It is easier to believe that Yankee professors would lie than that stones would fall from heaven.

Thomas Jefferson

University of Notre Dame

JINA Lecture Series on
Tools and Toys in Nuclear Astrophysics

Nuclear Reaction Network Techniques

Francis X. Timmes
fxt44@mac.com
cococubed.com/talk_pages/jina05.shtml

Los Alamos National Laboratory
Steward Observatory, University of Arizona

Sites of the week

- theory.itp.ucsb.edu/~paxton/
- flash.uchicago.edu/website/home/
- www.ucolick.org/~zingale/

Syllabus

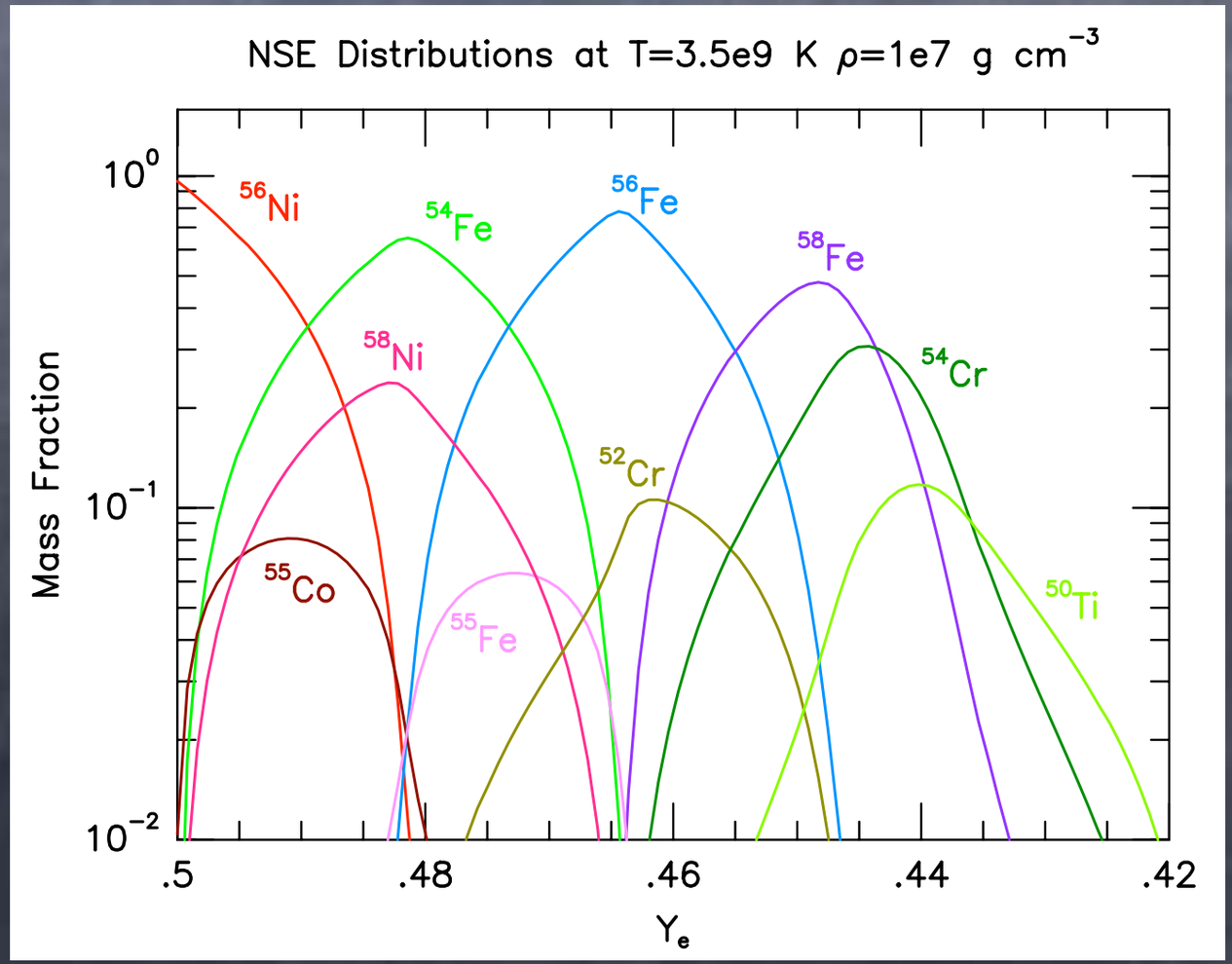
1	June 20	Purpose, Motivation, Forming a network, PP-chain code
2	June 21	Jacobian formation, Energy generation, Time integration, CNO-cycle code
3	June 22	Linear algebra, Thermodynamic trajectories, Alpha-chain code
4	June 23	Nuclear Statistical Equilibrium code, Big-Bang code
5	June 24	Networks in hydrodynamic simulations, General network code

Last Lecture

- The constraints of mass and charge conservation

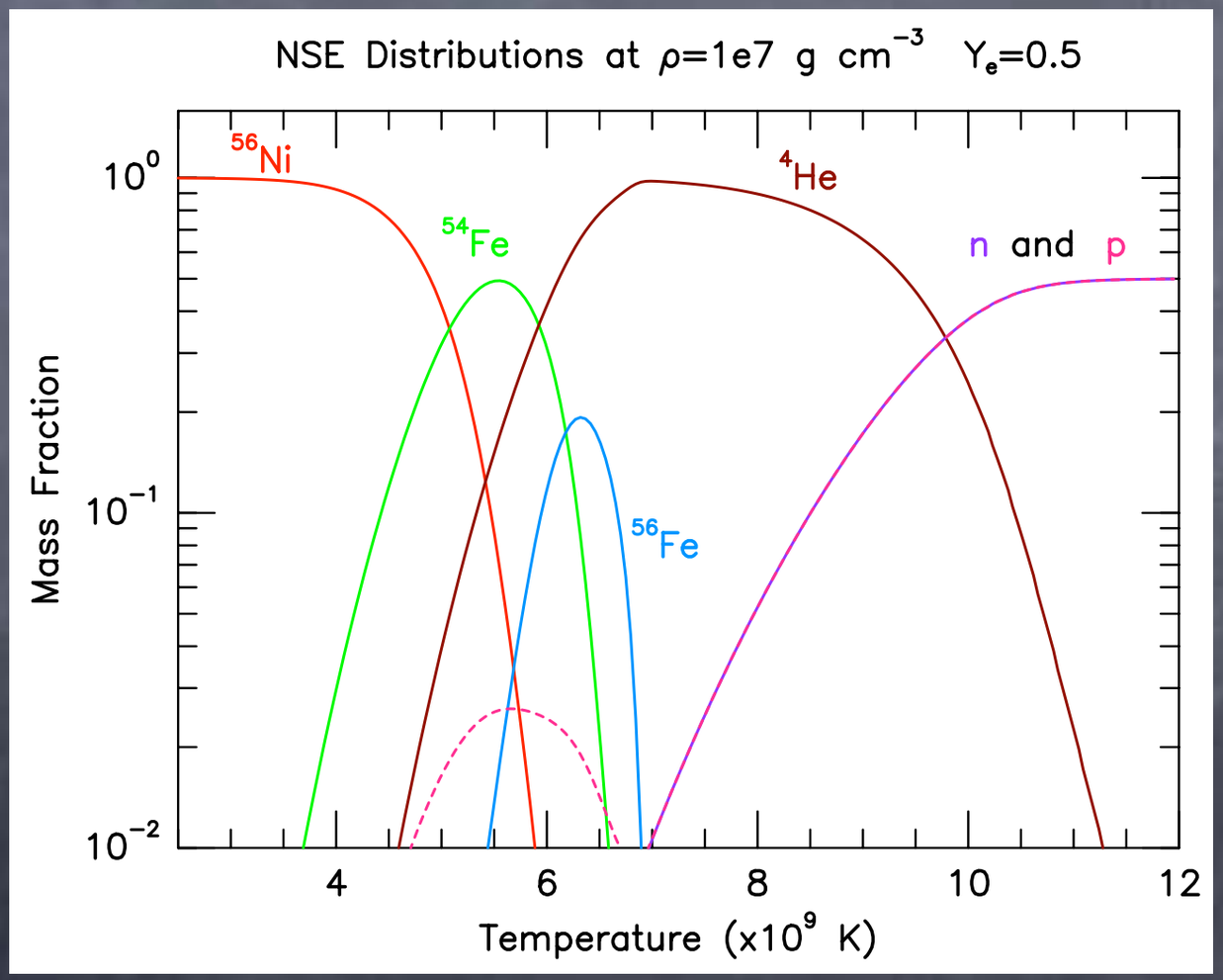
$$\sum_{i=1}^{n} X_i = 1 \qquad \sum_{i=1}^{n} Z_i Y_i = Y_e$$

give two equations for the two unknowns, μ_n and μ_p.



Last Lecture

- If weak interactions are also balanced (e.g., neutrino capture occurring as frequently on the daughter nucleus as electron capture on the parent), then only two parameters, ρ and T, specify the abundances.



- This last occurred for T > 10^9 K in the Big Bang.

Last Lecture

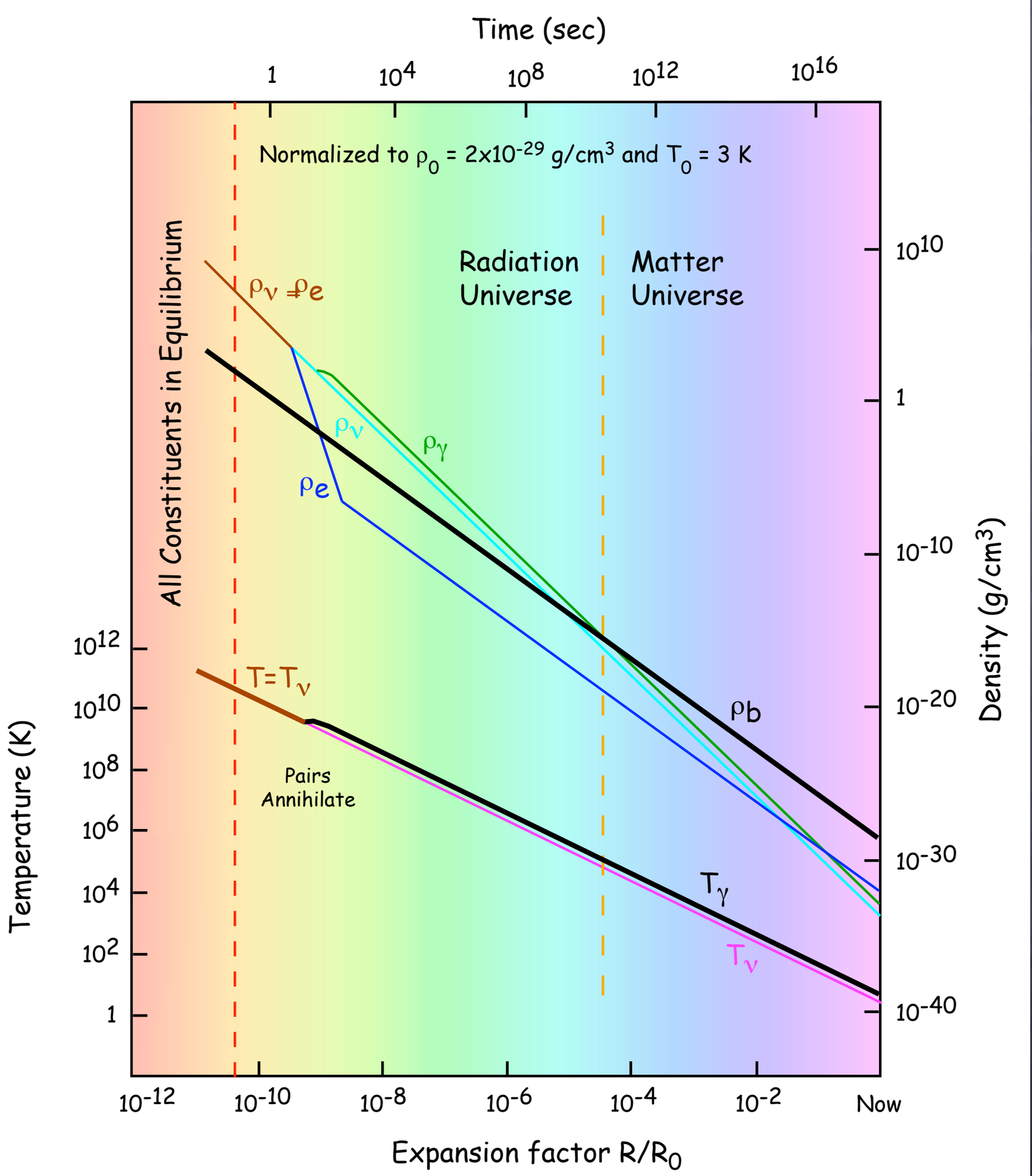
$$\frac{dT}{dt} = \sqrt{\frac{8\pi G a f(x)}{3c^2}}\, T^3 \left[x\frac{dg/dT}{3g} - 1\right]^{-1}$$

$$x = \frac{m_e c^2}{kT}$$

An ordinary differential equation for the photon temperature of the expanding universe.

