Use GNU AutoTools to generate the configure file for AstroBEAR

Baowei Liu
University of Rochester

October 20, 2014

Contents

Chapter 1

Introduction

This document record my notes of learning GNU AutoTools and working to generate the configure file for AstroBEAR. The version of Autotools used is GNU Autoconf 2.69. It's hard to find online tutorial materials using Gnu Autotools. Most of material found are usually outdated and almost all of them are for C/C++. Here's the route I've been through: hello world in C; hello world in Fortran; hello world in MPI Fortran; example with external library (paralllel HDF5 with mpif90); example with multiple source code files; AstroBEAR.

Chapter 2

Examples

2.1 C Hello World: amhello

changing main.c to hello.c and redo the procedures solved the problem!!

Under the subfolder src/ put these two files hello.c

```
#include <config.h>
#include <stdio.h>

int main(void)
{
  puts ("Hello␣World!");
  puts ("This␣is␣" PACKAGE_STRING ".");
  return 0;
}
```

Makefile.am

```
bin_PROGRAMS = hello
hello_SOURCE = hello.c
```

Under the root folder put these two files
configure.ac

```
AC_INIT([amhello], [1.0],[bliu@pas.rochester.edu])
AM_INIT_AUTOMAKE([foreign -Wall -Werror])
AC_PROG_CC
AC_CONFIG_HEADERS([config.h])
AC_CONFIG_FILES([Makefile src/Makefile])
AC_OUTPUT
```

Makefile.am

```
SUBDIRS = src
```

To test and create a distributed version, use these lines of commands

```
$ autoreconf --install
$ ./configure
$ make
$ make distcheck
```

2.2 Fortran Hello World: fhello

Under the subfolder src/ put these two files fhello.f

```
  program MAIN
    print*,'Hello World!'
  end
```

Makefile.am

```
bin_PROGRAMS = fhello
fhello_SOURCES = fhello.f
```

Under the root folder put these two files
configure.ac

```
AC_INIT([fhello], [0], [astrobear])
AM_INIT_AUTOMAKE([foreign -Wall -Werror])
AC_PROG_CC
AC_PROG_F77([gfortran mpif90])
AC_PROG_FC([gfortran mpif90])
AC_CONFIG_FILES([Makefile src/Makefile])
AC_OUTPUT
```

Makefile.am

```
SUBDIRS = src
```

To test use these lines of commands

```
$ autoreconf --install
$ ./configure
$ make
$ make distcheck
```

2.3 MPI Fortran Hello World: phello

Under the subfolder src/ put these two files phello.f

```
program phello
include 'mpif.h'
integer rank, size, ierror, tag, status(MPI_STATUS_SIZE)

```

```
call MPI_INIT(ierror)
call MPI_COMM_SIZE(MPI_COMM_WORLD, size, ierror)
call MPI_COMM_RANK(MPI_COMM_WORLD, rank, ierror)
print*, 'node', rank, ': Hello World'
call MPI_FINALIZE(ierror)
end
```

Makefile.am

```
bin_PROGRAMS = phello
phello_SOURCES = phello.f90
```

Under the root folder put these two files
configure.ac

```
AC_INIT([phello], [1.0], [bliu@pas.rochester.edu])
AM_INIT_AUTOMAKE([foreign -Wall -Werror])
AC_PROG_CC
AC_PROG_F77([mpif90 ifort])
AC_PROG_FC([mpif90 ifort])
AC_CONFIG_FILES([Makefile src/Makefile])
AC_OUTPUT
```

Makefile.am

```
SUBDIRS = src
```

To test use these lines of commands

```
$ autoreconf --install
$ ./configure
$ make
$ make distcheck
```

2.4 Prefix and Parallel HDF5: phdf5

This is an unsuccessful example which will be revisited in next section. The code (filecreate.f90) in this example will create a parallel hdf5. The Autotool is used to test if the header file hdf5.h (AC_CHECK_HEADERS) and if the h5create function can be found in the library. This example will also take the prefix directory.

The code can be compiled manually with these command

```
$ mpif90 -I/bamboodata/bliu/opt/hdf5_parallel_1.8.12/include -g -c
      -o filecreate-filecreate.o
$ mpif90 -L/bamboodata/bliu/opt/hdf5_parallel_1.8.12/lib -o
      filecreate filecreate-filecreate.o -lhdf5_fortran -lhdf5 -lz
```

Under the subfolder src/, we put these two files filecreate.f90

```
!
! This example creates HDF5 file in a parallel environment
!

    PROGRAM FILE_CREATE

    USE HDF5 ! This module contains all necessary modules

    IMPLICIT NONE

    INCLUDE 'mpif.h'
    CHARACTER(LEN=10), PARAMETER :: filename = "sds.h5" ! File
        name

    INTEGER(HID_T) :: file_id    ! File identifier
    INTEGER(HID_T) :: plist_id   ! Property list identifier
    INTEGER        :: error

    !
    ! MPI definitions and calls.
    !
    INTEGER :: mpierror     ! MPI error flag
    INTEGER :: comm, info
    INTEGER :: mpi_size, mpi_rank
    comm = MPI_COMM_WORLD
    info = MPI_INFO_NULL

    CALL MPI_INIT(mpierror)
    CALL MPI_COMM_SIZE(comm, mpi_size, mpierror)
    CALL MPI_COMM_RANK(comm, mpi_rank, mpierror)
    !
    ! Initialize FORTRAN predefined datatypes
    !
    CALL h5open_f(error)

    !
    ! Setup file access property list with parallel I/O access.
    !
    CALL h5pcreate_f(H5P_FILE_ACCESS_F, plist_id, error)
    CALL h5pset_fapl_mpio_f(plist_id, comm, info, error)

    !
    ! Create the file collectively.
    !
    CALL h5fcreate_f(filename, H5F_ACC_TRUNC_F, file_id, error,
        access_prp = plist_id)
```

```
45
46      !
47      ! Close property list and the file.
48      !
49      CALL h5pclose_f(plist_id, error)
50      CALL h5fclose_f(file_id, error)
51
52      !
53      ! Close FORTRAN interface
54      !
55      CALL h5close_f(error)
56
57      CALL MPI_FINALIZE(mpierror)
58
59      END PROGRAM FILE_CREATE
```

Makefile.am

```
1 bin_PROGRAMS = filecreate
2 filecreate_SOURCES = filecreate.f90
```

Under the root folder, we put these two files configure.ac

```
1  AC_INIT([phd5], [1.0], [Baowei])
2  AM_INIT_AUTOMAKE([foreign -Wall -Werror])
3  AC_PROG_F77([mpif90 ifort])
4  AC_PROG_FC([mpif90 ifort])
5  AC_ARG_VAR(MPICC,[MPI C compiler command])
6  AC_CHECK_PROGS(MPICC,mpicc,mpxlc,$CC)
7  CC="$MPICC"
8  AC_SUBST(MPICC)
9  AC_CONFIG_FILES([Makefile src/Makefile])
10
11 dnl
       ------------------------------------
12 dnl Where's HDF5? We need to find out where the HDF5 library
13 dnl and header files are if they aren't in system library/include places
14 dnl
15
16 AC_ARG_WITH([hdf5],
17              [AC_HELP_STRING([--with-hdf5=DIR],
18                [path for HDF5 library])],WITH_HDF5="yes"
                      HDF5PATH="$withval")
19
20 AM_CONDITIONAL(WITH_HDF5, test x$WITH_HDF5 = xyes)
21
22 if test -n "$WITH_HDF5"; then
23     HDF5_INC="$HDF5PATH/include"
24     HDF5_LIB="$HDF5PATH/lib"
```

```
    CPPFLAGS="$CPPFLAGS -I$HDF5_INC"
    LD_LIBRARY_PATH="$HDF5_LIB:$LD_LIBRARY_PATH"
    LDFLAGS="$LDFLAGS -L$HDF5_LIB"
fi

AC_SUBST(HDF5_INC)
AC_SUBST(HDF5_LIB)


dnl Check for parallel hdf5 header and functions


AC_CHECK_HEADERS(hdf5.h, break ,
                    AC_MSG_ERROR(1 cannot find HDF5 header files.))

AC_CHECK_LIB(hdf5, h5pcreate_f,,
                AC_MSG_ERROR(cannot find the parallel HDF5 library.))


AC_OUTPUT
```

Makefile.am

```
SUBDIRS = src
```

1. First we test the case with no hdf5 prefix provided

```
$ autoreconf --install
$ ./configure
```

The standard output file looks like this

```
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
/bamboodata/bliu/myTickets/255/myExamples/pHDF/missing:
      Unknown `--is-lightweight' option
Try `/bamboodata/bliu/myTickets/255/myExamples/pHDF/missing
      --help' for more information
configure: WARNING: 'missing' script is too old or missing
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking for mpif90... mpif90
checking whether the Fortran 77 compiler works... yes
checking for Fortran 77 compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU Fortran 77 compiler... no
checking whether mpif90 accepts -g... yes
```

```
checking for mpif90... mpif90
checking whether we are using the GNU Fortran compiler... no
checking whether mpif90 accepts -g... yes
checking for mpicc... no
checking for style of include used by make... GNU
checking for gcc... mpxlc
checking whether we are using the GNU C compiler... no
checking whether mpxlc accepts -g... no
checking for mpxlc option to accept ISO C89... unsupported
checking whether mpxlc understands -c and -o together... no
checking dependency style of /bamboodata/bliu/myTickets/255/
    myExamples/pHDF/compile mpxlc... none
checking how to run the C preprocessor... /lib/cpp
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for ANSI C header files... no
checking for sys/types.h... no
checking for sys/stat.h... no
checking for stdlib.h... no
checking for string.h... no
checking for memory.h... no
checking for strings.h... no
checking for inttypes.h... no
checking for stdint.h... no
checking for unistd.h... no
checking hdf5.h usability... no
checking hdf5.h presence... no
checking for hdf5.h... no
configure: error: 1 cannot find HDF5 header files.
```

So configure cannot find the HDF5 header file.

