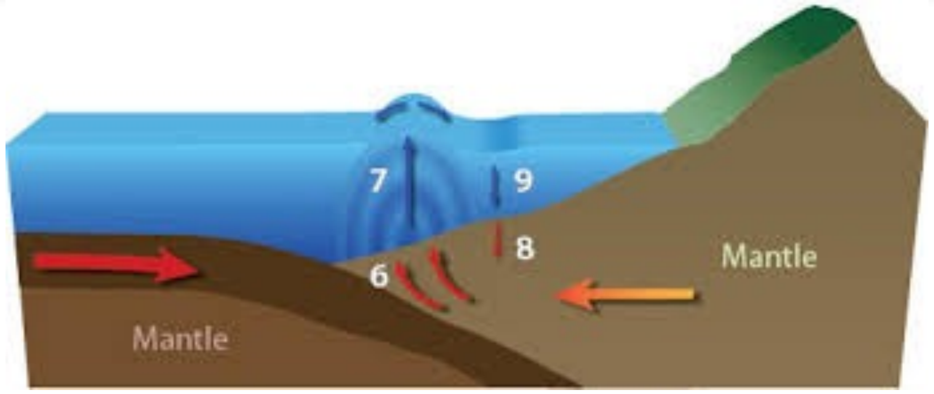
GeoClaw software development and testing

Donna Calhoun (Boise State University)

Randall J. LeVeque (Univ. of Washington)
Kyle Mandli (Columbia University)

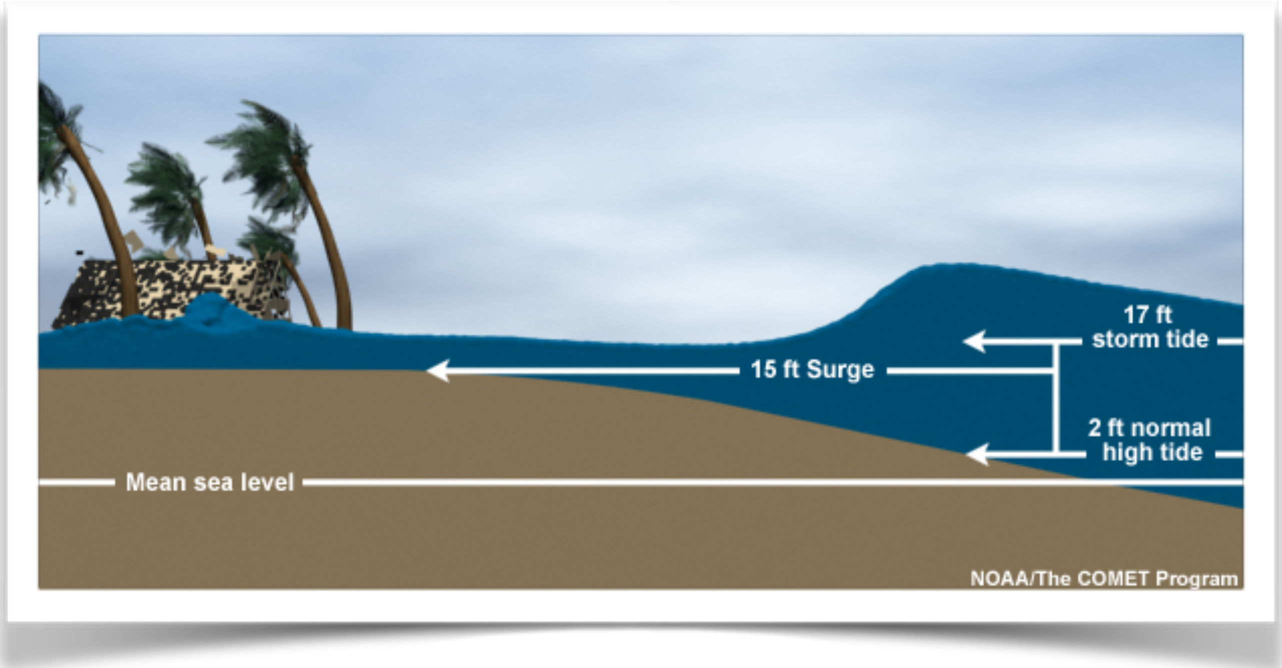
SIAM Geosciences
September 11-14, 2017
Erlangen, Germany

