

IBAMR install on OS X Sierra Version 10.11.6

June ?, 2016

Install gcc 4.9 using Mac Ports: (<https://www.macports.org/install.php>)

```
sudo port install gcc49
```

Setup

Create Directories where all software will be installed

```
cd $HOME
mkdir sfw
cd sfw
mkdir linux
```

First we will need to install all the supporting libraries for IBAMR

Boost

Create Directory where boost will be installed

```
cd $HOME/sfw/linux
mkdir boost
cd boost/
```

Obtain latest version of boost from the following website using wget

```
wget http://sourceforge.net/projects/boost/files/latest/download?source=files
```

Rename file to known latest version: Go to <https://sourceforge.net/projects/boost/files/?source=navbar> to determine it

```
mv download\?source\=files boost_1_63_0.tar.bz2
```

Untar and unzip downloaded boost file

```
tar xvjf boost_1_63_0.tar.bz2
```

Change directory name for convenience

```
mv boost_1_63_0 1.63.0
cd 1.63.0/
```

Change boost root directory

```
export BOOST_ROOT=$HOME/sfw/linux/boost/1.63.0/
```

Boost installation should be complete

HDF5

Create directory where HDF5 will be installed

```
cd $HOME/sfw/linux
```

Obtain latest version of HDF5 using wget

```
wget http://www.hdfgroup.org/ftp/HDF5/current/src/hdf5-1.10.1.tar
```

Untar downloaded HDF5 file

```
tar xvf hdf5-1.10.1.tar
cd hdf5-1.10.1
```

Configure with the mentioned compilers and preferred settings

```
./configure \
CC=gcc-mp-4.9 \
CXX=g++-mp-4.9 \
FC=gfortran-mp-4.9 \
F77=gfortran-mp-4.9 \
--enable-build-mode=production \
--prefix=$HOME/sfw/linux/hdf5/1.10.1
make
make check
make install
```

HDF5 installation should be complete

Silo

*Note: Mac OS X comes with zlib, so no need to install.

Make directory for Silo

```
cd $HOME/sfw/linux
```

Download the latest version from the following website:
<https://wci.llnl.gov/simulation/computer-codes/silo/downloads/>
Download Silo-4.10.2-tar.gz or the latest version
Untar and unzip the above downloaded Silo file

```
tar xvfz silo-4.10.2.tar.gz
```

```
cd silo-4.10.2
```

Configure with the mentioned compilers and preferred settings

```
./configure \
CC=gcc-mp-4.9 \
CXX=g++-mp-4.9 \
FC=gfortran-mp-4.9 \
F77=gfortran-mp-4.9 \
--prefix=$HOME/sfw/linux/silo/4.10.2 \
--disable-silex
```

```
make
make check //Error -rdynamic but no need to worry. Just continue
make install
```

Silo Installation should be complete

OpenMPI

Note: New 2. versions of openmpi have weird temp directories. Change it by adding the following to your .bash_profile in your \$HOME directory
export TMPDIR=/tmp

Make directory for OpenMPI

```
cd $HOME/sfw/linux/
```

Download the latest version of openmpi from the following website:
<https://www.open-mpi.org/software/ompi/v2.1/> and download the openmpi-2.1.1.tar.gz file
There might be a newer version when this is being read. The above tar file can be used in the build without any hiccups

Download openmpi-2.1.1.tar.gz

```
tar xvfz openmpi-2.1.1.tar.gz  
cd openmpi-1.10.2
```

Configure with the mentioned compilers and preferred settings

```
./configure \  
CC=gcc-mp-4.9 \  
CXX=g++-mp-4.9 \  
FC=gfortran-mp-4.9 \  
F77=gfortran-mp-4.9 \  
--prefix=$HOME/sfw/linux/openmpi/2.1.1 \  
--disable-mpi-cxx-seek \  
--disable-heterogeneous \  
--enable-orterun-prefix-by-default
```

```
make
```

```
make check
```

```
make install
```

