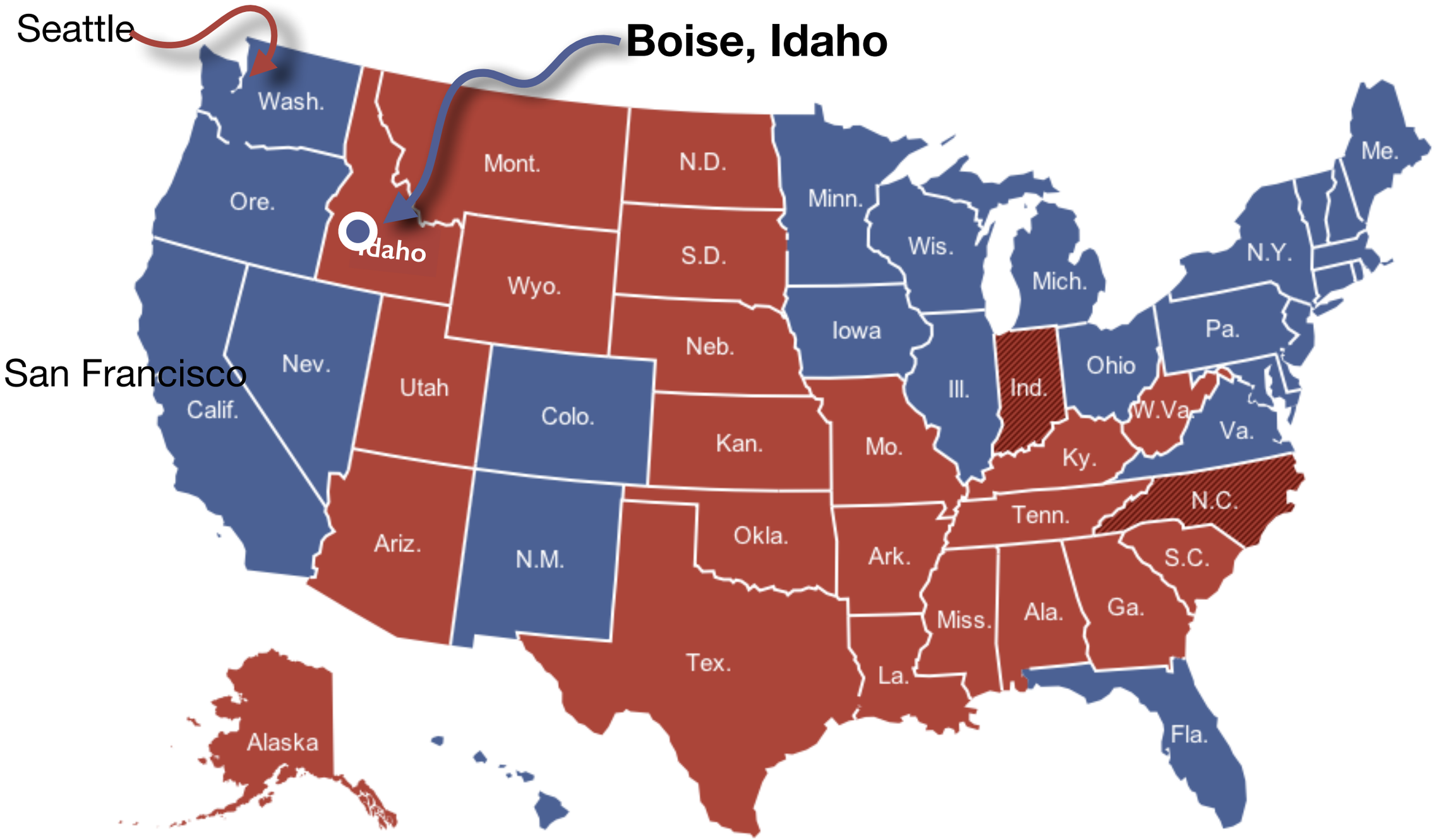
GeoClaw extension of ForestClaw

Donna Calhoun (Boise State University)

Hannah Spero (Boise State University),
Carsten Burstedde (Univ. of Bonn, Germany)
Melody Shih (Columbia/BSU); Kyle Mandli (Columbia Univ.); David George (USGS); R. J. LeVeque (UW); Marsha Berger (NYU)

GeoClaw Developer's Workshop
May 22, 2020

Where am I?



*2012 Electoral map (:-((

ForestClaw

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 numerical components as patch-based AMR (e.g. Berger-Oliger-Colella)

Thanks to NSF for supporting this work

www.forestclaw.org

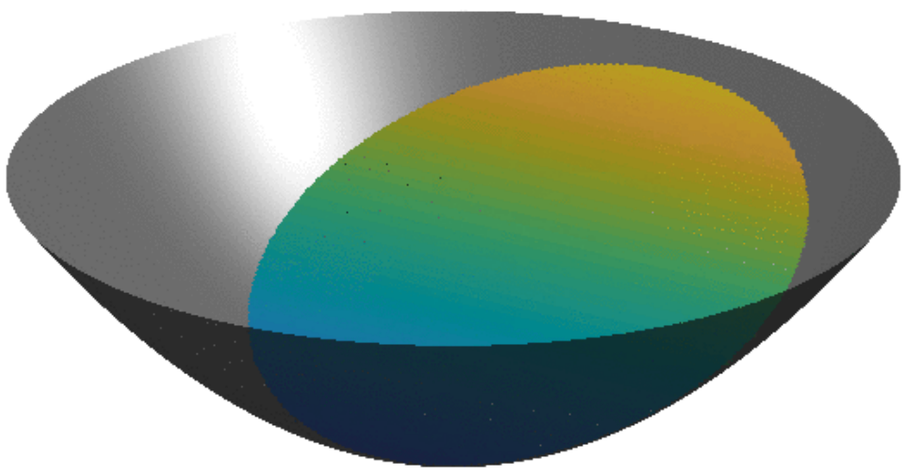
GeoClaw

GeoClaw library for depth-averaged geophysical flows (www.geoclaw.org)

GeoClaw overcomes several technical challenges

- **Riemann solver** robustly handles wet and dry states and discontinuities in topography - *no need to track the evolving flood boundary.*
- Seamlessly handles reading and interpolation of multiple, possibly overlapping, **topography files** (DEMs) for given computational domain
- **Well-balanced scheme** maintains steady states in presence of topography
- **Numerical gauges** allow for easy comparison with observational data
- Uses **OpenMP** (shared memory) parallelism
- Uses adaptive mesh refinement (**AMR**) to allocate resolution only where needed (dry land is resolved only at the coarsest levels)

