ForestClaw : Mapped, multiblock adaptive quadtrees

Donna Calhoun (Boise State University)

Carsten Burstedde (Univ. of Bonn)

HPC^3

November 9-13, 2014

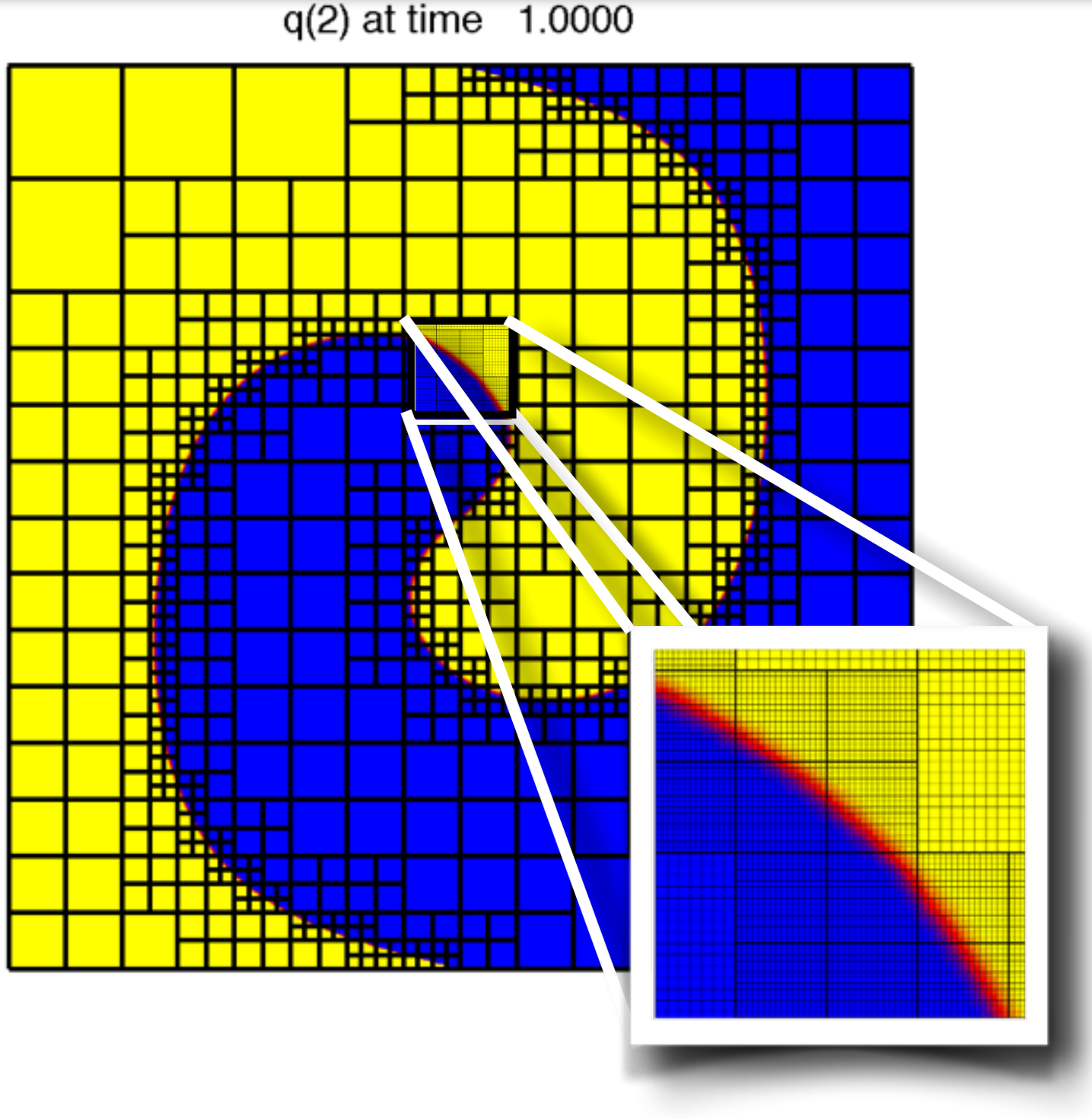
KAUST - Saudi Arabia

The ForestClaw Project

- Project to develop highly scalable adaptive mesh refinement (AMR) code that combines quadtree data structures with patch based refinement
- Uses the p4est (Carsten Burstedde, Univ. of Bonn) dynamic grid management library
- Explicit solver algorithms are those described by Berger-Oliger in 1984. Implicit solvers to be added.

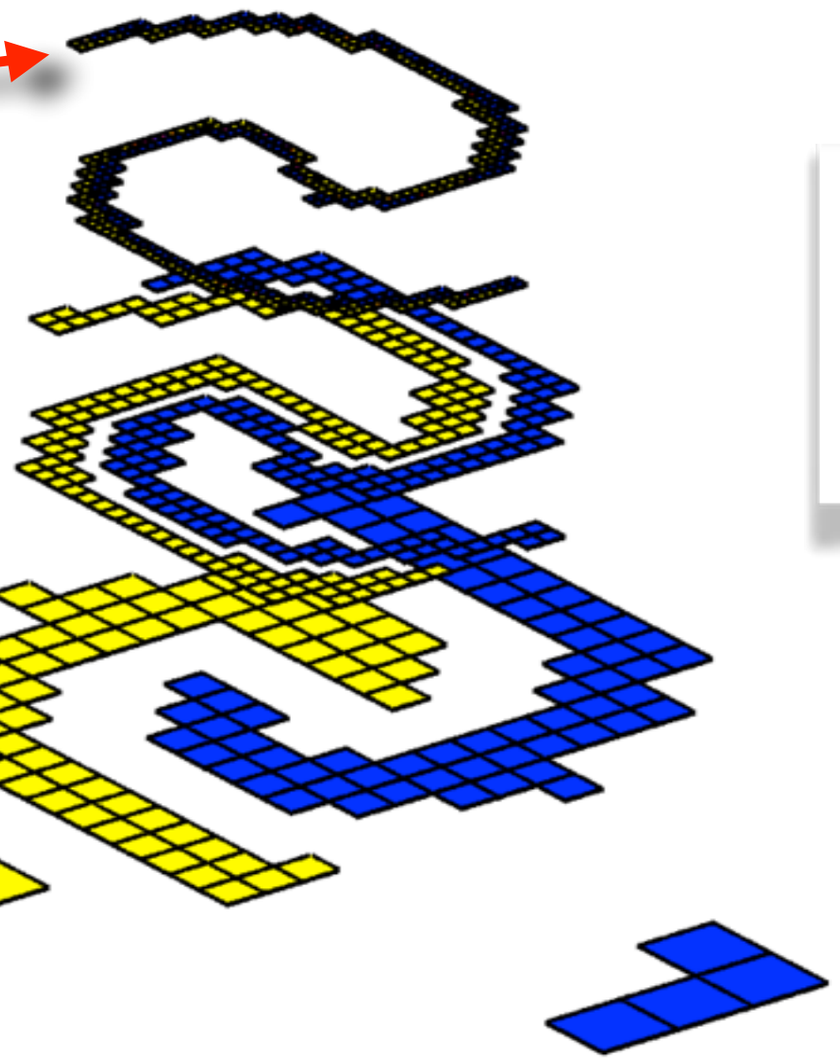
Design goal : Develop an easily accessible, highly scalable AMR framework to spur development of novel numerical methods and applications for AMR.

AMR using ForestClaw



AMR using ForestClaw

Fine level grids



Each grid has a layer of ghost cells (not shown) to facilitate communication

Coarse level grids

Quadtree based refinement

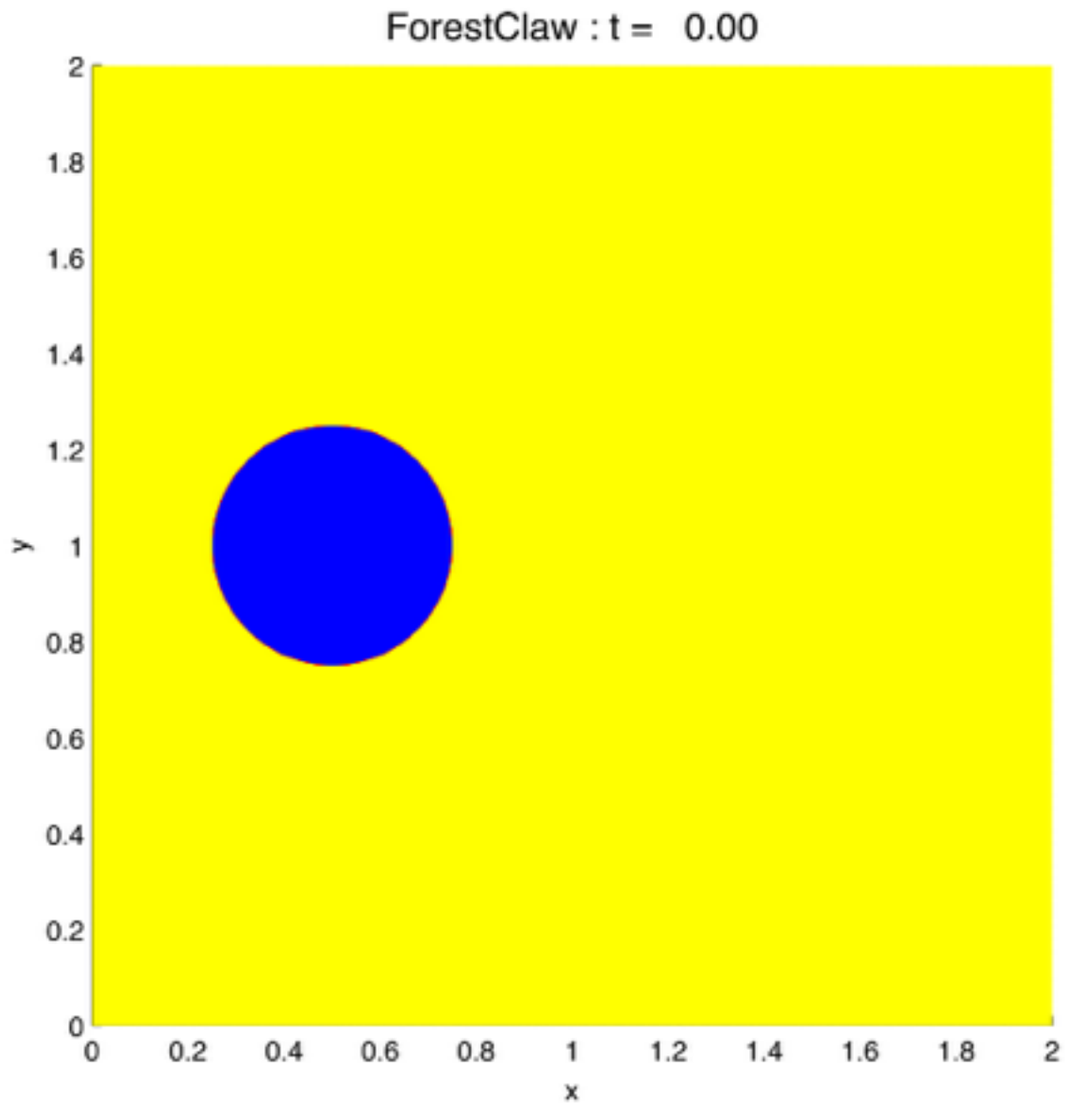
Progress on ForestClaw

ForestClaw started at [HPC]^3 2012.

Progress to date

- Multi-rate time stepping (Fall, 2012)
- General mappings available (Fall, 2012)
- MPI parallel capabilities implemented (Summer, 2013)
- Build system using GNU Autotools (Summer, 2014)
- Mapped, multiblock capabilities for several useful domains, including cubed disk and sphere (Summer, 2014)
- Periodic domains (Fall 2014)

