Example instructions for installing SOWFA-2.4.x with OpenFASTv2.0+
(1 July 2019, from Hannah Johlas)

I. **Find/install/load major prerequisites:**

1. Make sure you have the following installed:
   - Compilers (e.g. gcc, g++)
   - OpenMPI libraries
   - cmake
   - BLAS/LAPACK libraries
   - HDF5 libraries

II. **Install OpenFOAM 2.4.x to ~/OpenFOAM**

1. Download OpenFOAM-2.4.x and ThirdParty-2.4.x repos from github into ~/OpenFOAM:

```
[~/OpenFOAM]$ git clone git://github.com/OpenFOAM/OpenFOAM-2.4.x
[~/OpenFOAM]$ git clone git://github.com/OpenFOAM/ThirdParty-2.4.x
```

2. Create a function in ~/.bash_profile that sets environment for OpenFOAM-2.4.x (loads modules, sources ~/OpenFOAM/2.4.x/etc/bashrc). Source and execute before continuing. An example OpenFOAM environment function:

```
openfoam_env()
{       # Set up module environment for prerequisites
        module purge
        echo "Loading modules..."
        module load gnu/4.9.2
        module load openmpi_ib/1.8.4
        module load cmake/3.9.1
        module load lapack/3.6.0
        module load hdf5/1.8.14
        module list

      # Set OpenFOAM version
        export OPENFOAM_VERSION=2.4.x
        export OPENFOAM_NAME=OpenFOAM-$OPENFOAM_VERSION
        export FOAM_INST_DIR=~/OpenFOAM

        # Source OpenFOAM bashrc file
        foamBashrc=$FOAM_INST_DIR/$OPENFOAM_NAME/etc/bashrc
        if [ -f $foamDotFile ] ; then
                echo "Sourcing $foamBashrc..."
                source $foamBashrc
        fi
}
```

3. Compile Scotch 6.0.3 to ~/OpenFOAM/ThirdParty-2.4.x
   See https://openfoam.org/download/source/third-party-software/
   Download and unzip scotch_6.0.3.tar.gz from
   https://gforge.inria.fr/frs/download.php/file/34099/scotch_6.0.3.tar.gz :

```
[ThirdParty-2.4.x]$ source ~/.bash_profile
[ThirdParty-2.4.x]$ openfoam_env
[ThirdParty-2.4.x]$ tar -xzvf scotch_6.0.3.tar.gz
[ThirdParty-2.4.x]$ ./Allwmake > log.Allwmake 2>&1
```

4. Compile OpenFOAM-2.4.x to ~/OpenFOAM/OpenFOAM-2.4.x
   See https://openfoam.org/download/source/compiling-openfoam/
   An example compiling script on 24 cores:

```
source ~/.bash_profile
openfoam_env
cd ~/$FOAM_INST_DIR/OPENFOAM_NAME
export WM_NCOMPPROCS=24
./Allwmake > log.Allwmake 2>&1
```

III. Install OpenFAST (dev branch) to ~/OpenFAST

1. Download OpenFAST dev repo from github into ~/OpenFAST:

```
[~OpenFAST]$ git clone --single-branch --branch dev
       git://github.com/OpenFAST/openfast
```

2. Find yaml-cpp OR install yaml-cpp to ~/OpenFAST:
   A. Download yaml-cpp repo:

```
[~OpenFAST]$ git clone git://github.com/jbeder/yaml-cpp
```

   B. Compile yaml-cpp into OpenFAST/yaml-cpp/install
      An example compiling script:

```
source ~/.bash_profile
openfoam_env
cd ~/OpenFAST/yaml-cpp
rm -r build
mkdir build
cd build
cmake -DCMAKE_INSTALL_PREFIX=~/OpenFAST/yaml-cpp/install \
  ../ > ../log.cmake 2>&1
make > ../log.make 2>&1
make install > ../log.install 2>&1
```

3. Compile OpenFAST with CPP API
See https://openfast.readthedocs.io/en/master/source/install/install_cmake_linux.html
An example compiling script on 8 cores:

```
source ~/.bash_profile
openfoam_env
cd ~/OpenFAST/openfast
rm -r build
mkdir build
cd build
cmake \
  -DCMAKE_INSTALL_PREFIX=~/OpenFAST/openfast/install/ \
  -DCMAKE_BUILD_TYPE=RELEASE \
  -DBUILD_OPENFAST_CPP_API=ON \
  -DFPE_TRAP_ENABLED=ON \
  -DYAML_ROOT=/home/hjohlas/OpenFAST/yaml-cpp/install \
  ../ > ../log.cmake 2>&1
make -j 8 > ../log.make 2>&1
make install > ../log.install 2>&1
```

NOTE: -DFPE_TRAP_ENABLED must be set to ON; otherwise SOWFA simulations using an OpenFAST-linked solver will crash with a floating point exception during OpenFAST initialization.
NOTE: if cmake is unable to find your HDF5 install automatically, add the following cmake option your compile script:

```
-DHDF5_ROOT=${WHERE_HDF5_IS}
```