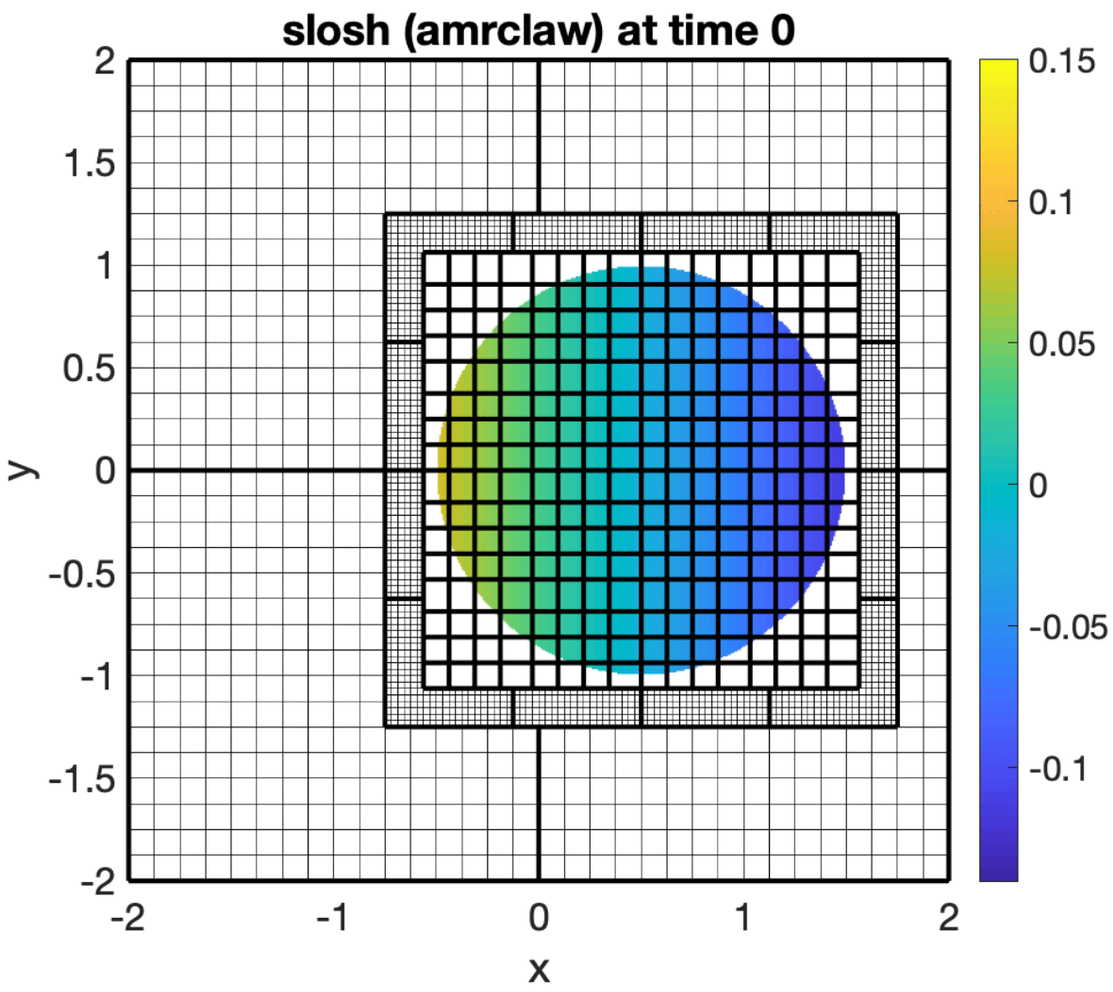
Bowl slosh



Bowl slosh example from GeoClaw (exact solution to the shallow water wave equations

GeoClaw

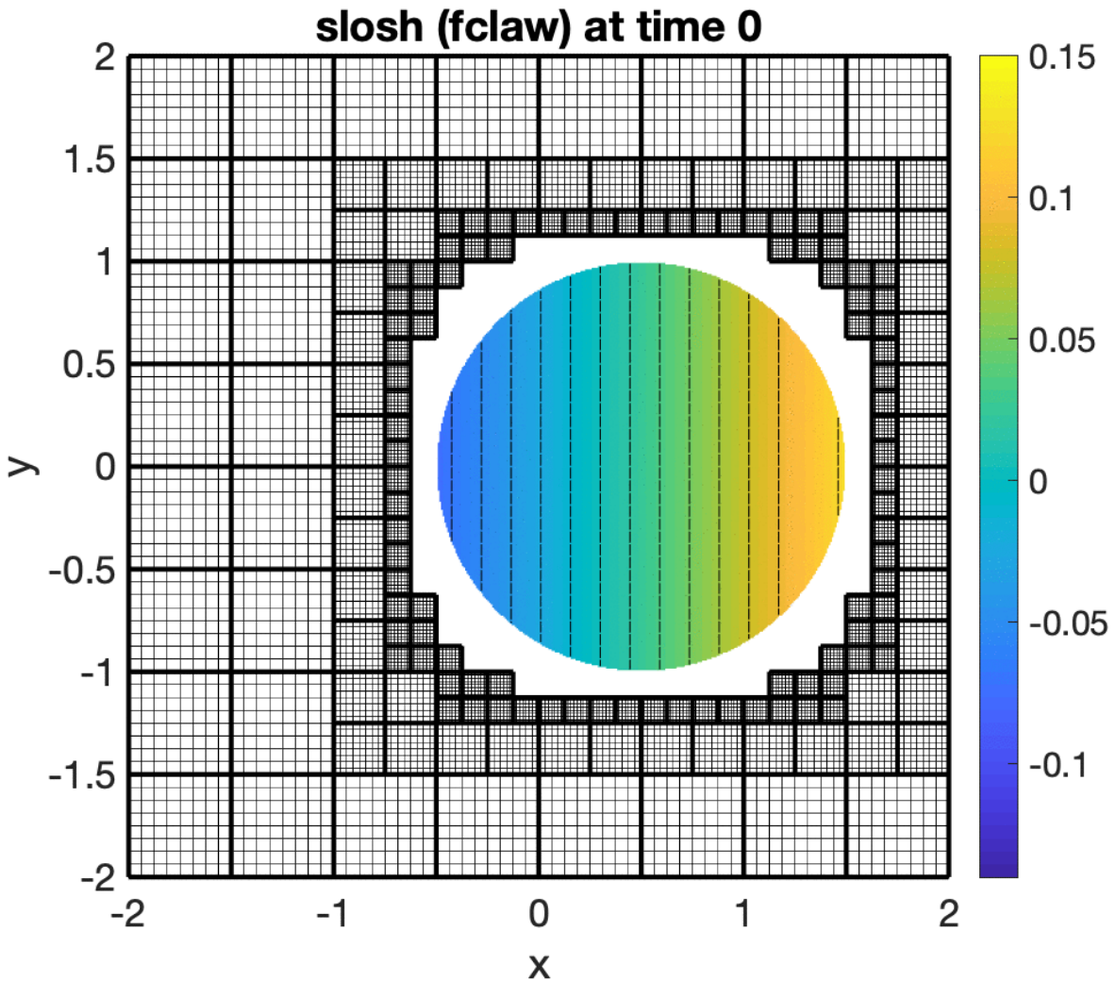
GeoClaw



- Patch-based AMR
- Refinement factors of 2,4,6,8, ...
- Rectangular patches can be of any size
- Finer grids are layered on top of coarser grids
- Uses original Berger-Rigoustos algorithm for dynamically adapting meshes.
- GeoClaw uses shared memory parallelism using OpenMP over patches. Mesh refinement is serial.
- CUDA version available

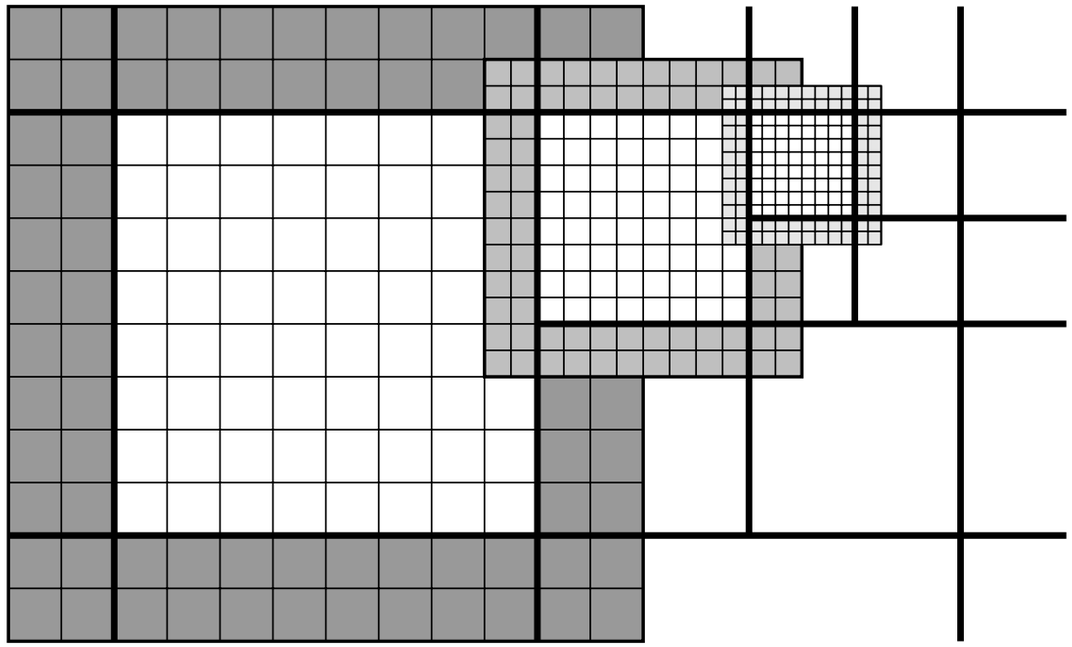
ForestClaw

ForestClaw



- Block-based AMR
- Refinement factors of 2
- Patches occupy **quadrants** or octants of a tree structure
- Grids partition the domain (no overlapping patches)
- Mesh management based on p4est (C. Burstedde, T. Isaac, L. Wilcox, www.p4est.org)
- Distributed parallelism using MPI
- CUDA version available
- Scaling demonstrated up to 65K cores on Jülich JUQUEEN (decommissioned)
- www.forestclaw.org