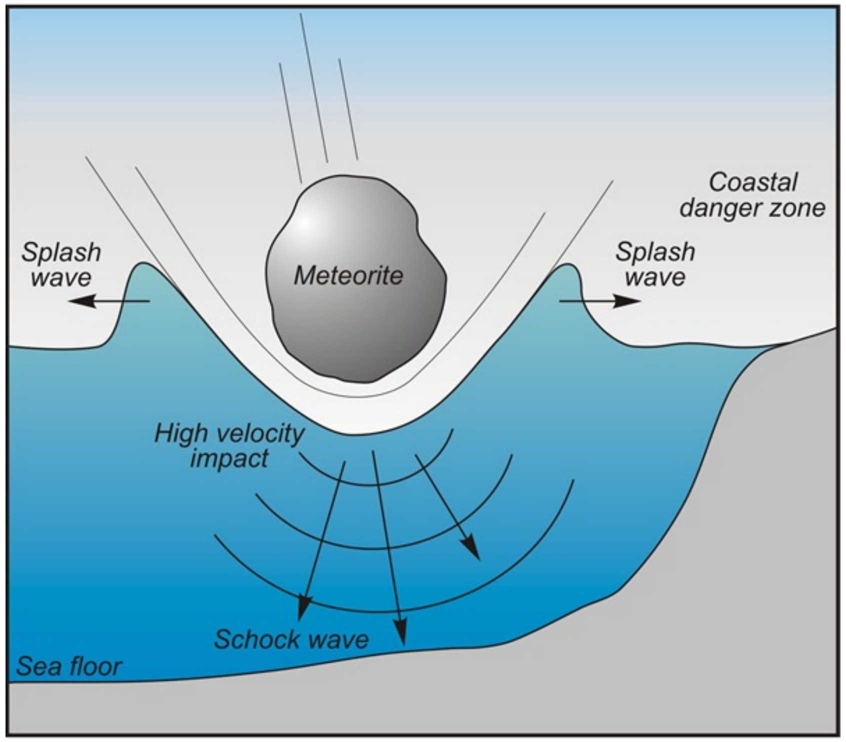
Depth-averaged flows with GeoClaw



Tsunami modeling (D. George, R. J. LeVeque, M. Berger, ….)

Storm surge modeling (K. Mandli)



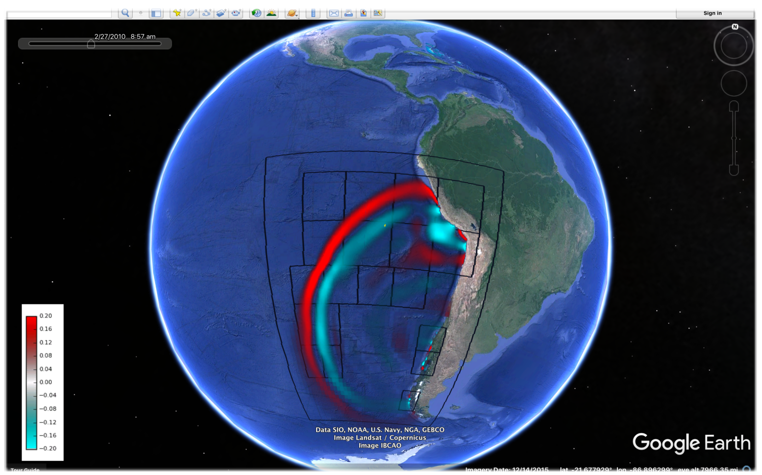


Asteroid-impact generated tsunamis (M. Berger, R. J. LeVeque)

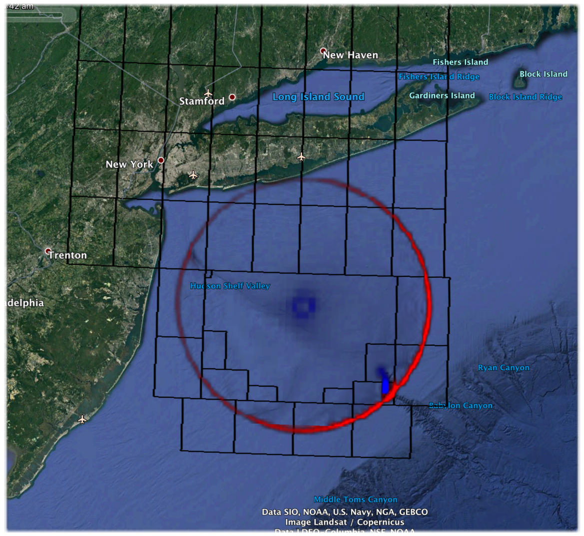
GeoClaw

GeoClaw is a depth-averaged (shallow water wave equations) code based on the finite volume, second order Cartesian grid methods in Clawpack

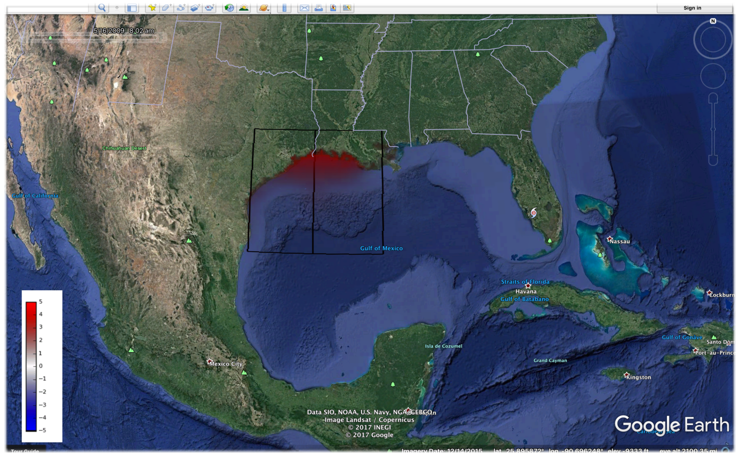
- Jointly developed by USGS, Univ. of Washington, NYU and Columbia researchers (D. George, R. J. LeVeque, M. Berger, K. Mandli)
- Widely used for modeling inundation and flooding from tsunamis, storm surges, landslides and debris flows



Chile 2010 Tsunami (Image : D. Calhoun)



Asteroid Impact M. Berger (Image : D. Calhoun)



Hurricane Ike storm surge K. Mandli (Image: D. Calhoun)

See ***http://www.geoclaw.org***

Brief overview of Clawpack

GeoClaw is based on Clawpack, a library for finite volume, wave-propagation algorithms developed by R. J. LeVeque,