OpenMPI Installation should be complete

PETSc

Create directory for PETSc

```
cd $HOME/sfw  
mkdir petsc  
cd petsc
```

Download the latest version of PETSc from the following website using wget:

```
wget http://ftp.mcs.anl.gov/pub/petsc/release-snapshots/petsc-3.7.tar.gz  
tar xvfz petsc-3.7.tar.gz
```

Change name of directory for convenience

```
mv petsc-3.7.6 3.7.6
```

cd 3.7.6/

We will need to install 2 versions of PETSc, one for debugging our IBAMR projects and the other for when we want to run an optimized build.

Install a debugging version of PETSc:

Make sure you are in the /3.7.6 directory

Change PETSc directory and ARCH

```
export PETSC_DIR=$PWD
export PETSC_ARCH=linux-debug
```

Configure using the specified compilers, flags, and preferred settings. It will have debugging, so
—with-debugging=1

```
./config/configure.py \
--CC=$HOME/sfw/linux/openmpi/2.1.1/bin/mpicc \
--CXX=$HOME/sfw/linux/openmpi/2.1.1/bin/mpicxx \
--FC=$HOME/sfw/linux/openmpi/2.1.1/bin/mpif90 \
--LDFLAGS="-L$HOME/sfw/linux/openmpi/2.1.1/lib -Wl,-rpath,$HOME/sfw/linux/openmpi/2.1.1/lib" \
--with-default-arch=0 \
--PETSC_ARCH=$PETSC_ARCH \
--with-debugging=1 \
--with-c++-support \
--with-hypre=1 \
--download-hypre=1
```

```
make
make test
```

PETSc debug installation should be complete

Install an optimized version of PETSc:

Make sure you are in the /3.7.6 directory

Change PETSc directory and ARCH

```
export PETSC_DIR=$PWD
export PETSC_ARCH=linux-opt
```

Configure using the specified compilers, flags, and preferred settings. It will NOT have debugging, so
—with-debugging=0

```
./config/configure.py \
--CC=$HOME/sfw/linux/openmpi/2.1.1/bin/mpicc \
--CXX=$HOME/sfw/linux/openmpi/2.1.1/bin/mpicxx \
--FC=$HOME/sfw/linux/openmpi/2.1.1/bin/mpif90 \
```

NOTE: The following flags have capital O not zero 0

```
--COPTFLAGS="-O3" \
--CXXOPTFLAGS="-O3" \
--FOPTFLAGS="-O3" \
--LDFLAGS="-L$HOME/sfw/linux/openmpi/2.1.1/lib -Wl,-rpath,$HOME/sfw/linux/openmpi/2.1.1/lib" \
```

```
--with-default-arch=0 \
--PETSC_ARCH=$PETSC_ARCH \
--with-debugging=0 \
--with-c++-support \
--with-hypre=1 \
--download-hypre=1 \
--with-x=0
```

```
make
make test
```

PETSc optimized installation should be complete

SAMRAI

*download SAMRAI-v2.4.4.tar.gz from <http://computation.llnl.gov/projects/samrai/software>

Create directory for SAMRAI

```
cd $HOME/sfw
mkdir samrai
cd samrai
mkdir 2.4.4
cd 2.4.4
```

Download SAMRAI-v2.4.4.tar.gz from the following website:

<https://computation.llnl.gov/projects/samrai/software>

Note: I think it has to be this version because of the patch that is downloaded below. This patch is specific to IBAMR

```
tar xvfz SAMRAI-v2.4.4.tar.gz
mv SAMRAI SAMRAI-2.4.4
```

Download the SAMRAI patch for IBAMR on [ibamr's github](https://github.com/IBAMR/IBAMR). The url is provided in the terminal command below

```
wget https://github.com/IBAMR/IBAMR/releases/download/v0.1-rc1/SAMRAI-v2.4.4-patch-121212.gz
cd SAMRAI-2.4.4/
```

Patch currently downloaded version of SAMRAI

```
./source/scripts/includes --link
gunzip -c ../SAMRAI-v2.4.4-patch-121212.gz | patch -p2
```

We will need to install 2 versions of SAMRAI, one for debugging our IBAMR projects and the other for when we want to run an optimized build.