Gridding strategies

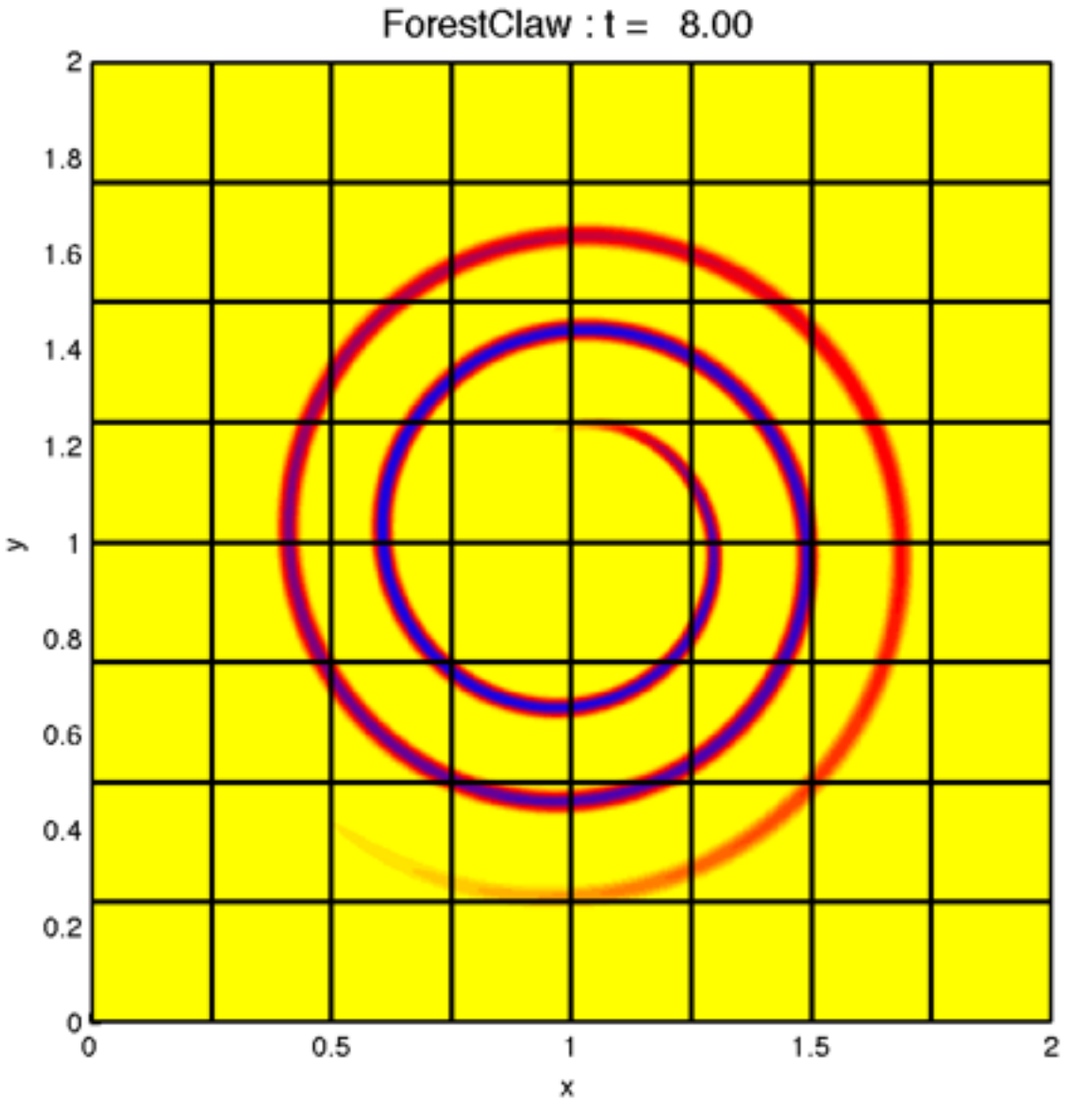


Compare patch-based vs. quad tree on this solution

Filament example

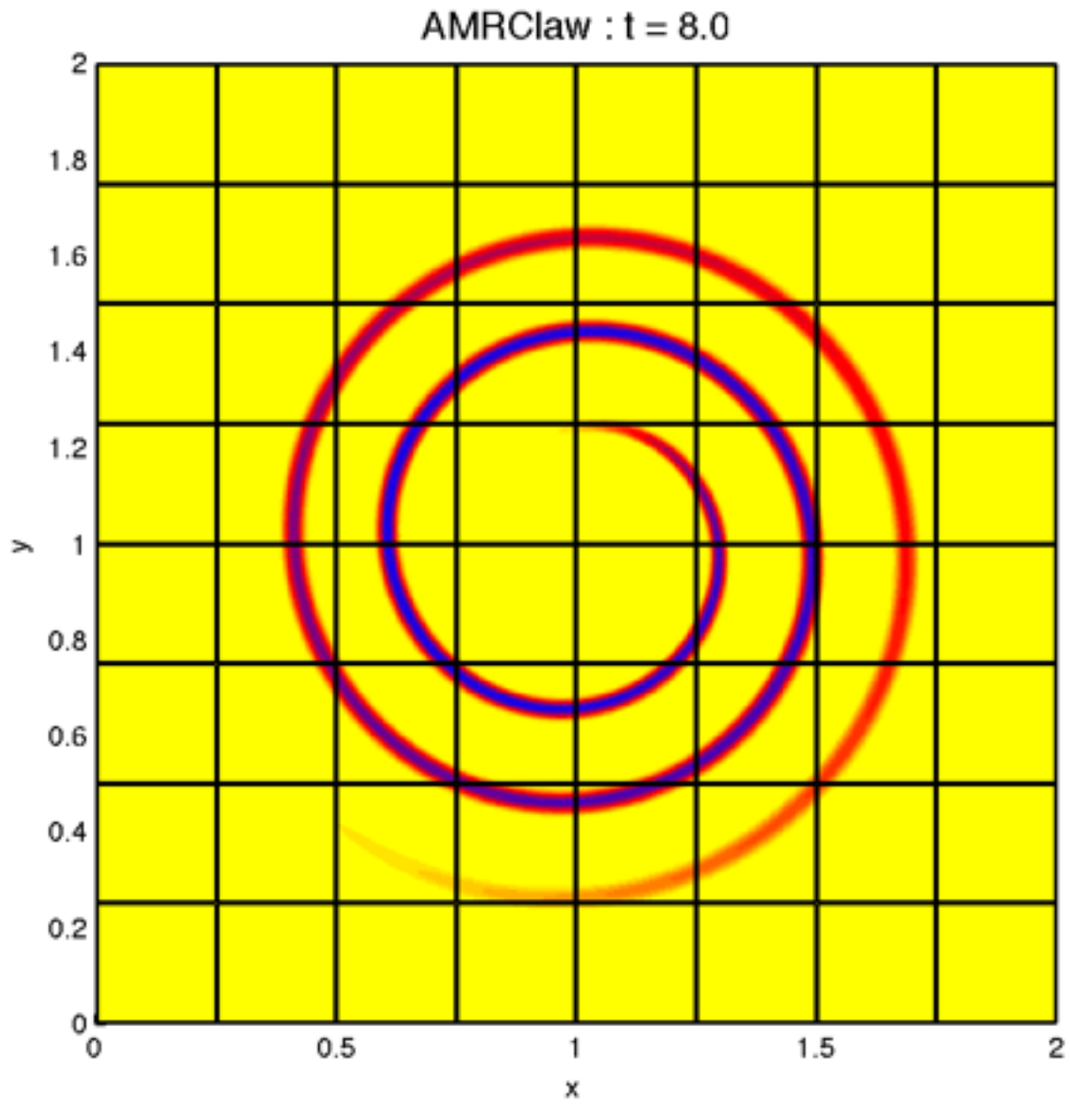
- Solve $q_t + \mathbf{u}\nabla q = 0$, $\mathbf{u} = \nabla \times \psi$, $\psi(x, y) = (4/3)r^3$.
- *Domain* : $[0, 2] \times [0, 2]$
- Fixed $\Delta t = 4 \times 10^{-3}$ for $\Delta x = 1/16$ (CFL approx. 0.74).
- Run to $T = 8.0$
- Refine to an effective resolution of 512×512 and 1024×1024.
- Use multirate time stepping
- Regrid at least every coarse grid time step
- Use MC limiter

Filament example (ForestClaw)



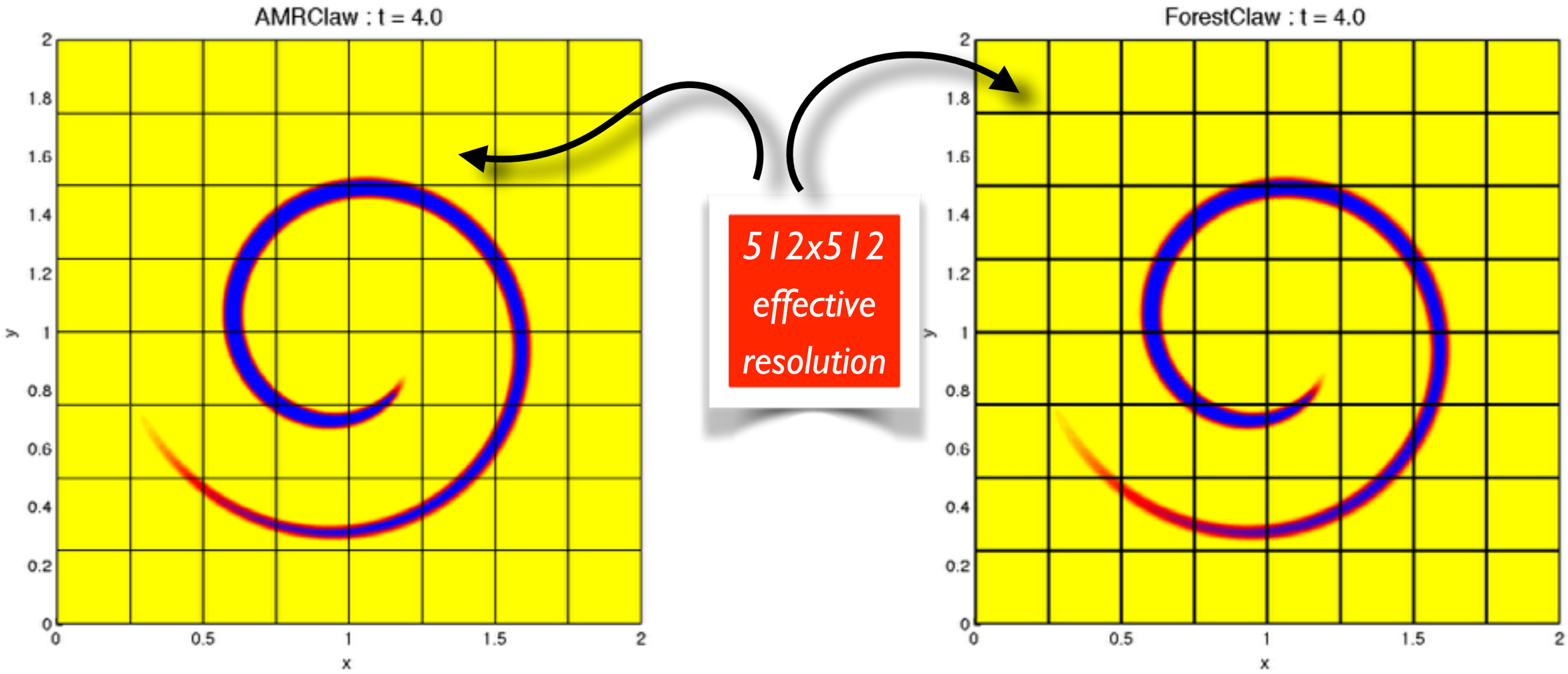
512x512 grid (mx=64)

Filament example (AMRClaw)



mx = 512 (maxld=80)

Gridding strategies (uniform)



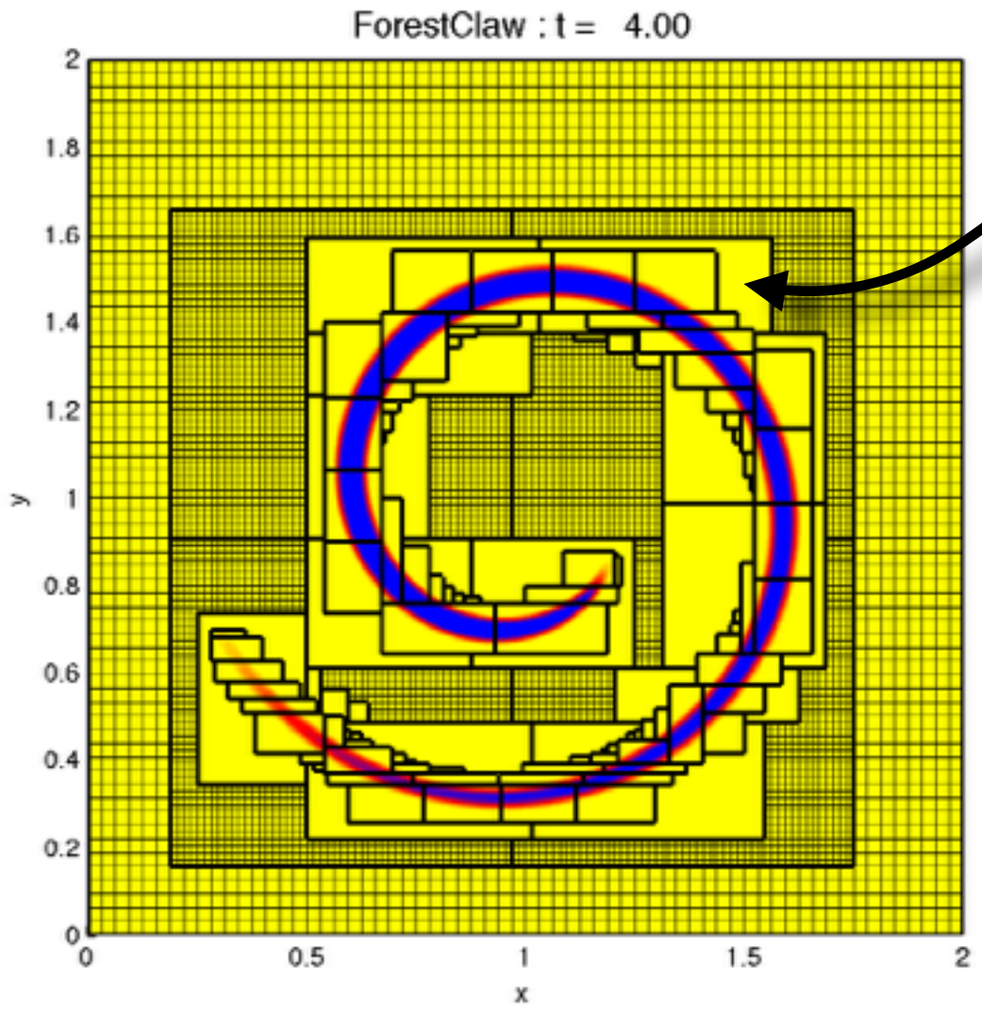
Levels 0 (maxld=80)

1751s

Levels 3 (mx=64)

1671s

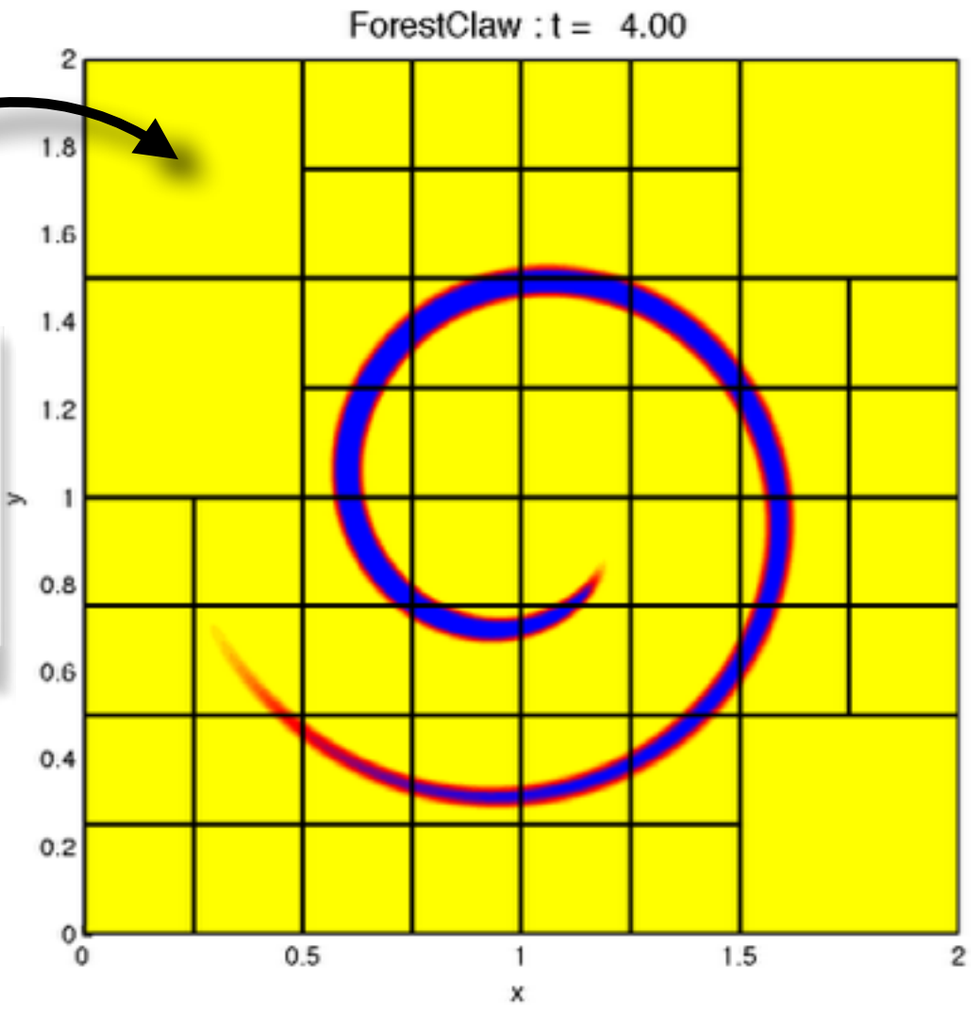
Gridding strategies (512)



Levels 0-3 (max1d=80)

