Development of a Multi-physics Code with Adaptive Mesh Refinement

Anthony Scannapieco, **William Dai,** Ted Cochran, Chong Chang, Britton Girard, Scott Runnel

Los Alamos National Laboratory

Contributors:

James Painter, Bob Robey, Elsie Sandford,
Lori Pritchett, Paul Webber, Lee Ankeny, Michael McKay,
Paul Woodward, Michael Steinkamp

LA-UR-14-24616

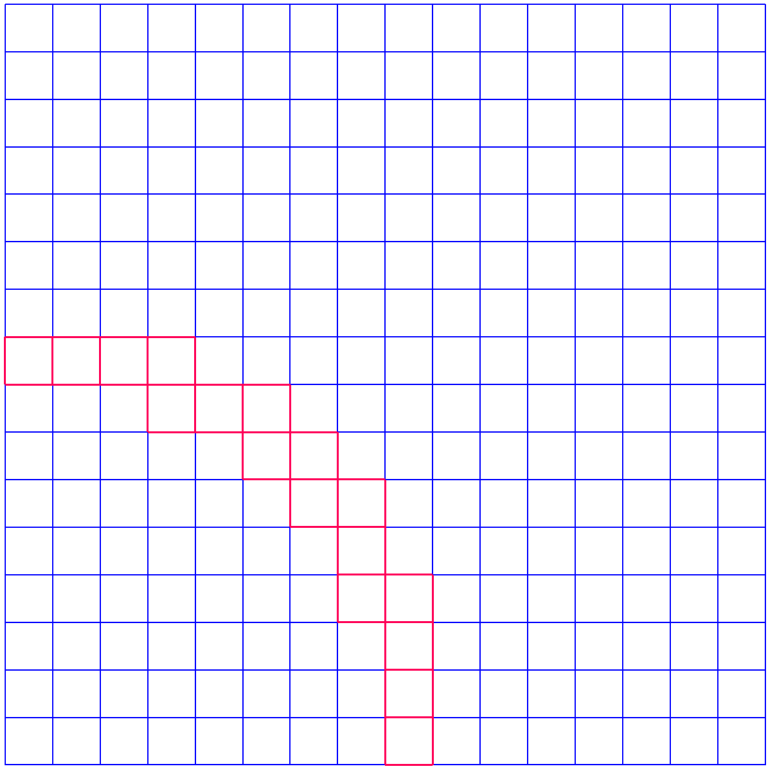
Outline

- What is Roxane.
- Adaptive mesh refinement
- Parallelization
- IO
- Testing
- **Numerical methods for plasma 3-T radiation diffusion**
- Toward advanced computer architectures

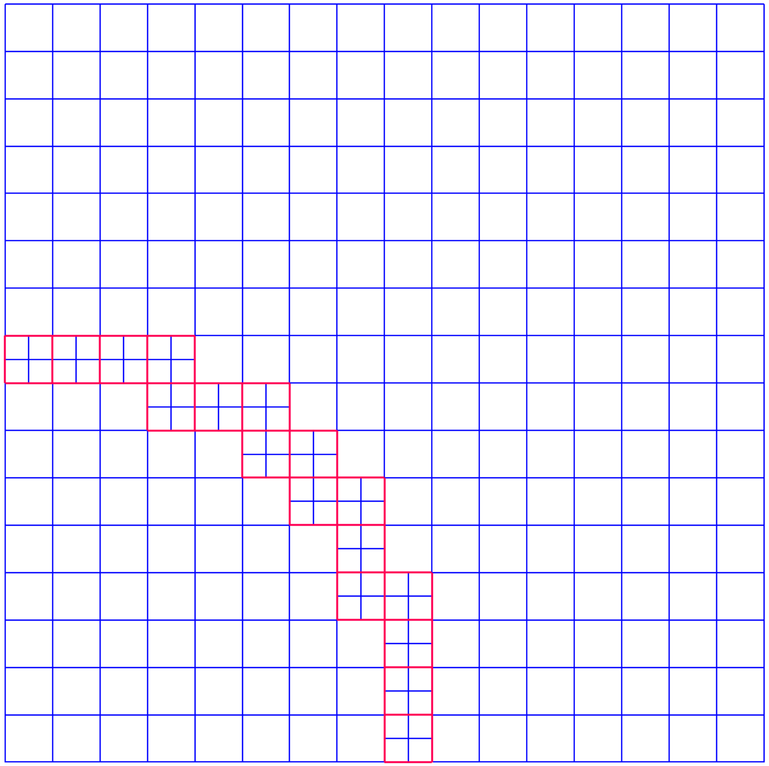
What is Roxane

- Developed for the program need.
- AMR, interface-aware hydro with strength in standard geometries.
- Non-equilibrium material pressure and temperature.
- Volume fraction material advection for sub-zonal physics models.
- Material strength models (Steinberg-Cochtran-Guinan and PTW)
- Dynamical material mix models(Turbo and KL).
- Sesame and analytic EOS
- Models of high explosives (DSD, Forest Fire, and more).
- **Plasma 3-T radiation diffusion.**
- Basic resistive MHD with circuit for experiments.
- ……

Cell-based AMR

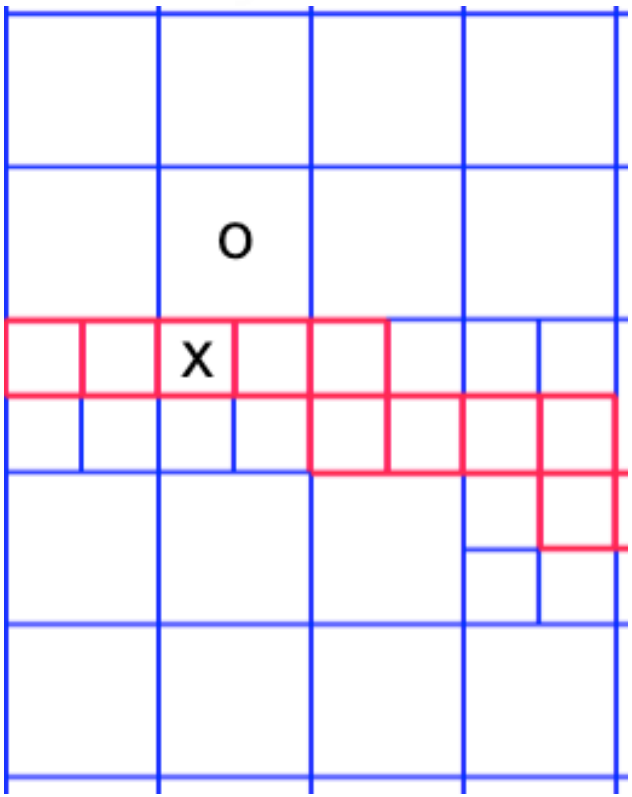


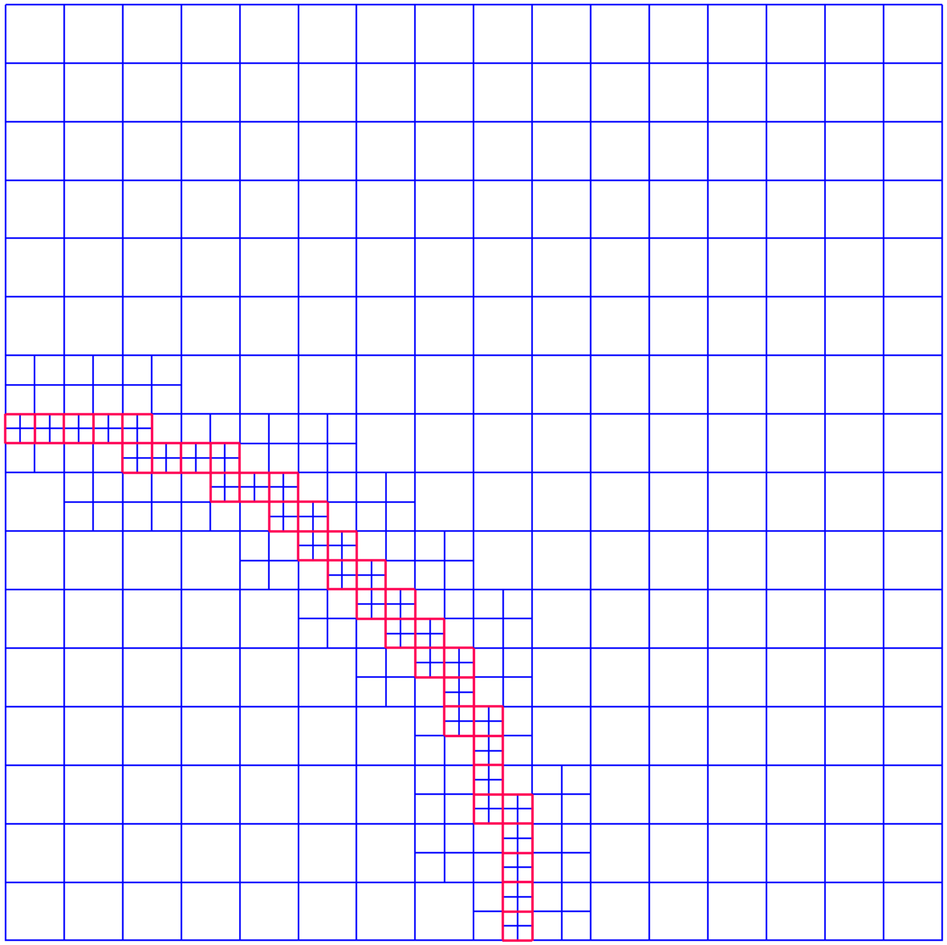
cells to be refined



1st level refinement

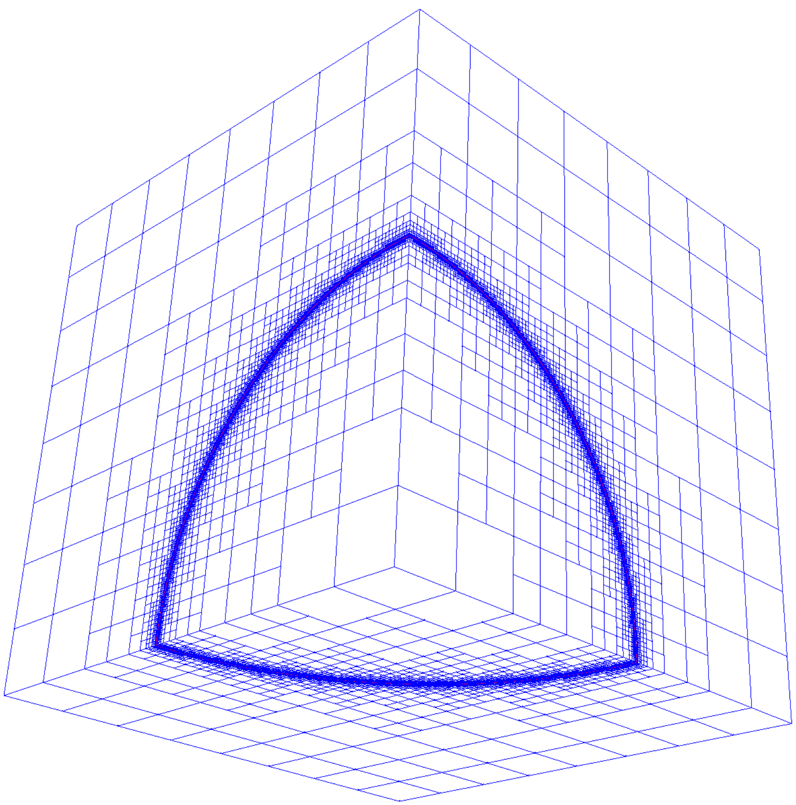
Smooth Transition in Refinement



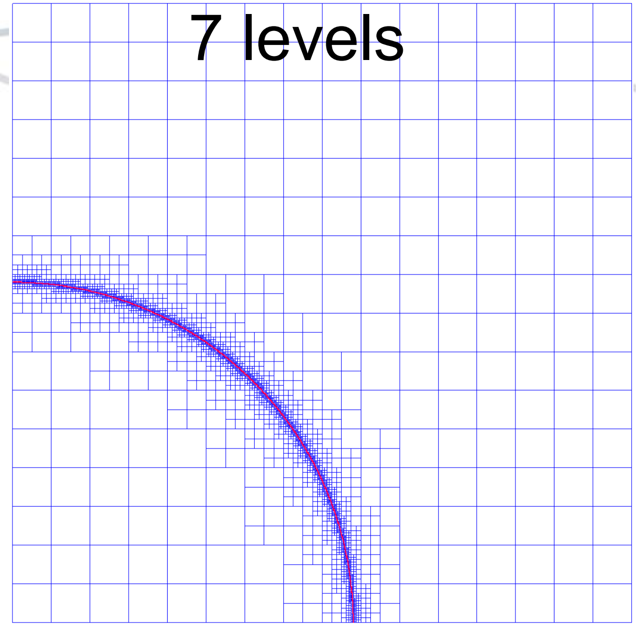


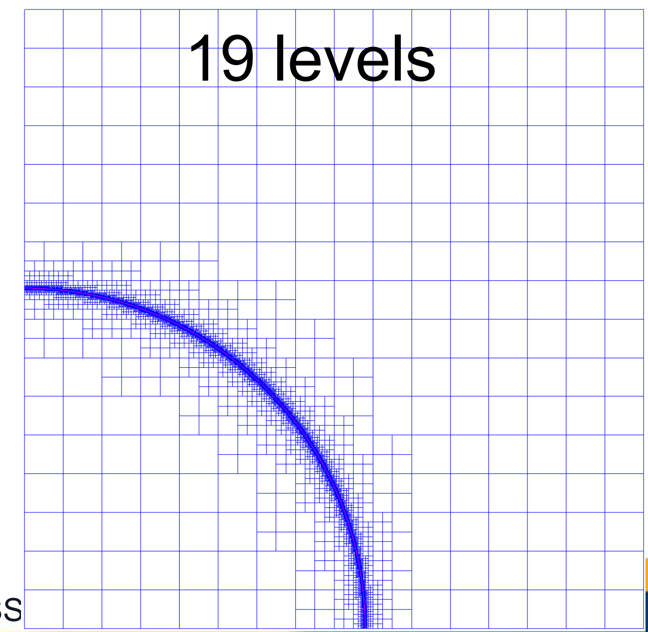
UNCLASSIFIED

Examples of Refinement



9 levels





Data Structures

- Cells: xe(nz, idim), dxe(nz, idim)
- List of cells

  do nz = 1, nzone

  ……

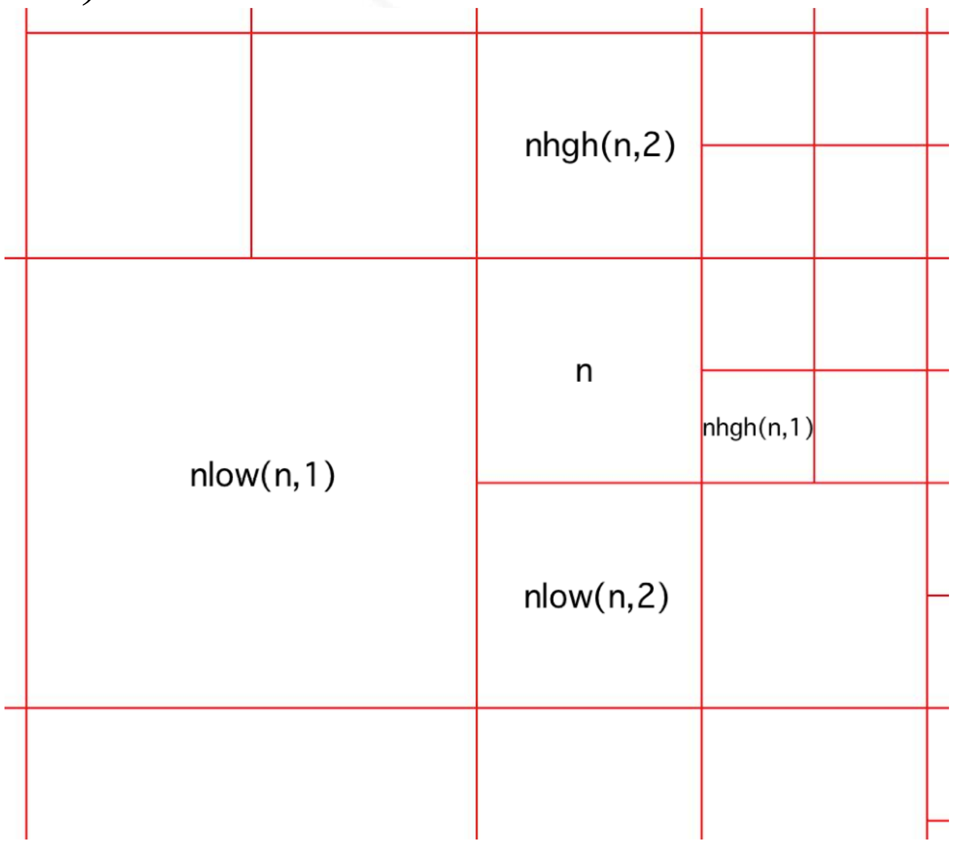
  enddo
- Connectivity

  nlow(nz, idim)

