LA-UR- 11-05019

Approved for public release; distribution is unlimited.

Title: Development of High-Performance AMR Codes for Multi-Physics Simulations

Author(s): Anthony Scannapieco, James Painter, Frederick Cochran, Robert Robey, William Dai, Michael Steinkamp

Intended for: International Conference on Numerical Methods for Multi-Material Fluid Flows (MultiMat 2011)

Los Alamos
NATIONAL LABORATORY
EST. 1943

Development of High-Performance AMR Codes for Multi-Physics Simulations

Anthony Scannapieco, James Painter, Frederick Cochran, Robert Robey, William Dai, Michael Steinkamp

Los Alamos National Laboratory

Other Contributors:

Lori Pritchett, Elsie Sandford, Lee Ankeny, Mike Mckay

Abstract

We have implemented a high performance code, Roxane, for multi-physics simulations, in which an advanced parallel strategy is employed. The scope of physics covered in Roxane include hydrodynamics, magentohydrodynamics, material mixing, radiation, isotopes, and etc, for applications such as heavy metals and lead isotopes in SDB stars. We introduced the KD-tree data structure and use it for the connectivity of cells in AMR, mapping between AMR meshes and unstructured meshes, linking between codes, and neighboring processor identification. In the advanced parallel implementation, we allow the variable number of ghost cells to fit different physics solvers, combine all the necessary variables and material compressed data together to reduce the number of communication in parallel environments. The parallel capability has been demonstrated for hydro calculations in one-, two-, and three-dimensions, including material mixing, isotopes, AMR. The parallel capability to link calculation from FLAG was developed and demonstrated. We also developed a scalable parallel IO capability for restarting and visualization. A set of tools to test the correctness of parallel calculations and a test system has been developed.

What is Roxane

- AMR, interface-aware, Q-based hydro with strength in 6 standard geometries.
- Stochastic volume fraction material advection for sub-zonal physics models.
- Interface-aware 3-T radiation diffusion, such as radiative processes in astrophysics
- Elastic-plastic strength models, such as asteroid impacts.
- Integrated physics capability.
- Mix models.
- Basic resistive toroidal MHD in 1D & 2D.
- Triangle link from other packages.
- Advanced parallel strategy and implementation.

Outline

- KD-tree implemented & its usage in Roxane.
- Procedure in parallelization in hydro & explicit solvers.
- Parallelization of AMR.
- Re-partition and load-balance.
- Parallel tri-link.
- Parallel IO: restarting & visualization dumps.
- Correctness.
- Testing and test system.
- Future work: parallel & modern computer architectures.

KD-tree in Roxane

- **KD-tree of structured mesh AMR, 1D, 2D, & 3D**

  Significantly simplified the logic

  Used in connectivity of AMR mesh: nlow and nhgh

  Used in mapping between different meshes of physics packages.

  Used for the identification of neighboring PEs.

- **KD-tree of unstructured mesh, both 2D and 3D**

  Used in tri-link

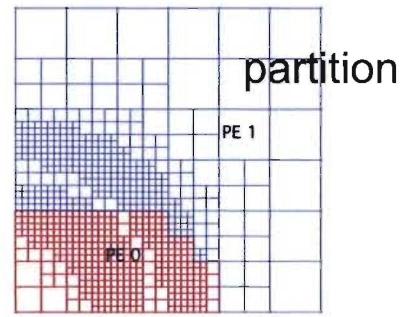
- **KD-tree of points**

  Used for the visualization of AMR data

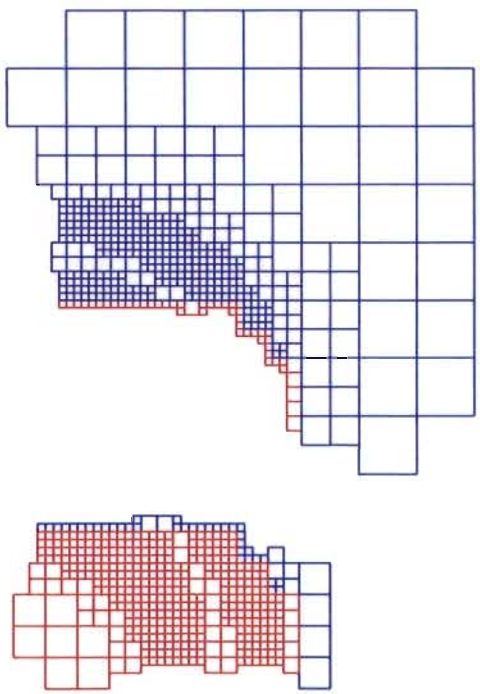
Design Principle for Parallelization

- Assumption: Given a partition, bad or good.
- Use KD-tree to determine the list of neighboring PEs in 2D & 3D
- O(Npe log Npe) operation to determine neighboring
- Allow any domain partitions.
- Each PE communications only to neighboring PEs for ghost cells
- No global arrays, ex, global_to_local(global_id) = local_id,
- The number of layers of ghost cells is a parameter to fit solvers.
- Mainly one communication for all variables, compressed material data, & isotopes.

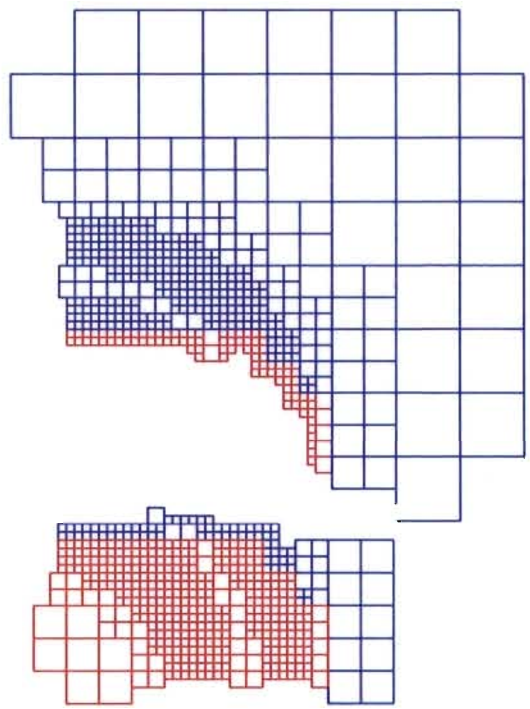
Illustration of Ghost Cells



1 layer of ghost cells



2 layer of ghost cells



3 layers of ghost cells

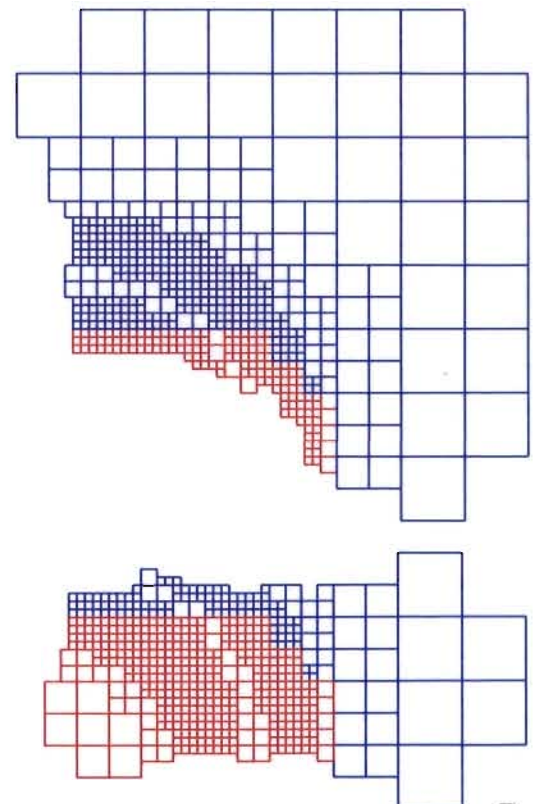
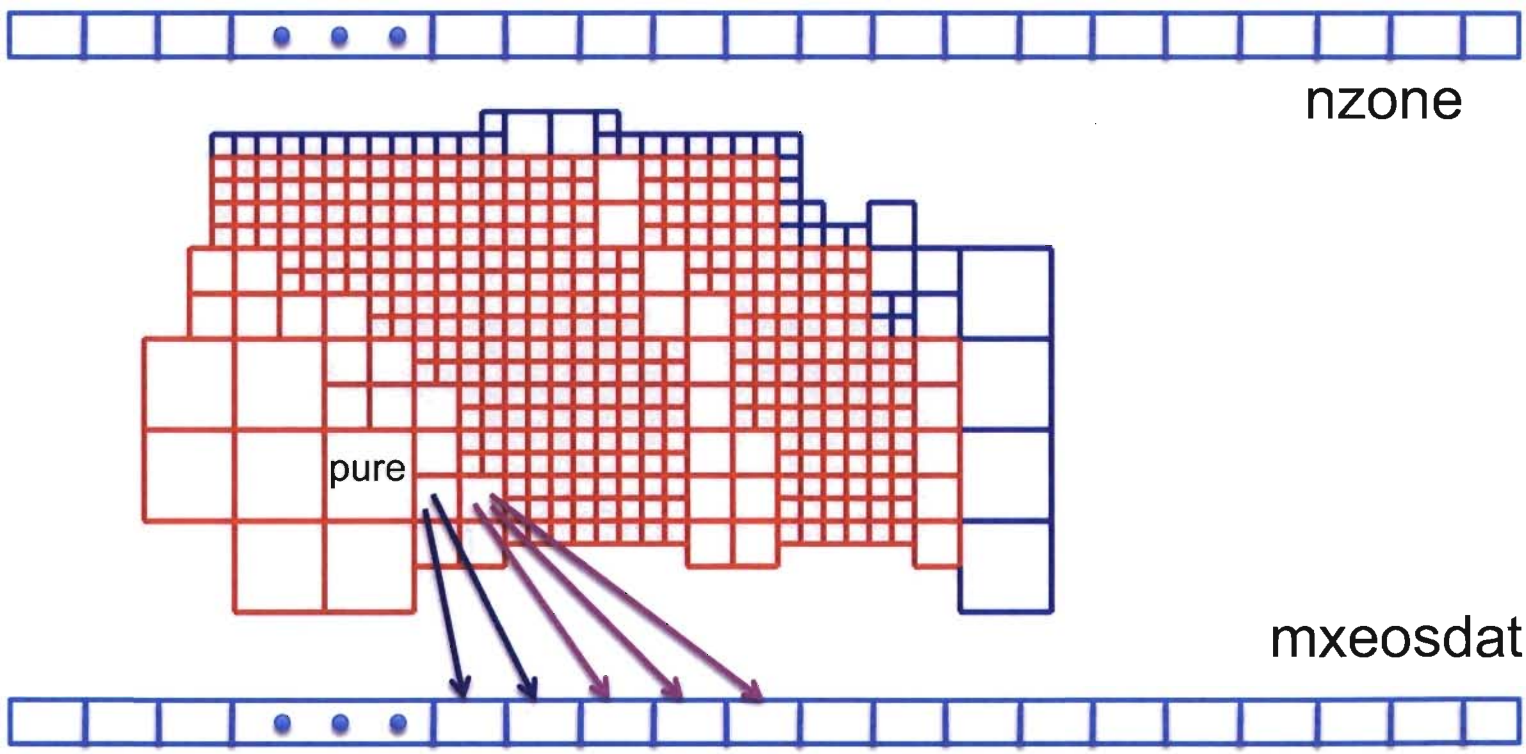