ForestClaw



In the “clawpatch” patch (used for finite volume solvers), each p4est quadrant is occupied by a single logically Cartesian grid, stored in contiguous memory, including ghost cells.

www.forestclaw.org

- Written mostly in object-oriented C
- Core routines are agnostic as to patch data, solvers used, etc.
- Most aspects of the PDE layer, including type of patch used, solver, interpolation and averaging, ghost-filling, can be customized
- Support for legacy codes
- ForestClaw is a “PDE layer” on top for the mesh management library p4est.

ForestClaw

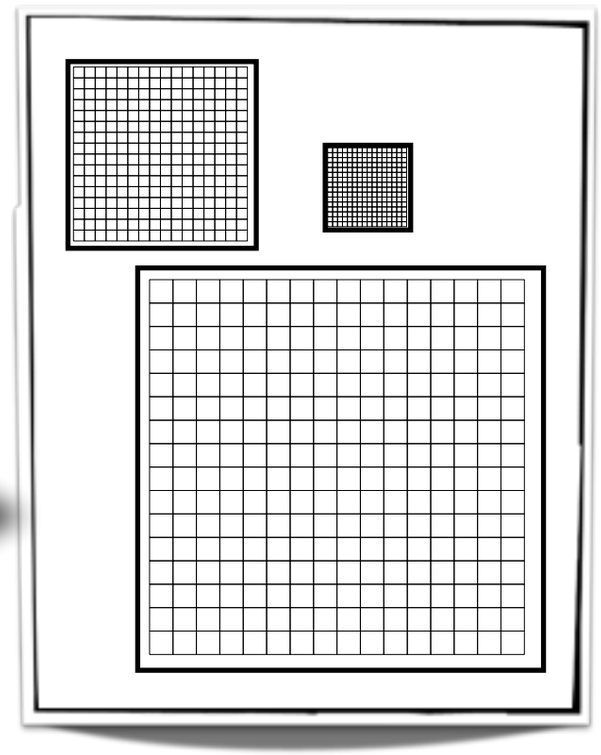
Core routines

p4est

ForestClaw

Time stepping, dynamic grid management, input/ output.

Patches



Stores solution data; provides tagging, ghost filling routines; customizes I/O.

Solvers

Clawpack4.6

Clawpack5

GeoClaw

ash3d

ThunderEgg (multigrid solver)

Provide routines for updating patches; store parameters;

Why use a tree-based AMR code?

- The tree based mesh refinement has some advantages over original patch-based AMR
  - Regular neighbor connectivity simplifies inter-grid communication
  - Non-overlapping composite grid structure simplifies post-processing.
  - Quadtree/octree well suited for emerging hardware - patches can all be processed simultaneously in CUDA blocks, for example,
  - Equal size patches and space-filling curve makes load-balancing straightforward, without the need for tiling patches.
  - Mesh management algorithms can be decentralized
  - Very limited meta-data requirements.
  - Potential disadvantage : 2:1 refinement ratio may lead to over-refinement in flows over large spatial domains (e.g. tsunamis)

Using an existing mesh library (e.g. p4est) reduces the burden of maintaining one of the most complex parts of an AMR code.

Why develop a new AMR code?

Adaptive mesh codes are big and difficult to maintain ...

Classic Clawpack (src/2d)

Language	files	blank	comment	code
Fortran 77	14	128	1093	1075
Fortran 90	12	212	287	587
SUM:	26	340	1380	1662

AMRClaw (src/2d)

Language	files	blank	comment	code
Fortran 77	99	1033	4291	6533
Fortran 90	42	1339	2187	4307
SUM:	141	2372	6478	10840

5.5x 6.5x

AMRClaw has over 5x number of files and over 6x number of lines of code

What is p4est?

- Highly scalable meshing library based on quadtree/octree refinement
- Manages a “forest-of-octrees” to allow for geometrically complex domains.
- Encapsulates AMR meshing details parallel load balancing, dynamic regridding, neighbor connectivity and so on
- Tree-based infrastructure allows for efficient searches needed to locate gauges, regions, and so on.
- Developed by Lucas Wilcox (NPS), Carsten Burstedde (Univ. of Bonn, Germany), Toby Issac (Georgia Tech) and others
- Used in Deal II, PETSc, NUMA and NUMO (NPS), Gorden Bell Prizes and other large scale projects.

http://www.p4est.org

GeoClaw extension of ForestClaw

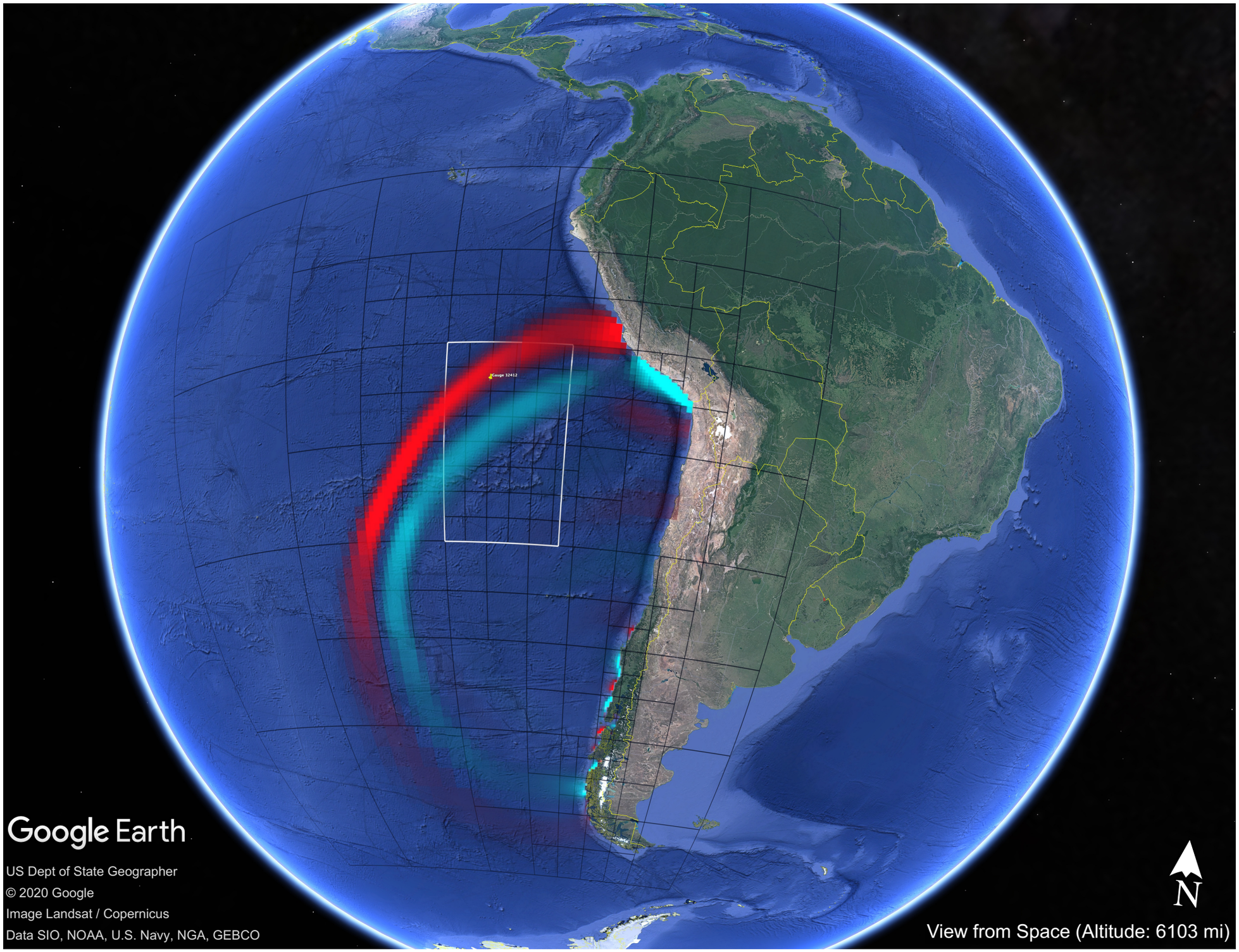
Extension uses GeoClaw solver routines as extension to core **ForestClaw** block-based AMR infrastructure