Install a debugging version of SAMRAI:

Make and go to the directory for our debugging version

```
cd $HOME/sfw/samrai/2.4.4
mkdir objs-debug
cd objs-debug/
```

Configure SAMRAI using the compilers and preferred settings below.

```
../SAMRAI-2.4.4/configure \
--prefix=$HOME/sfw/samrai/2.4.4/linux-g++-debug \
--with-CC=gcc-mp-4.9 \
--with-CXX=g++-mp-4.9 \
--with-F77=gfortran-mp-4.9 \
--with-MPICC=$HOME/sfw/linux/openmpi/2.1.1/bin/mpicc \
--with-hdf5=$HOME/sfw/linux/hdf5/1.10.1 \
--without-hypr \
--with-silo=$HOME/sfw/linux/silo/4.10.2 \
--without-blaslapack \
--without-cubes \
--without-eleven \
--without-kinsol \
--without-petsc \
--without-sundials \
--without-x \
--with-doxygen \
--with-dot \
--enable-debug \
--disable-opt \
--enable-implicit-template-instantiation \
--disable-deprecated
```

```
make
make install
```

SAMRAI debugging version should be complete

Install an optimized version of SAMRAI:

Make and go to the optimized directory

```
cd $HOME/sfw/samrai/2.4.4
mkdir objs-opt
cd objs-opt/
```

Configure SAMRAI using the specified compilers and preferred settings

```
../SAMRAI-2.4.4/configure \
CFLAGS="-O3" \
CXXFLAGS="-O3" \
FFLAGS="-O3" \
--prefix=$HOME/sfw/samrai/2.4.4/linux-g++-opt \
--with-CC=gcc-mp-4.9 \
--with-CXX=g++-mp-4.9 \
--with-F77=gfortran-mp-4.9 \
--with-MPICC=$HOME/sfw/linux/openmpi/2.1.1/bin/mpicc \
--with-hdf5=$HOME/sfw/linux/hdf5/1.10.1 \
--without-hypr \
--with-silo=$HOME/sfw/linux/silo/4.10.2 \
--without-blaslapack \
--without-cubes \
--without-eleven \
--without-kinsol \
--without-petsc \
```

```
--without-sundials \
--without-x \
--with-doxygen \
--with-dot \
--disable-debug \
--enable-opt \
--enable-implicit-template-instantiation \
--disable-deprecated
```

```
make
make install
```

SAMRAI optimized version installation should be complete

libMesh

Make directory for libMesh

```
cd $HOME/sfw/linux
mkdir libmesh
cd libmesh
mkdir 1.0.0
cd 1.0.0/
```

Download lib mesh-1.0.0.tar.gz from the website included in the terminal command below

```
wget https://github.com/libMesh/libmesh/releases/download/v1.0.0/libmesh-1.0.0.tar.gz
tar xvzf libmesh-1.0.0.tar.gz
```

Rename directory for convenience

```
mv libmesh-1.0.0 LIBMESH
```

We will need to install 2 versions of libMesh, one for debugging our IBAMR projects and the other for when we want to run an optimized build.

Install a debugging version of libMesh:

Make and go to libmesh debug directory

```
cd $HOME/sfw/linux/libmesh/1.0.0
mkdir objs-debug
cd objs-debug/
```

Make sure to specify the root directory for boost shown below

```
export BOOST_ROOT=$HOME/sfw/linux/boost/1.63.0/
```