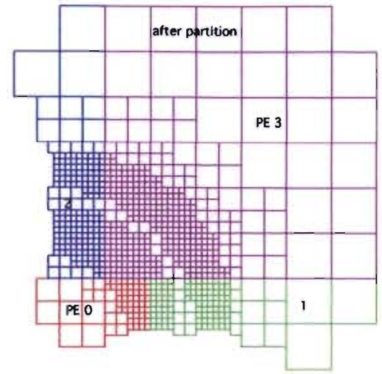
Illustration of compressed material variables

normal zonal variables: for example, d, mass density



mixing variables: e.x., df, density of each mat in each mixing cell

Procedure in implementation

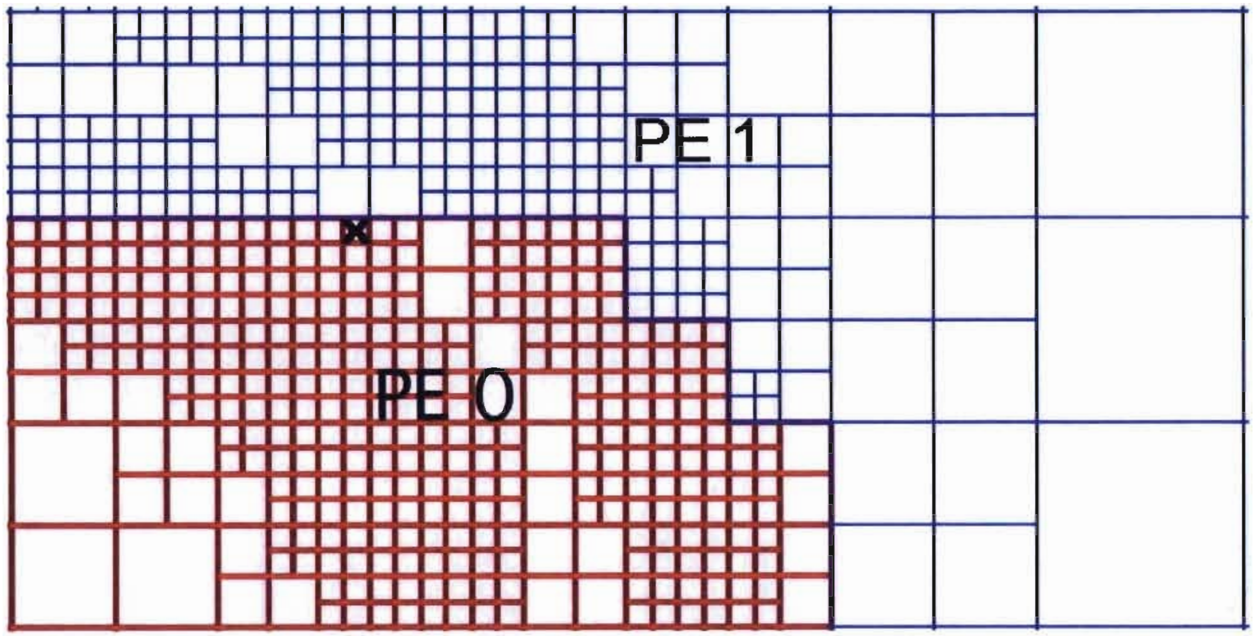


1. Determine neighboring PEs.
2. Determine which cells are ghost cells of neighboring PEs.
3. Pack variables, ghost cell geometry, material vars & composition, into buffers, one for each of neighboring PEs
4. Send each buffer to appropriate neighboring PE
5. Receive one buffer from each of neighboring PEs
6. Construct ghost cells from the geometry received from neighboring PEs
7. Unpack buffers received from neighboring PEs, put vars onto ghost cells.
8. Construct linked lists for material data & isotopes on ghost cells.
9. Run physics packages A, B, C, ….
10. End of cycle

Example of Subroutines for Parallel

```
call ghost_init
call ghost_var_sizes
call ghost_mxvar_sizes
call ghost_cpvar_sizes
call put_ghost_vars
call put_ghost_mxvars
call put_ghost_cpvars
call comm_all_vars
call get_ghost_vars
call get_ghost_mxvars
call get_ghost_cpvars
```

Parallelization of AMR



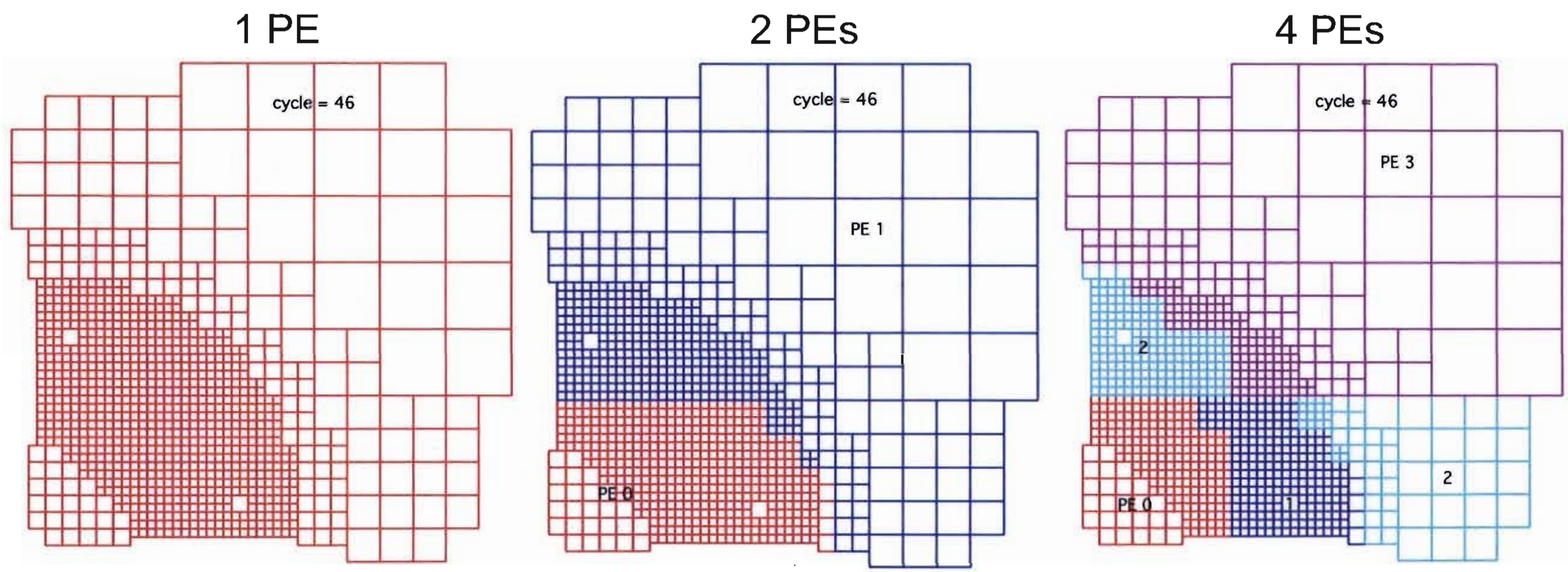
Each level involves one communication.

```
!    Spread the mesh potential.

     do lev = levmx, 0, -1
        update_ghost_var(mpot)
        .....
     enddo
```

