharedBins $HOME/dafoam/OpenFOAM/sharedLibs $HOME/dafoam/repos && \

echo '#!/bin/bash' > $HOME/dafoam/loadDAFoam.sh && \

echo '# DAFoam root path' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export DAFOAM\_ROOT\_PATH=$HOME/dafoam' >> $HOME/dafoam/loadDAFoam.sh && \

chmod 755 $HOME/dafoam/loadDAFoam.sh && \

. $HOME/dafoam/loadDAFoam.sh

## Python-3.8

Install Miniconda3 by running this command:

cd $HOME/dafoam/packages && \

wget https://repo.anaconda.com/miniconda/Miniconda3-py38\_4.10.3-Linux-x86\_64.sh && \

chmod 755 Miniconda3-py38\_4.10.3-Linux-x86\_64.sh && \

./Miniconda3-py38\_4.10.3-Linux-x86\_64.sh -b -p $HOME/dafoam/packages/miniconda3 && \

echo '# Miniconda3' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export PATH=$DAFOAM\_ROOT\_PATH/packages/miniconda3/bin:$PATH' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:$DAFOAM\_ROOT\_PATH/packages/miniconda3/lib' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export PYTHONUSERBASE=no-local-libs' >> $HOME/dafoam/loadDAFoam.sh && \

. $HOME/dafoam/loadDAFoam.sh

In the above, we use “export PYTHONUSERBASE=no-local-libs” to bypass the site-packages in user’s .local directory because they may conflict with the DAFoam packages. Then we can upgrade the pip utility:

pip install --upgrade pip && \

pip install numpy==1.21.2 && \

pip install scipy==1.7.1 && \

pip install cython==0.29.21 && \

pip install numpy-stl==2.16.0 && \

pip install pynastran==1.3.3 && \

pip install nptyping==1.4.4 && \

pip install swig==4.1.1 && \

pip install tensorflow-cpu==2.12

## OpenMPI-3.1.6

First append relevant environmental variables by running:

echo '# OpenMPI-3.1.6' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export MPI\_INSTALL\_DIR=$DAFOAM\_ROOT\_PATH/packages/openmpi-3.1.6/opt-gfortran' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:$MPI\_INSTALL\_DIR/lib' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export PATH=$MPI\_INSTALL\_DIR/bin:$PATH' >> $HOME/dafoam/loadDAFoam.sh&& \

. $HOME/dafoam/loadDAFoam.sh

Then, configure and build OpenMPI:

cd $HOME/dafoam/packages && \

wget https://download.open-mpi.org/release/open-mpi/v3.1/openmpi-3.1.6.tar.gz && \

tar -xvf openmpi-3.1.6.tar.gz && \

cd openmpi-3.1.6 && \

./configure --prefix=$MPI\_INSTALL\_DIR && \

make all install

To verify the installation, run:

mpicc -v

You should see the version of the compiled OpenMPI.

Finally, install mpi4py-3.1.3:

pip install mpi4py==3.1.3

## Petsc-3.14.6

First append relevant environmental variables by running:

echo '# Petsc-3.14.6' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export PETSC\_DIR=$DAFOAM\_ROOT\_PATH/packages/petsc-3.14.6' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export PETSC\_ARCH=real-opt' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:$PETSC\_DIR/$PETSC\_ARCH/lib' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export PETSC\_LIB=$PETSC\_DIR/$PETSC\_ARCH/lib' >> $HOME/dafoam/loadDAFoam.sh && \

. $HOME/dafoam/loadDAFoam.sh

Then, configure and compile:

cd $HOME/dafoam/packages && \

wget https://www.mcs.anl.gov/petsc/mirror/release-snapshots/petsc-3.14.6.tar.gz && \

tar -xvf petsc-3.14.6.tar.gz && \

cd petsc-3.14.6 && \

./configure --PETSC\_ARCH=real-opt --with-scalar-type=real --with-debugging=0 --download-metis=yes --download-parmetis=yes --download-superlu\_dist=yes --download-fblaslapack=yes --with-shared-libraries=yes --with-fortran-bindings=1 --with-cxx-dialect=C++11 && \

make PETSC\_DIR=$HOME/dafoam/packages/petsc-3.14.6 PETSC\_ARCH=real-opt all

Finally, install petsc4py-3.14.1:

cd $PETSC\_DIR/src/binding/petsc4py && \

pip install .