- Handful of key routines are ported directly from GeoClaw
  - -- Riemann solvers
  - -- Files for reading topo files and setting up topography in aux arrays
  - -- Adaptive criteria for tagging cells to re-generate mesh
- Numerical gauges are part of the core ForestClaw routines
- Visualization can be done using VisClaw, Matlab, VisIt, ParaView or Google Earth

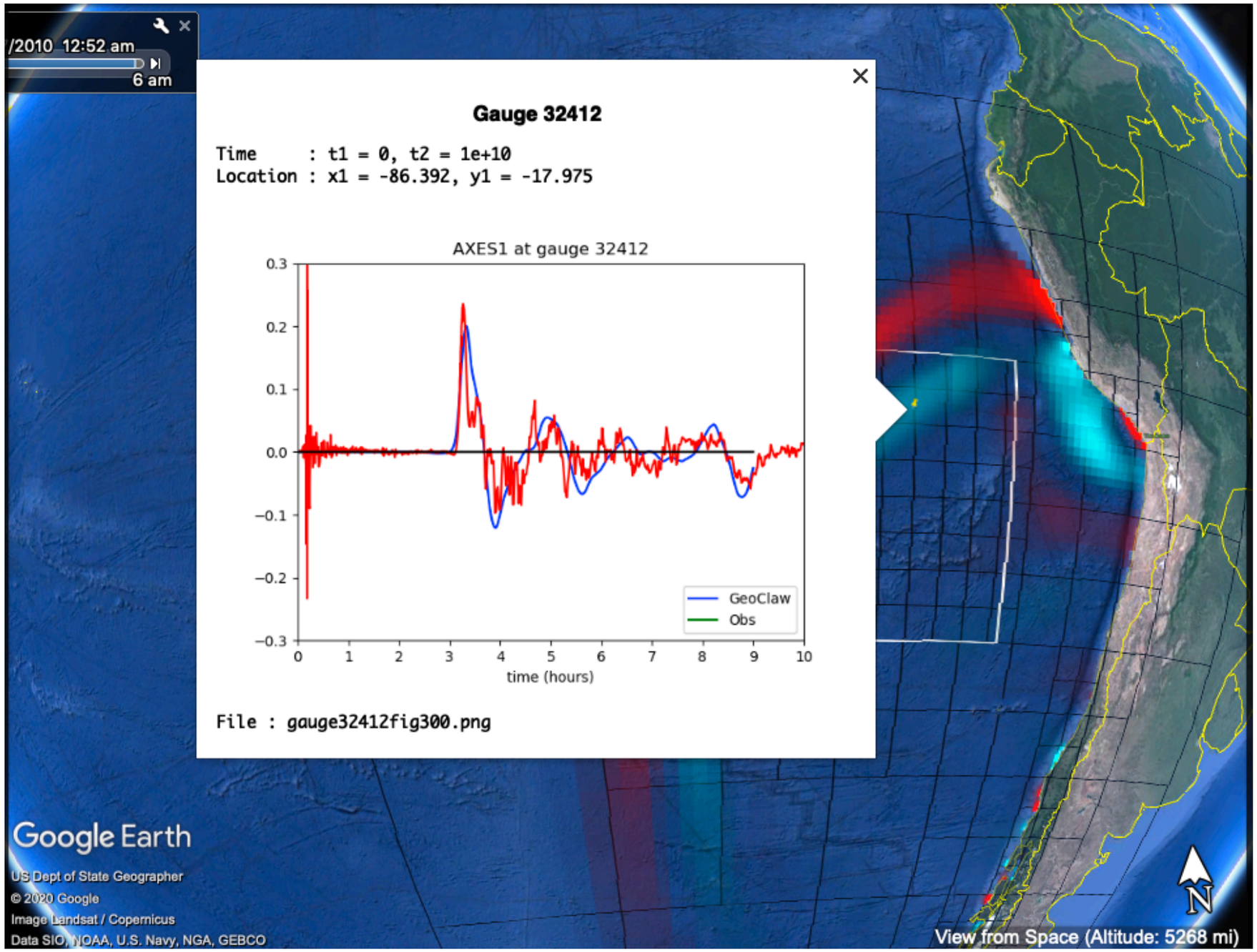
Parallel capabilities of ForestClaw

- Dynamic grid adaption is fully distributed to MPI processes/cores
- Space filling curve used for load balancing.