  nhgh(nz, idim)
- Cell-based data

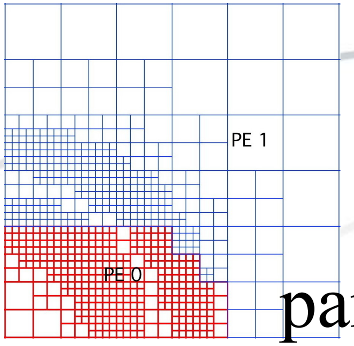
  d(nz), p(nz), t4(nz), …
- Material-based data

  irf(m), vf(m), pf(m), ef(m), t4f(m), …



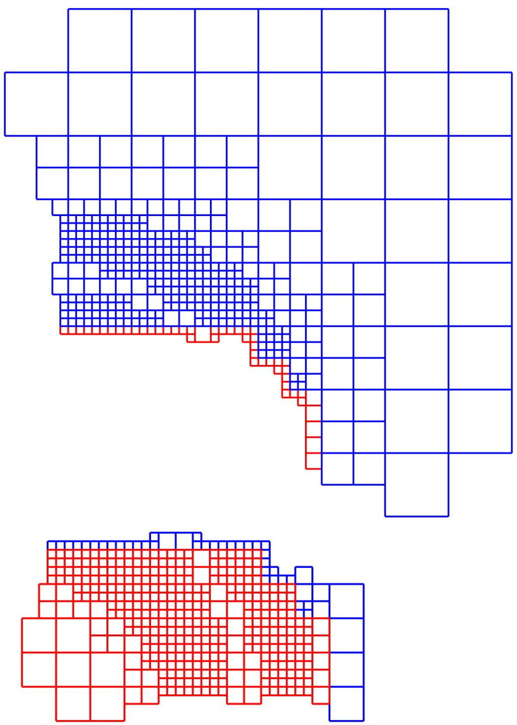
UNCLASSIFIED

Illustration of Ghost Cells

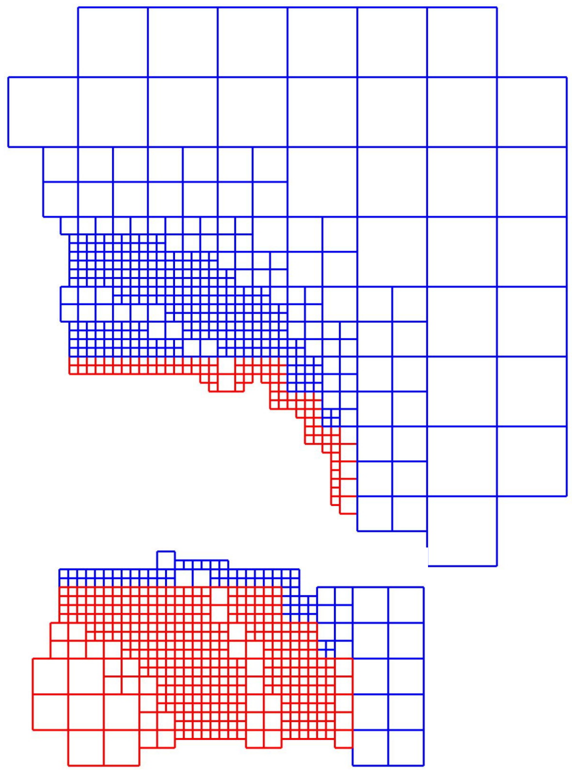


partition

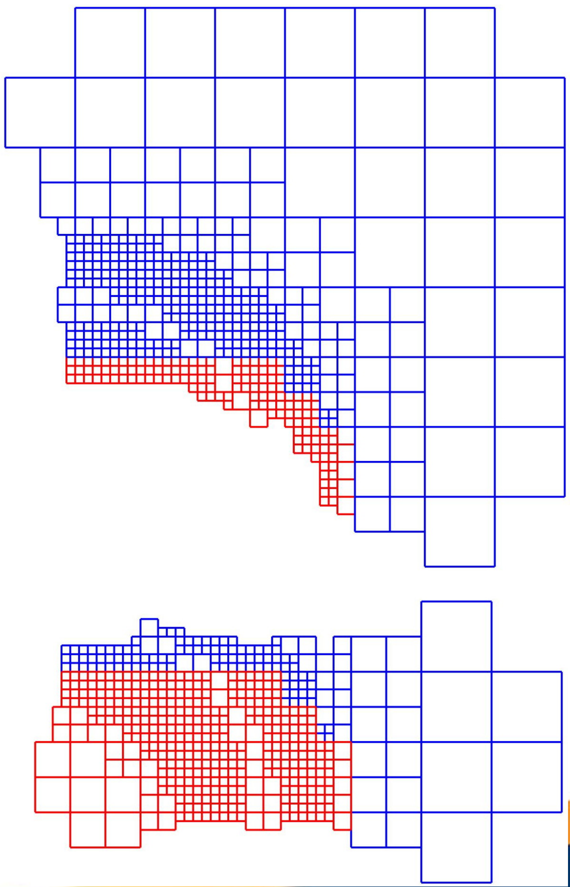
1 layer of ghost cells



2 layer of ghost cells

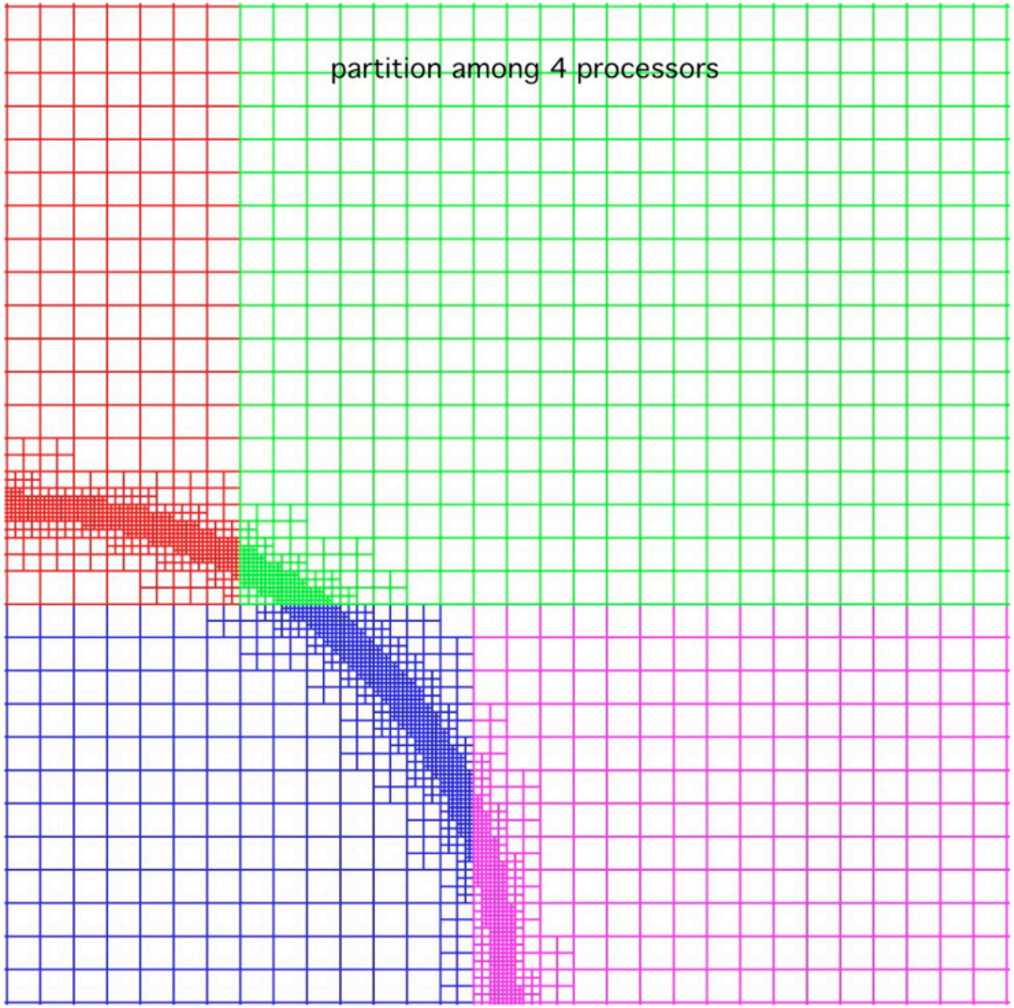


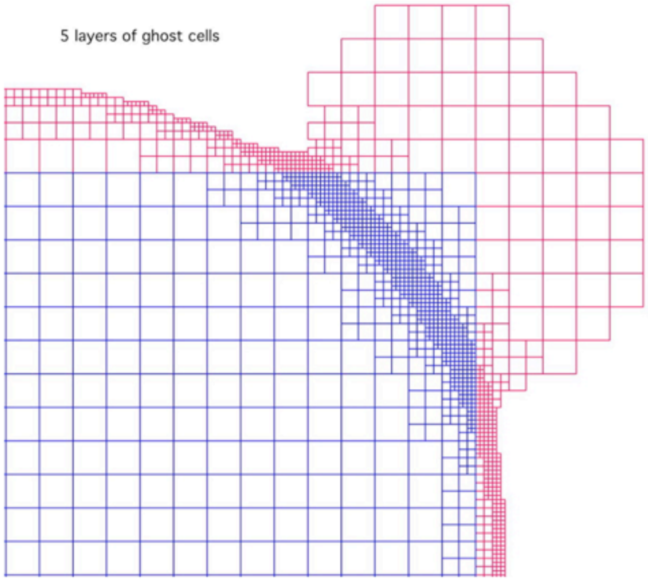
3 layers of ghost cells



UNCLASSIFIED

Mesh setup & dynamic repartition

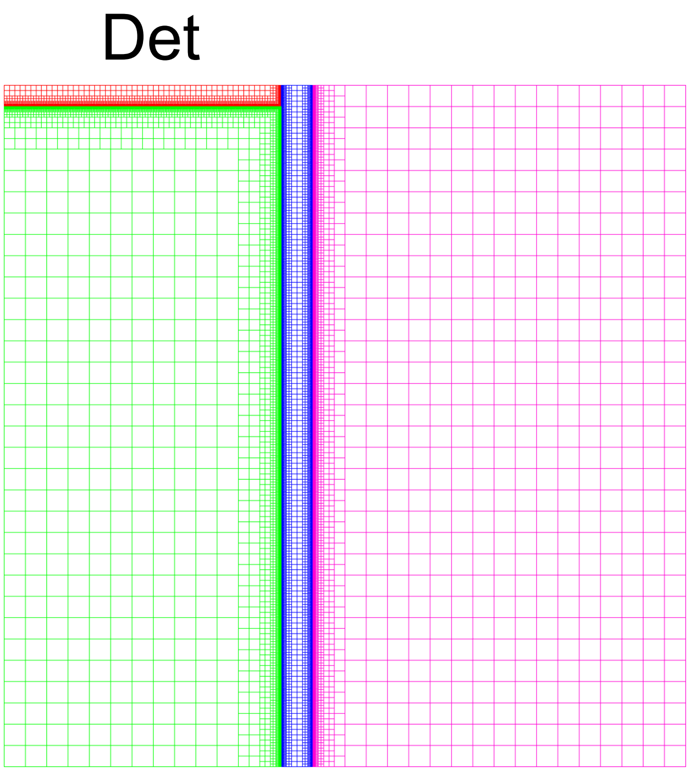




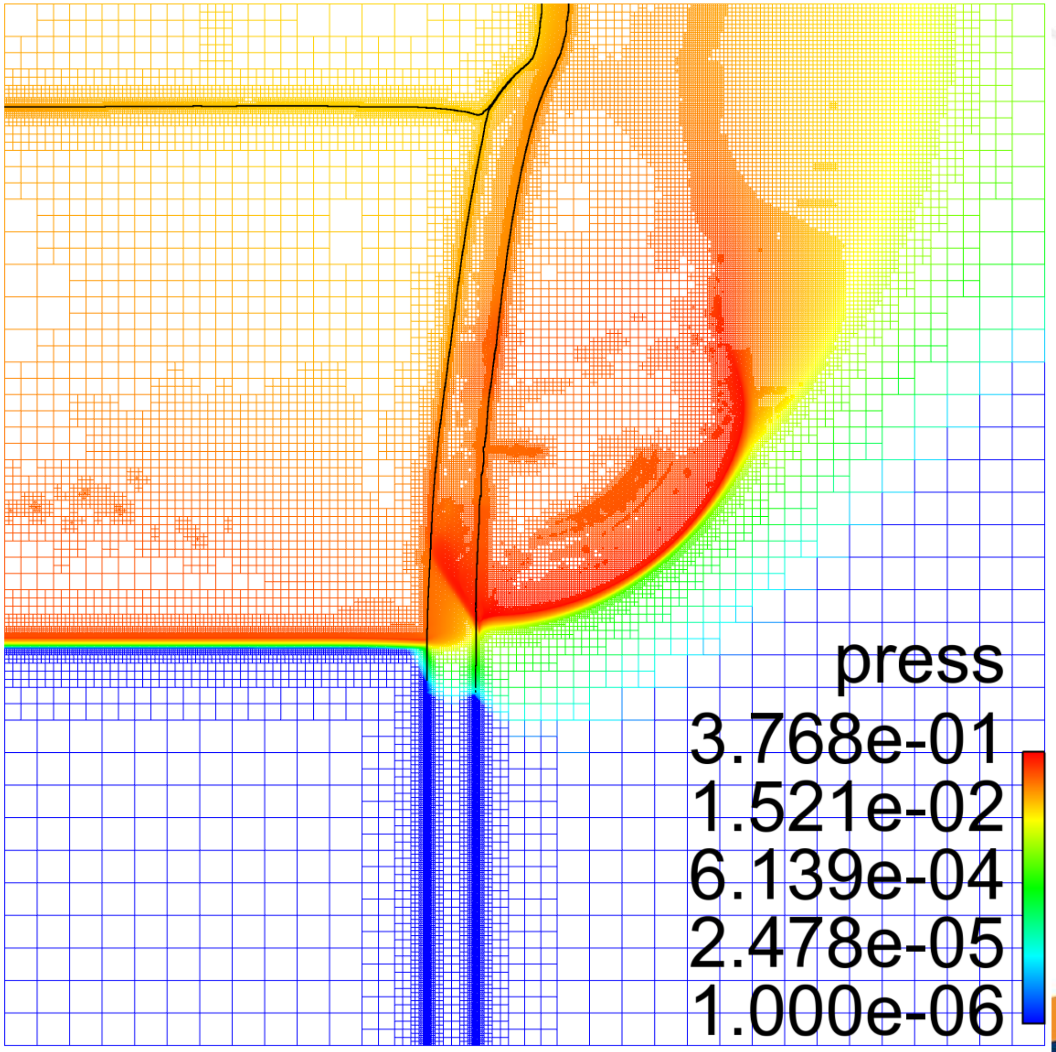
Parallelization

- Determine neighboring processors
- Identify cells which are ghost of neighboring processors.
- Pack coordinates, variables, material, isotope into buffers.
- Send each buffer to an appropriate neighboring processor.
- Receive one buffer from each of neighboring processors.
- Construct ghost cells from the buffer received from neighboring processors.
- Construct data structures on ghost cells from the buffer received from neighboring processors.
- Construct material and isotope data.