## CGNS-4.1.2

First append relevant environmental variables by running:

echo '# CGNS-4.1.2' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export CGNS\_HOME=$DAFOAM\_ROOT\_PATH/packages/CGNS-4.1.2/opt-gfortran' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export PATH=$PATH:$CGNS\_HOME/bin' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:$CGNS\_HOME/lib' >> $HOME/dafoam/loadDAFoam.sh && \

. $HOME/dafoam/loadDAFoam.sh

Then, configure and compile:

cd $HOME/dafoam/packages && \

wget https://github.com/CGNS/CGNS/archive/v4.1.2.tar.gz && \

tar -xvaf v4.1.2.tar.gz && \

cd CGNS-4.1.2 && \

mkdir -p build && \

cd build && \

cmake .. -DCGNS\_ENABLE\_FORTRAN=1 -DCMAKE\_INSTALL\_PREFIX=$CGNS\_HOME -DCGNS\_BUILD\_CGNSTOOLS=0 && \

make all install

## IPOPT-3.13.2

Download Ipopt-3.13.2 and set up the relevant environmental variables to loadDAFoam.sh by running:

echo '# Ipopt' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export IPOPT\_DIR=$DAFOAM\_ROOT\_PATH/packages/Ipopt' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:$IPOPT\_DIR/lib' >> $HOME/dafoam/loadDAFoam.sh && \

. $HOME/dafoam/loadDAFoam.sh

Next, compiles the ThirdParty dependencies Metis and Mumps by running:

cd $HOME/dafoam/packages && \

git clone -b stable/3.13 https://github.com/coin-or/Ipopt.git && \

cd $IPOPT\_DIR && \

git clone -b stable/2.1 https://github.com/coin-or-tools/ThirdParty-Mumps.git && \

cd ThirdParty-Mumps && \

./get.Mumps && \

./configure --prefix=$IPOPT\_DIR && \

make && \

make install

Finally, compile Ipopt by running:

cd $IPOPT\_DIR && \

mkdir build && \

cd build && \

../configure --prefix=${IPOPT\_DIR} --disable-java --with-mumps --with-mumps-lflags="-L${IPOPT\_DIR}/lib -lcoinmumps" --with-mumps-cflags="-I${IPOPT\_DIR}/include/coin-or/mumps" && \

make && \

make install

## MACH-Aero framework

The supported repo versions in the MACH-Aero framework for DAFoam-v3.0.7 is as follows

| baseclasses | pySpline | pyGeo | multipoint | pyHyp | cgnsUtilities | IDWarp | pyOptSparse | pyOFM | DAFoam |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| v1.6.1 | v1.5.2 | v1.12.2 | v1.4.0 | v2.5.0 | v2.6.0 | v2.6.0 | v2.10.1 | v1.2.1 | v3.0.7 |

Now run this command to install all the repos for MACH-Aero:

. $HOME/dafoam/loadDAFoam.sh && \

cd $HOME/dafoam/repos && \

wget https://github.com/mdolab/baseclasses/archive/v1.6.1.tar.gz -O baseclasses.tar.gz && \

tar -xvf baseclasses.tar.gz && cd baseclasses-1.6.1 && pip install . && \

cd $HOME/dafoam/repos && \

wget https://github.com/mdolab/pyspline/archive/v1.5.2.tar.gz -O pyspline.tar.gz && \

tar -xvf pyspline.tar.gz && cd pyspline-1.5.2 && \

cp config/defaults/config.LINUX\_GFORTRAN.mk config/config.mk && \

make && pip install . && \

cd $HOME/dafoam/repos && \

wget https://github.com/mdolab/pygeo/archive/4641625e28b0bae97e05351c23d9083a693734d3.tar.gz -O pygeo.tar.gz && \

tar -xvf pygeo.tar.gz && mv pygeo-\* pygeo && cd pygeo && pip install . && \

cd $HOME/dafoam/repos && \

wget https://github.com/mdolab/multipoint/archive/v1.4.0.tar.gz -O multipoint.tar.gz && \