Chile 2010 example



Chile 2010

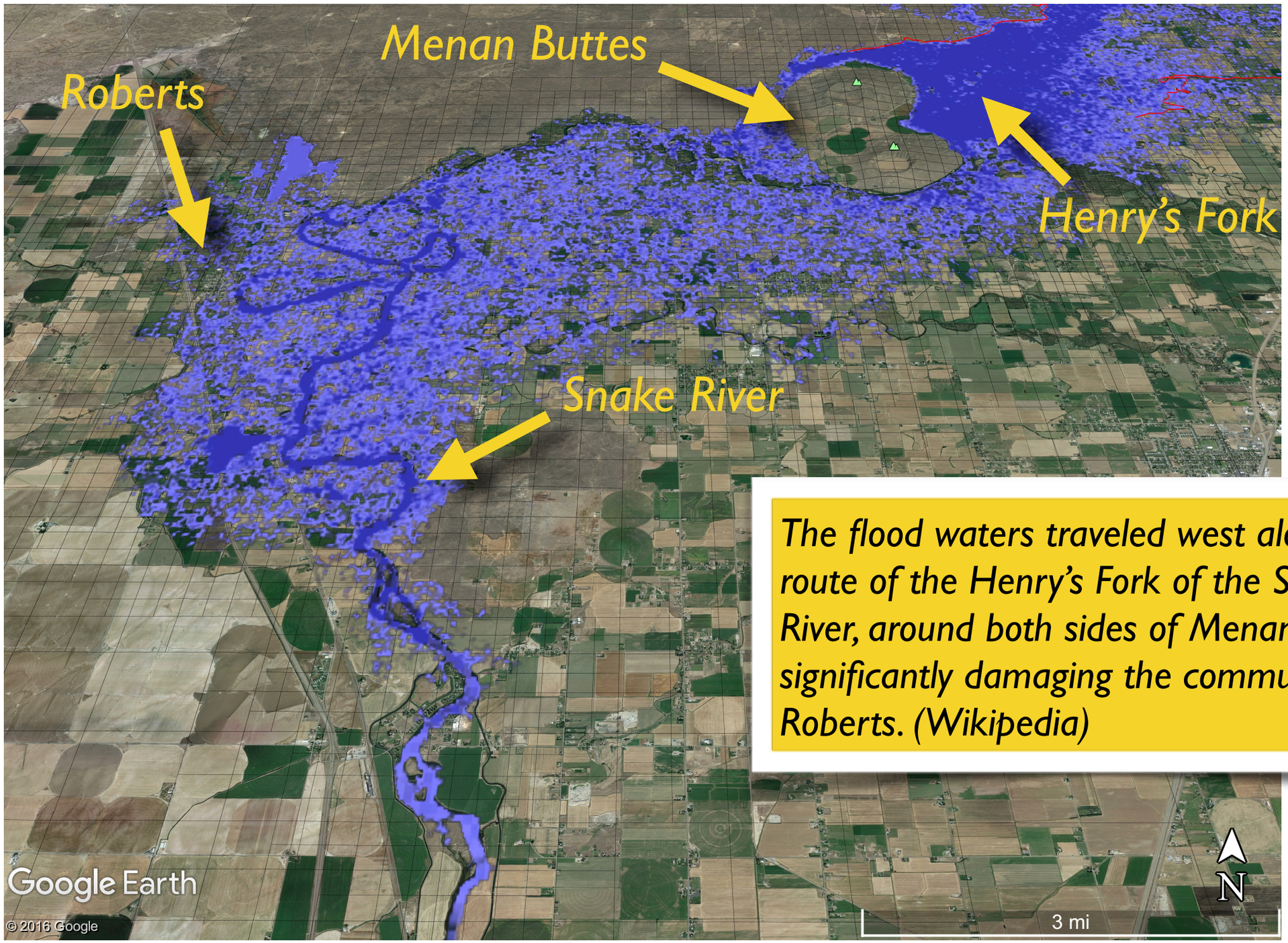


Google Earth visualization tools available in VisClaw

1976 Teton Dam Failure

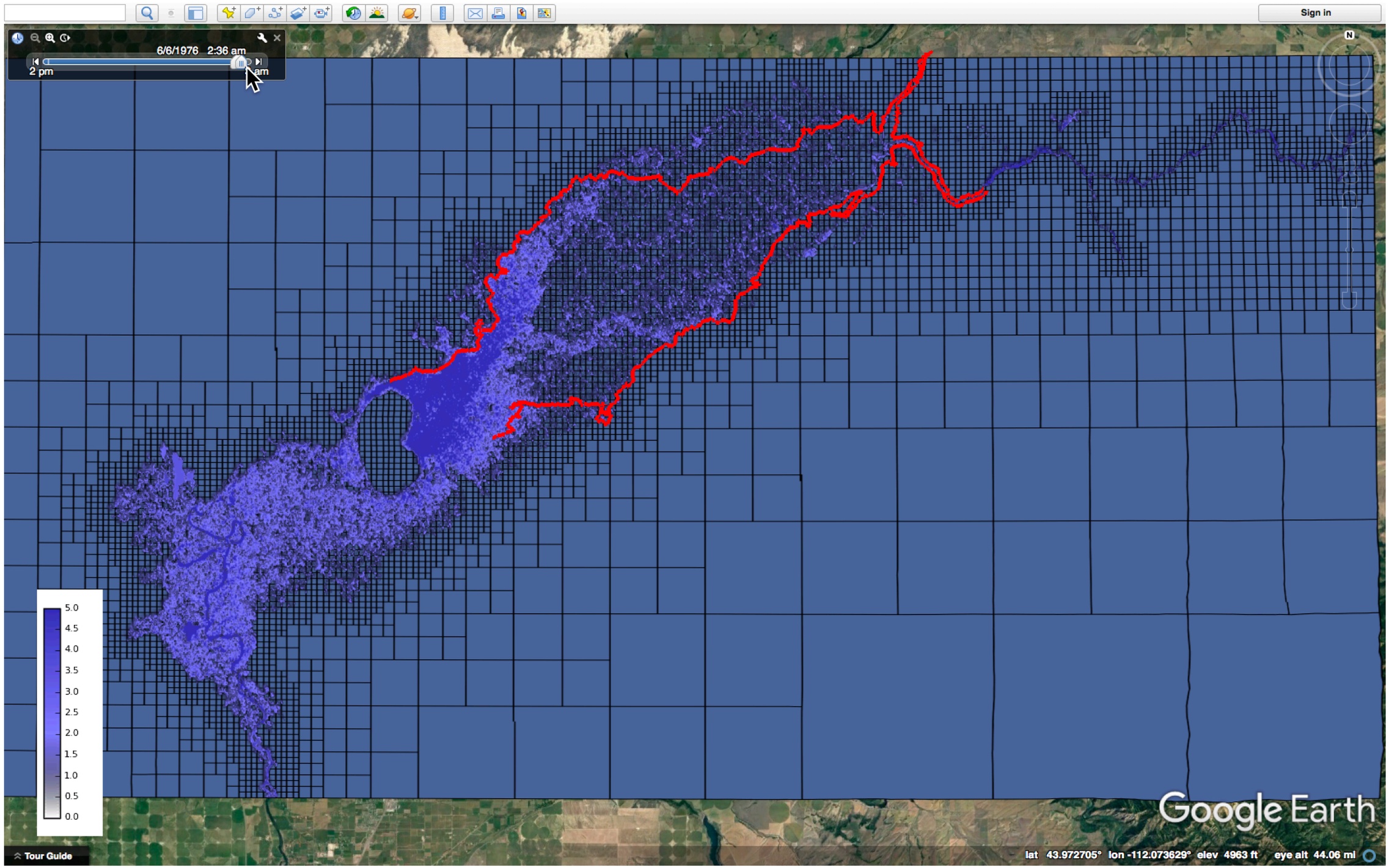
Teton Dam Failure in eastern Idaho - Project with BSU undergraduate Hannah Spero)

Snake River



The flood waters traveled west along the route of the Henry's Fork of the Snake River, around both sides of Menan Buttes, significantly damaging the community of Roberts. (Wikipedia)

Teton Dam failure



Parallel/AMR Efficiency

~ 10m resolution (8192 x 4096)

6.5 hours vs. 30 minutes

Procs	14	28	56	112	224
Wall (s)	23601.9	12510.6	6626.7	3499.7	1872.9
Speed-up	1.00	1.89	3.56	6.74	12.60
Efficiency	**100%**	**94%**	**89%**	**84%**	**79%**
Grids per processor	**670**	**334**	**167**	**83**	**41**

Procs	Wall	Advance	(%)	Ghost Comm	(%)	Ghost fill	(%)	Regrid	(%)	Speed-up	Par. eff.
14	23601.9	17706.4	**75%**	4500.4	**19%**	1343.3	**6%**	28.5	**0%**	1.0	**100%**
28	12510.6	8863.0	**71%**	2838.0	**23%**	772.4	**6%**	17.0	**0%**	1.9	**94%**
56	6626.7	4453.7	**67%**	1714.5	**26%**	432.6	**7%**	9.1	**0%**	3.6	**89%**
112	3499.7	2229.0	**64%**	1002.8	**29%**	248.1	**7%**	5.3	**0%**	6.7	**84%**
224	1872.9	1114.1	**59%**	602.8	**32%**	138.6	**7%**	3.3	**0%**	12.6	**79%**