2. Then we test the case with serial HDF library

```
$ ./configure --with-hdf5=/usr/local/alt/hdf5_185p1_ifort/
```

Set the location in configure.ac The standard output file looks like this

```
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
/bamboodata/bliu/myTickets/255/myExamples/pHDF/missing:
    Unknown `--is-lightweight' option
Try `/bamboodata/bliu/myTickets/255/myExamples/pHDF/missing
    --help' for more information
configure: WARNING: 'missing' script is too old or missing
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking for mpif90... mpif90
checking whether the Fortran 77 compiler works... yes
checking for Fortran 77 compiler default output file name... a.out
```

```
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU Fortran 77 compiler... no
checking whether mpif90 accepts -g... yes
checking for mpif90... mpif90
checking whether we are using the GNU Fortran compiler... no
checking whether mpif90 accepts -g... yes
checking for mpicc... no
checking for style of include used by make... GNU
checking for gcc... mpxlc
checking whether we are using the GNU C compiler... no
checking whether mpxlc accepts -g... no
checking for mpxlc option to accept ISO C89... unsupported
checking whether mpxlc understands -c and -o together... no
checking dependency style of /bamboodata/bliu/myTickets/255/
    myExamples/pHDF/compile mpxlc... none
checking how to run the C preprocessor... /lib/cpp
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for ANSI C header files... no
checking for sys/types.h... no
checking for sys/stat.h... no
checking for stdlib.h... no
checking for string.h... no
checking for memory.h... no
checking for strings.h... no
checking for inttypes.h... no
checking for stdint.h... no
checking for unistd.h... no
checking hdf5.h usability... no
checking hdf5.h presence... yes
configure: WARNING: hdf5.h: present but cannot be compiled
configure: WARNING: hdf5.h: check for missing prerequisite headers?
configure: WARNING: hdf5.h: see the Autoconf documentation
configure: WARNING: hdf5.h: section "Present But Cannot Be
    Compiled"
configure: WARNING: hdf5.h: proceeding with the compiler's result
configure: WARNING: ## ---------------------- ##
configure: WARNING: ## Report this to Baowei ##
configure: WARNING: ## ---------------------- ##
checking for hdf5.h... no
configure: error: 1 cannot find HDF5 header files.
```

3. Next we test the case with parallel HDF library

```
$ ./configure --with-hdf5=/bamboodata/bliu/opt/hdf5_parallel_1
    .8.12
```

And the standard output file looks like this

```
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
/bamboodata/bliu/myTickets/255/myExamples/pHDF/missing:
      Unknown `--is-lightweight' option
Try `/bamboodata/bliu/myTickets/255/myExamples/pHDF/missing
      --help' for more information
configure: WARNING: 'missing' script is too old or missing
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking for mpif90... mpif90
checking whether the Fortran 77 compiler works... yes
checking for Fortran 77 compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU Fortran 77 compiler... no
checking whether mpif90 accepts -g... yes
checking for mpif90... mpif90
checking whether we are using the GNU Fortran compiler... no
checking whether mpif90 accepts -g... yes
checking for mpicc... no
checking for style of include used by make... GNU
checking for gcc... mpxlc
checking whether we are using the GNU C compiler... no
checking whether mpxlc accepts -g... no
checking for mpxlc option to accept ISO C89... unsupported
checking whether mpxlc understands -c and -o together... no
checking dependency style of /bamboodata/bliu/myTickets/255/
      myExamples/pHDF/compile mpxlc... none
checking how to run the C preprocessor... /lib/cpp
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for ANSI C header files... no
checking for sys/types.h... no
checking for sys/stat.h... no
checking for stdlib.h... no
checking for string.h... no
checking for memory.h... no
checking for strings.h... no
checking for inttypes.h... no
checking for stdint.h... no
checking for unistd.h... no
checking hdf5.h usability... no
checking hdf5.h presence... no
checking for hdf5.h... no
configure: error: 1 cannot find HDF5 header files.
```

The configure cannot find the HDF5 header file again. This is probably due to the gcc couldn't find the HDF5 compiled with mpicc.

2.5 Parallel HDF5 revisit

Use AC_CHECK_FUNC instead. And this new configure.ac seems working although with warnings. Pay attention to how the linking library and flags are set in configure.ac and Makefile.am – AC_SUBST has to be used to define the paths so they can be used in Makefile.am. The .f90 code and folder structures are same as above.

Under the subfolder src/, we put these two files filecreate.f90

```
1  !
2  ! This example creates HDF5 file in a parallel environment
3  !
4
5      PROGRAM FILE_CREATE
6
7      USE HDF5 ! This module contains all necessary modules
8
9      IMPLICIT NONE
10
11     INCLUDE 'mpif.h'
12     CHARACTER(LEN=10), PARAMETER :: filename = "sds.h5" ! File
           name
13
14     INTEGER(HID_T) :: file_id    ! File identifier
15     INTEGER(HID_T) :: plist_id   ! Property list identifier
16     INTEGER      :: error
17
18     !
19     ! MPI definitions and calls.
20     !
21     INTEGER :: mpierror     ! MPI error flag
22     INTEGER :: comm, info
23     INTEGER :: mpi_size, mpi_rank
24     comm = MPI_COMM_WORLD
25     info = MPI_INFO_NULL
26
27     CALL MPI_INIT(mpierror)
28     CALL MPI_COMM_SIZE(comm, mpi_size, mpierror)
29     CALL MPI_COMM_RANK(comm, mpi_rank, mpierror)
30     !
31     ! Initialize FORTRAN predefined datatypes
32     !
33     CALL h5open_f(error)
```

```
34
35      !
36      ! Setup file access property list with parallel I/O access.
37      !
38      CALL h5pcreate_f(H5P_FILE_ACCESS_F, plist_id, error)
39      CALL h5pset_fapl_mpio_f(plist_id, comm, info, error)
40
41      !
42      ! Create the file collectively.
43      !
44      CALL h5fcreate_f(filename, H5F_ACC_TRUNC_F, file_id, error,
            access_prp = plist_id)
45
46      !
47      ! Close property list and the file.
48      !
49      CALL h5pclose_f(plist_id, error)
50      CALL h5fclose_f(file_id, error)
51
52      !
53      ! Close FORTRAN interface
54      !
55      CALL h5close_f(error)
56
57      CALL MPI_FINALIZE(mpierror)
58
59      END PROGRAM FILE_CREATE
```

Makefile.am

```
1 bin_PROGRAMS = filecreate
2 filecreate_SOURCES = filecreate.f90
3 filecreate_FCFLAGS = -I$(HDF5_INC)
4 filecreate_LDFLAGS = -L$(HDF5_LIB)
5 filecreate_LDADD = $(HDF5_LDFLAGS)
```

Under the root folder, we put these two files configure.ac

```
1  AC_INIT([phd5], [1.0], [Baowei])
2  AM_INIT_AUTOMAKE([foreign -Wall -Werror])
3  AC_LANG(Fortran)
4  AC_PROG_F77([mpif90 ifort])
5  AC_PROG_FC([mpif90 ifort])
6  dnl AC_ARG_VAR(MPICC,[MPI C compiler command])
7  dnl AC_CHECK_PROGS(MPICC,mpicc,mpxlc,$CC)
8  CC="$MPICC"
9  AC_SUBST(MPICC)
10 AC_CONFIG_FILES([Makefile src/Makefile])
11
```

```
12 dnl
       ------------------------------------
13 dnl Where's HDF5? We need to find out where the HDF5 library
14 dnl and header files are if they aren't in system library/include places
15 dnl
16
17 dnl AC_HELP_STRING is obsolete for Autotools 2.69
18 dnl AC_ARG_WITH([hdf5],
19 dnl [AC_HELP_STRING([--with-hdf5=DIR],
20 dnl       [path for HDF5 library])],WITH_HDF5="yes" HDF5PATH="
       $withval")
21 dnl
22
23 AC_ARG_WITH([hdf5],
24              [AS_HELP_STRING([--with-hdf5=DIR],
25                [enable path for HDF5 library])],
26                                  [WITH_HDF5="yes" HDF5PATH="
                                        $withval"],
27                                  [WITH_HDF5="no"])
28
29 AM_CONDITIONAL(WITH_HDF5, test x$WITH_HDF5 = xyes)
30
31 if test -n "$WITH_HDF5"; then
32    HDF5_INC="$HDF5PATH/include"
33    HDF5_LIB="$HDF5PATH/lib"
34    HDF5_LDFLAGS="-lhdf5_fortran -lhdf5 -lz"
35 fi
36
37
38 dnl Check for parallel hdf5 header and functions
39
40
41 AC_CHECK_HEADERS(hdf5.h, [HAVE_HDF5_HEADER="yes"],
42                     AC_MSG_ERROR(1 cannot find HDF5 header files.))
43
44 AC_CHECK_FUNC(hdf5, h5pcreate_f,[HAVE_HDF5_FUNC="yes"],
45                AC_MSG_ERROR(cannot find the parallel HDF5 library.))
46
47 dnl if the hdf5 header file and library found
48 dnl define Include and lib paths and flags so they could be used in
       Makefile.am
49 AC_SUBST(HDF5_INC)
50
51 LD_LIBRARY_PATH="$HDF5_LIB:$LD_LIBRARY_PATH"
52 AC_SUBST(HDF5_LIB)
53 AC_SUBST(HDF5_LDFLAGS)
54
55 dnl echo
56 dnl echo OutputVariables: {F90FLAGS=$all_includes} \
```

```
dnl {F90LIBS=$all_libs}
dnl echo

AC_OUTPUT
```

Makefile.am

```
SUBDIRS = src
```

Here we try with parallel HDF library

```
$ autoreconf --install
$ ./configure --with-hdf5=/bamboodata/bliu/opt/hdf5_parallel_1.8.12
$ make
```

And the corresponding standard output look like these

1. autoreconf

```
configure.ac:41: warning: AC_LANG_PREPROC(Fortran): No preprocessor defined for Fortran
../../lib/autoconf/fortran.m4:245: AC_LANG_PREPROC(Fortran) is expanded from...
../../lib/autoconf/lang.m4:372: AC_LANG_PREPROC_REQUIRE is expanded from...
../../lib/autoconf/general.m4:2518: AC_PREPROC_IFELSE is expanded from...
../../lib/m4sugar/m4sh.m4:639: AS_IF is expanded from...
../../lib/m4sugar/m4sh.m4:385: AS_REQUIRE is expanded from...
../../lib/autoconf/general.m4:181: AC_REQUIRE_SHELL_FN is expanded from...
../../lib/autoconf/headers.m4:129: _AC_CHECK_HEADER_MONGREL is expanded from...
../../lib/autoconf/headers.m4:67: AC_CHECK_HEADER is expanded from...
../../lib/m4sugar/m4sh.m4:607: AS_FOR is expanded from...
../../lib/autoconf/headers.m4:249: AC_CHECK_HEADERS is expanded from...
configure.ac:41: the top level
configure.ac:41: warning: AC_LANG_PROGRAM(Fortran): ignoring PROLOGUE: [#include <stdlib.h>
configure.ac:41: #include <stdarg.h>
configure.ac:41: #include <string.h>
configure.ac:41: #include <float.h>
configure.ac:41: ]
../../lib/autoconf/lang.m4:224: AC_LANG_SOURCE is expanded from...
../../lib/autoconf/lang.m4:241: AC_LANG_PROGRAM is expanded from...
../../lib/autoconf/lang.m4:193: AC_LANG_CONFTEST is expanded from...
../../lib/autoconf/general.m4:2590: _AC_COMPILE_IFELSE is expanded from...
```

```
../../lib/autoconf/general.m4:2606: AC_COMPILE_IFELSE is expanded
    from...
../../lib/m4sugar/m4sh.m4:639: AS_IF is expanded from...
../../lib/autoconf/general.m4:2031: AC_CACHE_VAL is expanded from
    ...
../../lib/autoconf/general.m4:2052: AC_CACHE_CHECK is expanded
    from...
../../lib/autoconf/headers.m4:677: AC_HEADER_STDC is expanded
    from...
../../lib/autoconf/headers.m4:297:
    _AC_INCLUDES_DEFAULT_REQUIREMENTS is expanded from...
../../lib/autoconf/headers.m4:353: AC_INCLUDES_DEFAULT is
    expanded from...
../../lib/autoconf/headers.m4:129: _AC_CHECK_HEADER_MONGREL
    is expanded from...
../../lib/autoconf/headers.m4:67: AC_CHECK_HEADER is expanded
    from...
../../lib/m4sugar/m4sh.m4:607: AS_FOR is expanded from...
../../lib/autoconf/headers.m4:249: AC_CHECK_HEADERS is expanded
    from...
configure.ac:41: the top level
```