617s

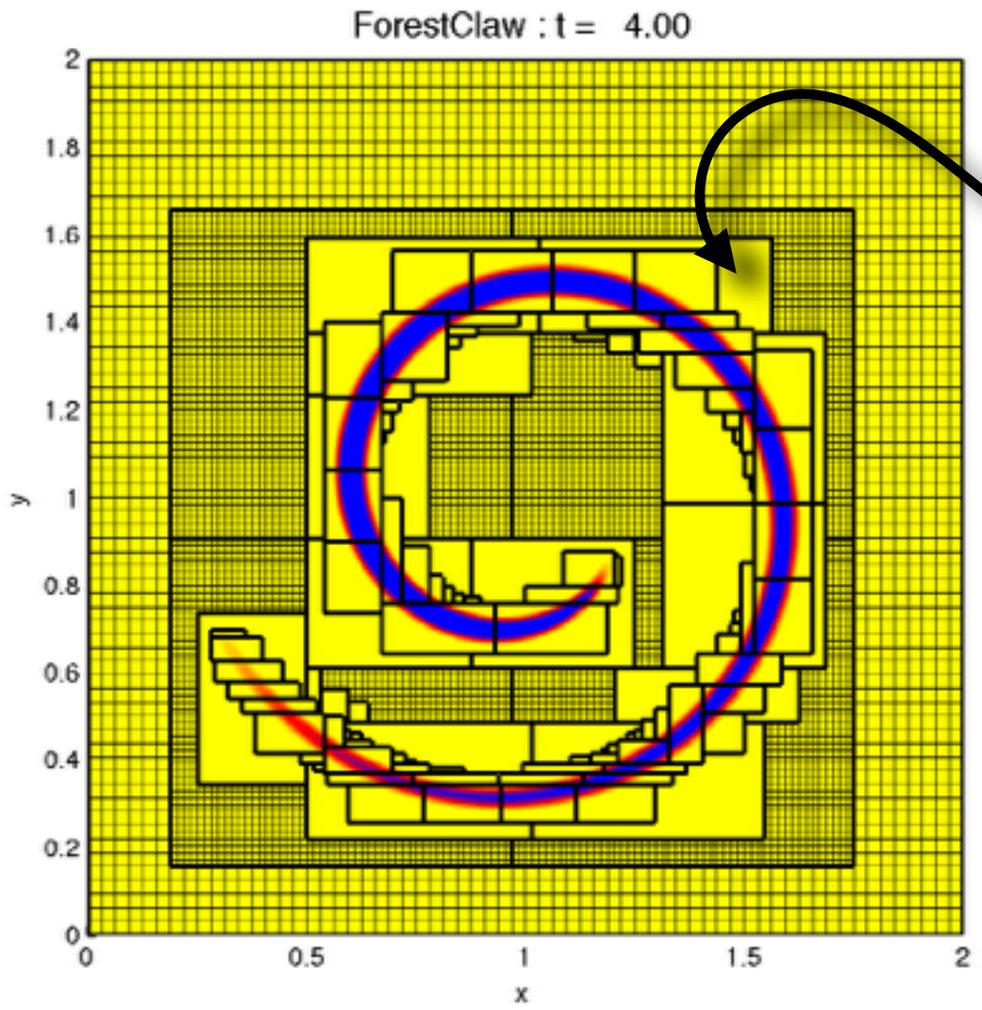
256x256 effective resolution



Levels 2-3 (mx=64)

1178s

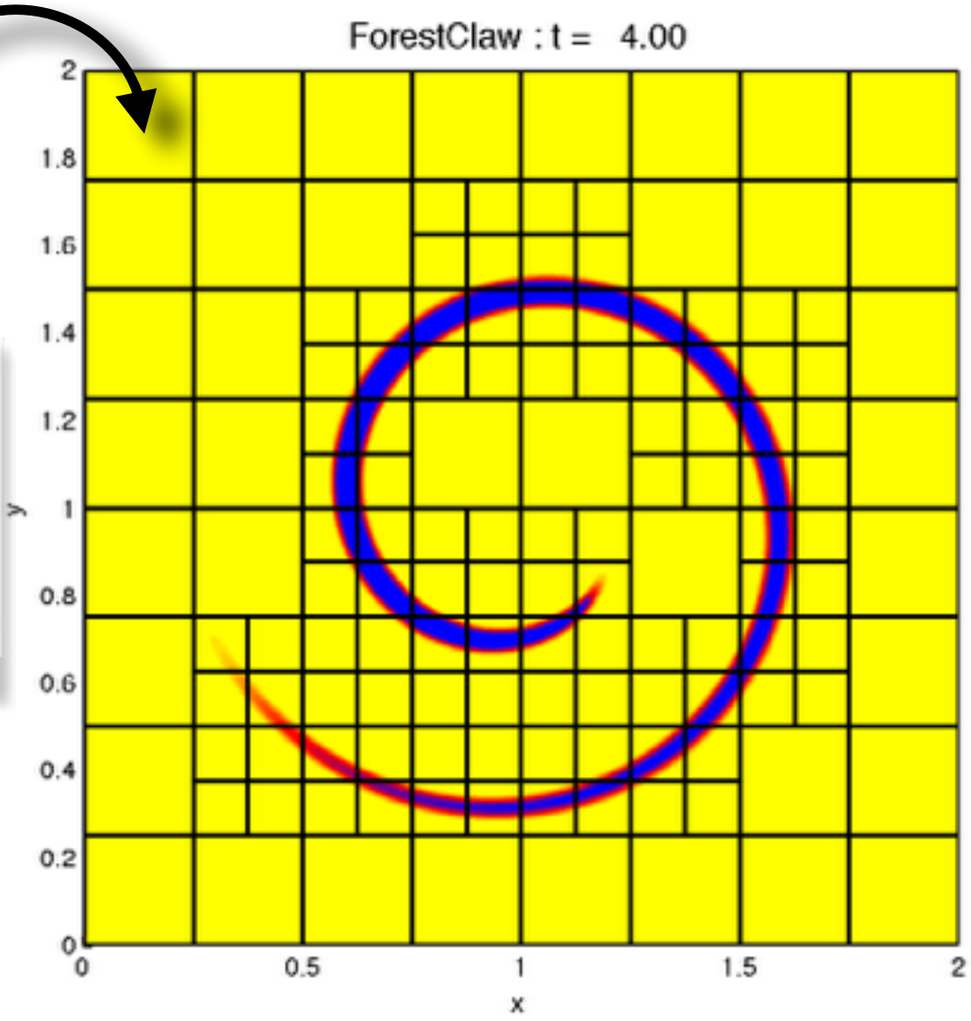
Gridding strategies (512)



Levels 0-3 (maxld=80)

617s

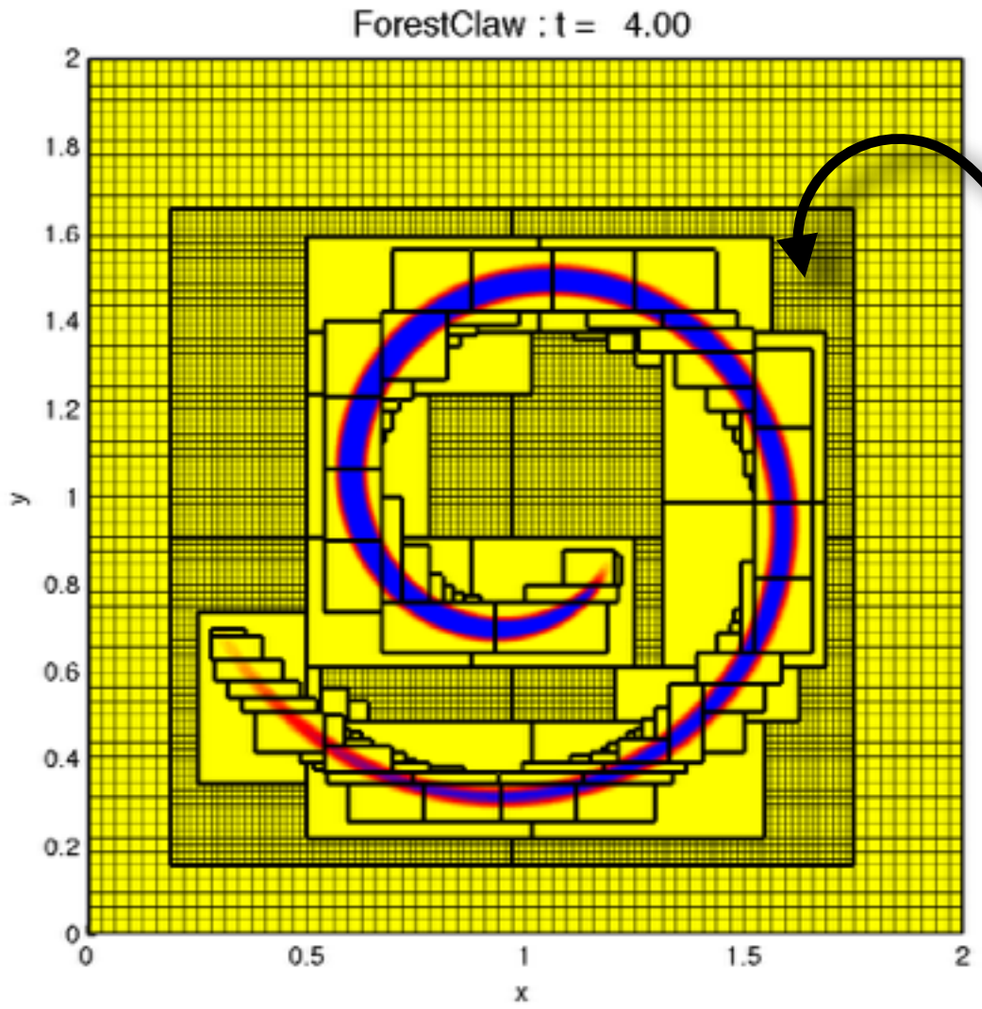
256x256 effective resolution



Levels 3-4 (mx=32)

904s

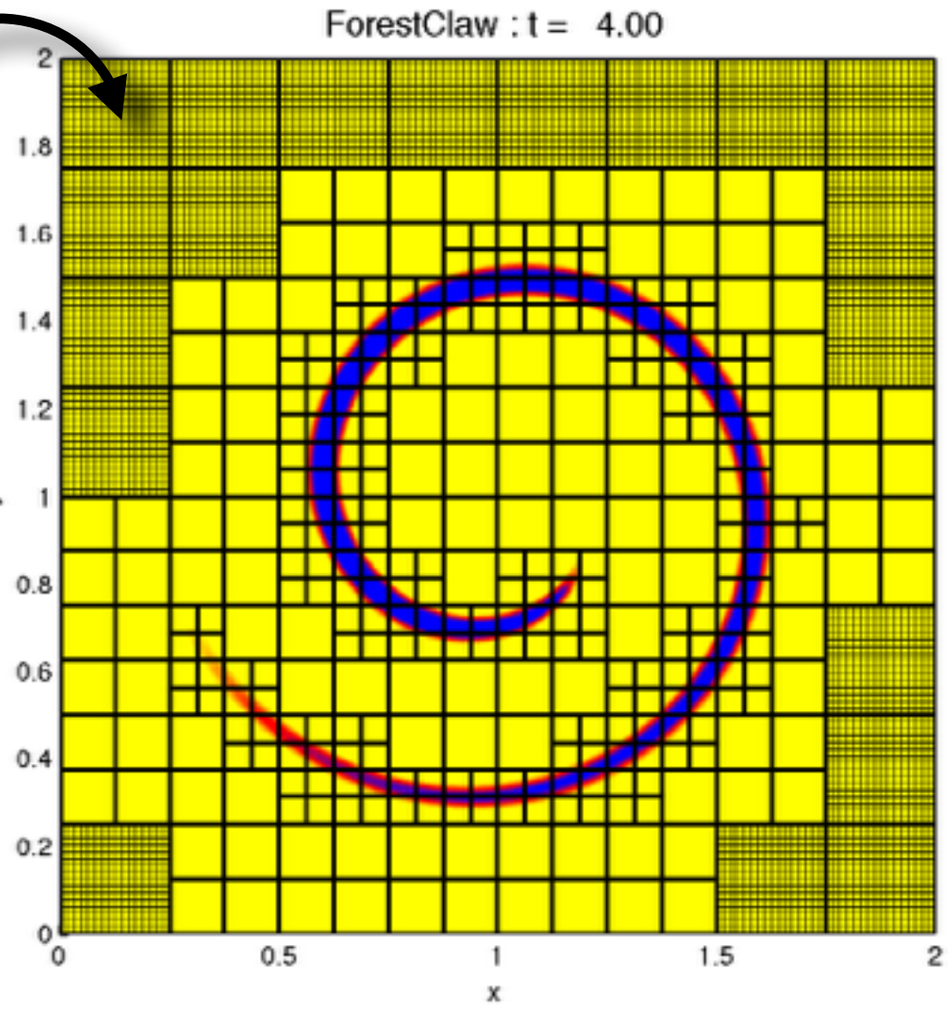
Gridding strategies (512)



Levels 0-3 (max1d=80)

617s

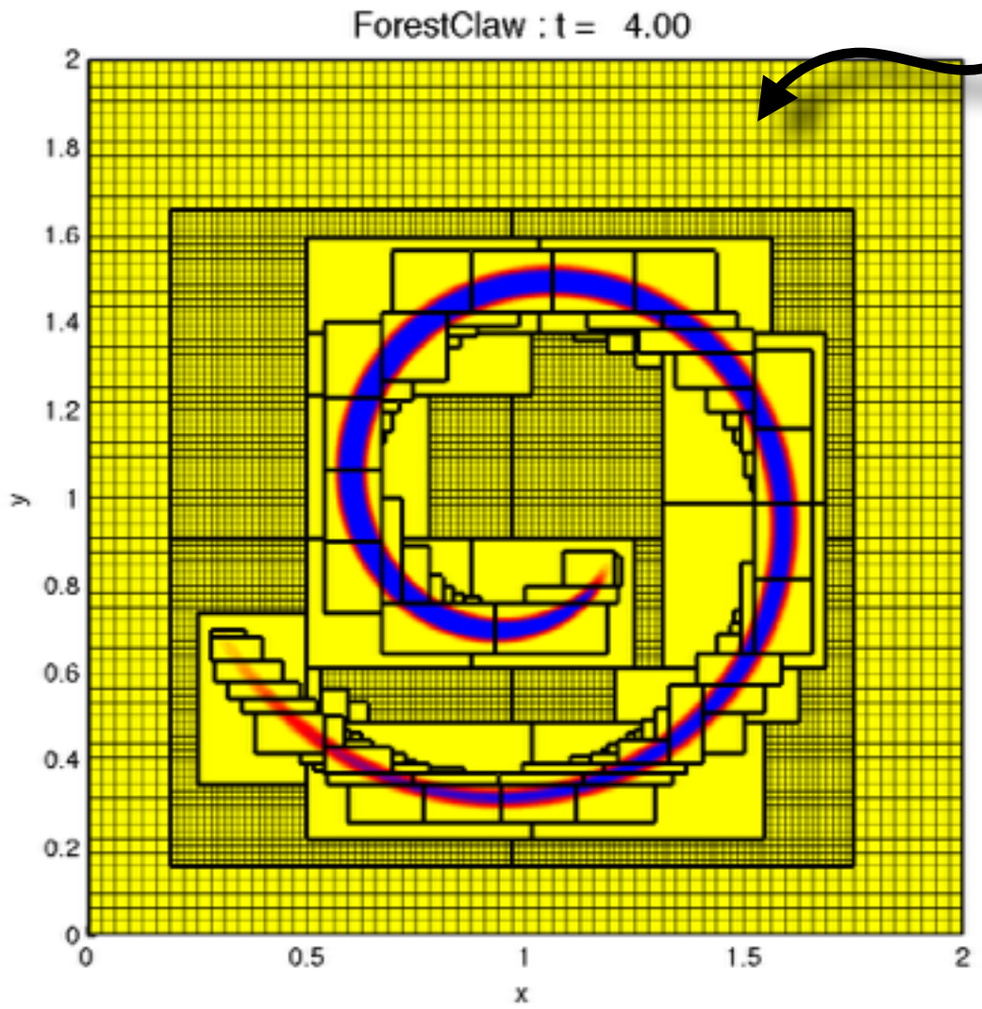
128x128 effective resolution



Levels 3-5 (mx=16)

