Jim he allowed the stars was made, but I allowed they happened. Jim said the moon could'a laid them; well, that looked kind of reasonable, so I didn't say nothin against it, because I've seen a frog lay almost as many, so of course it could be done.

Mark Twain

The Adventures of Huckleberry Finn

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

- pntpm.ulb.ac.be/nacre.htm
- quasar.physik.unibas.ch/~tommy/adndt.html
- www-pat.llnl.gov/Research/RRSN/
- www.nr.com
- www.netlib.org/utk/people/JackDongarra/la-sw.html

# Syllabus

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

# Last Lecture

- We may write our initial value problem as

$$\dot{Y}_i = \sum_j C_i R_j Y_j + \sum_{jk} \frac{C_i}{C_j! C_k!} \rho N_A \, R_{jk} \, Y_j \, Y_k + \sum_{jkl} \frac{C_i}{C_j! C_k! C_l!} \rho^2 N_A^2 \, R_{jkl} \, Y_j \, Y_k Y_l$$

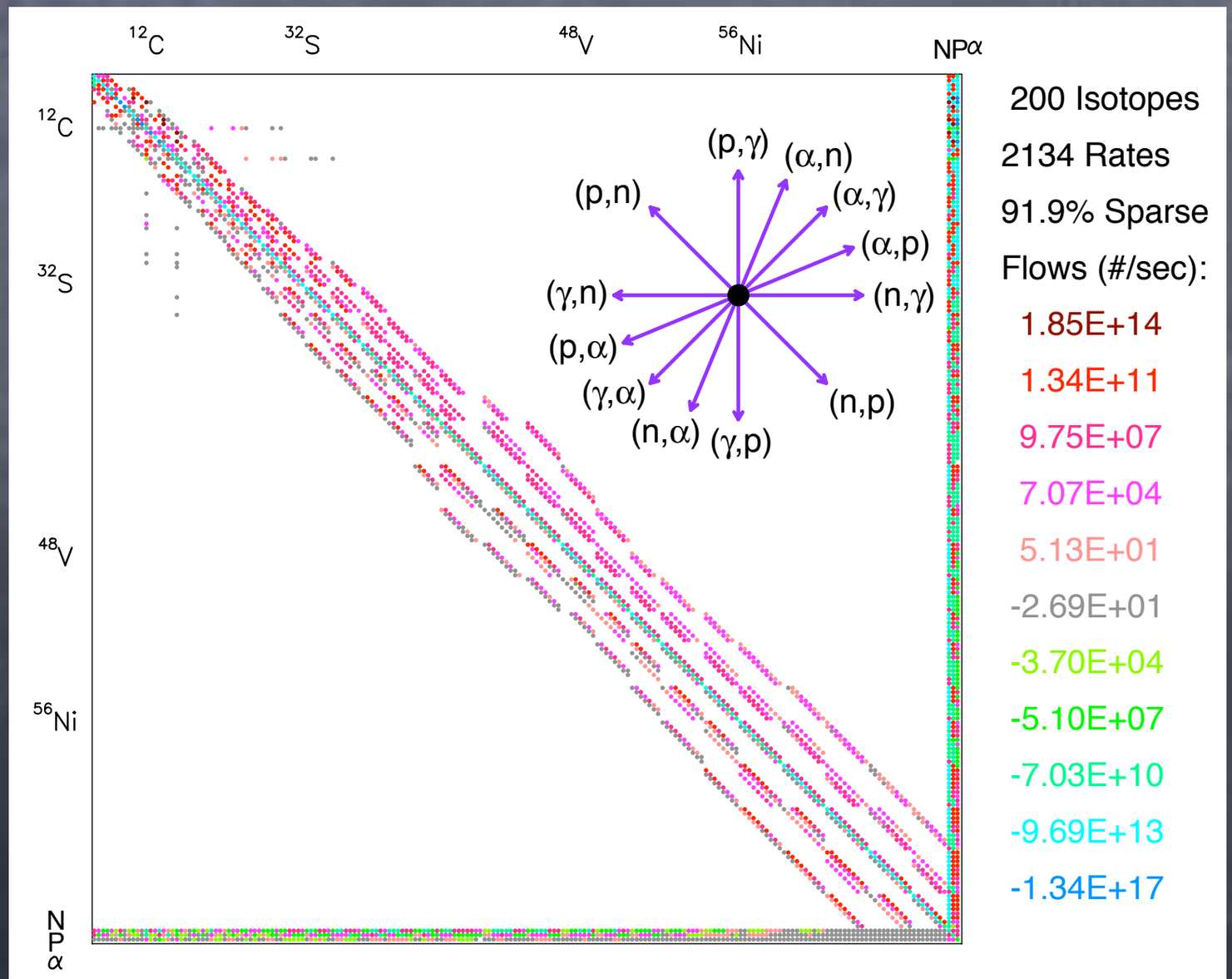
  or, in vector notation as

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y})$$

- The ODE coefficients span orders of magnitude since the reaction rates depend on high powers of the temperature, and since the abundances themselves may traverse orders of magnitude. As a result, nuclear reaction networks are "stiff".

# Last Lecture

- Thus, with a few important exceptions, for each nucleus we need only consider twelve reactions linking it to its nuclear neighbors by the capture of an n, p, α or γ and release a different one of these four.



# Last Lecture

- One of the most important consequences of changing the composition is the release (or absorption) of energy. The energy generation rate is given by

$$\dot{\epsilon}_{\rm nuc} = -\sum_i N_A M_i c^2 \dot{Y}_i - \dot{\epsilon}_\nu \quad ({\rm erg\ g^{-1}\ s^{-1}})$$

where $M_i c^2$ is the rest mass energy of species i.

- The mass of a nucleus is conveniently expressed in terms of tabulations of the the atomic mass excess

$$M_i = A_i m_\mu + M_{{\rm ex},i}$$

where $m_\mu$ is the atomic mass unit.

# Last Lecture

- This method has a minimum costs of 1 Jacobian and 8 right-hand side evaluations, 2 matrix reductions, and 10 backsubstitutions for a time step that meets the specified integration accuracy.

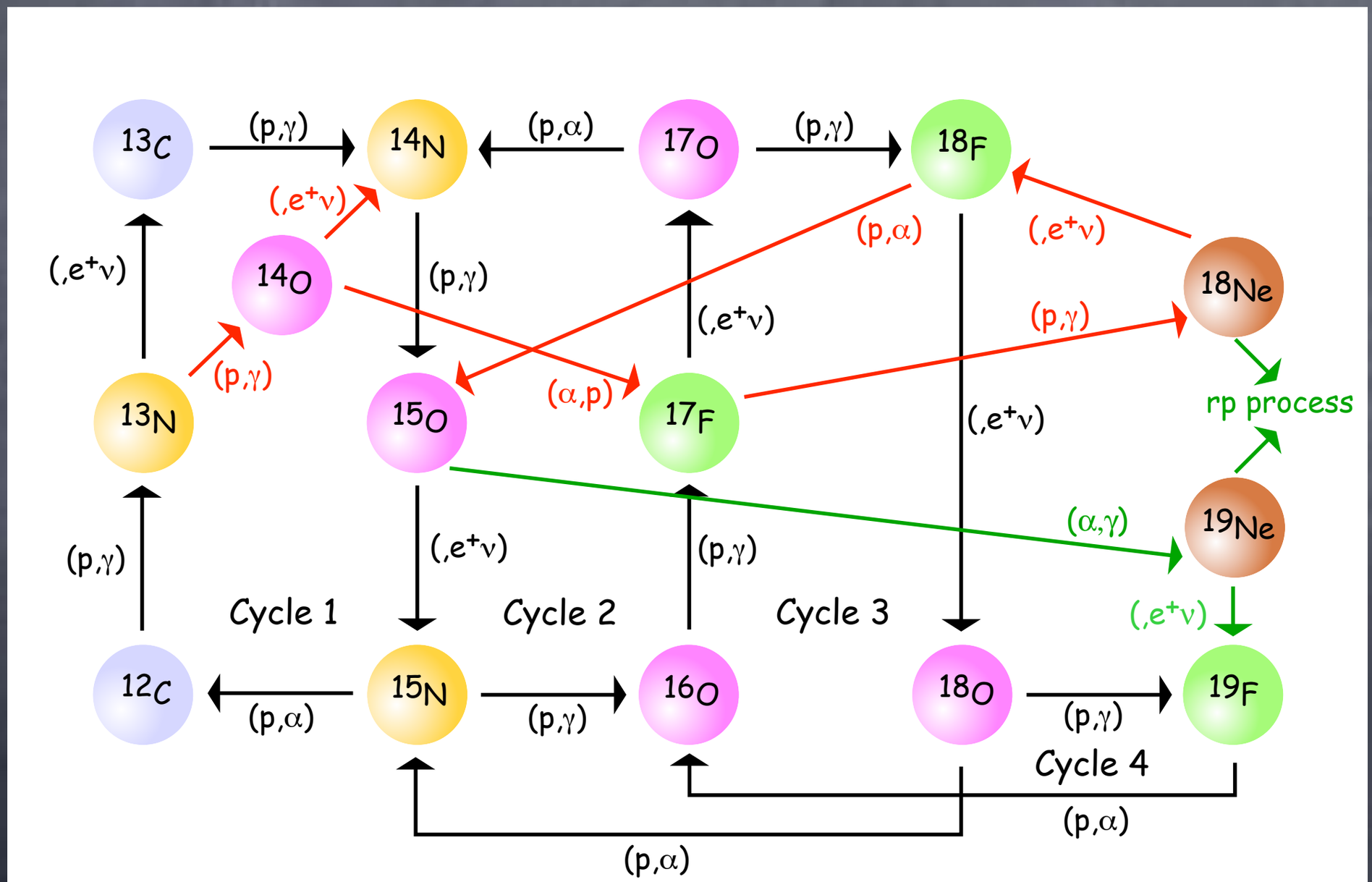
- The cost increases by 1 matrix reduction + m backsubstitutions for every order increase.

- The cost per step is at least twice as large as the simple Euler or Kaps-Rentrop method, but it may be more efficient globally if accurate steps are at least twice as big can be taken.

- The Bader-Deuflard method is used in the codes I'm providing for these JINA lectures.

# CNO cycles

- If $T_9 > 0.5$, then one breaks out of the beta-limited CNO cycles and begins a journey on the rapid-proton capture process.




CNO: $T_9 < 0.2$ Hot CNO: $0.2 < T_9 < 0.5$ rp process: $T_9 > 0.5$

# Abundance Variables

$$\text{neutron excess}: \quad \eta = \sum (A_i - 2Z_i) Y_i$$

$$(\text{pure protons}) - 1 \leq \eta \leq 1 (\text{pure neutrons})$$

Let $f_i$ be the fraction of isotope i that is ionized.

$f_i = 0$ = neutral atom. $f_i = 1$ = fully ionized.

$$\text{electron fraction}: \quad Y_e = \frac{n_e}{N_A \rho} = \sum Z_i f_i Y_i$$

$$Y_e = (1 - \eta)/2 = \overline{Z}/\overline{A} \quad \text{if fully ionized}$$

# Abundance Variables

$$\mu_{ion} = \frac{1}{\sum Y_i} = \overline{\mathrm{A}}$$

$$\mu_{ele} = \frac{1}{\sum Y_i \mathrm{Z}_i f_i} = \frac{1}{Y_e}$$

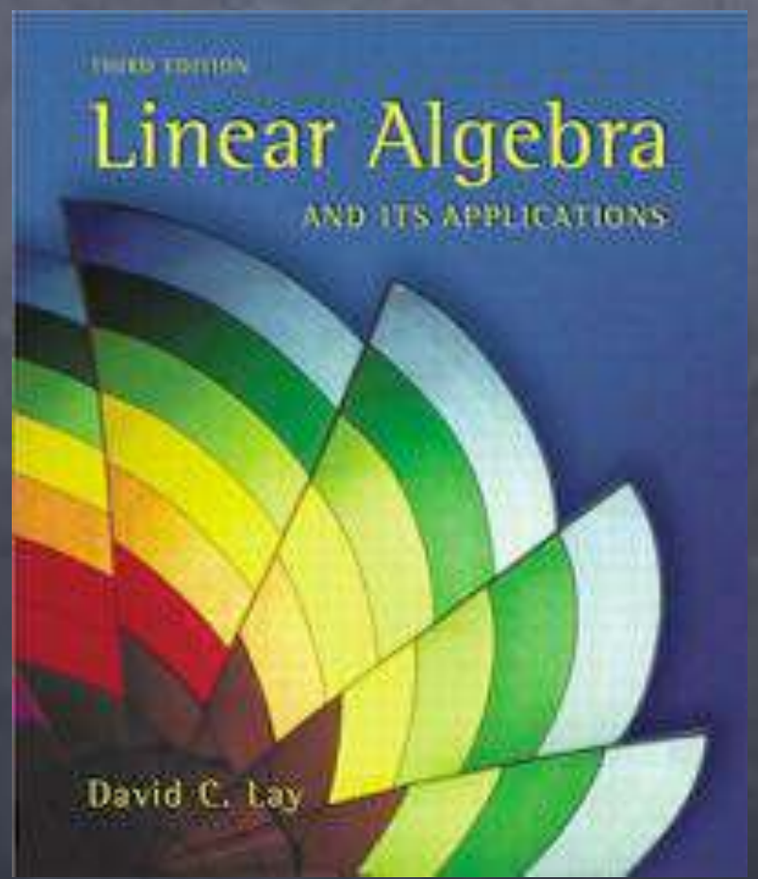
mean molecular weight : $\mu = \left[\frac{1}{\mu_{ion}} + \frac{1}{\mu_{ele}}\right]^{-1} = \frac{1}{\sum Y_i (Z_i f_i + 1)}$

$$\mu = \frac{\overline{A}}{\overline{Z} + 1} \quad \text{if fully ionized}$$

# Linear algebra

- Since we must use an implicit integration method we'll be solving (large) systems of linear equations. As the linear algebra will generally dominate the time to obtain a solution, we'll want to use efficient solvers.