2. configure

```
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
/bamboodata/bliu/myTickets/255/myExamples/pHDF2/missing:
    Unknown `--is-lightweight' option
Try `/bamboodata/bliu/myTickets/255/myExamples/pHDF2/missing
    --help' for more information
configure: WARNING: 'missing' script is too old or missing
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking for mpif90... mpif90
checking whether the Fortran 77 compiler works... yes
checking for Fortran 77 compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU Fortran 77 compiler... no
checking whether mpif90 accepts -g... yes
checking for mpif90... mpif90
checking whether we are using the GNU Fortran compiler... no
checking whether mpif90 accepts -g... yes
checking for grep that handles long lines and -e... /bin/grep
checking for egrep... /bin/grep -E
checking for ANSI C header files... no
checking for sys/types.h... yes
```

```
checking for sys/stat.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for memory.h... yes
checking for strings.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for unistd.h... yes
checking hdf5.h usability... yes
checking hdf5.h presence... no
configure: WARNING: hdf5.h: accepted by the compiler, rejected by the
    preprocessor!
configure: WARNING: hdf5.h: proceeding with the compiler's result
checking for hdf5.h... yes
checking for hdf5... no
checking that generated files are newer than configure... done
configure: creating ./config.status
config.status: creating Makefile
config.status: creating src/Makefile
```

3. make

```
make: Warning: File 'Makefile' has modification time 17 s in the future
Making all in src
make[1]: Entering directory '/mnt/net/bamboodata/bliu/myTickets
    /255/myExamples/pHDF2/src'
make[1]: Warning: File 'Makefile' has modification time 17 s in the
    future
mpif90 -I/bamboodata/bliu/opt/hdf5_parallel_1.8.12/include -g -c -
    o filecreate-filecreate.o `test -f 'filecreate.f90' || echo './'`
    filecreate.f90
mpif90 -I/bamboodata/bliu/opt/hdf5_parallel_1.8.12/include -g -L/
    bamboodata/bliu/opt/hdf5_parallel_1.8.12/lib -o filecreate
    filecreate-filecreate.o -lhdf5_fortran -lhdf5 -lz
make[1]: warning: Clock skew detected. Your build may be incomplete.
make[1]: Leaving directory '/mnt/net/bamboodata/bliu/myTickets
    /255/myExamples/pHDF2/src'
make[1]: Entering directory '/mnt/net/bamboodata/bliu/myTickets
    /255/myExamples/pHDF2'
make[1]: Warning: File 'Makefile' has modification time 13 s in the
    future
make[1]: Nothing to be done for 'all-am'.
make[1]: warning: Clock skew detected. Your build may be incomplete.
make[1]: Leaving directory '/mnt/net/bamboodata/bliu/myTickets
    /255/myExamples/pHDF2'
make: warning: Clock skew detected. Your build may be incomplete.
```

2.6 Multiple Files: Dependencies

The GNU Autotools handles the dependencies for Fortran90 in a very tricky way: you would have to put the source files in a correct order –dependencies first and the EXTRA_SOURCE Macro which works for C does not work for current Fortran. "Automatic dependency tracking is, as far as I know, not (yet) possible. This could change with the addition of submodules, which are defined in Fortran 2008 but not yet implemented in any popular compiler" according to `http://stackoverflow.com/questions/17864763/automake-with-fortran-order-of-file`. In this example for the Makefile.am

```
bin_PROGRAMS = main
main_SOURCES = subfile.f90 main.f90
```

If you switch the order of subfiles.f90 and main.f90, the compiler won't know to compile subfile.o. Here's the output if you do

```
$ make
make[1]: Entering directory '/mnt/net/bamboodata/bliu/myTickets
    /255/myExamples/multiFiles/src'
make[1]: Warning: File 'Makefile' has modification time 13 s in the
    future
gfortran -g -O2 -c -o main.o main.f90
main.f90:2.6:

  use subfile, only:sub1
       1
Fatal Error: Can't open module file 'subfile.mod' for reading at (1): No
    such file or directory
make[1]: *** [main.o] Error 1
```

The complete example is shown here.
Under the root folder:
1. configure.ac

```
AC_INIT([MultiFiles], [0], [astrobear])
AM_INIT_AUTOMAKE([foreign -Wall -Werror])
AC_PROG_CC
AC_PROG_F77([gfortran mpif90])
AC_PROG_FC([gfortran mpif90])
AC_CONFIG_FILES([Makefile src/Makefile])

AC_OUTPUT
```

2. Makefile.am

```
SUBDIRS = src
```

Under the src subfolder
1. two code files

```
module subfile

contains

subroutine sub1()
  print *, "Hello from file1"
end subroutine sub1

end module subfile
```

and

```
program multiFile
  use subfile, only:sub1
  print *, "Hello from file2(main)"
  call sub1()
end program multiFile
```

2. Makefile.am as already shown above

```
bin_PROGRAMS = main
main_SOURCES = subfile.f90 main.f90
```

And the commands to try are same

```
$ autoreconf --install
$ ./configure
$ make
```

2.7 Non-recursive Make: Dependency with More Files and Subfolders

This example is to test more Fortran90 files and subfolders for dependencies. We use AM_INIT_AUTOMAKE([subdir-objects]) in configure.ac to enable the non-recursive make – no need to put Makefile.am under each subfolders. Also pay attention to how the sources are given in Makefile.am

Under the root folder

1. Configure.ac

```
AC_INIT([MultiFiles], [0], [astrobear])
AM_INIT_AUTOMAKE([subdir-objects])
AC_PROG_CC
AC_PROG_F77([gfortran mpif90])
AC_PROG_FC([gfortran mpif90])
AC_CONFIG_FILES([Makefile src/Makefile])

AC_OUTPUT
```

2. Makefile.am

```
SUBDIRS = src
```

Under the src/ subfolder

1. two code files:

```
module subfile

contains

subroutine sub1()
  use subfile2, only:sub2
  print *, "Hello from file1"
  call sub2()
end subroutine sub1

end module subfile
```

and

```
program multiFile
  use subfile, only:sub1
  print *, "Hello from file2(main)"
  call sub1()
end program multiFile
```

2. Makefile.am

```
SUBDIRS = src
```

Under the src/folder1/ subsubfolder, only need the code file

```
module subfile2

contains

subroutine sub2()
  print *, "Hello from subfolder"
end subroutine sub2

end module subfile2
```

And the commands to try are same

```
$ autoreconf --install
$ ./configure
$ make
```

2.8 Preprocessing: Conditional compilation

This example is to test the preprocessing or passing the source code through a C preprocessor to allow for conditional compilation. Automake assumes preprocessed Fortran with capitalized suffixes,.F(), .F95. So if you use them, you can stuff preprocessing flags globallly in configure.ac

Under the root folder

1. Configure.ac

```
AC_INIT([Preprocessing], [0], [astrobear])
AM_INIT_AUTOMAKE([foreign -Wall -Werror])
AC_PROG_CC([gcc])
AC_PROG_F77([gfortran mpif90])
AC_PROG_FC([gfortran mpif90])
AC_CONFIG_FILES([Makefile src/Makefile])

PREPROCESSOR_FLAG="-cpp"
AC_SUBST(PREPROCESSOR_FLAG)

AC_OUTPUT
```

2. Makefile.am

```
SUBDIRS = src
```

Under the src/ subfolder

1. Code file with .F90 suffix

```
program main

# if defined XX
   print*, "Hello XX!"
# endif

# if defined YY
   print*, "Hello YY!"
# else
   print*, "Hello  !"
# endif

end
```

2. Makefile.am

```
bin_PROGRAMS = main
main_SOURCES = main.F90
main_FCFLAGS = $(PREPROCESSOR_FLAG)
```

And the commands to try are same

```
$ autoreconf --install
$ ./configure
$ make
```

Chapter 3

Other Autoconf and Automake Grammers

3.1 Splitting Long Lines

Makefile.am: To formate makefile for readability in Makefile.am, use a backslash character.

3.2 Comment Lines

configure.ac

- The comment lines for configure.ac start with dnl will not appear in the resulting configure script.
- But the lines commented with '#' that occur after AC_INT will appear in the resulting configure
- For m4 macros use dnl comments as m4 macros do not appear in configure script, only their expansion

Makefile.am

- Makefile.am is not processed by m4 but by automake
- comment lines start with '##' will NOT carry through to Makefile.in
- commnet lines start with '###' will carry through to Makefile.in

3.3 Conditionals

Automake supports a simple type of conditionals. Automake conditionals are checked at configure time by the configure script, and affect the translation from

Makefile.in to Makefile. They are based on options passed to configure and on results that configure has discovered about the host system.