777s

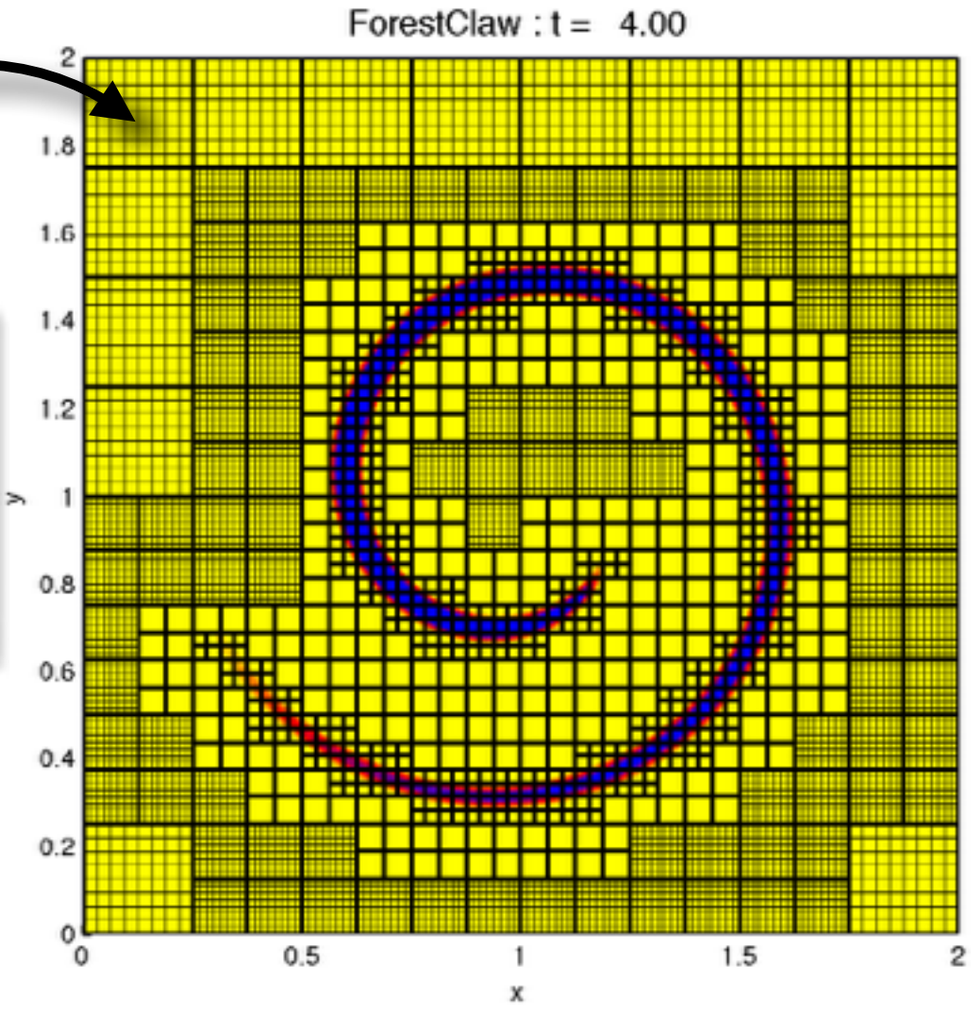
Gridding strategies (512)



Levels 0-3 (max1d=80)

617s

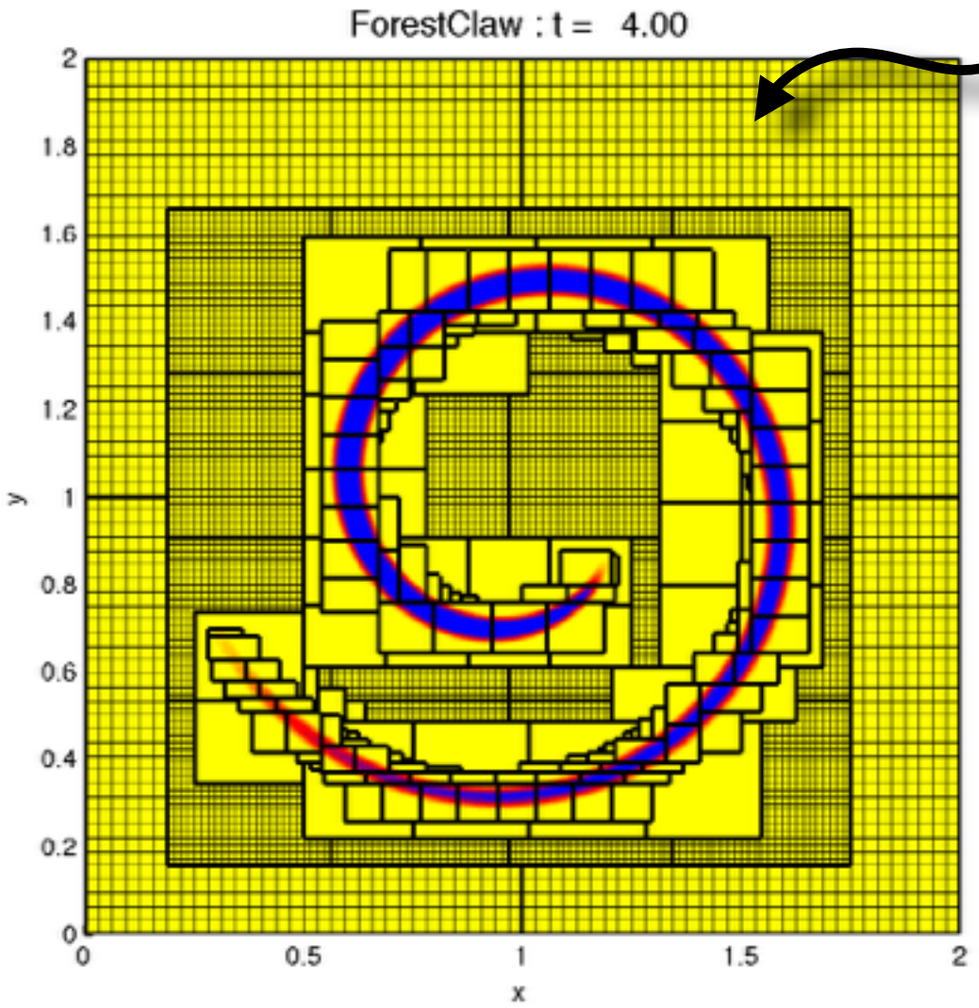
64x64 effective resolution



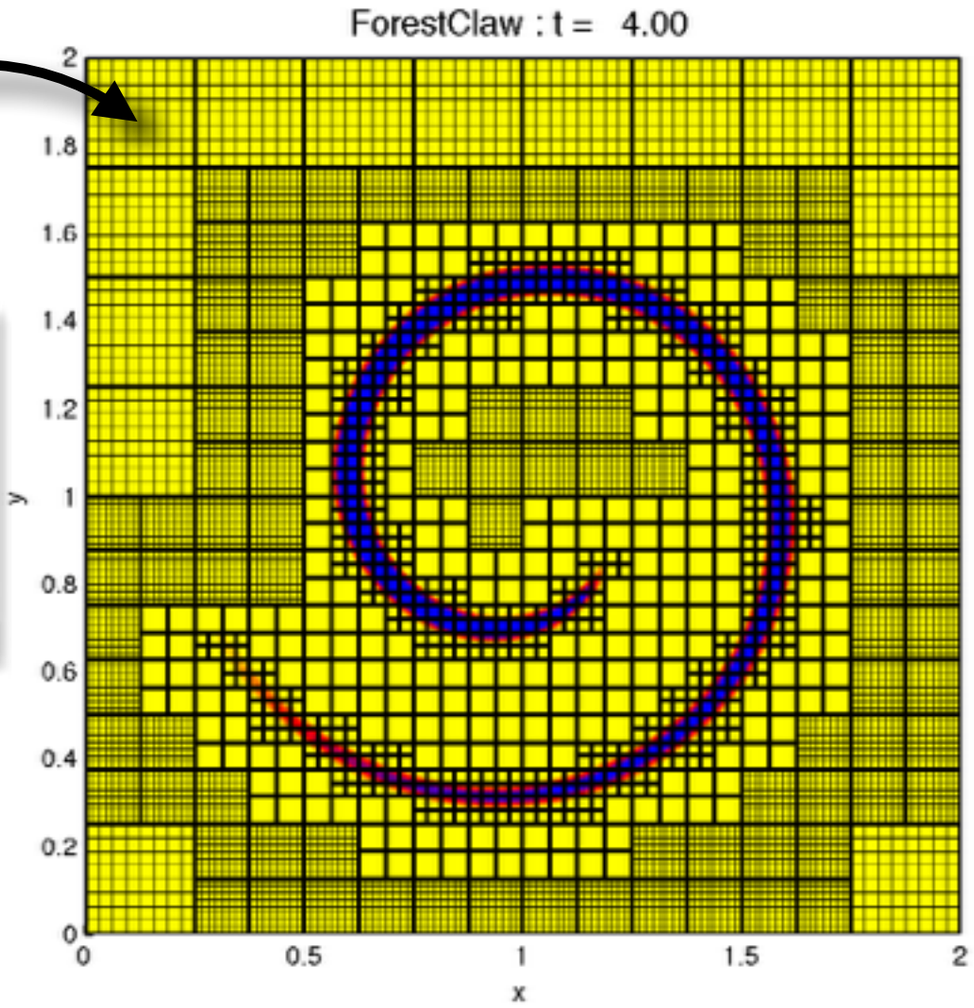
Levels 3-6 (mx=8)

811s

Gridding strategies (512)



64x64 effective resolution



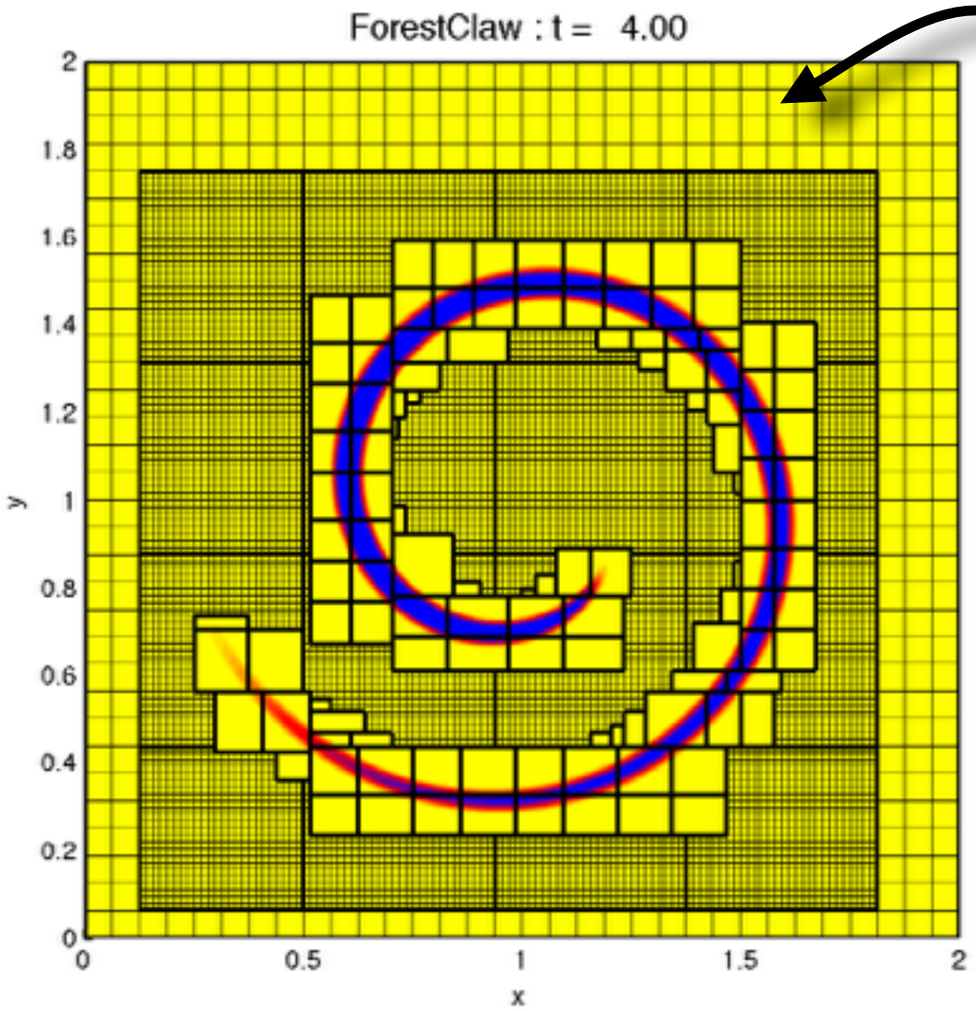
Levels 0-3 (maxld=40)

720s

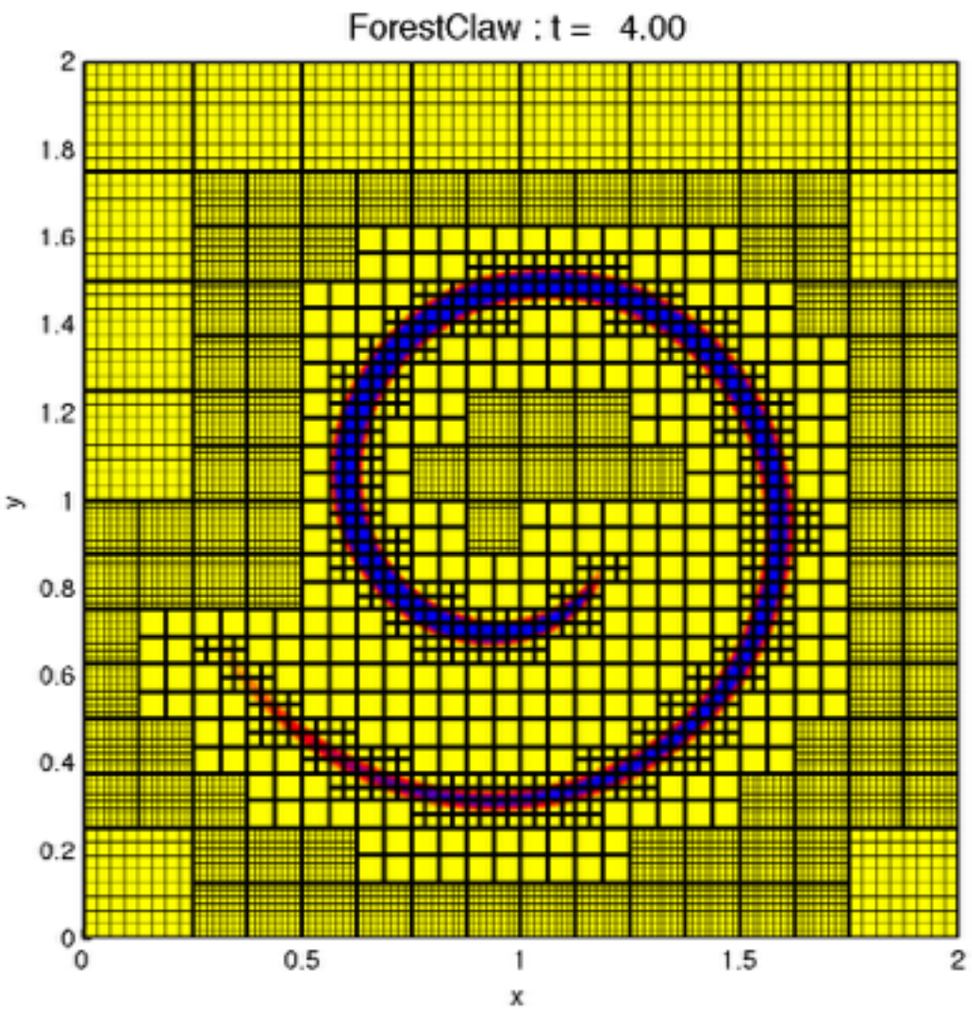
Levels 3-6 (mx=8)