tar -xvf multipoint.tar.gz && cd multipoint-1.4.0 && pip install . && \

cd $HOME/dafoam/repos && \

wget https://github.com/mdolab/pyhyp/archive/v2.5.0.tar.gz -O pyhyp.tar.gz && \

tar -xvf pyhyp.tar.gz && cd pyhyp-2.5.0 && \

cp -r config/defaults/config.LINUX\_GFORTRAN\_OPENMPI.mk config/config.mk && \

make && pip install . && \

cd $HOME/dafoam/repos && \

wget https://github.com/mdolab/cgnsutilities/archive/v2.6.0.tar.gz -O cgnsutilities.tar.gz && \

tar -xvf cgnsutilities.tar.gz && cd cgnsutilities-2.6.0 && \

cp config/defaults/config.LINUX\_GFORTRAN.mk config/config.mk && \

make && pip install . && \

cd $HOME/dafoam/repos && \

wget https://github.com/mdolab/idwarp/archive/v2.6.0.tar.gz -O idwarp.tar.gz && \

tar -xvf idwarp.tar.gz && cd idwarp-2.6.0 && \

cp -r config/defaults/config.LINUX\_GFORTRAN\_OPENMPI.mk config/config.mk && \

make && pip install . && \

cd $HOME/dafoam/repos && \

wget https://github.com/mdolab/pyoptsparse/archive/v2.10.1.tar.gz -O pyoptsparse.tar.gz && \

tar -xvf pyoptsparse.tar.gz && cd pyoptsparse-2.10.1 && \

pip install .

## OpenFOAM-v1812

**There are three versions of OpenFOAM to compile: original, reverse-mode AD (ADR), and forward-mode AD (ADF).** The reverse-mode AD enables the JacobianFree adjoint option, and the forward-mode AD enables the brute-force AD for verifying the adjoint accuracy.

Build Original

Run the following:

cd $HOME/dafoam/OpenFOAM && \

wget https://sourceforge.net/projects/openfoam/files/v1812/OpenFOAM-v1812.tgz/download -O OpenFOAM-v1812.tgz && \

wget https://sourceforge.net/projects/openfoam/files/v1812/ThirdParty-v1812.tgz/download -O ThirdParty-v1812.tgz && \

tar -xvf OpenFOAM-v1812.tgz && \

tar -xvf ThirdParty-v1812.tgz && \

cd $HOME/dafoam/OpenFOAM/OpenFOAM-v1812 && \

sed -i 's/$HOME/$DAFOAM\_ROOT\_PATH/g' etc/bashrc && \

wget https://github.com/DAFoam/files/releases/download/v1.0.0/UPstream.C && \

mv UPstream.C src/Pstream/mpi/UPstream.C && \

echo '# OpenFOAM-v1812' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'source $DAFOAM\_ROOT\_PATH/OpenFOAM/OpenFOAM-v1812/etc/bashrc' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export LD\_LIBRARY\_PATH=$DAFOAM\_ROOT\_PATH/OpenFOAM/sharedLibs:$LD\_LIBRARY\_PATH' >> $HOME/dafoam/loadDAFoam.sh && \

echo 'export PATH=$DAFOAM\_ROOT\_PATH/OpenFOAM/sharedBins:$PATH' >> $HOME/dafoam/loadDAFoam.sh && \

. $HOME/dafoam/loadDAFoam.sh && \

export WM\_NCOMPPROCS=4 && \

./Allwmake

**Note:** In the above command, we replaced the OpenFOAM-v1812’s built-in UPstream.C file with a customized one because we need to prevent OpenFOAM from calling the MPI\_Finialize function when wrapping OpenFOAM functions using Cython.

**Note:** The above command will compile OpenFOAM using 4 CPU cores. If you want to compile OpenFOAM using more cores, change the WM\_NCOMPPROCS parameter before running ./Allwmake

Finally, verify the installation by running:

simpleFoam -help

It should see some basic information of OpenFOAM.