- Both conservative and non-conservative terms are handled

$$q_t + A(x)q_x = 0 \qquad q_t + f(q)_x = 0 \qquad q_t + f(x,q)_x = 0$$

$$\downarrow$$

$$q_t + f'(q)q_x = 0$$

- Source terms (e.g. bathymetry) are handled using **f-waves**

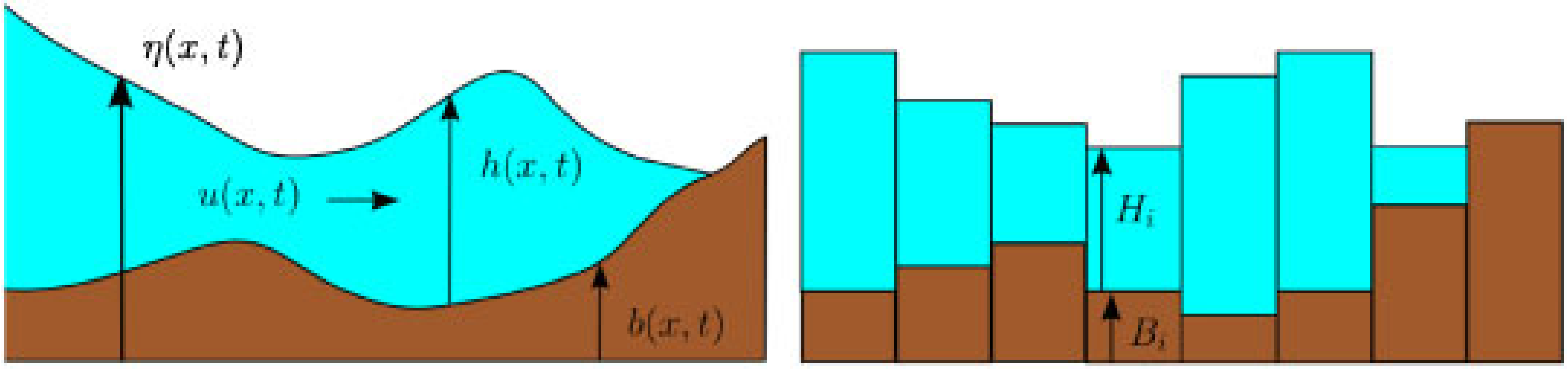
$$q_t + f(q)_x = \Psi(x,q) \longrightarrow q_t + \underbrace{(f(q) - \Delta x \Psi(x,q))_x}_{} = 0$$

Flux is split into waves to balance scheme

- Specialized Riemann solvers are designed to handle wetting drying

GeoClaw for depth averaged flows

$$
\begin{aligned}
&\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) + \frac{\partial}{\partial y}(hv) = 0, \\
&\frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) + \frac{\partial}{\partial y}(huv) = -gh\frac{\partial b}{\partial x} + S_{fx}, \\
&\frac{\partial}{\partial t}(hv) + \frac{\partial}{\partial x}(huv) + \frac{\partial}{\partial y}\left(hv^2 + \frac{1}{2}gh^2\right) = -gh\frac{\partial b}{\partial y} + S_{fy},
\end{aligned}
$$



D. L. George, "Adaptive finite volume methods with well-balanced Riemann solvers for modeling floods in rugged terrain: Application to the Malpasset dam-break flood (France, 1959)", *Int. J. Numer. Methods. Fluids*, 66 (2011), pp. 1000–1018.

GeoClaw

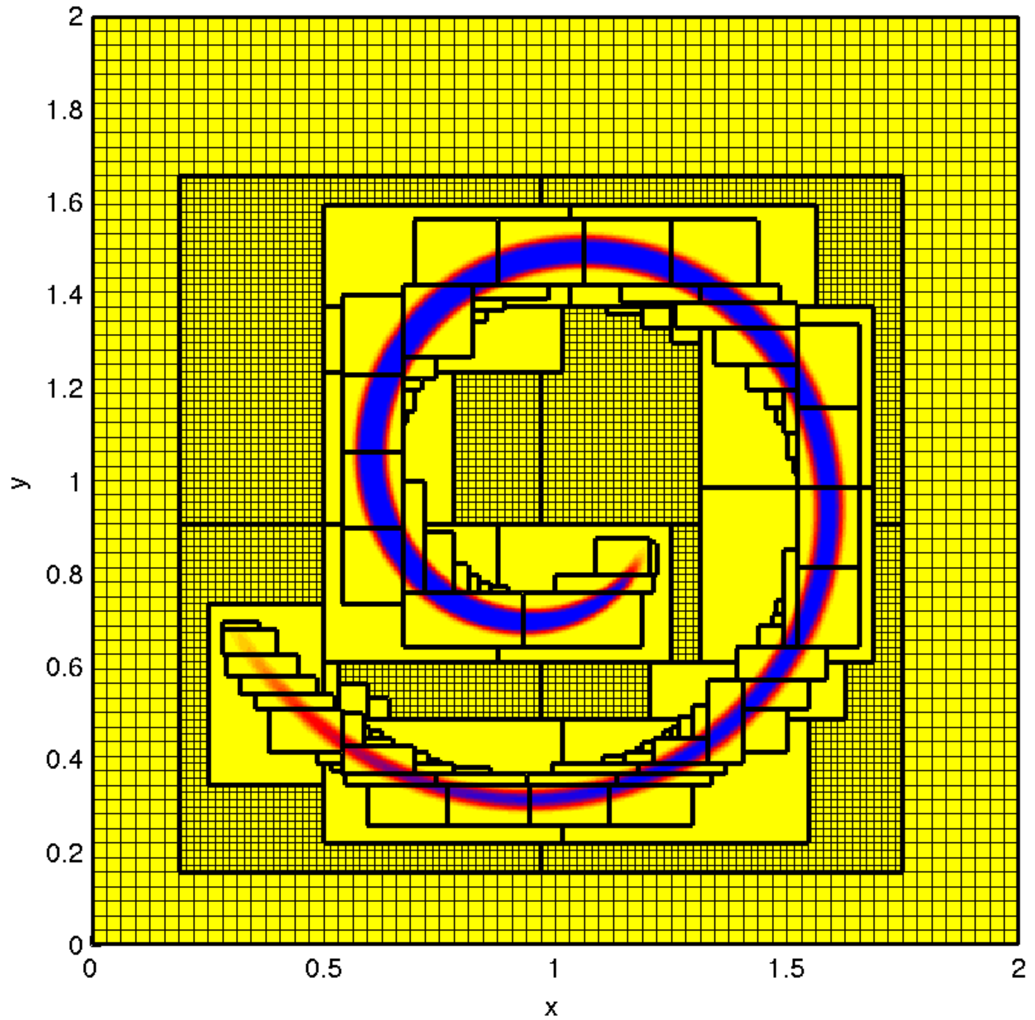
GeoClaw overcomes several technical challenges