811s

Gridding strategies (512)



32x32 coarsest level grid



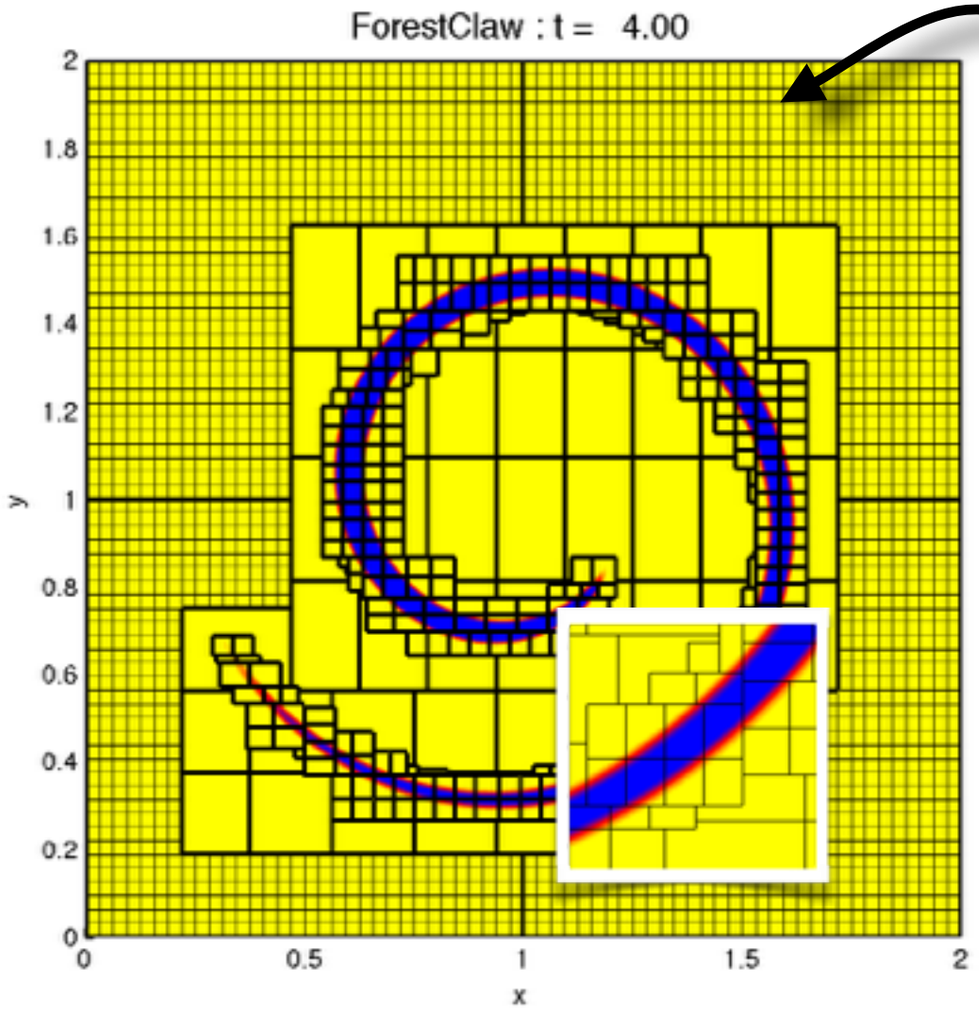
Levels 0-2 (maxld=40; factor 4)

636s

Levels 3-6 (mx=8)

811s

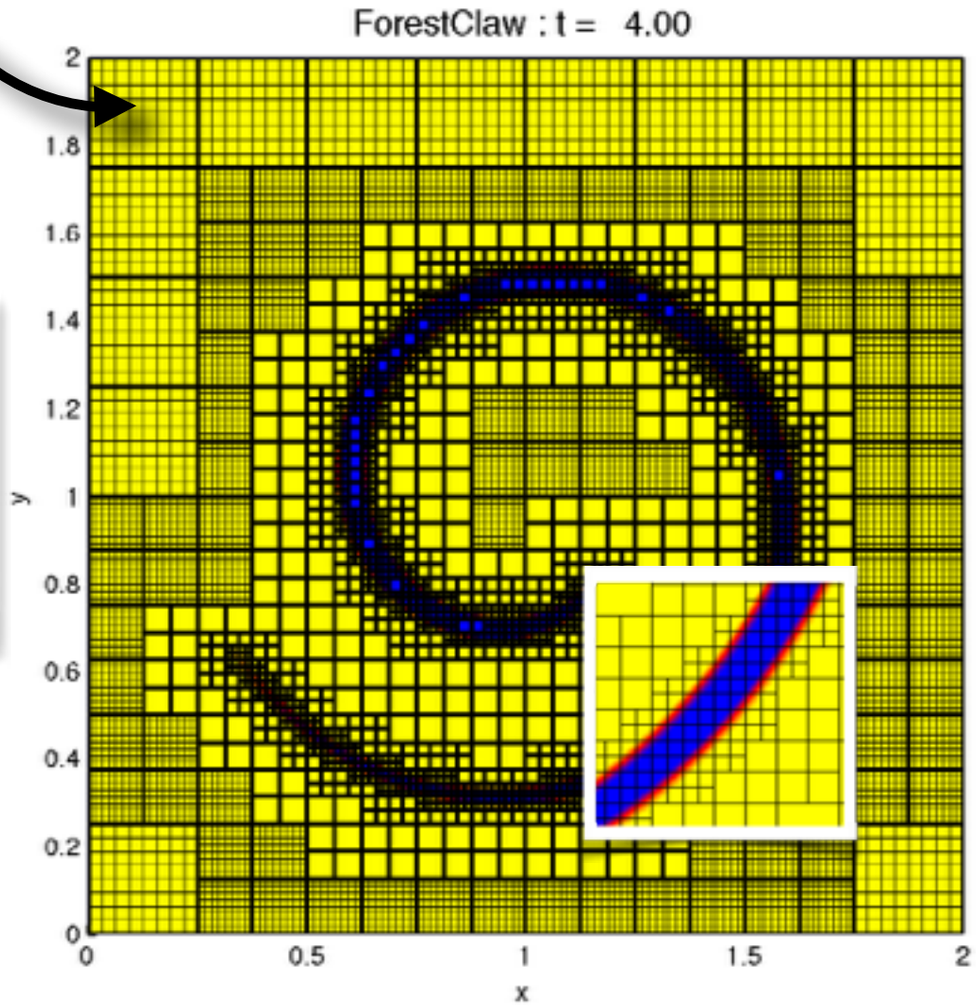
Gridding strategies (1024)



Levels 0-2 (R Factor : 4,4)

3212s

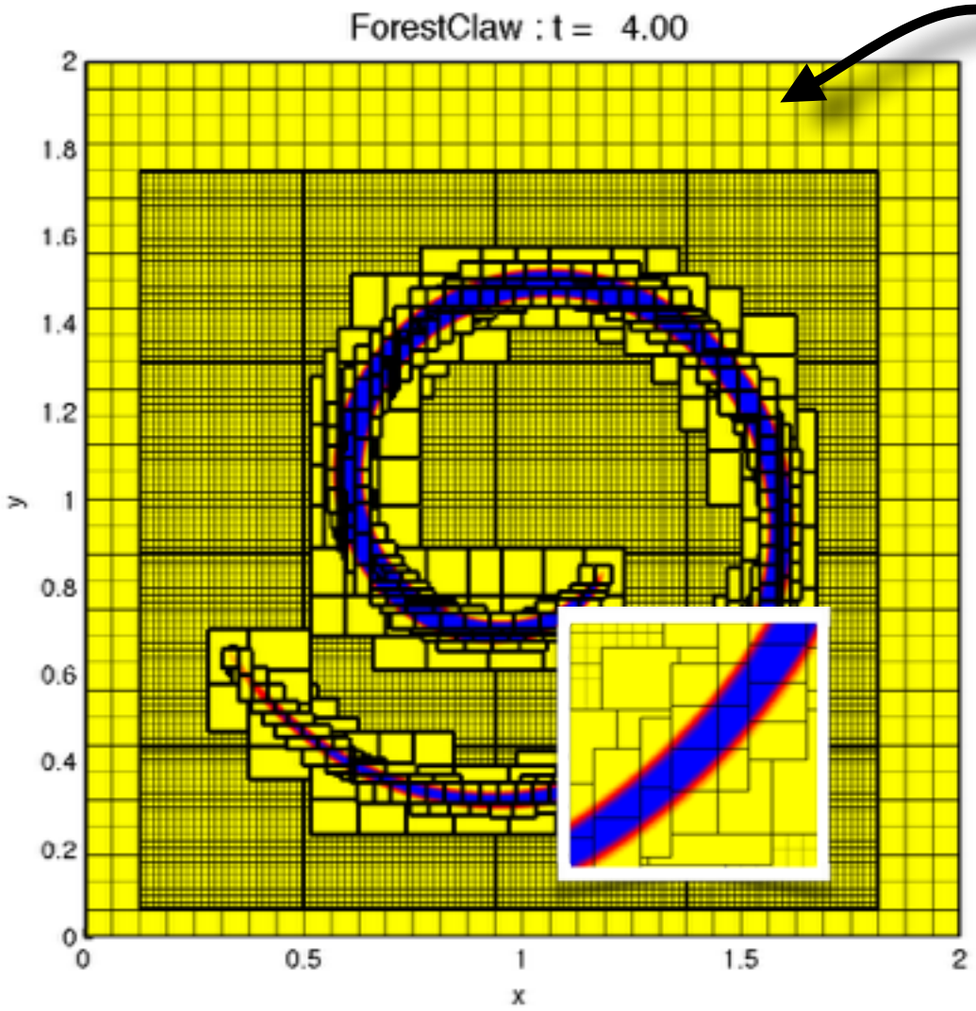
64x64 effective resolution



Levels 3-7 (mx=8)

3567s

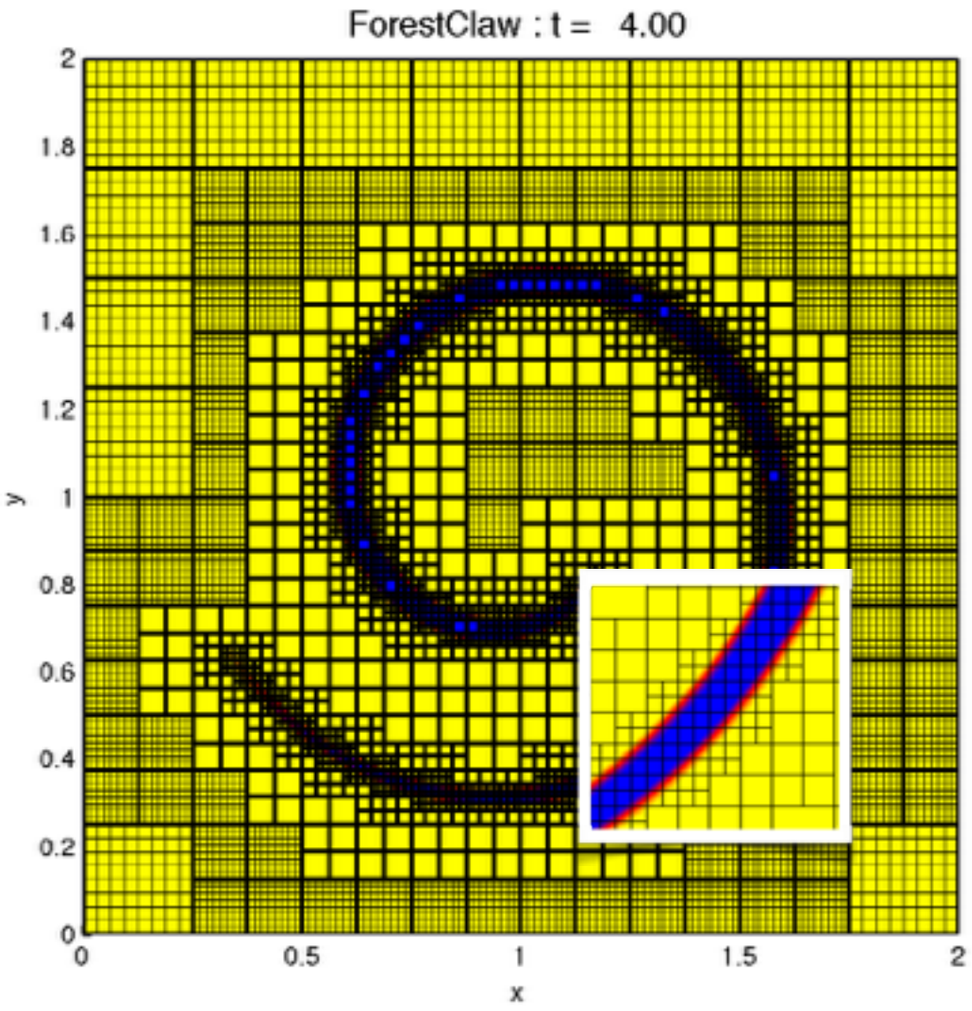
Gridding strategies



Levels 0-3 (R Factor : 4,4,2)