Using ForestClaw

Consider trying out ForestClaw/GeoClaw if you

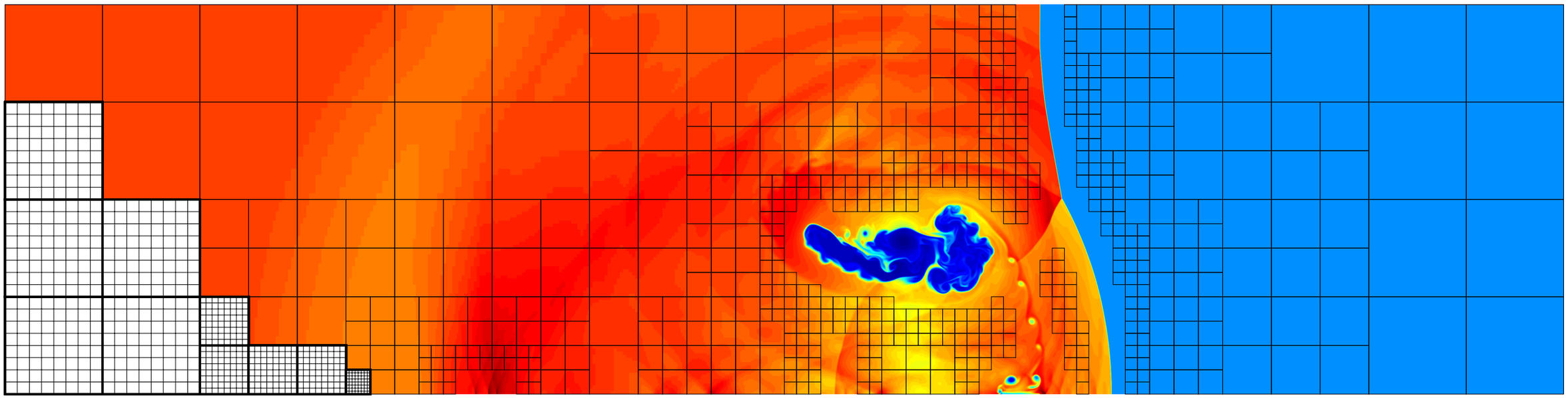
- Would like to run in a distributed environment
- Are interested in software development and would like to contribute!
- Would like to use more complex geometry (cubed-sphere, for example)
- Might like to see how GPUs performs

See Wiki on GitHub for install instructions

- Clone from git repo at www.github.com/ForestClaw
- Uses autoconf/automake to install
- Need to install MPI, C/C++, Fortran compilers
- The p4est library is installed as a submodule
- Uses familiar setrun.py (“make_data.py”) and setplot.py (“make_plots.py”)

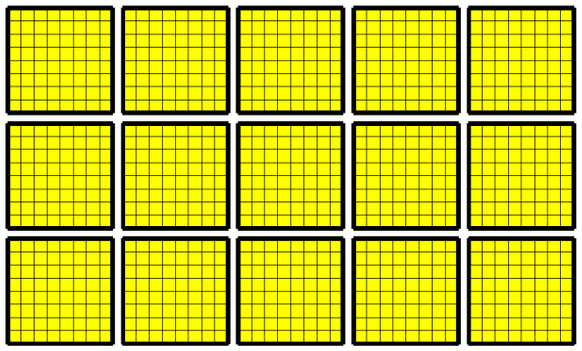
Questions?

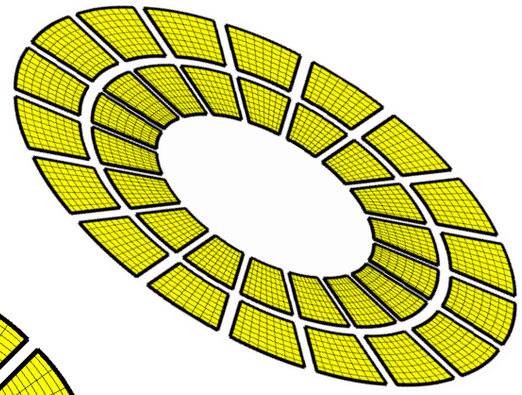
ForestClaw - multi block features

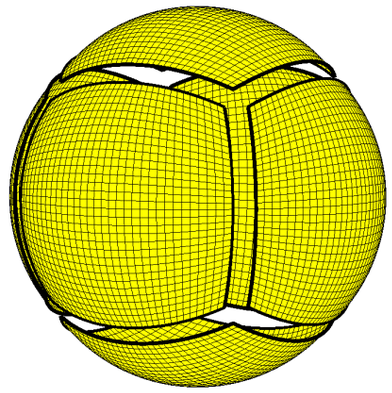


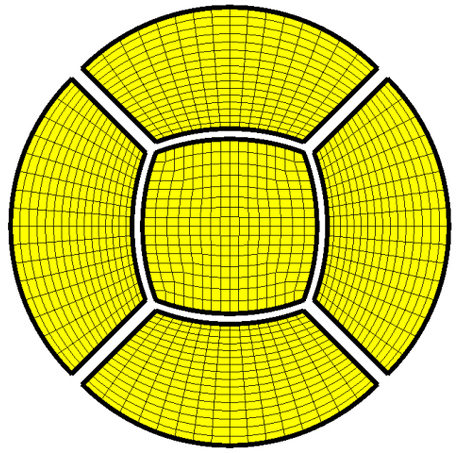
Shockbubble simulation using Clawpack (www.clawpack.org)

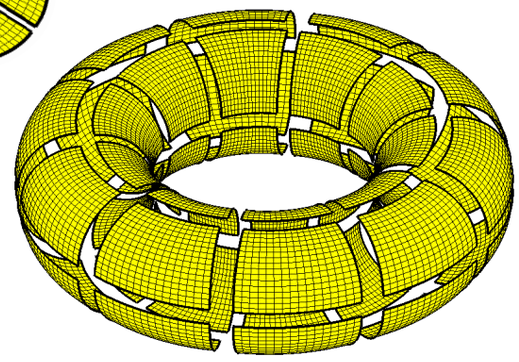
extension of ForestClaw on 4x1 multi block domain