3.3.1 Usage of Conditionals

Before using a conditional, you must define it by using AM_CONDITIONAL in the configure.ac file. An example

```
AC_ARG_ENABLE([debug],
              [ --enable-debug Turn on debugging],
              [case "${enableval}" in
                    yes) debug=true ;;
                     no) debug=false ;;
                      *) AC_MSG_ERROR([bad value ${enableval} for
                            --enable-debug]) ;;
              esac],[debug=false])
AM_CONDITIONAL([DEBUG], [test x$debug = xtrue])
```

Here is an example of how to use the above conditional in Makefile.am

```
if DEBUG
  DBG = debug
else
  DBG =
endif
noinst_PROGRAMS = $(DBG)
```

3.3.2 Shell Commands

AC_OUTPUT_COMMANDS Specify additional shell commands to run at the end of 'config.status', and shell commands to initialize any variables from configure. This macro. Can be put anywhere after AC_INIT and before AC_OUTPUT. Example:

```
AC_INIT([fhello], [0], [])
...
AC_OUTPUT_COMMANDS([rm -rf src/Problem; cp -r src/module1
    src/Problem])
...
AC_OUTPUT
```

This macro may be called multiple times. It is obsolete, replaced by AC_CONFIG_COMMANDS. Use the latter will be much easier to pass vaiables to the Shell commands as this example shows:

```
AC_CONFIG_COMMANDS([cmdTest],[echo $MODULENAME],[
    MODULENAME=$MODULENAME])
```

3.3.3 Documents

```
automake --add-missing
```

The –add-missing option copies some boilerplate files from your Automake into the current directory. Files such as COPYING which contain the GNU General Public License change infrequently, and so can be generated without user intervention. A number of utility scripts are also installed–these are used by the generated 'Makefile's. particularly by the install target. Notice that some required files are still missing. These are: 'NEWS': A record of user-visible changes to a package. The format is not strict, but the chcnages to the most recent version should appear at the top of the file 'README': The first place a user will look to get an overview for the purpose of a package, and perhaps special installation instructionss. 'AUTHORS': Lists the names and usually mail addresses, of individuals who worked on the package. 'ChangeLog': The ChangeLog is an important file–it records the changes that are made to a package. The format of this file is quite strict.

3.3.4 Conditional and Exit if Error

```
if test x"$WITH_MODULE" == xyes; then
    AC_OUTPUT_COMMANDS([echo compile with module
        $MODULENAME])
else
    AC_MSG_ERROR([Please specify the problem module with configure
        option --with-module... check the module list in README
        or INSTALL.])
fi
```

Or

```
if test x"$WITH_MODULE" != xyes; then
    AC_MSG_ERROR([Please specify the problem module with configure
        option --with-module... check the module list in README
        or INSTALL.])
fi
```

3.3.5 Check if the folder exist

```
  AC_CHECK_FILE("$RUNDIR", [RUNDIREXIST="yes"],[
      RUNDIREXIST="no"])
if test x"$RUNDIREXIST" == xyes; then
    AC_MSG_WARN([echo run directory $RUNDIR exists. A backup will
        be made.])
    AC_CONFIG_COMMANDS([BackupAndCreatRunDir],[now=`date
        +%s`; bkRunDir=run_dir_$now; mv $RUNDIR $bkRunDir;
        mkdir $RUNDIR; cp -r ./src/modules/Problem/* $RUNDIR
        ;],[RUNDIR=$RUNDIR])
```

```
else
    AC_CONFIG_COMMANDS([CreatRunDir],[rm -rf $RUNDIR; mkdir
        $RUNDIR; cp -r ./src/modules/Problem/* $RUNDIR;],[
        RUNDIR=$RUNDIR])
fi
```

3.3.6 Test if the string is empty

```
dnl if HDF5PATH is not empty
if test -n x$HDF5PATH; then
    HDF5INC="-I$HDF5PATH/include"
    HDF5LIBS="-L$HDF5PATH/lib"
    CPPFLAGS="$CPPFLAGS -I$HDF5_INC"
    LD_LIBRARY_PATH="$HDF5_LIB:$LD_LIBRARY_PATH"
    LDFLAGS="$LDFLAGS -L$HDF5_LIB"
    HDF5LDFLAGS="-lhdf5_fortran -lhdf5 -lz"
else AC_MSG_ERROR([cannot find HDF5 package installed. If the
    HDF5 library has been installed on your computer, you may try to
    set the path manually with configure option '--with-hdf5='.
    Otherwise you may download the latest version of hdf5 package
    from http://www.hdfgroup.org/downloads/index.html and install
    it.])
fi
```

3.4 Other Important Errors and Fixes

3.4.1 ipo: error #11021

The error looks like this

```
ipo: error #11021: OBJREAD Error: Could not create mapping for /usr
    /local/alt/hypre_260b_ifort/include
```

The fix is: When doing library include, each library needs a "-I" Otherwise it will got errors:

```
HDF5INC=-I/usr/local/alt/hdf5_185p1_ifort/include
HYPREINC=-I/usr/local/alt/hypre_260b_ifort/include
```

When compiling it will be something like

```
mpif90 -I/usr/local/alt/hdf5_185p1_ifort/include -I/usr/local/alt/
    hypre_260b_ifort/include ....
```

3.4.2 undefined reference to 'MAIN__'

The error looks like this

```
1 /opt/intel/fce/10.1.018/lib/for_main.o: In function 'main':
2 /export/users/nbtester/efi2linuxx86_nightly/branch-10_1/20080805
      _000000/libdev/frtl/src/libfor/for_main.c:(.text+0x26): undefined
      reference to 'MAIN__'
3 make[1]: *** [astrobear] Error 1
```

The main program file for AstroBEAR is scrambler.f90 or F90 and this file need to be included to avoid this error.

Chapter 4

Autotools files for AstroBEAR

4.1 Introduction

For AstroBEAR we use the same layer structures as the examples above: we put all the .f90 code in src/ directory which includes a Makefile.am. Under the root we have configure.ac, Makefile.am. Some of the AstroBEAR code files include C macros which needs preprocessing. One way to do the preprocessing is to use file name as .F90 and let the mpif90 compiler to do the preprocessing. Somehow this ways doesn't work out correctly with Autotools. So we gcc instead. We set these options for the configure file