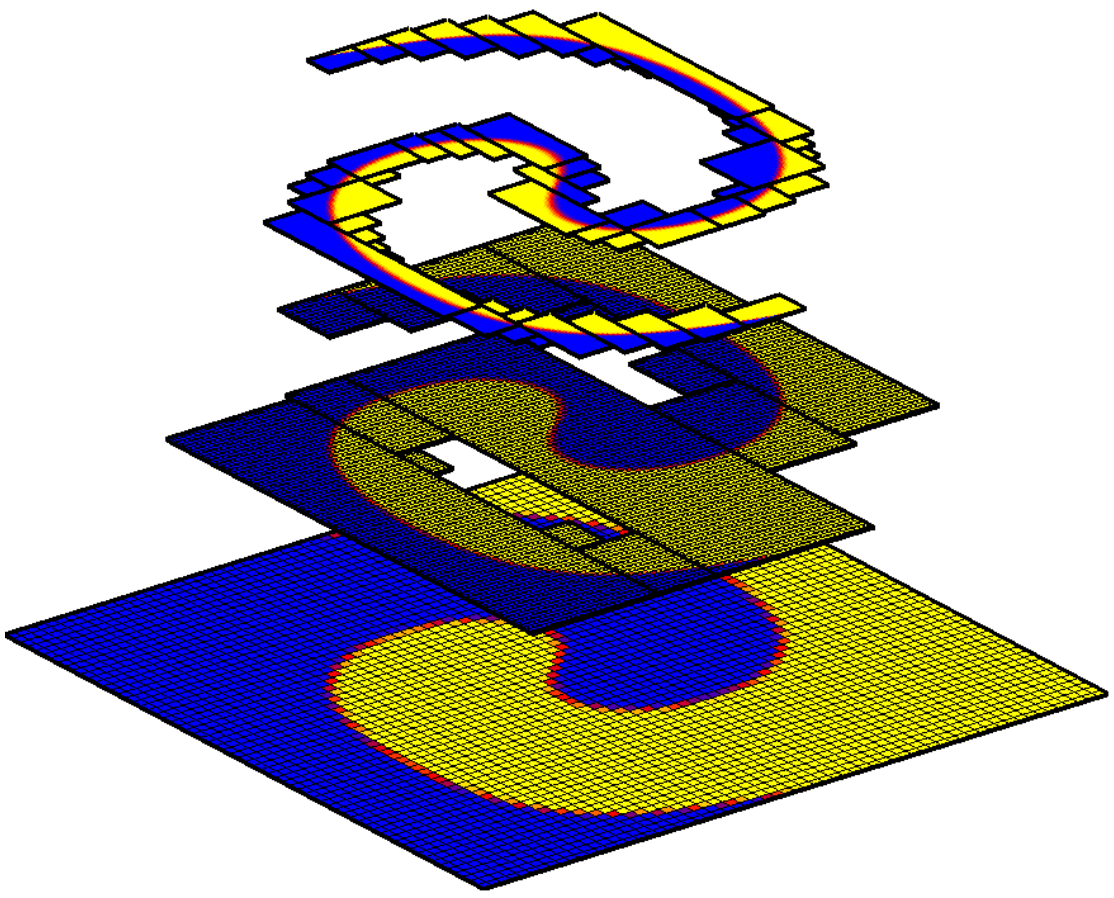
- Seamlessly handles reading and interpolation of multiple, possibly overlapping, **bathymetry or topography** files for given computational domain
- Riemann solver robustly handles **wet and dry states** and discontinuities in bathymetry
- **Well-balanced scheme** maintains steady states in presence of bathymetry (f-waves)
- Use of **adaptive mesh refinement** (AMR) means that resolution is allocated only where needed (dry land is resolved only at the coarsest levels)
- **Numerical gauges** allow for easy comparison with observational data

Adaptive Mesh Refinement (AMR)

Overlapping patch-based AMR (Structured AMR or SAMR)

Original approach (Berger, 1984)





Codes : Chombo (LBL), AMRClaw and GeoClaw (UW, NYU) , **Boxlib* (LBL)**, SAMRAI (LLNL), AMROC (Univ. of South Hampton) and many others

GeoClaw resources

- GeoClaw is part of the Clawpack "organization" on GitHub.
- Submodules include : classic, amrclaw, riemann, visclaw, pyclaw, and geoclaw.
- Routines written in Fortran and F90.
- Python scripts are used to build and run examples,
- Make files build Fortran code.

```
$ cd geoclaw/examples/tsunami/chile2010
$ make .exe
$ make data
$ make output
$ make plots
```

Quality assurance?

Group core developers (R. J. LeVeque, K. Mandli, D. Ketcheson) who approve pull requests.

