# Compile from source (HPC)

<https://dafoam.github.io/mydoc_installation_source_hpc.html#setup-environment-loader>

**Note:** This section provides instructions on compiling the DAFoam optimization package on an HPC system (Stampede 2) with Intel compilers. Installation on other intel-based clusters may differ, but follows the same general procedure

The DAFoam package can be compiled with various dependency versions. For this installation, we will use the following modules already installed on Stampede 2:

| TACC | git | intel | libfabric | impi | boost | petsc | python3 |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N / A | 2.24.1 | 18.0.2 | 1.7.0 | 18.0.2 | 1.68 | 3.11-nohdf5 | 3.7.0 |

For this installtion, we will store all of the DAFoam installation files within $WORK/DAFoam. Follow this guide line-by-line, reproducing each command on your terminal. Ensure that each command exits successfully before moving to the next one. Once the installation is complete, the directory structure of the installation should be the following:

DAFoam/

└───DAFOAM\_VENV/

│

└───OpenFOAM/

│ └───OpenFOAM-v1812/

│ └───OpenFOAM-v1812-ADF/

│ └───OpenFOAM-v1812-ADR/

│ └───sharedBins/

│ └───sharedLibs/

│ └───ThirdParty-v1812/

│

└───packages/

│ └───CGNS-3.3.0/

│

└───repos/

│ └───baseclasses/

│ └───cgnsutilities/

│ └───dafoam/

│ └───idwarp/

│ └───mphys/

│ └───multipoint/

│ └───pygeo/

│ └───pyhyp/

│ └───pyofm/

│ └───pyoptsparse/

│ └───pyspline/

│

└───ENV\_DAFOAM.sh/

Begin by navigating to the $WORK directory and generating the DAFoam code directory and its subdirectory structure:

mkdir DAFoam

cd DAFoam

mkdir OpenFOAM packages repos

## Setup Environment Loader

To simplify the process of initializing the DAFoam environment when logging into Stampede, we will create a bash script that automates the steps required to load the required modules, export the required environment variables, and source the Python virtual environment. To generate this script, enter $WORK/DAFoam and create the file: ENV\_DAFOAM.sh. Add the following to ENV\_DAFOAM.sh:

#!/bin/bash

echo "=============== Initializing DAFoam Environment ==============="

DIR="$( cd "$( dirname "${BASH\_SOURCE[0]}" )" >/dev/null 2>&1 && pwd )"

# Modules

echo "Loading Modules"

module purge

module load TACC

module load git/2.24.1

module load intel/18.0.2

module load libfabric/1.7.0

module load impi/18.0.2

module load boost/1.68

module load petsc/3.11-nohdf5

module load python3/3.7.0

# DAFoam

echo "Initializing DAFoam"

export DAFOAM\_ROOT\_PATH=$DIR

# CGNS

echo "Initializing CGNS"

export CGNS\_HOME=$DIR/packages/CGNS-3.3.0/opt-gfortran

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

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

# OpenFOAM

echo "Initializing OpenFOAM"

source $DIR/OpenFOAM/OpenFOAM-v1812/etc/bashrc

export LD\_LIBRARY\_PATH=$DIR/OpenFOAM/sharedLibs:$LD\_LIBRARY\_PATH

PATH=$DIR/OpenFOAM/sharedBins:$PATH

# Python Virtual Environment

echo "Initializing Python Virtual Environment"

source $DIR/DAFOAM\_VENV/bin/activate

echo "============= Done Initializing DAFOAM Environment ============="

This file will be responsible for loading the DAFoam environment once everything is installed. A command to source the file can be added to your bashrc, or you can call the file by navigating to its directory and running the following command (run this command now):

. ENV\_DAFOAM.sh

This command will output information stating that the DAFoam environment is being initialized. Since DAFoam and the assorted environment components are not yet installed, the script will output some errors.