4.1.1 configure.ac

```
1  AC_INIT([AstroBEAR],[3.0],[baowei.liu@rochester.edu])
2  AM_INIT_AUTOMAKE([foreign subdir-objects])
3  dnl AM_INIT_AUTOMAKE([foreign -Wall -Werror])
4  AC_LANG(Fortran)
5  AC_PROG_CC([mpicc gcc])
6  AC_PROG_F77([mpif90])
7  AC_PROG_FC([mpif90])
8  AC_CONFIG_FILES([Makefile src/Makefile])
9
10
11 dnl === Preprocessing Flag
12 PREPROCESSOR_FLAG="-cpp -E -p"
13
14 CC="mpicc"
15 CPC="gcc"
16 AC_SUBST([CC])
17 AC_SUBST([CPC])
```

```
18 AC_SUBST([PREPROCESSOR_FLAG])
19
20 dnl ========== Must choose a Problem module
       ==========
21 AC_ARG_WITH([module],
22               [AS_HELP_STRING([--with-module=name],
23               [Get problem module name])],
24                       [WITH_MODULE="yes" MODULENAME="
                             $withval"],
25                       [WITH_MODULE="no"])
26
27 dnl AM_CONDITIONAL(WITH_MODULE, test x$WITH_MODULE =
       xyes)
28 if test x"$WITH_MODULE" == xyes; then
29     AC_OUTPUT_COMMANDS([echo compile with module
           $MODULENAME])
30 else
31 dnl AC_MSG_ERROR([Please specify the problem module with
       configure option --with-module... check the module list in
       README or INSTALL.])
32     AC_MSG_NOTICE([          ])
33     AC_MSG_NOTICE([****************** MODULE PROBLEM
           SECTION **********************])
34     AC_MSG_NOTICE([* *])
35     AC_MSG_NOTICE([* building with the default Template problem
           module. *])
36     AC_MSG_NOTICE([* To change the problem module, use the option
            --with-module. *])
37     AC_MSG_NOTICE([* To see a list of problem module directories use
            command: *])
38     AC_MSG_NOTICE([* ls src/modules/ *])
39     AC_MSG_NOTICE([* *])
40     AC_MSG_NOTICE
           ([************************************************************])

41     AC_MSG_NOTICE([          ])
42     MODULENAME="Template"
43 fi
44
45 AC_SUBST(MODULENAME)
46
47 dnl ==== generate the Problem Directory =======
48 dnl AC_OUTPUT_COMMANDS([echo $MODULENAME])
49 AC_CONFIG_COMMANDS([LinkModule],[cd src; rm -rf Problem; ln -
       s modules/$MODULENAME Problem; cd ../],[MODULENAME=
       $MODULENAME])
50 dnl AC_CONFIG_COMMANDS([cdmTest],[echo $MODULENAME;
       echo success ],[MODULENAME=$MODULENAME])
51
```

```
52 dnl AC_OUTPUT_COMMANDS([cd src/modules; rm -rf Problem; ln
       -s $MODULENAME Problem; cd ../../])
53
54 dnl === Create run directory ==========
55 AC_ARG_WITH([rundir],
56              [AS_HELP_STRING([--with-run_dir=DIR],
57              [Get run directory path])],
58                      [WITH_RUNDIR="yes" RUNDIR="$withval"],
59                      [WITH_RUNDIR="no" RUNDIR="./run_dir/"])
60
61 dnl check if the run directory exists
62 AC_CHECK_FILE("$RUNDIR", [RUNDIREXIST="yes"],[
       RUNDIREXIST="no"])
63 if test x"$RUNDIREXIST" == xyes; then
64     AC_MSG_WARN([run directory $RUNDIR exists. A backup will be
           made.])
65     AC_CONFIG_COMMANDS([BackupAndCreatRunDir],[now='date
           +%s'; bkRunDir=run_dir_$now; mv $RUNDIR $bkRunDir;
           mkdir $RUNDIR; mkdir $RUNDIR/out; cp -r ./src/Problem/*
            $RUNDIR;],[RUNDIR=$RUNDIR])
66 else
67     AC_CONFIG_COMMANDS([CreatRunDir],[rm -rf $RUNDIR; mkdir
           $RUNDIR; mkdir $RUNDIR/out; cp -r ./src/Problem/*
           $RUNDIR;],[RUNDIR=$RUNDIR])
68 fi
69
70
71 RUNDIR2="../$RUNDIR"
72 AC_SUBST(RUNDIR)
73 AC_SUBST(RUNDIR2)
74
75 dnl ============================
76
77 dnl ============ Check for HDF5 library
       ============
78
79 AC_ARG_WITH([hdf5],
80              [AS_HELP_STRING([--with-hdf5=DIR],
81              [enable path for HDF5 library])],
82                                  [WITH_HDF5="yes" HDF5PATH="
                                        $withval"],
83                                  [WITH_HDF5="no"])
84
85 AM_CONDITIONAL(WITH_HDF5, test x$WITH_HDF5 = xyes)
86
87 dnl if HDF5PATH is not empty
88 if test -n x$HDF5PATH; then
89     HDF5INC="-I$HDF5PATH/include"
90     HDF5LIBS="-L$HDF5PATH/lib"
91     CPPFLAGS="$CPPFLAGS -DHDF5"
```

```
92      LD_LIBRARY_PATH="$HDF5_LIB:$LD_LIBRARY_PATH"
93      LDFLAGS="$LDFLAGS -L$HDF5_LIB"
94      HDF5LDFLAGS="-lhdf5_fortran -lhdf5 -lz"
95      dnl Check for parallel hdf5 header and functions
96      AC_CHECK_HEADERS(hdf5.h, [HAVE_HDF5_HEADER="yes"] ,
97                       AC_MSG_ERROR(cannot find HDF5 header file. If
                              the HDF5 library has been installed on your
                             computer, you may try to set the path
                             manually with configure option '--with-
                             hdf5='. Otherwise you may download the
                             latest version of hdf5 package from http://
                             www.hdfgroup.org/downloads/index.html
                             and install it. And make sure you have --
                             enable-fortran when doing configure.))
98
99      dnl AC_CHECK_LIB(hdf5, h5pcreate_f,,
100     dnl AC_MSG_ERROR(cannot find the parallel HDF5 library.))
101
102     AC_SUBST(HDF5INC)
103     AC_SUBST(HDF5LIBS)
104     AC_SUBST(HDF5LDFLAGS)
105
106 else
107     AC_MSG_ERROR([cannot find HDF5 installation path. If the HDF5
              library has been installed on your computer, you may try to set
              the path manually with configure option '--with-hdf5='.
              Otherwise you may download the latest version of hdf5 package
               from http://www.hdfgroup.org/downloads/index.html and
              install it.])
108 fi
109
110 dnl ========== HDF5 library check complete
        ==================
111
112 dnl ========== Check for FFTW3 library
        =============
113
114 AC_ARG_WITH([fftw3],
115              [AS_HELP_STRING([--with-fftw3=DIR],
116                              [enable path for fftw3 library])],
117                              [WITH_fftw3="yes" FFTW3PATH="
                                     $withval"],
118                              [WITH_fftw3="no"])
119
120 AM_CONDITIONAL(WITH_fftw3, test x$WITH_fftw3 = xyes)
121
122 if test -n "$FFTW3PATH"; then
123     FFTW3INC="-I$FFTW3PATH/include"
124     FFTW3LIBS="-L$FFTW3PATH/lib"
125     CPPFLAGS="$CPPFLAGS -DFFTW3"
```

```
126     FFTW3LDFLAGS="-lfftw3"
127     AC_CHECK_HEADERS(fftw3.h, [HAVE_fftw3_HEADER="yes"],
128                           AC_MSG_ERROR(cannot find fftw3 header files. If
                                  fftw3 has been installed on your computer,
                                  you may try to set the path mannually with
                                  configure option '--with-fftw3='.
                                  Otherwise you may download the latest
                                  version of fftw3 package from http://www.
                                  fftw.org/download.html and install it.))
129
130
131     AC_SUBST(FFTW3INC)
132     AC_SUBST(FFTW3LIBS)
133     AC_SUBST(FFTW3LDFLAGS)
134 else
135     AC_MSG_ERROR([cannot find fftw3 installation path. If fftw3 has
            been installed on your computer, you may try to set the path
            mannually with configure option '--with-fftw3='. Otherwise
            you may download the latest version of fftw3 package from
            http://www.fftw.org/download.html and install it.])
136 fi
137
138
139 dnl ============== FFTW3 check complete
        ==========
140
141
142 dnl ============= Check for Hypre
        ====================
143 AC_ARG_WITH([hypre],
144              [AS_HELP_STRING([--with-hypre=DIR],
145                              [enable path for hypre library])],
146                              [WITH_hypre="yes" HYPREPATH="
                                     $withval"],
147                              [WITH_hypre="no"])
148
149 AM_CONDITIONAL(WITH_hypre, test x$WITH_hypre = xyes)
150
151 if test -n "$HYPREPATH"; then
152     HYPREINC="-I$HYPREPATH/include"
153     HYPREINC_C="-I$HYPREPATH/include"
154     HYPRELIBS="-L$HYPREPATH/lib"
155     CPPFLAGS="$CPPFLAGS -DHYPRE"
156     HYPRELDFLAGS="-lHYPRE"
157     AC_CHECK_HEADERS(HYPREf.h, [HAVE_hypre_HEADER="yes"],
158                       AC_MSG_ERROR(cannot find hypre header files. If
                              hypre has been installed on your computer, you
                              may try to set the path mannually with
                              configure option '--with-hypre='. Otherwise
                              you may download the latest hypre package
```

```
                        from https://computation-rnd.llnl.gov/
                        linear_solvers/software.php and install it. And
                        make sure you have --enable-fortran when
                        doing configure))
159
160     AC_SUBST(HYPREINC)
161     AC_SUBST(HYPREINC_C)
162     AC_SUBST(HYPRELIBS)
163     AC_SUBST(HYPRELDFLAGS)
164
165 else
166     AC_MSG_ERROR([cannot find hypre installation path. If hypre has
            been installed on your computer, you may try to set the path
            mannually with configure option '--with-hypre='. Otherwise
            you may download the latest hypre package from https://
            computation-rnd.llnl.gov/linear_solvers/software.php and
            install it.])
167 fi
168
169
170 dnl check if PTHREAD is enabled default is no,
171 dnl equivalent to PTHREADFLAG = 1
172 AC_ARG_WITH([pthread],
173               [AS_HELP_STRING([--with-pthread=DIR],
174                               [enable path for pthread library])],
175                               [WITH_pthread="yes" PTHREADPATH
                                      ="$withval"],
176                               [WITH_pthread="no"])
177
178 AM_CONDITIONAL(PTHREAD, test x$WITH_pthread = xyes)
179
180 if test "$WITH_pthread" = yes; then
181     PTHREADLDFLAGS="-DPTHRREADS"
182 else
183     PTHREADLDFLAGS=
184 fi
185
186 AC_SUBST(PTHREADLDFLAGS)
187
188 dnl check if PTHREAD2 is enabled default is no,
189 dnl equivalent to PTHREADFLAG = 1
190 AC_ARG_WITH([pthread2],
191               [AS_HELP_STRING([--with-pthread2=DIR],
192                               [enable path for pthread2 library])],
193                               [WITH_pthread2="yes"
                                      PTHREAD2PATH="$withval"],
194                               [WITH_pthread2="no"])
195
196 AM_CONDITIONAL(PTHREAD2, test x$WITH_pthread2 = xyes)
197
```

```
198 if test "$WITH_pthread2" = yes; then
199     PTHREAD2INC="-I$PTHREAD2PATH/include"
200     PTHREAD2LIBS="-L$PTHREAD2PATH/lib"
201     PTHREAD2LDFLAGS="-DPTH"
202 else
203     PTHREAD2INC=
204     PTHREAD2LIBS=
205     PTHREAD2LDFLAGS=
206 fi
207
208 AC_SUBST(PTHREAD2INC)
209 AC_SUBST(PTHREAD2LIBS)
210 AC_SUBST(PTHREAD2LDFLAGS)
211
212
213 PROBLEM="Problem"
214 AC_SUBST(PROBLEM)
215
216 AC_OUTPUT_COMMANDS([touch ./src/Problem/problem.f90])
217
218 AC_OUTPUT
```