- Over the next few slides we'll briefly examine an example of a dense solver, direct sparse solver, and an iterative sparse solver.



# Linear algebra

- LEQS is a routine which solves a system of linear equations by Gaussian elimination, the method you probably first learned.

- Matrix $\tilde{A}$ is reduced to upper triangular form in tandem with a right-hand side b by Gaussian elimination, and backsubstitution on the upper triangular matrix yields the solution to $\tilde{A} \cdot x = b$.

$$\begin{bmatrix} 2 & -1 & 1 \\ -2 & 2 & -3 \\ 2 & -4 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -7 \\ 3 \end{bmatrix}$$

1. Step: annihilate $x_1$ from rows 2, 3

new row 2 = row 1 + row 2

new row 3 = row 3 − row 1

$$\begin{bmatrix} 2 & -1 & 1 \\ 0 & 1 & -2 \\ 0 & -3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -4 \\ 0 \end{bmatrix}$$

2. step: annihilate $x_2$ from row 3

new row 3 = row 3 + 3 · row 2

$$\begin{bmatrix} 2 & -1 & 1 \\ 0 & 1 & -2 \\ 0 & 0 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -4 \\ -12 \end{bmatrix}$$

$\Rightarrow x_3 = 3$

$x_2 - 2x_3 = 4 \Rightarrow x_2 = 2$

$2x_1 - x_2 + x_3 = 3 \Rightarrow x_1 = 1$

# Linear algebra

- The origin of this legacy routine (LEQS) is somewhat obscure, in use by at least 1962, and is probably the most common linear algebra package presently used for evolving reaction networks.

- LEQS is used in the codes I'm providing for the JINA lectures.

Ford-Seattle
1962

# Linear algebra

- The maximum element in each row serves as the pivot element, but no row or column interchanges are performed, so LEQS may become unstable if used on matrices that are not diagonally dominant.

- A small amount of effort is devoted to minimizing calculations with matrix elements that are zero.

- All Gaussian elimination routines have the disadvantage that for a staged sequence of right-hand sides, the entire matrix must be decomposed for each right-hand side.

# Linear algebra

- Suppose we are able to write the matrix $\tilde{A}$ as a product of two matrices, $\tilde{A} = \tilde{L} \cdot \tilde{U}$, where L is a lower triangular matrix (elements on the diagonal and below) and U is an upper triangular matrix (elements on the diagonal and above).

- We could use such a decomposition to solve the linear set

$$\tilde{A} \cdot \mathbf{x} = \left(\tilde{L} \cdot \tilde{U}\right) \cdot \mathbf{x} = \tilde{L} \cdot \left(\tilde{U} \cdot \mathbf{x}\right) = \mathbf{b}$$

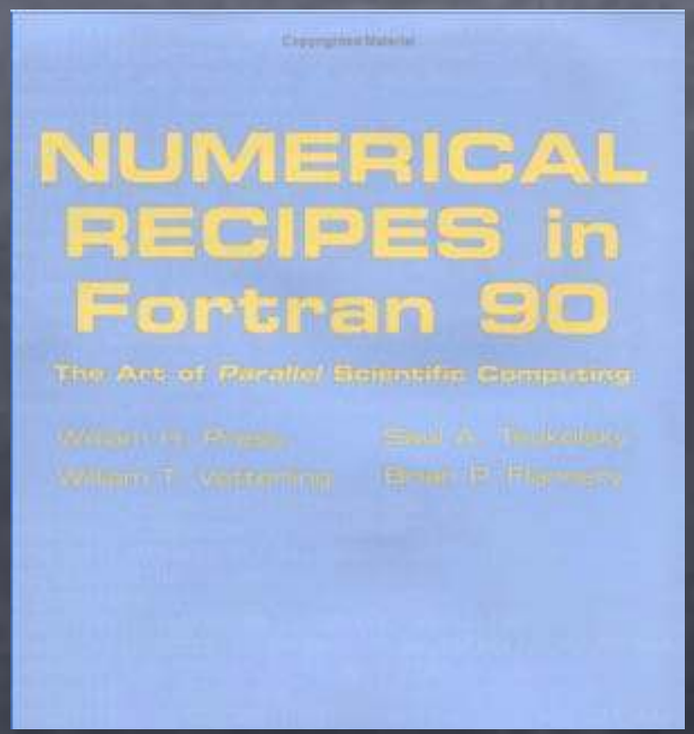
By first solving $\tilde{L} \cdot \mathbf{y} = \mathbf{b}$ for y and then $\tilde{U} \cdot \mathbf{x} = \mathbf{y}$ for x.

# Linear algebra

- What is the advantage of breaking up one linear set into two successive ones?

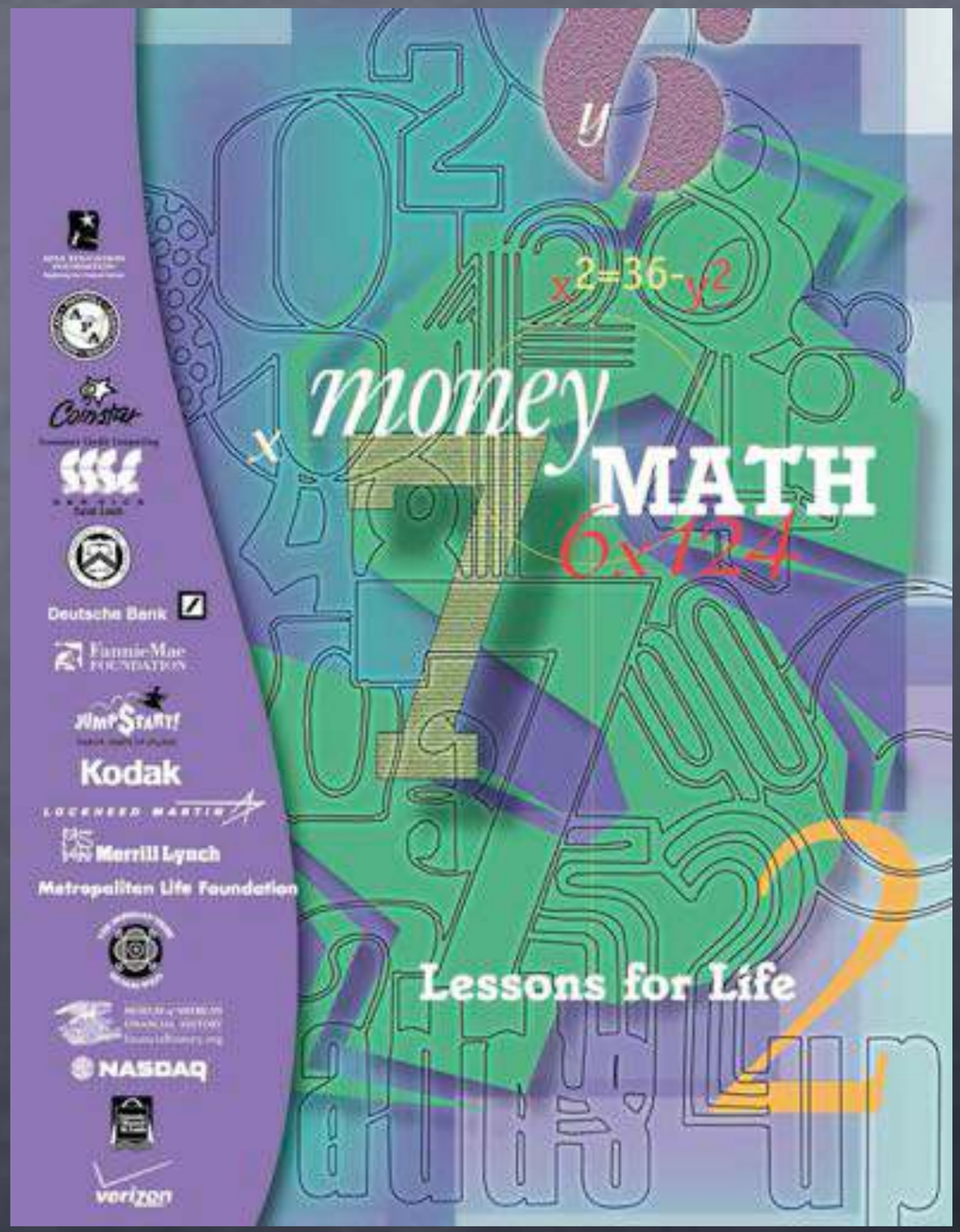
- One advantage is that the solution of a triangular system is trivial, as we have just seen in Gaussian elimination.

- Another advantage is that once we have the LU decomposition of a matrix, we can solve for as many right hand sides as we want, one at a time.



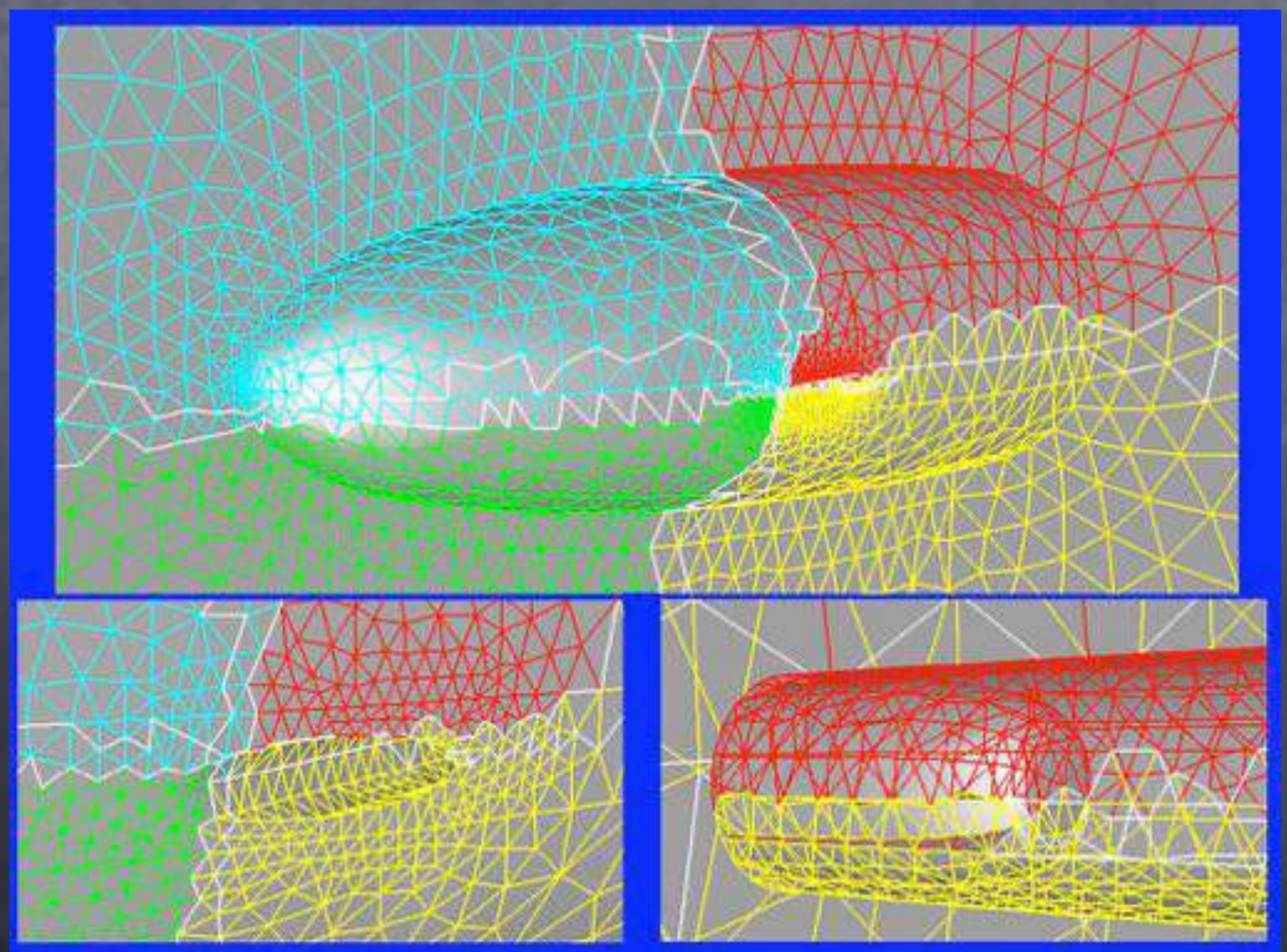
# Linear algebra

- There are two methods for solving sparse linear systems of equations; direct and iterative.