## Create Python Virtual Environment

To modularize the DAFoam environment, we will use a Python virtual environment. Create a Python virtual environment using the package venv:

python3 -m venv DAFOAM\_VENV

Re-source the environment initialization script to activate the virtual environment:

. ENV\_DAFOAM.sh

After re-sourcing the environment initialization script your terminal prompt should include the label (ENV\_DAFOAM). Install the Python dependencies in the virtual environment:

pip3 install --upgrade pip

pip3 install numpy==1.21.2

pip3 install scipy==1.7.1

pip3 install cython==0.29.21

pip3 install numpy-stl==2.16.0

pip3 install petsc4py==3.11.0

## Install OpenFOAM

We will begin by 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. These packages should be installed in the OpenFoam/ directory.

Build Original

Begin by downloading OpenFOAM-v1812 and the associated third-party libraries, and unzip the tarballs:

cd OpenFOAM

wget https://sourceforge.net/projects/openfoamplus/files/v1812/OpenFOAM-v1812.tgz/download -O OpenFOAM-v1812.tgz

wget https://sourceforge.net/projects/openfoamplus/files/v1812/ThirdParty-v1812.tgz/download -O ThirdParty-v1812.tgz

tar -xvf OpenFOAM-v1812.tgz

tar -xvf ThirdParty-v1812.tgz

Enter the OpenFOAM-v1812 directory:

cd OpenFOAM-v1812

Replace the built-in UPstream.C file with a customized version:

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

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

We will make several modifications to the configuration files within OpenFOAM. The following steps show which files need to be changed and what changes are required.

Edit OpenFOAM-v1812/etc/bashrc:

projectDir="$HOME/OpenFOAM/OpenFoam-$WM\_PROJECT\_VERSION"

->

projectDir="$DAFOAM\_ROOT\_PATH/OpenFOAM/OpenFoam-$WM\_PROJECT\_VERSION"

-------------------------------------------------

export WM\_COMPILER=Gcc

->

export WM\_COMPILER=Icc

-------------------------------------------------

export WM\_MPLIB=SYSTEMOPENMPI

->

export WM\_MPLIB=INTELMPI

-------------------------------------------------

foamOldDirs="$WM\_PROJECT\_DIR $WM\_THIRD\_PARTY\_DIR \

$HOME/$WM\_PROJECT/$USER $FOAM\_USER\_APPBIN $FOAM\_USER\_LIBBIN \

$WM\_PROJECT\_SITE $FOAM\_SITE\_APPBIN $FOAM\_SITE\_LIBBIN"

->

foamOldDirs="$WM\_PROJECT\_DIR $WM\_THIRD\_PARTY\_DIR \

$DAFOAM\_ROOT\_PATH/$WM\_PROJECT/$USER $FOAM\_USER\_APPBIN $FOAM\_USER\_LIBBIN \

$WM\_PROJECT\_SITE $FOAM\_SITE\_APPBIN $FOAM\_SITE\_LIBBIN"

-------------------------------------------------

export WM\_PROJECT\_USER\_DIR=$HOME/$WM\_PROJECT/$USER-$WM\_PROJECT\_VERSION

->

export WM\_PROJECT\_USER\_DIR=$DAFOAM\_ROOT\_PATH/$WM\_PROJECT/$USER-$WM\_PROJECT\_VERSION

Edit OpenFOAM-v1812/etc/config.sh/CGAL:

boost\_version=boost\_1\_64\_0

->

boost\_version=boost-system

-------------------------------------------------

export BOOST\_ARCH\_PATH=$WM\_THIRD\_PARTY\_DIR/platforms/$WM\_ARCH$WM\_COMPILER/$boost\_version

->

export BOOST\_ARCH\_PATH=$TACC\_BOOST\_DIR

Edit OpenFOAM-v1812/etc/config.sh/settings:

export WM\_CC="gcc"

export WM\_CXX="g++"

export WM\_CFLAGS="-fPIC"

export WM\_CXXFLAGS="-fPIC -std=c++11"

->

export WM\_CC="mpicc"

export WM\_CXX="mpicxx"

export WM\_CFLAGS="-O3 -fPIC"

export WM\_CXXFLAGS="-O3 -fPIC -std=c++11"

-------------------------------------------------

64)