Configure libMesh using the following compilers and preferred settings. Note: we use the PETSc debugging version

```
../LIBMESH/configure \
--prefix=$HOME/sfw/linux/libmesh/1.0.0/1.0.0-debug \
--with-methods=dbg \
PETSC_DIR=$HOME/sfw/petsc/3.7.6 \
PETSC_ARCH=linux-debug \
```

```
CC=$HOME/sfw/linux/openmpi/2.1.1/bin/mpicc \  
CXX=$HOME/sfw/linux/openmpi/2.1.1/bin/mpicxx \  
FC=$HOME/sfw/linux/openmpi/2.1.1/bin/mpif90 \  
F77=$HOME/sfw/linux/openmpi/2.1.1/bin/mpif90 \  
--enable-triangle \  
--disable-cxx11 \  
--disable-openmp \  
--disable-perflog \  
--disable-strict-lgpl \  
--disable-pthreads \  
--disable-cppthreads \  
--disable-unique-ptr
```

```
make  
make install
```

****Note: If using libMesh, it is necessary to use an external boost with IBAMR. Change the name of the boost directory in the include folder so IBAMR uses the external library:**

```
cd $HOME/sfw/linux/libmesh/1.0.0/1.0.0-debug/include  
mv boost/ boost_name_change
```

libMesh debugging version installation should be complete

Install an optimized version of libMesh:

Make and go to the optimized directory

```
cd sfw/linux/libmesh/1.0.0  
mkdir objs-opt  
cd objs-opt/
```

Specify the boost root directory

```
export BOOST_ROOT=$HOME/sfw/linux/boost/1.63.0
```

Configure libMesh using the below compilers and preferred settings. Note: we use the PETSc optimized build here

```
../LIBMESH/configure \  
--prefix=$HOME/sfw/linux/libmesh/1.0.0/1.0.0-opt \  
--with-methods=opt \  
PETSC_DIR=$HOME/sfw/petsc/3.7.6 \  
PETSC_ARCH=linux-opt \  
CC=$HOME/sfw/linux/openmpi/2.1.1/bin/mpicc \  
CXX=$HOME/sfw/linux/openmpi/2.1.1/bin/mpicxx \  
FC=$HOME/sfw/linux/openmpi/2.1.1/bin/mpif90 \  
F77=$HOME/sfw/linux/openmpi/2.1.1/bin/mpif90 \  
--enable-triangle \  
--disable-cxx11 \  
--disable-openmp \  
--disable-perflog \  
--disable-strict-lgpl \  
--disable-pthreads \  
--disable-cppthreads \  
--disable-unique-ptr
```

```
make
make install
```

****Note: If using libMesh, it is necessary to use an external boost with IBAMR. Change the name of the boost directory in the include folder so IBAMR uses the external library:**

```
cd $HOME/sfw/linux/libmesh/1.0.0/1.0.0-opt/include
mv boost/ boost_name_change
```

libMesh Optimized version installation should be complete

IBAMR

Now comes the fun part. If everything was done correctly above, then this part should go swimmingly. If it does not go as planned, make sure you record what errors have occurred and where they did

Create directory for IBAMR

```
cd $HOME/sfw
mkdir ibamr
cd ibamr
```

Clone the ibamr git using the terminal command below

```
git clone https://github.com/IBAMR/IBAMR.git
```

We will need to install 2 versions of IBAMR, one for debugging our projects and the other for when we want to run an optimized build.

Install a debugging version of IBAMR:

Make and go to the debug directory. You can name it something else if you'd like

```
mkdir ibamr-objs-dbg
cd ibamr-objs-dbg
```

Include changes to the boost root directory and PETSc directory and build version

```
export BOOST_ROOT=$HOME/sfw/linux/boost/1.63.0/
export PETSC_ARCH=linux-debug
export PETSC_DIR=$HOME/sfw/petsc/3.7.6
```

