Python, Clawpack, PyClaw, and PetClaw (using PETSc)

Randall J. LeVeque
Kyle Mandli

Applied Mathematics
University of Washington

Conservation Laws Package

www.clawpack.org

Supported in part by NSF, ONR, NIH, DOE,
Founders Term Professorship

CLAWPACK — Conservation Laws Package

High-resolution finite volume methods for hyperbolic problems

- Open source, 1d, 2d, 3d **www.clawpack.org**
- Originally f77 with Matlab graphics.
- Moving to f95 with Python.
- Adaptive mesh refinement (AMRClaw – Marsha Berger).
- OpenMP and MPI (under development).

User supplies:

- Riemann solver, splitting data into waves and speeds
- Boundary condition routine to extend data to ghost cells
- Initial conditions — `qinit.f`
- Source terms — `src1.f`

Recent work on Clawpack: www.clawpack.org

- PyClaw: Python interface, graphics (Kyle Mandli)
- EagleClaw: Easy Access Graphical Laboratory for Exploring Conservation Laws (web interface)
- GeoClaw: Geophysical flows (David George, Marsha Berger, Kyle Mandli)
- SharpClaw: Semi-discrete High Accuracy Runge-Kutta Package (David Ketcheson)
  - High order Weighted Essentially Non-Oscillatory (WENO) methods
  - SSP Runge-Kutta methods
  - Extension to hyperbolic problems not in conservation form
  - Clawpack framework and Riemann solvers.
- ChomboClaw: Interface to CHOMBO (Donna Calhoun)
  - CHOMBO: AMR in C++ with MPI interface
    P. Collela et. al., Lawrence Berkeley Lab

Setting runtime parameters

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

```
$ make .data
```

converts into file of input parameters read by Fortran.

Advantages:

- Easier to document,
- Parameters can be set in any order,
- Can use conditionals, loops, `linspace`, ...
- Easy to script to perform a series of runs.
- Can more easily add new parameters with default values, without breaking previous apps.

Python plotting tools

Fortran code outputs solution in standard format.

Originally used Matlab for plotting.

Currently using matplotlib module for 1d and 2d plots

3d under development with VisIt (LLNL) and Mayavi2 (EPD).

Advantages of Python:

- Open source and free (as is Clawpack)
- Excellent plotting tools for 3d
- More powerful general language for scripting regression tests, apps gallery, web interface using cgi-scripts, etc.
- Virtualization