WM\_ARCH=linux64

export WM\_COMPILER\_LIB\_ARCH=64

export WM\_CFLAGS="$WM\_CFLAGS -m64"

export WM\_CXXFLAGS="$WM\_CXXFLAGS -m64"

export WM\_LDFLAGS="-m64"

->

64)

WM\_ARCH=linux64

export WM\_COMPILER\_LIB\_ARCH=64

export WM\_CFLAGS="$WM\_CFLAGS"

export WM\_CXXFLAGS="$WM\_CXXFLAGS"

export WM\_LDFLAGS="-O3"

Edit OpenFOAM-v1812/wmake/rules/linux64Icc/c:

cc = icc

->

cc = mpicc

Edit OpenFOAM-v1812/wmake/rules/linux64Icc/c++:

CC = icpc -std=c++11 -fp-trap=common -fp-model precise

->

CC = mpicxx -std=c++11 -fp-trap=common -fp-model precise

Copy mplibINTELMPI:

cp OpenFOAM-v1812/wmake/rules/linux64Icc/mplibINTELMPI OpenFOAM-v1812/wmake/rules/General/mplibINTELMPI

Edit ThirdParty-v1812/makeCGAL:

: ${BOOST\_ARCH\_PATH:=$installBASE/$boostPACKAGE} # Fallback

->

: ${BOOST\_ARCH\_PATH:=$TACC\_BOOST\_DIR} # Fallback

Once the configuration file modifications are complete, source the OpenFOAM bashrc file:

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

We will begin the installation by making CGAL. To do this, run the following command:

cd $WM\_THIRD\_PARTY\_DIR and ./makeCGAL > log.mkcgal 2>&1

Once the CGAL installation is complete, we will install OpenFOAM using the following command:

cd $WM\_PROJECT\_DIR and ./Allwmake > log.make 2>&1

Once the original version of OpenFOAM is installed, we will install the AD versions of the code. These packages should be in the OpenFOAM/ directory. Download the AD version of OpenFOAM using the following command:

cd $DAFOAM\_ROOT\_PATH/OpenFOAM

wget https://github.com/DAFoam/OpenFOAM-v1812-AD/archive/v1.2.9.tar.gz -O OpenFOAM-v1812-AD.tgz

Reverse Mode

Unzip the downloaded AD code, move the AD version of OpenFOAM to mark it as reverse-mode, and enter its code directory:

tar -xvf OpenFOAM-v1812-AD.tgz

mv OpenFOAM-v1812-AD-\* OpenFOAM-v1812-ADR

cd OpenFOAM-v1812-ADR

As with the original version of OpenFOAM, we need to make several modifications to the configuration files within the reverse-mode AD version of OpenFOAM. The following steps show which files need to be changed and what changes are required.

Edit OpenFOAM-v1812-ADR/etc/bashrc:

export WM\_PROJECT\_VERSION=v1812-AD

->

export WM\_PROJECT\_VERSION=v1812-ADR

-------------------------------------------------

projectDir="$HOME/OpenFOAM/OpenFoam-$WM\_PROJECT\_VERSION"

->

projectDir="$DAFOAM\_ROOT\_PATH/OpenFOAM/OpenFoam-$WM\_PROJECT\_VERSION"

-------------------------------------------------

export WM\_COMPILER=Gcc

->

export WM\_COMPILER=Icc

-------------------------------------------------

export WM\_MPLIB=SYSTEMOPENMPI

->

export WM\_MPLIB=INTELMPI

-------------------------------------------------

export WM\_CODI\_AD\_MODE=CODI\_AD\_FORWARD

->

export WM\_CODI\_AD\_MODE=CODI\_AD\_REVERSE

-------------------------------------------------

foamOldDirs="$WM\_PROJECT\_DIR $WM\_THIRD\_PARTY\_DIR \

$HOME/$WM\_PROJECT/$USER $FOAM\_USER\_APPBIN $FOAM\_USER\_LIBBIN \

$WM\_PROJECT\_SITE $FOAM\_SITE\_APPBIN $FOAM\_SITE\_LIBBIN"