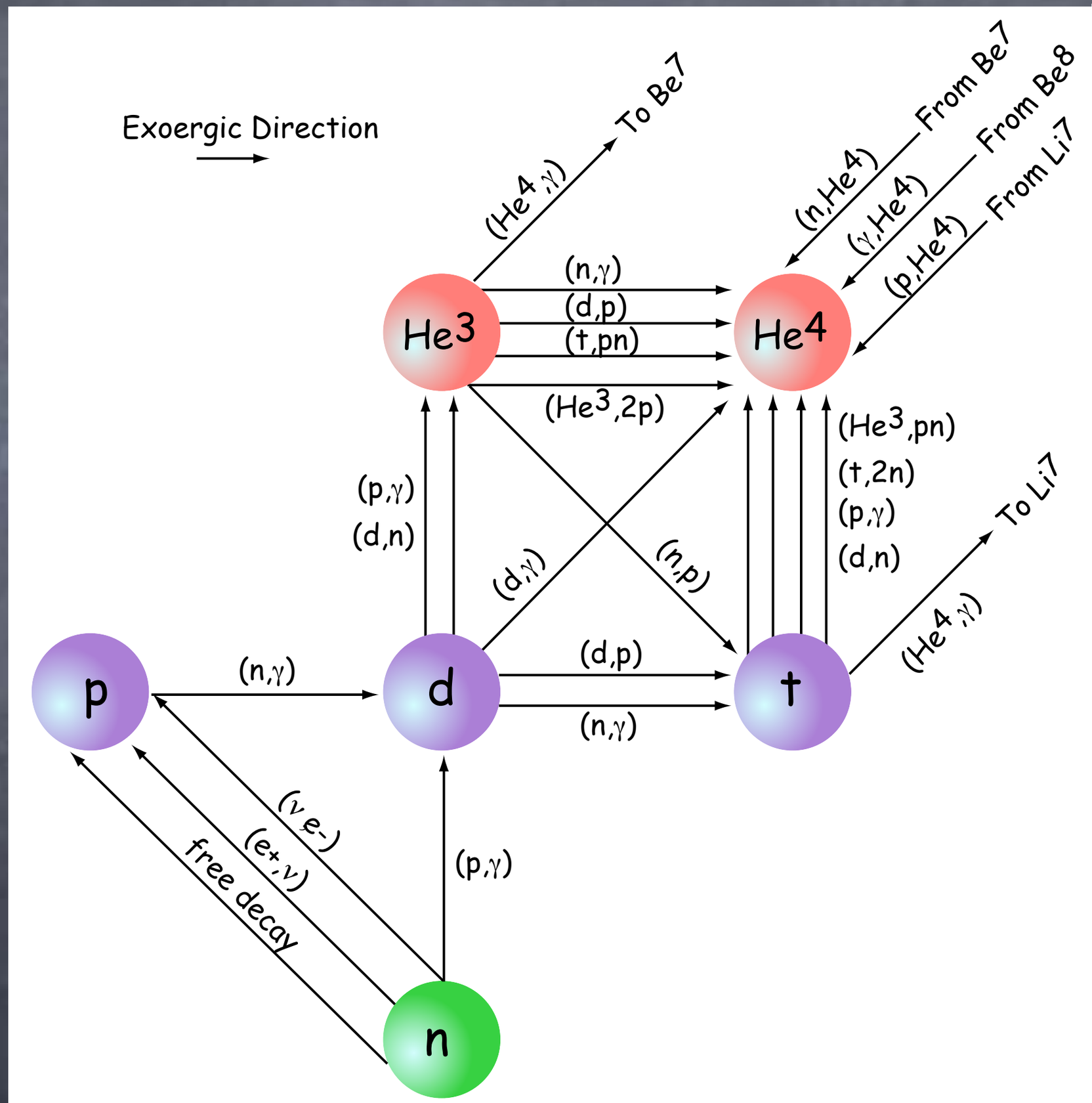
$$g(x) = 1 + N_\nu \frac{7}{8}\left[\frac{4}{11} f(x)\right]^{4/3} + \int_0^\infty \sqrt{x^2+y^2}\left[\exp\left(\sqrt{x^2+y^2}\right) + 1\right]^{-1} y^2 dy$$

- How our universe cools down as it expands.



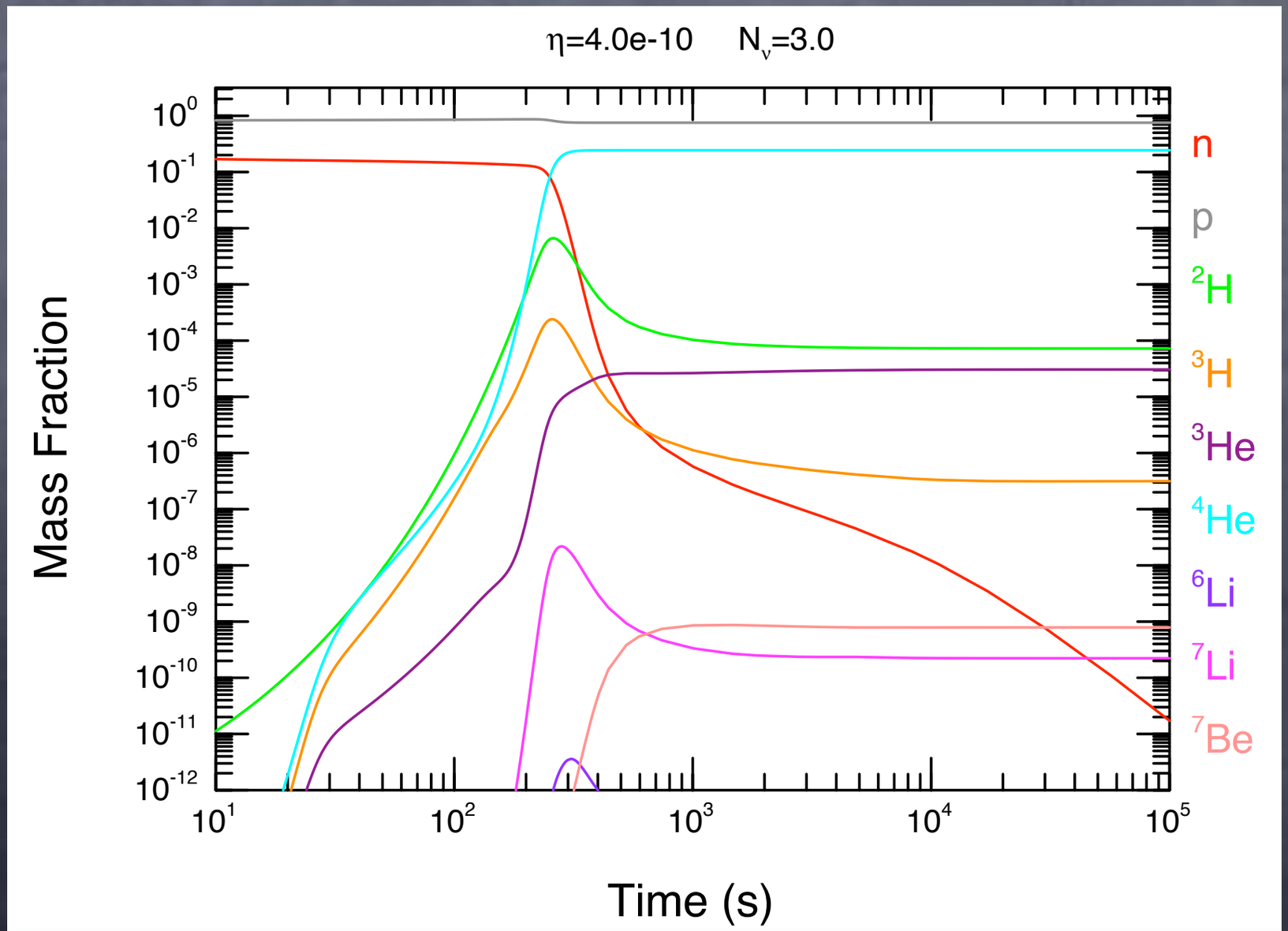
Last Lecture

- A typical Big Bang reaction network.



Last Lecture

- By 35 min nucleosynthesis is essentially complete.

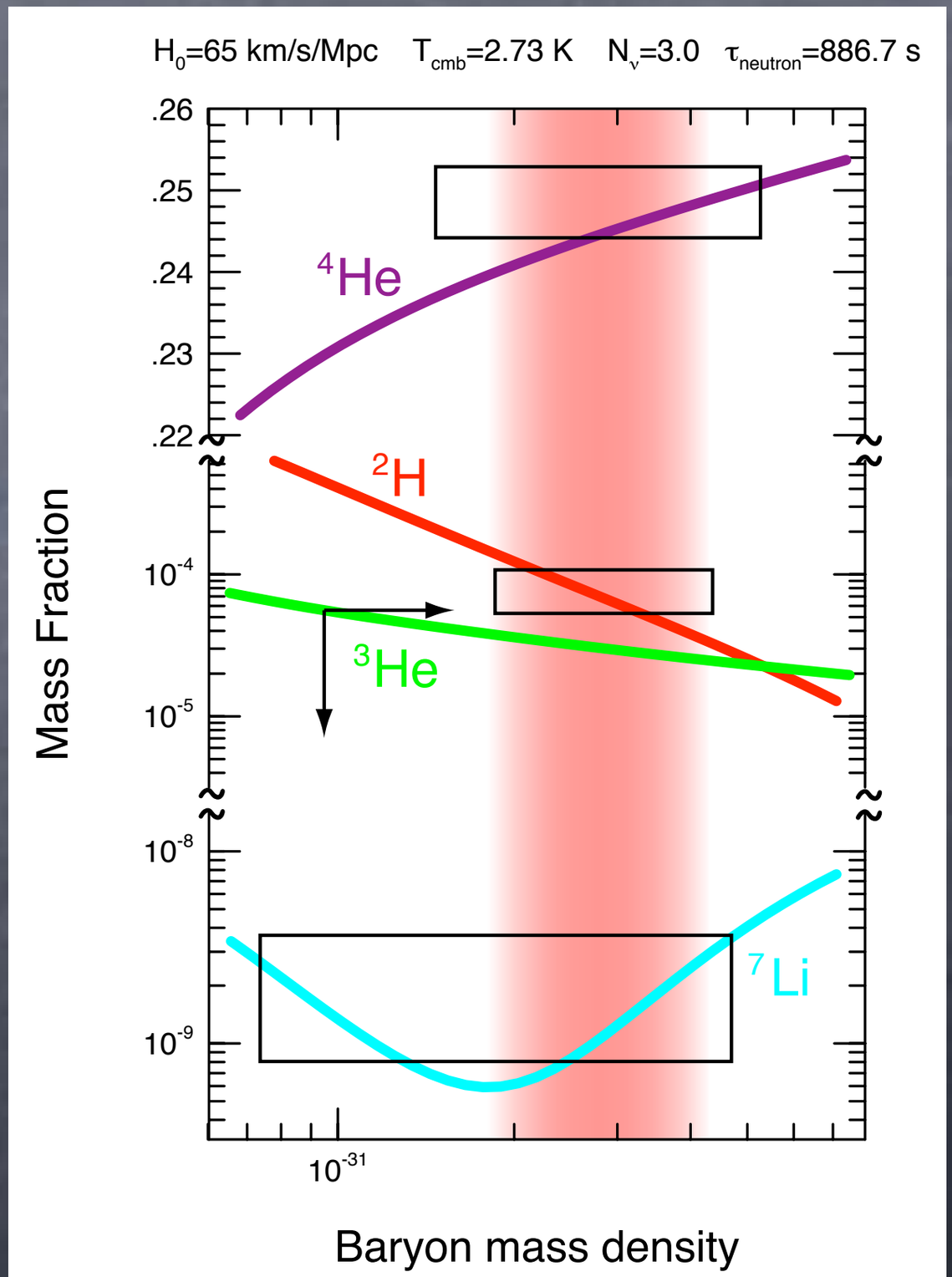


Last Lecture

- The observed abundances of the light elements imply the density of normal matter in the universe is about 3.5×10^{-31} g/cm^3.

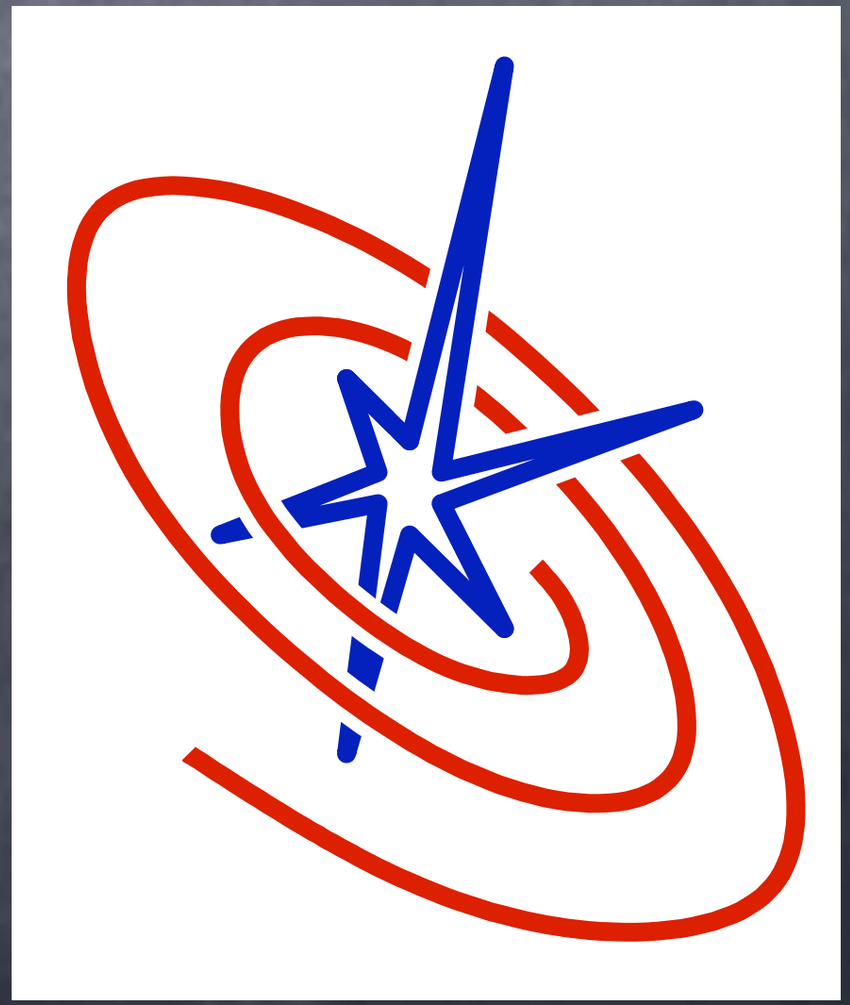
- Four independent measurements of four different elements lead to a consistent constraint.

- This gives us confidence that BBN provides a correct explanation of light element formation.