4.1.2 Makefile.am

```
1 SUBDIRS = src
```

4.1.3 src/Makefile.am

```
1  AM_CFLAGS = $(HYPREINC_C)
2  bin_PROGRAMS = astrobear
3
4  PROBLEM = Problem
5  FFTWLIBS = $(FFTW3LIBS) -lfftw3
6
7  HDFINC =$(HDF5INC) $(HDF5LIBS) $(HDF5LDFLAGS)
8
9  ###MPIPLIBS = -L$(MPIPPATH)/lib -lmpiP
10
11 ###ifeq ($(HYPREFLAG),1)
12 ###HYPRELIBS=$(HYPRELIBS)
13 ###HYPREINC=$(HYPREINC)
14
15
16 HYPRE_OBJ_FILES = \
17         elliptic/extended_hypre_wrapper.o \
18         elliptic/self_gravity/multipole.o \
19         elliptic/self_gravity/poisson.o \
```

```
20          elliptic/diffusion/diffusionbc.o \
21          elliptic/diffusion/diffusion.o \
22          elliptic/radtransfer/radtransfer.o \
23          elliptic/elliptic_control.o
24  HCPPFLAGS = -DHYPRE
25  ###endif
26
27  EXPLICIT_OBJ_FILES = \
28                  explicit/resistive.o \
29          explicit/conductive.o \
30          explicit/bviscosity.o \
31          explicit/explicit_control.o
32
33  if PTHREAD
34
35  PTHREAD_OBJ_FILES = \
36                  threads/fpthread_lib.o \
37                  threads/thread_declarations.o \
38                  threads/thread_control.o
39
40  PCPPFLAGS = -DPTHREADS
41  endif
42
43  if PTHREAD2
44
45  PTHREAD_OBJ_FILES = \
46                  pth/fpth_lib.o \
47                  pth/pth_declarations.o \
48                  pth/pth_control.o
49
50  PCPPFLAGS = -DPTH
51
52  PTHREADLIBS=-L/$(PTH_PATH)/lib -lpth
53  PTHREADINC=-I/$(PTH_PATH)/include
54  endif
55
56
57
58  OBJ_FILES = global_declarations.o \
59                  processing/splines.o \
60                  objects/profiles.o \
61                  communication/message_declarations.o \
62                  communication/communication_declarations.o \
63                  data/boundary.o \
64                  data/slopelim.o \
65                  data/data_declarations.o \
66                  tree/tree_declarations.o \
67                  physics/physics_declarations.o \
68                  elliptic/elliptic_declarations.o \
69                  explicit/explicit_declarations.o \
```

```
70              hyperbolic/hyperbolic_declarations.o \
71              physics/specfun.o \
72              physics/EOS.o \
73              physics/EOS.o \
74              physics/abundances.o \
75              physics/common_functions.o \
76              physics/BE_module.o \
77              physics/Bondi.o \
78              timing.o \
79              amr/scheduling.o \
80              hyperbolic/riemann_solvers.o \
81              data/data_info_ops.o \
82              objects/object_declarations.o \
83              objects/tables.o \
84              physics/opacity.o \
85              source/chemistry.o \
86              source/AnalyticCooling.o \
87              source/IICooling.o \
88              source/DMCooling.o \
89              source/ZCooling.o \
90              source/cooling.o \
91              source/cylindrical.o \
92              source/rotating.o \
93              source/uniformgravity.o \
94              source/pointgravity.o \
95              objects/outflows.o \
96              source/selfgravity.o \
97              source/source_control.o \
98      $(EXPLICIT_OBJ_FILES) \
99              particle/particle_declarations.o \
100             particle/particle_info_ops.o \
101             particle/particle_level_ops.o \
102             particle/particle_advance.o \
103             particle/particle_comms.o \
104             particle/particle_control.o \
105             objects/shapes.o \
106             processing/emissions.o \
107             processing/fields.o \
108     processing/processing_declarations.o \
109             objects/profile_functions.o \
110             objects/interfaces.o \
111             objects/perturbations.o \
112             objects/vectorperturbations.o \
113             objects/ambients.o \
114             objects/clumps.o \
115             objects/disks.o \
116             objects/winds.o \
117             objects/uniformregions.o \
118             objects/splitregions.o \
119             objects/collidingflows.o \
```

```
120                 physics/physics_control.o \
121                 io/io_bov.o \
122                 io/io_curve.o \
123                 io/io_ppm.o \
124                 layouts/layout_declarations.o \
125                 layouts/layout_io.o \
126                 layouts/layout_comms.o \
127                 physics/PFFT.o \
128                 layouts/layout_control.o \
129                 processing/images.o \
130                 processing/histograms.o \
131                 processing/spectra.o \
132                 processing/pdfs.o \
133                 processing/cameras.o \
134                 processing/movies.o \
135                 processing/projections.o \
136                 processing/pointsamples.o \
137                 processing/totals.o \
138                 objects/refinements.o \
139                 objects/tests.o \
140                 objects/object_control.o \
141                 $(PROBLEM_OBJ_FILES) \
142                 module_control.o \
143                 hyperbolic/sweep/stencil_declarations.o \
144                 hyperbolic/sweep/sweep_declarations.o \
145                 hyperbolic/sweep/stencil_control.o \
146                 hyperbolic/sweep/sweep_scheme.o \
147                 hyperbolic/MUSCL/MUSCL_scheme.o \
148                 hyperbolic/hyperbolic_control.o \
149                 distribution/distribution_declarations.o \
150                 distribution/hilbert.o \
151                 distribution/distribution_control.o \
152                 communication/mpi_transmission.o \
153                 communication/mpi_packing.o \
154                 communication/communication_control.o \
155                 tree/tree_node_ops.o \
156                 tree/tree_parsing.o \
157                 tree/tree_level_comms.o \
158                 data/data_parsing.o \
159                 data/data_level_comms.o \
160                 data/data_level_ops.o \
161                 tree/tree_level_ops.o
162
163
164 POST_ELLIPTIC_OBJ_FILES = \
165                 processing/processing_control.o
166
167 POST_PTHREADS_OBJ_FILES = \
168                 amr/time_step.o \
169                 io/io_declarations.o \
```

```
170                 io/hdf5_declarations.o \
171                 io/chombo_declarations.o \
172                 io/io_chombo.o \
173                 io/io_okc.o \
174                 io/io_comms.o \
175                 io/io_parsing.o \
176                 io/io_control.o \
177         amr/amr_control.o \
178                 scrambler.o
179
180 PROBLEM_OBJ_FILES = \
181                 $(PROBLEM)/problem.o
182
183 MOD_FILES = *.mod
184
185 METHOD_INCLUDE_FILES = \
186           hyperbolic/i_dependencies.f90 \
187           hyperbolic/riemann_solvers.f90
188
189 EXECUTABLE = astrobear
190
191
192
193 default: $(EXECUTABLE)
194
195 bov2jpeg: global_declarations.o io/io_bov.o io/io_ppm.o processing/
        images.o bov2jpeg.o
196   $(FC) $(LFLAGS) bov2jpeg.o global_declarations.o io/io_bov.o io/
          io_ppm.o processing/images.o bov2jpeg.o -o bov2jpeg
197
198 astrobear: $(OBJ_FILES) $(HYPRE_OBJ_FILES) $(
        POST_ELLIPTIC_OBJ_FILES) $(PTHREAD_OBJ_FILES) $(
        POST_PTHREADS_OBJ_FILES)
199   $(FC) $(LFLAGS) $(OBJ_FILES) $(HYPRE_OBJ_FILES) $(
          POST_ELLIPTIC_OBJ_FILES) $(PTHREAD_OBJ_FILES) $(
          POST_PTHREADS_OBJ_FILES) $(HDFLIBS) $(HDFINC) $(
          HYPRELIBS) $(HYPREINC) $(HYPRELDFLAGS) $(
          PTHREADLIBS) $(FFTWLIBS) -o $(EXECUTABLE)
200   @echo 'Be sure to update/remove your profile.data file'; cp $(
          EXECUTABLE) $(RUNDIR2)
201
202 scrambler.o : scrambler.f90
203   $(CPP) $(PREPROCESSOR_FLAG) $(HCPPFLAGS) $(PCPPFLAGS
          ) scrambler.f90 -o scrambler.cpp.f90
204   $(FC) -c $(FFLAGS) scrambler.cpp.f90 -o scrambler.o
205
206 io/io_bov.o : io/io_bov.f90
207   $(CPP) $(PREPROCESSOR_FLAG) $(HCPPFLAGS) $(PCPPFLAGS
          ) io/io_bov.f90 -o io/io_bov.cpp.f90
208   $(FC) -c $(FFLAGS) io/io_bov.cpp.f90 -o io/io_bov.o
```

```
209
210 amr/amr_control.o : amr/amr_control.f90
211   $(CPP) $(PREPROCESSOR_FLAG) $(HCPPFLAGS) $(PCPPFLAGS
          ) amr/amr_control.f90 -o amr/amr_control.cpp.f90
212   $(FC) -c $(FFLAGS) amr/amr_control.cpp.f90 -o amr/amr_control.
          o
213
214 processing/processing_control.o : processing/processing_control.f90
215   $(CPP) $(PREPROCESSOR_FLAG) $(HCPPFLAGS) $(PCPPFLAGS
          ) processing/processing_control.f90 -o processing/
          processing_control.cpp.f90
216   $(FC) -c $(FFLAGS) processing/processing_control.cpp.f90 -o
          processing/processing_control.o
217
218 amr/time_step.o : amr/time_step.f90
219   $(CPP) $(PREPROCESSOR_FLAG) $(HCPPFLAGS) $(PCPPFLAGS
          ) amr/time_step.f90 -o amr/time_step.cpp.f90
220   $(FC) -c $(FFLAGS) amr/time_step.cpp.f90 -o amr/time_step.o
221
222 if PTHREAD2
223 communication/mpi_transmission.o : communication/
        message_declarations.o communication/mpi_transmission.f90 pth/
        fpth_lib.o pth/pth_declarations.o
224   $(CPP) $(PREPROCESSOR_FLAG) $(PCPPFLAGS) communication
          /mpi_transmission.f90 -o communication/mpi_transmission.cpp.
          f90
225   $(FC) -c $(FFLAGS) communication/mpi_transmission.cpp.f90 -o
          communication/mpi_transmission.o
226 else
227 communication/mpi_transmission.o : communication/
        message_declarations.o communication/mpi_transmission.f90
228   $(CPP) $(PREPROCESSOR_FLAG) $(PCPPFLAGS) communication
          /mpi_transmission.f90 -o communication/mpi_transmission.cpp.
          f90
229   $(FC) -c $(FFLAGS) communication/mpi_transmission.cpp.f90 -o
          communication/mpi_transmission.o
230
231 endif
232 #
233 #test: test.o global_declarations.o data/boundary.o data/slopelim.o
        data/node_info.o tree/amr_node.o data/syncgrids.o
234 # $(FC) $(FFLAGS) test.o global_declarations.o data/boundary.o data
        /slopelim.o data/node_info.o tree/amr_node.o data/syncgrids.o -
        o astrobear
235 #
236 #scrambler.o : global_declarations.o data/boundary.o data/slopelim.o
        data/data_declarations.o tree/tree_declarations.o module_control.o
         physics/physics_control.o io/io_control.o scrambler.f90
237 # $(FC) -c $(FFLAGS) scrambler.f90 -o scrambler.o
238 #
```

```
239 #test.o : global_declarations.o data/boundary.o data/slopelim.o data/