Configure the ibamr build with the following compilers, flags and all the previous software versions we have just downloaded

```
../IBAMR/configure \
  CFLAGS="-g -O1 -Wall" \
  CXXFLAGS="-g -O1 -Wall" \
  FCFLAGS="-g -O1 -Wall" \
  CC=$HOME/sfw/linux/openmpi/2.1.1/bin/mpicc \
  CXX=$HOME/sfw/linux/openmpi/2.1.1/bin/mpicxx \
  FC=$HOME/sfw/linux/openmpi/2.1.1/bin/mpif90 \
  CPPFLAGS="-DOMPI_SKIP_MPICXX" \
  --with-hypre=$PETSC_DIR/$PETSC_ARCH \
  --with-samrai=$HOME/sfw/samrai/2.4.4/linux-g++-debug \
  --with-hdf5=$HOME/sfw/linux/hdf5/1.10.1 \
  --with-silo=$HOME/sfw/linux/silo/4.10.2 \
  --with-boost=$HOME/sfw/linux/boost/1.63.0 \
```

```
--enable-libmesh \  
--with-libmesh=$HOME/sfw/linux/libmesh/1.0.0/1.0.0-debug \  
--with-libmesh-method=dbg
```

When this is complete, the configure process should tell you how to make the libraries and examples. You will do both.

The libraries are necessary for your future projects, and the examples are important since they should run without any problems if ibamr is compiled correctly.

Note: Some examples I have not been able to run properly (time step change error...unsure). But if you run the example mentioned below and it runs properly, then all went well

```
make lib  
make examples
```

Run the IBAMR examples. For example:

```
cd $HOME/sfw/ibamr/ibamr-objs-dbg/examples/IB/explicit/ex0  
./main2d input2d
```

IBAMR Debugging version installation should be complete!

Install an optimized build of IBAMR:

Make and go to the optimized build directory. You may change the name if you'd like

```
cd $HOME/sfw/ibamr  
mkdir ibamr-objs-opt  
cd ibamr-objs-opt
```

Mention changes to the boost root directory, PETSc directory, and PETSc build

```
export BOOST_ROOT=$HOME/sfw/linux/boost/1.63.0/  
export PETSC_ARCH=linux-opt  
export PETSC_DIR=$HOME/sfw/petsc/3.7.6
```

Configure ibamr with the following flags, compilers, and all the previous software versions we have just downloaded

```
../IBAMR/configure \  
CC=$HOME/sfw/linux/openmpi/2.1.1/bin/mpicc \  
CXX=$HOME/sfw/linux/openmpi/2.1.1/bin/mpicxx \  
F77=$HOME/sfw/linux/openmpi/2.1.1/bin/mpif90 \  
FC=$HOME/sfw/linux/openmpi/2.1.1/bin/mpif90 \  
MPICC=$HOME/sfw/linux/openmpi/2.1.1/bin/mpicc \  
MPICXX=$HOME/sfw/linux/openmpi/2.1.1/bin/mpicxx \  
CFLAGS="-O3 -Wall" \  
CXXFLAGS="-O3 -Wall -std=gnu++11" \  
FFLAGS="-O3 -Wall" \  
FCFLAGS="-O3 -Wall" \  
CPPFLAGS="-DOMPI_SKIP_MPICXX" \  
--with-hypre=$PETSC_DIR/$PETSC_ARCH \  
--with-samrai=$HOME/sfw/samrai/2.4.4/linux-g++-opt \  
--with-hdf5=$HOME/sfw/linux/hdf5/1.10.1 \  
--with-silo=$HOME/sfw/linux/silo/4.10.2 \  
--with-boost=$HOME/sfw/linux/boost/1.63.0 \  
--enable-libmesh \  
--with-libmesh=$HOME/sfw/linux/libmesh/1.0.0/1.0.0-opt \  

```

```
--with-libmesh-method=opt
```

Same as the debug version!! Almost done!

```
make lib  
make examples
```

Run the IBAMR examples. For example:

```
cd $HOME/sfw/ibamr/ibamr-objs-opt/examples/IB/explicit/ex0  
./main2d input2d
```

Congratulations! You have completed the installation of IBAMR on your local system. Now comes even more fun!

After experimenting with some examples and possibly modifying the example to make your own project, it's probably a good idea to install this again (I know) on KillDevil