Build Reverse Mode AD

Run the following:

cd $HOME/dafoam/OpenFOAM && \

wget https://github.com/DAFoam/OpenFOAM-v1812-AD/archive/v1.3.0.tar.gz -O OpenFOAM-v1812-AD.tgz && \

tar -xvf OpenFOAM-v1812-AD.tgz && mv OpenFOAM-v1812-AD-\* OpenFOAM-v1812-ADR && \

cd $HOME/dafoam/OpenFOAM/OpenFOAM-v1812-ADR && \

sed -i 's/WM\_PROJECT\_VERSION=v1812-AD/WM\_PROJECT\_VERSION=v1812-ADR/g' etc/bashrc && \

sed -i 's/$HOME/$DAFOAM\_ROOT\_PATH/g' etc/bashrc && \

sed -i 's/export WM\_CODI\_AD\_MODE=CODI\_AD\_FORWARD/export WM\_CODI\_AD\_MODE=CODI\_AD\_REVERSE/g' etc/bashrc && \

. $HOME/dafoam/loadDAFoam.sh && \

source etc/bashrc && \

export WM\_NCOMPPROCS=4 && \

./Allwmake 2> warningLog.txt

Then, verify the installation by running:

DASimpleFoamReverseAD -help

It should see some basic information of DASimpleFoamReverseAD.

**Note:** We use CodiPack to differentiate the OpenFOAM libraries. During the compliation, it will generate a lot of warning messages, which are saved to the warningLog.txt file. After the compilation is done, remember to delete this warning file, which can be larger than 1 GB.

After OpenFOAM-v1812-ADR is compiled and verified, we need to link all the compiled AD libraries to the original OpenFOAM-v1812 folder. Note that we need to link the relative path because we want this to be portable.

cd $HOME/dafoam/OpenFOAM/OpenFOAM-v1812/platforms/\*/lib && \

ln -s ../../../../OpenFOAM-v1812-ADR/platforms/\*/lib/\*.so . && \

cd $HOME/dafoam/OpenFOAM/OpenFOAM-v1812/platforms/\*/lib/dummy && \

ln -s ../../../../../OpenFOAM-v1812-ADR/platforms/\*/lib/dummy/\*.so . && \

cd $HOME/dafoam/OpenFOAM/OpenFOAM-v1812/platforms/\*/lib/openmpi-system && \

ln -s ../../../../../OpenFOAM-v1812-ADR/platforms/\*/lib/openmpi-system/\*.so .

Build Forward Mode AD

Run the following:

cd $HOME/dafoam/OpenFOAM && \

wget https://github.com/DAFoam/OpenFOAM-v1812-AD/archive/v1.3.0.tar.gz -O OpenFOAM-v1812-AD.tgz && \

tar -xvf OpenFOAM-v1812-AD.tgz && mv OpenFOAM-v1812-AD-\* OpenFOAM-v1812-ADF && \

cd $HOME/dafoam/OpenFOAM/OpenFOAM-v1812-ADF && \

sed -i 's/WM\_PROJECT\_VERSION=v1812-AD/WM\_PROJECT\_VERSION=v1812-ADF/g' etc/bashrc && \

sed -i 's/$HOME/$DAFOAM\_ROOT\_PATH/g' etc/bashrc && \

. $HOME/dafoam/loadDAFoam.sh && \

source etc/bashrc && \

export WM\_NCOMPPROCS=4 && \

./Allwmake 2> warningLog.txt

After OpenFOAM-v1812-ADF is compiled and verified, we need to link all the compiled AD libraries to the original OpenFOAM-v1812 folder. Note that we need to link the relative path because we want this to be portable.

cd $HOME/dafoam/OpenFOAM/OpenFOAM-v1812/platforms/\*/lib && \

ln -s ../../../../OpenFOAM-v1812-ADF/platforms/\*/lib/\*.so . && \

cd $HOME/dafoam/OpenFOAM/OpenFOAM-v1812/platforms/\*/lib/dummy && \

ln -s ../../../../../OpenFOAM-v1812-ADF/platforms/\*/lib/dummy/\*.so . && \

cd $HOME/dafoam/OpenFOAM/OpenFOAM-v1812/platforms/\*/lib/openmpi-system && \

ln -s ../../../../../OpenFOAM-v1812-ADF/platforms/\*/lib/openmpi-system/\*.so .

Once done, unset the AD environment, and re-source the original OpenFOAM-v1812. **This step is needed everytime you compile an AD versions of DAFoam or OpenFOAM!**

unset WM\_CODI\_AD\_MODE && \

. $HOME/dafoam/loadDAFoam.sh

## pyOFM-1.2.1

Compile pyOFM by running:

cd $HOME/dafoam/repos && \

wget https://github.com/mdolab/pyofm/archive/v1.2.1.tar.gz -O pyofm.tar.gz && \

tar -xvf pyofm.tar.gz && cd pyofm-1.2.1 && \

. $HOME/dafoam/loadDAFoam.sh && \

make && pip install .