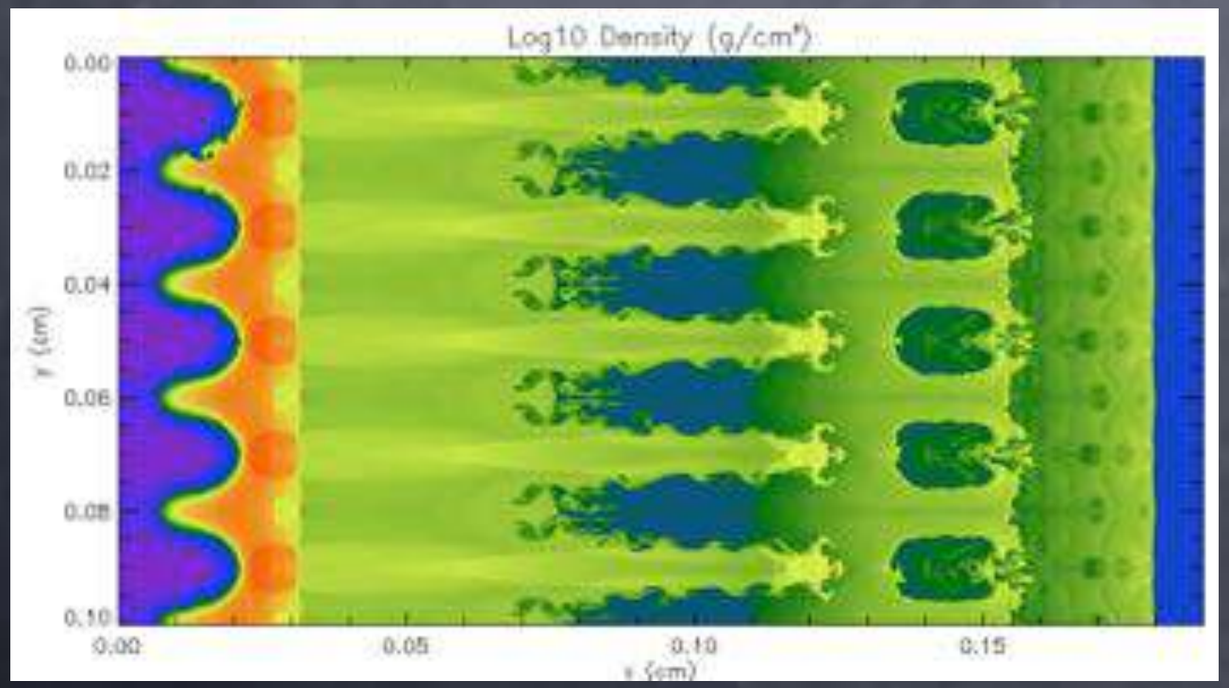
Reactive flows

- The Center for Astrophysical Thermonuclear Flashes, or Flash Center, was founded at the University of Chicago in 1997 under contract to the United States Department of Energy as part of its Accelerated Strategic Computing Initiative (ASCI).

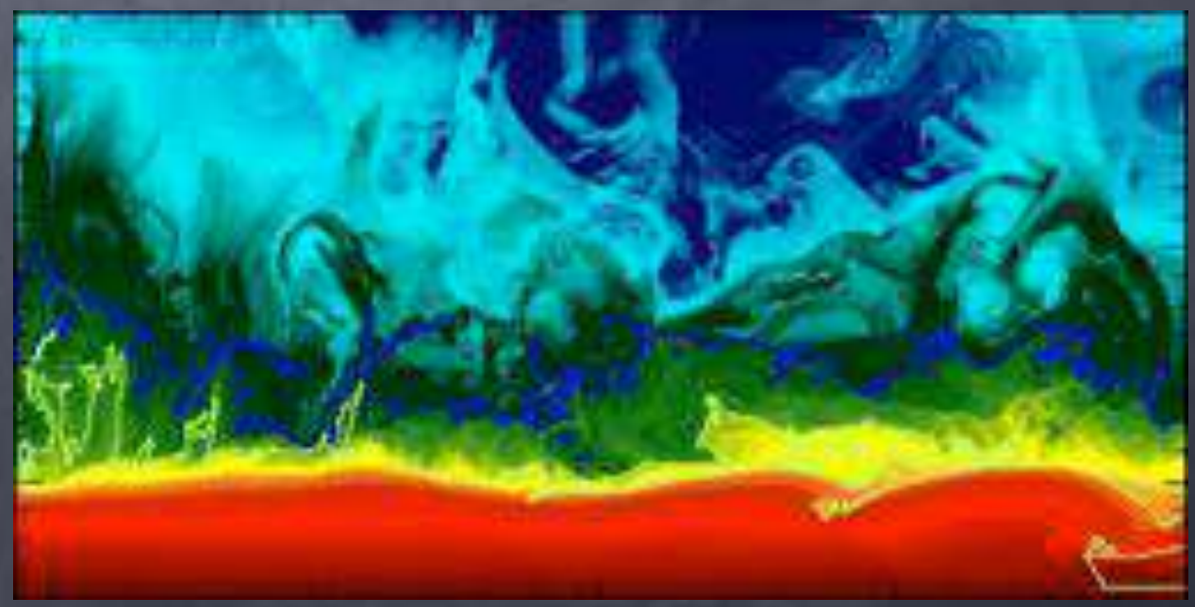


Reactive flows

- The goal of the Center is to significantly advance the solution to several problems related to thermonuclear flashes on the surfaces and in the interiors of compact stars, in particular X-ray bursts, Type Ia supernovae, and classical novae.



Laser impnging on copper-foam target, Calder et al 2002

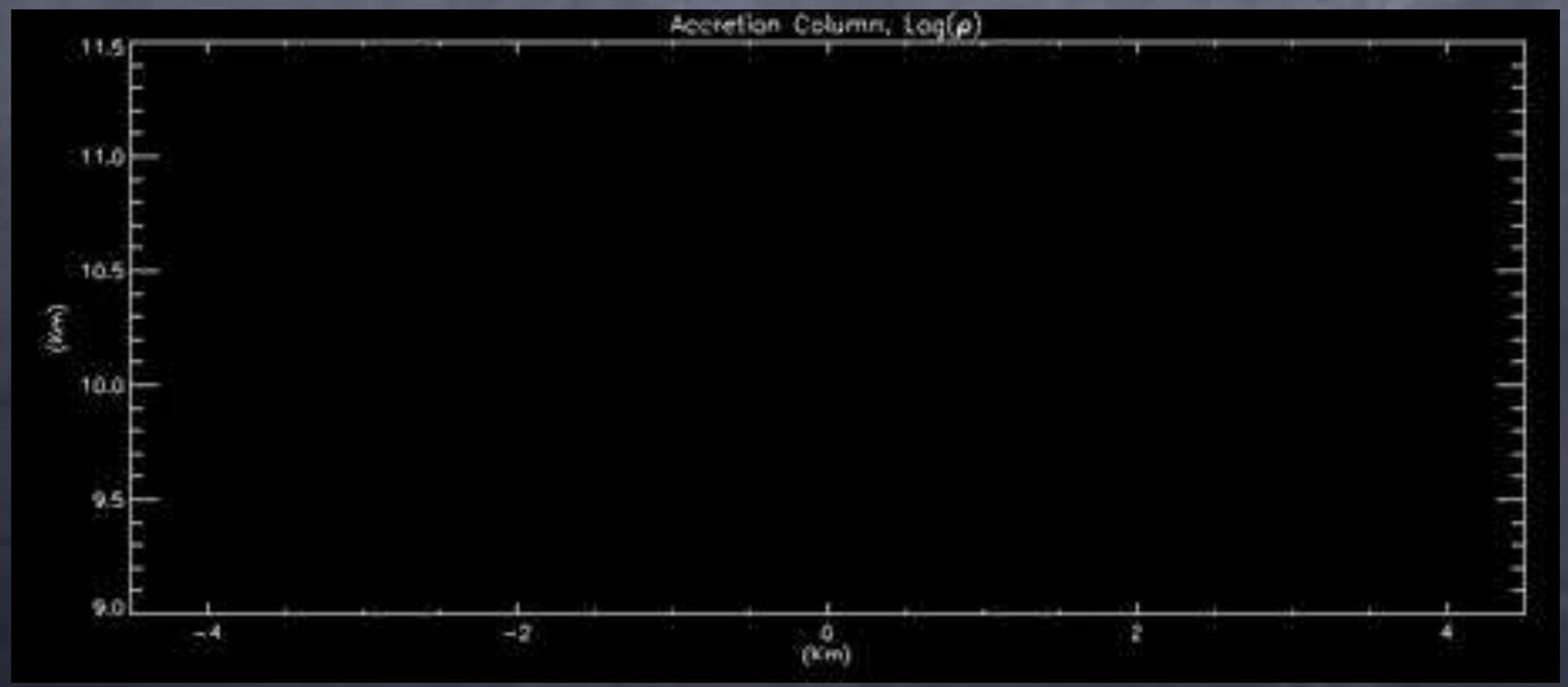


Helium burning on neutron stars, Zingale et al 2001

Reactive flows

- These problems are remarkable for the breadth of physical phenomena involved.

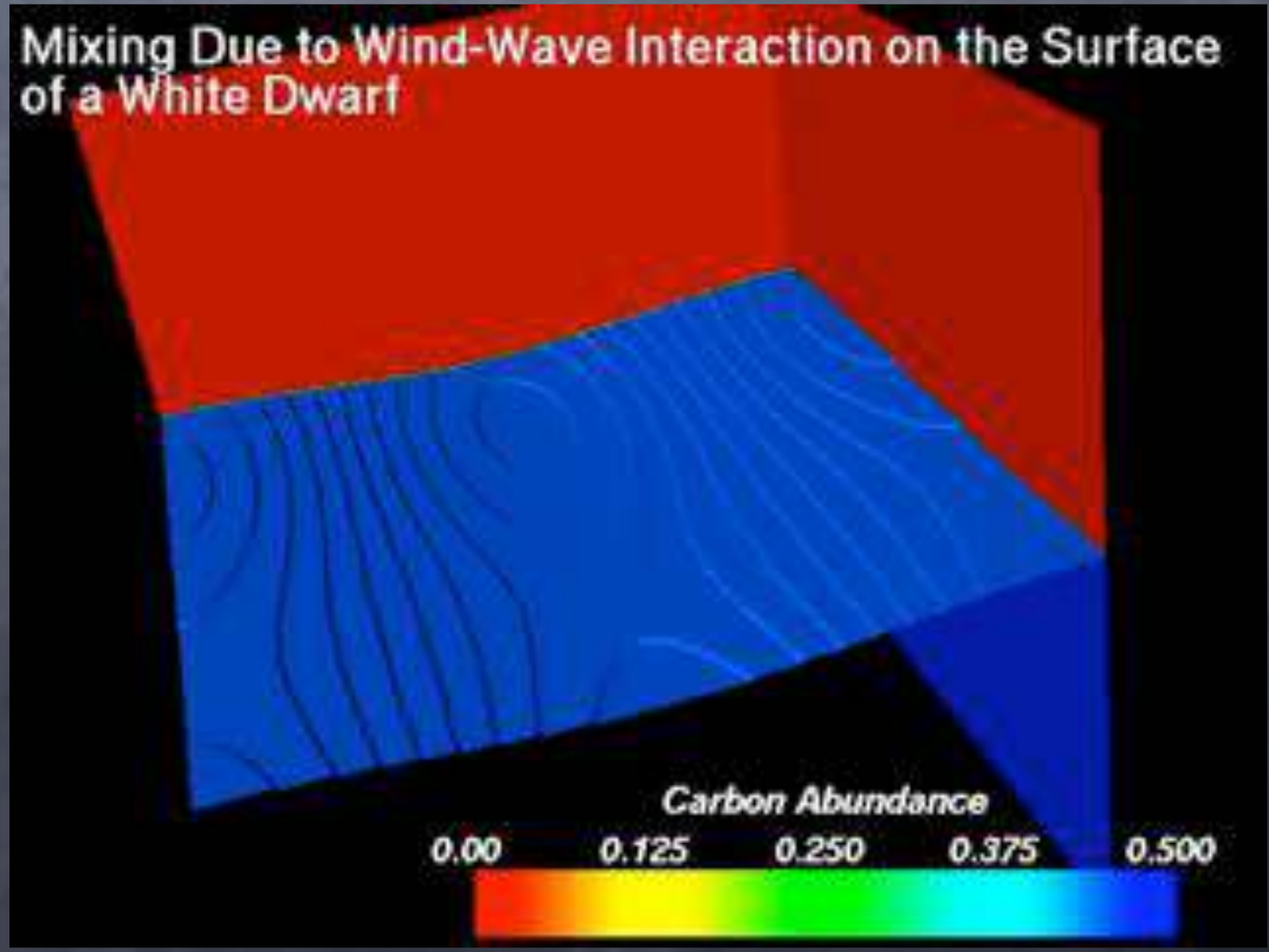
- They range from accretion flow onto the surfaces of the compact stars to shear flow and Rayleigh-Taylor instabilities on the stellar surfaces, ignition of nuclear burning under conditions leading to convection, deflagration and detonation flame fronts, and stellar envelope expansion.