3891s

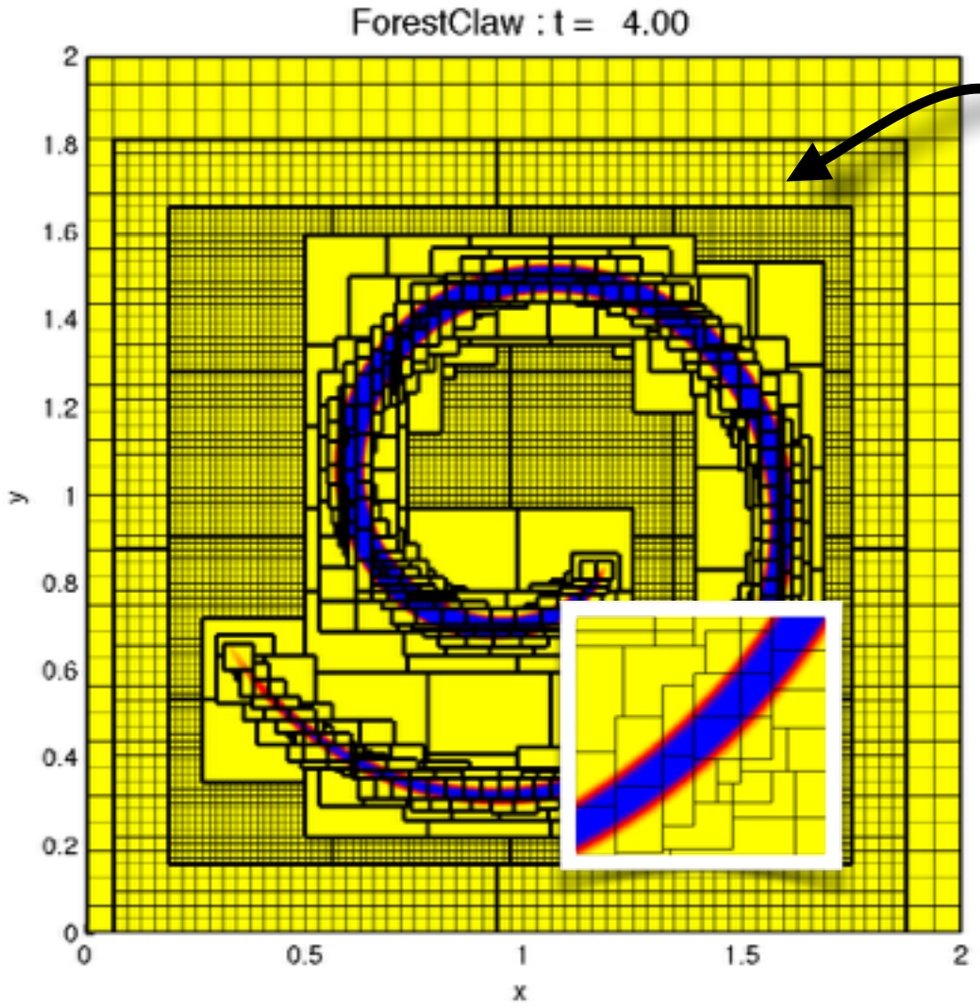
32x32 coarsest level grid



Levels 3-7 (mx=8)

3567s

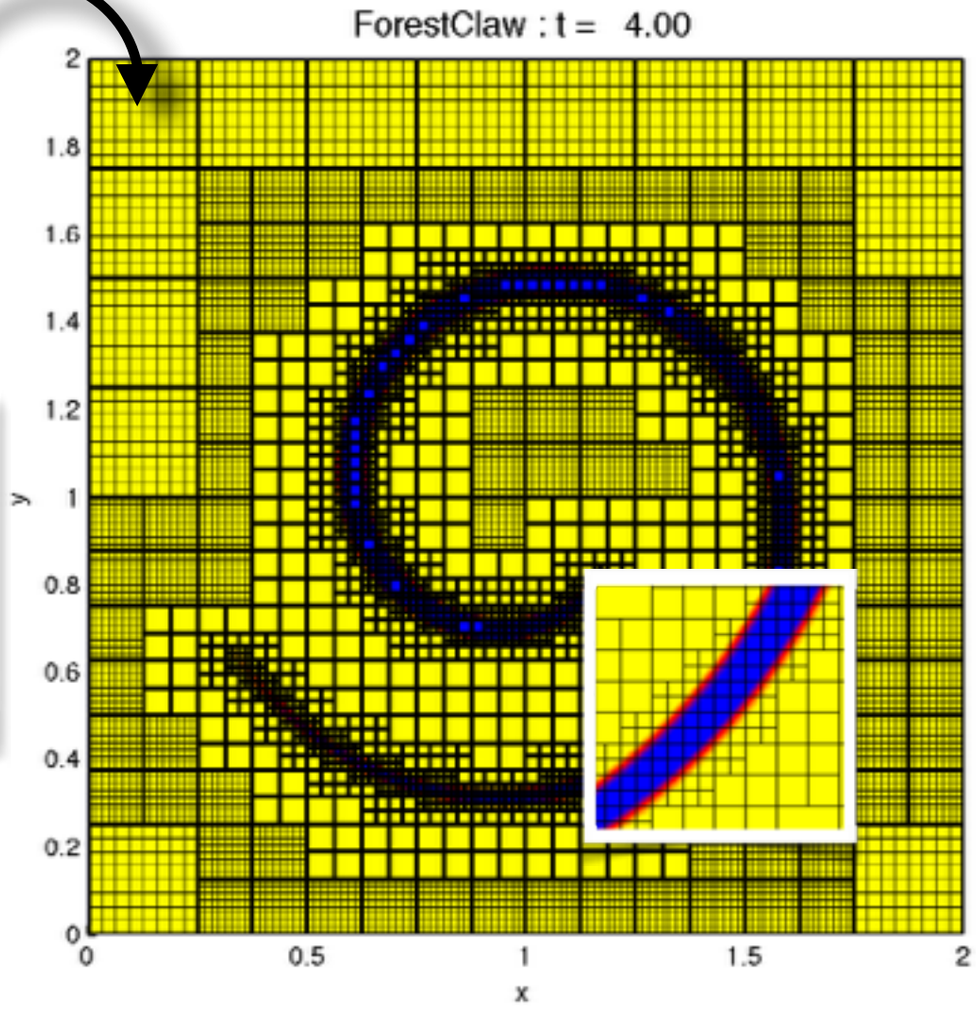
Gridding strategies



Levels 0-5 (R Factor : 2)

4060s

64x64 effective resolution



Levels 3-7 (mx=8)

3567s

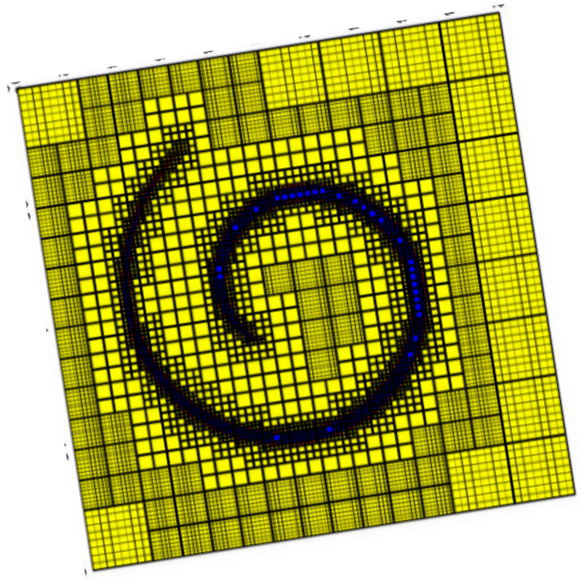
Summary

	AMRClaw			ForestClaw				
	Total	**Regrid**		**Total**	**Advance**	**Exchange**	**Regrid**	
512x512 grid : Uniform refinement (ForestClaw 5% faster)								
Example 0	**1751**		*max1d=80*	**1671**				*mx = 64*
512x512 effective resolution (AMRClaw about 12% faster)								
Example 1	617	19.8%	*max1d=80*	1178	96.8%	1.5%	1.7%	*mx = 64*
Example 2				904	90.0%	3.8%	5.6%	*mx = 32*
Example 3				777	83.0%	8.6%	7.6%	*mx = 16*
Example 4				**811**	70.7%	17.5%	11.9%	*mx = 8*
Example 5	**720**	21.9%	*max1d=40*					
Example 6	636	7.5%	*max1d=40; RF=4*					
1024x1024 effective resolution (ForestClaw and AMRClaw are within about 15% of each other)								
Example 7	3212	8.4%	*max1d=40; RF=4*	3567	77.7%	18.7%	3.6%	*mx=8*
Example 8	3891	24.9%	*max1d=40; RF=4,4,2*					
Example 9	4060	27.1%	*max1d=40; RF=2*					

AMRClaw prefers larger refinement ratio; ForestClaw prefers smaller grids

Timings all done on a desktop

Why use quadtrees?

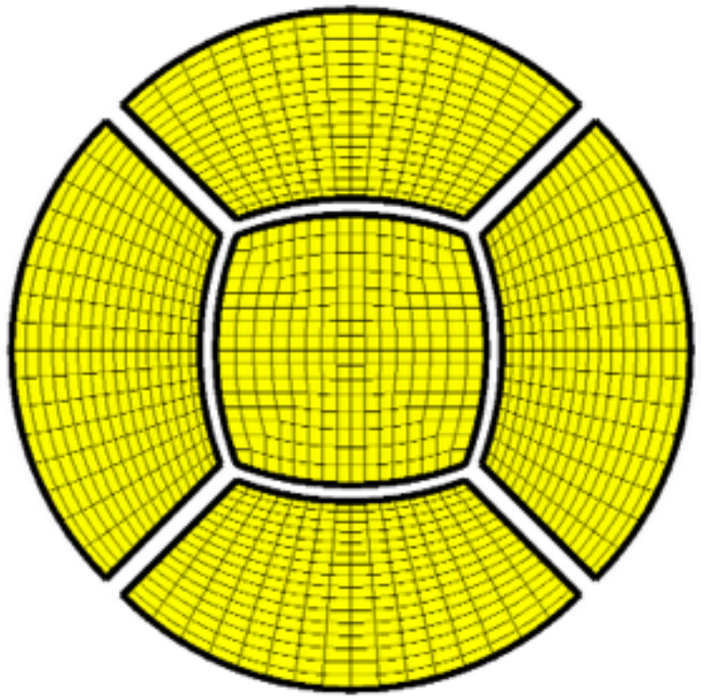


- Refinement patches and parallel “units” are the same
- Tree-based grid layout makes communication between patches and blocks much simpler, especially in 3d.
- Has the aesthetic appeal of a quad/octree refinement
- Make use of existing literature on make quad/octrees efficient and scalable.

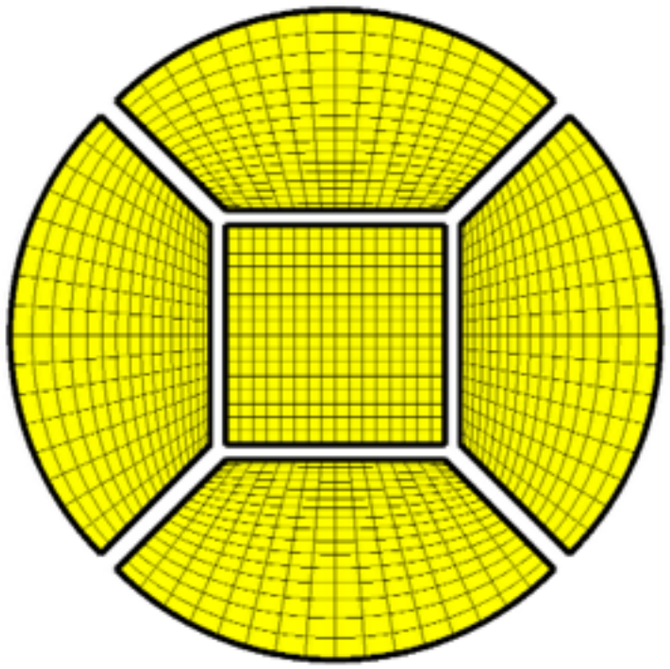
What next?

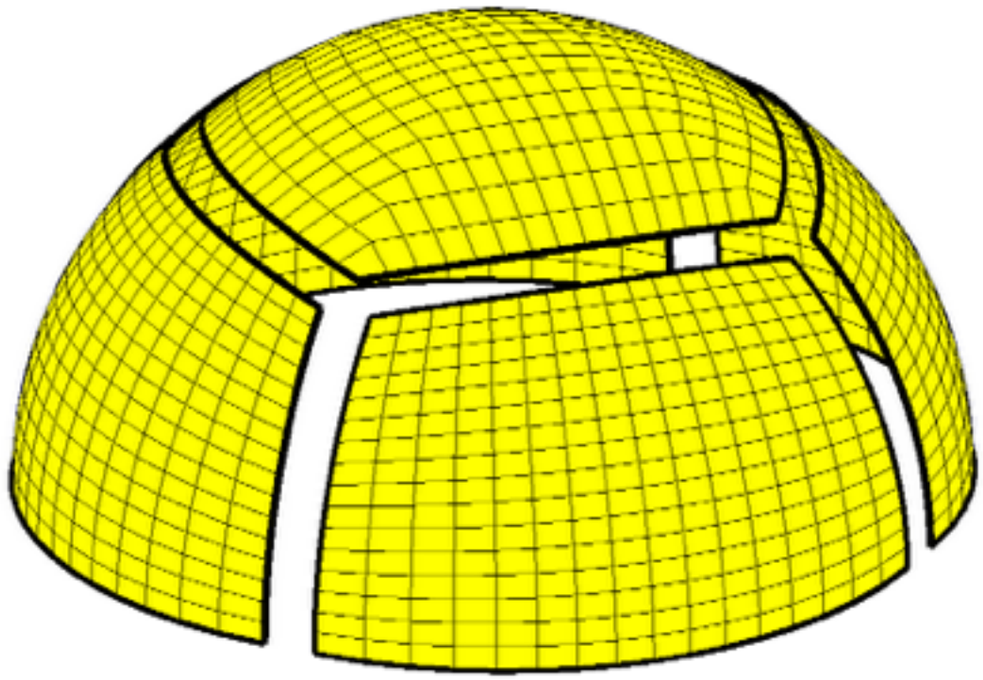
- Grid exchanges and communication
- Grid mappings and multiblock
- Multi-rate time stepping