->

foamOldDirs="$WM\_PROJECT\_DIR $WM\_THIRD\_PARTY\_DIR \

$DAFOAM\_ROOT\_PATH/$WM\_PROJECT/$USER $FOAM\_USER\_APPBIN $FOAM\_USER\_LIBBIN \

$WM\_PROJECT\_SITE $FOAM\_SITE\_APPBIN $FOAM\_SITE\_LIBBIN"

-------------------------------------------------

export WM\_PROJECT\_USER\_DIR=$HOME/$WM\_PROJECT/$USER-$WM\_PROJECT\_VERSION

->

export WM\_PROJECT\_USER\_DIR=$DAFOAM\_ROOT\_PATH/$WM\_PROJECT/$USER-$WM\_PROJECT\_VERSION

Edit OpenFOAM-v1812-ADR/etc/config.sh/CGAL:

boost\_version=boost\_1\_64\_0

->

boost\_version=boost-system

-------------------------------------------------

export BOOST\_ARCH\_PATH=$WM\_THIRD\_PARTY\_DIR/platforms/$WM\_ARCH$WM\_COMPILER/$boost\_version

->

export BOOST\_ARCH\_PATH=$TACC\_BOOST\_DIR

Edit OpenFOAM-v1812-ADR/etc/config.sh/settings:

export WM\_CC="gcc"

export WM\_CXX="g++"

export WM\_CFLAGS="-fPIC"

export WM\_CXXFLAGS="-fPIC -std=c++11"

->

export WM\_CC="mpicc"

export WM\_CXX="mpicxx"

export WM\_CFLAGS="-O3 -fPIC"

export WM\_CXXFLAGS="-O3 -fPIC -std=c++11"

-------------------------------------------------

64)

WM\_ARCH=linux64

export WM\_COMPILER\_LIB\_ARCH=64

export WM\_CFLAGS="$WM\_CFLAGS -m64"

export WM\_CXXFLAGS="$WM\_CXXFLAGS -m64"

export WM\_LDFLAGS="-m64"

->

64)

WM\_ARCH=linux64

export WM\_COMPILER\_LIB\_ARCH=64

export WM\_CFLAGS="$WM\_CFLAGS"

export WM\_CXXFLAGS="$WM\_CXXFLAGS"

export WM\_LDFLAGS="-O3"

Edit OpenFOAM-v1812-ADR/wmake/rules/linux64Icc/c:

cc = icc

->

cc = mpicc

Edit OpenFOAM-v1812-ADR/wmake/rules/linux64Icc/c++:

CC = icpc -std=c++11 -fp-trap=common -fp-model precise

->

CC = mpicxx -std=c++11 -fp-trap=common -fp-model precise

Copy mplibINTELMPI:

cp OpenFOAM-v1812-ADR/wmake/rules/linux64Icc/mplibINTELMPI OpenFOAM-v1812-ADR/wmake/rules/General/mplibINTELMPI

Once the configuration file modifications are complete, source the OpenFOAM reverse-mode bashrc file:

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

Install OpenFOAM reverse-mode AD using the following command:

cd $WM\_PROJECT\_DIR and ./Allwmake > log.make 2>&1

Once the reverse-mode AD version of the code is build, we need to link it to the original version. To do this, run the following commands to create soft links:

cd $DAFOAM\_ROOT\_PATH/OpenFOAM/OpenFOAM-v1812/platforms/\*/lib

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

cd $DAFOAM\_ROOT\_PATH/OpenFOAM/OpenFOAM-v1812/platforms/\*/lib/dummy

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

cd $DAFOAM\_ROOT\_PATH/OpenFOAM/OpenFOAM-v1812/platforms/\*/lib/mpi

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

Forward Mode

Again unzip the AD code, move the AD version of OpenFOAM to mark it as forward-mode, and enter its code directory:

tar -xvf OpenFOAM-v1812-AD.tgz

mv OpenFOAM-v1812-AD-\* OpenFOAM-v1812-ADF

cd OpenFOAM-v1812-ADF

As with the original and reverse-mode versions of OpenFOAM, we need to make several modifications to the configuration files within the forward-mode AD version of OpenFOAM. The following steps show which files need to be changed and what changes are required.