Accretion onto neutron star, Mignone et al. 2003

Reactive flows

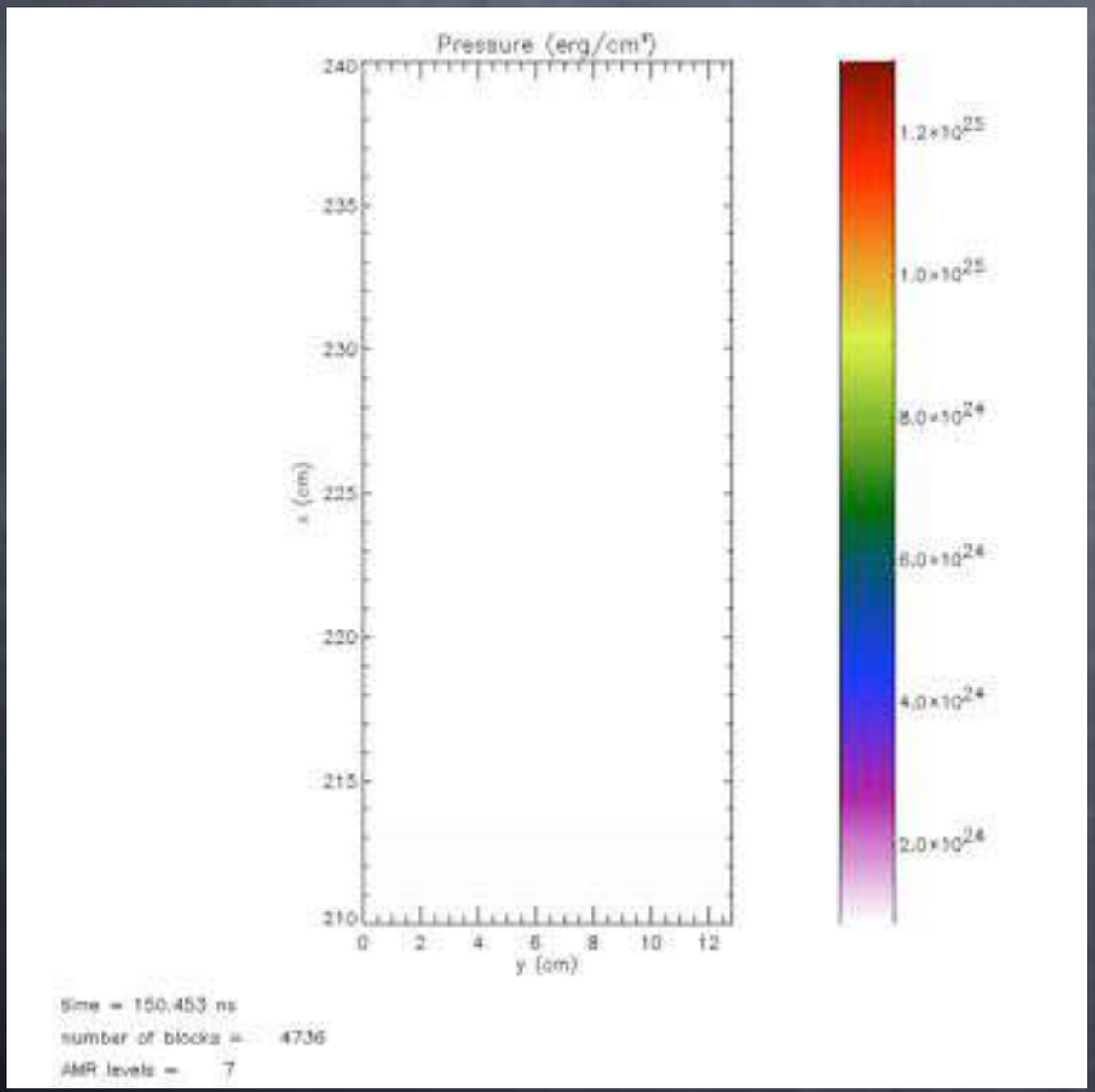
- The physical processes include convection and turbulence at large Reynolds and Rayleigh numbers, convective penetration of stable matter at very high densities, equations of state for relativistic and degenerate matter, thermonuclear burning, mixing instabilities, burning front propagation, and radiation hydrodynamics.



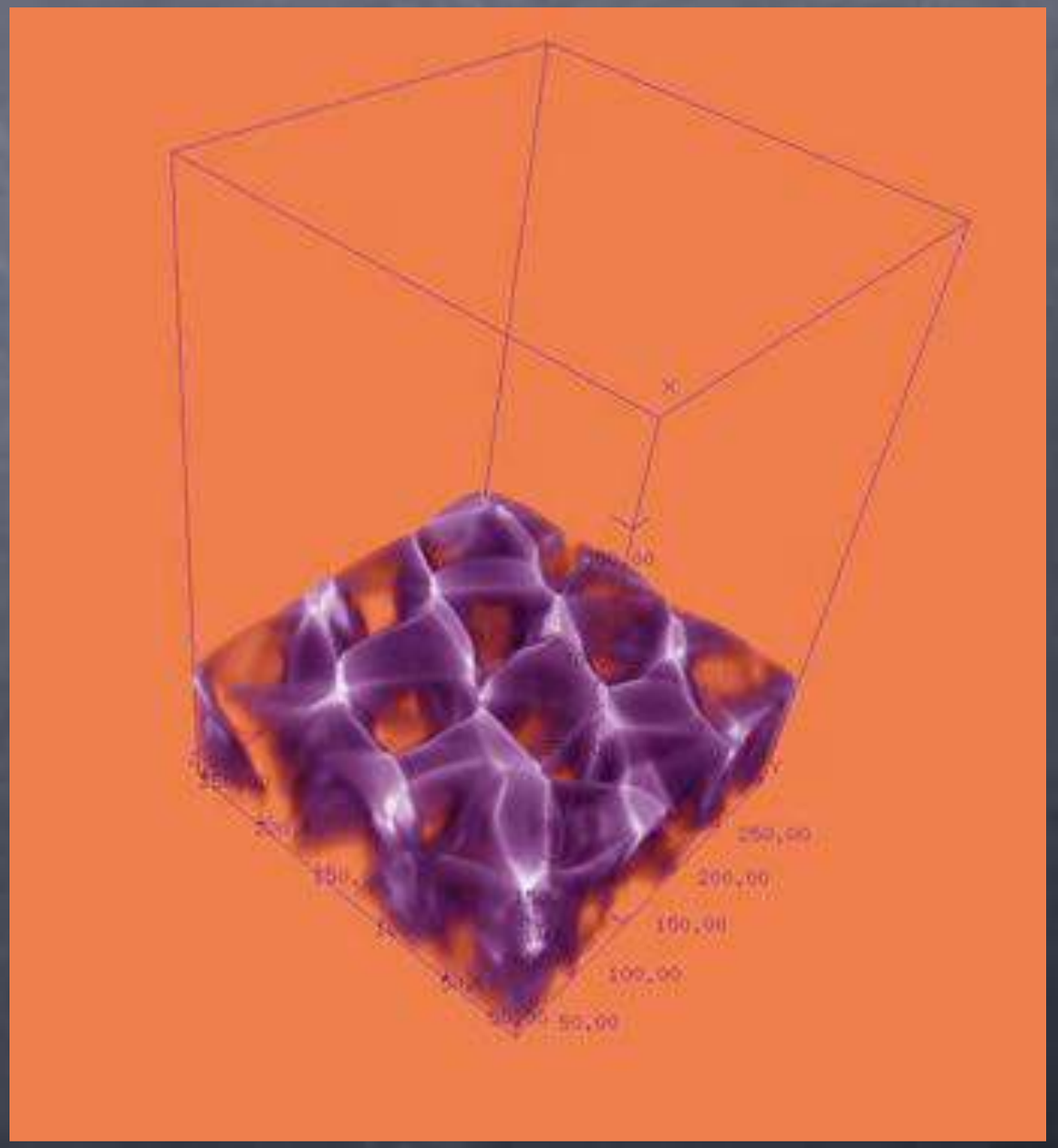
Mixing at white dwarf surface, Alexakis et al,2004

Reactive flows

- Few astrophysical problems present a substantially greater level of physical complexity.

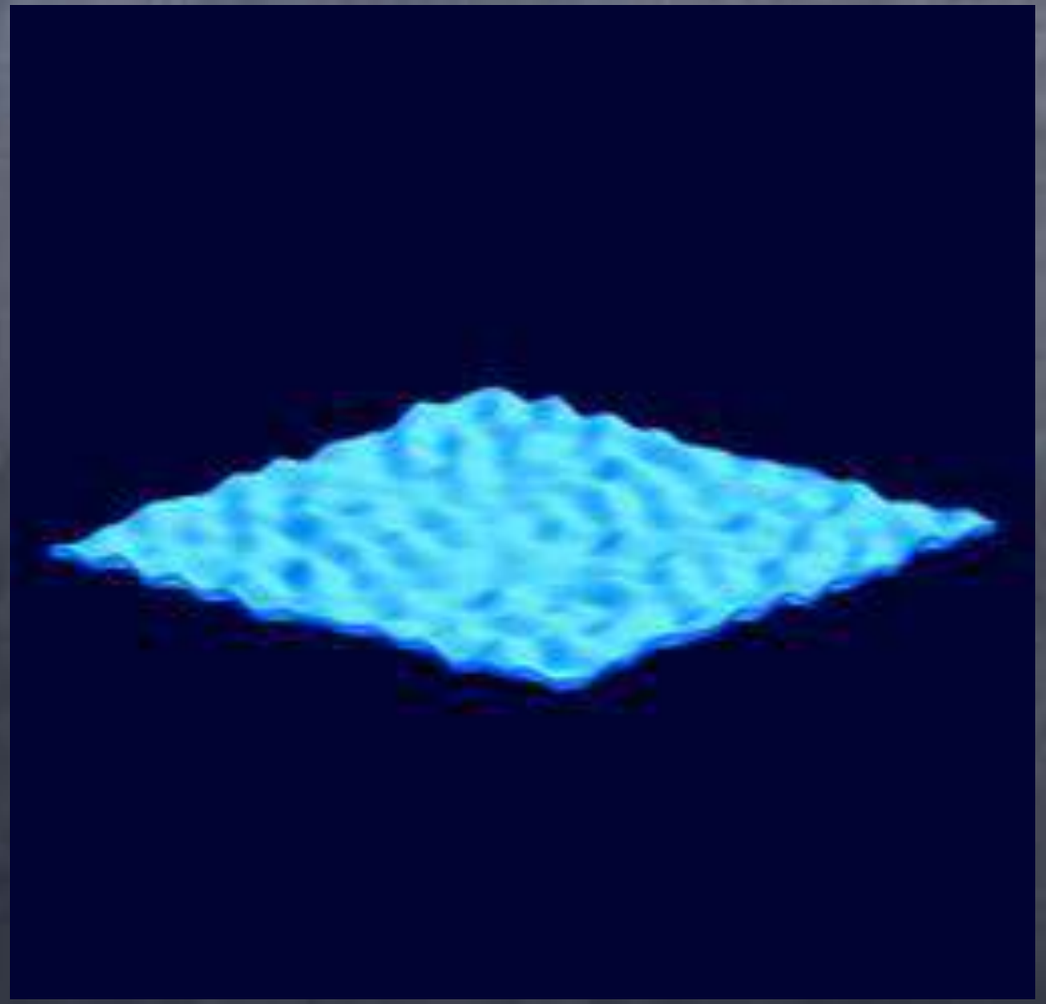


Cellular detonation
Timmes et al 2001



Reactive flows

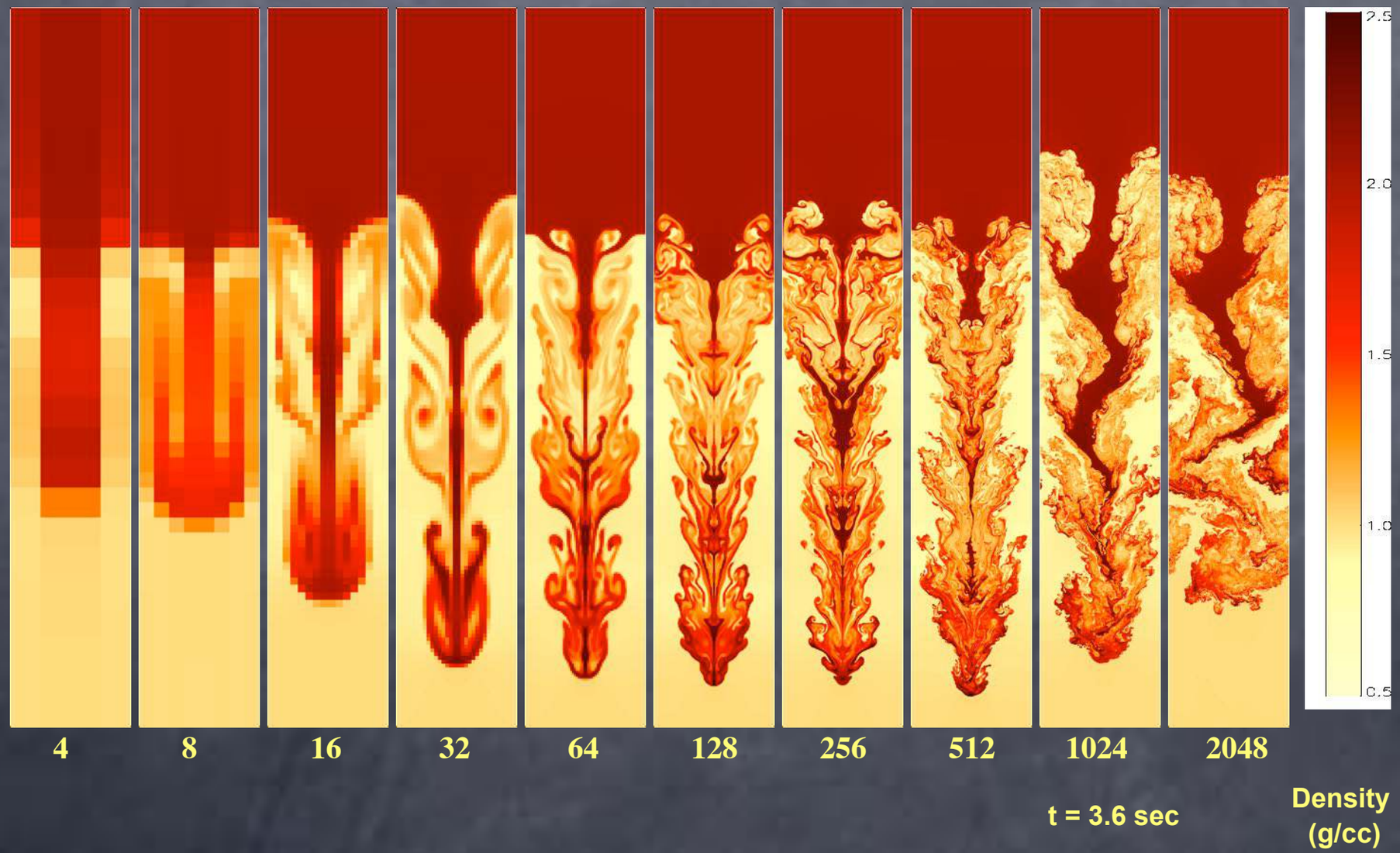
- To advance the solution of these astrophysical problems requires the development of new simulation tools capable of handling the extreme resolution and physical requirements imposed by these thermonuclear flashes and to do so while making efficient use of the parallel supercomputers developed by the ASCI project, the most powerful constructed to date.



Buring RT instability,
Zingale et al. 2005

Reactive flows

- The FLASH code represents a step along the road to this goal.