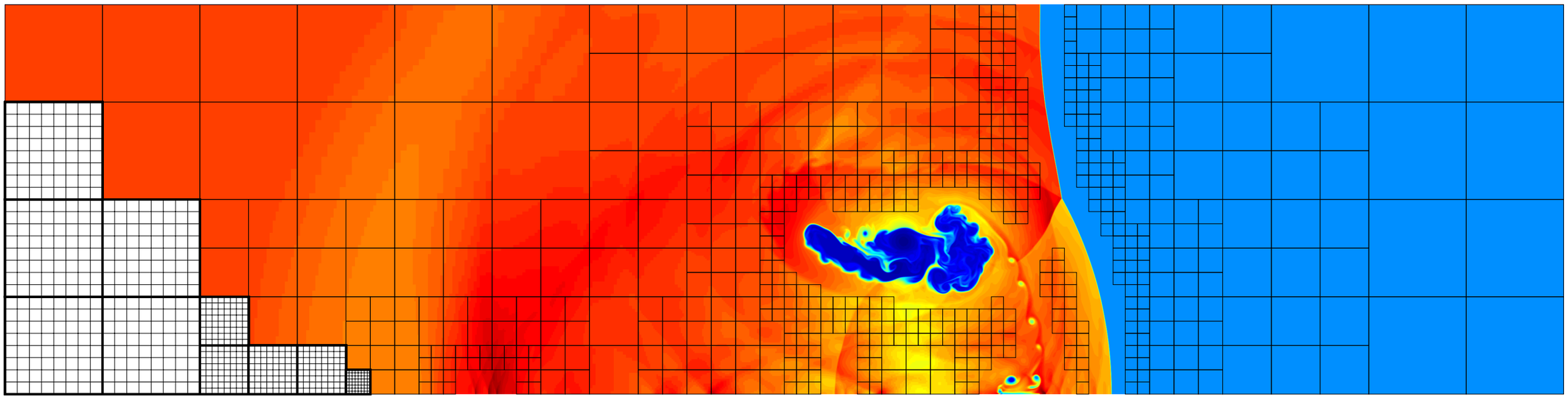






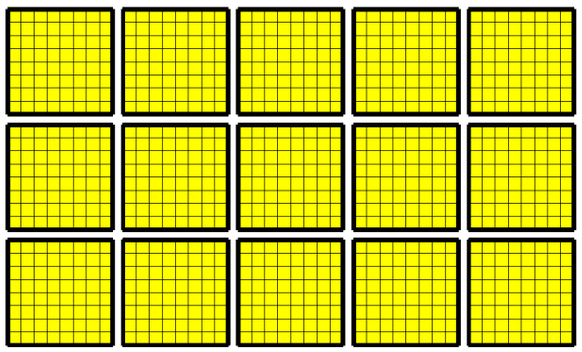


ForestClaw - multi block features

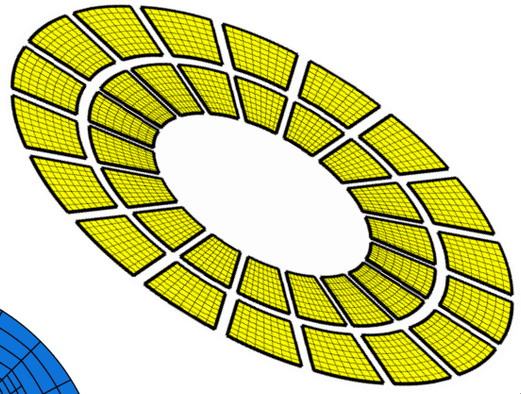


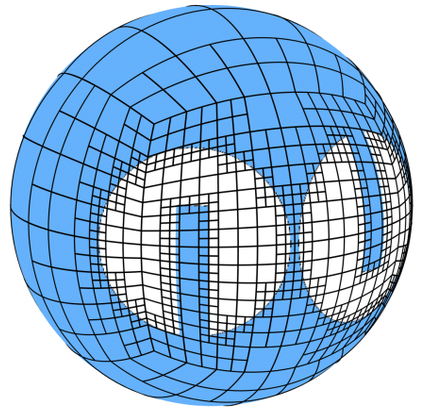
Shockbubble simulation using Clawpack (www.clawpack.org)

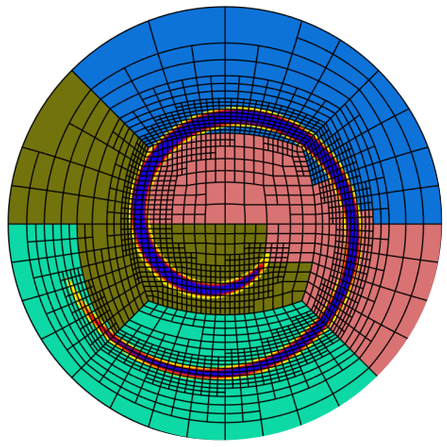
extension of ForestClaw on 4x1 multi block domain

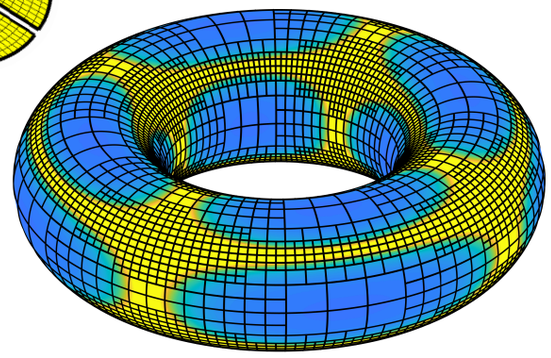


Solvers based on finite volume wave propagation algorithms in Clawpack (R. J. LeVeque)