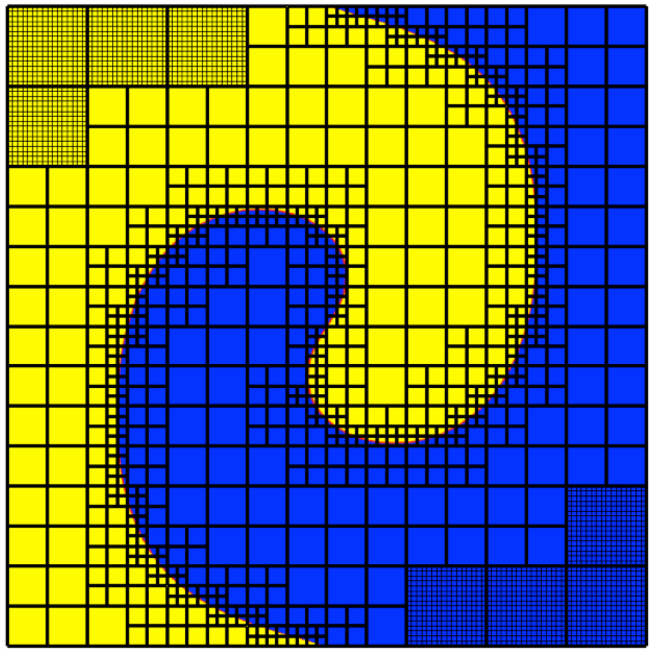
- Robust discussions on pull requests result in consistent interface decisions.
- Continuous integration with **Travis CI** assures that changes do not affect build.
  - Scripts ".travis.yml" run in the background on GitHub to ensure that all committed code can be compiled and run.
- Nosetests for run pre-defined tests
- Engaged user community on Google Groups raises questions, proposes new features and promotes widespread use of GeoClaw.

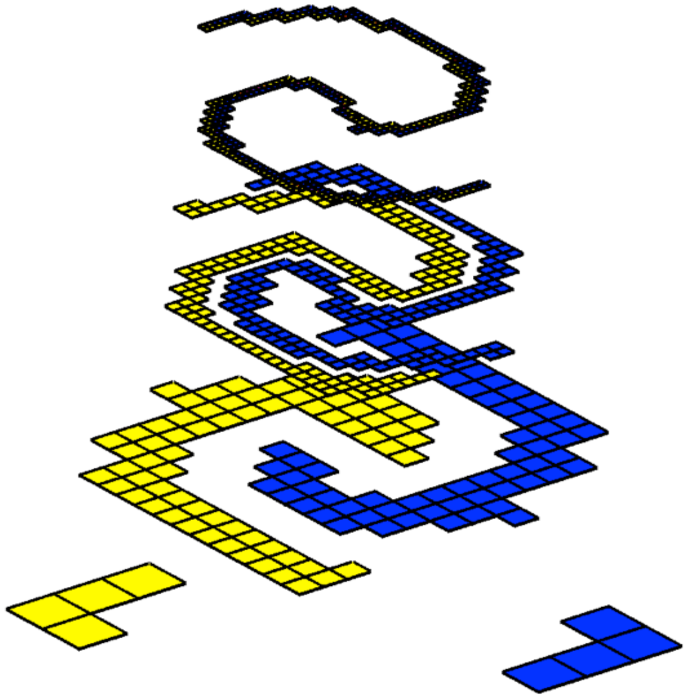
GeoClaw + ForestClaw

Some potential downsides to GeoClaw :

- Based on AMRClaw, a legacy code for adaptive mesh refinement,
- Relies on shared memory parallelism (i.e. OpenMP) - no distributed memory capabilities.
- Difficult to extend with new functionality, i.e. dispersive terms

Idea : Use core single grid routines in a new AMR framework





ForestClaw Project

A parallel, adaptive library for logically Cartesian, mapped, multi-block domains

Features of ForestClaw include :

- Uses the **highly scalable p4est** dynamic grid management library (C. Burstedde, Univ. of Bonn, Germany)
- Each leaf of the quadtree contains a fixed, uniform grid,
- Optional multi-rate time stepping strategy,
- Has **mapped, multi-block** capabilities, (cubed-sphere, for example) to allow for flexibility in physical domains,
- Modular design gives user flexibility in extending ForestClaw with Cartesian grid based solvers and packages.
- Uses essentially the same algorithmic components as patch-based AMR

Thanks to NSF for supporting this work

www.forestclaw.org

Extending ForestClaw

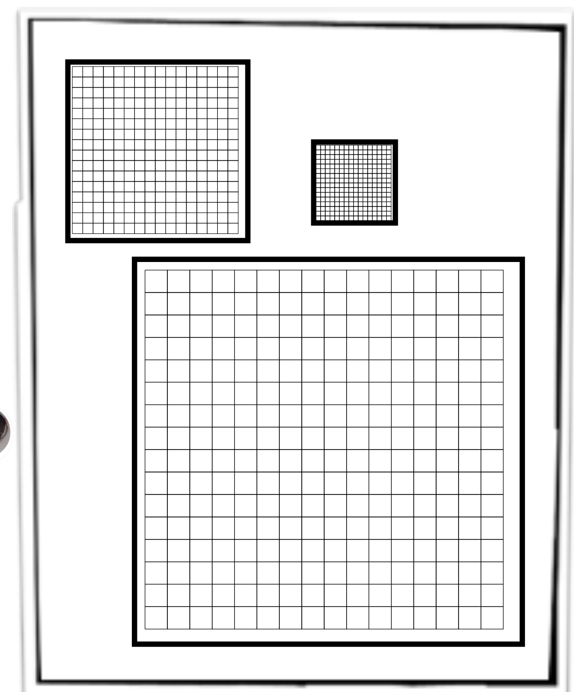
Core routines

p4est

ForestClaw

Time stepping, dynamic grid management, input/output.