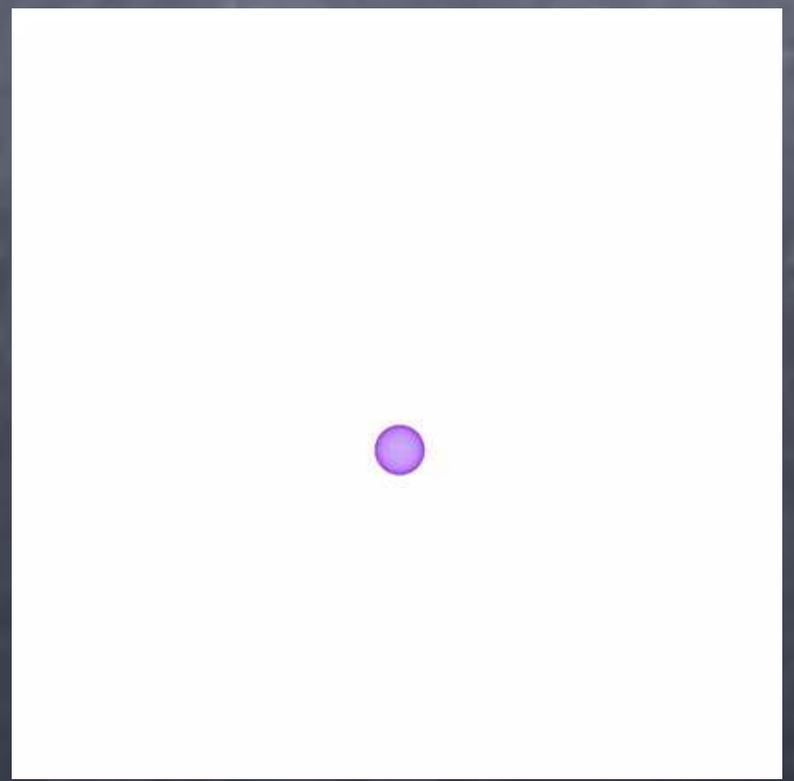


Single mode RT instability,
Calder et al. 2003

Reactive flows

- FLASH is a modular, adaptive, and parallel simulation code capable of handling general compressible flow problems in astrophysical environments.

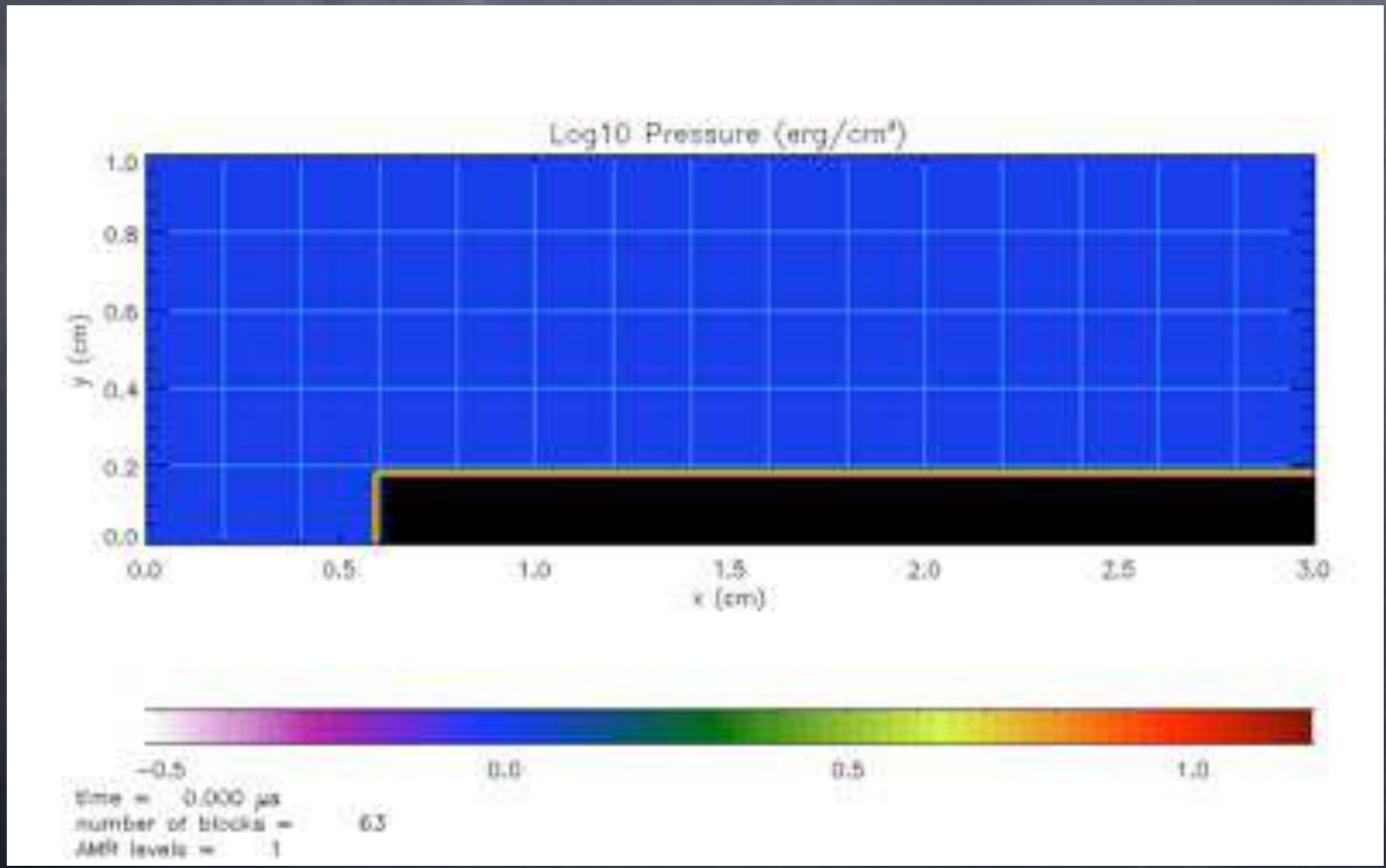
- It is modular: FLASH has been designed to allow users to configure initial and boundary conditions, change algorithms, and add new physical effects with minimal effort.



Off Center SNIa deflagration,
Plewa et al. 2005

Reactive flows

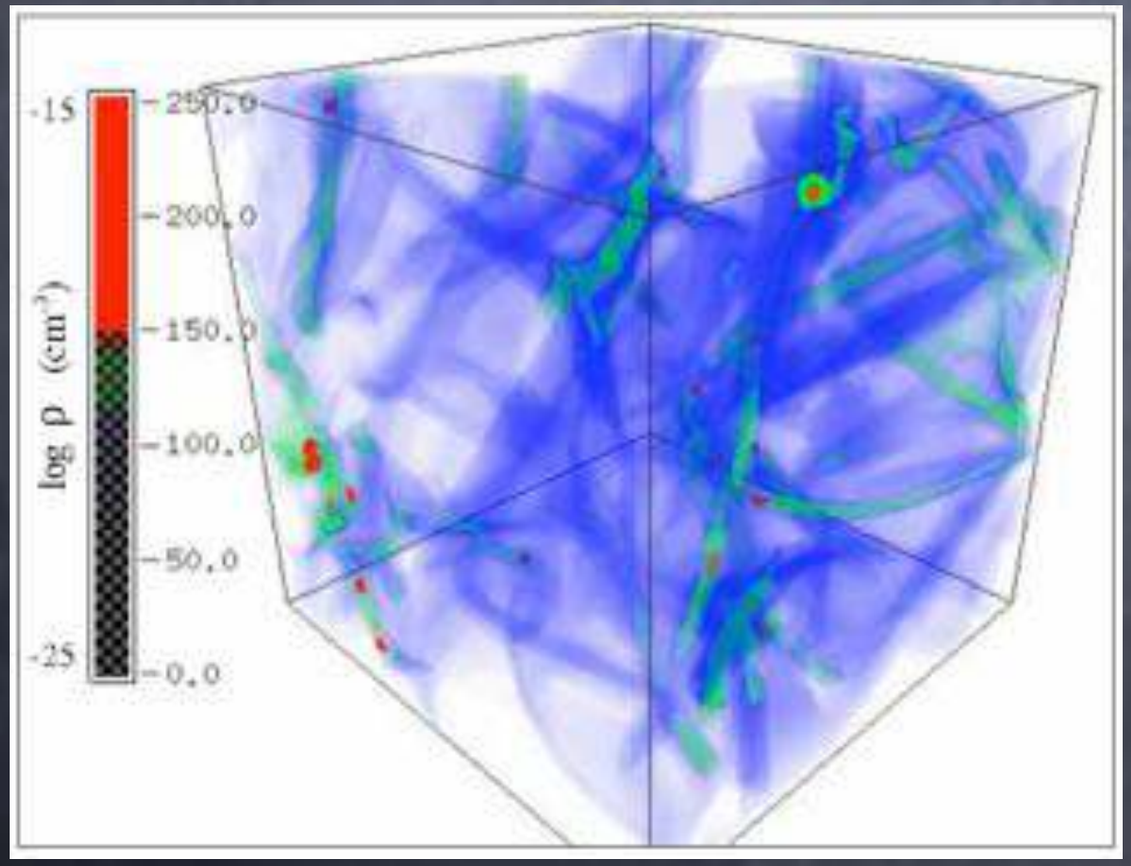
- It is adaptive: FLASH uses a block-structured adaptive grid, placing resolution elements only where they are needed most.



Emery wind tunnel
Ricker et al, 2000

Reactive flows

- It is parallel: FLASH uses the Message-Passing Interface library (mpich) to achieve portability and scalability on a variety of different message-passing parallel computers.



Jean's mass collapse
Ricker et al, 2002, 2004

Reactive flows

FLASH 1.7
Architecture,
Optimization, 2002

FLASH 2.5
Data Layout,
Modularity, 2005

FLASH 0.0
Initial, 1999

FLASH 1.0
Organization, 2000

Reactive flows

- FLASH a hydrodynamics code for 1, 2 or 3D reactive flows.
- Implementation in Fortran 90 + C + Pearl + IDL
  - ~28,000 lines in driver + adaptive mesh refinement modules
  - ~79,000 lines in physics modules
- Object-oriented framework
- Portable - runs efficiently on a wide class of machines

Reactive flows

- Parallel using Argonne's Message Passing Interface Library (mpich) library and uses the Hierarchical Data Format (HDF 5) or the Network Common Data Form (NETCDF) for parallel I/O

- High-performance - Gordon Bell prize @ Supercomputing 2000

- Ongoing verification and validation with analytic test problems Laboratory experiments

- Available for distribution to the astrophysics community flash.uchicago.edu/website/home/

AMR in FLASH

- Reduces time to solution and improves accuracy by concentrating grid points in regions which require higher resolution

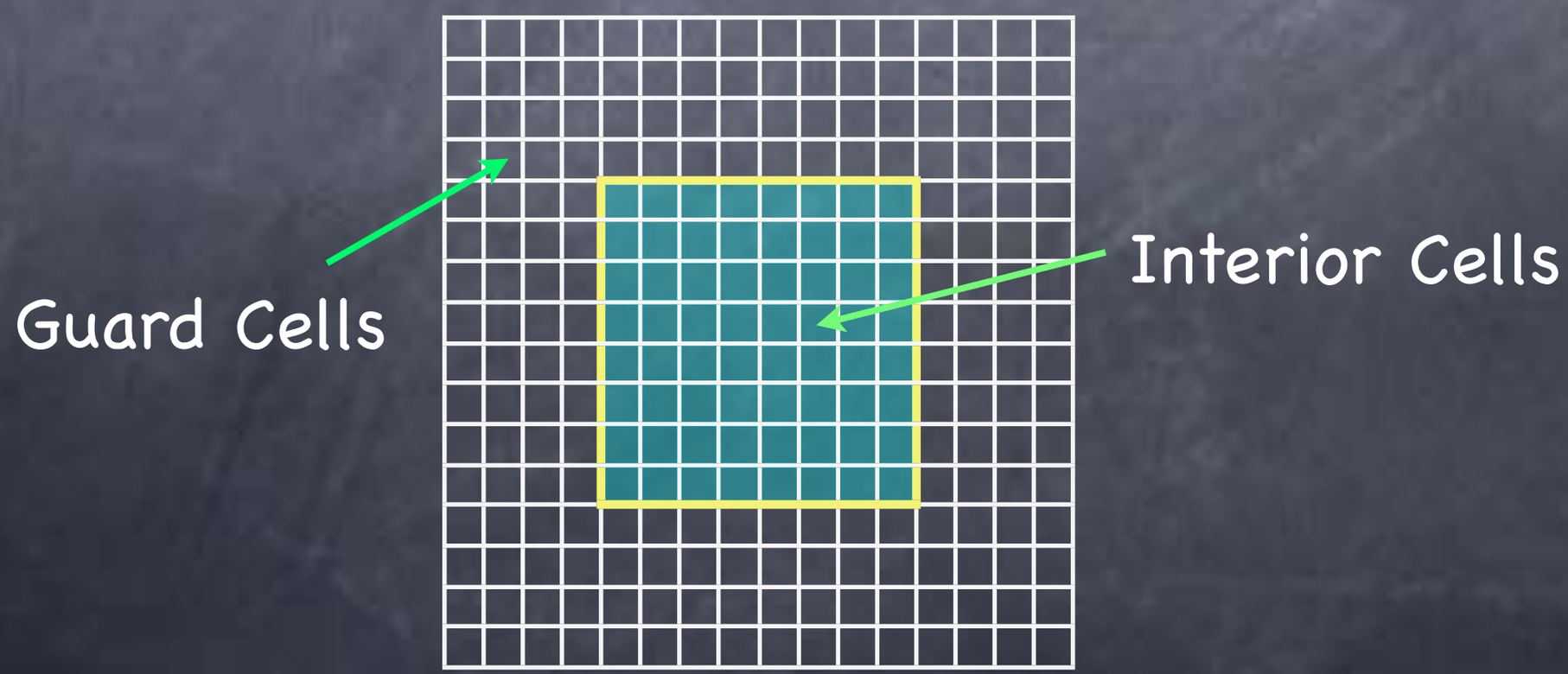
- PARAMESH (NASA / GSFC)
  ct.gsfc.nasa.gov/paramesh/Users_manual/amr.html

- Block structured refinement (8 x 8 x 8 blocks)

- User-defined refinement criterion - currently using second derivatives of density and pressure