## DAFoam-3.0.7

Similar to OpenFOAM, we need to compile three versions of DAFoam: original, reverse-mode AD (ADR), and forward-mode AD (ADF). It can be done by running the following commands:

cd $HOME/dafoam/repos && \

wget https://github.com/mdolab/dafoam/archive/v3.0.7.tar.gz -O dafoam.tar.gz && \

tar -xvf dafoam.tar.gz && cd dafoam-\* && \

. $HOME/dafoam/loadDAFoam.sh && \

export DAFOAM\_NO\_WARNINGS=1 && \

./Allmake && \

source $HOME/dafoam/OpenFOAM/OpenFOAM-v1812-ADR/etc/bashrc && \

./Allclean && ./Allmake && \

source $HOME/dafoam/OpenFOAM/OpenFOAM-v1812-ADF/etc/bashrc && \

./Allclean && ./Allmake && \

pip install .

Once done, unset the AD environment, and re-source the original OpenFOAM-v1812. **This step is needed everytime you compile an AD version of DAFoam!**

unset WM\_CODI\_AD\_MODE && \

. $HOME/dafoam/loadDAFoam.sh

**Note:** Before running any jobs, source the loadDAFoam.sh file to load DAFoam environment!

## MDAO packages

To perform multi-displinary design optimization, we need to install the following packages:

OpenMDAO-3.26

[OpenMDAO](https://openmdao.org/) is an open-source multidisciplinary optimization framework.

. $HOME/dafoam/loadDAFoam.sh && \

pip install openmdao==3.26

Mphys

[Mphys](https://github.com/OpenMDAO/mphys) is an interface that facilitate the interaction between low- and high-fidelity tools within OpenMDAO.

. $HOME/dafoam/loadDAFoam.sh && \

cd $HOME/dafoam/repos && \

wget https://github.com/OpenMDAO/mphys/archive/b6db107d05937d95a46b392b6f5759677a93e46d.tar.gz -O mphys.tar.gz && \

tar -xvf mphys.tar.gz && mv mphys-\* mphys && \

cd mphys && pip install -e .

FUNtoFEM-0.3

[FUNtoFEM](https://github.com/smdogroup/funtofem) is a generic aeroelastic analysis and adjoint-based gradient evaluation tools.

. $HOME/dafoam/loadDAFoam.sh && \

cd $HOME/dafoam/repos && \

wget https://github.com/smdogroup/funtofem/archive/refs/tags/v0.3.tar.gz -O funtofem.tar.gz && \

tar -xvf funtofem.tar.gz && mv funtofem-\* funtofem && \

cd funtofem && cp Makefile.in.info Makefile.in && \

sed -i "s/git/dafoam\/repos/g" Makefile.in && \

make && pip install -e .

TACS-3.2.1

[TACS](https://github.com/smdogroup/tacs) is a finite-element library for analysis and adjoint-based gradient evaluation

. $HOME/dafoam/loadDAFoam.sh && \

cd $HOME/dafoam/repos && \

wget https://github.com/smdogroup/tacs/archive/refs/tags/v3.2.1.tar.gz -O tacs.tar.gz && \

tar -xvf tacs.tar.gz && mv tacs-\* tacs && \

cd tacs/extern && \

wget https://github.com/DAFoam/files/releases/download/TACS\_Extern/TACS\_extern.tar.gz && tar -xzf TACS\_extern.tar.gz && \

rm -rf metis-4.0.3\* && \

wget https://github.com/DAFoam/files/releases/download/TACS\_Extern/metis-5.1.0.tar.gz && \

tar -czvf TACS\_extern.tar.gz metis\*.tar.gz UFconfig\*.tar.gz AMD\*.tar.gz &&\

tar -xzf metis\*.tar.gz && \

cd metis-5.1.0 && make config prefix=$HOME/dafoam/repos/tacs/extern/metis/ CFLAGS="-fPIC" && make install && \

cd ../../ && \

cp Makefile.in.info Makefile.in && \

ls && \

sed -i "s/git/dafoam\/repos/g" Makefile.in && \

make && pip install -e . && \

cd extern/f5tovtk && make && cp f5tovtk $HOME/dafoam/OpenFOAM/sharedBins

## DAFoam regression tests

To verify the DAFoam installation, you can run the regression tests:

cd $HOME/dafoam/repos/dafoam-\*/tests && ./Allrun

The regression tests should take less than 30 minutes. The test progress will be printed to screen. Make sure you see this at the end:

