

GeoClaw group outline

Monday morning:

- Intro to using Clawpack:
 setting parameters, plotting results
- Intro to Makefiles, Python, IPython
- Clawpack examples in [\\$CLAW/apps/advection](#)
- GeoClaw examples in [\\$CLAW/apps/tsunami](#)
- Acquiring topography data

Tuesday morning:

- Earthquake sources, Okada model

Wednesday morning:

- More about algorithms, adaptive refinement

GeoClaw group outline

Afternoons:

Today:

- Learn to run GeoClaw.
- Download topo files.
- Start to form groups.

Tuesday — Thursday:

- Mostly work in groups.
- Tutorials as needed.

Compiling, running, plotting

Applications directories contain a **Makefile**.

\$ make help **For list of options**

\$ make .data **Uses `setrun.py` to make Fortran data**

\$ make .exe **Compiles Fortran codes**

\$ make .output **Runs code, produces `_output`**

\$ make .plots **Plots results, produces `_plots`**

\$ make .htmls **Produces html versions of source files**

Application Makefile

Documentation: www.clawpack.org/users/makefiles.html

Several variables are set, e.g.

where to find `setrun` function and where to put output:

```
CLAW_setrun_file = setrun.py
CLAW_OUTDIR = _output
```

where to find `setplot` function and where to put plots:

```
CLAW_setplot_file = setplot.py
CLAW_PLOTDIR = _plots
```

Usually these do not need to be changed.

Application Makefile (cont.)

List of local Fortran files:

```
CLAW_SOURCES = \  
  driver.f \  
  qinit.f \  
  rp1.f \  
  setprob.f
```

List of library files:

```
# Clawpack library to be used:  
CLAW_LIB = $(CLAW)/clawpack/ld/lib  
  
CLAW_LIBSOURCES = \  
  $(CLAW_LIB)/clawlez.f \  
  $(CLAW_LIB)/bc1.f \  
  etc.
```

Application Makefile (cont.)

Ends with...

```
# Include Makefile containing standard
# definitions and make options:
CLAWMAKE = $(CLAW)/util/Makefile.common
include $(CLAWMAKE)
```

The file `$(CLAW)/util/Makefile.common` contains rules for various targets.

For possible targets, type

```
$ make help
```

Setting runtime parameters

The file `setrun.py` contains a function `setrun` that returns an object `rundata` of class `ClawRunData`.

Never need to write from scratch...
Modify an existing example!

Don't need to know much if anything about Python!

Lots of comments in the sample versions.

Documentation: www.clawpack.org/doc/setrun.html

Copying and modifying an example

Find an example similar to the one you want to create and copy the directory, e.g.

```
$ cd $CLAW/apps/acoustics/1d
$ cp -r example2 $CLAW/myclaw/newexample
```

Do not need to put in the `$CLAW/myclaw` directory, should work as long as environment variable set properly.

Python plotting tools

Directory `_output` contains files `fort.t000N`, `fort.q000N` of data at **frame N** (N'th output time).

`fort.t000N`: Information about this time,
`fort.q000N`: Solution on all grids at this time

There may be many grids at each output time.

Python tools provide a way to specify what plots to produce for each frame:

- One or more **figures**,
- Each figure has one or more **axes**,
- Each axes has one or more **items**,
(Curve, contour, pcolor, etc.)

setplot function for specifying plots

The file `setplot.py` contains a function `setplot`
Takes an object `plotdata` of class `ClawPlotData`,
Sets various attributes, and returns the object.

Documentation: www.clawpack.org/users/setplot.html

Example: 1 figure with 1 axes showing 1 item:

```
def setplot(plotdata):  
    plotfigure = plotdata.new_plotfigure(name, num)  
    plotaxes = plotfigure.new_plotaxes(title)  
    plotitem = plotaxes.new_plotitem(plot_type)  
    # set attributes of these objects  
    return plotdata
```

setplot function for specifying plots

Example: plot first component of q as blue curve, red circles.

```
plotfigure = plotdata.new_plotfigure('Q', 1)
plotaxes = plotfigure.new_plotaxes('axes1')

plotitem = plotaxes.new_plotitem('1d_plot')
plotitem.plotvar = 0 # Python indexing!
plotitem.plotstyle = '-'
plotitem.color = 'b' # or [0,0,1] or '#0000ff'

plotitem = plotaxes.new_plotitem('1d_plot')
# plotitem now points to a new object!
plotitem.plotvar = 0
plotitem.plotstyle = 'ro'
```

Plotting examples and documentation

General plotting information:

www.clawpack.org/users/plotting.html

Use of setplot, possible attributes:

www.clawpack.org/users/setplot.html

Examples:

1d: www.clawpack.org/users/plotexamples.html

2d: www.clawpack.org/users/plotexamples2d.html

FAQ: www.clawpack.org/users/plotting_faq.html

Gallery of applications:

www.clawpack.org/users/apps.html

Plotting options

Create a set of webpages showing all plots:

```
$ make .plots
```

Disadvantages:

- May take a while to plot all frames
- Can't zoom in dynamically or explore data

View plots interactively:

```
$ ipyclaw # alias defined in setenv.bash
```

```
In[1]: ip = Iplotclaw()
```

```
In[2]: ip.plotloop()
```

```
PLOTCLAW>> ?
```

```
PLOTCLAW>> q
```

```
In[3]: Quit
```