Edit OpenFOAM-v1812-ADF/etc/bashrc:

export WM\_PROJECT\_VERSION=v1812-AD

->

export WM\_PROJECT\_VERSION=v1812-ADF

-------------------------------------------------

projectDir="$HOME/OpenFOAM/OpenFoam-$WM\_PROJECT\_VERSION"

->

projectDir="$DAFOAM\_ROOT\_PATH/OpenFOAM/OpenFoam-$WM\_PROJECT\_VERSION"

-------------------------------------------------

export WM\_COMPILER=Gcc

->

export WM\_COMPILER=Icc

-------------------------------------------------

export WM\_MPLIB=SYSTEMOPENMPI

->

export WM\_MPLIB=INTELMPI

-------------------------------------------------

foamOldDirs="$WM\_PROJECT\_DIR $WM\_THIRD\_PARTY\_DIR \

$HOME/$WM\_PROJECT/$USER $FOAM\_USER\_APPBIN $FOAM\_USER\_LIBBIN \

$WM\_PROJECT\_SITE $FOAM\_SITE\_APPBIN $FOAM\_SITE\_LIBBIN"

->

foamOldDirs="$WM\_PROJECT\_DIR $WM\_THIRD\_PARTY\_DIR \

$DAFOAM\_ROOT\_PATH/$WM\_PROJECT/$USER $FOAM\_USER\_APPBIN $FOAM\_USER\_LIBBIN \

$WM\_PROJECT\_SITE $FOAM\_SITE\_APPBIN $FOAM\_SITE\_LIBBIN"

-------------------------------------------------

export WM\_PROJECT\_USER\_DIR=$HOME/$WM\_PROJECT/$USER-$WM\_PROJECT\_VERSION

->

export WM\_PROJECT\_USER\_DIR=$DAFOAM\_ROOT\_PATH/$WM\_PROJECT/$USER-$WM\_PROJECT\_VERSION

Edit OpenFOAM-v1812-ADF/etc/config.sh/CGAL:

boost\_version=boost\_1\_64\_0

->

boost\_version=boost-system

-------------------------------------------------

export BOOST\_ARCH\_PATH=$WM\_THIRD\_PARTY\_DIR/platforms/$WM\_ARCH$WM\_COMPILER/$boost\_version

->

export BOOST\_ARCH\_PATH=$TACC\_BOOST\_DIR

Edit OpenFOAM-v1812-ADF/etc/config.sh/settings:

export WM\_CC="gcc"

export WM\_CXX="g++"

export WM\_CFLAGS="-fPIC"

export WM\_CXXFLAGS="-fPIC -std=c++11"

->

export WM\_CC="mpicc"

export WM\_CXX="mpicxx"

export WM\_CFLAGS="-O3 -fPIC"

export WM\_CXXFLAGS="-O3 -fPIC -std=c++11"

-------------------------------------------------

64)

WM\_ARCH=linux64

export WM\_COMPILER\_LIB\_ARCH=64

export WM\_CFLAGS="$WM\_CFLAGS -m64"

export WM\_CXXFLAGS="$WM\_CXXFLAGS -m64"

export WM\_LDFLAGS="-m64"

->

64)

WM\_ARCH=linux64

export WM\_COMPILER\_LIB\_ARCH=64

export WM\_CFLAGS="$WM\_CFLAGS"

export WM\_CXXFLAGS="$WM\_CXXFLAGS"

export WM\_LDFLAGS="-O3"

Edit OpenFOAM-v1812-ADF/wmake/rules/linux64Icc/c:

cc = icc

->

cc = mpicc

Edit OpenFOAM-v1812-ADF/wmake/rules/linux64Icc/c++:

CC = icpc -std=c++11 -fp-trap=common -fp-model precise

->

CC = mpicxx -std=c++11 -fp-trap=common -fp-model precise

Copy mplibINTELMPI:

cp OpenFOAM-v1812-ADF/wmake/rules/linux64Icc/mplibINTELMPI OpenFOAM-v1812-ADF/wmake/rules/General/mplibINTELMPI