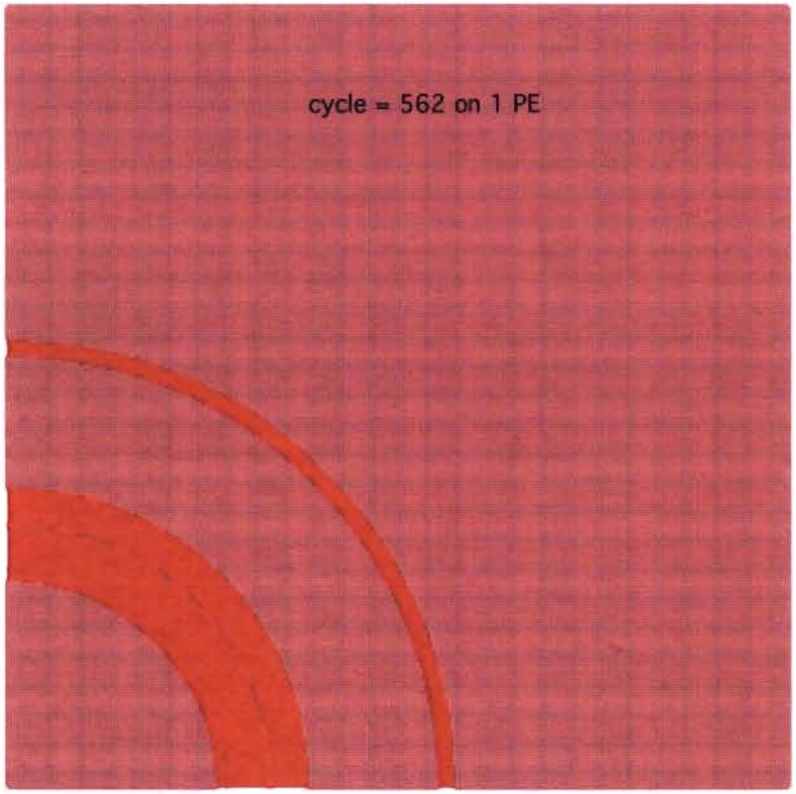
Example: 1, 2, & 4 PEs

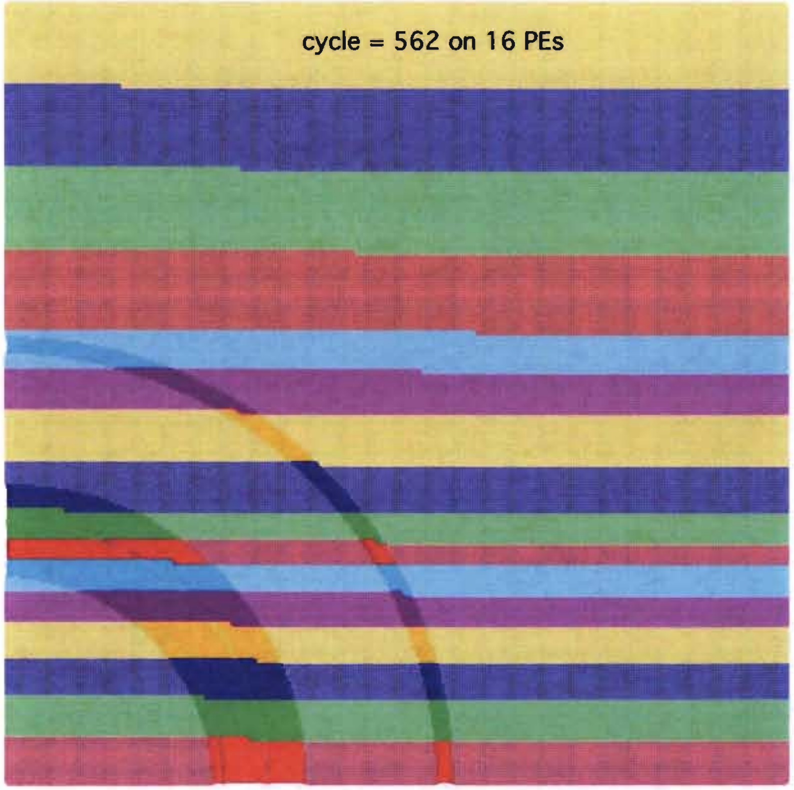
4 materials, 4 levels
Colored by PE.
Solutions are same.



Example: 16 PEs and 1 PE

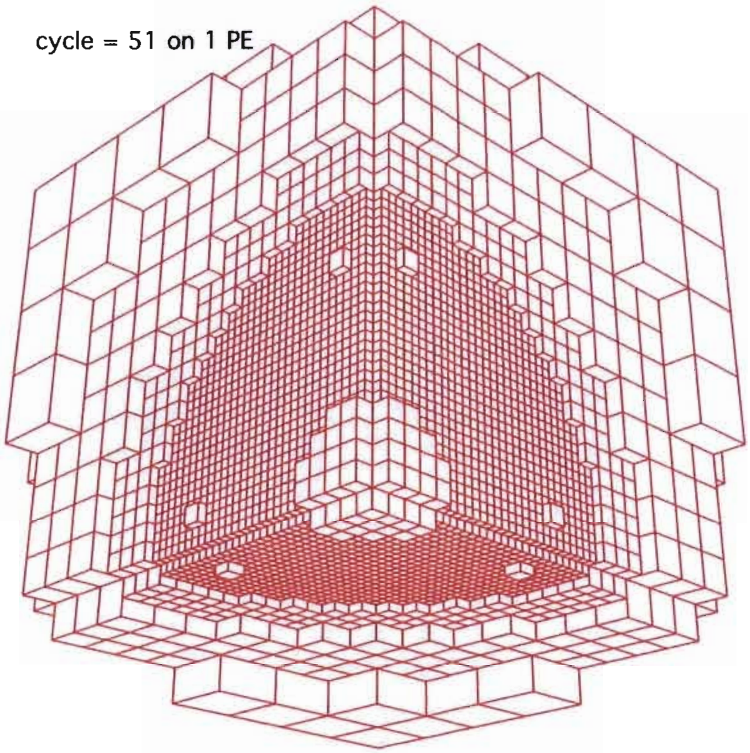
4 materials, 4 levels
Colored by PE
Solutions are same