# Linear algebra

- Direct methods for sparse matrices typically divide the solution of $\tilde{A} \cdot x = b$ into a symbolic LU decomposition, numerical LU decomposition, and a backsubstitution phase.



Domain decomposition for four processors for a torpedo launch calculation. 1996, NRL.

# Linear algebra

- In the symbolic LU decomposition phase the matrix is not (usually) decomposed; only the steps to do so are stored.

- The pivot order is determined, and a sequence of decomposition operations which minimize the amount of fill-in is recorded.

- Fill-in refers to zero matrix elements which become nonzero (e.g., a sparse matrix times a sparse matrix is generally a denser matrix).

# Linear algebra

- Since the nonzero pattern of a nuclear reaction network (generally) does not change, the symbolic LU decomposition is a one-time initialization cost for a given pivot ordering (diagonal).

- In the numerical LU decomposition phase, a matrix with the same pivot order and nonzero pattern as a previously factorized matrix is numerically decomposed into its lower-upper form.

- This phase must be done only once for each staged set of linear equations.

# Linear algebra

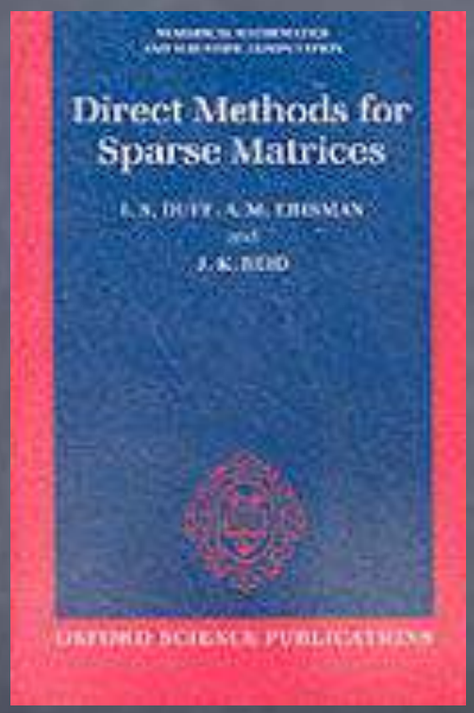
- In the backsubstitution phase, a set of linear equations is solved with the factors calculated from a previous numerical decomposition.

- The backsubstitution phase may be performed with as many right-hand sides as needed, and not all of the right-hand sides need to be known in advance.

- Most sparse matrix packages accept the nonzero entries of the matrix in three vectors, i, j, $a_{ij}$. Most sparse routubes consume about 50 to 99.999% less storage than a dense matrix.

# Linear algebra

- MA28 is the Coke classic of sparse matrix solvers.
  hsl.rl.ac.uk/archive/hslarchive.html
  Duff, Erisman & Reid "Direct Methods for Sparse Matrices".




- UMFPACK is a modern, direct sparse matrix solver.
  www.cise.ufl.edu/research/sparse/umfpack

# Linear algebra

- Iterative, or matrix-free, methods seek to minimize a function whose gradient is typically $\tilde{A} \cdot x - b$ and equal to zero when the function is minimized.

- These methods are attractive because they tend to only require matrix-times-vector operations and usually have smaller storage requirements than direct methods.

# Linear algebra

- However, the number of iterations required to converge to a solution is not known a priori, and generally increases with the number of unknowns.

- The total number of iterations, hence the overall speed, depends crucially on the initial guess and on the stringency of the convergence criteria.

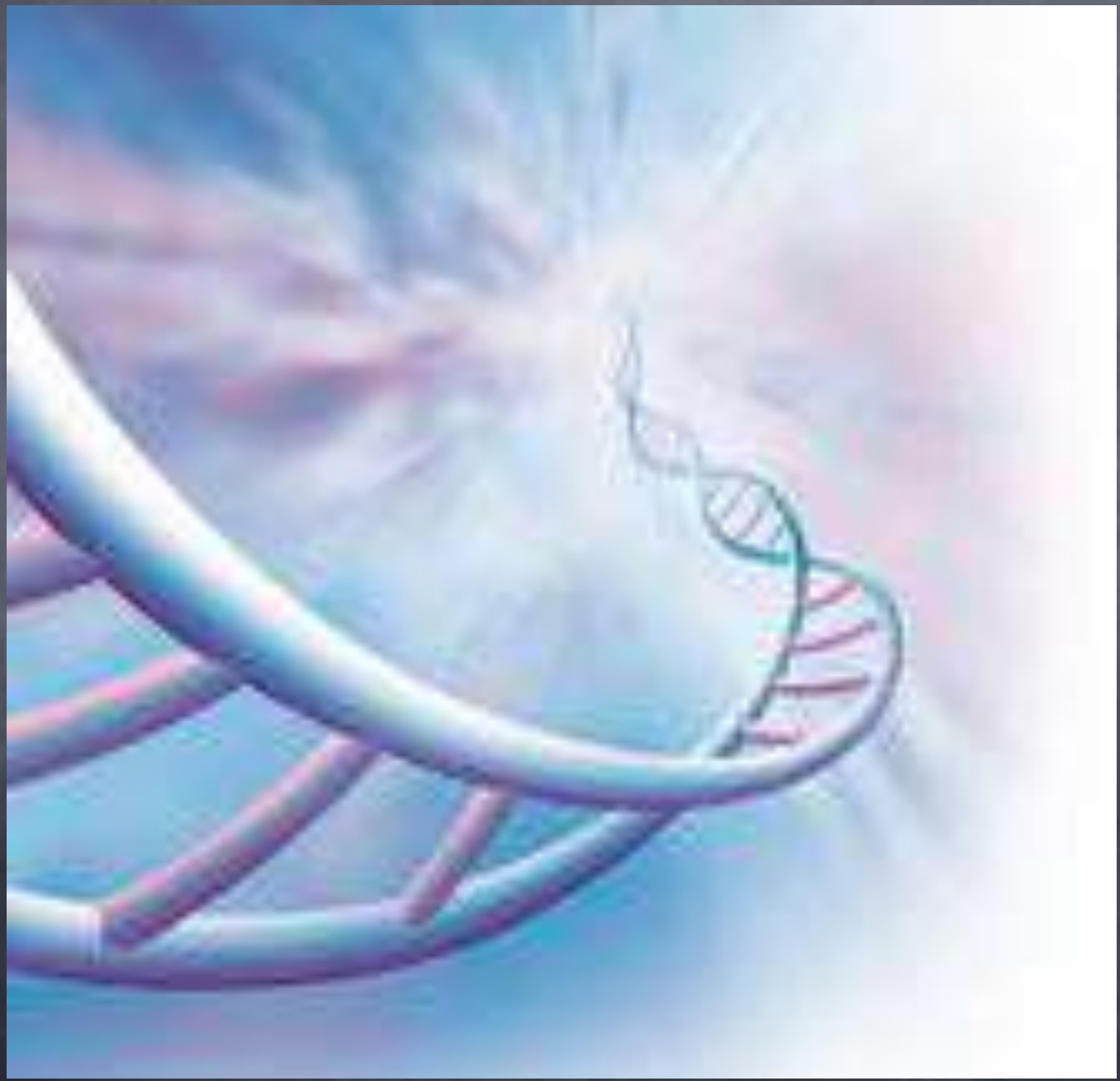
# Linear algebra

- BiCG is described by Barret et al in "Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods". netlib2.cs.utk.edu/linalg/html_templates/Templates.html

- SPARSKIT is a modern, iterative sparse matrix solver. www-users.cs.umn.edu/~saad/software/SPARSKIT/sparskit.html


- Both methods generate a sequence of vectors for the matrix $\tilde{A}$ and another sequence for the transpose matrix $\tilde{A}^T$. These vector sequences are the residuals of the iterations and are made mutually orthogonal, or bi-orthogonal.

# Interlude

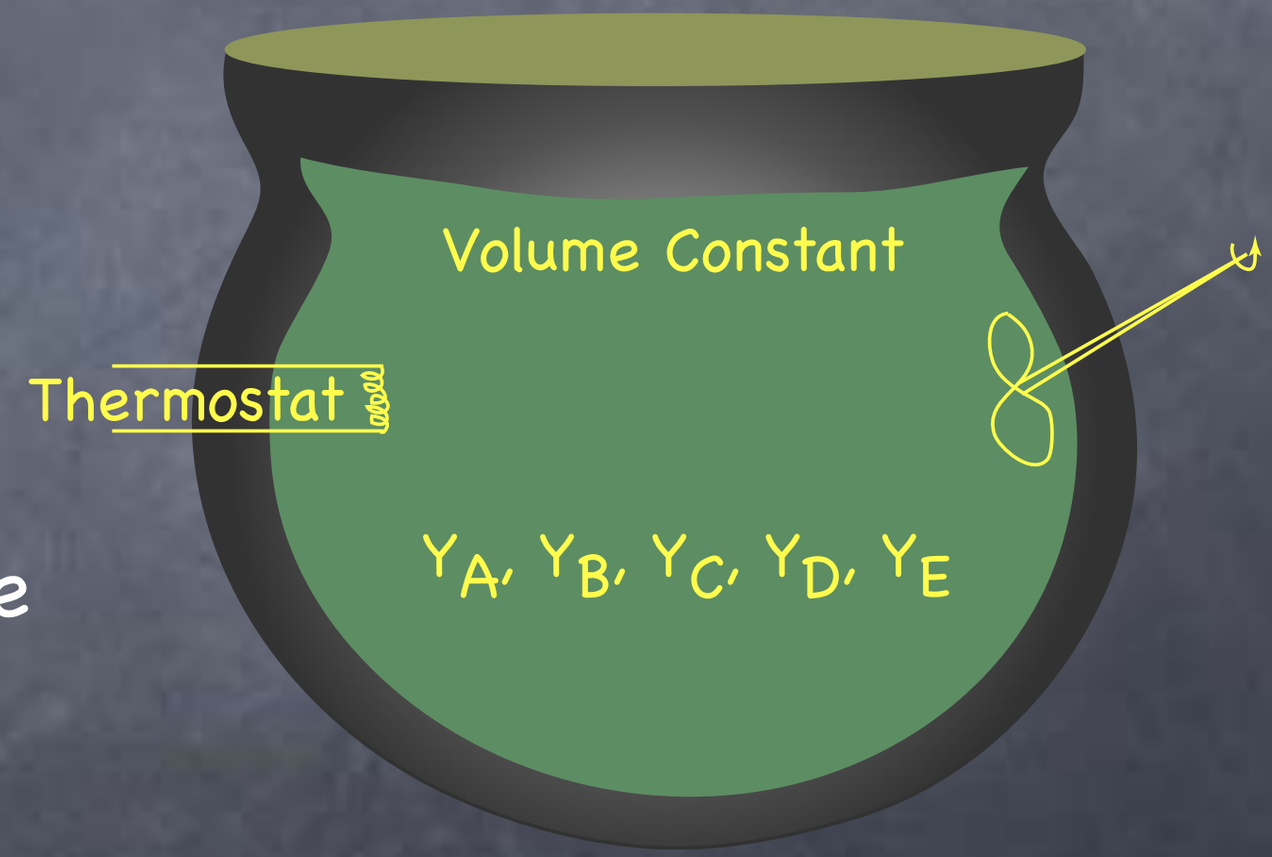


# Thermodynamic trajectories

- We've been considering situations where the temperature and density history are constant. This is the same as saying