IV. Install SOWFA-2.4.x into ~/OpenFOAM
See https://github.com/NREL/SOWFA/blob/master/README.SOWFA

1. Download SOWFA repo from github into ~/OpenFOAM:

```
[OpenFOAM]$ git clone git://github.com/NREL/SOWFA
```

2. Rename SOWFA repo to SOWFA-2.4.x:

```
[OpenFOAM]$ mv SOWFA SOWFA-2.4.x
```

3. Create a function in ~/.bash_profile that sets environment for SOWFA-2.4.x (loads OpenFOAM environment, adds environment variables). Source and execute before continuing. An example SOWFA environment function:

```
sowfa_env
{  # Unset OpenFOAM environment variables.
   if [ -z "$FOAM_INST_DIR" ]; then
      echo "Nothing to unset..."
   else
      echo "Unsetting OpenFOAM environment variables..."
      . $FOAM_INST_DIR/OpenFOAM-$OPENFOAM_VERSION/etc/config/unset.sh
   fi

   # Load OpenFOAM environment
   Openfoam_env

   # Set SOWFA-specific variables
   export SOWFA_DIR=~/OpenFOAM/SOWFA-$OPENFOAM_VERSION-$USER
   export OPENFAST_DIR=~/OpenFAST/openfast/install

   export HDF5_DIR=${WHERE_HDF5_IS}
   export BLASLIB="-L/opt/lapack/gnu/lib -lblas -llapack"

   export LD_LIBRARY_PATH=$SOWFA_DIR/lib/$WM_OPTIONS:$OPENFAST_DIR/lib:
      $HDF5_DIR/lib:$LD_LIBRARY_PATH
   export PATH=$SOWFA_DIR/applications/bin/$WM_OPTIONS:$PATH
   export WM_PROJECT_USER_DIR=$SOWFA_DIR
}
```

NOTE: check to see if $HDF5_DIR and $BLASLIB are set by default (echo $HDF5_DIR before running sowfa_env); if so, you do not need to set them in your SOWFA environment function. These variables are needed when compiling SOWFA-OpenFAST solvers.

4. Check/update SOWFA’s compiling options:
   A. In $SOWFA_DIR/src/turbineModels/turbineModelsOpenFAST/Make/options :
      1. add -I$(HDF5_DIR)/include to EXE_INC list
      2. add -L$(HDF5_DIR)/lib/ , -lhdf5_hl, -lhdf5 to LIB_LIBS list

```
         3. add -lversioninfolib to LIB_LIBS list (for OpenFAST v2.0+)
   B. In $SOWFA_DIR/applications/solvers/incompressible/windEnergy/
      pisoFoamTurbine.ALMAdvancedOpenFAST/Make/options:
         1. add -I$(HDF5_DIR)/include to EXE_INC list
         2. add -lversioninfolib to LIB_LIBS list (for OpenFAST v2.0+)
   C. In applications/solvers/incompressible/windEnergy/
      windPlantSolver.ALMAdvancedOpenFAST/Make/options:
         1. add -lversioninfolib to LIB_LIBS list (for OpenFAST v2.0+)
```

NOTE: Explicitly including and linking HDF5 libraries in all OpenFAST-related solvers and libraries may not be necessary if HDF5 is installed in $OPENFAST_DIR

NOTE: adding -lversioninfolib is necessary for OpenFAST v2.0+, which introduced a new library $OPENFAST_DIR/lib/libversioninfolib.so

5. Compile SOWFA-2.4.x
Example compiling script on 24 cores:

```
source ~/.bash_profile
sowfa_env
cd $SOWFA_DIR
export WM_NCOMPPROCS=24
./Allwclean > log.Allwclean1 2>&1
./Allwmake > log.Allwmake 2>&1
./Allwclean > log.Allwclean2 2>&1
```

V. Common error help cheat-sheet

1. cmake, Allwmake can’t find BLAS libraries
   See IV.3: try setting $BLASLIB
2. cmake, Allwmake can’t find HDF5 libraries
   See III.3: try adding -DHDF5_ROOT=${WHERE_HDF5_IS}
   See IV.3: try setting $HDF5_DIR
   See IV.4: try adding missing HDF5-related options to Make/options files
3. cmake, Allwmake can’t find yaml-cpp
   See III.3: try setting -DYAML_ROOT=${WHERE_YAML_IS}
4. cmake, Allwmake say undefined reference to version_info function
   See IV.4: try adding -lversioninfolib to Make/options files
5. Everything compiles, but windPlantSolver.ALMAdvancedOpenFAST or pisoFoamTurbine.ALMAdvancedOpenFAST crashes with a floating point exception while initializing OpenFAST
   See III.3: try setting -DFPE_TRAP_ENABLED=ON when compiling OpenFAST