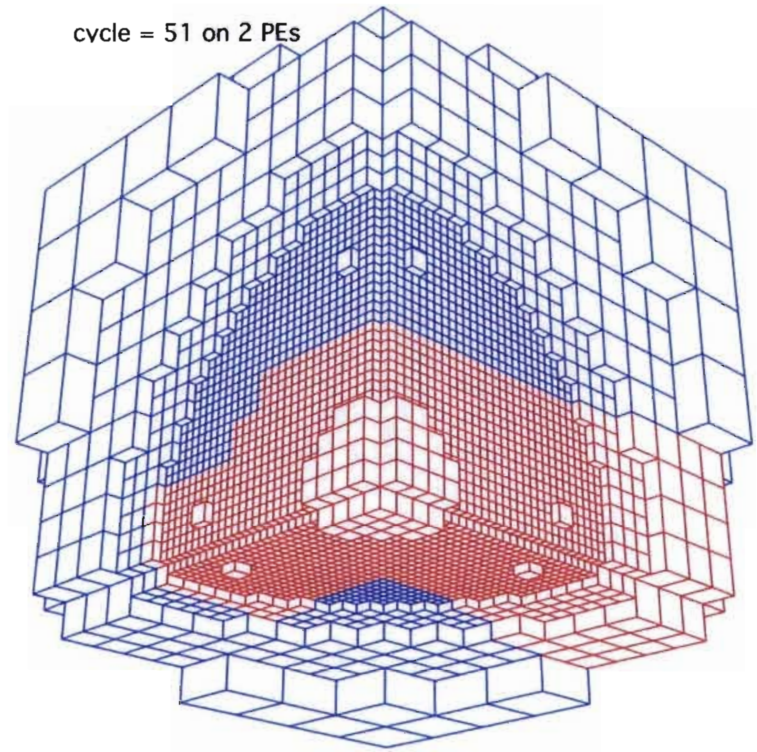


4 materials, 4 levels
Colored by PE
Solutions are same

3D Example: 2 PEs vs 1 PE



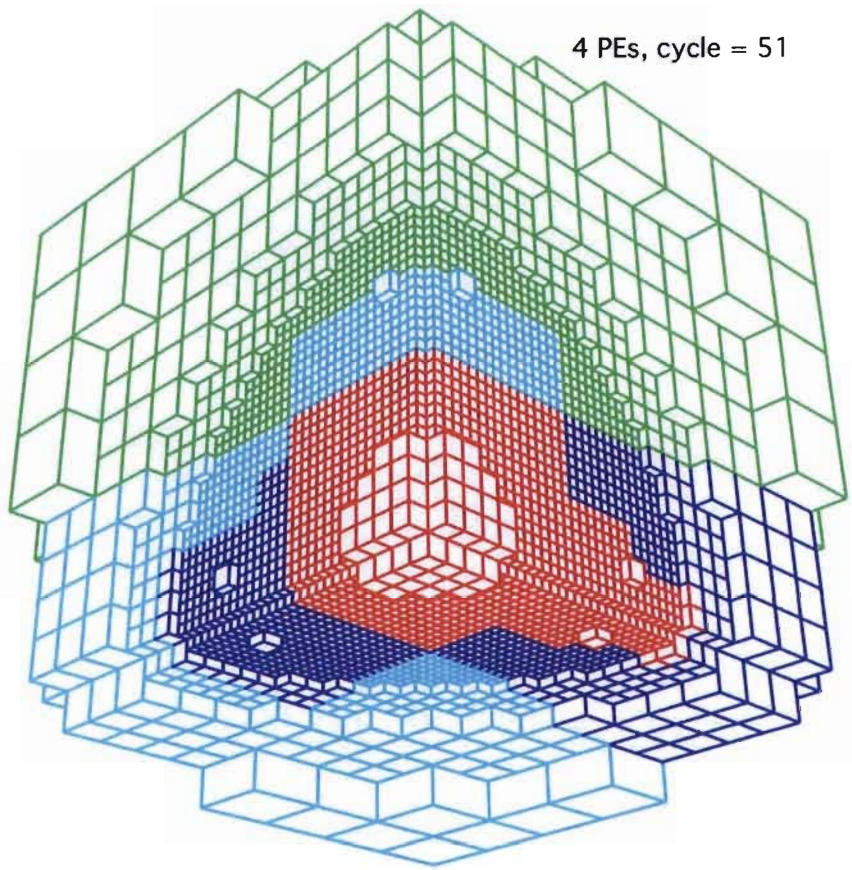
1 PE



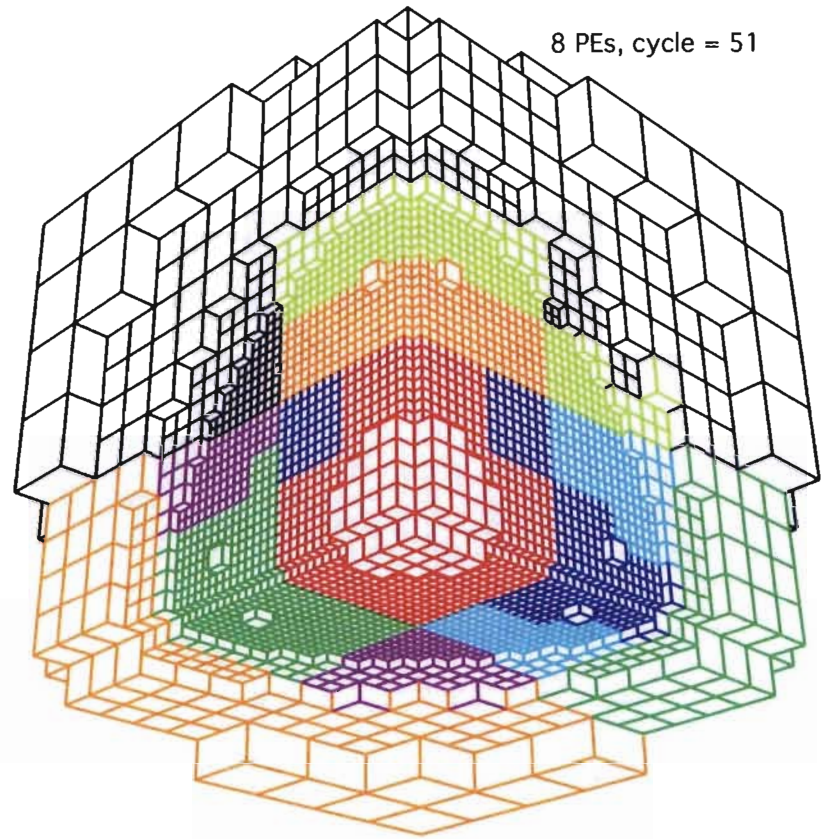
2 PEs

3D Example: 4 & 8 PEs

4 materials 4 levels
Colored by PE
Solutions are same



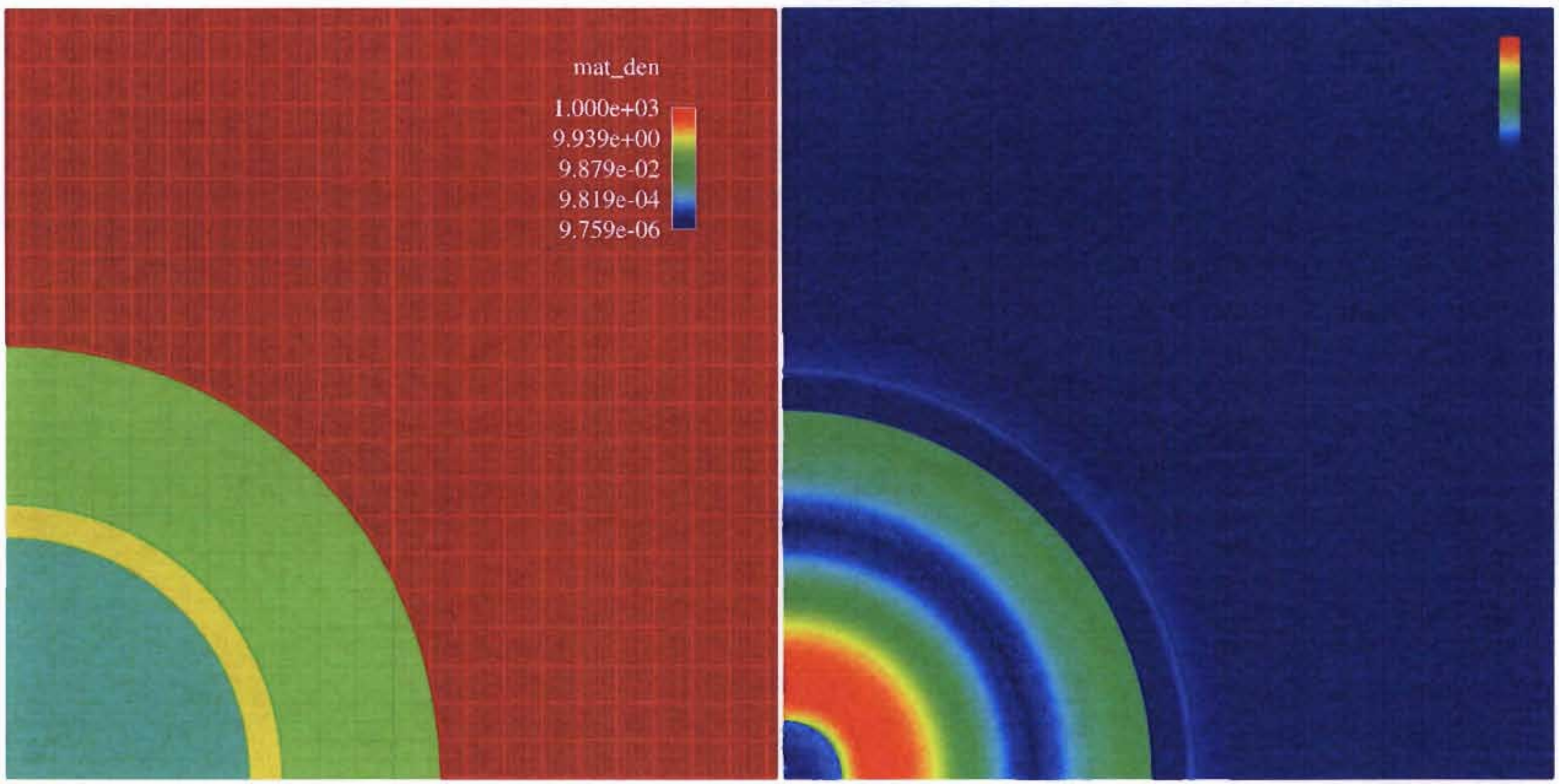
4 PEs



8 PEs

Example: 64 PEs, 2.2M cells, 12000 time steps, r-z

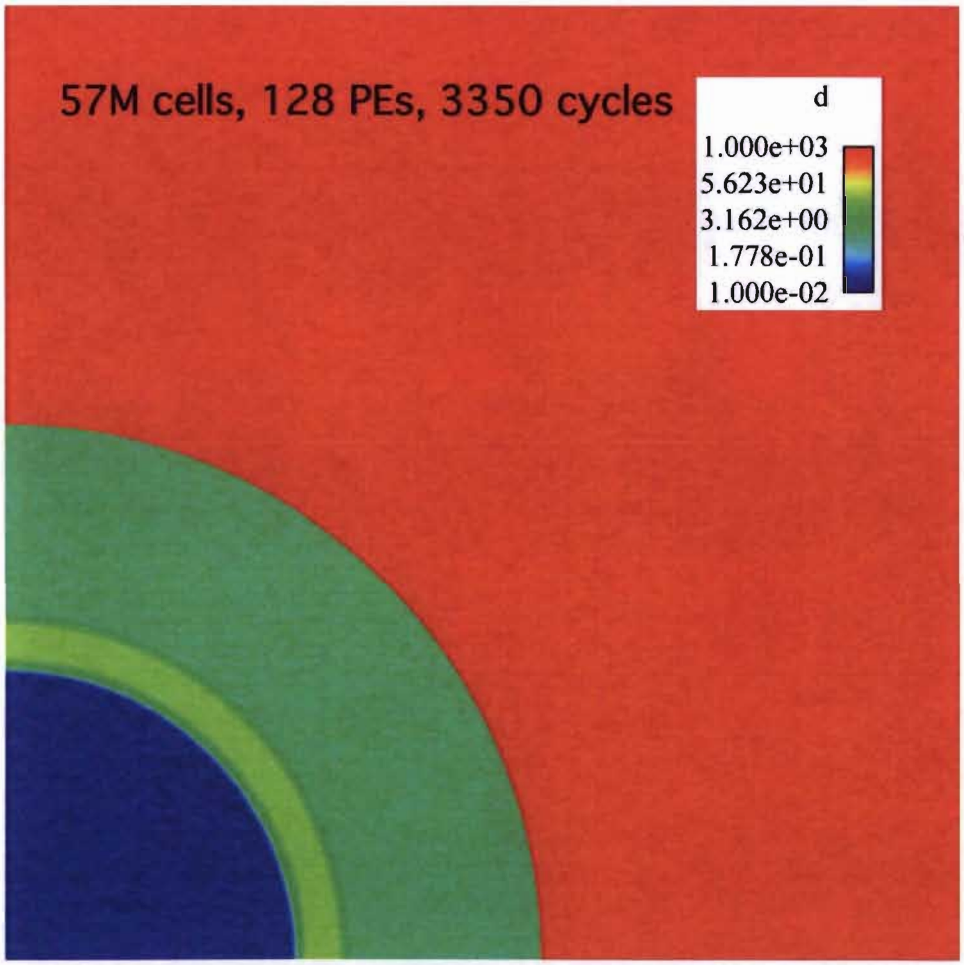
4 materials 4 levels



t=0, colored by mat density

ncycle = 12,000, colored by flow velocity

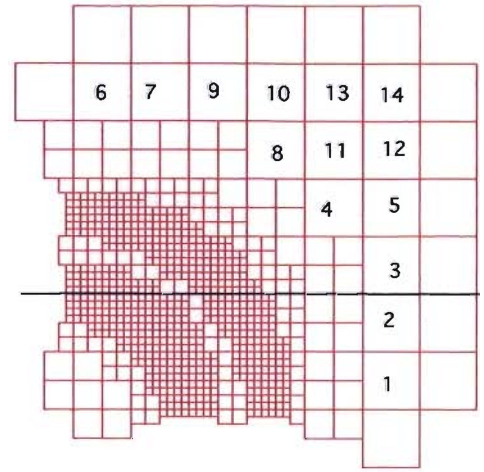
Example: 128 PEs, 57 M cells, 3350 time steps



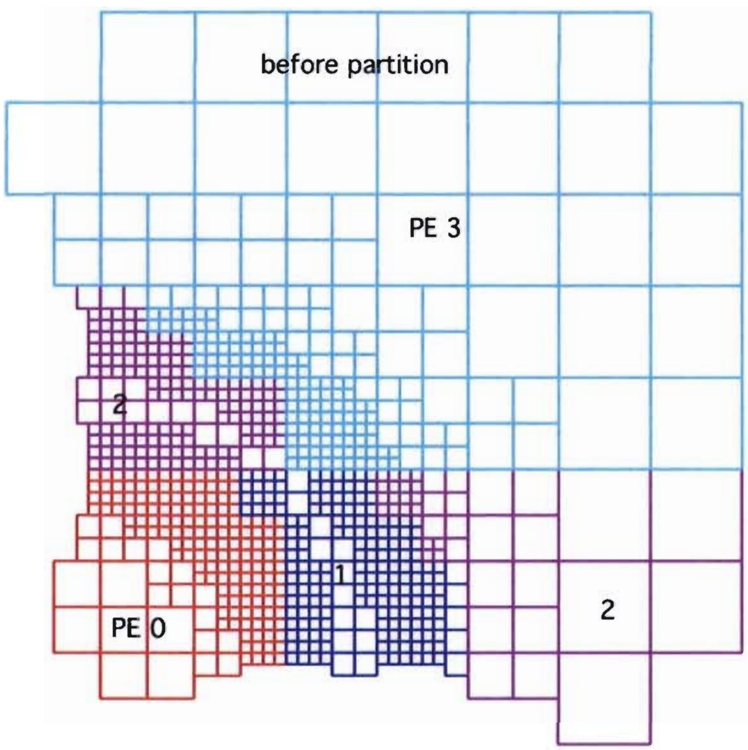
4 materials, 4 levels
Colored by density.

Why re-partition

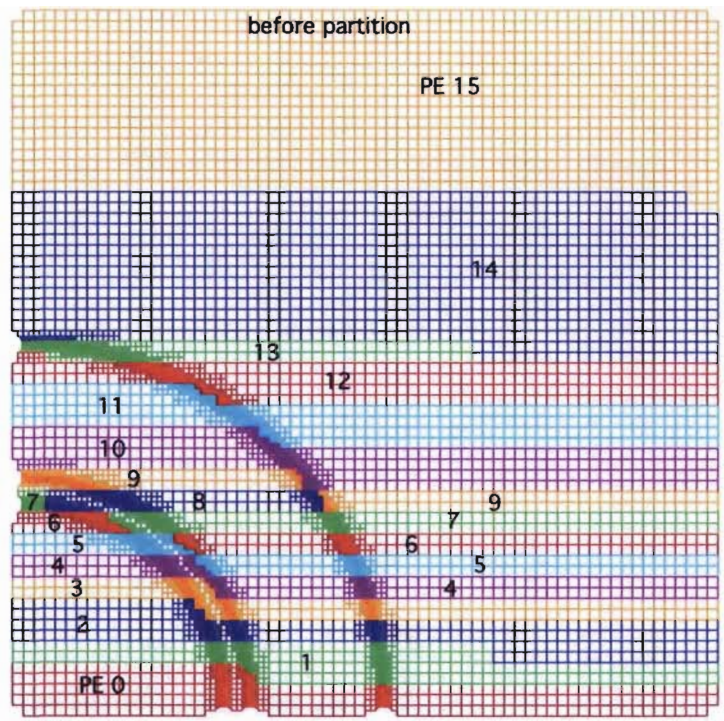
Colored by PE



4 PEs



16 PEs



PEs: 5, 6, 7, 9

Re-partition and load-balance