Example of Calculations



initial mesh



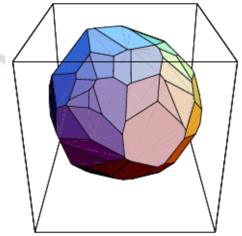
UNCLASSIFIED

IO in multi-physics code

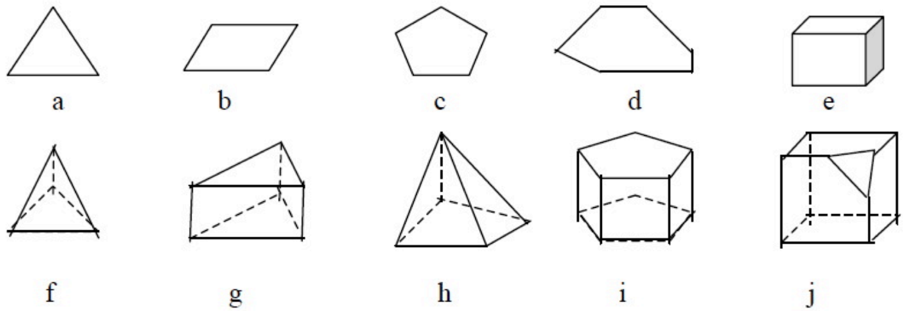
self-described, N-to-M, multi-level buffered

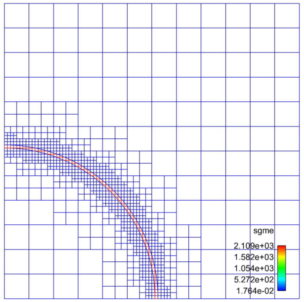
- bio_file_open(filename, filemode, iomode, fid)
- bio_file_close(fid)
- bio_attr_write(objid, attrname, datatype, size, buffer)
- bio_attr_list(objid, filter, nattr, list)
- bio_attr_read(objid, attrname, datatype, size, buffer)
- bio_write(fid, arrayname, datatype, offset, size, gsize, buffer)
- bio_list(fid, filter, to_read_value_or_not, narray, list)
- bio_read(fid, arrayname, datatype, offset, size, buffer)
- bio_buffer_init(fid, buffername, buffer_size, bufferid)
- bio_buffer_finalize(bufferid)
- bio_set_mfile(num_files)

IO for Viz and Link

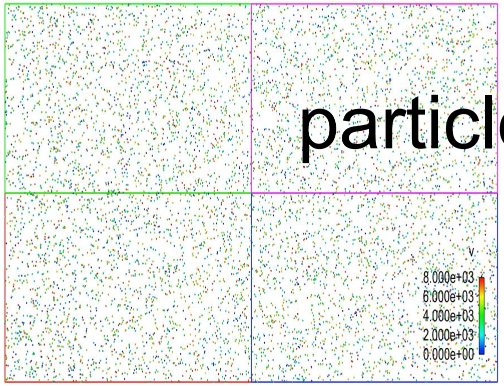


unstructured mesh





AMR



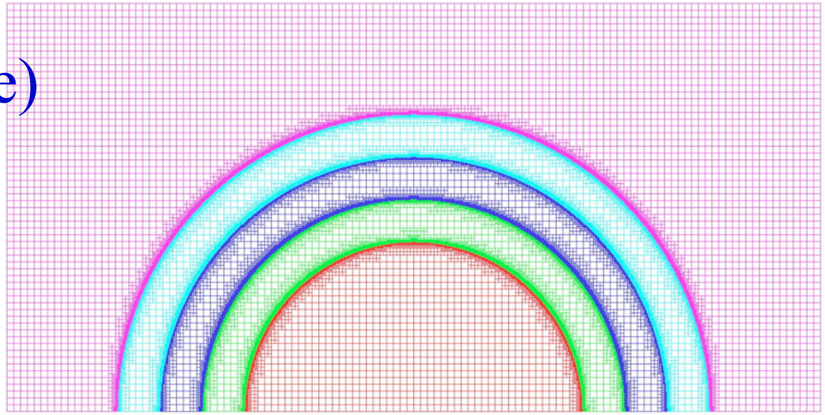
particles

- mio_open(filename, filemode, fileid)
- mio_close(fileid)
- mio_init(objtype, fileid, obj)
- mio_write(objtype, fileid, obj)
- mio_query(objtype, fileid, filter, nobjs, objs)
- mio_clean(objtype, nobjs, objs)
- mio_get_size(objtype, domain, fileid, obj)
- mio_read(objtype, domain, fileid, objs, nobjs)

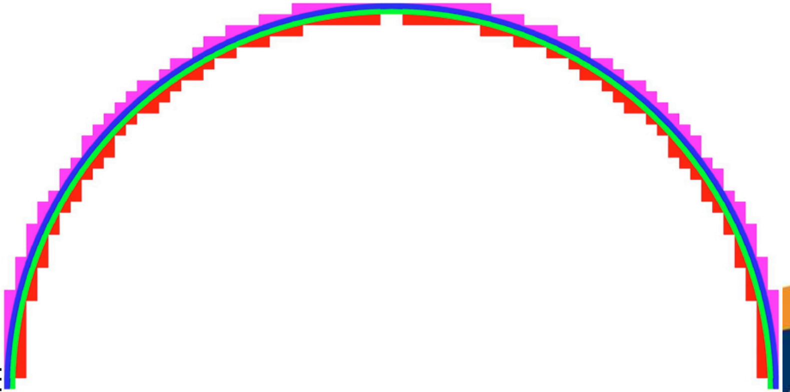
Buffering

- meta data
- problem-size data :
  each mesh associated with many variables, several meshes

mio_init_buffer(fid, bufname, bufsize)



- No change is usage
- Reduce many writes into one



Query/Read

File structure without buffering of problem-size data:

header, pe0_var1, pe1_var1, ... pe0_var2, pe1_var2, ..., tail

File structure with buffering of problem-size data:

header, pe0_var1, pe0_var2, ... pe1_var1, pe1_var2, ..., tail

- Read all meta data and file structure once when the file is open
- Read dada between two processors
- Read data for restarting calculations : the whole original buffer
- Read data for visualization : read variable by variables

mio_buffer_mode(flag_for_buffered_read)

Testing

- Single PE on different machines

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

- 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.

Correctness of Parallelization

Difference between parallel and serial calculations

$$diff < 10^{-40}$$

- Implicit calculations excluded
- On same machine
- Compiling with -g

Plasma 3-T radiation diffusion

$$\frac{\partial a T_r^4}{\partial t} = -\nabla \cdot \vec{F}_r + S_r,$$

$$C_{ve} \frac{\partial T_e}{\partial t} = -\nabla \cdot \vec{F}_e - S_r + S_e,$$

$$C_{vp} \frac{\partial T_p}{\partial t} = -\nabla \cdot \vec{F}_p - S_e.$$

$$\vec{F}_r \equiv -\sigma_r \nabla T_r^4,$$

$$\vec{F}_e \equiv -\sigma_e \nabla T_e$$

$$\vec{F}_p \equiv -\sigma_p \nabla T_i$$

$$S_r \equiv ac\rho\kappa_p (T_e^4 - T_r^4)$$

$$S_e \equiv C_{ve}\kappa_{ie}(T_p - T_e)$$

a : radiation constant.

C_{ve}, C_{vp} : heat capacities.

$\sigma_r, \sigma_e, \sigma_p$: heat conductivities.

κ_p : material absorption coefficient.

κ_{pe} : coeffiecient for interaction.

Issues in plasma 3-T radiation diffusion

- nonlinearity
- linearization: T or T^4
- treatment of source
- order of accuracy
- material discontinuity
- operator-split or not
- system of algebraic Eqs

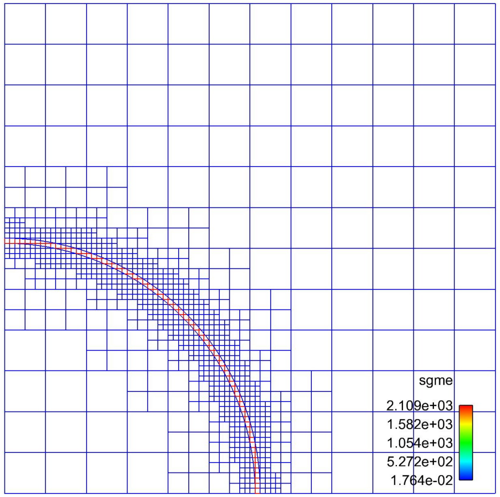
$$\frac{\partial a T_r^4}{\partial t} = -\nabla \cdot \vec{F}_r + S_r,$$

$$C_{ve} \frac{\partial T_e}{\partial t} = -\nabla \cdot \vec{F}_e - S_r + S_e,$$

$$C_{vp} \frac{\partial T_p}{\partial t} = -\nabla \cdot \vec{F}_p - S_e.$$

$$S_r \equiv ac\rho\kappa_p (T_e^4 - T_r^4)$$

$$S_e \equiv C_{ve}\kappa_{ie}(T_p - T_e)$$



Desired Features for Solver

- accurate treatment for arbitrary material discontinuity
- second order accurate in time (and space)
- any size of time step & formulation for correct steady state

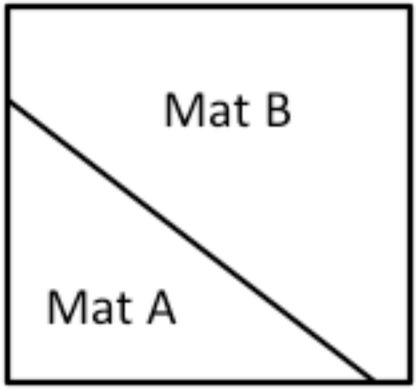
- Dai & Woodward, *Numerical simulations for nonlinear heat transfer in systems of multi-materials*, JCP 139, 58-78, 1998.
- Dai & Woodward, *A second-order iterative implicit-explicit hybrid scheme for hyperbolic systems of conservation laws*, JCP, 1996.
- Dai & Scannapieco, Second-order Accurate Interface- and Discontinuity-aware Diffusion Solvers in Two and Three Dimensions, JCP, 2015.

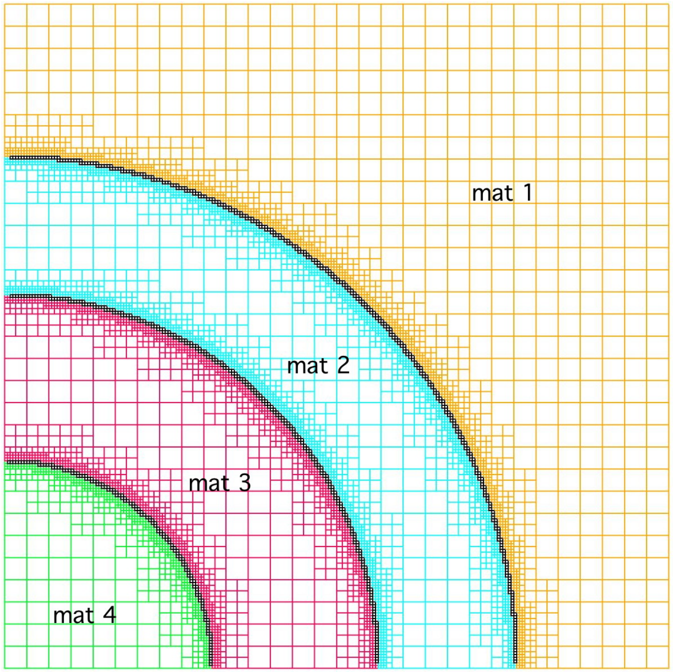
UNCLASSIFIED

Plasma 3-T radiation diffusion

Why interface reconstruction for T diffusion

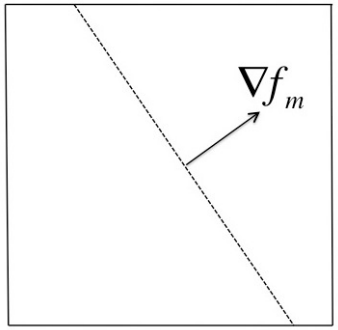
- Non-equilibrium sub-cells
- EOS unavailable for mixture of materials
- Temperature-sensitive subsequent physics

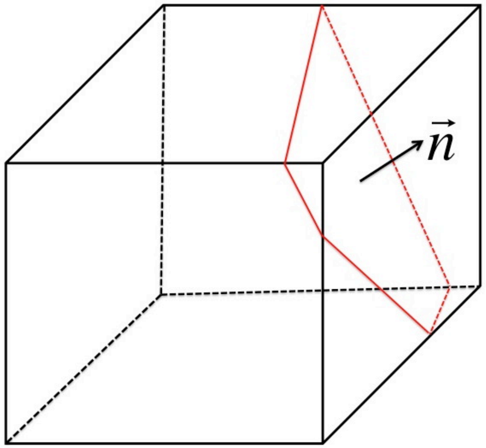




Procedure of interface reconstruction

- Find gradient of each material within a mixed cell
- Determine one gradient of the cell through *q(m)*
- Find the orientation of the interfaces
- Determine the order of materials *p(m)*
- Move the interfaces to uniquely determine locations

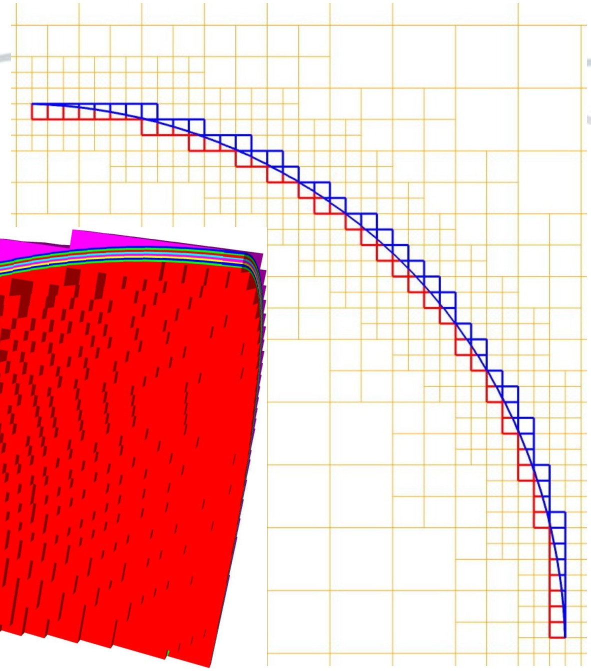


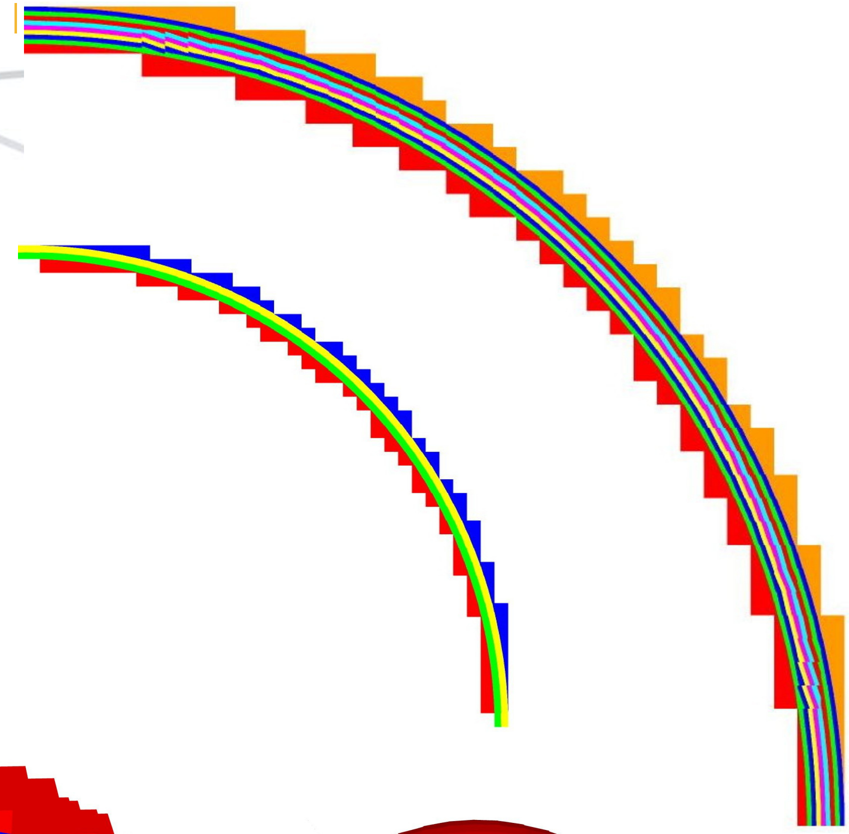


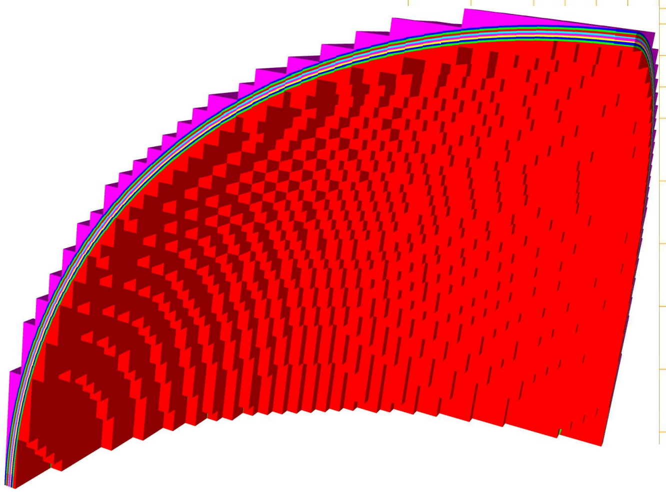
$$q(m) \equiv \| \nabla f_m \|^2 \sqrt{f_m}.$$