Patches



Tasks (tagging, building patches, etc) customized through use of function pointers stored in "virtual tables"

Extensions

clawpack4.6

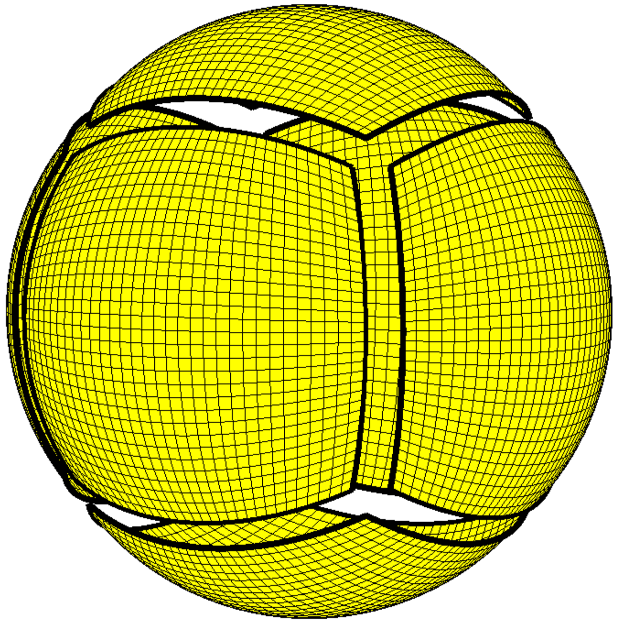
clawpack5

geoclaw

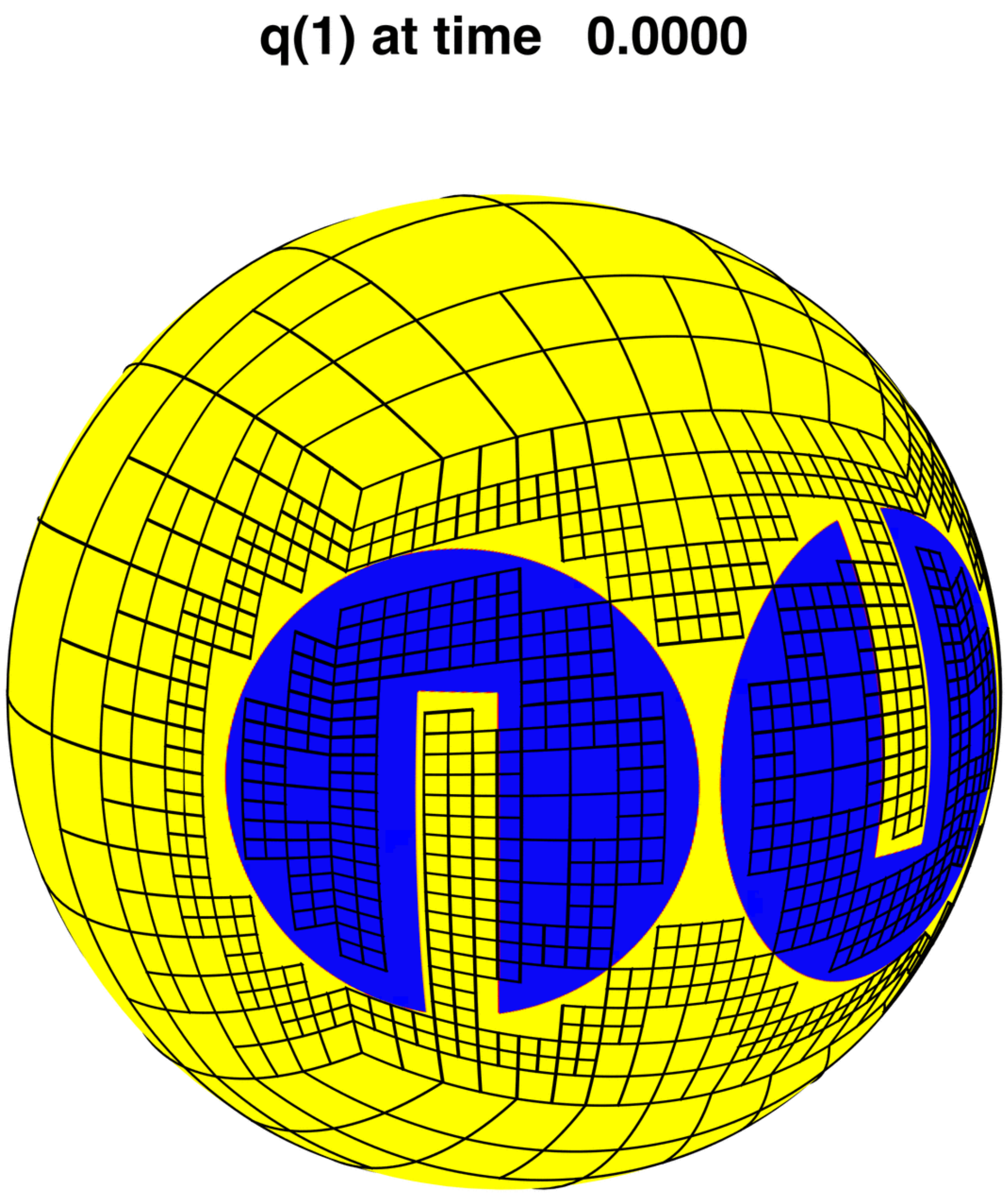
ash3d

ForestClaw is mostly C; solvers extension libraries left largely untouched (in original Fortran)

Cubed sphere with ForestClaw



Cubed sphere



Lauritzen, P. H., Skamarock, W. C., Prather, M. J., and and, M. A. Taylor. A standard test case suite for two-dimensional linear transport on the sphere. Geoscientific Model Development 5 (2012), 887–901.