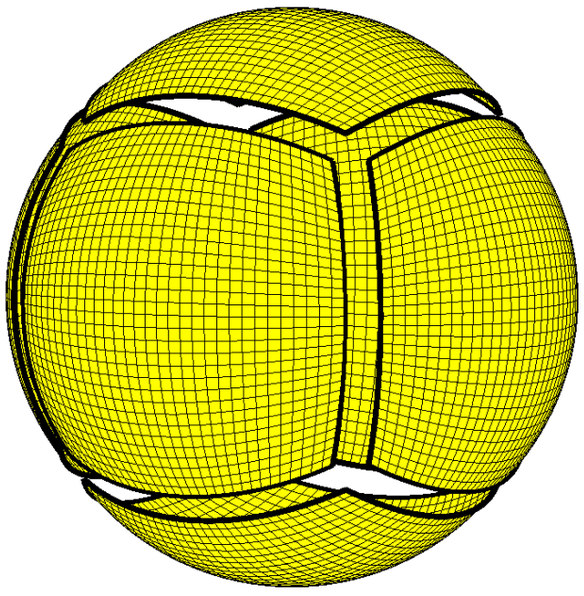






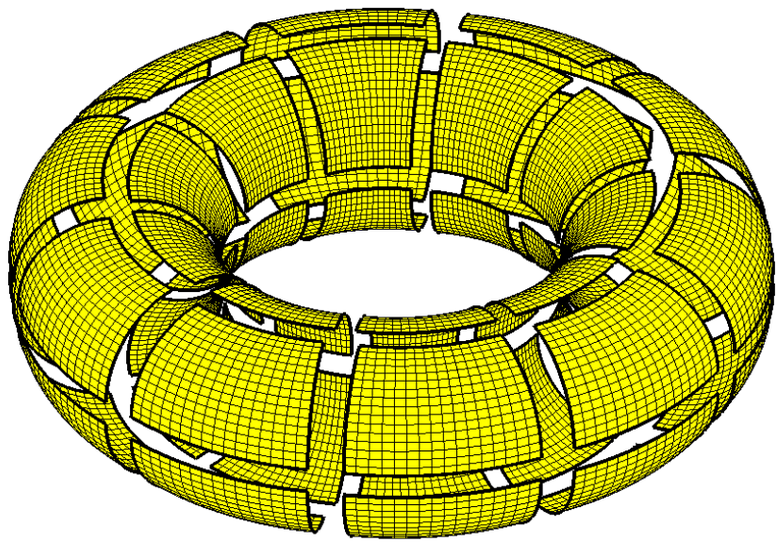


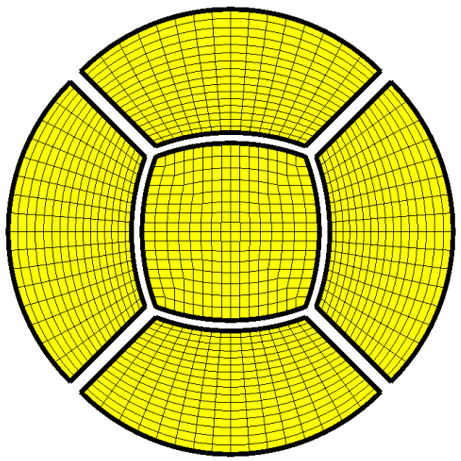
Multi-block capabilities

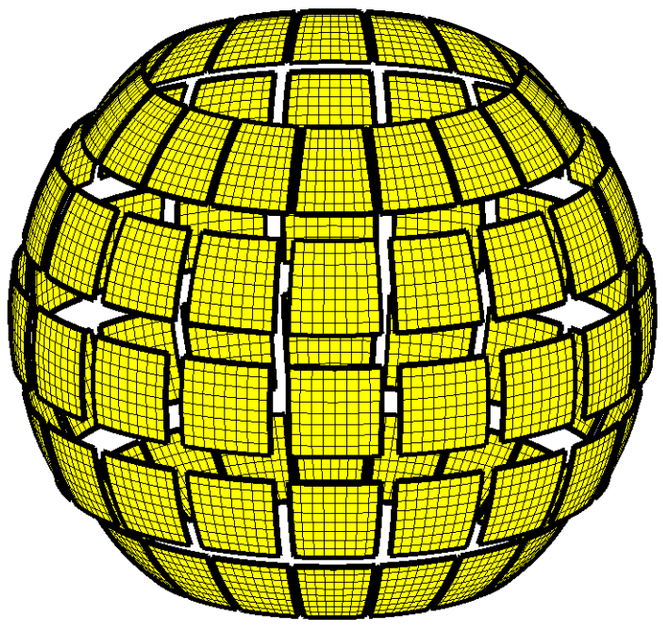


Cubed sphere

Torus

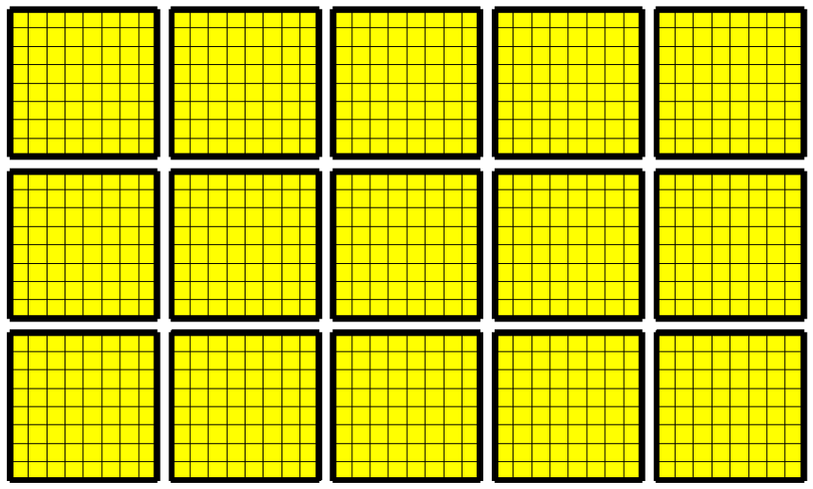




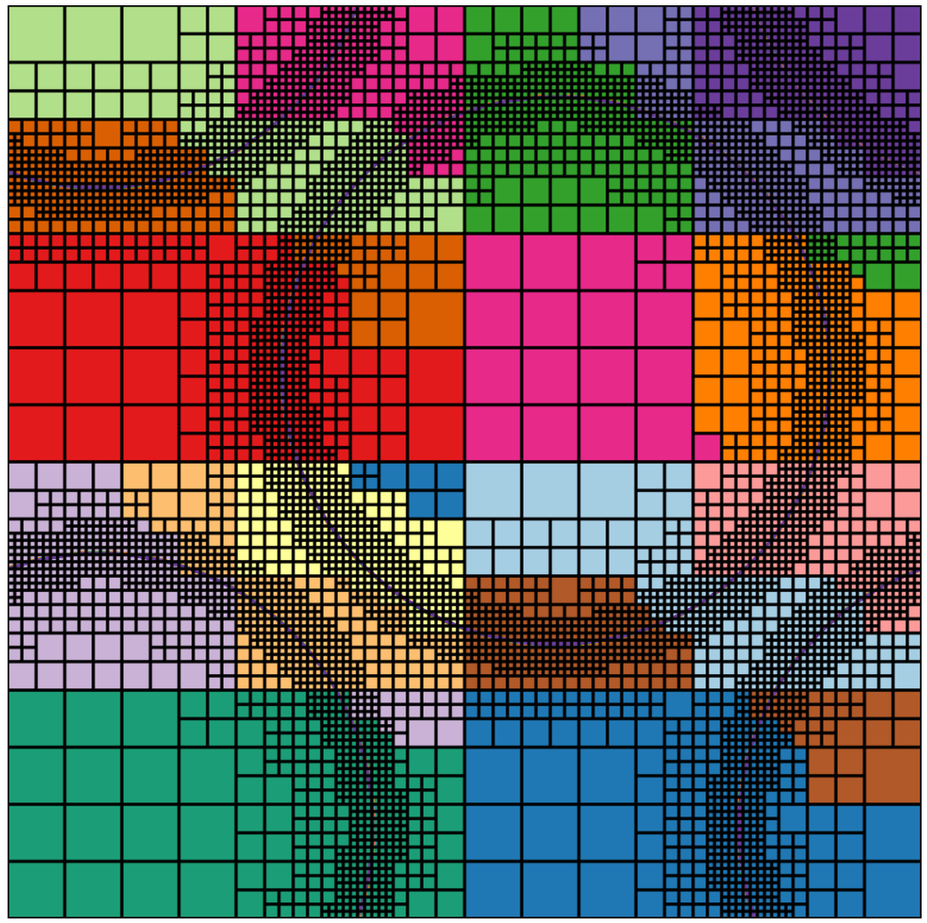


Lat/Long coordinates

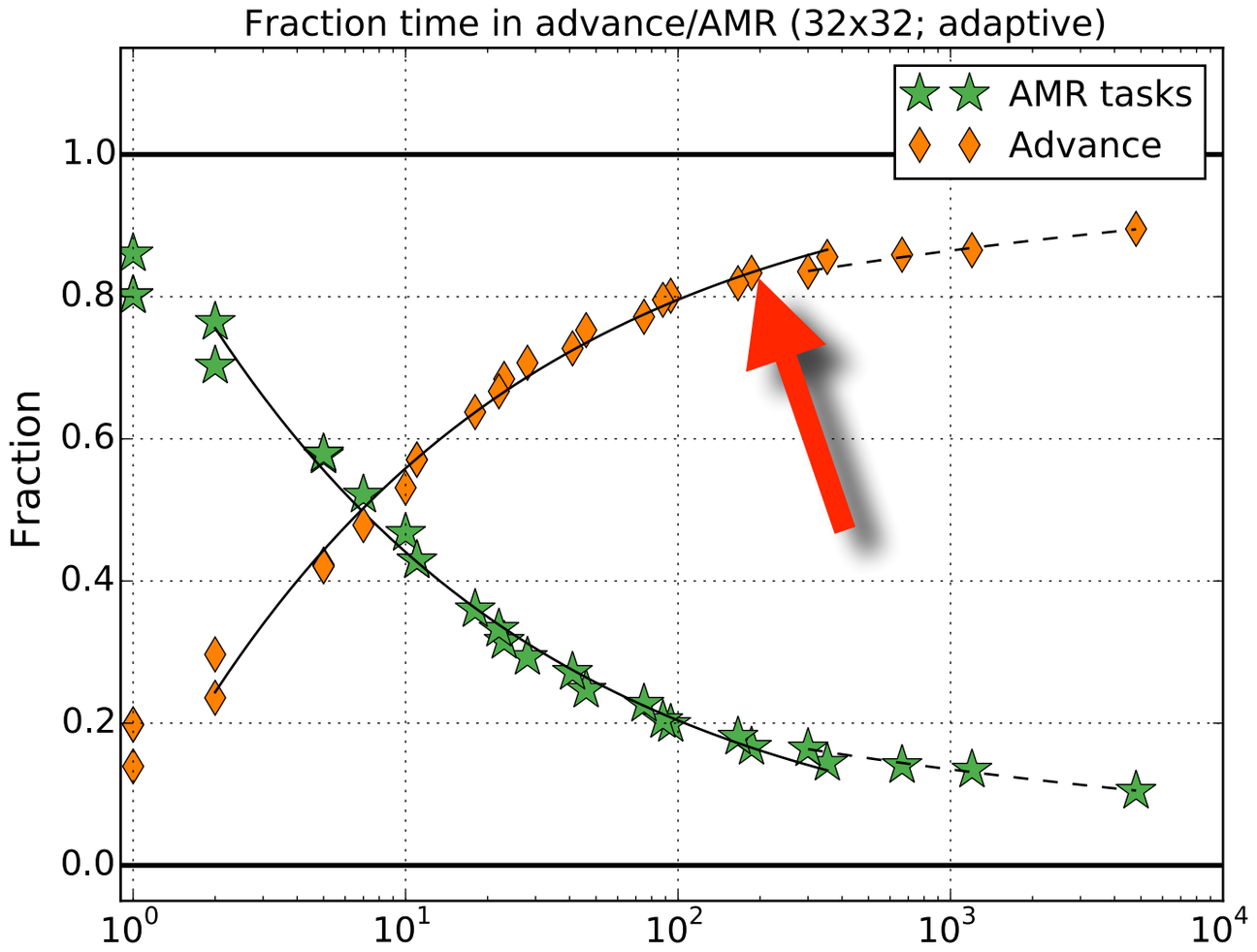
"Brick domain"



Parallel scaling (BlueGene/Q)



Strong scaling for single grid



80% AMR efficiency at approx. 100 grids per core

D. Duplyakin , J. Brown, D. Calhoun, "Applying Active Learning to Adaptive Mesh Refinement Simulations", (submitted) IEEE (2017)