        data_declarations.o tree/tree_declarations.o module_control.o data
        /data_info_ops.o test.f90
240 # $(FC) -c $(FFLAGS) test.f90 -o test.o
241 #
242 #module_control.o : module_control.f90 hyperbolic/
        hyperbolic_declarations.o physics/physics_declarations.o
        global_declarations.o data/data_declarations.o $(
        PROBLEM_OBJ_FILES)
243 # $(FC) -c $(FFLAGS) module_control.f90 -o module_control.o
244 #
245 #hyperbolic/hyperbolic_control.o : hyperbolic/hyperbolic_control.f90
        hyperbolic/hyperbolic_declarations.o hyperbolic/sweep/
        sweep_scheme.o data/data_declarations.o
246 # $(FC) -c $(FFLAGS) hyperbolic/hyperbolic_control.f90 -o
        hyperbolic/hyperbolic_control.o
247 #
248 if PTHREAD2
249 hyperbolic/sweep/sweep_scheme.o : hyperbolic/sweep/sweep_scheme.
        f90 global_declarations.o data/data_declarations.o hyperbolic/
        hyperbolic_declarations.o module_control.o pth/pth_declarations.o
250   $(CPP) $(PREPROCESSOR_FLAG) $(HCPPFLAGS) $(PCPPFLAGS
          ) hyperbolic/sweep/sweep_scheme.f90 -o hyperbolic/sweep/
          sweep_scheme.cpp.f90
251   $(FC) -c $(FFLAGS) hyperbolic/sweep/sweep_scheme.cpp.f90 -o
          hyperbolic/sweep/sweep_scheme.o
252 else
253 hyperbolic/sweep/sweep_scheme.o : hyperbolic/sweep/sweep_scheme.
        f90 global_declarations.o data/data_declarations.o hyperbolic/
        hyperbolic_declarations.o module_control.o
254   $(CPP) $(PREPROCESSOR_FLAG) $(HCPPFLAGS) $(PCPPFLAGS
          ) hyperbolic/sweep/sweep_scheme.f90 -o hyperbolic/sweep/
          sweep_scheme.cpp.f90
255   $(FC) -c $(FFLAGS) hyperbolic/sweep/sweep_scheme.cpp.f90 -o
          hyperbolic/sweep/sweep_scheme.o
256 endif
257 hyperbolic/MUSCL/MUSCL_scheme.o : hyperbolic/MUSCL/
        MUSCL_scheme.f90 global_declarations.o data/data_declarations.o
        hyperbolic/hyperbolic_declarations.o module_control.o#
258   $(CPP) $(PREPROCESSOR_FLAG) $(HCPPFLAGS) $(PCPPFLAGS
          ) hyperbolic/MUSCL/MUSCL_scheme.f90 -o hyperbolic/
          MUSCL/MUSCL_scheme.cpp.f90
259   $(FC) -c $(FFLAGS) hyperbolic/MUSCL/MUSCL_scheme.cpp.f90 -
          o hyperbolic/MUSCL/MUSCL_scheme.o
260
261 if PTHREAD2
262 hyperbolic/sweep/stencil_control.o : hyperbolic/sweep/stencil_control.
        f90 global_declarations.o data/data_declarations.o hyperbolic/
        hyperbolic_declarations.o pth/pth_declarations.o
```

```
263   $(CPP) $(PREPROCESSOR_FLAG) $(HCPPFLAGS) $(PCPPFLAGS
          ) hyperbolic/sweep/stencil_control.f90 -o hyperbolic/sweep/
          stencil_control.cpp.f90
264   $(FC) -c $(FFLAGS) hyperbolic/sweep/stencil_control.cpp.f90 -o
          hyperbolic/sweep/stencil_control.o
265 else
266 hyperbolic/sweep/stencil_control.o : hyperbolic/sweep/stencil_control.
      f90 global_declarations.o data/data_declarations.o hyperbolic/
      hyperbolic_declarations.o
267   $(CPP) $(PREPROCESSOR_FLAG) $(HCPPFLAGS) $(PCPPFLAGS
          ) hyperbolic/sweep/stencil_control.f90 -o hyperbolic/sweep/
          stencil_control.cpp.f90
268   $(FC) -c $(FFLAGS) hyperbolic/sweep/stencil_control.cpp.f90 -o
          hyperbolic/sweep/stencil_control.o
269 endif
270
271 #
272 #elliptic/elliptic_control.o : elliptic/elliptic_control.f90
273 # $(FC) -c $(HDFINC) $(FFLAGS) $(EFLAGS) elliptic/
      elliptic_control.f90 -o elliptic/elliptic_control.o
274
275 pth/pth_control.o : pth/pth_control.f90 hyperbolic/hyperbolic_control.o
276   $(FC) -c $(FFLAGS) pth/pth_control.f90 -o pth/pth_control.o
277
278 source/source_control.o : source/cooling.o io/chombo_declarations.o
      source/source_control.f90
279   $(FC) -c $(HDFINC) $(FFLAGS) source/source_control.f90 -o
          source/source_control.o
280
281
282 elliptic/self_gravity/poisson.o : elliptic/self_gravity/poisson.f90
283   $(FC) -c $(HDFINC) $(FFLAGS) -I$(FFTW_PATH)/include elliptic
          /self_gravity/poisson.f90 -o elliptic/self_gravity/poisson.o
284
285 Problem/problem.o : Problem/problem.f90
286   $(FC) -c $(HDFINC) $(FFLAGS) -I$(FFTW_PATH)/include
          Problem/problem.f90 -o Problem/problem.o
287
288 objects/outflows.o : objects/outflows.f90 io/chombo_declarations.o io/
      hdf5_declarations.o
289   $(FC) -c $(HDFINC) $(FFLAGS) objects/outflows.f90 -o objects/
          outflows.o
290
291 physics/PFFT.o : physics/PFFT.f90
292   $(FC) -c $(HDFINC) $(FFLAGS) -I$(FFTW_PATH)/include
          physics/PFFT.f90 -o physics/PFFT.o
293
294 source/pointgravity.o : source/pointgravity.f90 io/chombo_declarations.
      o io/hdf5_declarations.o
```

```
295   $(FC) -c $(HDFINC) $(FFLAGS) source/pointgravity.f90 -o source
          /pointgravity.o
296
297 source/cooling.o : source/cooling.f90 io/chombo_declarations.o io/
        hdf5_declarations.o
298   $(FC) -c $(HDFINC) $(FFLAGS) source/cooling.f90 -o source/
          cooling.o
299
300 particle/particle_control.o : particle/particle_declarations.o particle/
        particle_level_ops.o particle/particle_comms.o particle/
        particle_advance.o particle/particle_info_ops.o particle/
        particle_control.f90
301   $(FC) -c $(HDFINC) $(FFLAGS) particle/particle_control.f90 -o
          particle/particle_control.o
302
303 particle/particle_level_ops.o : particle/particle_declarations.o particle/
        particle_info_ops.o particle/particle_level_ops.f90
304   $(FC) -c $(HDFINC) $(FFLAGS) particle/particle_level_ops.f90 -o
          particle/particle_level_ops.o
305
306 particle/particle_declarations.o : physics/physics_declarations.o
        global_declarations.o io/chombo_declarations.o io/
        hdf5_declarations.o source/pointgravity.o particle/
        particle_declarations.f90
307   $(FC) -c $(HDFINC) $(FFLAGS) particle/particle_declarations.f90 -
          o particle/particle_declarations.o
308
309 io/io_control.o : io/io_chombo.o io/io_declarations.o io/io_control.f90
310   $(FC) -c $(HDFINC) $(FFLAGS) io/io_control.f90 -o io/io_control.
          o
311
312 io/io_chombo.o : global_declarations.o data/data_level_ops.o physics/
        physics_declarations.o distribution/distribution_control.o tree/
        tree_level_comms.o io/hdf5_declarations.o io/io_comms.o io/
        chombo_declarations.o io/io_chombo.f90
313   $(FC) -c $(HDFINC) $(FFLAGS) io/io_chombo.f90 -o io/
          io_chombo.o
314
315 io/io_comms.o : io/io_comms.f90 tree/tree_declarations.o data/
        data_declarations.o global_declarations.o io/chombo_declarations.o
        communication/communication_declarations.o communication/
        message_declarations.o io/io_parsing.o communication/
        mpi_packing.o
316   $(FC) -c $(HDFINC) $(FFLAGS) io/io_comms.f90 -o io/io_comms.
          o
317
318 io/io_parsing.o : io/io_parsing.f90 tree/tree_declarations.o data/
        data_declarations.o global_declarations.o io/chombo_declarations.o
        communication/mpi_packing.o
```

```
319   $(FC) -c $(HDFINC) $(FFLAGS) io/io_parsing.f90 -o io/io_parsing.
          o
320
321 io/chombo_declarations.o : global_declarations.o io/hdf5_declarations.o
        io/io_declarations.o io/chombo_declarations.f90
322   $(FC) -c $(HDFINC) $(FFLAGS) io/chombo_declarations.f90 -o io
          /chombo_declarations.o
323
324 io/hdf5_declarations.o : global_declarations.o io/hdf5_declarations.f90
325   $(FC) -c $(HDFINC) $(FFLAGS) io/hdf5_declarations.f90 -o io/
          hdf5_declarations.o
326
327 distribution/distribution_control.o : distribution/hilbert.o data/
        data_info_ops.o distribution/distribution_control.f90
328   $(FC) -c $(HDFINC) $(FFLAGS2) distribution/distribution_control.
          f90 -o distribution/distribution_control.o
329
330 layouts/layout_io.o : layouts/layout_io.f90 layouts/layout_declarations.
        f90
331   $(FC) -c $(HDFINC) $(FFLAGS) layouts/layout_io.f90 -o layouts/
          layout_io.o
332
333 #io/io_declarations.o : global_declarations.o io/io_declarations.f90
334 # $(FC) -c $(FFLAGS) io/io_declarations.f90 -o io/io_declarations.o
335 #
336 #data/data_declarations.o : global_declarations.o data/boundary.o data
        /data_declarations.f90
337 # $(FC) -c $(FFLAGS) data/data_declarations.f90 -o data/
        data_declarations.o
338 #
339 #data/boundary.o : global_declarations.o data/boundary.f90
340 # $(FC) -c $(FFLAGS) data/boundary.f90 -o data/boundary.o
341 #
342 #data/slopelim.o : global_declarations.o data/slopelim.f90
343 # $(FC) -c $(FFLAGS) data/slopelim.f90 -o data/slopelim.o
344 #
345 #data/data_info_ops.o : data/data_declarations.o data/data_info_ops.
        f90
346 # $(FC) -c $(FFLAGS) data/data_info_ops.f90 -o data/data_info_ops.
        o
347 #
348 #tree/tree_level_ops.o : tree/tree_level_ops.f90 tree/tree_node_ops.o
        tree/tree_declarations.o $(PROBLEM_OBJ_FILES)
349 # $(FC) -c $(FFLAGS) tree/tree_level_ops.f90 -o tree/tree_level_ops.
        o
350 #
351 #data/data_level_ops.o : data/data_level_ops.f90 amr/scheduling.o data
        /data_info_ops.o hyperbolic/hyperbolic_control.o data/
        data_declarations.o tree/tree_declarations.o module_control.o $(
        PROBLEM_OBJ_FILES)
```

```
352 # $(FC) -c $(FFLAGS) data/data_level_ops.f90 -o data/
        data_level_ops.o
353 #
354 ##amr/time_step.o : amr/time_step.f90 elliptic_control.o
355 ## $(FC) -c $(FFLAGS) amr/time_step.f90 -o amr/time_step.o
356 #
357 ##physics/physics_control.o : physics/physics_declarations.o physics/
        physics_control.f90
358 ## $(FC) -c $(FFLAGS) physics/physics_control.f90 -o physics/
        physics_control.o
359 #
360 #tree/tree_level_comms.o : tree/tree_declarations.o communication/
        mpi_packing.o tree/tree_parsing.o tree/tree_level_comms.f90
361 # $(FC) -c $(FFLAGS) tree/tree_level_comms.f90 -o tree/
        tree_level_comms.o
362 #
363 #tree/tree_parsing.o : tree/tree_declarations.o tree/tree_node_ops.o
        communication/mpi_packing.o tree/tree_parsing.f90
364 # $(FC) -c $(FFLAGS) tree/tree_parsing.f90 -o tree/tree_parsing.o
365 #
366 #data/data_level_comms.o : tree/tree_declarations.o data/
        data_declarations.o data/data_level_comms.f90
367 # $(FC) -c $(FFLAGS) data/data_level_comms.f90 -o data/
        data_level_comms.o
368 #
369 #communication/communication_declarations.o : global_declarations.o
        communication/message_declarations.o communication/
        communication_declarations.f90
370 # $(FC) -c $(FFLAGS) communication/communication_declarations.