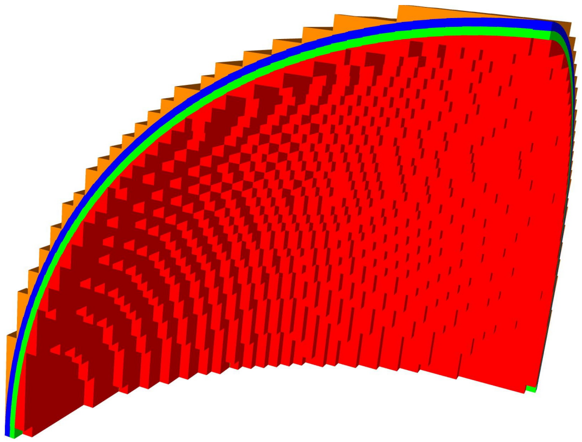
$$p(m) \equiv \vec{n}(m) \cdot \vec{n}(m_0).$$

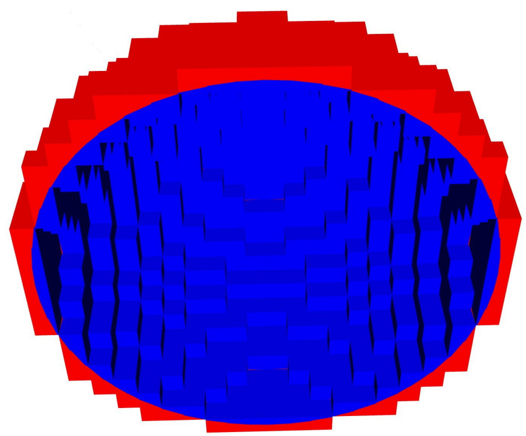
Examples

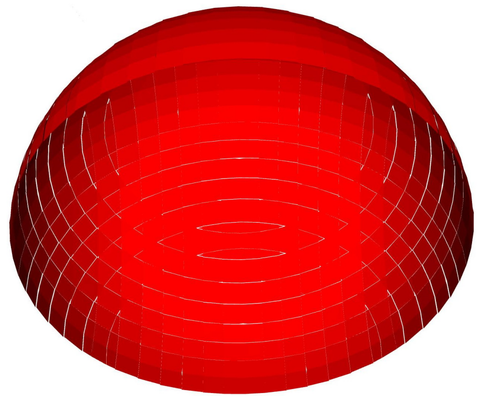












$$\frac{\partial u}{\partial t} = -\nabla \cdot \vec{F} + S.$$

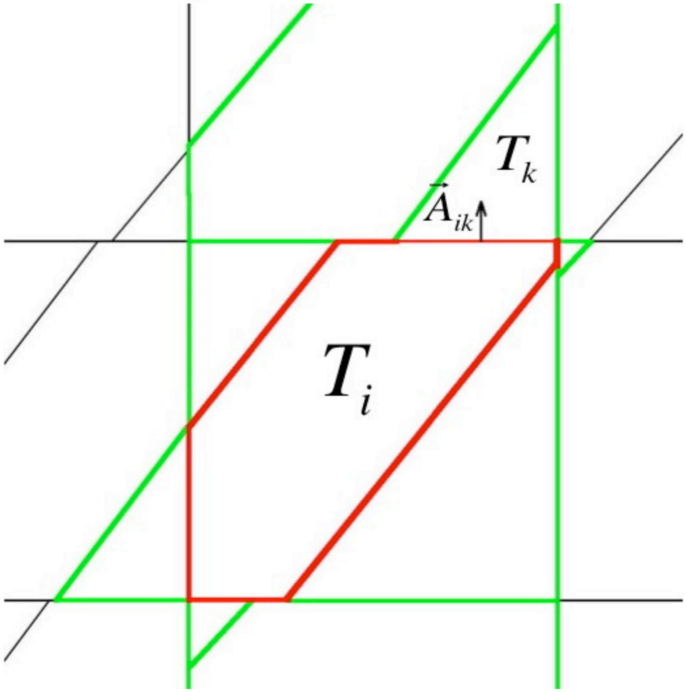
Integrate Eq. over ΔV and Δt,

$$u_i^n = u_i - \frac{\Delta t}{\Delta V_i} \sum_k \overline{F}_{ik} A_{ik} + \overline{S}_i \Delta t.$$

$$u_i^n \equiv \frac{1}{\Delta V_i} \int_{\Delta V_i} u(\Delta t, \vec{r}) dV.$$

$$\overline{S}_i \equiv \frac{1}{\Delta t \Delta V_i} \int_0^{\Delta t} \int_{\Delta V_i} S(t, \vec{r}) dV dt.$$

$$\overline{F}_{ik} \equiv \frac{1}{\Delta t} \int_0^{\Delta t} \left[\frac{1}{\Delta A_{ik}} \int_{A_{ik}} \vec{F} \cdot d\vec{a} \right] dt.$$



Coupled nonlinear difference equations

$$a(T_r^4)_i^n + \frac{\Delta t}{\Delta V_i}\left(\sum_k \kappa_{ik}^h A_{ik}\right)T_{ri}^h = a(T_r^4)_i + \frac{\Delta t}{\Delta V_i}\sum_k(\kappa_{ik}^h A_{ik} T_{rk}^h) + S_{ri}^h \Delta t$$

$$a(T_r^4)_i^h + \frac{3\Delta t}{4\Delta V_i}\left(\sum_k \kappa_{ik}^h A_{ik}\right)T_{ri}^h - \frac{\Delta t}{4\Delta V_i}\left(\sum_k \kappa_{ik}^n A_{ik}\right)T_{ri}^n$$

$$= a(T_r^4)_i + \frac{3\Delta t}{4\Delta V_i}\sum_k(\kappa_{ik}^h A_{ik} T_{rk}^h) - \frac{\Delta t}{4\Delta V_i}\sum_k(\kappa_{ik}^n A_{ik} T_{rk}^n) + \frac{1}{2}\Delta t \bar{S}_{ri}^h$$

$$C_{ei}T_{ei}^n + \frac{\Delta t}{\Delta V_i}\left(\sum_k \sigma_{eik}^h A_{ik}\right)T_{ei}^h = C_{ei}T_{ei} + \frac{\Delta t}{\Delta V_i}\sum_k(\sigma_{eik}^h A_{ik} T_{ek}^h) - \Delta t(S_{ri}^h - S_{ei}^h)$$

$$C_{ei}T_{ei}^h + \frac{3\Delta t}{4\Delta V_i}\left(\sum_k \sigma_{eik}^h A_{ik}\right)T_{ei}^h - \frac{\Delta t}{4\Delta V_i}\left(\sum_k \sigma_{eik}^n A_{ik}\right)T_{ei}^n = C_{ei}T_{ei} + \frac{3\Delta t}{4\Delta V_i}\sum_k(\sigma_{eik}^h A_{ik} T_{ek}^h) - \frac{\Delta t}{4\Delta V_i}\sum_k(\sigma_{eik}^n A_{ik} T_{ek}^n) - \frac{1}{2}\Delta t(\bar{S}_{ri}^h - \bar{S}_{ei}^h)$$

$$C_{pi}T_{pi}^n + \frac{\Delta t}{\Delta V_i}\left(\sum_k \sigma_{pik}^h A_{ik}\right)T_{pi}^h = C_{pi}T_{pi} + \frac{\Delta t}{\Delta V_i}\sum_k(\sigma_{pik}^h A_{ik} T_{pk}^h) - S_{pi}^h \Delta t$$

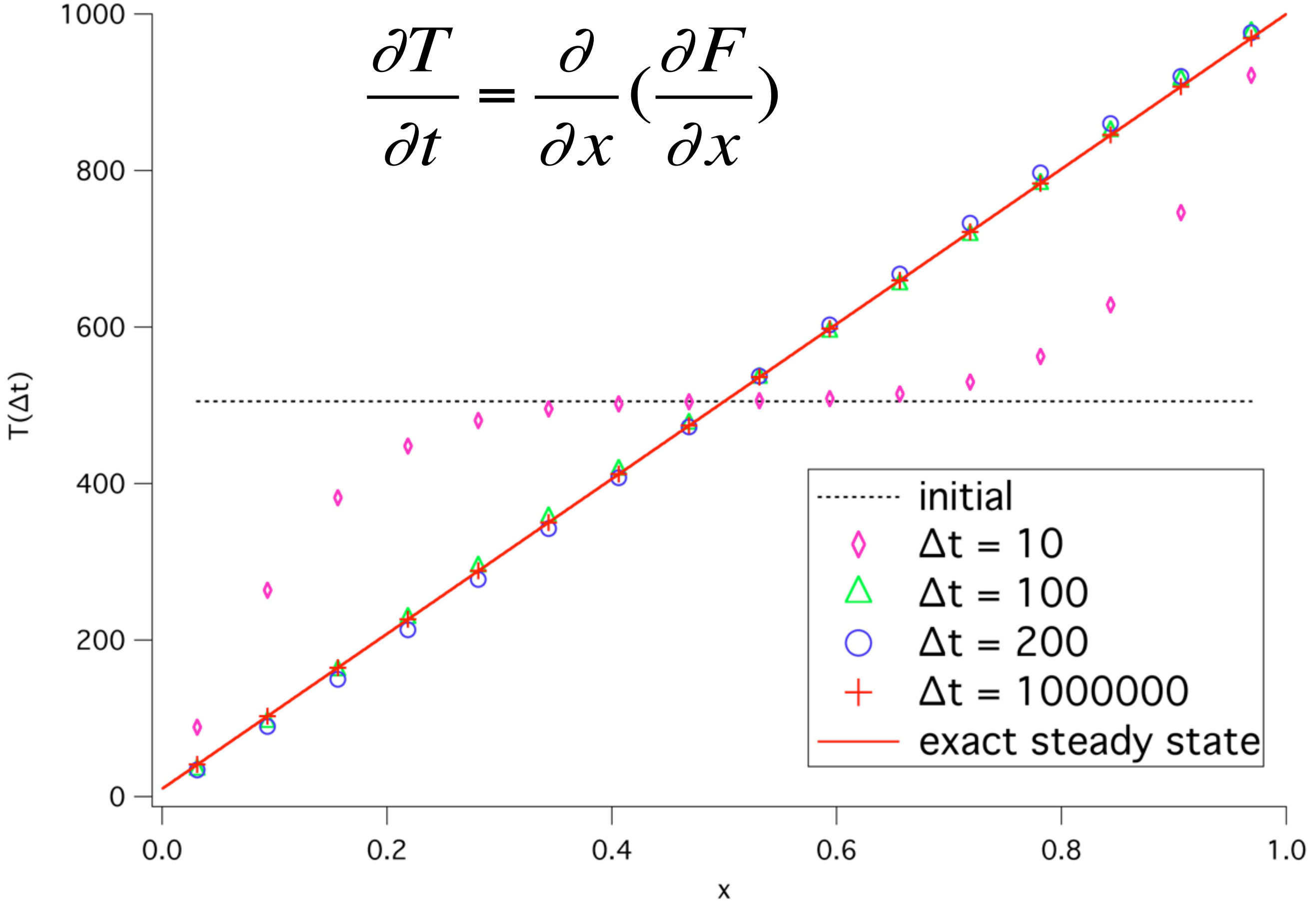
$$C_{pi}T_{pi}^h + \frac{3\Delta t}{4\Delta V_i}\left(\sum_k \sigma_{pik}^h A_{ik}\right)T_{pi}^h - \frac{\Delta t}{4\Delta V_i}\left(\sum_k \sigma_{pik}^n A_{ik}\right)T_{pi}^n = C_{pi}T_{pi} + \frac{3\Delta t}{4\Delta V_i}\sum_k(\sigma_{pik}^h A_{ik} T_{pk}^h) - \frac{\Delta t}{4\Delta V_i}\sum_k(\sigma_{pik}^n A_{ik} T_{ek}^n) - \frac{1}{2}\Delta t \bar{S}_{pi}^h$$

$$\bar{S}_i^h = \frac{3}{2}S_i^h - \frac{1}{2}S_i^n \qquad S_{ri}^h = ac\rho\kappa_p(T_{ei}^{h4} - T_{ri}^{h4}). \qquad S_{ri}^n = ac\rho\kappa_p(T_{ei}^{n4} - T_{ri}^{n4}).$$

Numerical Example

1D temperature after one time step

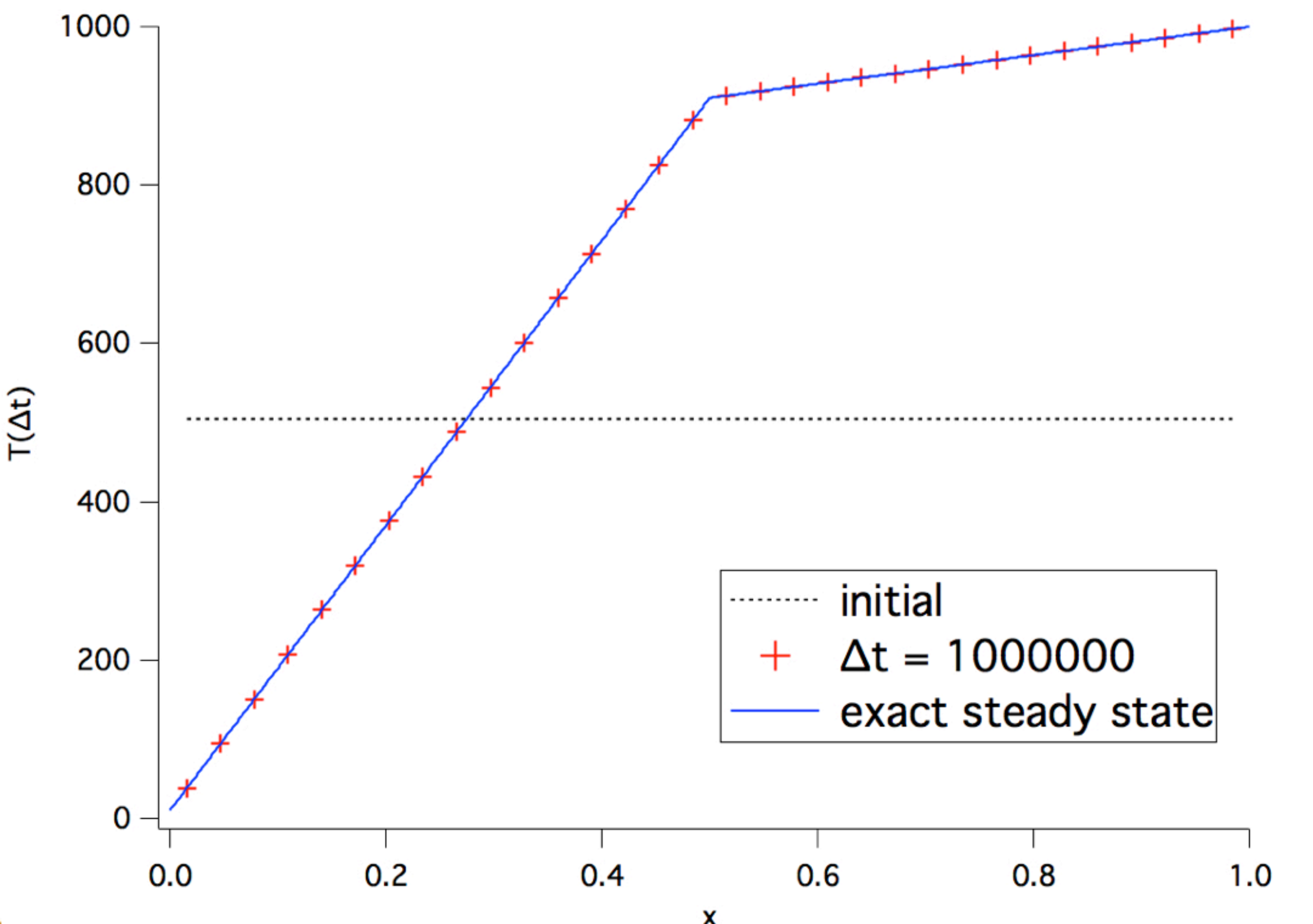
$$\frac{\partial T}{\partial t} = \frac{\partial}{\partial x}\left(\frac{\partial F}{\partial x}\right)$$