Multi-block disk

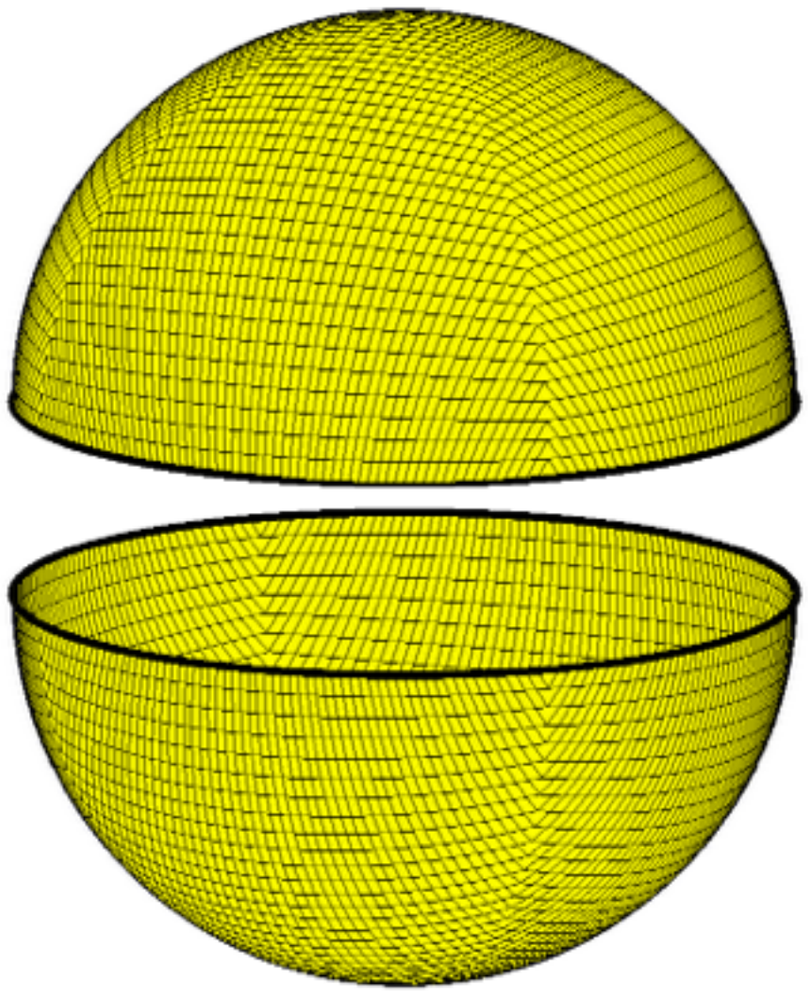


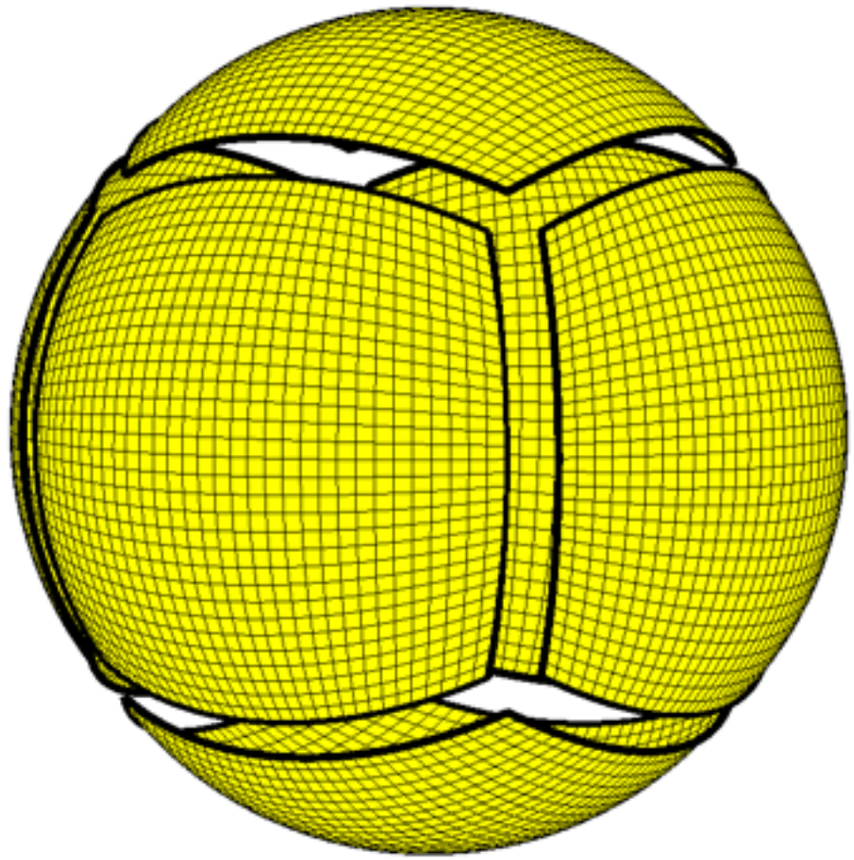
Multiblock capabilities are "inherited" from p4est.