Once the configuration file modifications are complete, source the OpenFOAM forward-mode bashrc file:

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

Install OpenFOAM forward-mode AD using the following command:

cd $WM\_PROJECT\_DIR and ./Allwmake > log.make 2>&1

As with the reverse-mode AD code, we need to link the built forward-mode AD code to the original version. To do this, run the following commands to create soft links:

cd $DAFOAM\_ROOT\_PATH/OpenFOAM/OpenFOAM-v1812/platforms/\*/lib

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

cd $DAFOAM\_ROOT\_PATH/OpenFOAM-v1812/platforms/\*/lib/dummy

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

cd $DAFOAM\_ROOT\_PATH/OpenFOAM-v1812/platforms/\*/lib/mpi

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

## Install CGNS

CGNS is a required dependency for DAFoam that should be installed within the packages/ subdirectory of the DAFoam directory. Enter the packages directory, download CGNS, untar the code files, and enter the CGNS directory:

cd $DAFOAM\_ROOT\_PATH/packages/

wget https://github.com/CGNS/CGNS/archive/v3.3.0.tar.gz

tar -xvaf v3.3.0.tar.gz

cd CGNS-3.3.0

We will configure the CGNS build with several options. We will enable Fortran bindings, set the installation location, disable 64bit architecture, and specify the paths for the required Fortran and C compilers. Run this configuration step using the following command:

cmake -D CGNS\_ENABLE\_FORTRAN=ON -D CMAKE\_INSTALL\_PREFIX=$CGNS\_HOME -D CGNS\_ENABLE\_64BIT=OFF -D CGNS\_BUILD\_CGNSTOOLS=OFF -D CMAKE\_C\_COMPILER=$(which icc) -D CMAKE\_Fortran\_COMPILER=$(which ifort) .

Once the configuration process is complete, make and install CGNS using the following command:

make all install

## Install MACH-Aero

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 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| v1.2.0 | v1.2.0 | v1.5.0 | v1.2.0 | v2.2.0 | v2.5.0 | v2.2.1 | v2.3.0 | v1.2.1 | v3.0.7 |

The MACH-Aero packages are considered part of the DAFoam ecosystem and should be installed in the repos/ subdirectory of the DAFoam/ directory. Move to the repos directory:

cd $DAFOAM\_ROOT\_PATH/repos/

The following steps go through the process of installing the MACH-Aero codes one at a time, downloading, unpacking, building, and installing each package. For simplicity, the packages will be installed in-place.

baseClasses

Download the repository and enter the directory:

wget https://github.com/mdolab/baseclasses/archive/v1.2.0.tar.gz -O baseclasses.tar.gz

tar -xvf baseclasses.tar.gz

cd baseclasses-1.2.0

Install the package:

pip3 install -e .

Return to the repos directory:

cd $DAFOAM\_ROOT\_PATH/repos/

**pySpline**

Download the repository and enter the directory:

wget https://github.com/mdolab/pyspline/archive/v1.2.0.tar.gz -O pyspline.tar.gz

tar -xvf pyspline.tar.gz

cd pyspline-1.2.0

Copy the configuration file:

cp config/defaults/config.LINUX\_INTEL.mk config/config.mk

Compile the code:

make

Install the package:

pip3 install -e .

Return to the repos directory:

cd $DAFOAM\_ROOT\_PATH/repos/

pyGeo

Download the repository and enter the directory:

wget https://github.com/mdolab/pygeo/archive/v1.5.0.tar.gz -O pygeo.tar.gz

tar -xvf pygeo.tar.gz

cd pygeo-1.5.0

Install the package:

pip3 install -e .

Return to the repos directory:

cd $DAFOAM\_ROOT\_PATH/repos/

multipoint

Download the repository and enter the directory:

wget https://github.com/mdolab/multipoint/archive/v1.2.0.tar.gz -O multipoint.tar.gz

tar -xvf multipoint.tar.gz

cd multipoint-1.2.0

Install the package:

pip3 install -e .

