CASL: Some tips for Mac users (MacOS X Leopard)

August 19, 2008

1 Basic assumptions

It is assumed you have already gcc and gfortan (you'll need to install Xcode beforehand). That should be more or less all you need.

2 Graphics package

I did not uncounter difficult problems to install the graphics package. Just make sure you have the subdirectory gp with all the .h and .f files. Just modify runme, see my own runme below to include a mac option.

```
#! /bin/csh

if ($#argv == 1) then
   set MACHINE = $1
else
   echo 'Machine type (PC/DEC/CRAY/SG/SUN/MAC)?'
   set MACHINE=$<
   if (($MACHINE != PC) && ($MACHINE != DEC) && ($MACHINE != CRAY) &&
       ($MACHINE != SG) && ($MACHINE != SUN) && ($MACHINE != MAC)) then
      echo Wrong type. Only PC/DEC/CRAY/SG/SUN/MAC allowed
      exit(1)
   endif
endif

if ($MACHINE == CRAY) then
   alias f77 cf77
   set CFLAGS=""
   set FFLAGS=""
else if ($MACHINE == PC) then
   alias f77 f77
   alias cc gcc
   set CFLAGS="-g"
   set FFLAGS="-g -w"
else if ($MACHINE == MAC) then
   echo It is assumed you have the GNU compilers gcc/gfortran
   alias f77 gfortran
   alias cc gcc
   set CFLAGS="-g"
   set FFLAGS="-g -w"
```

```
else
   set CFLAGS="-g"
   set FFLAGS="-g"
endif

set HEDFIL = " gaxcom.h gaxmrg.h gdefax.h gdefpm.h gdeftx.h gpcons.h gpgmap.h gpshad.h
gpsmap.h gpzmap.h gpzstk.h gtxcom.h gpconsc.h "
set SRCFIL="gquick.f gplot.f gzoom.f gcoord.f gstnum.f gtext.f gaxes.f"
set OBJFIL="gquick.o gplot.o gzoom.o gcoord.o gstnum.o gtext.o gaxes.o"

echo Making glib.o
f77 -c $FFLAGS $SRCFIL
ar crv glib.o $OBJFIL
if (($MACHINE != CRAY)||($MACHINE != VPX)) ranlib glib.o
/bin/rm $OBJFIL

echo Making gpolar.o, gdump.o
f77 -c $FFLAGS gpolar.f
cc -c $CFLAGS -DON_${MACHINE} gdump.c
echo Making gpps.o, gpX.o
f77 -c $FFLAGS gpps.f
f77 -c $FFLAGS gpxf.f
cc -c $CFLAGS -DON_${MACHINE} -DBUFFERED gpxc.c
ar crv gpX.o gpxf.o gpxc.o
if (($MACHINE != CRAY)||($MACHINE != VPX)) ranlib gpX.o

echo Making gpX1.o
ar crv gpX1.o gpxf.o gpxc.o
if (($MACHINE != CRAY)||($MACHINE != VPX)) ranlib gpX1.o
/bin/rm gpxf.o

echo Done
```

When running the script a few error message are displayed. It seems you can ignore them. Same thing when compiling.

3 Update the get????? scripts from cs/bin-sh/

3.1 get_machine

In get_machine, add yours, example add something like this just before the final endif and replace jean-reinauds-macbook by the name of your mac, see below

```
else if ( $host == jean-reinauds-macbook ) then

   echo 'mac'
```

3.2 get_compiler

Modify then get_compiler. Find below an example

```
#!/bin/csh
```

```
# Assigns the compiler given the machine type; modify as needed.

set mach = `~/cs/bin-sh/get_machine`

if ( $mach == intel ) then
   echo 'ifort -O3'
else if ( $mach == opteron ) then
   echo 'f77 -O3 -I.'
else if ( $mach == up2000 ) then
   echo 'fort -fast'
else if ( $mach == digital ) then
   echo 'f90 -O3'
else if ( $mach == cray ) then
   echo 'cf77'
else if ( $mach == cluster ) then
   echo 'mpif90'
else if ( $mach == laptop ) then
   echo 'ifort -O3'
else if ( $mach == other ) then
   echo 'f77 -O3'
else if ( $mach == mac ) then
   echo 'gfortran -O3'
endif
```

3.3 get_home

Add your own directory, example

```
#!/bin/csh

# Finds the home disk given the machine type (uses get_machine for this)

# This is needed, for example, when trying to execute pre-compiled routines
# with different processors.  The processors typically differ from machine
# to machine.

# Change, or add more options if required.

set mach = `~/cs/bin-sh/get_machine`

if ( $mach == intel ) then

   echo '/disk3'

else if ( $mach == opteron ) then

   echo '/disk3'

else if ( $mach == up2000 ) then

   echo '/disk10'
```

```
else if ( $mach == digital ) then
   echo '/disk6'
else if ( $mach == cluster ) then
   echo '/scratch'
else if ( $mach == laptop ) then
   echo '/home'
else if ( $mach == other ) then
   echo '/scratch'
else if ( $mach == mac ) then
  echo '/Users'
endif
exit
```

3.4 get_lib & get_nag

First install fftw 2.1.5 (freely available from hhttp://www.fftw.org). The install procedure works just as on Linux. Then add the mac in get_lib & get_nag:

```
#!/bin/csh
# Finds the FFT library link given the machine type (uses get_machine for this)
# Add more options if required.
set mach = `~/cs/bin-sh/get_machine`
if ( $mach == intel ) then
# FFTW:
   echo '-lrfftw -lfftw'
else if ( $mach == opteron ) then
# FFTW:
   echo '-lrfftw -lfftw'
else if ( $mach == up2000 ) then
# Compaq eXtended Mathematical Library:
   echo '-lcxml'
```

```
else if ( $mach == digital ) then

# Digital eXtended Mathematical Library:
   echo '-ldxml'

else if ( $mach == cluster ) then

# FFTW:
   echo '-lrfftw -lfftw'

else if ( $mach == laptop ) then

# FFTW:
   echo '-lrfftw -lfftw'

else if ( $mach == other ) then

# FFTW:
   echo '-lrfftw -lfftw'

else if ( $mach == mac ) then

# FFTW:
   echo '-lrfftw -lfftw'

endif

#!/bin/csh

# Locate NAG or NAG-equivalent routines for use in compiling
#=============================================================================

set mach = `~/cs/bin-sh/get_machine`

if ( ( $mach == laptop ) || ( $mach == intel ) || ( $mach == opteron ) || ( $mach == mac ) )
#   use local routines:
   set local_home = `~/cs/bin-sh/get_home`
   set lib_path = $local_home/$user/Libdir
   echo $lib_path/double_fft.f $lib_path/double_tran.f $lib_path/f02agf.f
else
#   use NAG library:
   echo '-lnag'
endif
```

3.5 other scripts

Other (old) scripts might need updates. Newer scripts should work straightforwardly with the get_???? updated.