        f90 -o communication/communication_declarations.o
371 #
372 #communication/mpi_packing.o : global_declarations.o tree/
        tree_declarations.o communication/message_declarations.o
        communication/mpi_transmission.o communication/mpi_packing.
        f90
373 # $(FC) -c $(FFLAGS) communication/mpi_packing.f90 -o
        communication/mpi_packing.o
374 #
375 #communication/mpi_transmission.o : communication/
        message_declarations.o communication/mpi_transmission.f90
376 # $(FC) -c $(FFLAGS) communication/mpi_transmission.f90 -o
        communication/mpi_transmission.o
377 #
378 #communication/message_declarations.o : communication/
        message_declarations.f90
379 # $(FC) -c $(FFLAGS) communication/message_declarations.f90 -o
        communication/message_declarations.o
380 #
381 #tree/tree_declarations.o : global_declarations.o data/data_declarations.
        o tree/tree_declarations.f90
```

```
382 # $(FC) -c $(FFLAGS) tree/tree_declarations.f90 -o tree/
        tree_declarations.o
383 #
384 #tree/distribution/hilbert.o : tree/distribution/hilbert.c
385 # $(CC) -c $(CFLAGS) tree/distribution/hilbert.c -o tree/
        distribution/hilbert.o
386 #
387 ##physics/physics_declarations.o : physics/physics_declarations.f90
388 ## $(FC) -c $(FFLAGS) physics/physics_declarations.f90 -o physics/
        physics_declarations.o
389 #
390 #
391 #FieldLoopAdvection/field_loop_advection.o : FieldLoopAdvection/
        field_loop_advection.f90 global_declarations.o data/
        data_declarations.o
392 # $(FC) -c $(FFLAGS) FieldLoopAdvection/field_loop_advection.f90
        -o FieldLoopAdvection/field_loop_advection.o
393 #
394 elliptic/extended_hypre_wrapper.o : elliptic/extended_hypre_wrapper.c
395   $(CC) -c $(CFLAGS) $(HYPREINC_C) elliptic/
          extended_hypre_wrapper.c -o elliptic/extended_hypre_wrapper.o
396
397 distribution/hilbert.o : distribution/hilbert.c
398     $(CC) -c $(CFLAGS) distribution/hilbert.c -o distribution/hilbert
            .o
399
400 if PTHREAD
401 threads/fpthread_lib.o : threads/fpthread_lib.c
402   $(CC) -c $(CFLAGS) threads/fpthread_lib.c -pthread -o threads/
          fpthread_lib.o
403
404 endif
405
406 if PTHREAD2
407 pth/fpth_lib.o : pth/fpth_lib.c
408   $(CC) -c $(CFLAGS) pth/fpth_lib.c -I$(PTH_PATH)/include -o
          pth/fpth_lib.o
409
410 endif
411
412 #Template/template.o : Template/template.f90 objects/winds.o
        objects/clumps.o
413 # $(FC) -c $(FFLAGS) Template/template.f90 -o Template/
        template.o
414
415
416 %.o : %.f90
417   $(FC) -c $(FFLAGS) $< -o $@
418
419 %.o : %.c
```

```
420   $(CC) -c $(CFLAGS) $< -o $@
421
422 ###clean:
423 ### rm -f $(OBJ_FILES) $(MOD_FILES) $(PROBLEM_OBJ_FILES)
        $(PROBLEM_MOD_FILES) $(EXECUTABLE)
424 ### rm -f $(HYPRE_OBJ_FILES) $(POST_ELLIPTIC_OBJ_FILES) $
      (PTHREAD_OBJ_FILES) $(POST_PTHREADS_OBJ_FILES)
425 ### rm -f *.lst
426 ### rm -f *.cpp.f90 */*.cpp.f90 */*/*.cpp.f90
427
428 ##======= Hypre source files ============
429 HYPRE_SRC_FILES= \
430       elliptic/extended_hypre_wrapper.c \
431       elliptic/self_gravity/multipole.f90 \
432       elliptic/self_gravity/poisson.f90 \
433       elliptic/diffusion/diffusionbc.f90 \
434       elliptic/diffusion/diffusion.f90 \
435       elliptic/radtransfer/radtransfer.f90 \
436       elliptic/elliptic_control.f90
437
438 ##====== Explicit source files ==========
439 EXPLICIT_SRC_FILES = \
440         explicit/resistive.f90 \
441       explicit/conductive.f90 \
442       explicit/bviscosity.f90 \
443       explicit/explicit_control.f90
444
445 ##==== PTHREAD is enabled or not
446  PTHREAD_SRC_FILES = \
447         threads/fpthread_lib.c \
448         threads/thread_declarations.f90 \
449         threads/thread_control.f90
450
451 ##==== PTHREAD2 is enabled or not
452  PTHREAD2_SRC_FILES = \
453         pth/fpth_lib.c \
454         pth/pth_declarations.f90 \
455         pth/pth_control.f90
456
457 POST_ELLIPTIC_SRC_FILES = \
458               processing/processing_control.f90
459
460 POST_PTHREADS_SRC_FILES = \
461               amr/time_step.f90 \
462               io/io_declarations.f90 \
463               io/hdf5_declarations.f90 \
464               io/chombo_declarations.f90 \
465               io/io_parsing.f90 \
466               io/io_comms.f90 \
467               io/io_chombo.f90 \
```

```
468                 io/io_okc.f90 \
469                 io/io_control.f90 \
470               amr/amr_control.f90 \
471                 scrambler.f90
472
473 PROBLEM_SRC_FILES = \
474                 */*
475 ##MOD_FILES = *.mod \
476 ## amr/*.mod \
477 ## physics/*.mod \
478 ## data/*.mod \
479 ## *.mod \
480 ## hyperbolic/*.mod \
481 ## io/*.mod \
482 ## communication/*.mod \
483 ## source/*.mod
484
485 METHOD_INCLUDE_FILES = \
486           hyperbolic/i_dependencies.f90 \
487           hyperbolic/riemann_solvers.f90
488
489 SRC_FILES = global_declarations.f90 \
490                 processing/splines.f90 \
491                 objects/profiles.f90 \
492                 communication/message_declarations.f90 \
493                 communication/communication_declarations.f90 \
494                 data/boundary.f90 \
495                 data/slopelim.f90 \
496                 data/data_declarations.f90 \
497                 tree/tree_declarations.f90 \
498                 physics/physics_declarations.f90 \
499                 elliptic/elliptic_declarations.f90 \
500                 explicit/explicit_declarations.f90 \
501                 hyperbolic/hyperbolic_declarations.f90 \
502                 physics/specfun.f90 \
503                 physics/EOS.f90 \
504                 physics/EOS.f90 \
505                 physics/abundances.f90 \
506                 physics/common_functions.f90 \
507                 physics/BE_module.f90 \
508                 physics/Bondi.f90 \
509                 timing.f90 \
510                 amr/scheduling.f90 \
511                 hyperbolic/riemann_solvers.f90 \
512                 data/data_info_ops.f90 \
513                 objects/object_declarations.f90 \
514                 objects/tables.f90 \
515                 physics/opacity.f90 \
516                 source/chemistry.f90 \
517                 source/AnalyticCooling.f90 \
```

```
518                 source/IICooling.f90 \
519                 source/DMCooling.f90 \
520                 source/ZCooling.f90 \
521                 source/cooling.f90 \
522                 source/cylindrical.f90 \
523                 source/rotating.f90 \
524                 source/uniformgravity.f90 \
525                 io/io_declarations.f90 \
526                 io/hdf5_declarations.f90 \
527                 io/chombo_declarations.f90 \
528                 source/pointgravity.f90 \
529                 objects/outflows.f90 \
530                 source/selfgravity.f90 \
531                 source/source_control.f90 \
532         $(EXPLICIT_SRC_FILES) \
533                 particle/particle_declarations.f90 \
534                 particle/particle_info_ops.f90 \
535                 particle/particle_level_ops.f90 \
536                 particle/particle_advance.f90 \
537                 particle/particle_comms.f90 \
538                 particle/particle_control.f90 \
539                 objects/shapes.f90 \
540                 processing/emissions.f90 \
541                 processing/fields.f90 \
542               processing/processing_declarations.f90 \
543                 objects/profile_functions.f90 \
544                 objects/interfaces.f90 \
545                 objects/perturbations.f90 \
546                 objects/vectorperturbations.f90 \
547                 objects/ambients.f90 \
548                 objects/clumps.f90 \
549                 objects/disks.f90 \
550                 objects/winds.f90 \
551                 objects/uniformregions.f90 \
552                 objects/splitregions.f90 \
553                 objects/collidingflows.f90 \
554                 physics/physics_control.f90 \
555                 io/io_bov.f90 \
556                 io/io_curve.f90 \
557                 io/io_ppm.f90 \
558                 layouts/layout_declarations.f90 \
559                 layouts/layout_io.f90 \
560                 layouts/layout_comms.f90 \
561                 physics/PFFT.f90 \
562                 layouts/layout_control.f90 \
563                 processing/images.f90 \
564                 processing/histograms.f90 \
565                 processing/spectra.f90 \
566                 processing/pdfs.f90 \
567                 processing/cameras.f90 \
```

```
568                 processing/movies.f90 \
569                 processing/projections.f90 \
570                 processing/totals.f90 \
571                 objects/refinements.f90 \
572                 objects/tests.f90 \
573                 objects/object_control.f90 \
574                 $(PROBLEM_SRC_FILES) \
575                 module_control.f90 \
576                 hyperbolic/sweep/stencil_declarations.f90 \
577                 hyperbolic/sweep/sweep_declarations.f90 \
578                 hyperbolic/sweep/stencil_control.f90 \
579                 hyperbolic/sweep/sweep_scheme.f90 \
580                 hyperbolic/MUSCL/MUSCL_scheme.f90 \
581                 hyperbolic/hyperbolic_control.f90 \
582                 distribution/distribution_declarations.f90 \
583                 distribution/hilbert.c \
584                 distribution/distribution_control.f90 \
585                 communication/mpi_transmission.f90 \
586                 communication/mpi_packing.f90 \
587                 communication/communication_control.f90 \
588                 tree/tree_node_ops.f90 \
589                 tree/tree_parsing.f90 \
590                 tree/tree_level_comms.f90 \
591                 data/data_parsing.f90 \
592                 data/data_level_comms.f90 \
593                 data/data_level_ops.f90 \
594                 tree/tree_level_ops.f90
595 HEADFILES=\
596     elliptic/extended_hypre_wrapper.h
597
598 dist_astrobear_SOURCES = $(SRC_FILES) $(HYPRE_SRC_FILES) $(
        POST_ELLIPTIC_SRC_FILES) $(PTHREAD_SRC_FILES) $(
        PTHREAD2_SRC_FILES) $(POST_PTHREADS_SRC_FILES) $(
        HEADFILES)
```

4.2 Generate the Configure File

To generate the configure file we still need these commands

```
1 $ autoreconf --install
2 $ ./configure
3 $ make
4 $ make distcheck
```