$$\frac{dT}{dt} = 0 \qquad \frac{d\rho}{dt} = 0$$

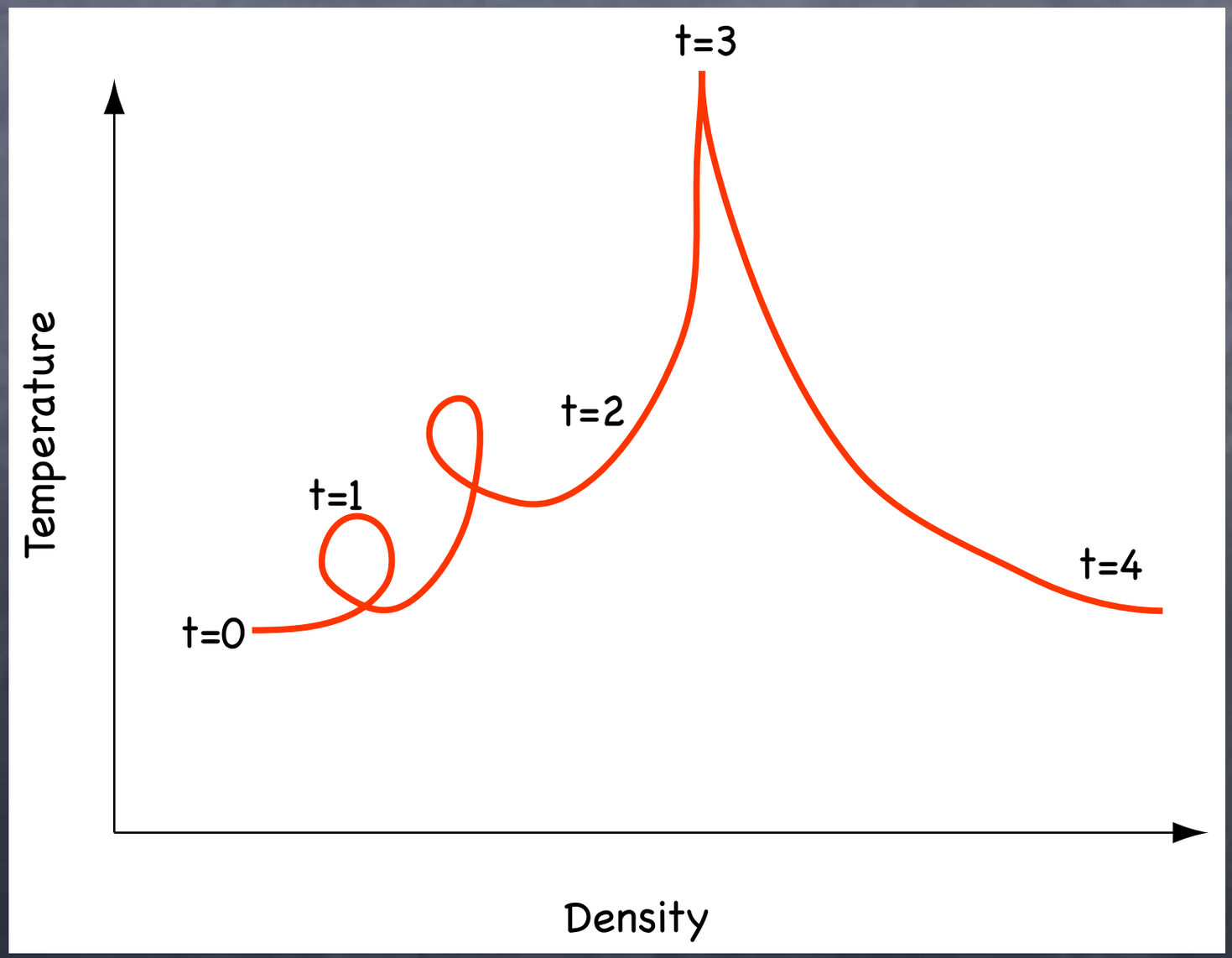
- Such conditions are called "hydrostatic" burning, since the local energy release doesn't change the temperature or density, as during the hydrostatic phases of a star's evolution.




- We now wish to relax these assumptions.

# Thermodynamic trajectories

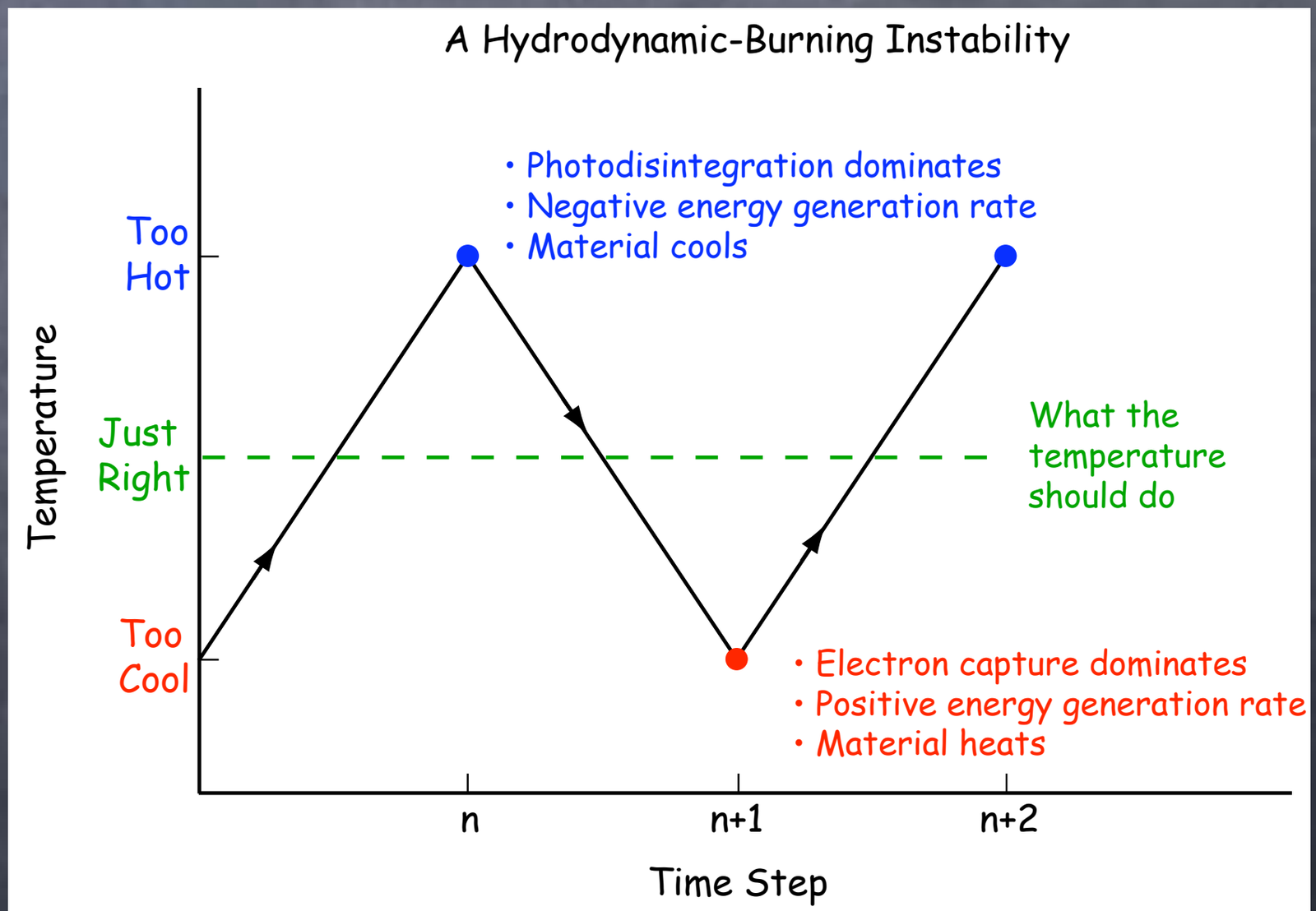
- This presents us with a choice. We can evolve the temperature and density equations separately and then evolve the reaction network, or we can evolve the temperature and density equations simultaneously with the reaction network.



# Thermodynamic trajectories

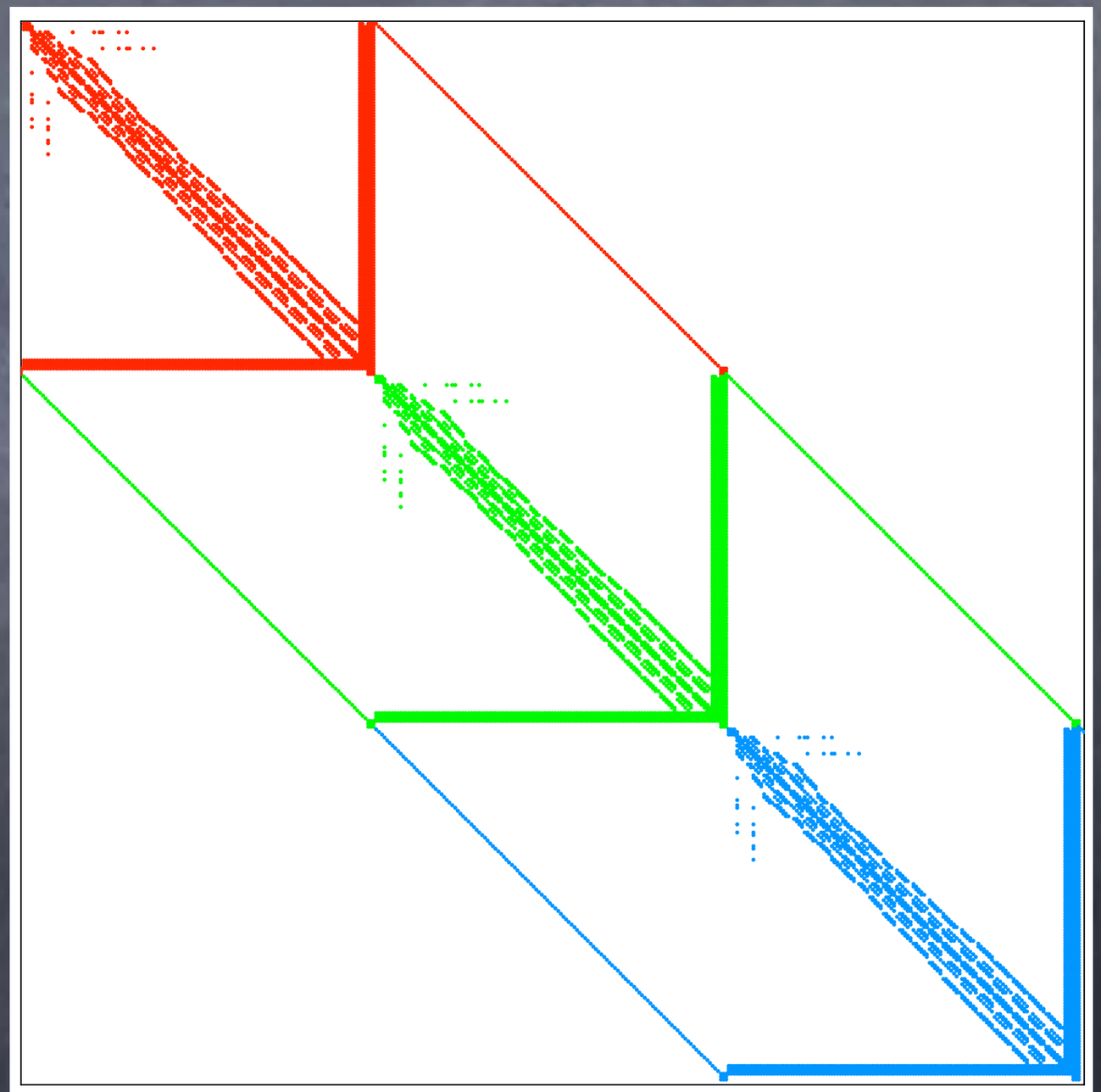
- The first choice is called "operator splitting", and assumes that the timescale for a temperature or density change is much longer than the burning timescale. Operator splitting is easy to implement and, by far, the most common choice.

An example when the assumption of loose coupling breaks down.



# Thermodynamic trajectories

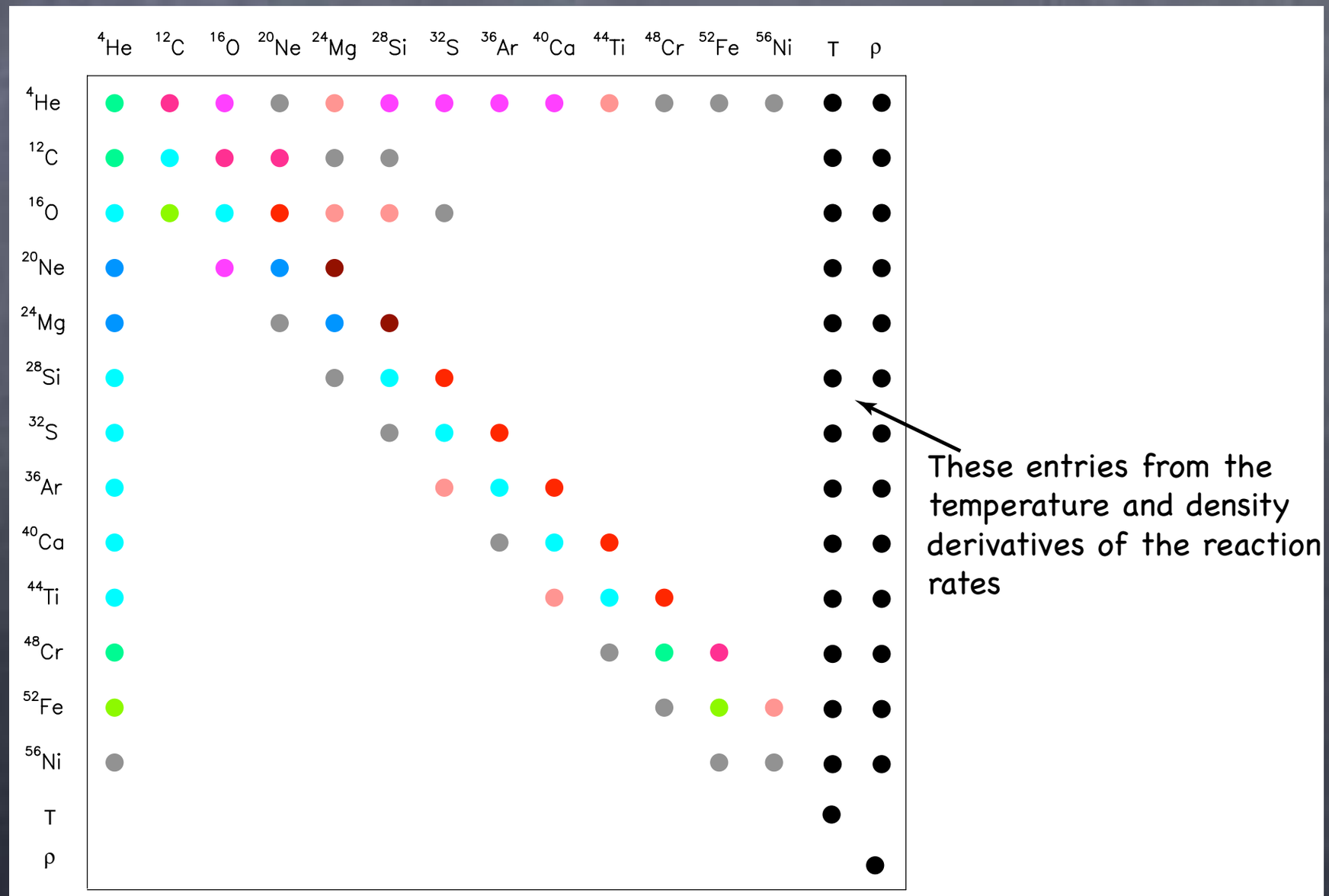
- The second choice is called "unsplit". It avoids the coupling between processes issue, but has the disadvantage of being more difficult to implement, particularly for implicit integrations.