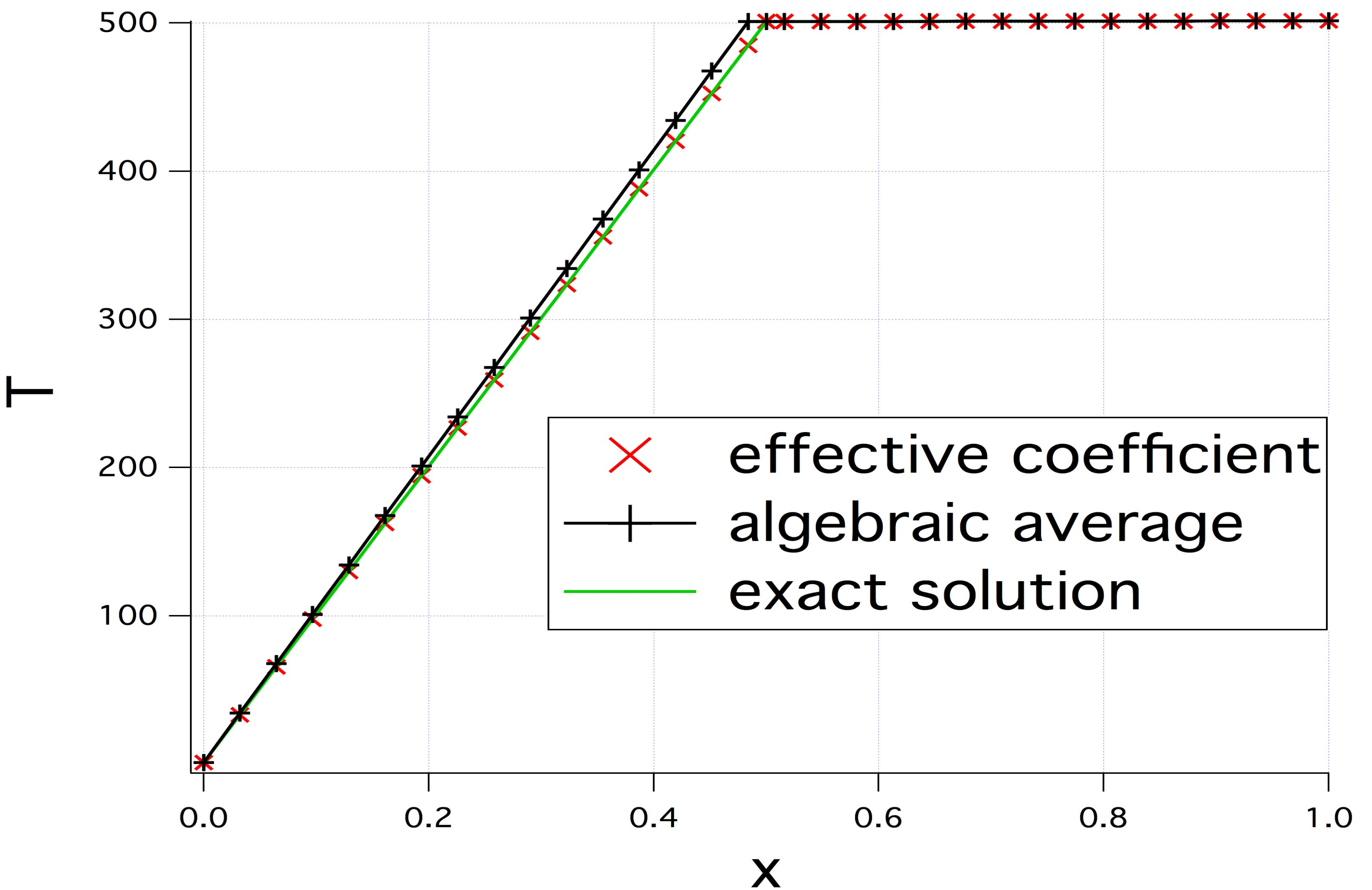
UNCLASSIFIED

Numerical Example

1D temperature after one time step, 2 mats

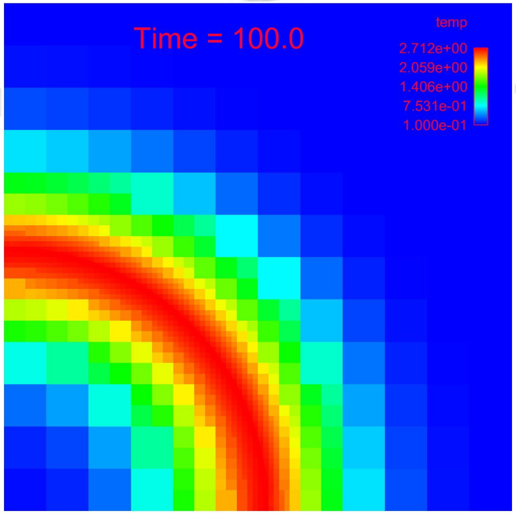


Example

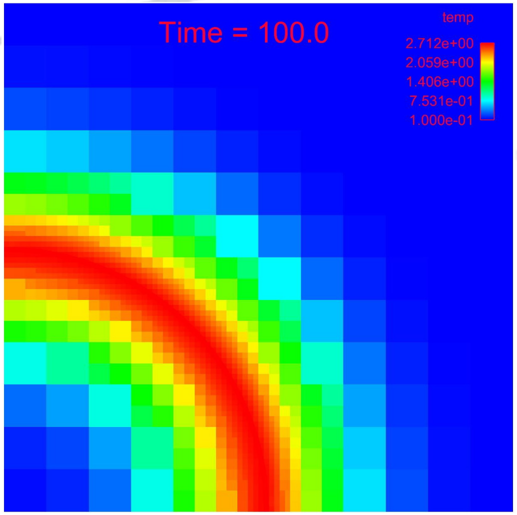


500
400
300
200
100
T
0.0
0.2
0.4
0.6
0.8
1.0
x
effective coefficient
algebraic average
exact solution

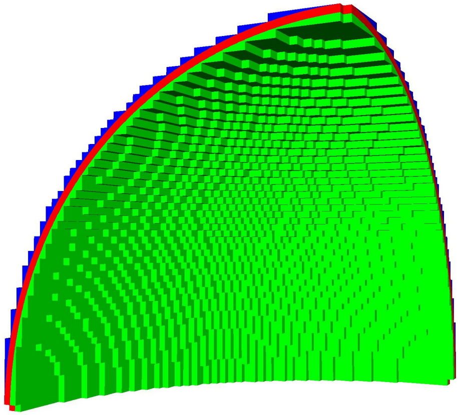
Example for different time steps

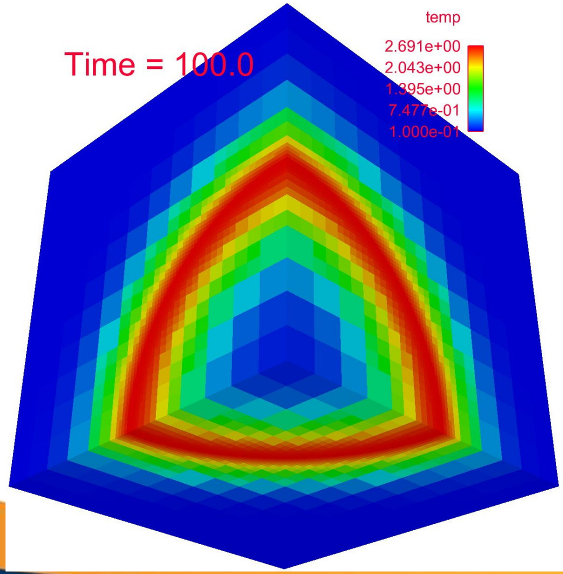


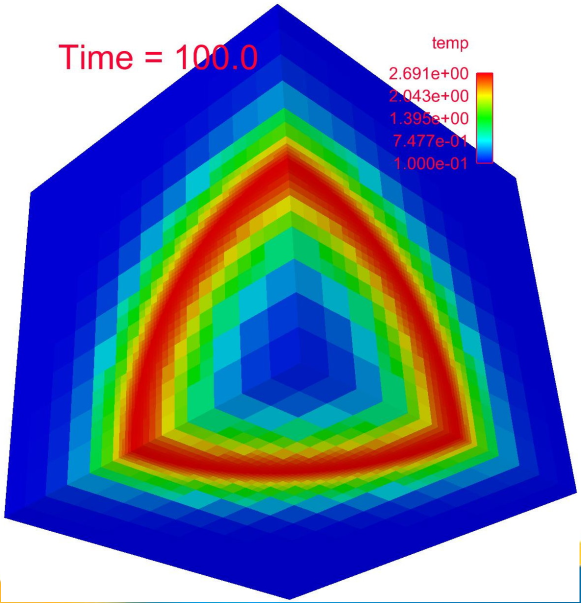
$$\Delta t = 0.001$$



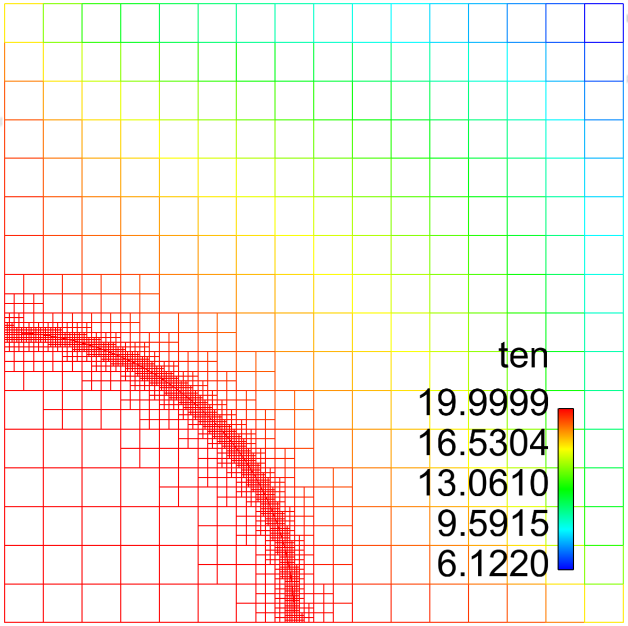
$$\Delta t = 1.0$$

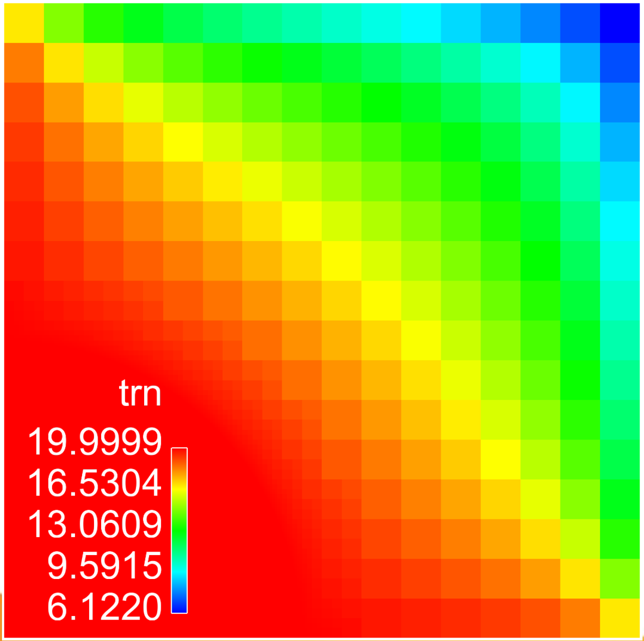


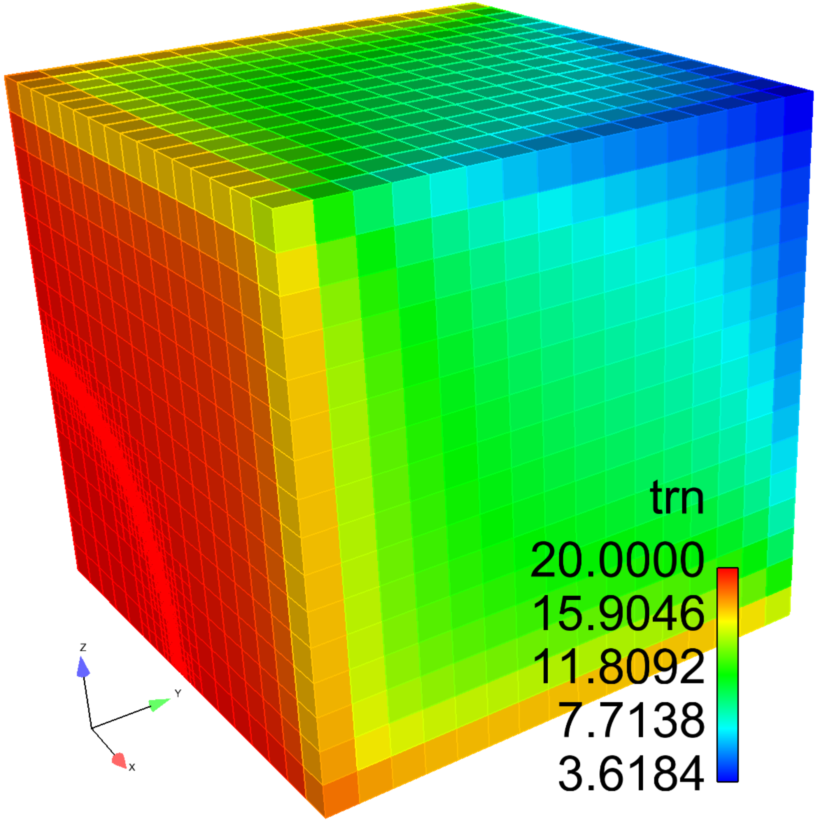




Example of Steady State of 3-T after one time step



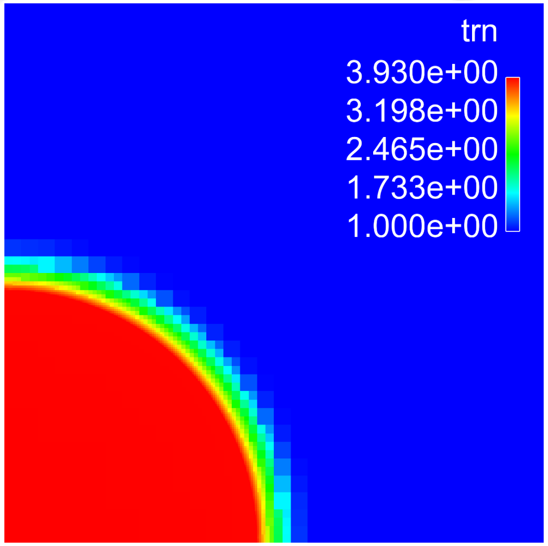




UNCLASSIFIED

Solution after one time step

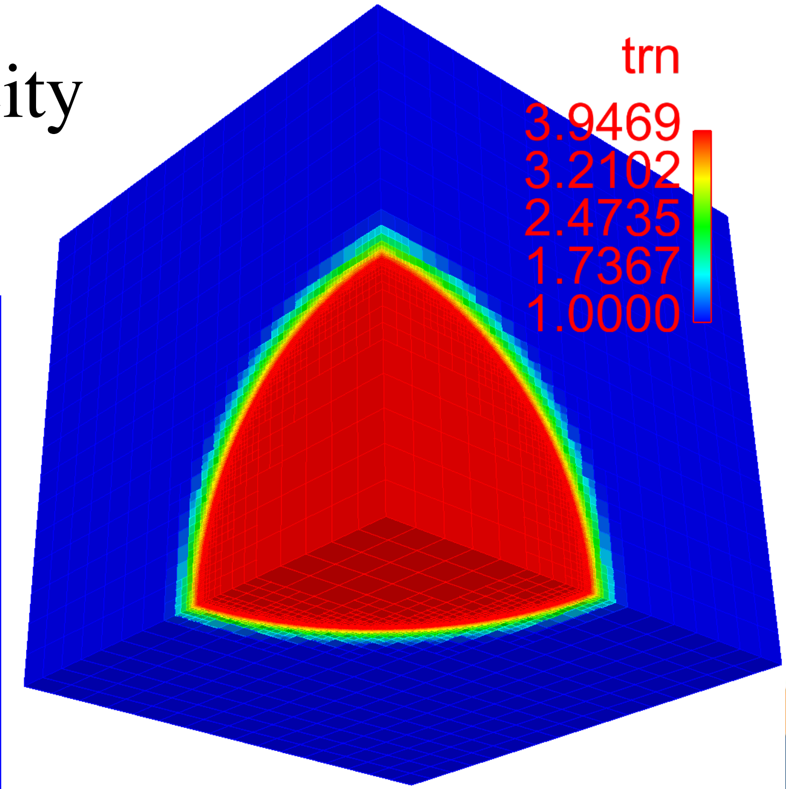
Fully nonlinear, fully coupled, fully implicit