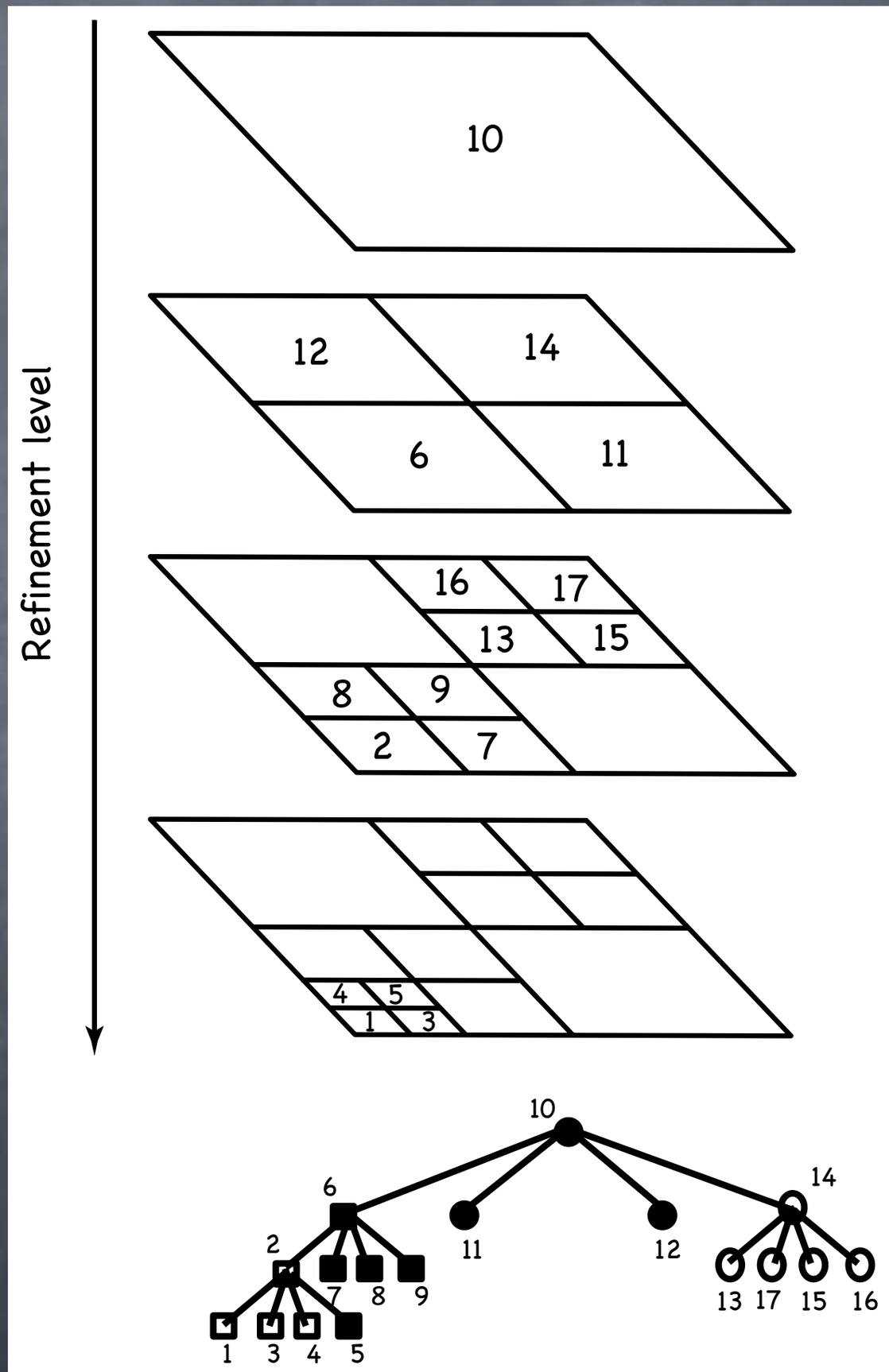
AMR in FLASH

- Local physics (hydro, eos, burn, etc.) occurs on a block. When guardcells are filled, operators act on each block as if isolated.
- Number of guardcells depends on stencil size.
- Number of interior points must balance tradeoff:
  More cells - more efficient (until block too big for cache)
  Fewer cells - can refine more quickly in smaller area.



AMR in FLASH

- Blocks and refinement are arranged in an 2^d-tree structure.
- When a block is refined, 2^d child blocks are created, each with a factor of 2 resolution increase over its parent.
- Blocks assigned indices via space-filling curve.

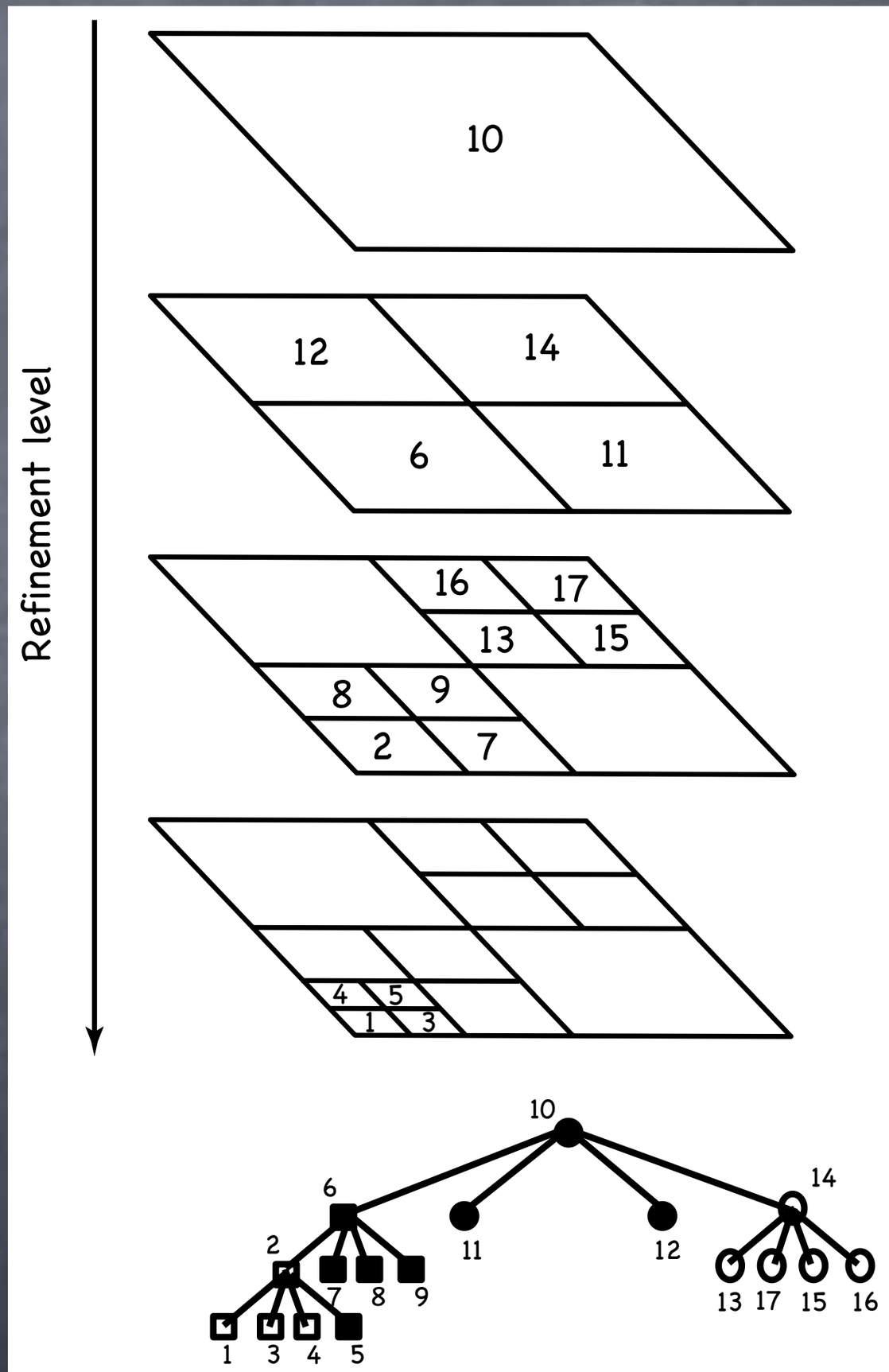


AMR in FLASH

- Neighboring blocks differ by at most one level of refinement.

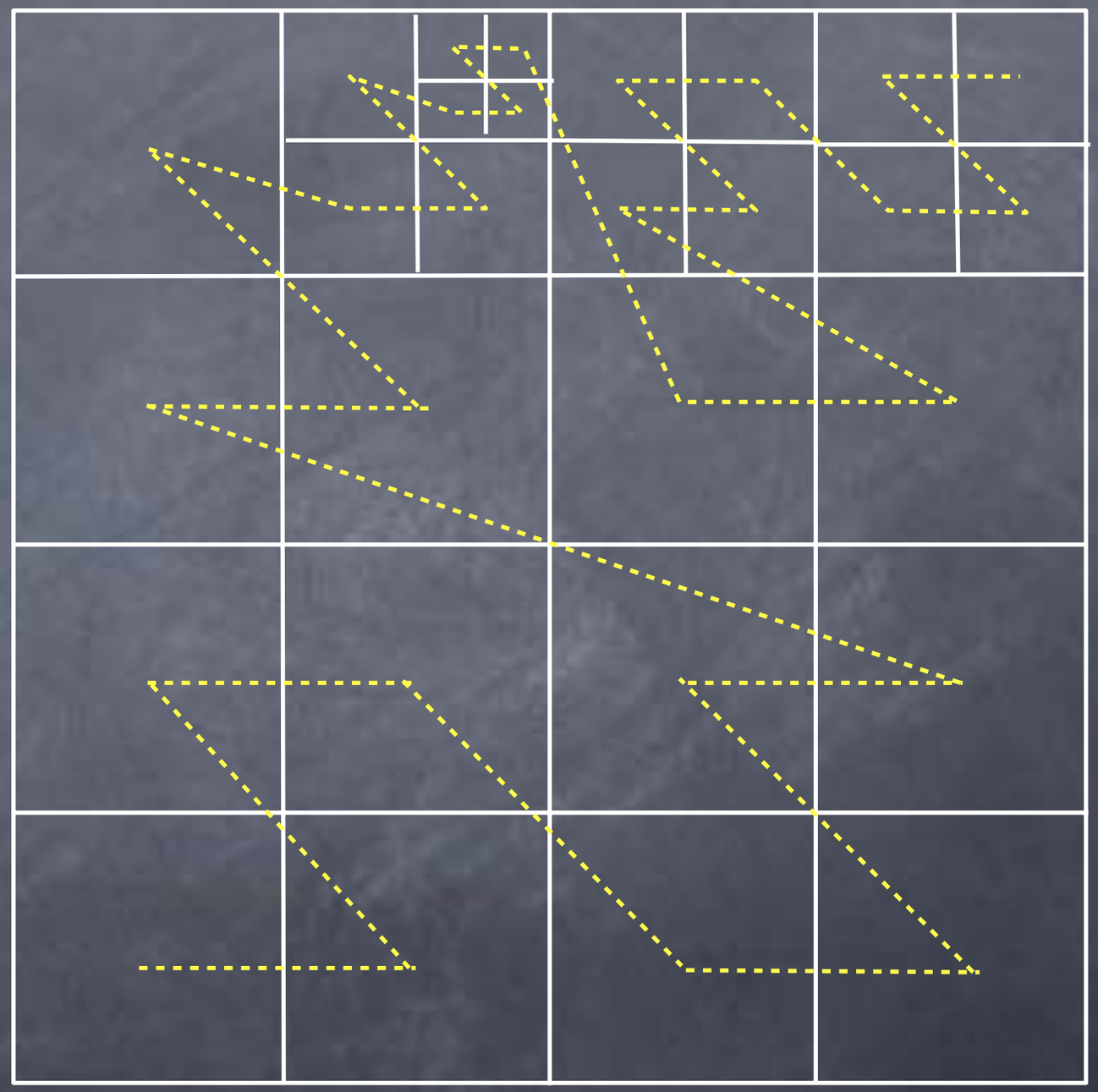
  Drawback: resolution can only fall of linearly in distance.

  Feature: simplifies, speeds up accurate calculation of "boundary conditions" (guardcells)



AMR in FLASH

- Blocks are distributed across processors using a space-filling curve method.

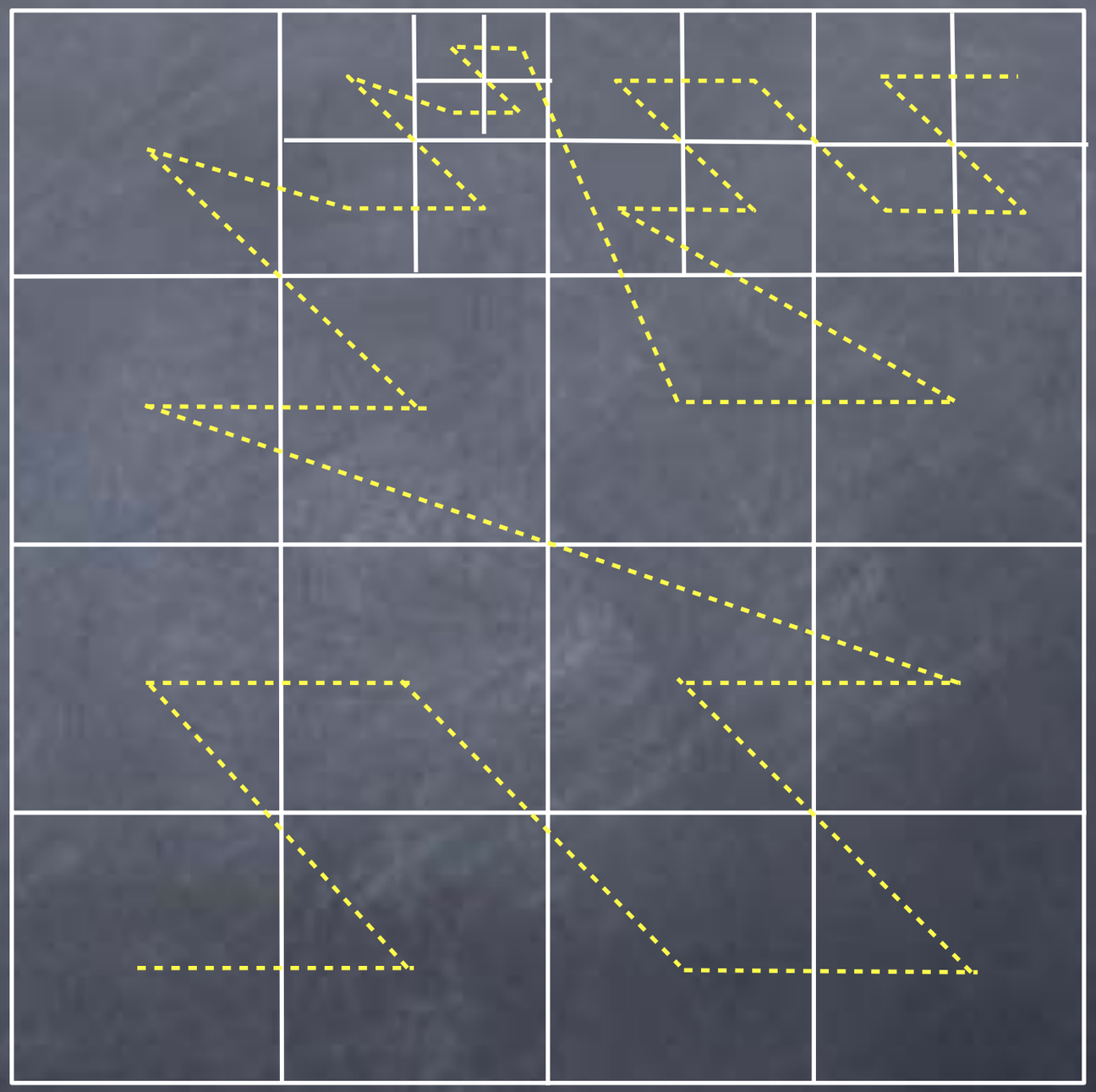


- A curve (typically a Morton curve) is threaded through the mesh, with weights according to how much each block "costs" in CPU use, communication, memory.