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* All DAFoam tests passed! \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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

In summary, here is the folder structures for all the installed packages:

$HOME/dafoam

loadDAFoam.sh

- OpenFOAM

- OpenFOAM-v1812

- OpenFOAM-v1812-ADF

- OpenFOAM-v1812-ADR

- ThirdParty-v1812

- packages

- Ipopt

- miniconda3

- CGNS-4.1.2

- openmpi-3.1.6

- petsc-3.14.6

- repos

- baseclasses

- cgnsutilities

- dafoam

- funtofem

- idwarp

- multipoint

- mphys

- pygeo

- pyhyp

- pyofm

- pyoptsparse

- pyspline

- tacs

The loadDAFoam.sh file should look like this:

#!/bin/bash

# DAFoam root path

export DAFOAM\_ROOT\_PATH=$HOME/dafoam

# Miniconda3

export PATH=$DAFOAM\_ROOT\_PATH/packages/miniconda3/bin:$PATH

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:$DAFOAM\_ROOT\_PATH/packages/miniconda3/lib

export PYTHONUSERBASE=no-local-libs

# OpenMPI-3.1.6

export MPI\_INSTALL\_DIR=$DAFOAM\_ROOT\_PATH/packages/openmpi-3.1.6/opt-gfortran

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:$MPI\_INSTALL\_DIR/lib

export PATH=$MPI\_INSTALL\_DIR/bin:$PATH

# PETSC

export PETSC\_DIR=$DAFOAM\_ROOT\_PATH/packages/petsc-3.14.6

export PETSC\_ARCH=real-opt

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:$PETSC\_DIR/$PETSC\_ARCH/lib

export PETSC\_LIB=$PETSC\_DIR/$PETSC\_ARCH/lib

# CGNS-4.1.2

export CGNS\_HOME=$DAFOAM\_ROOT\_PATH/packages/CGNS-4.1.2/opt-gfortran

export PATH=$PATH:$CGNS\_HOME/bin

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:$CGNS\_HOME/lib

# Ipopt

export IPOPT\_DIR=$DAFOAM\_ROOT\_PATH/packages/Ipopt

export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:$IPOPT\_DIR/lib

# OpenFOAM-v1812

source $DAFOAM\_ROOT\_PATH/OpenFOAM/OpenFOAM-v1812/etc/bashrc

export LD\_LIBRARY\_PATH=$DAFOAM\_ROOT\_PATH/OpenFOAM/sharedLibs:$LD\_LIBRARY\_PATH

export PATH=$DAFOAM\_ROOT\_PATH/OpenFOAM/sharedBins:$PATH

## Make the DAFoam package portable (optional)

This step is only needed if you want to change the root path of your installation, e.g., copy your compiled DAFoam packages to another directory.

The only thing you need to do is to modify the interpreter lines “#!” for files in $HOME/dafoam/packages/miniconda3/. This is because Miniconda hard codes the Python path, so we need to chagne it to “#!/usr/bin/env python”

First find an example of the hard-coded interpreter line from $HOME/dafoam/packages/miniconda3/bin/conda. Run this command

head -1 $HOME/dafoam/packages/miniconda3/bin/conda

You may see an output like this:

#!/home/replace\_this\_with\_your\_username/dafoam/packages/miniconda3/bin/python

Then run this command to replace all the hard-coded interpreter lines:

sed -i 's,^#\!/home/replace\_this\_with\_your\_username/dafoam/packages/miniconda3/bin/python,#!/usr/bin/env python,g' $HOME/dafoam/packages/miniconda3/\*/\*

Finally, you can change the DAFOAM\_ROOT\_PATH value (in loadDAFoam.sh) to your new directory, source the “loadDAFoam.sh” script again, and run DAFoam without compiling everything again.