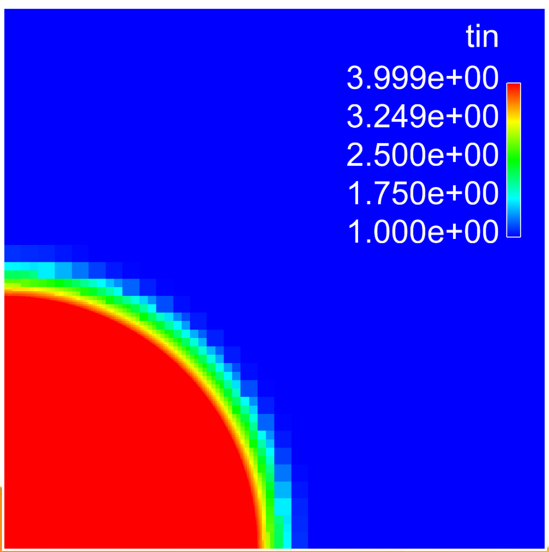


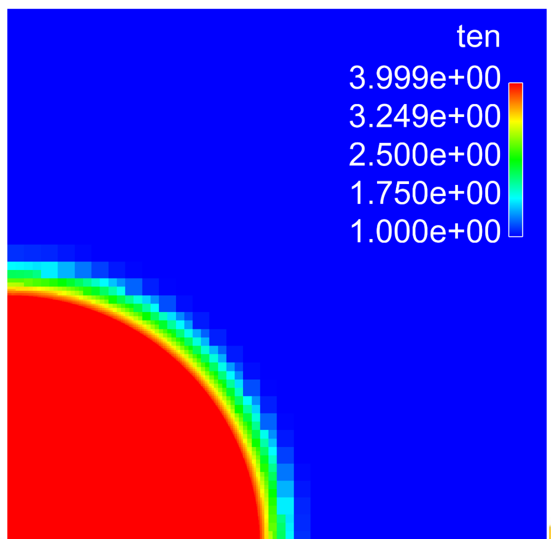
large σ_r, small σ_e, σ_p

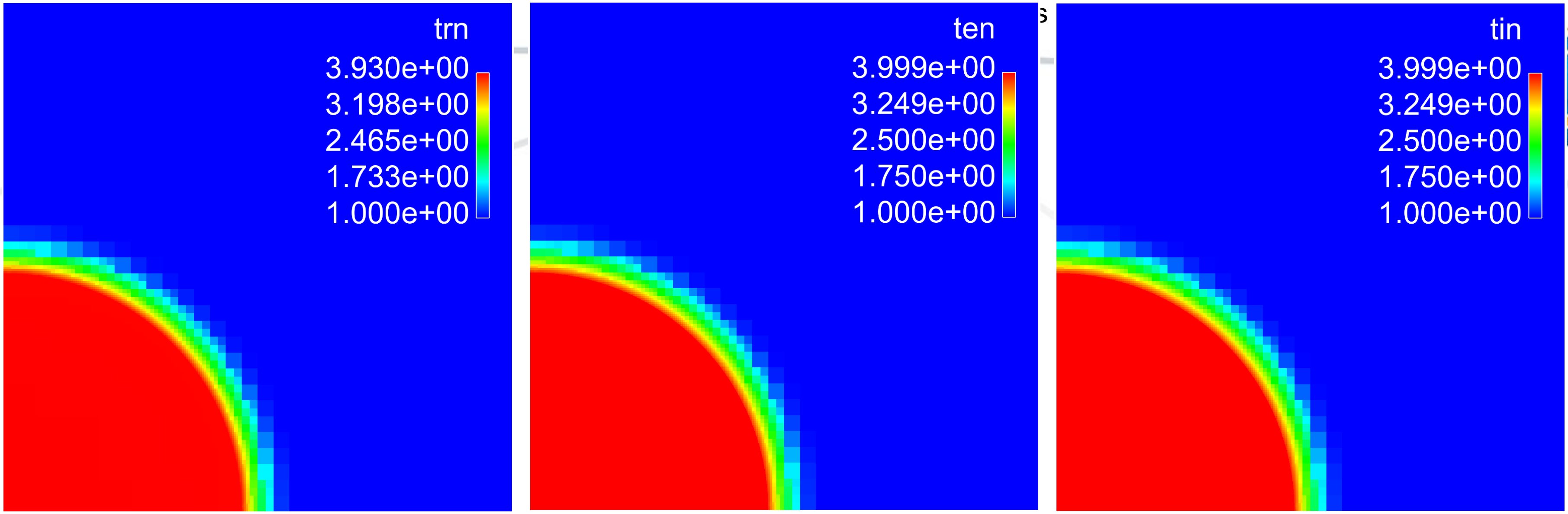
strong coupling

large heat capacity



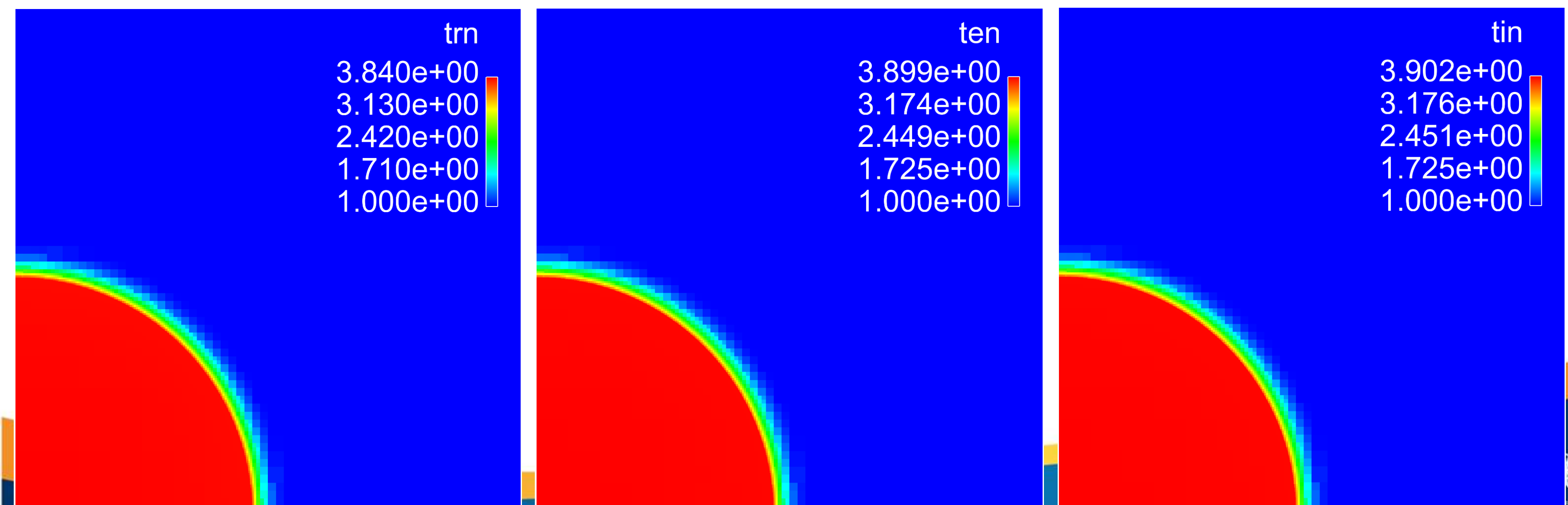






Above: fully implicit, fully coupled, and fully nonlinear

Below: fully implicit, fully coupled, but linear



Inappropriate split algorithm for this problem

Step1)

$$\frac{\partial a T_r^4}{\partial t} = -\nabla \cdot (\sigma_r \nabla T_r),$$

$$C_{ve} \frac{\partial T_e}{\partial t} = -\nabla \cdot (\sigma_e \nabla T_e),$$

$$C_{vp} \frac{\partial T_p}{\partial t} = -\nabla \cdot (\sigma_p \nabla T_p).$$

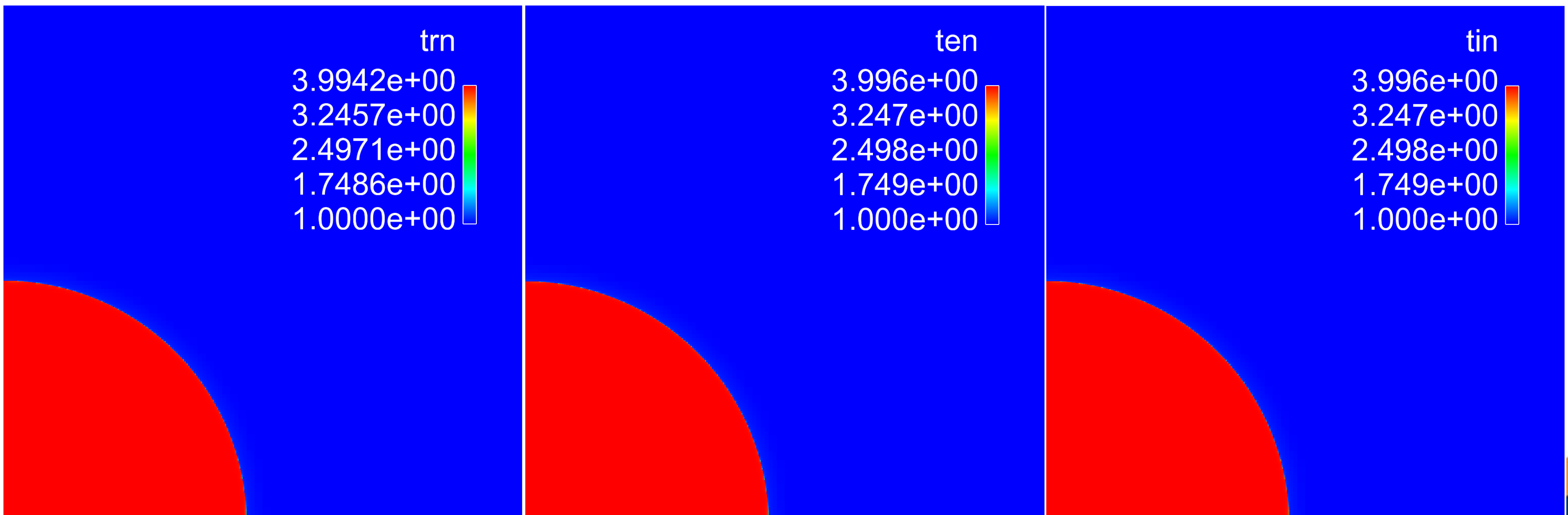
all fully implicit

Step2)

$$\frac{\partial a T_r^4}{\partial t} = ac\rho\kappa_p (T_e^4 - T_r^4),$$

$$C_{ve} \frac{\partial T_e}{\partial t} = -ac\rho\kappa_p (T_e^4 - T_r^4) + C_{ve}\kappa_{ie}(T_p - T_e),$$

$$C_{vp} \frac{\partial T_p}{\partial t} = -C_{ve}\kappa_{ie}(T_p - T_e).$$



Tr diffused, but absorbed

Another inappropriate split algorithm

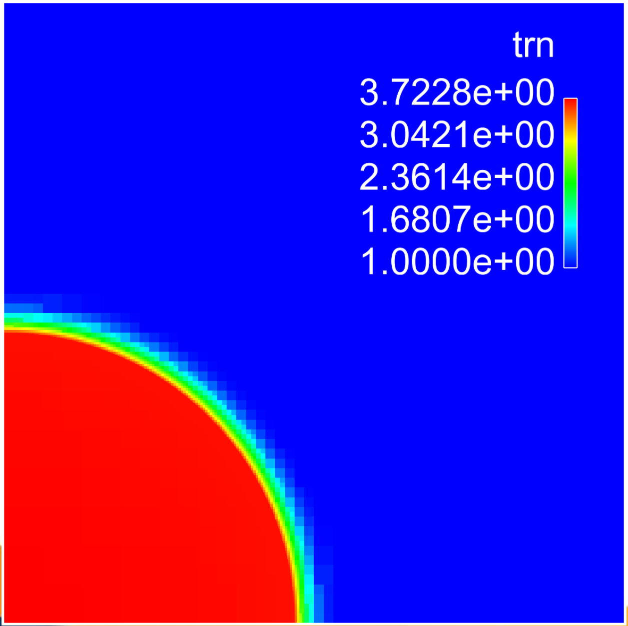
for this problem, all fully implicit

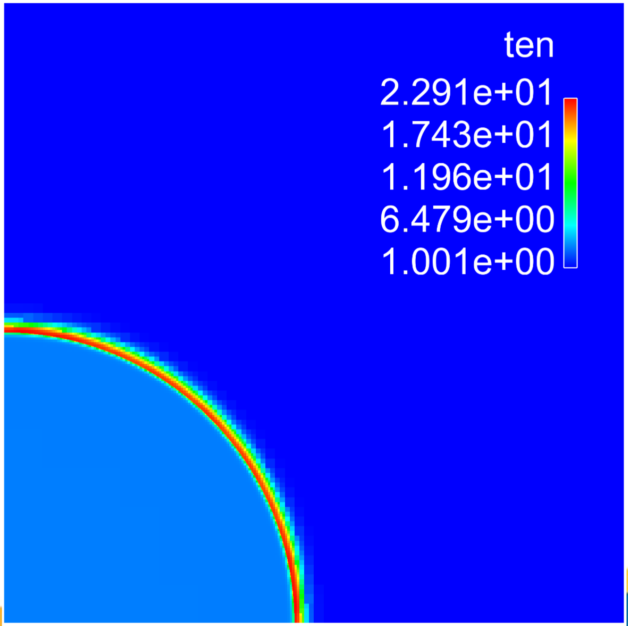
$$\frac{\partial a T_r^4}{\partial t} = -\nabla \cdot \vec{F}_r + S_r,$$
$$C_{ve} \frac{\partial T_e}{\partial t} = -S_r + \frac{2}{3} S_e,$$
$$C_{vp} \frac{\partial T_p}{\partial t} = -\frac{2}{3} S_e.$$

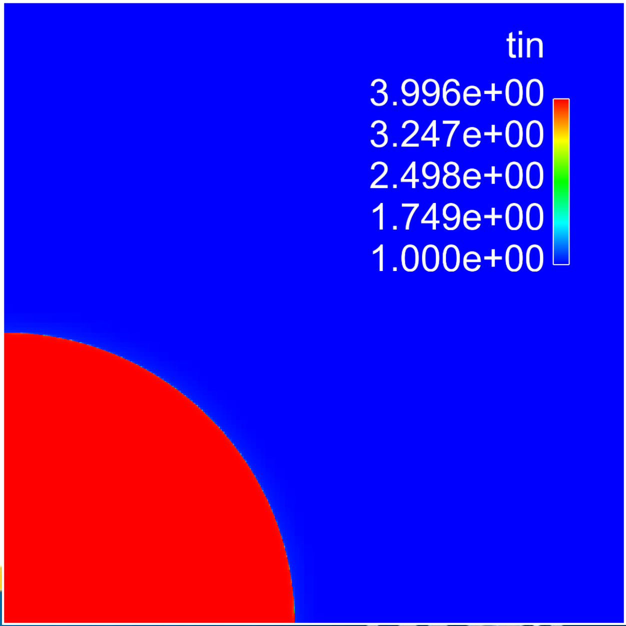
$$C_{ve} \frac{\partial T_e}{\partial t} = -\nabla \cdot \vec{F}_e + \frac{1}{3} S_e,$$
$$C_{vp} \frac{\partial T_p}{\partial t} = -\frac{1}{3} S_e.$$

$$C_{vp} \frac{\partial T_p}{\partial t} = -\nabla \cdot \vec{F}_p.$$

$$(T_r^4 - T_e^4)^n \approx (T_r^4 - T_e^4) + 4(T_e^3 T_e^n - T_r^3 T_r^n)$$







Toward advanced computer architectures

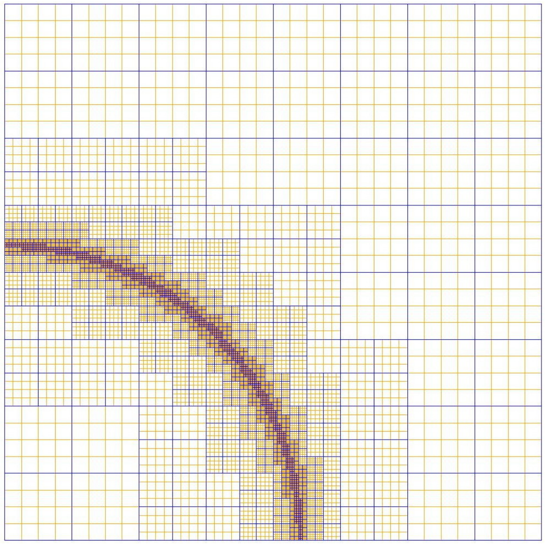
Main difficulties of cell-based AMR

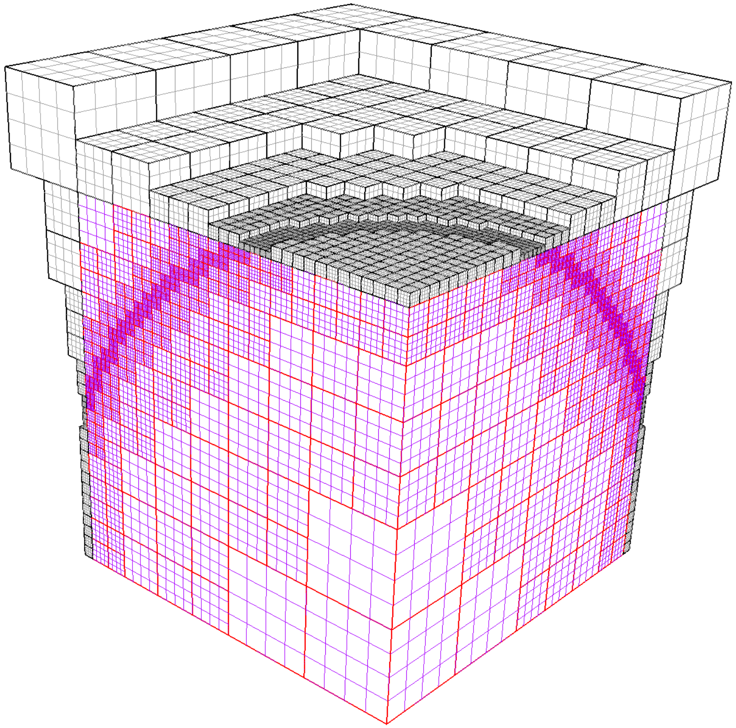
- Data locality, (long distance) indirect address.
- Difficult to implement pipeline, i.e., list of tasks (tiles).
- Unstructured aspect of physics packages.