Example of coupling three burning zones via diffusive processes.

# Thermodynamic trajectories

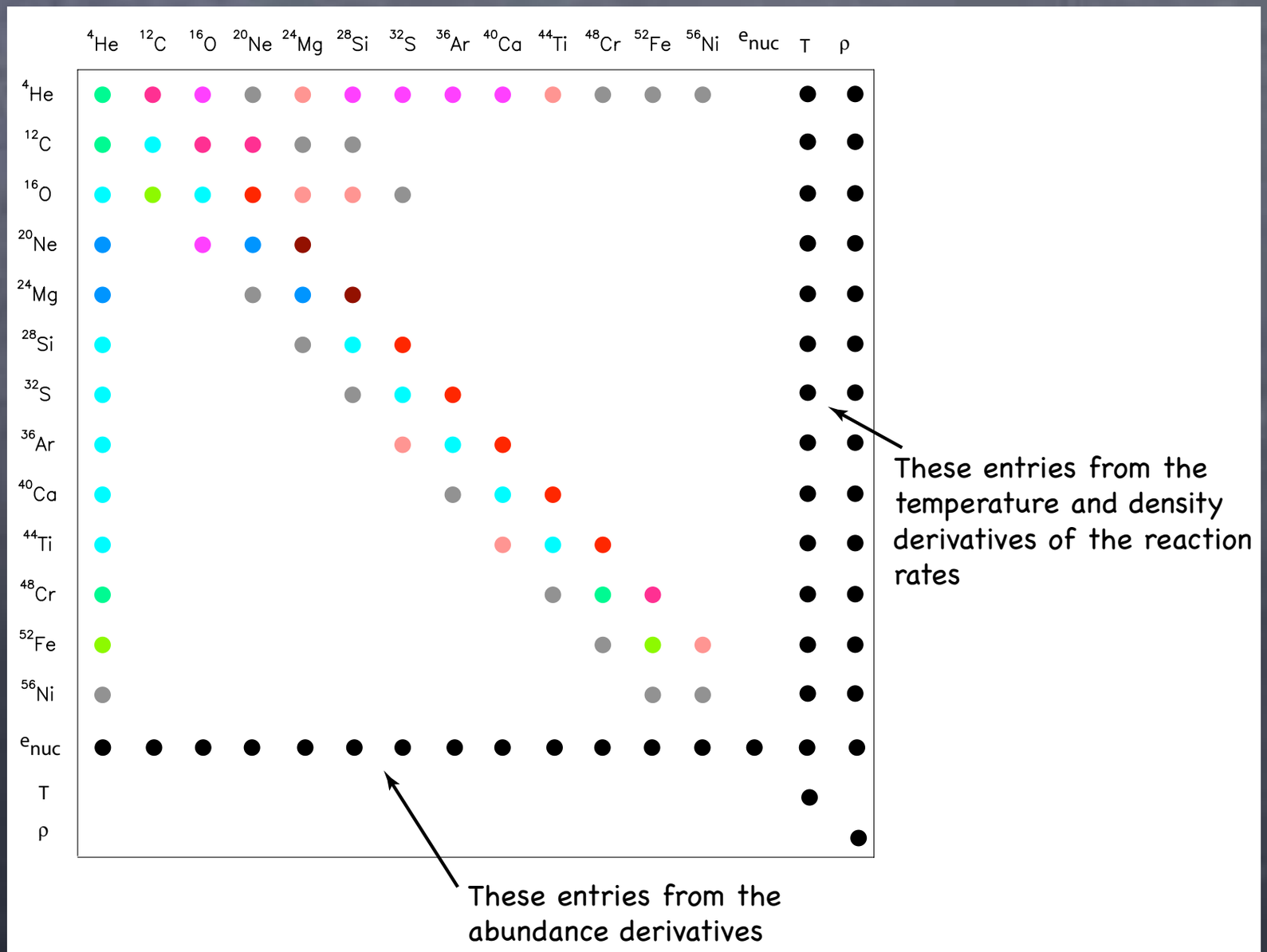
- The reaction network that I'm distributing for this JINA school uses the unsplit method. The hydrostatic burning ODEs dT/dt=0 and dρ/dt=0 are easy to append to our reaction network. The Jacobian then looks like this:



# Thermodynamic trajectories

- Its usually improves mass and energy conservation to append the energy generation rate to our set of ODEs

$$\dot{\epsilon}_{\rm nuc} = -\sum_i N_A M_i c^2 \dot{Y}_i - \dot{\epsilon}_\nu$$



# Thermodynamic trajectories

- A type of burning called "explosive burning" models a region where a shock has heated and compressed the material to some peak temperature $T_0$ and density $\rho_0$.

- This region subsequently expands adiabatically (if the energy from shock heating exceeds that from nuclear burning) as a radiation dominated gas.

- The density and temperature decline with time over a hydrodynamic timescale (a free-fall timescale)

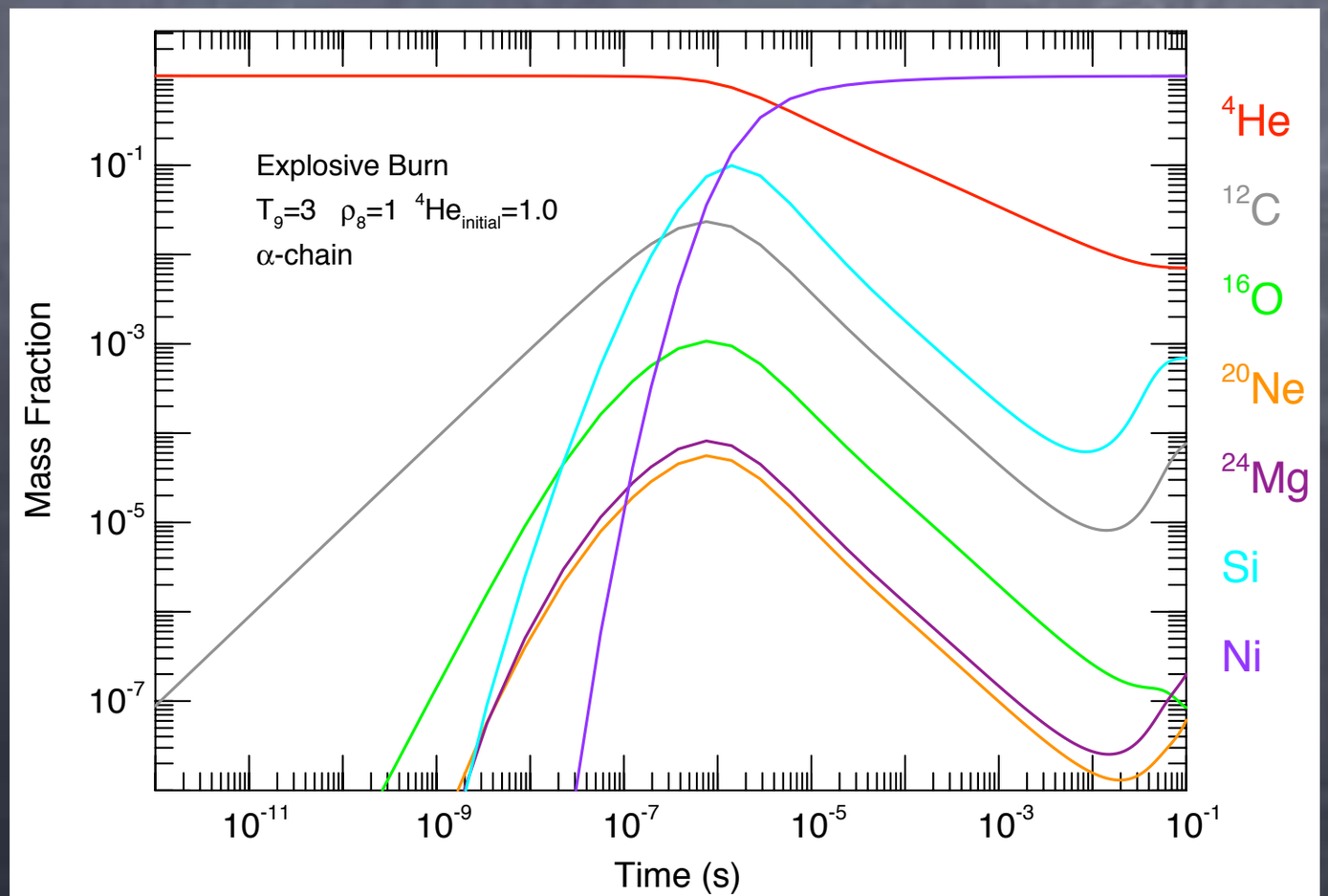
$$\tau_\rho = \frac{1}{24\pi G\rho} = \frac{446.0}{\sqrt{\rho}} \qquad \tau_T = 3\tau_\rho$$

# Thermodynamic trajectories

- For explosive nucleosynthesis we thus take

$$\frac{d\rho}{dt} = -\frac{\rho}{\tau_\rho} \qquad \frac{dT}{dt} = -\frac{T}{\tau_T}$$

$$\rho = \rho_0 \exp\left(-\frac{\rho}{\tau_\rho}\right) \qquad T = T_0 \exp\left(-\frac{T}{\tau_T}\right)$$



# Thermodynamic trajectories

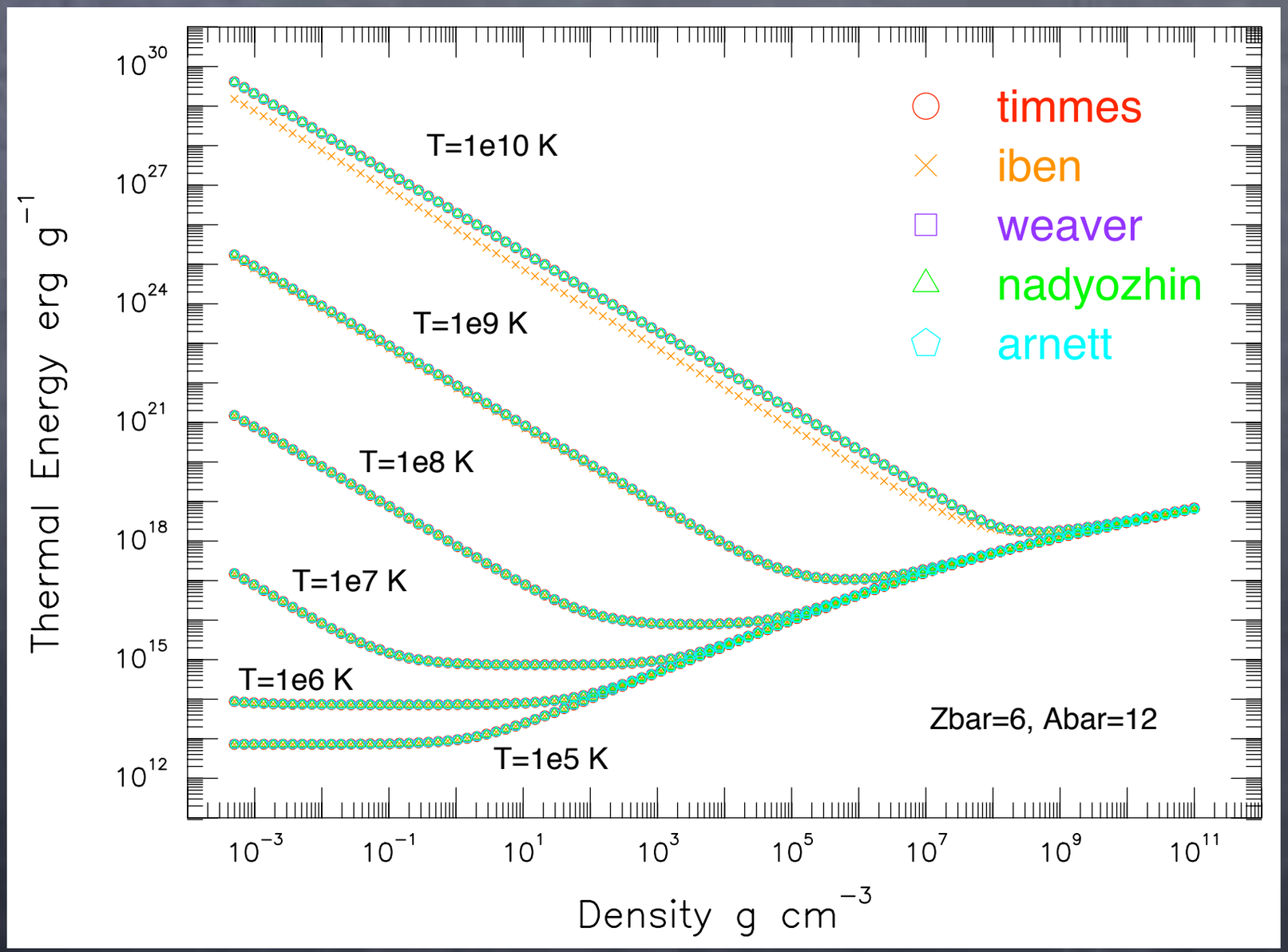
- Nucleosynthetic changes in composition and the resultant energy release produce local changes in hydrodynamic quantities like pressure and temperature.