Quality Assurance in ForestClaw?

- Lots of examples - acoustics, Burgers equation, Euler equations, advection, shallow water wave equations
- Lots of mappings - flat square and rectangular ("brick") domains, torus, sphere, five-patch disk,
- Different solvers - Both Clawpack 4.x and 5.0 to assure that the library mechanism is really modular, plus others.
- "Unit" tests - static problems with no solver.
- "patch free" problems to isolate data from core routines
- GNU Autotools build system, including "diskcheck" - checks that all files are included in Makefiles
- Git, issue tracker

Main goal of testing is to make sure code is really modular.

Example : Parallel testing

Write mini .ini file used to describe range of processors, levels or mesh sizes to vary.

Use Python scripts to :

- Expand these .ini files to sequence of full .ini files to run series
- Launch sequence of jobs on target platform
- Read resulting output files and scan for timing results
- Produce plots needed to check scalability.

Replicated problem

Grids per processor - **uniform**

Procs	1x1	2x2	4x4	8x8	16x16	32x32	64x64	128x128	256x256
1	4096	---	---	---	---	---	---	---	---
4	1024	4096	---	---	---	---	---	---	---
16	256	1024	4096	---	---	---	---	---	---
64	64	256	1024	4096	---	---	---	---	---
256	16	64	256	1024	4096	---	---	---	---
1024	---	16	64	256	1024	4096	---	---	---
4096	---	---	16	64	256	1024	4096	---	---
16384	---	---	---	16	64	256	1024	4096	---
65536	---	---	---	---	16	64	256	1024	4096

Grids per processor - **adaptive**

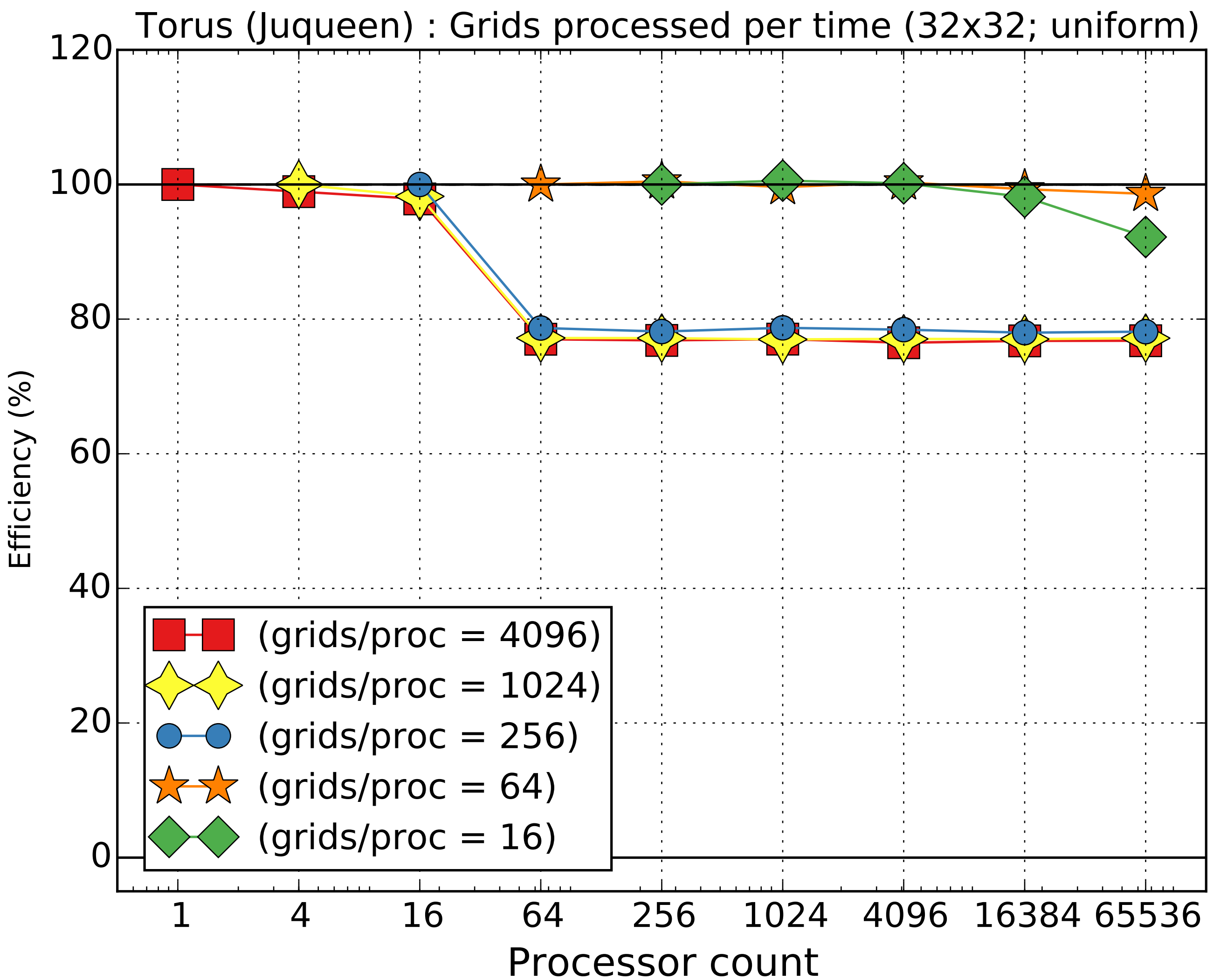
Procs	1x1	2x2	4x4	8x8	16x16	32x32	64x64	128x128	256x256
1	**5498**	---	---	---	---	---	---	---	---
4	1374	**5498**	---	---	---	---	---	---	---
16	343	1374	**5498**	---	---	---	---	---	---
64	85	343	1374	**5498**	---	---	---	---	---
256	21	85	343	1374	**5498**	---	---	---	---
1024	---	21	85	343	1374	**5498**	---	---	---
4096	---	---	21	85	343	1374	**5498**	---	---
16384	---	---	---	21	85	343	1374	**5498**	---
65536	---	---	---	---	21	85	343	1374	**5498**