Why briquette-based AMR?

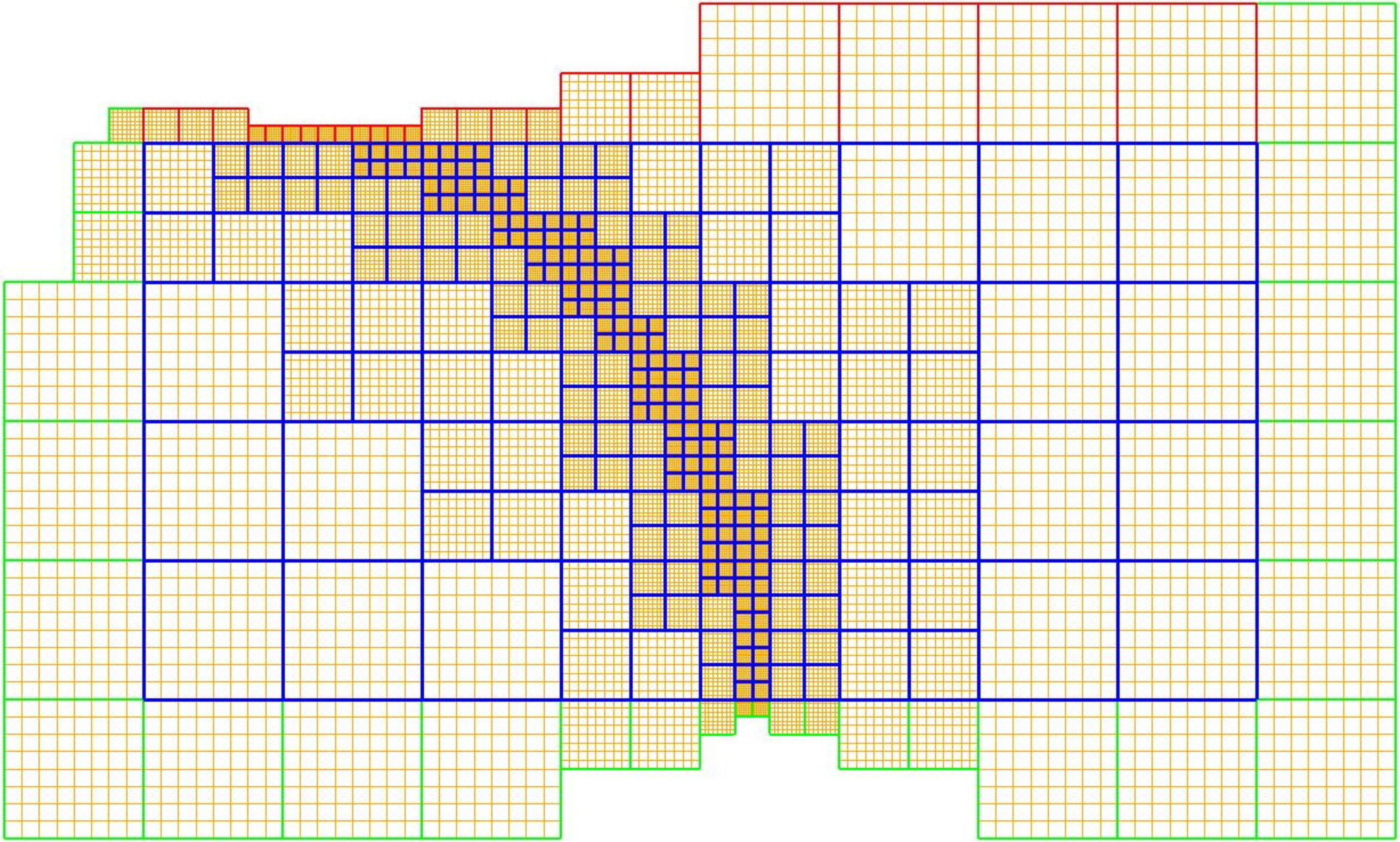
- to overcome these difficulties,
- to take advantage of current cell-based AMR.

briquette-based

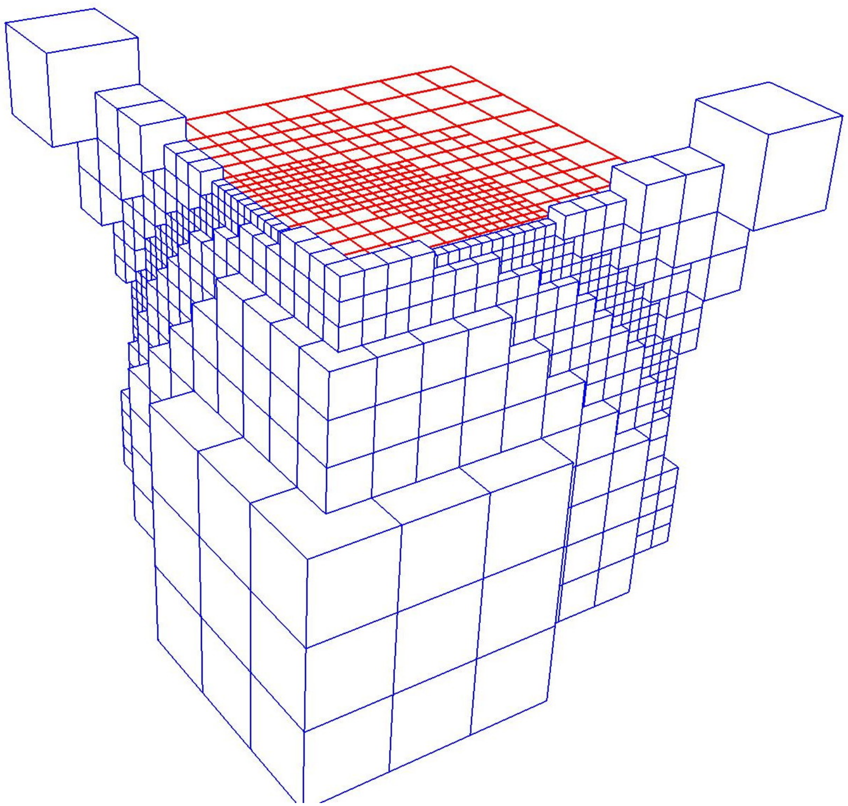


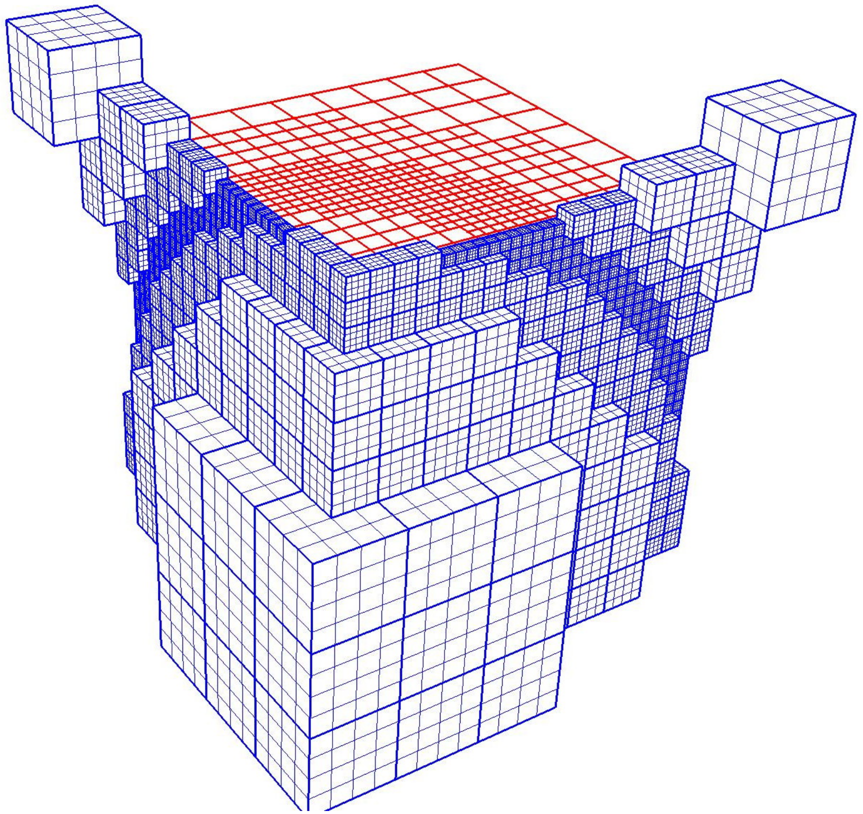


Boundary and ghost briquettes



3D briquettes in AMR with 5 levels



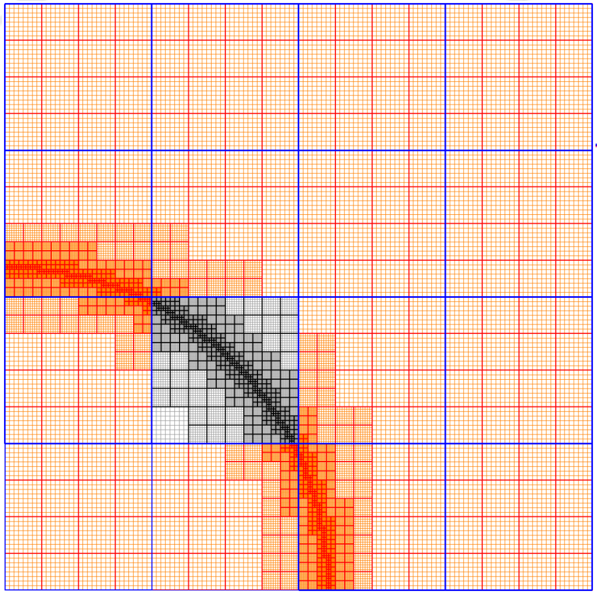


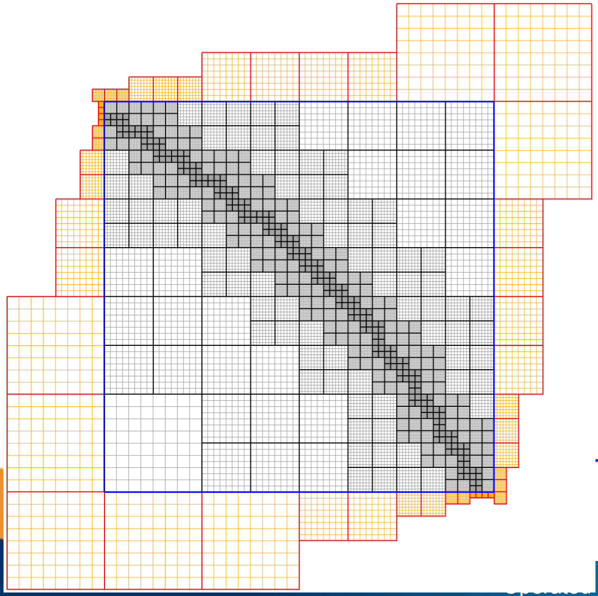
UNCLASSIFIED

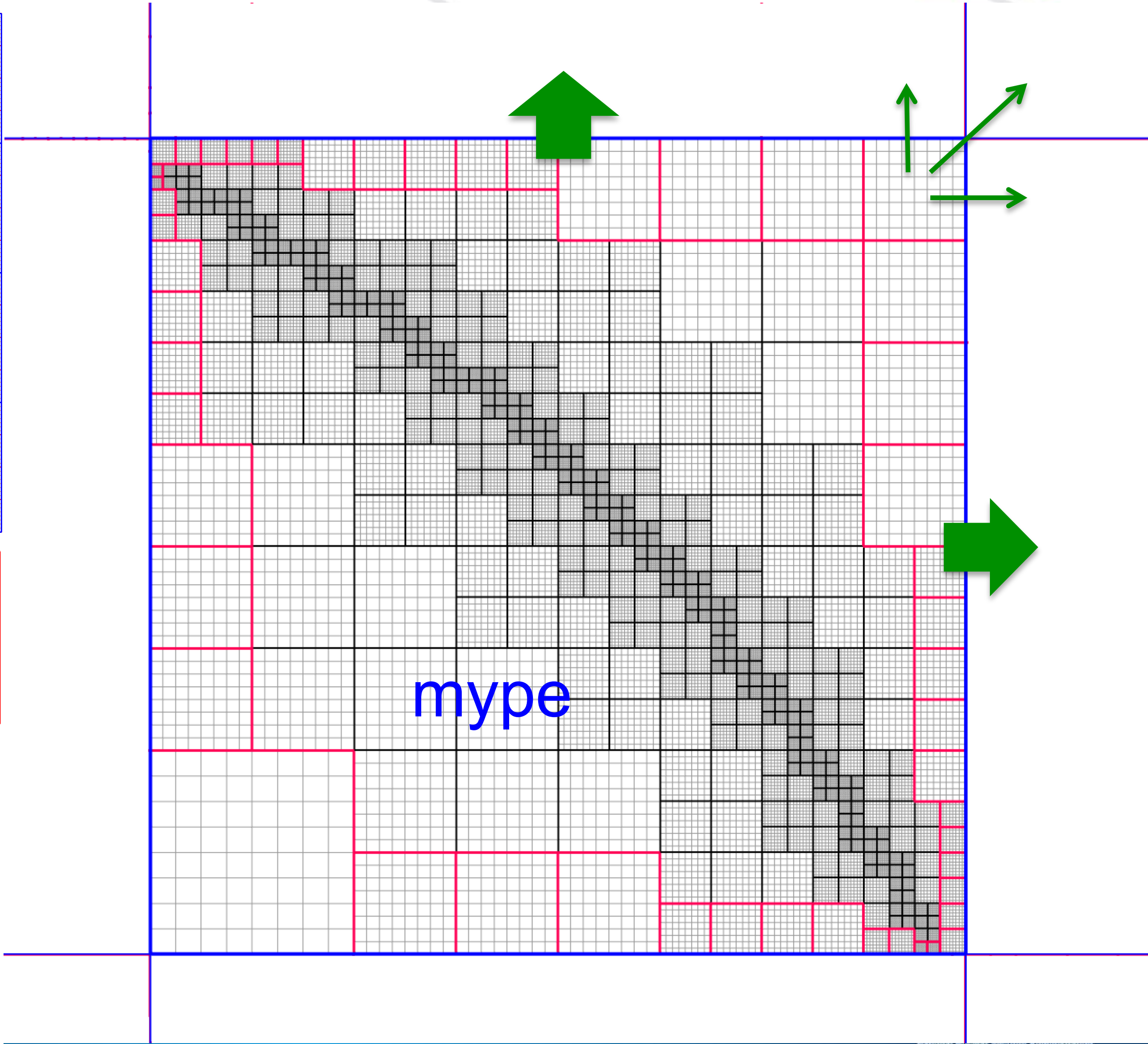
Data structures in briquette-based AMR

- (Physics) package-based.
- Structured mesh within each briquette/task/tile.
- Variable ghost cells, nghost.
- Flexible size of each task, ncell.
- Variables continuous in memory.
- Uncompressed data in physics packages.
- Option to locally un-compress or compress mat data.