- The strongest of these local couplings is the release (or absorption) of energy and the resultant change in temperature.

- Changes in temperature are particularly important because of the exponential nature of the temperature dependence of thermonuclear reaction rates.

# Thermodynamic trajectories

- One way to model this in a reaction network is to deposit the energy generated into internal energy and derive an ODE that gives the temperature in accordance with an equation of state.



# Thermodynamic trajectories

- Sometimes this model is called "self-heating". From basic thermodynamics we have

$$dE = \frac{\partial E}{\partial T} dT + \frac{\partial E}{\partial \rho} d\rho + \sum \frac{\partial E}{\partial Y_i} dY_i$$

applying the 1/dt operator and assuming the equation of state composition dependence is characterized $\overline{A}$ and $\overline{Z}$ leads to

$$\epsilon_{nuc} = c_V \frac{dT}{dt} + \frac{\partial E}{\partial \rho} \frac{d\rho}{dt} + \sum \frac{\partial E}{\partial \overline{A}} \frac{\partial \overline{A}}{\partial Y_i} \frac{dY_i}{dt} + \sum \frac{\partial E}{\partial \overline{Z}} \frac{\partial \overline{Z}}{\partial Y_i} \frac{dY_i}{dt}$$

# Thermodynamic trajectories

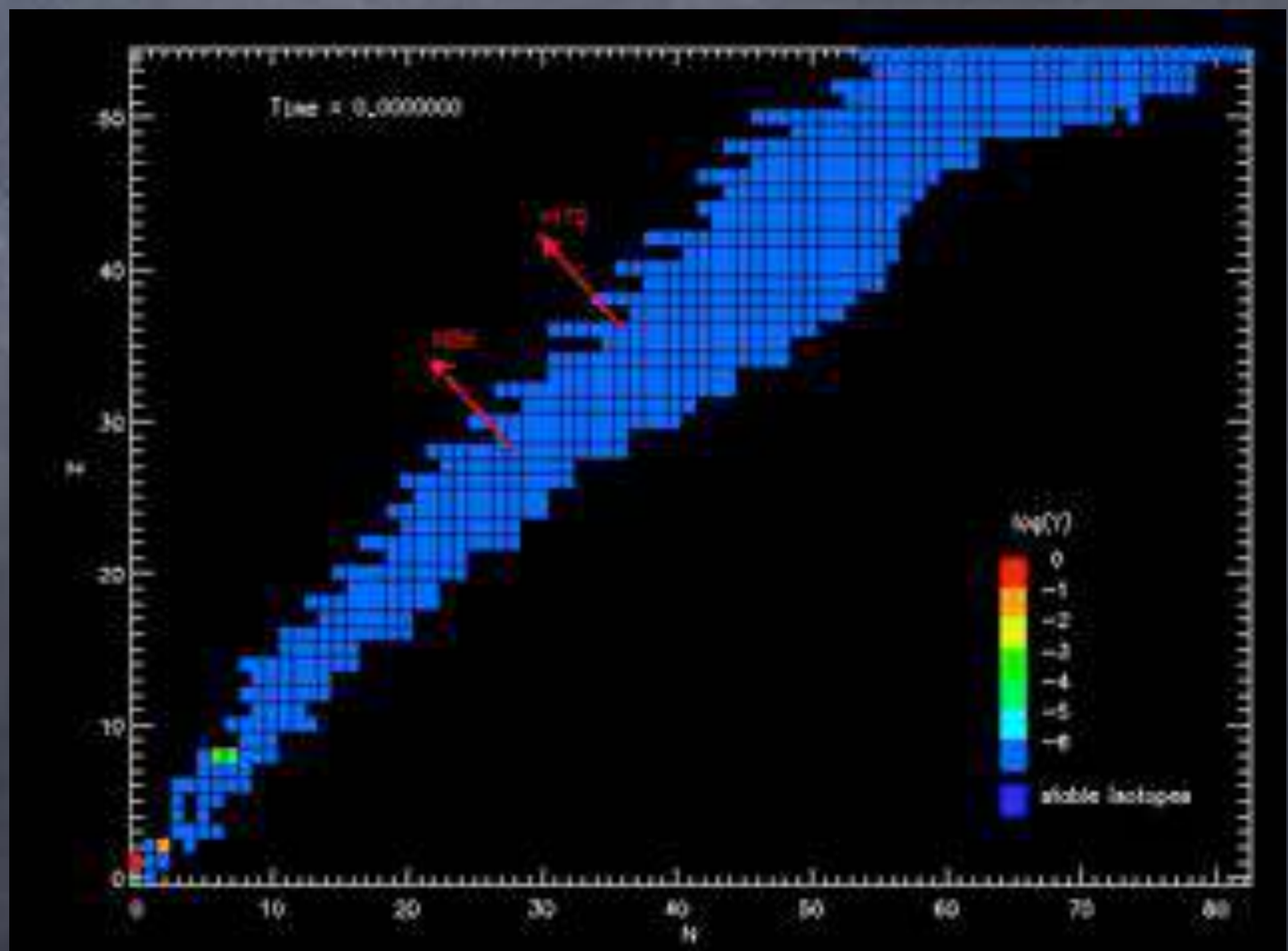
- Solving for dT/dt yields our self-consistent temperature ODE

$$\frac{dT}{dt} = \frac{1}{c_V}\left[\epsilon_{nuc} - \frac{\partial E}{\partial \rho}\frac{d\rho}{dt} - \frac{\partial E}{\partial \overline{A}}\overline{A}^2 \sum \frac{dY_i}{dt} - \frac{\partial E}{\partial \overline{Z}}\overline{A}\sum (Z_i - \overline{Z})\frac{dY_i}{dt}\right]$$

$$\frac{d\rho}{dt} = 0$$

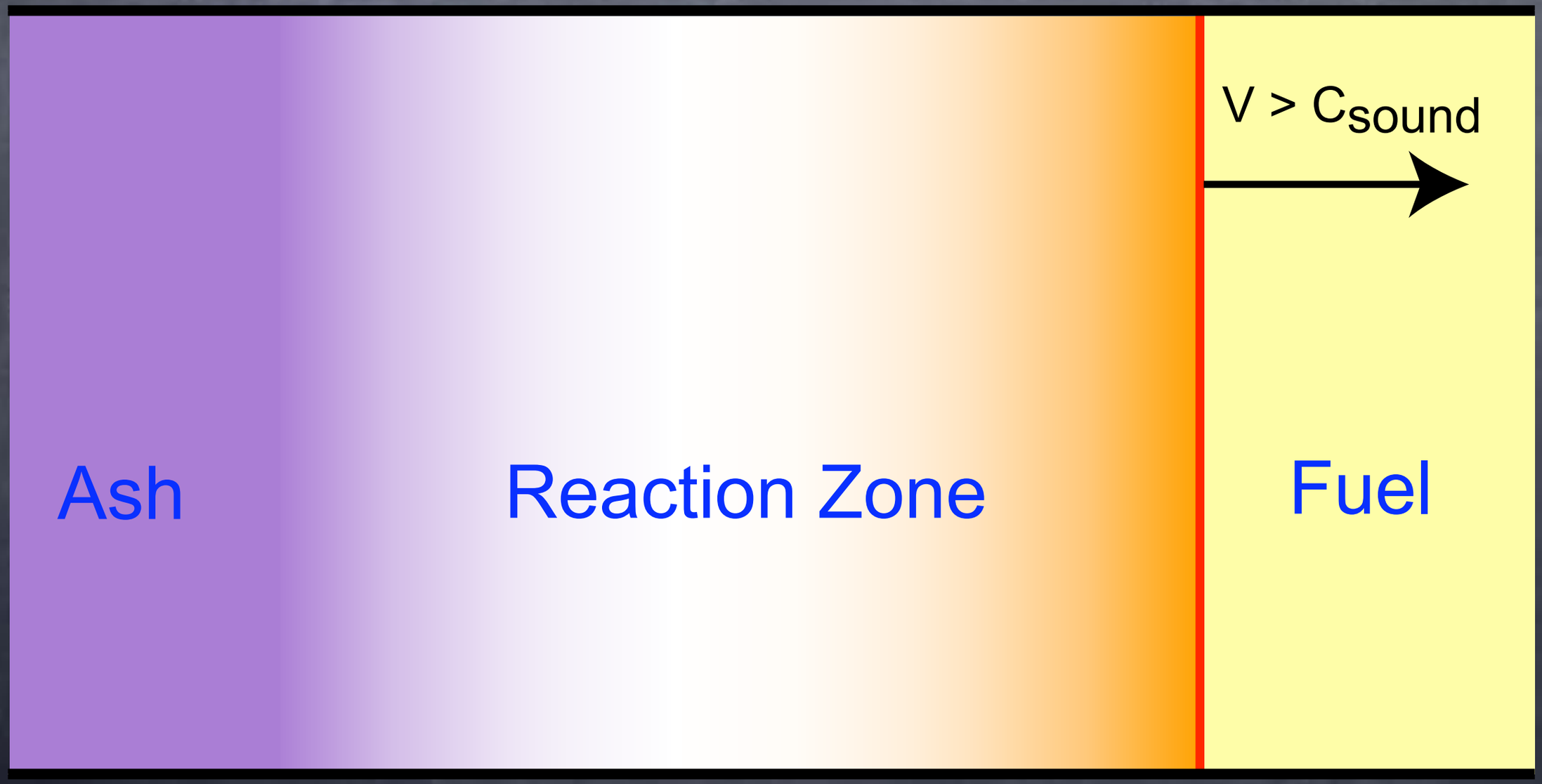
This self-heating mode can (and has) been used (in operator split form) for x-ray burst models

Fang Peng & Ed Brown 2003



# Thermodynamic trajectories

- One-dimensional detonations



# Thermodynamic trajectories

- Given the thermodynamics of the fuel, and that the ashes exist in an equilibrium state, the Chapman-Jouguet (1890) solution follows from a consistent solution to

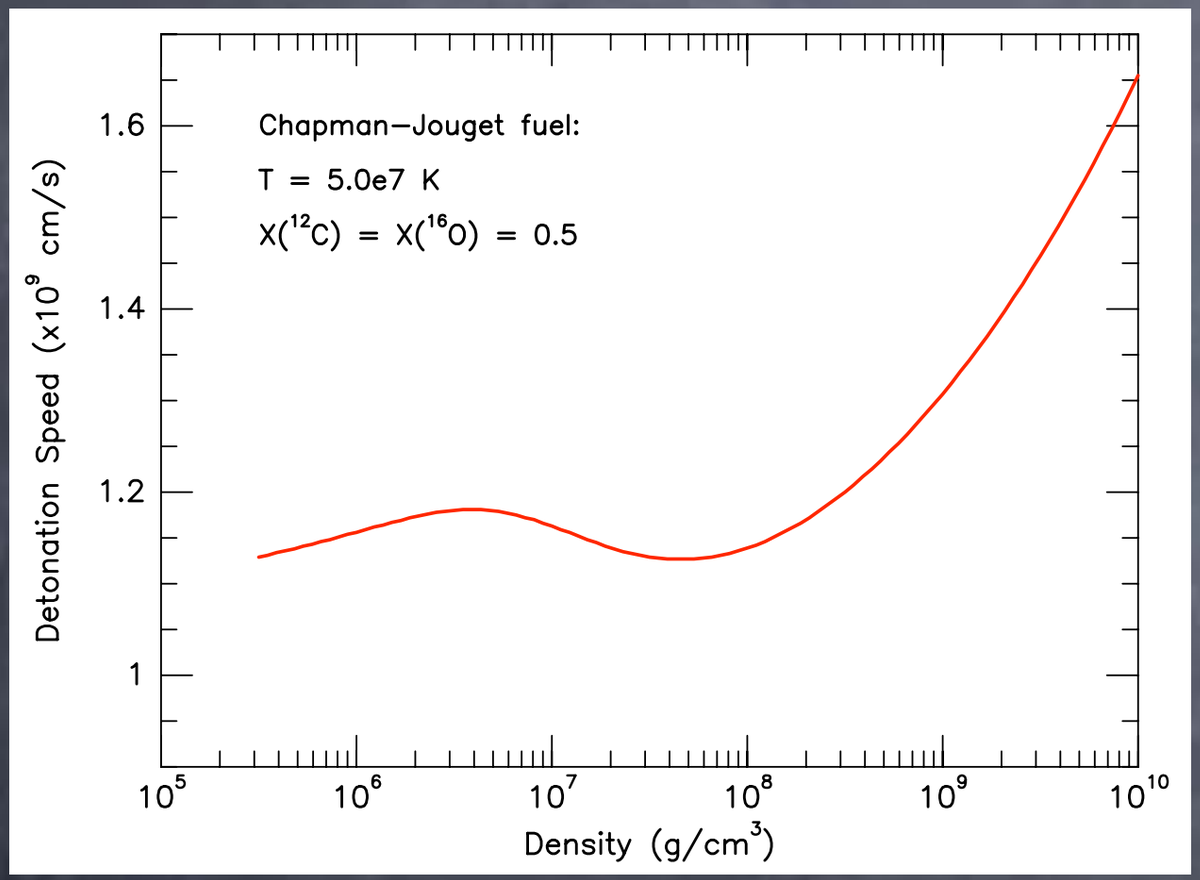
$$P_2 = P_1 - (v_2\rho_2)^2 \left(\frac{1}{\rho_1} - \frac{1}{\rho_2}\right)$$