- Add weighting functions.
- Requirement from (nlayer > 1) of ghost cells.

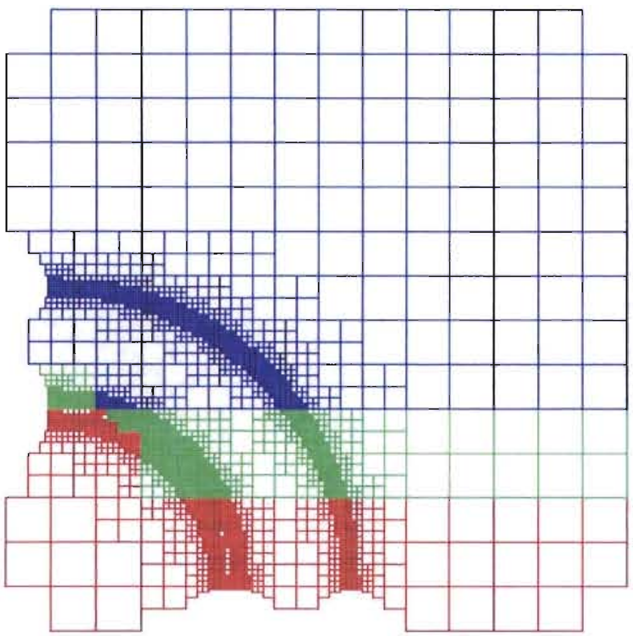
Procedure involved in load balance

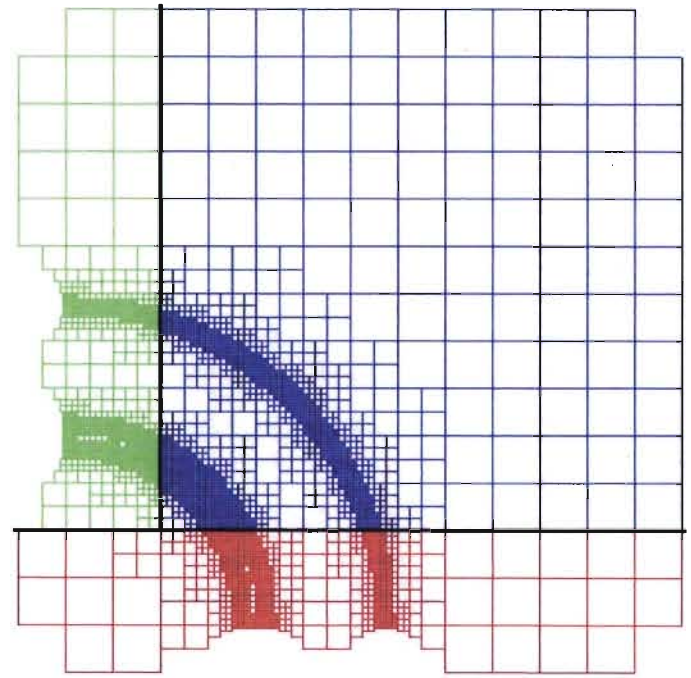
- Weighting function for partition, ex. W(nz) = 1 + 0.1 * [nmat(nz) – 1]
- Check balance.
- Partition to determine which_pe(nz) through Zoltan or our own tool.
- Pack cells, vars, material data, & compositions, send them to other PEs.
- Take cells, vars, material data, and compositions out from mype.
- Communication.
- Rearrange vars & material/composition link lists for the remaining cells.
- Unpack variables, material data, & composition data.
- Build material/comp link lists of those cells from other PEs.

- Build ghost cells for parallel calculations.

Re-partition and Load-balance

Zoltan: more choices & and easy to use, scalability ?
Our own: simple, but only geometry partition.