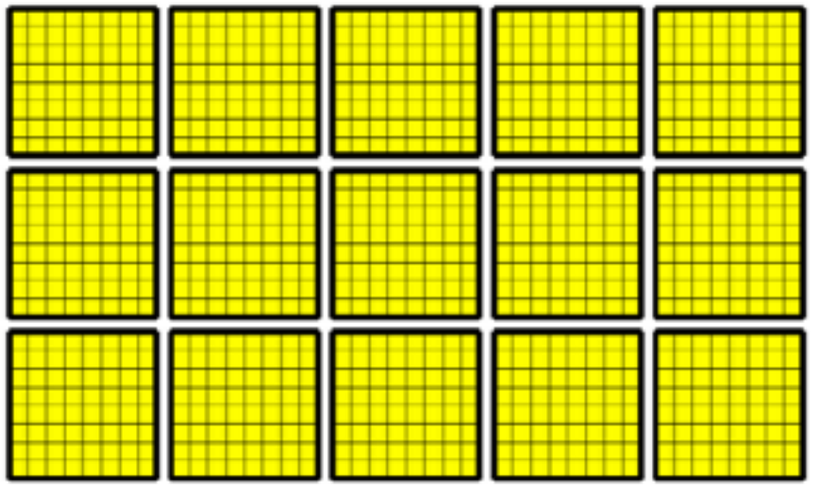


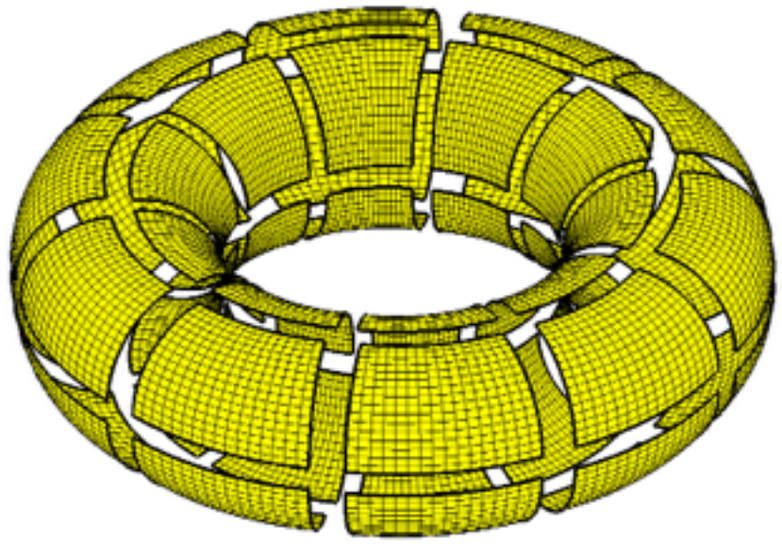
Sphere domains

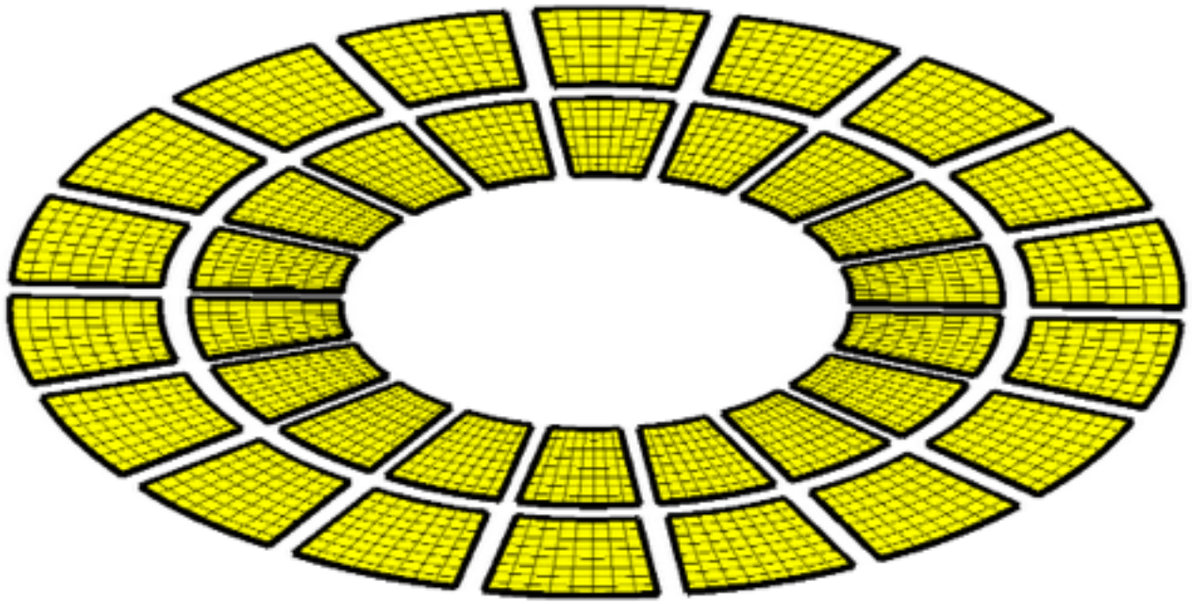


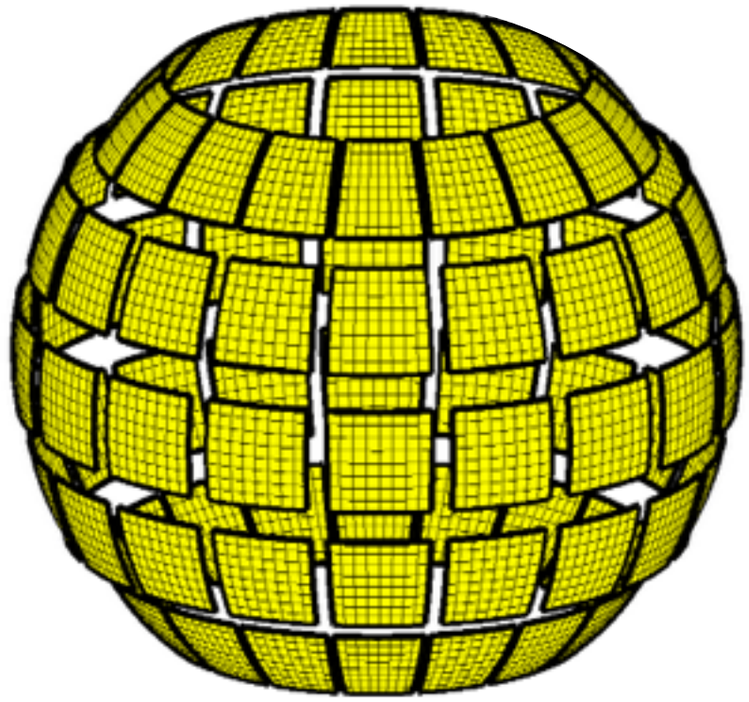


Periodic brick domains

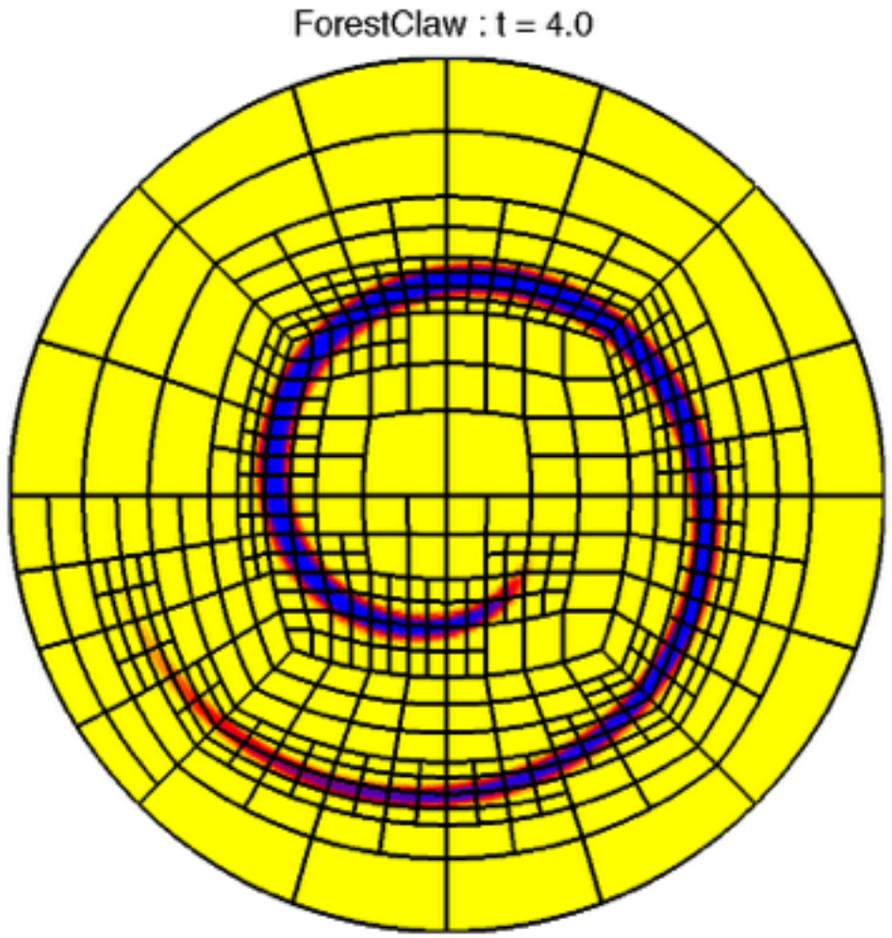


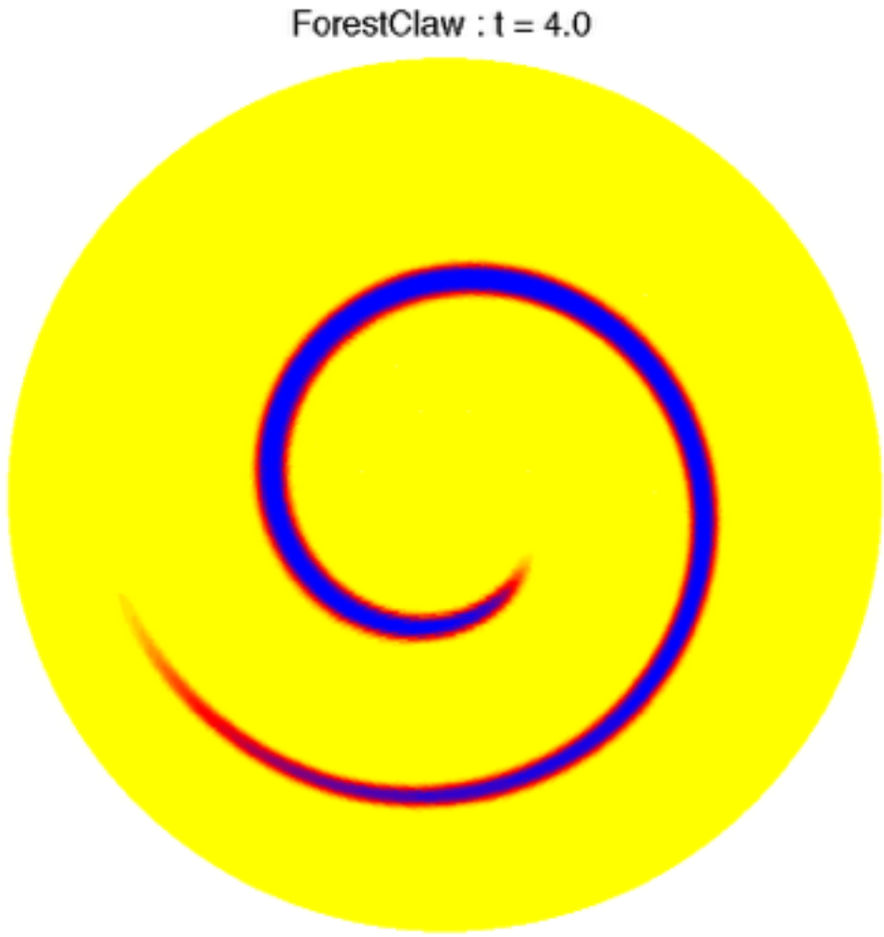






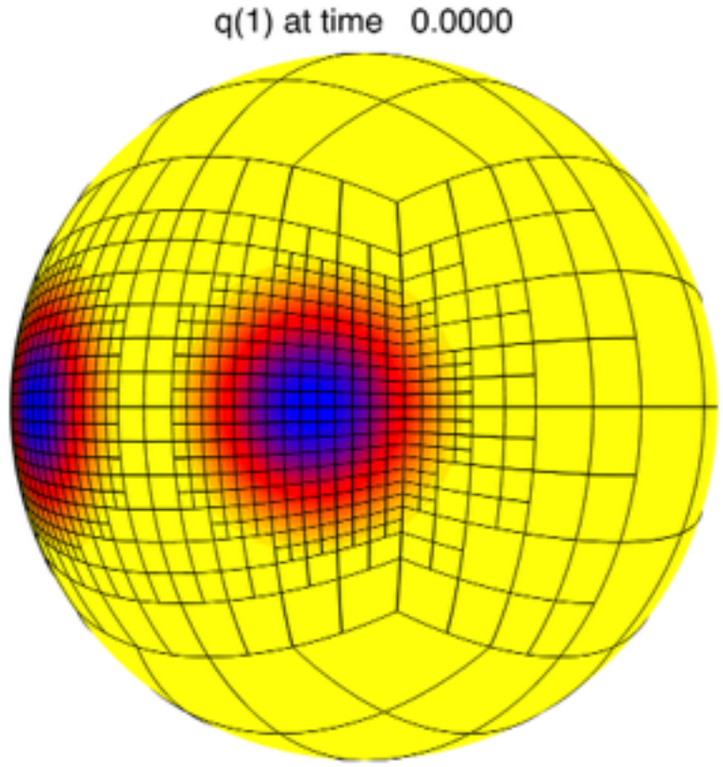
Multiblock disk





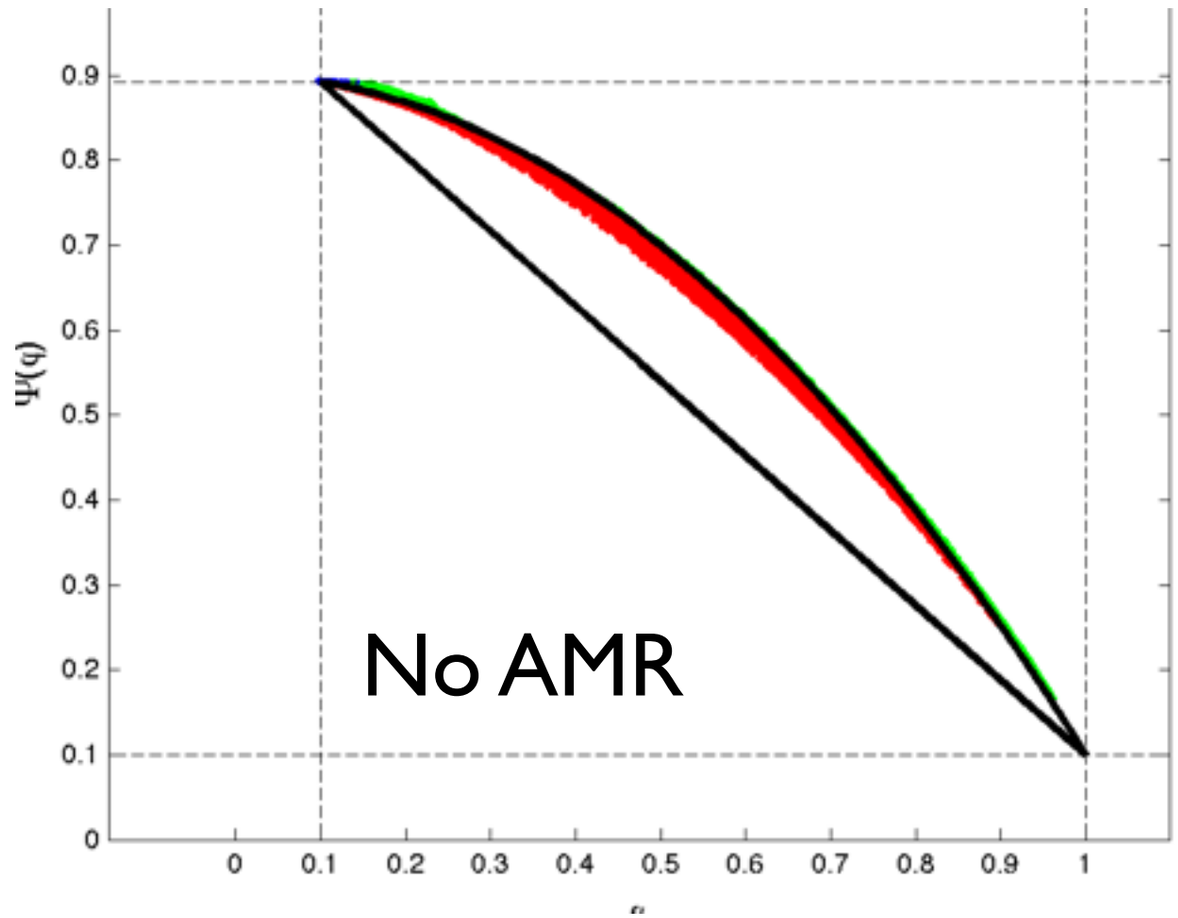
Mixing diagnostic

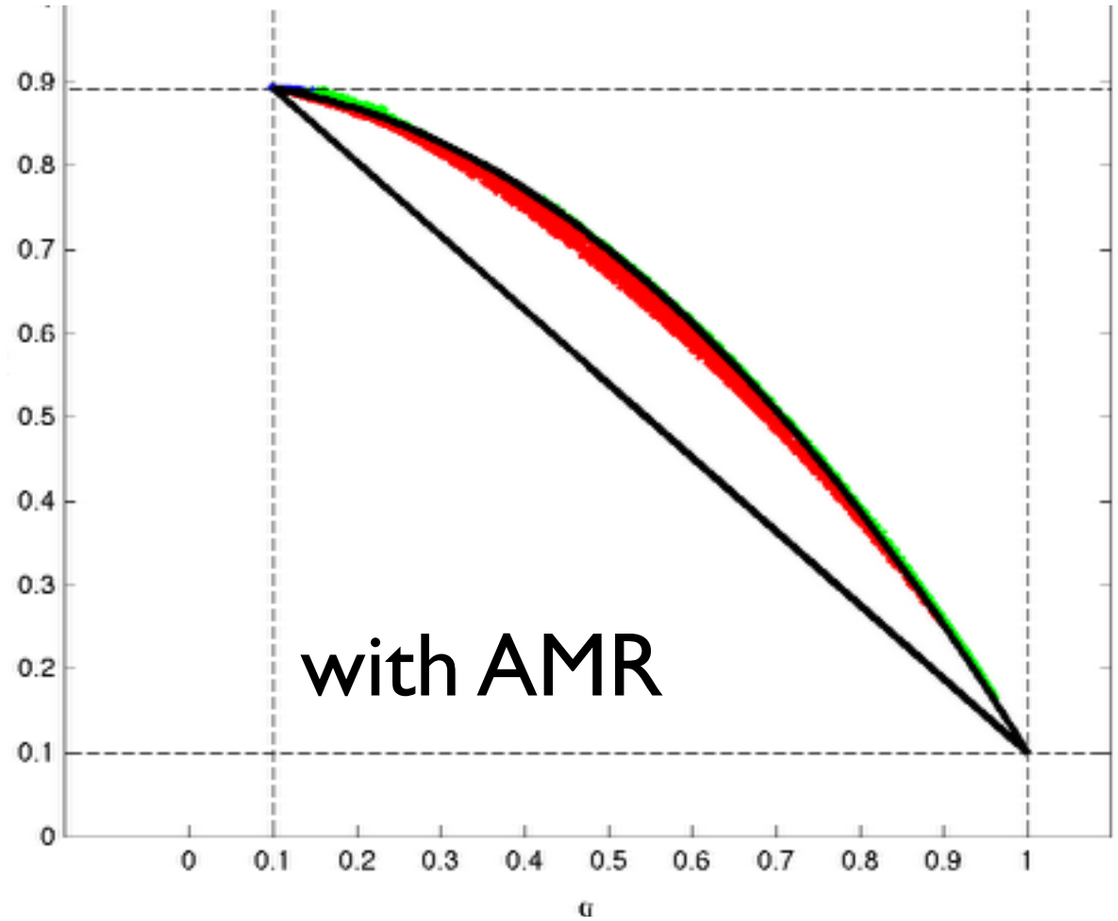
Preservation of functional relationship between tracers.



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

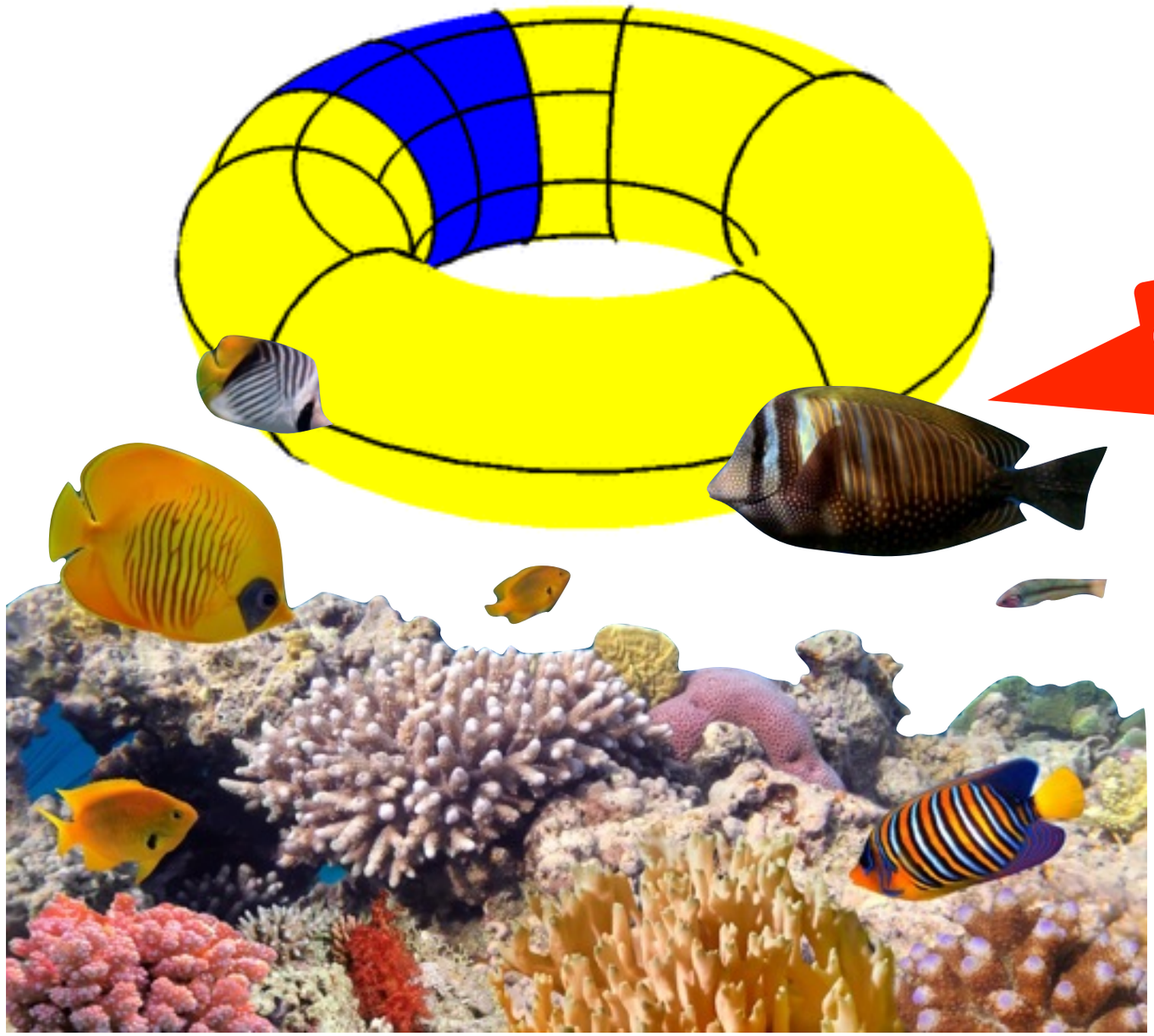
Mixing diagnostics (256 x 256)





	Diagnostic	Fraction	Diagnostic	Fraction
Real mixing (r)	5.45E-04	0.7565	1.68E-04	0.7709
Range preserving mixing (g)	1.49E-04	0.2070	4.2E-05	0.1925
Under and over shoots (b)	2.63E-05	0.0365	7.98E-06	0.0366

Flow on a torus



Handling multiblock boundaries

```
c       # Exchange at left neighbor
        if (idir .eq. 0) then
           do j = 1,my
              do ibc = 1,mbc
                 do mq = 1,meqn
                  if (iface .eq. 0) then
c                       # Left face
                        i1 = 1-ibc
                        j1 = j
                     elseif (iface .eq. 1) then
c                       # Right face
                        i1 = mx+ibc
                        j1 = j
                     endif
                     call fclaw2d_transform_face(i1,j1,i2,j2,
     &                    transform_ptr)
                     qthis(mq,i1,j1) = qneighbor(mq,i2,j2)
                 enddo
              enddo
           enddo
```

Loop over all ghost cells at left edge

This is the code used for every exchange at a left edge even if it is not a block boundary.

*transform_face maps (i1,j1) ghost cell values to neighbor values, using information about block boundary orientations encoded in **transform_ptr**.*

Mapping contexts

```
fclaw2d_map_context_t* fclaw2d_map_new_squareddisk(const double alpha)
{
    fclaw2d_map_context_t *cont;
    cont = FCLAW_ALLOC_ZERO (fclaw2d_map_context_t, 1);
    cont->query = fclaw2d_map_query_squareddisk;
    cont->mapc2m = fclaw2d_map_c2m_squareddisk;

    cont->user_double[0] = alpha;
}
```

No need to rely on a fixed argument list; arguments specific to the mapping can be passed into the mapping routine.

```
c       call mapc2m(xc,yc ,xd1,yd1,zd1)
        call fclaw2d_map_c2m(map_context,blockno,xc,yc,xd1,yd1,zd1)
```

```
static void
fclaw2d_map_c2m_squareddisk(fclaw2d_map_context_t *cont, int blockno,
                            double xc, double yc,
                            double *xp, double *yp, double *zp)
{
    double alpha = cont->user_double[0];
    mapc2m_squareddisk(&blockno,&xc,&yc,xp,yp,zp,&alpha);
}
```

Current practical limitations

Multiblock limitations

- Users cannot yet design their own block connectivity
  - p4est needs to first set up the connectivity
  - Need a file format for specifying general connectivities
- At most four blocks can meet at a corner

Solver limitations

- Metric terms limit solutions to second order (most likely)
- Solution errors at mapping seams not properly handled.

Software

- Visualization done in Matlab (no VisClaw, Visit, etc support yet)
- Regridding done only at coarse time step, not at intermediate time steps.

Future?

- Extend code to 3d
- Parallel performance testing
- Better treatment of metric terms
- Documentation

Recent NSF funding to

- Develop multistage methods in an AMR Framework (D. Ketcheson)
- Extend existing Ash3d code to AMR (USGS, D. George)

Alpha testers needed!

Let me know if you'd like to out ForestClaw
- Test installation procedure
- Run existing examples
- Port Clawpack examples
- Get some VisClaw graphics support?

No prior experience needed!

Thanks!

Thanks to David Ketcheson, Heba, Ting and KAUST for organizing hosting this event!

Work-Precision