AMR in FLASH

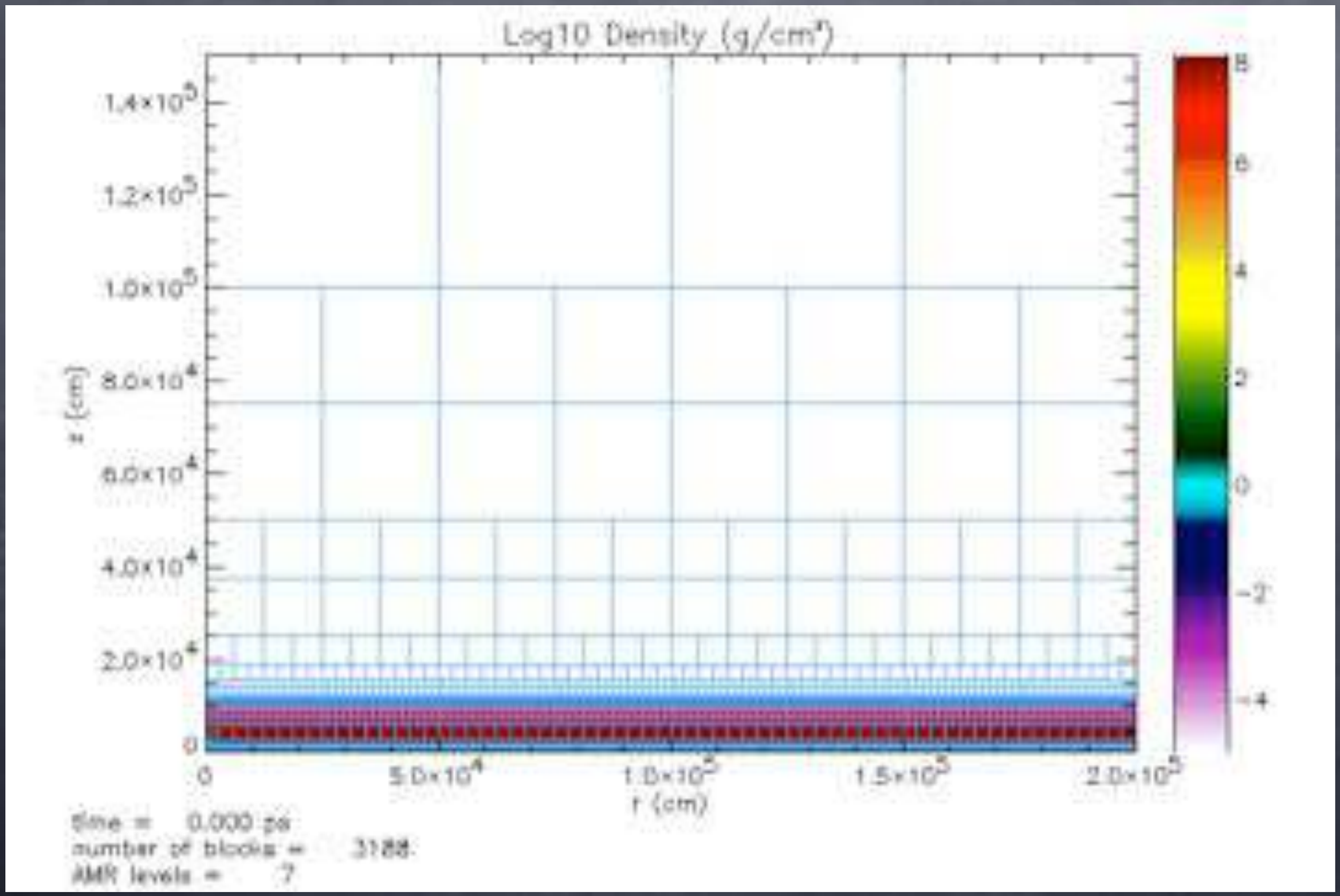
- Other curves (Hilbert, Peano) with better clustering properties have been tried; no performance enhancement.

- Refinement and redistribution of blocks every four time steps.



AMR in FLASH

zingale et al, 2000



Reactive flows

- The most common choice of the hydrodynamic module in the current version of the FLASH code solves Euler's equations for compressible gas dynamics. The equations can be written in conservative form as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \rho \tilde{\mathbf{v}} = 0$$

$$\frac{\partial \rho \tilde{\mathbf{v}}}{\partial t} + \nabla \cdot \rho \tilde{\mathbf{v}} \tilde{\mathbf{v}} + \nabla P = \rho \tilde{\mathbf{g}}$$

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot (\rho E + P) \, \tilde{\mathbf{v}} = \rho \tilde{\mathbf{v}} \cdot \tilde{\mathbf{g}}$$

$$E = \epsilon + \frac{1}{2} v^2$$

Reactive flows

- The pressure is obtained from the internal energy and the density by an equation of state.

- For the case of a simple gamma-law equation of state, the pressure is given by

$$P = (\gamma - 1)\rho\epsilon$$

but more general equations of state are readily available.

Eos
Goddess of Dawn
Aurora to Romans

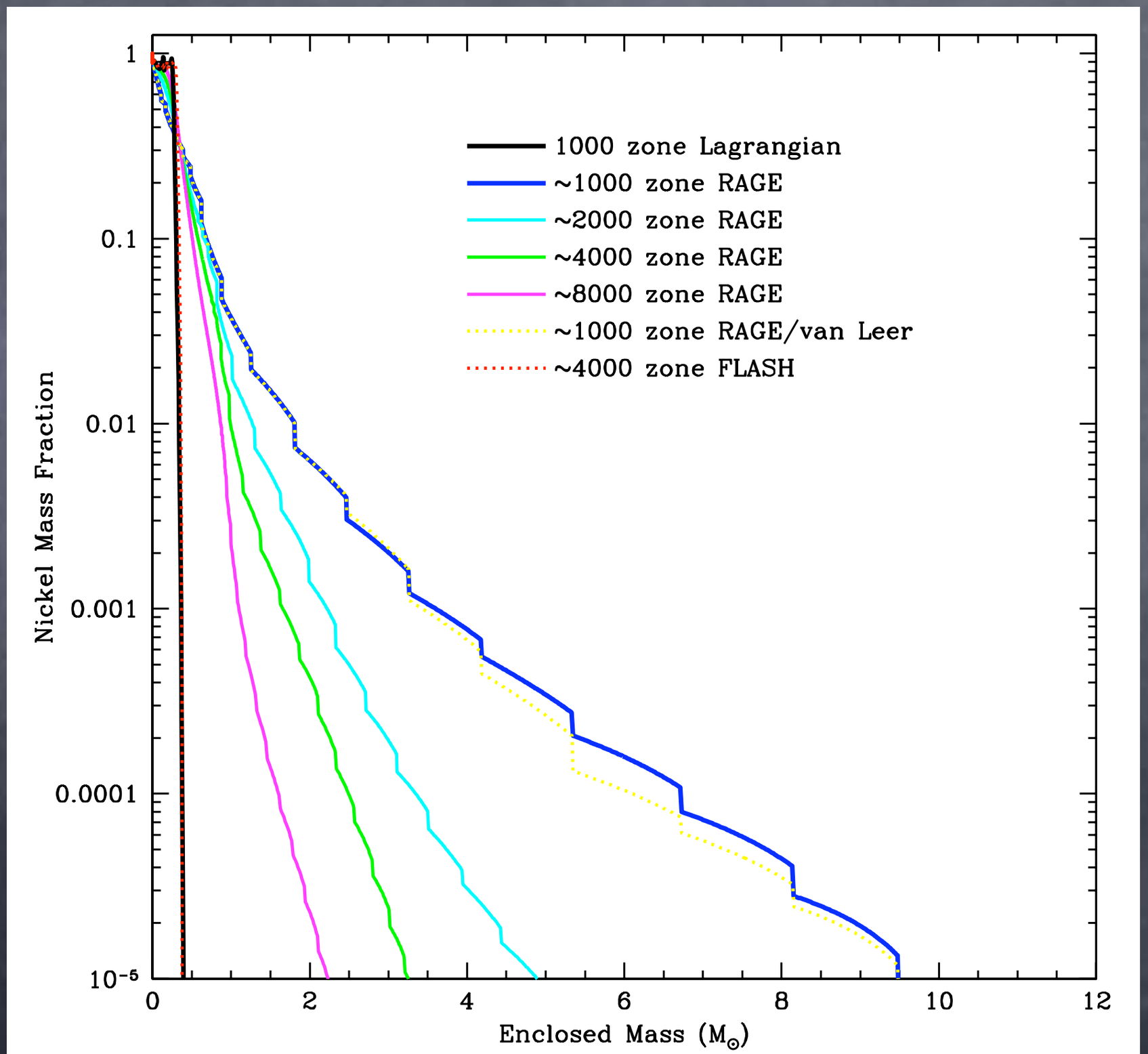
Reactive flows

- For reactive flows, a separate advection equation must be solved for each chemical or nuclear species

$$\frac{\partial \rho X_i}{\partial t} + \nabla \cdot \rho X_i \tilde{\mathbf{v}} = 0$$

- The code does not explicitly track interfaces between the fluids, so that a small amount of numerical mixing can be expected during the course of the calculation.

Reactive flows



Fryer et al, 2005

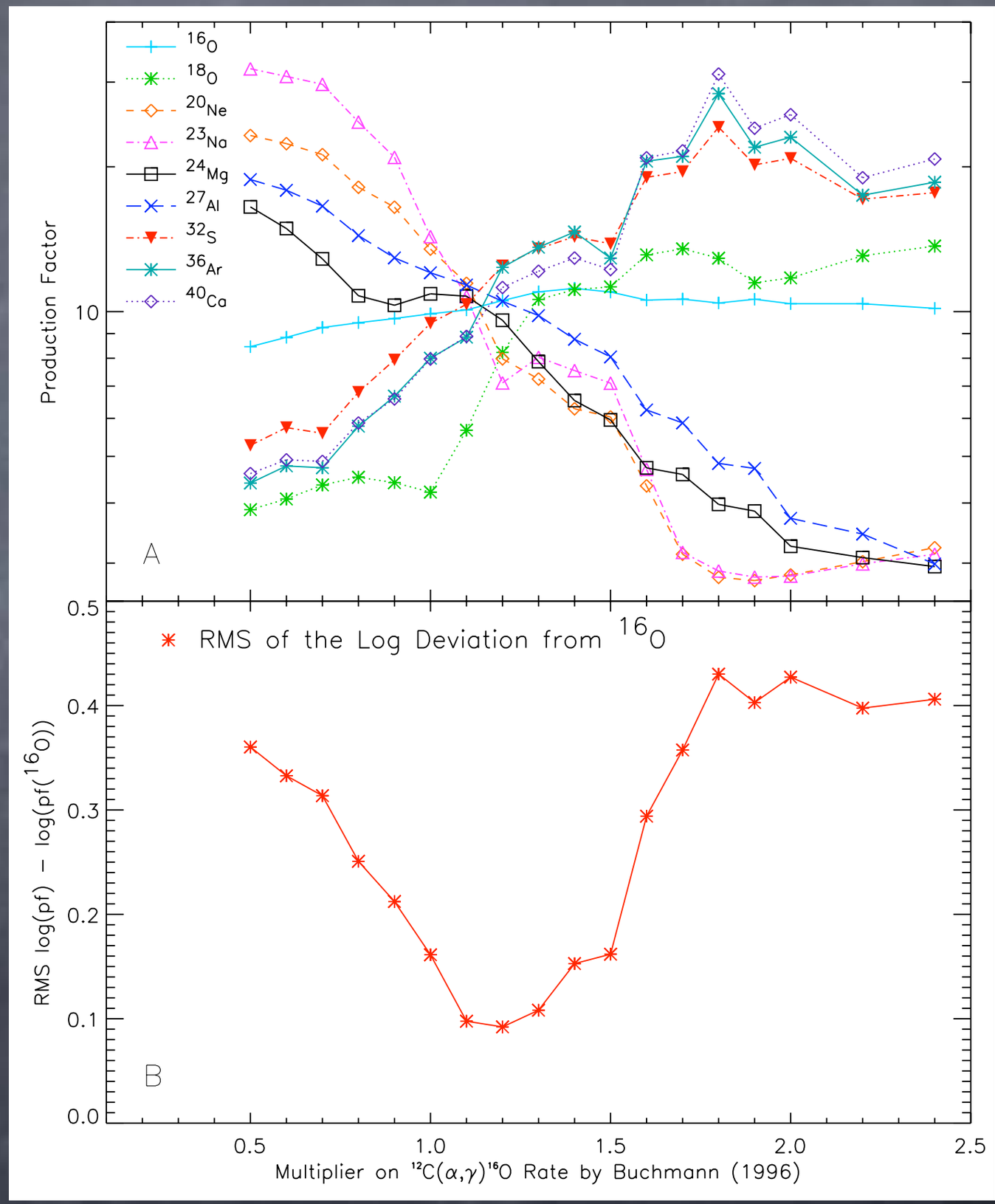
Reactive flows

- Physical and numerical errors are unavoidable in modeling thermonuclear reactive flows.