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 speciying 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 speciying 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’
```

Virtualization

Clawpack requires:

- Unix/Linux
- gfortran (and OpenMP, MPI)
- Python (preferably IPython)
- Plotting modules
- Sphinx for documentation

Virtualization

Clawpack requires:

- Unix/Linux
- gfortran (and OpenMP, MPI)
- Python (preferably IPython)
- Plotting modules
- Sphinx for documentation

VirtualClaw: Simple way to provide complete OS and software.

VM image for VirtualBox, runs on Linux / Windows / Mac.

Download: **www.clawpack.org/VM**

See also poster today by Jonathan Claridge on creating a VM.

Reproducible research

VirtualClaw can simplify archiving codes used for publications.

For example, “ this code runs with VirtualClaw-4.5.1.”

This contains not just Clawpack 4.5.1, also the same version of gfortran, Python, etc. used originally.

For some papers with accompanying codes, see
www.clawpack.org/links

PyClaw

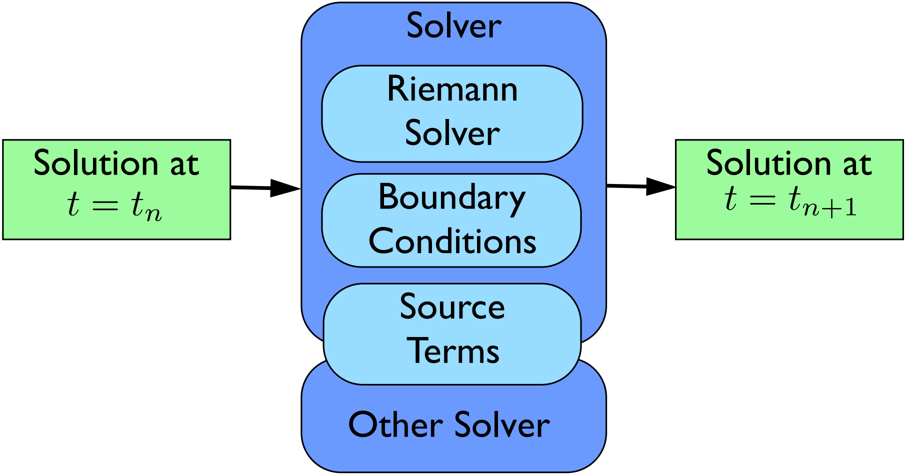
PyClaw contains:

- Tools for manipulating input data
- Tools for plotting results
- Pure Python version of parts of Clawpack

Major structures:

- `Solution` objects representing grid at a particular time
- `Solver` objects representing a type of solver - extensible

PyClaw Flow

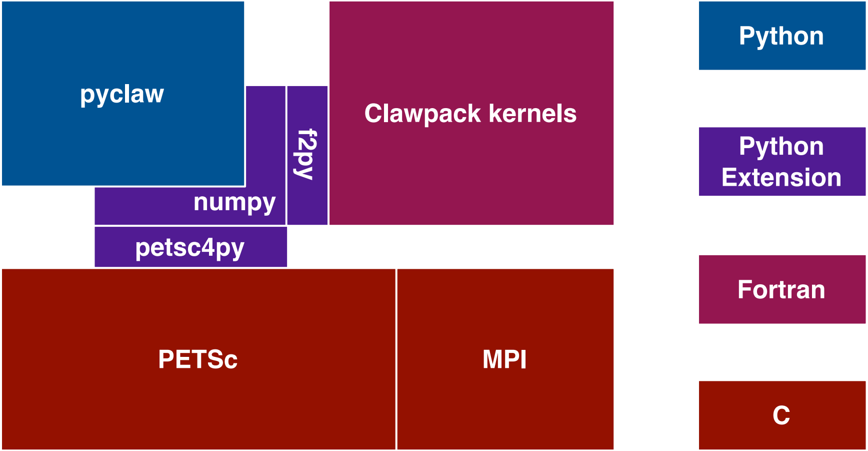

Solution at
$t = t_n$
Solver
Riemann Solver
Boundary Conditions
Source Terms
Other Solver
Solution at
$t = t_{n+1}$

PetClaw

PetClaw: A scalable distributed-memory solver for time-dependent nonlinear wave propagation.

- Authors
  - Amal Alghamdi, Aron Ahmadia, David I. Ketcheson (KAUST)
  - Matthew G. Knepley (U. Chicago)
  - Lisandro Dalcin (CIMEC)
- Computational strategy
  - **Clawpack** for computational kernel
  - **PETSc** for communication and implicit solvers (future)
  - Use python as the "glue", **petsc4py** and **PyClaw**
- Total code comprising PetClaw extension = 300 lines

PetClaw: Architecture



pyclaw

numpy

f2py

Clawpack kernels

petsc4py

PETSc

MPI

Python

Python Extension

Fortran

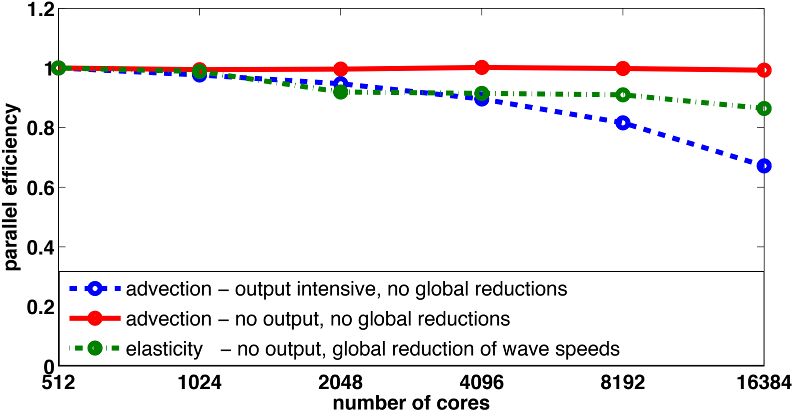
C

Timing Results

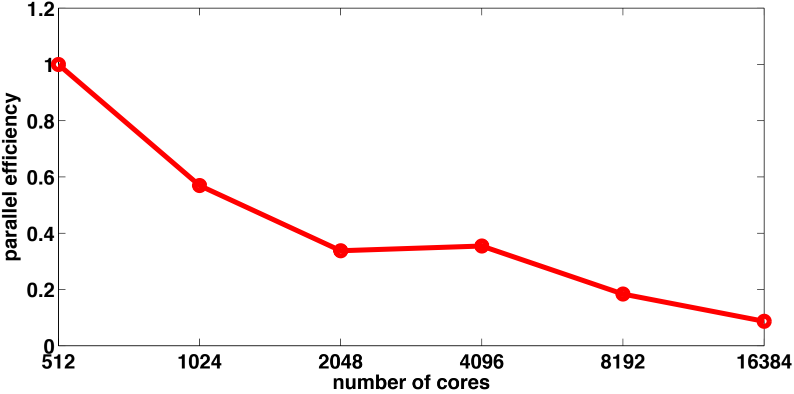
	Clawpack	**PetClaw** (Python)	**PetClaw** (Hybrid)
Advection	10.0 s	42.0 s	17.6 s
Elasticity	17.9 s	82.1 s	25.4 s

- All problems involve 10,000 cells and about 10,000 time steps
- Run on an Intel 3.06 GHz Intel Core 2 Duo Laptop

Weak Scaling



Catastrophic Loading



PetClaw References

- *"PetClaw: A Scalable Parallel Nonlinear Wave Propagation Solver for Python"*, with Amal Alghamdi, Aron Ahmadia, David I. Ketcheson, Matthew G. Knepley, and Lisandro Dalcin. Accepted for the 19th High Performance Computing Symposium, 2011.
- Obtain the code and reproduce the tests presented: `http://web.kaust.edu.sa/faculty/davidketcheson/PetClaw_HPC_paper.html`