Return to the repos directory:

cd $DAFOAM\_ROOT\_PATH/repos/

pyHyp

Download the repository and enter the directory:

wget https://github.com/mdolab/pyhyp/archive/v2.2.0.tar.gz -O pyhyp.tar.gz

tar -xvf pyhyp.tar.gz

cd pyhyp-2.2.0

Copy the configuration file:

cp config/defaults/config.LINUX\_INTEL\_OPENMPI.mk config/config.mk

Compile the code:

make

Install the package:

pip3 install -e .

Return to the repos directory:

cd $DAFOAM\_ROOT\_PATH/repos/

cgnsUtilities

Download the repository and enter the directory:

wget https://github.com/mdolab/cgnsutilities/archive/v2.5.0.tar.gz -O cgnsutilities.tar.gz

tar -xvf cgnsutilities.tar.gz

cd cgnsutilities-2.5.0

Copy the configuration file:

cp config/defaults/config.LINUX\_INTEL.mk config/config.mk

Compile the code:

make

Install the package:

pip3 install -e .

Return to the repos directory:

cd $DAFOAM\_ROOT\_PATH/repos/

IDWarp

Download the repository and enter the directory:

wget https://github.com/mdolab/idwarp/archive/v2.2.1.tar.gz -O idwarp.tar.gz

tar -xvf idwarp.tar.gz

cd idwarp-2.2.1

Copy the configuration file:

cp config/defaults/config.LINUX\_INTEL\_OPENMPI.mk config/config.mk

Compile the code:

make

Install the package:

pip3 install -e .

Return to the repos directory:

cd $DAFOAM\_ROOT\_PATH/repos/

pyOptSparse

**NOTE:** pyOptSparse can be modified before installing to include additional optimizers. Refer to the pyOptSparse documentation for more information

Download the repository and enter the directory:

wget https://github.com/mdolab/pyoptsparse/archive/v2.3.0.tar.gz -O pyoptsparse.tar.gz

tar -xvf pyoptsparse.tar.gz

cd pyoptsparse-2.3.0

Install the package:

pip3 install -e .

Return to the repos directory:

cd $DAFOAM\_ROOT\_PATH/repos/

## Install pyOFM

In addition to installing the main components of MACH-Aero, we will install pyOFM to interface with OpenFOAM meshes. Download the repository and enter its code directory:

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

Make the package:

make

Install the package:

pip3 install -e .

Return to the repos directory:

cd $DAFOAM\_ROOT\_PATH/repos/

## Install DAFoam

Once OpenFOAM, CGNS, and the MACH-Aero packages are installed, we can install DAFoam. Similar to OpenFOAM, we need to compile three versions of DAFoam: original, reverse-mode AD (ADR), and forward-mode AD (ADF). Download the repository and enter its code directory:

wget https://github.com/mdolab/dafoam/archive/v2.2.9.tar.gz -O dafoam.tar.gz

tar -xvf dafoam.tar.gz

cd dafoam-2.2.9

Build DAFoam original:

. $DAFOAM\_ROOT\_PATH/ENV\_DAFOAM.sh

./Allmake 2> errorLog.txt

Build DAFoam reverse-mode:

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

./Allclean && ./Allmake 2> errorLog.txt

Build DAFoam forward-mode:

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

./Allclean && ./Allmake 2> errorLog.txt

Install the package:

pip3 install -e .

Once the installtion is complete, unset the AD environment and re-source the environment loading script. **This step is needed every time you compile an AD version of DAFoam!**

unset WM\_CODI\_AD\_MODE\

. $DAFOAM\_ROOT\_PATH/ENV\_DAFOAM.sh

## Regression Tests

Once DAFoam and all of its dependencies are installed, you can run the regression tests located in the DAFoam/repos/dafoam/tests directory. To run the cases, use the command:

cd $HOME/dafoam/repos/dafoam/tests && ./Allrun

If successful, the test will output the following message:

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

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

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

If for any reason the tests fail, check the output message to understand if the issue is due to numerical precision errors caused by differing hardware / compiler versions or if there is an issue with the installation.