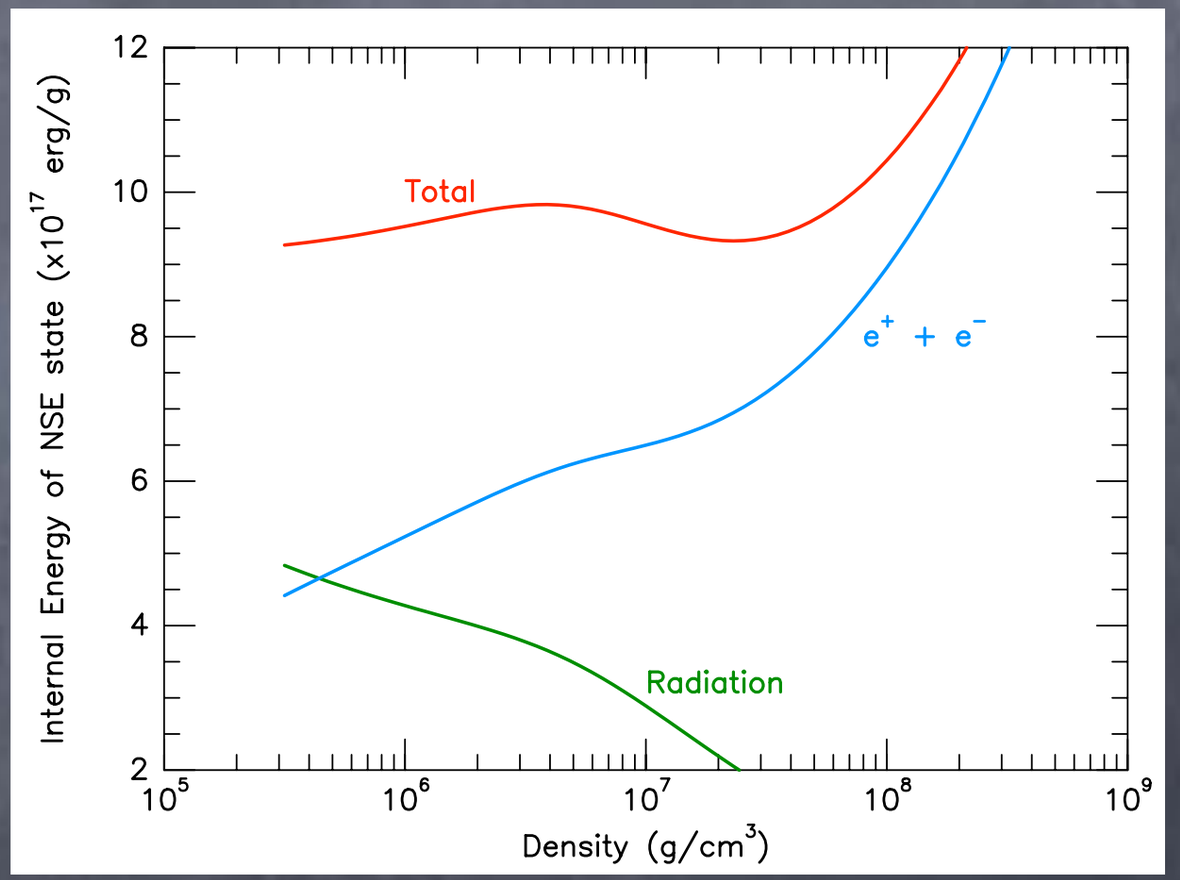
$$E_2 = \frac{1}{2}\left[(P_1 + P_2)\left(\frac{1}{\rho_1} - \frac{1}{\rho_2}\right)\right] + E_1 + q_{\rm burn}$$

$$\sum_{i=1}^{n} X_i = 1 \qquad \sum_{i=1}^{n} \frac{X_i\, Z_i}{A_i} = Y_{\rm e}$$

# Thermodynamic trajectories

- The CJ solution gives the detonation front speed and the ash thermodynamics.




- The CJ solution doesn't tell you about the width of the fuel-ash region, spatial variations of variables, or if the solution is a self-sustaining detonation.

# Thermodynamic trajectories

- The Zeldovich-Von Neumann-Doring (1943) solution follows from integrating three ODEs:

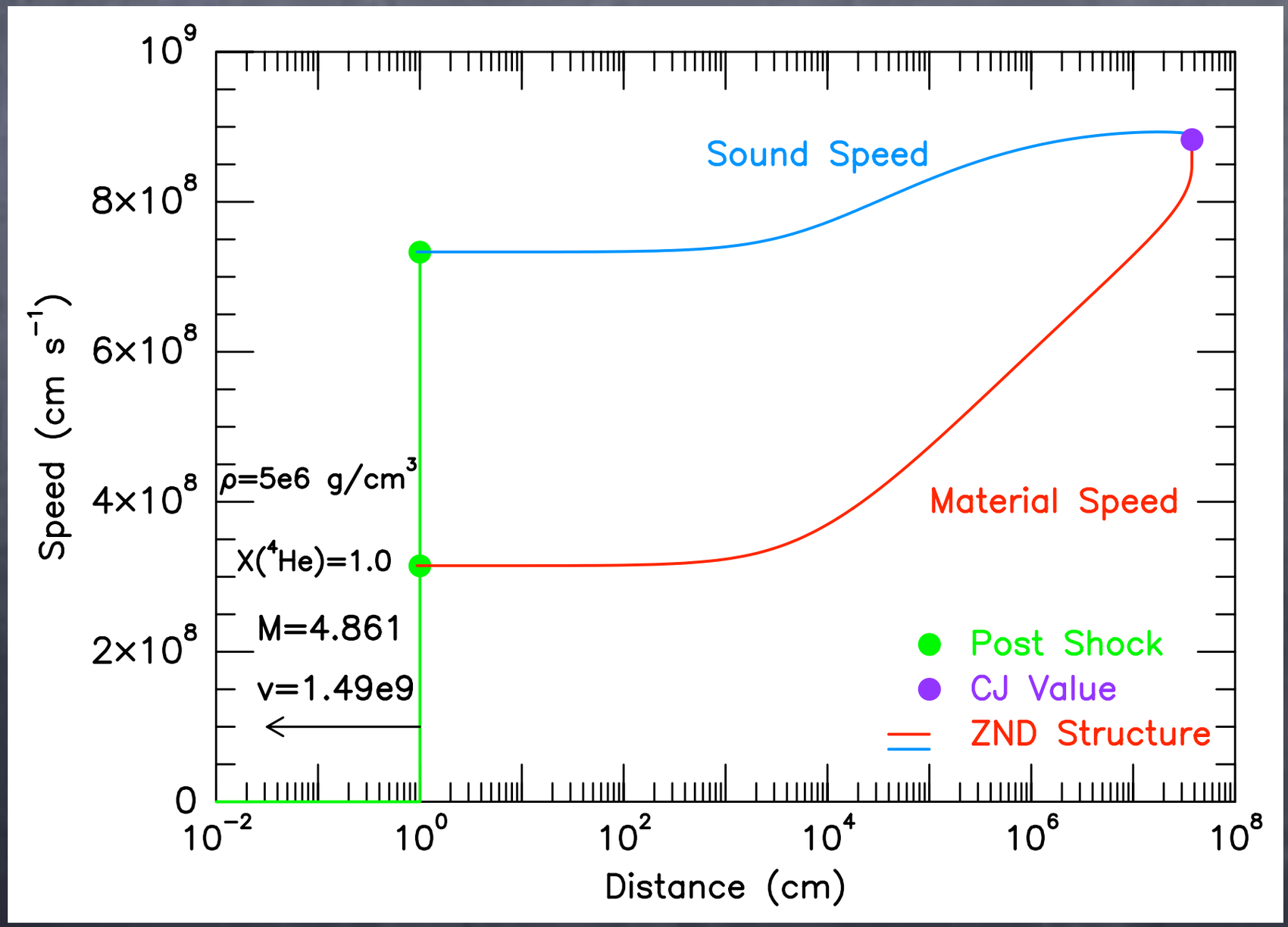
$$\frac{dP}{dx} = \frac{v\ \phi}{v^2 - c_s^2} \qquad \frac{d\rho}{dx} = \frac{1}{v}\ \frac{\phi}{v^2 - c_s^2} \qquad \frac{dv}{dx} = -\frac{1}{\rho}\ \frac{\phi}{v^2 - c_s^2}$$

$$\phi = \left.\frac{\partial P}{\partial E}\right|_\rho \left[\epsilon_{nuc} - \left.\frac{\partial E}{\partial A}\right|_P \frac{dA}{dt}\right]$$

- These can be added (with some degree of difficulty) to a reaction network.

# Thermodynamic trajectories

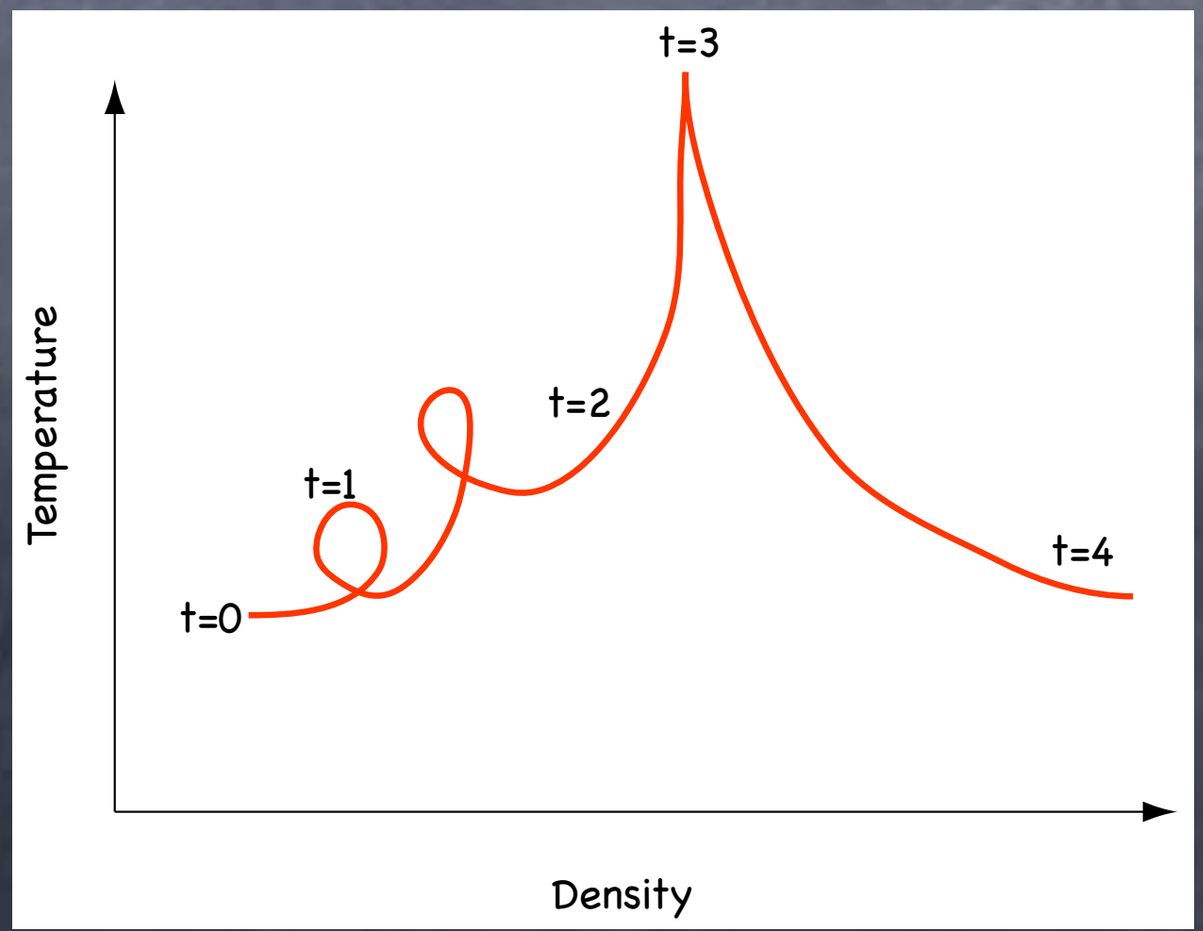
- The ZND solution gives the width of the fuel-ash region, spatial variations of the quantities, the self-sustaining solution, and global integrals which reduce to the CJ solution.