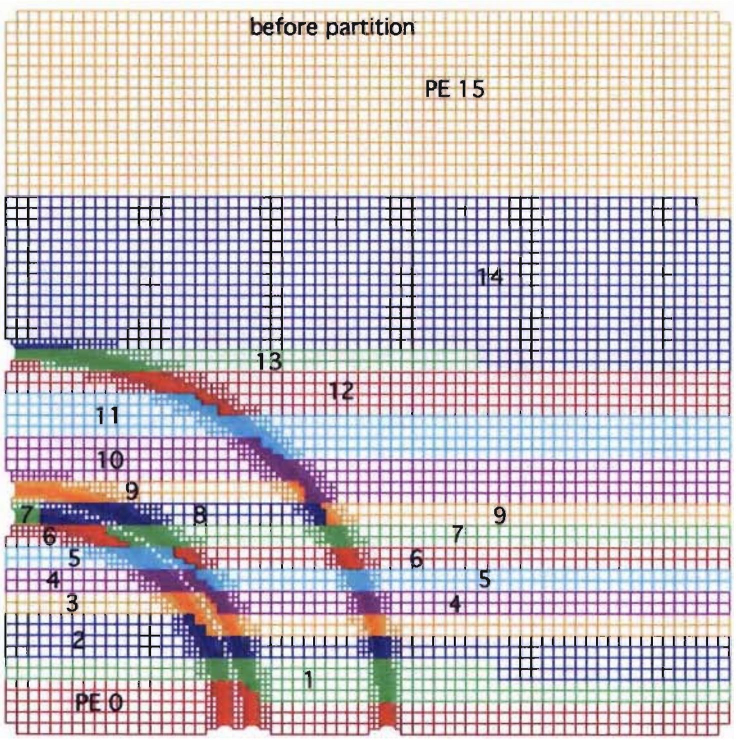


At no instant, does any PE have cells of other PEs.