Strong scaling

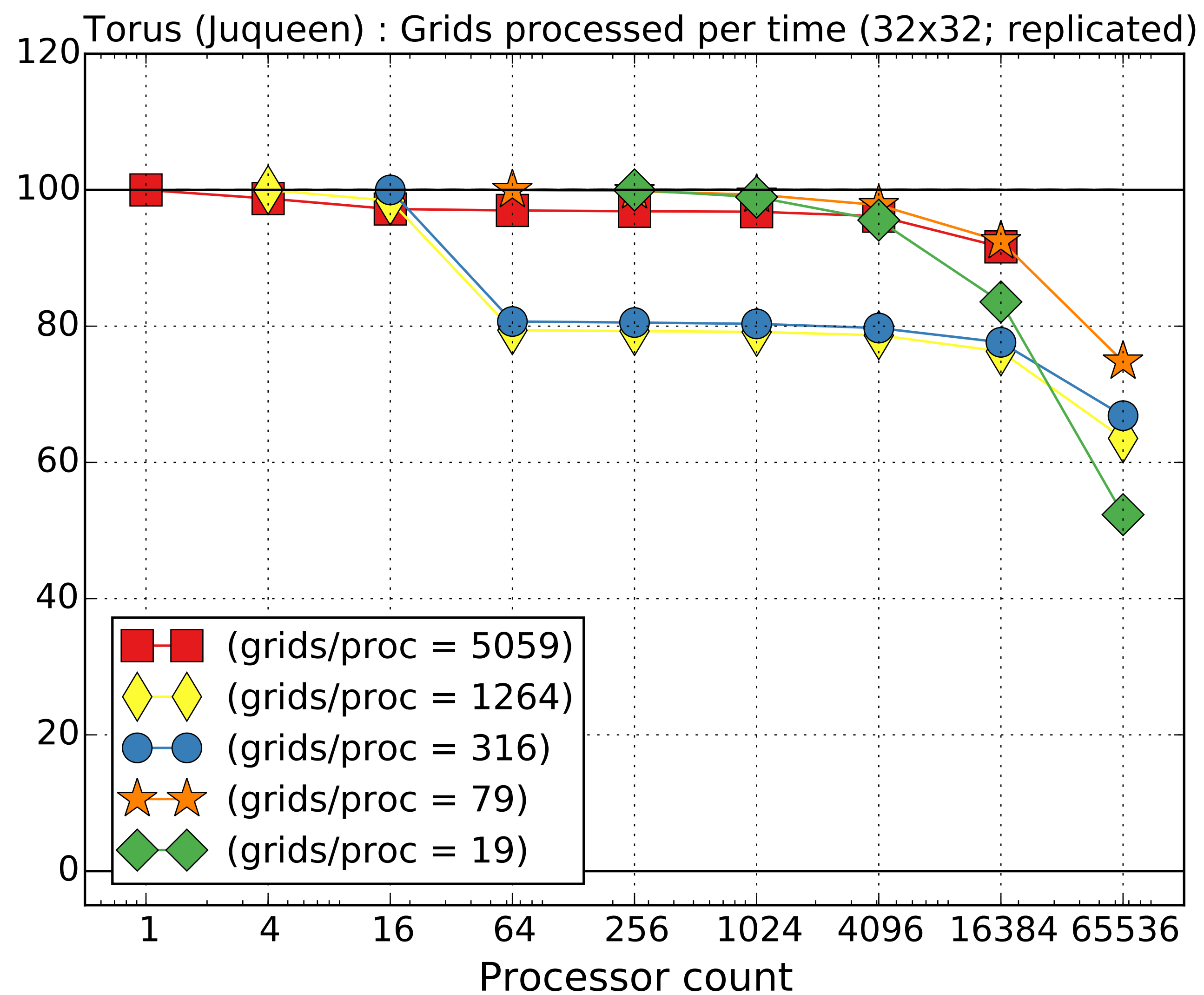
Weak scaling

Weak Scaling - 32x32 grids

Uniform



Adaptive



Grids processed per total wall time

Clawpack Ecosystem

See **www.clawpack.org/about.html**

PeerJ CompSci

✔ PEER-REVIEWED

Clawpack: building an open source ecosystem for solving hyperbolic PDEs

Distributed and Parallel Computing | Scientific Computing and Simulation

Kyle T. Mandli[1], Aron J. Ahmadia[2], Marsha Berger[3], Donna Calhoun[4], David L. George[5], Yiannis Hadjimichael[6], David I. Ketcheson[6], Grady I. Lemoine[7], Randall J. LeVeque[8]

August 8, 2016

10.7717/peerj-cs.68

For more information...

Clawpack and GeoClaw

http://www.clawpack.org

http://www.geoclaw.org

http://www.github.com/clawpack

ForestClaw and GeoClaw

http://www.forestclaw.org

http://www.github.com/ForestClaw