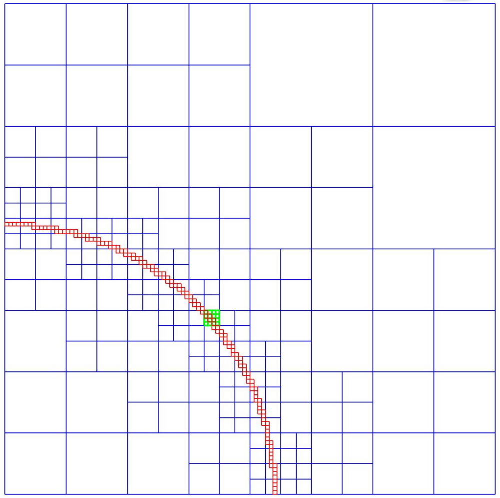
Dynamic scheduling of tasks & communications

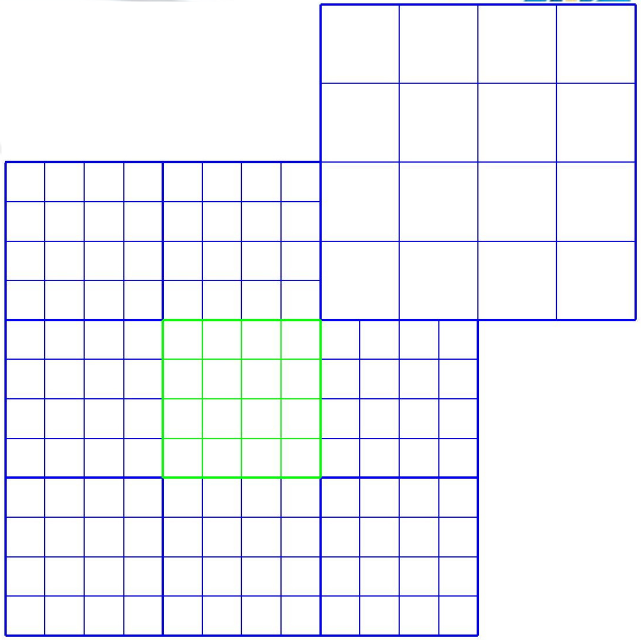


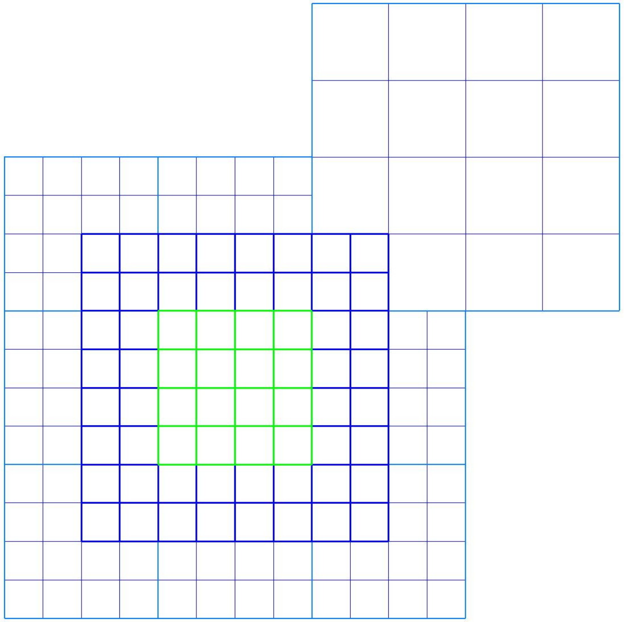


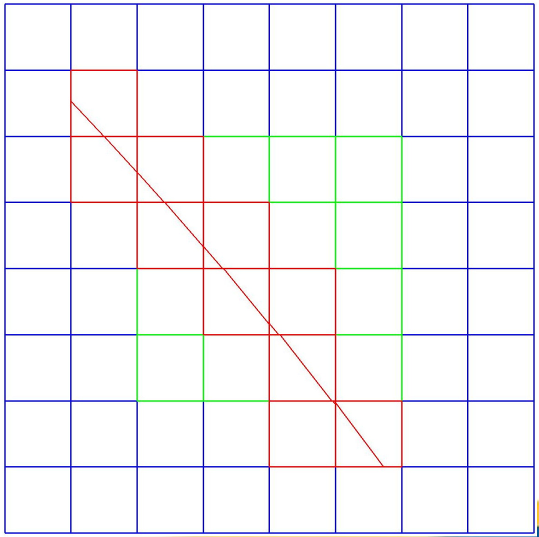


Physics on uniform mesh

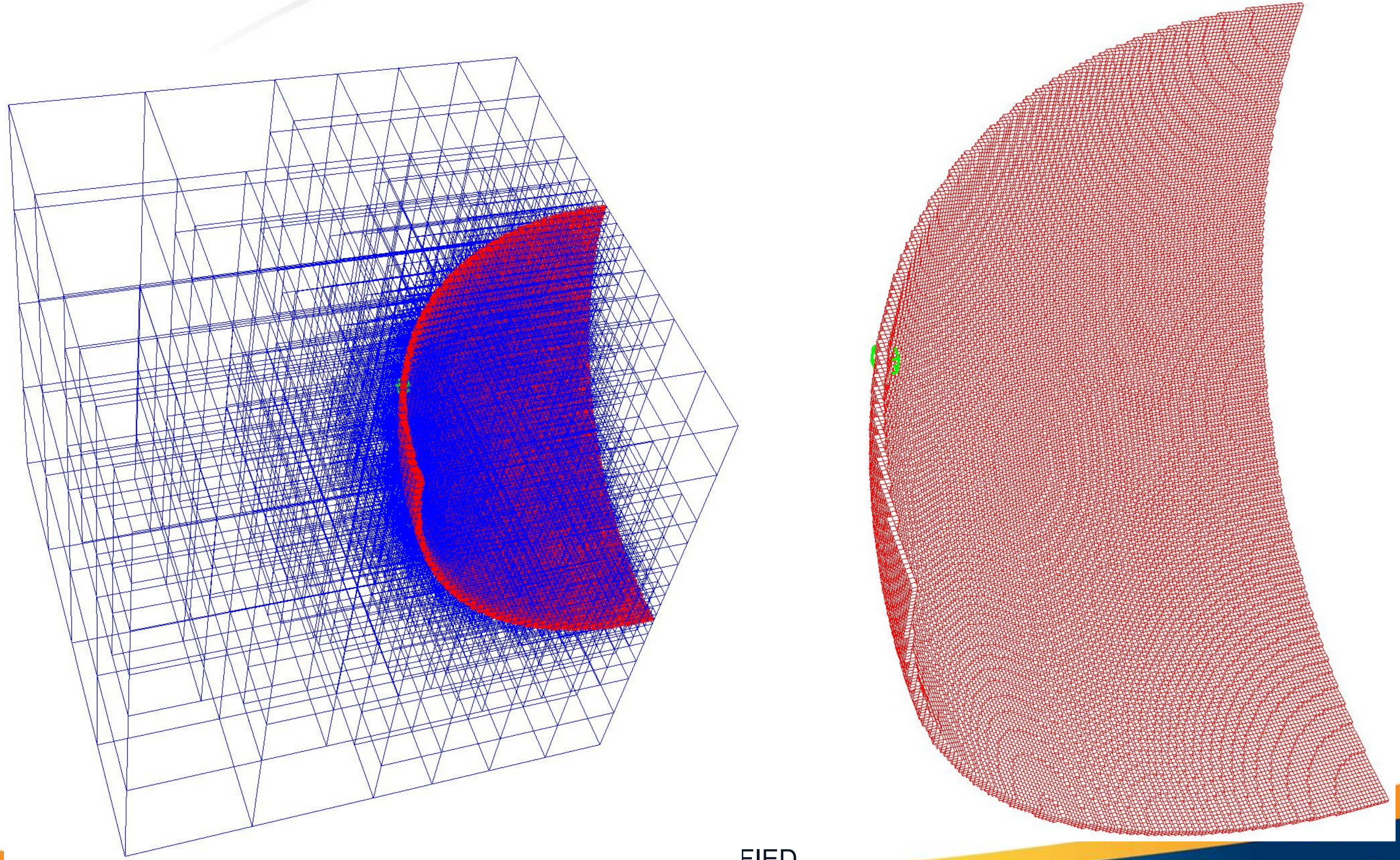






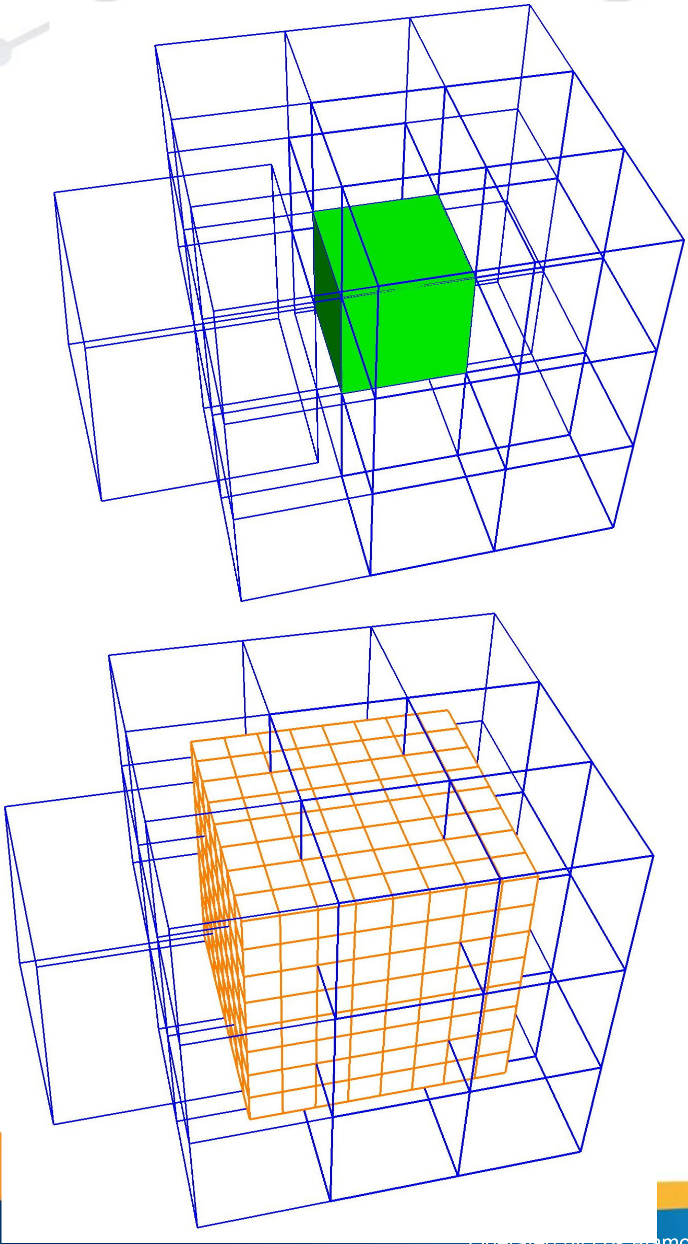


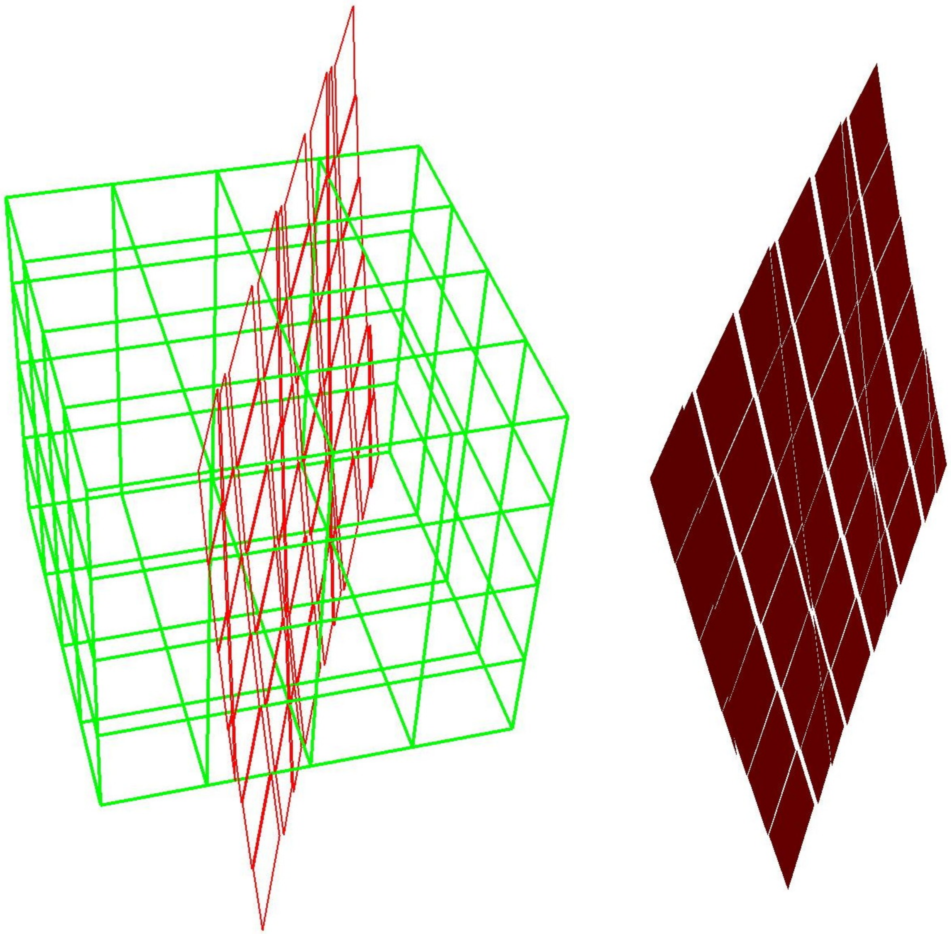
Physics on uniform mesh (3D)



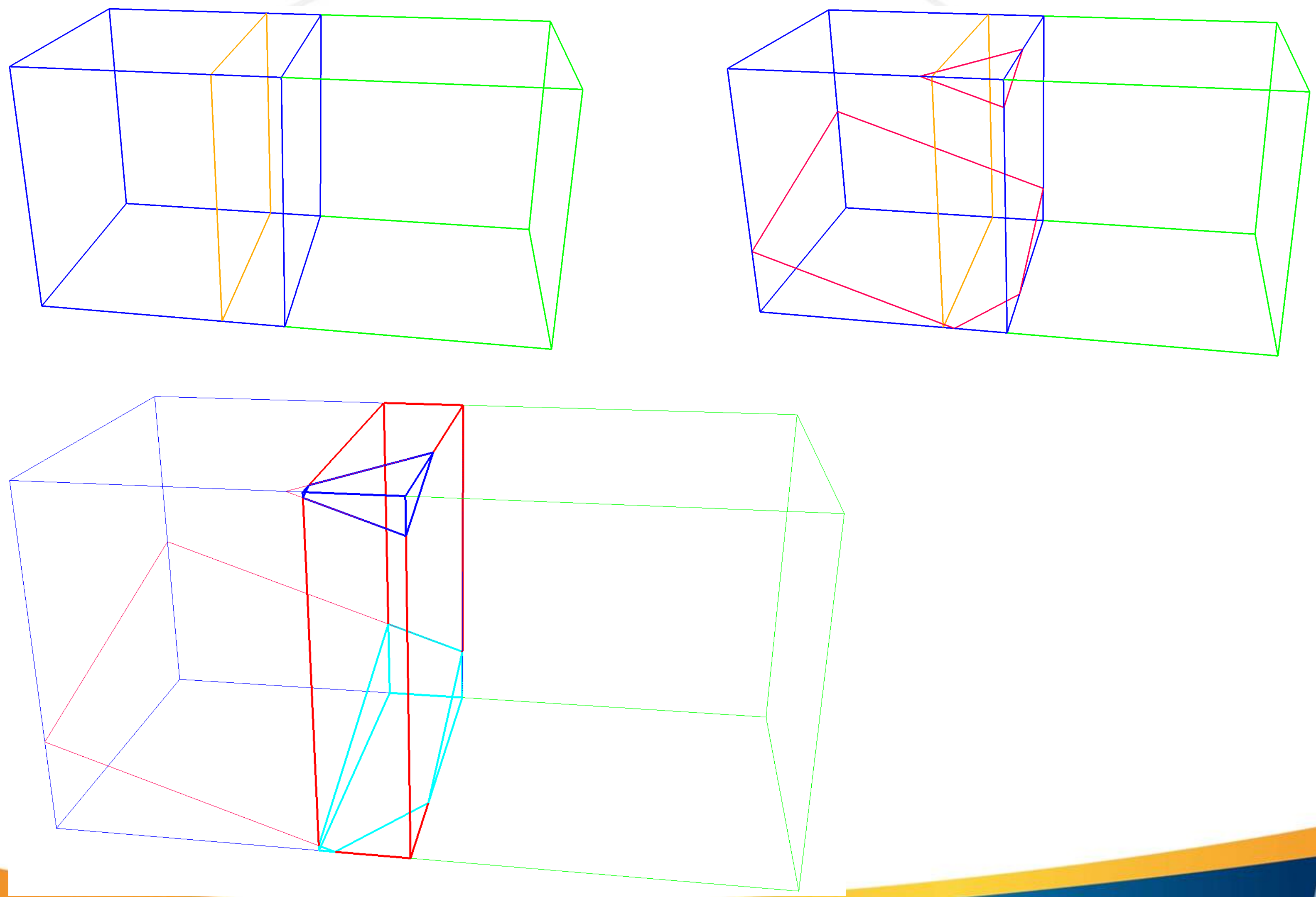
UNCLASSIFIED

Physics on uniform mesh (3D)





Advection on uniform mesh (3D)



Thank You.