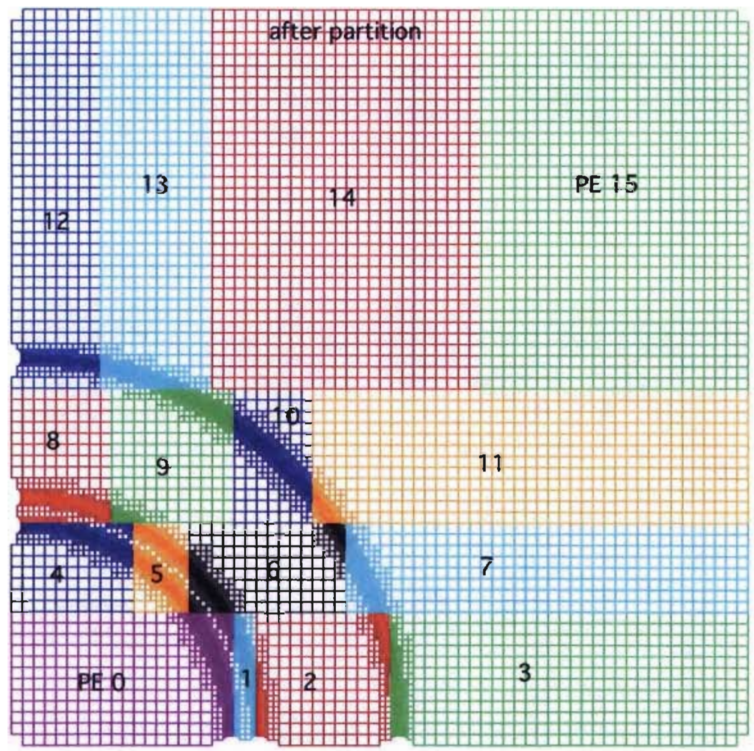
Example re-partition on 16 Pes

Colored by PE

Before



After



PEs: 5, 6, 7, 9

Example re-partition on 64 PEs

Colored by PE

AMR: 4 levels

mats: 4

Before

After

Example re-partition on 256 PEs
Colored by PE

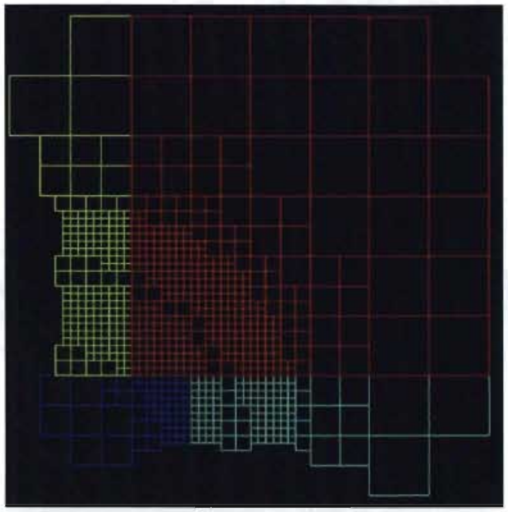
AMR: 4 levels
mats: 4

Before

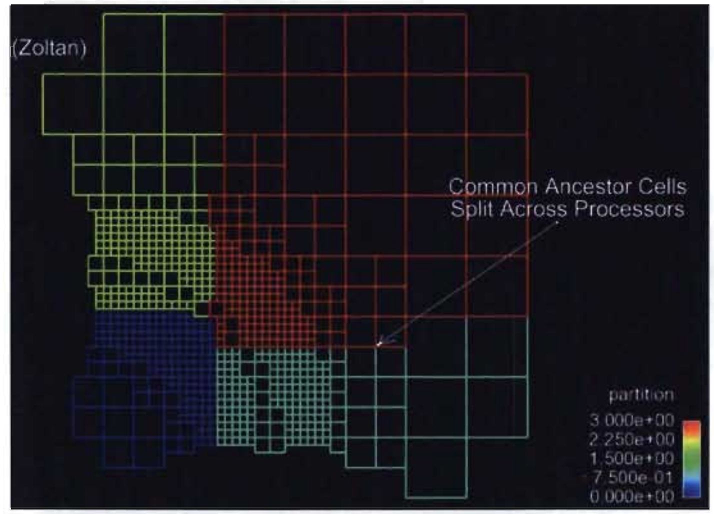
After

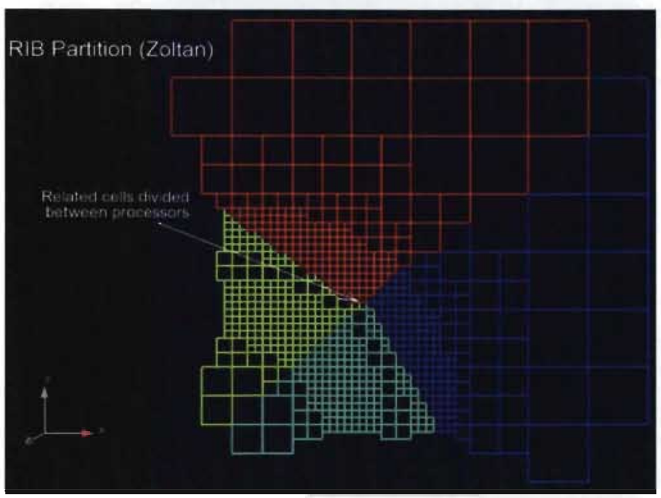
Home grown tool vs Zoltan

Home grown tool

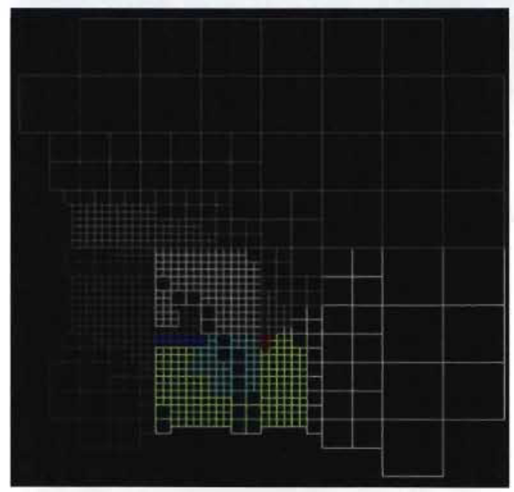


Zoltan



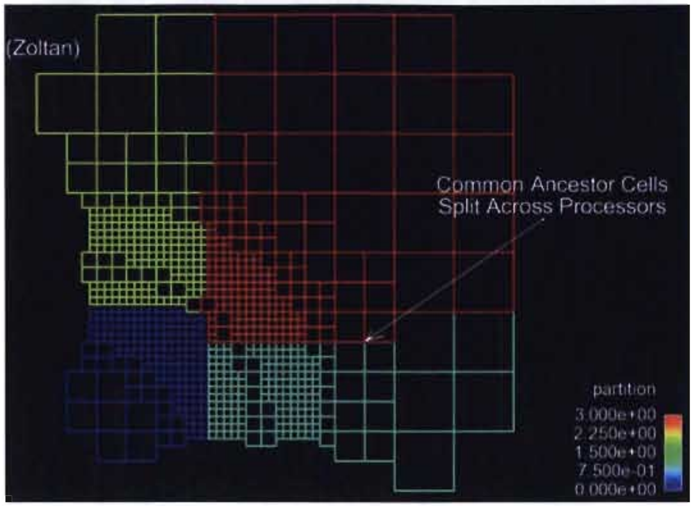


Zoltan



Zoltan

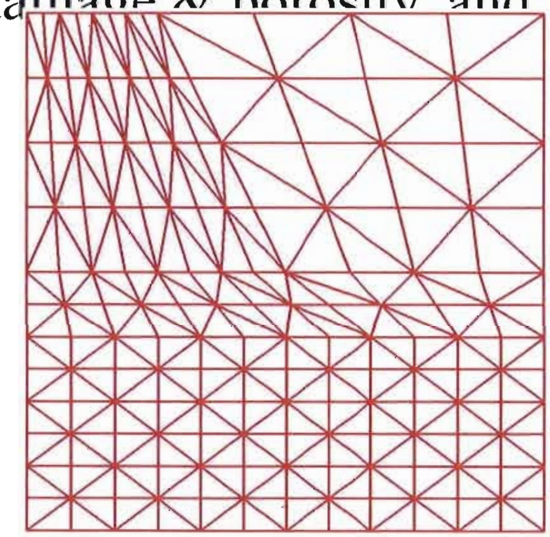
Making Zoltan work for Rozane



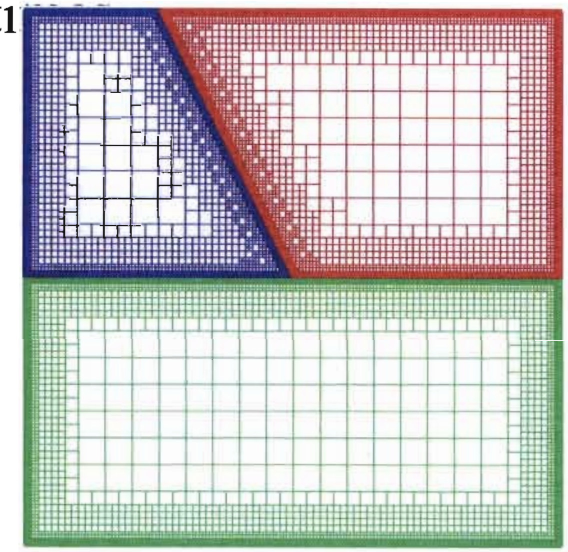
base_cell_num(nbase)
base_cell_info(ncell_ea_base, nmat_ea_base, nbase, mybase, nzone)
call zoltan(base_owner)
zone_owner(base_owner, nbase, mybase, myowner, nzone)

Parallel Tri-link

- Tri-link: Link from Flag to Roxane to continue calculation, for example, for mixing of materials and turbulence study.
- Second order accurate, analytic, iterative AMR linker
- Variables linked include: kinetic, thermodynamic, strength, damage & porosity, and various ti



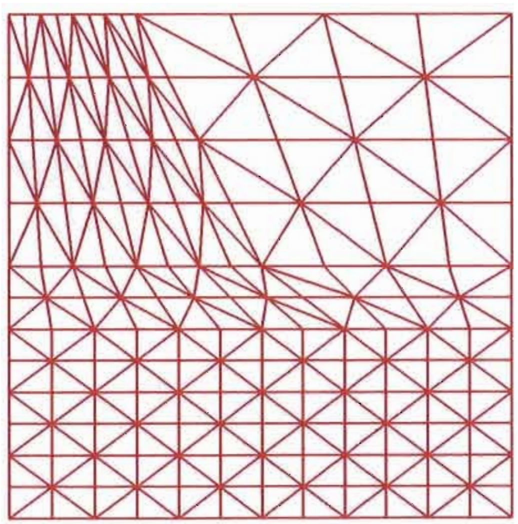
Triangles from Flag



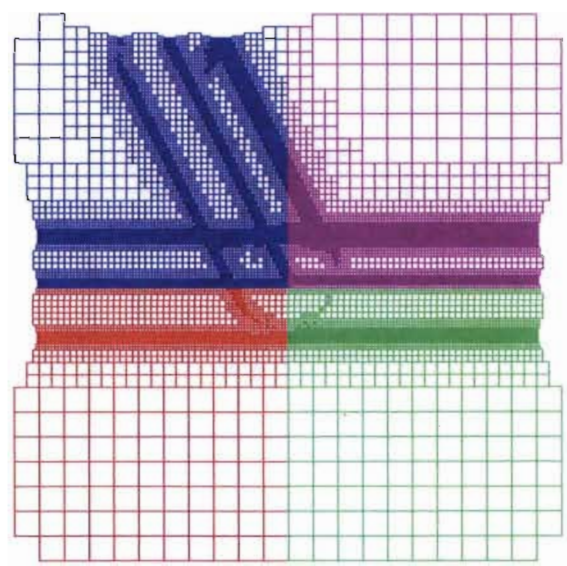
Roxane mesh after link, colored by mat

Procedure in Parallel Tri-link

- Space partition to define the region of Each PE
- Determine relevant triangles of my own PE
- Each PE reads relevant triangles & variables defined on them
- Each PE do serial interpolation, mapping, AMR adaption & meshing.
- Trim out the cells that are out of the region of each PE.



Triangles from Flag

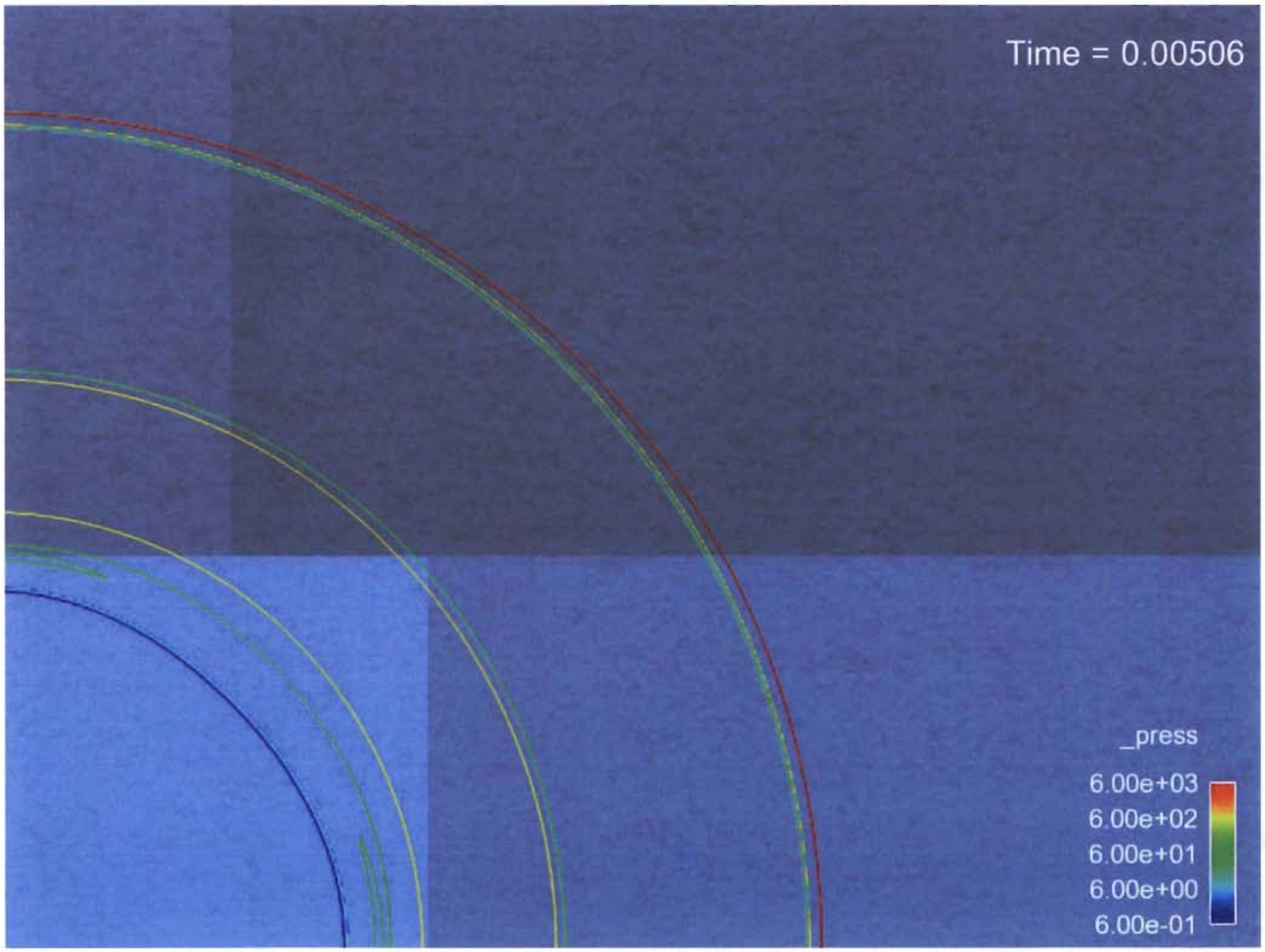


mesh run at the end, colored by PE

Parallel IO: restarting & visualization dumps

- Parallel restart capability developed and scalable.
- Parallel viz dump capability developed and scalable.
- Capability to draw iso-surfaces
- Capability to rendering large data set (across boundaries between servers through ghost cells)

Example of Viz through multi-servers: contours / iso-surfaces



Correctness

- Single PE on different machines, ex, Turing vs Yellowrail

$$diff \approx 10^{-12}$$

- Single PE on same machine, but with vs w/o –g in compiling

$$diff \approx 10^{-13}$$

- Single PE runs may be very different due to AMR and/or physical instability on different machines, or different ways of compiling.

- (A+B) + C is not equal to A + (B + C).
- The order is changed in a parallel run, ex. coarsening criteria.
- Keep the exactly same order of math operations as much as we can.