- Physical errors are introduced by using approximations to known pieces of physics, or by using estimates for incompletely known physics.

Reactive flows

- An incompleteness error committed by the networks in FLASH are the approximations used for the reaction rates.
- Many key rates aren't known experimentally to within a factor of 2-10.
- How large an error is made in a stellar model that uses these imprecisely known reaction rate is highly problem dependent.



Reactive flows

- Another physical error made by the small reaction networks in FLASH is that they are all approximations to using a large and complete reaction network.

- As we've noted, a carefully designed α-chain and heavy ion reaction network can usually reproduce the energy generation rate of a large nuclear reaction network to within 30%.

- For conditions where Y_e remains near 0.5 the accuracy is much better that 30%, but in regimes where Y_e differs from 0.5, the accuracy is significantly reduced.

Reactive flows

- Numerical errors are introduced by the algorithms and methods employed in a calculation.

- The goal for the reaction networks in FLASH is to reduce the numerical integration error below the formal accuracy of the hydrodynamic algorithm; to have the accuracy of the hydrodynamics be the limiting source of numerical error.

Train wreck at
Montparnasse, France, 1865

Reactive flows

- For fluid flows with very strong shock features, the absolute accuracy of the hydro algorithm is about two or three digits.

- For smooth fluid flows the absolute accuracy is better, perhaps to five or six digits, but highly problem dependent.

The Weir
Dhruva Mistry, 2005

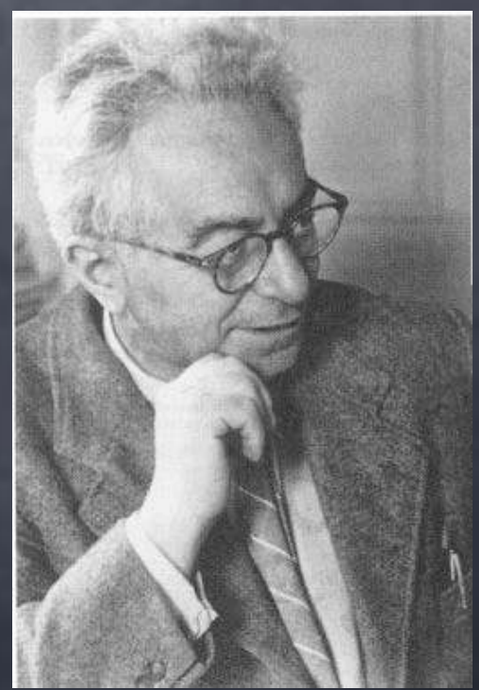
Reactive flows

- The time integration routines in the reaction networks permit an integration accuracy that is well below the formal accuracy of the hydrodynamic algorithm, even for smooth fluid flows.

- By calculating the approximate reactions networks as accurately as possible, we strive to eliminate the integration as a potential source of systematic numerical error.

Reactive flows

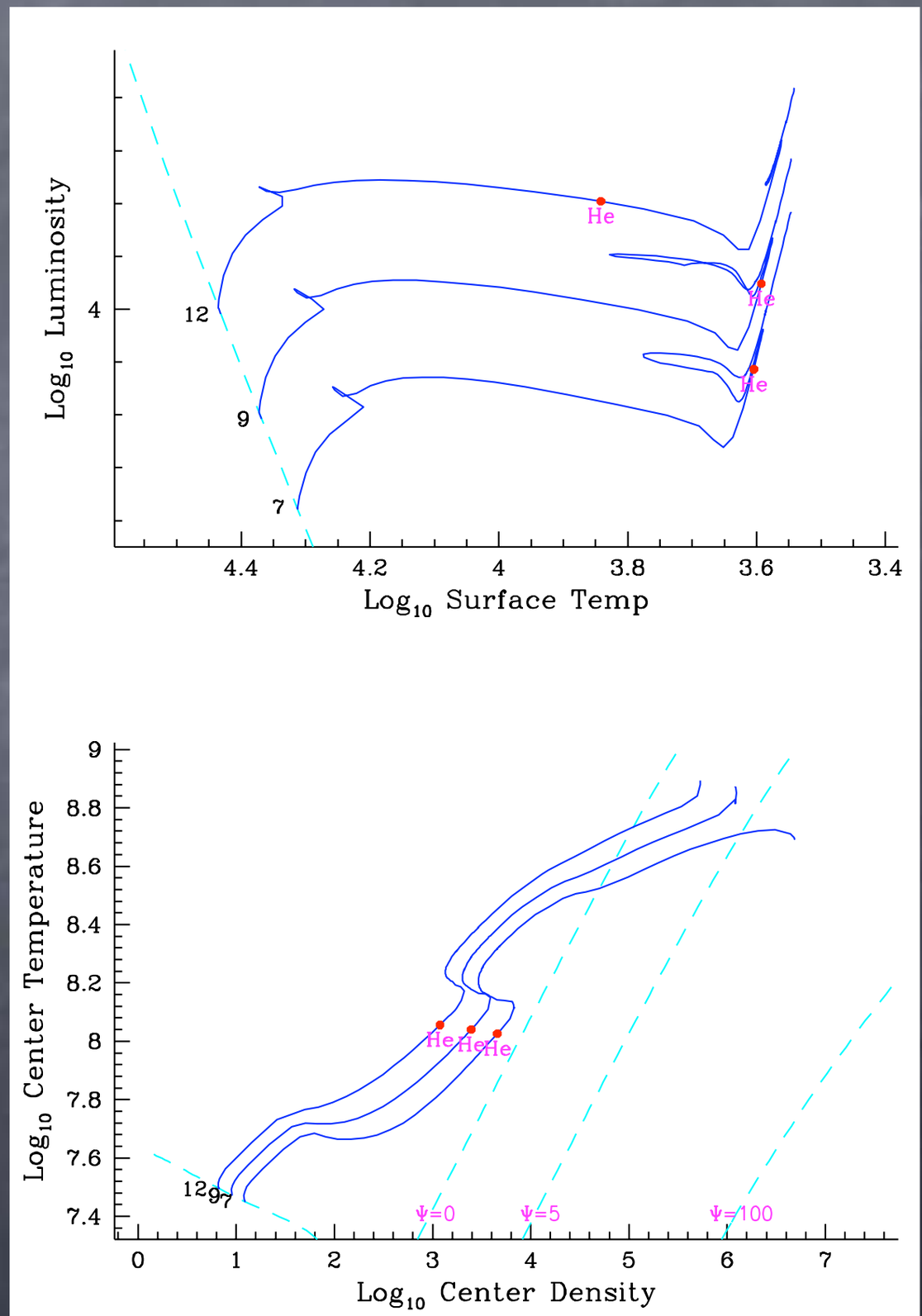
- FLASH is an explicit solver for compressible hydrodynamics. At best it is limited to time steps no larger than time it takes sound waves to cross a grid cell. So we are limited to "fast" phenomena - perhaps a few million sound crossing times.
- There are many instances where one is interested on evolutions on much longer time scales, such as stars or various thermal timescale events in stars (e.g., slow neutron captures).



Richard Courant
circa 1958

Bill Paxton

- EZ stellar evolution code
- Tioga



Kavli Institute for Theoretical Physics,
UC Santa Barbara

Interlude

General network

- All the previous networks (big bang, hydrogen burners, alpha chains) are examples of hardwired networks.

- Each of those networks are carefully crafted by hand.
  They have the advantage of being fast, but the disadvantage of being inflexible.

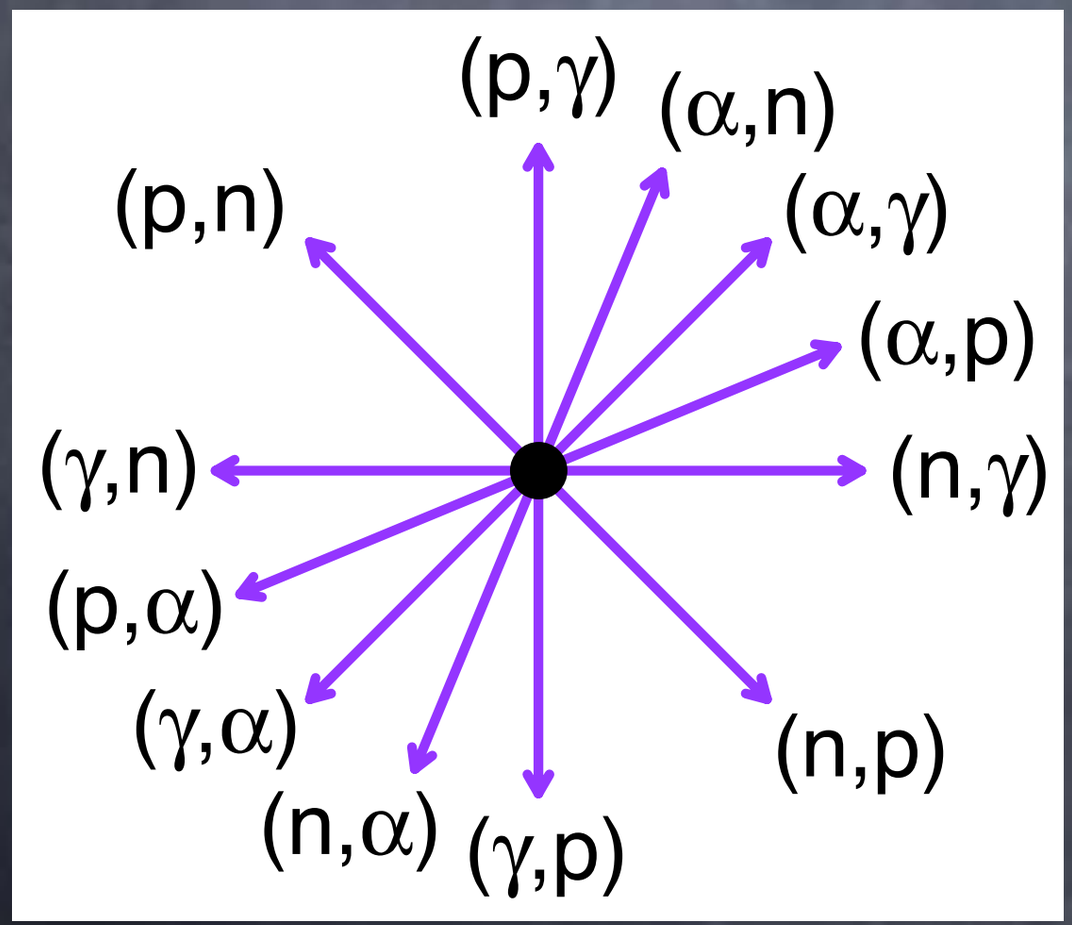
General network

- A general network, capable of doing any reaction network, is generally softwired.

Softwired
2005
kenn brown and chris wren

General network

- First, connect isotope i with isotope k by reaction type n




```
      subroutine naray

c..this routine builds the nrr(7,i) array, which specifies
c..the location of isotopes coupled to i by various reactions.
c..the first index on nrr refers to reactions of the form
c..  1=ng 2=pn 3=pg 4=ap 5=an 6=ag 7=b-

c..initialize
      jz(1) = 0
      jz(2) = 1
      jz(3) = 1
      jz(4) = 1
      jz(5) = 2
      jz(6) = 2
      jz(7) = -1

      jn(1) = 1
      jn(2) = -1
      jn(3) = 0
      jn(4) = 2
      jn(5) = 1
      jn(6) = 2
      jn(7) = 1

c..isotope i connected to isotope k by reaction type n
      do i=ionbeg,ionend
       do n=1,7
        nrr(n,i) = 0
        kz = int(zion(i)) + jz(n)
        kn = int(nion(i)) + jn(n)
        do k=ionbeg,ionend
         if (kz.eq.int(zion(k)) .and. kn.eq.int(nion(k))) nrr(n,i)=k
        enddo
       enddo
      enddo

      return
      end
```

General network

- Second, form the ODEs.

```
c..for every isotope in the network
      do j=ionbeg,ionend

c..set up the y(j)(n,g)y(k) and y(k)(g,n)y(j) components
        k = nrr(1,j)
        if (k .gt. 0) then
          b1 = -aan*sig(1,j)*y(j) + sig(2,j)*y(k)
          dydt(j) = dydt(j) + b1
          dydt(in) = dydt(in) + b1
          dydt(k) = dydt(k) - b1
        end if
```

General network

- Third, form the Jacobian

- Forth, evolve the system in time like any of our hardwired networks.

```
c..for every isotope in the network
      do j=ionbeg,ionend

c..set up the (n,g) components
        k = nrr(1,j)
        if (k .gt. 0) then
          a1 = sig(1,j) * aan
          a2 = sig(2,j)
          a3 = sig(1,j) * y(j)
          dfdy(j,j) = dfdy(j,j) - a1
          dfdy(j,k) = dfdy(j,k) + a2
          dfdy(j,in) = dfdy(j,in) - a3
          dfdy(k,j) = dfdy(k,j) + a1
          dfdy(k,k) = dfdy(k,k) - a2
          dfdy(k,in) = dfdy(k,in) + a3
          dfdy(in,j) = dfdy(in,j) - a1
          dfdy(in,k) = dfdy(in,k) + a2
          dfdy(in,in) = dfdy(in,in) - a3
        end if
```

Tasks for the day

- Download, compile, and run the torch code from www.cococubed.com/code_pages/burn.shtml
  How do you set which isotopes are in the network?

- Run one of the task problems from the previous four days with the torch network. Do you get the same answers?
  Why or why not?

- Extra credit: The reading the nuclear data file is a bit clumsy. Modify the code to use the NON-SMOKER data base of reaction rates. Modify the code to use the LAnganke et al weak reaction rates.

Tools and Toys in Nuclear Astrophysics