Testing

- All pass with tol = 1.0e-40 when compiled with –g
- Blue pass with tol = 1.0e-40 when compiled w/o –g
- Red pass with larger tol when compiled w/o -g

problemsls	ncycle		problems	ncycle
roxpk_blake_sph1d_2m1lg	1,200		roxpk_rtz_1d	7000
roxpk_hyd_2m4lg	50		roxpk_sed_1d_1m4lg	44522
roxpk_hyd_2m4lg_p	50		roxpk_sed_1d_1m4lg_qsm_0	2500
roxpk_hyd_2m4lg_rs	50		roxpk_sodr2m3lg	1000
roxpk_sed_2d_0_005	231		roxpk_sodz2m1lg	2000
roxpk_sed_2d_1m5lg	231		roxpk_sodz2m3lg	2000
roxpk_sph_2dcart	46		roxpk_sph_1dsphere	42
roxpk_sph_3dcart	51		roxpk_verney_cyl1d_2m1lg	10000
roxpk_sphin	46		roxpk_verney_sph1d_2m1lg	5500
roxpk_sphin_p	46		roxpk_verney_sph1d_2m3lg	6000
roxpk_sphin_qsm_0	46		roxpk_x_str_sg_2m4lg	500
roxpk_sphin_rs	46		roxpk_taylor_rz_sg_2m1lg	500
roxpk_tri4m5lg	100		roxpk_mhd_cyl1d_2m1lg	2000
roxpk_verney_cartxy_2m1lg	400			
roxpk_mhd_cyl2d	100			

Summary

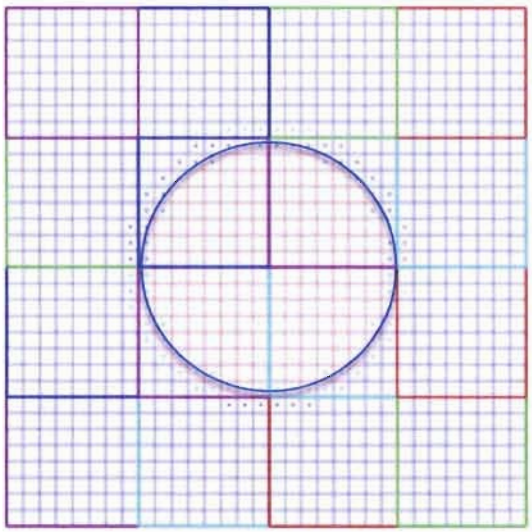
- Many physics packages developed.
- Physics packages parallelized: hydro, advection, strength, eos, mixing, isotopes, explicit and implicit MHD calculations, for atrophysics problems, such as heavy metals and lead isotopes in SDB stars, asteroid impacts.
- AMR parallelized.
- 1D, 2D, & 3D parallelized.
- Parallel restarting capability developed.
- Parallel viz dump capability developed.
- Capability to visualized Roxane data through Ensight developed.
- Tools for rigorous test for the correctness of parallel runs developed.

Future Work

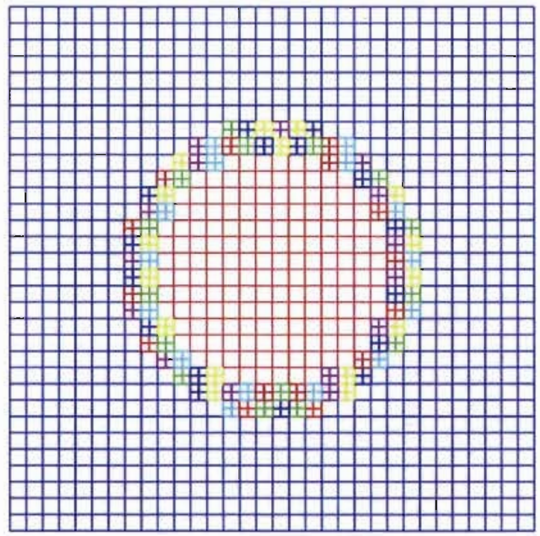
- Continue to check the correctness of parallel runs.
- Continue Zoltan integration.
- Parallel runs of realistic problems.
- Large runs of Roxane.
- Parallel implicit solvers for radiation.
- Large scale visualization of Roxane data, Ensight/VisIt.

- Move AMR and EAP/Roxane to modern architectures.

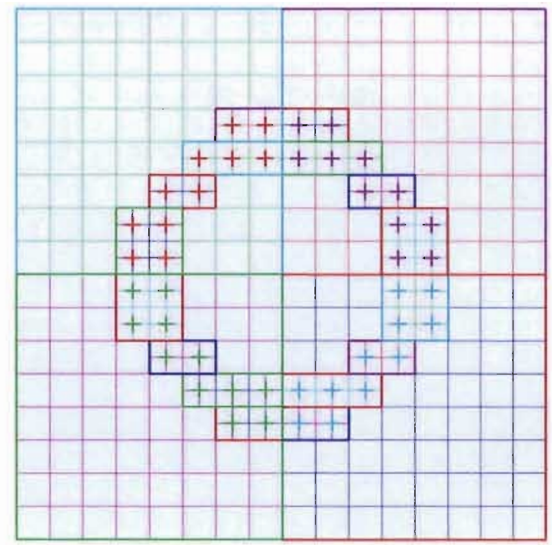
Typical AMR Approaches



block-based AMR
over-refined



cell-based AMR
data-locality not good

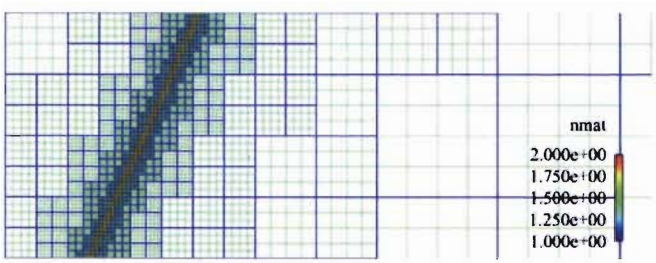


patch-based AMR
uniformity not good

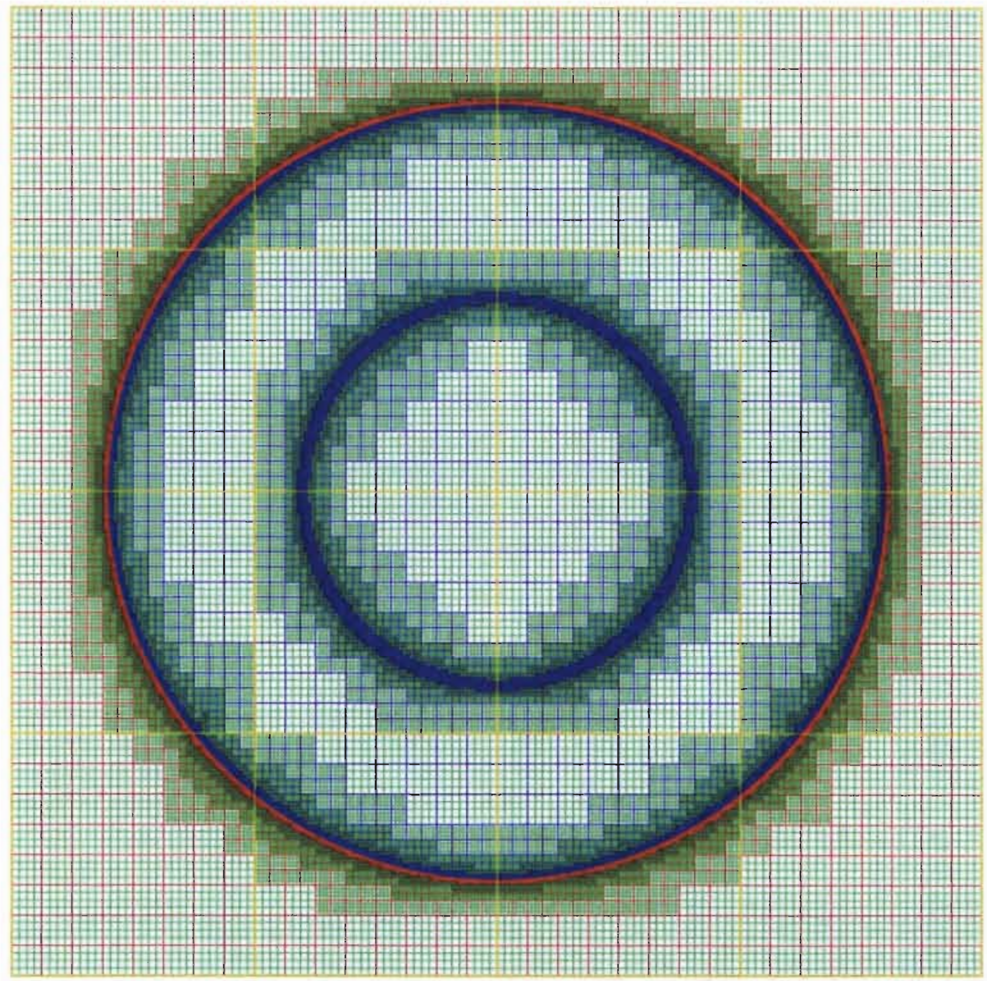
Combine block-, patch-, cell-AMR to seek greater uniformity.

Each *briquette* has the same size.

Refinement near shock and material interface



near mixing cells



colored by pressure