# Thermodynamic trajectories

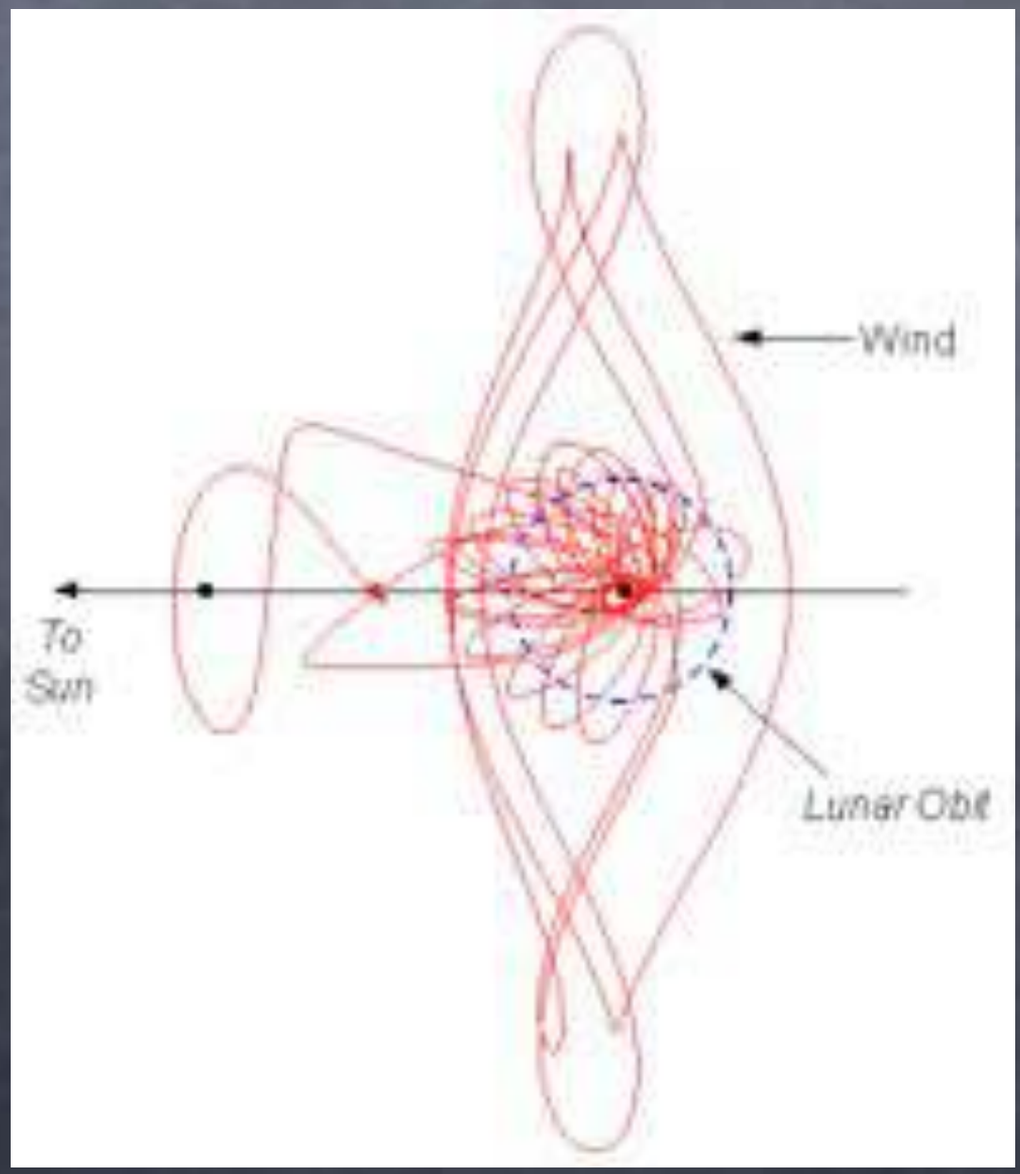
- One also encounters cases where the post-processing of a previously calculated thermodynamic trajectory is desired.

- In this case one interpolates T(t) and ρ(t) for time point demanded by the integration, and one uses the hydrostatic ODEs dT/dt=0 and dρ/dt=0.

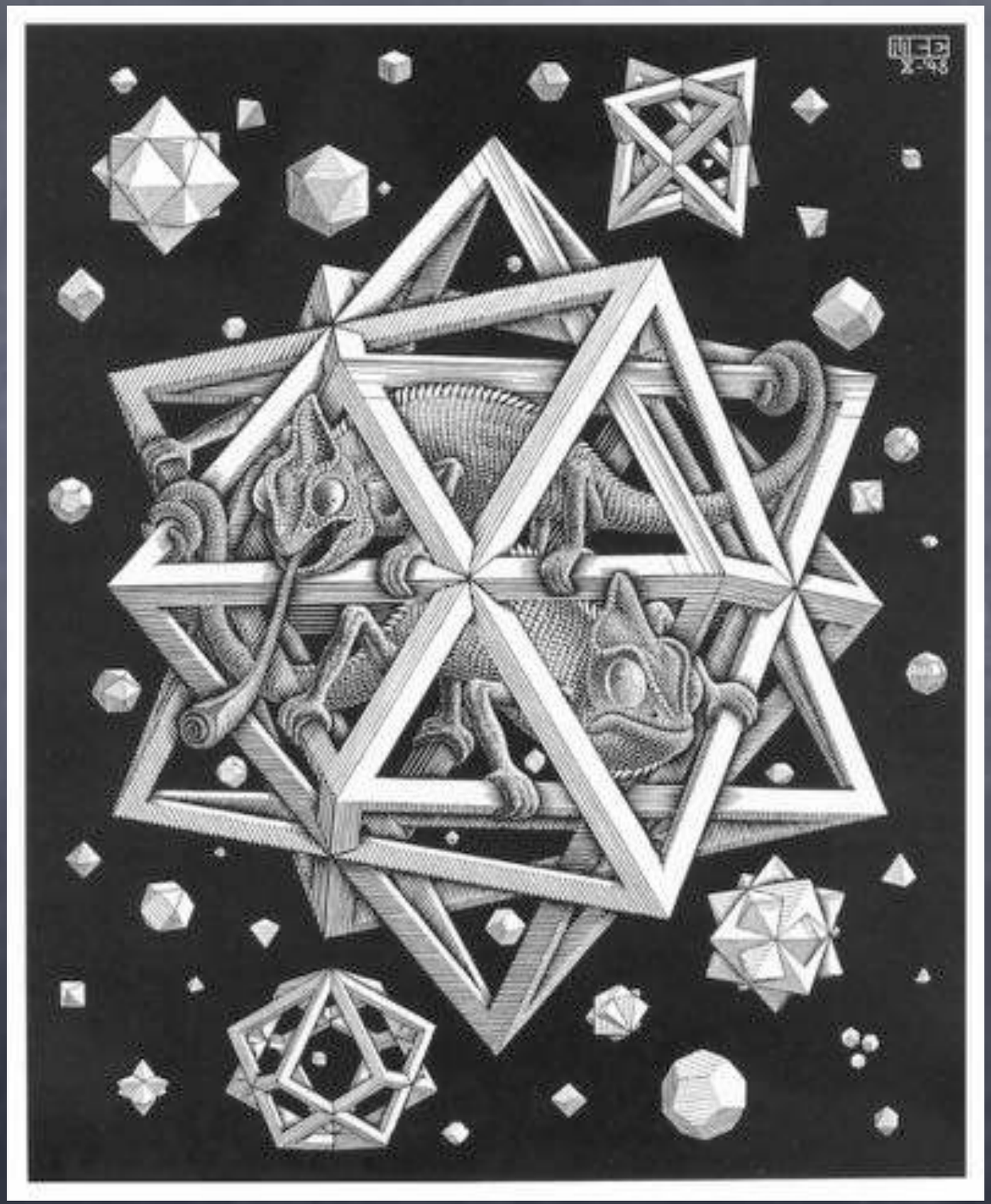


# Thermodynamic trajectories

- While not the most efficient, accurate integrations can be obtained for any arbitrary thermodynamic trajectory.



# Interlude



Stars
M.C. Escher
1948
Woodcut

# Alpha-chain networks

- Integration of the ordinary differential equations which represent the abundance levels of a set of isotopes serves two functions in models of stellar events.

- The primary function, as far as the hydrodynamics is concerned, is to provide the magnitude and sign of the thermonuclear energy generation rate.

- The second function is to describe the evolution of the abundance levels of the nuclear species. These abundance levels are, of course, fundamental to our understanding of the origin and evolution of the elements.

# Alpha-chain networks

- Obtaining accurate values for the energy generation rate is expensive in terms of computer memory and CPU time.

- The largest block of memory in a stellar hydrodynamic program is reserved for storing the abundances at every grid point.

- This memory requirement can be quite restrictive for 3-D models on present parallel computer architectures.

- Even with modern methods for solving reaction networks, evolving the abundances begins to dominate the total cost of a multi-D model when the number of species is about 30.

# Alpha-chain networks

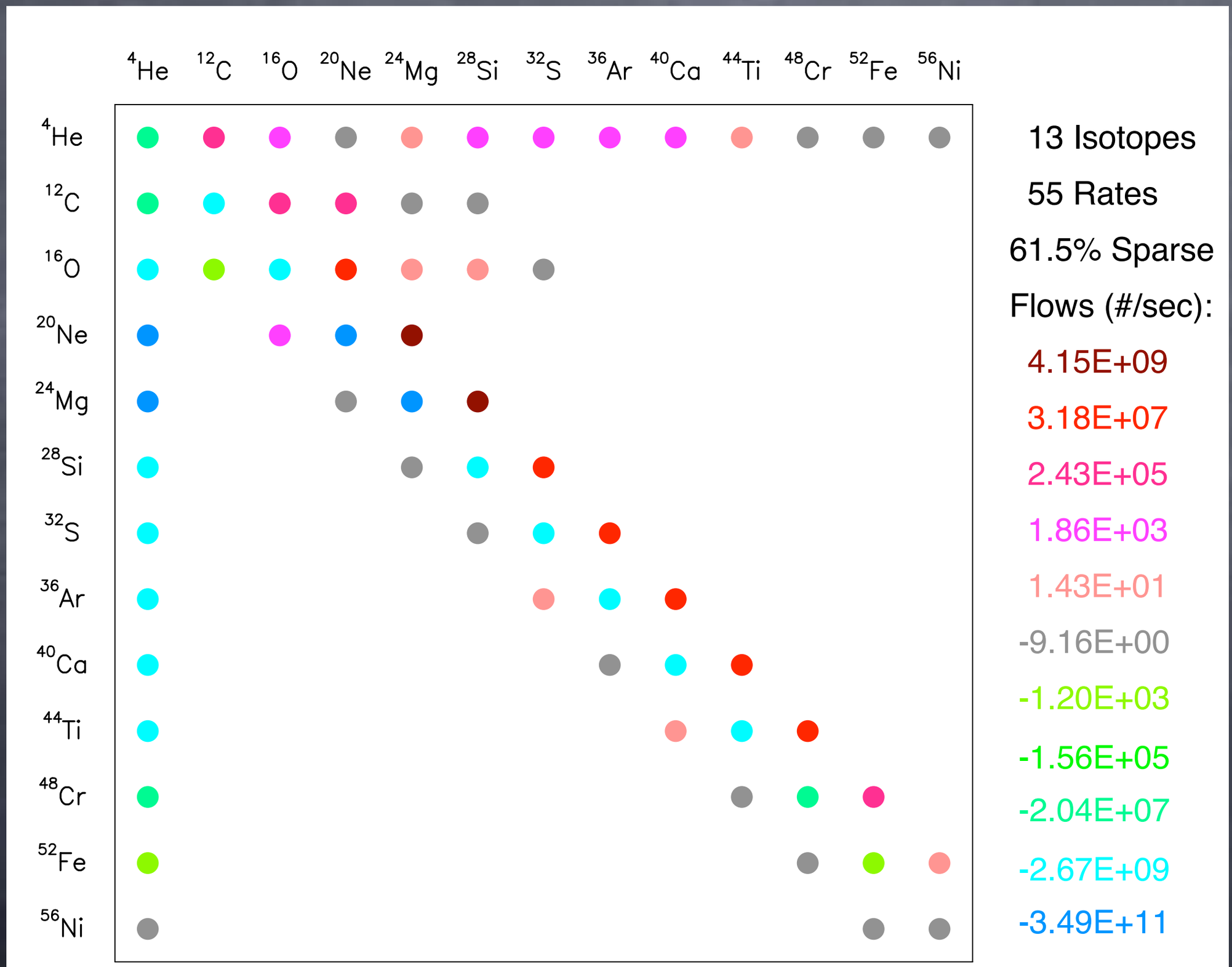
- To decrease the resources usage means making a choice between having fewer isotopes in the reaction network or having less spatial resolution.

- The general response to this tradeoff has been to evolve a limited number of isotopes, and thus calculate an approximate thermonuclear energy generation rate.

- The set of 13 nuclei most commonly used for this purpose are $^{4}He$, $^{12}C$, $^{16}O$, $^{20}Ne$, $^{24}Mg$, $^{28}Si$, $^{32}S$, $^{36}Ar$, $^{40}Ca$, $^{44}Ti$, $^{48}Cr$, $^{52}Fe$, $^{56}Ni$.

- This minimal set of nuclei, usually called an $\alpha$-chain network, can reasonably track the abundance levels from helium burning through nuclear statistical equilibrium.

# Alpha-chain networks

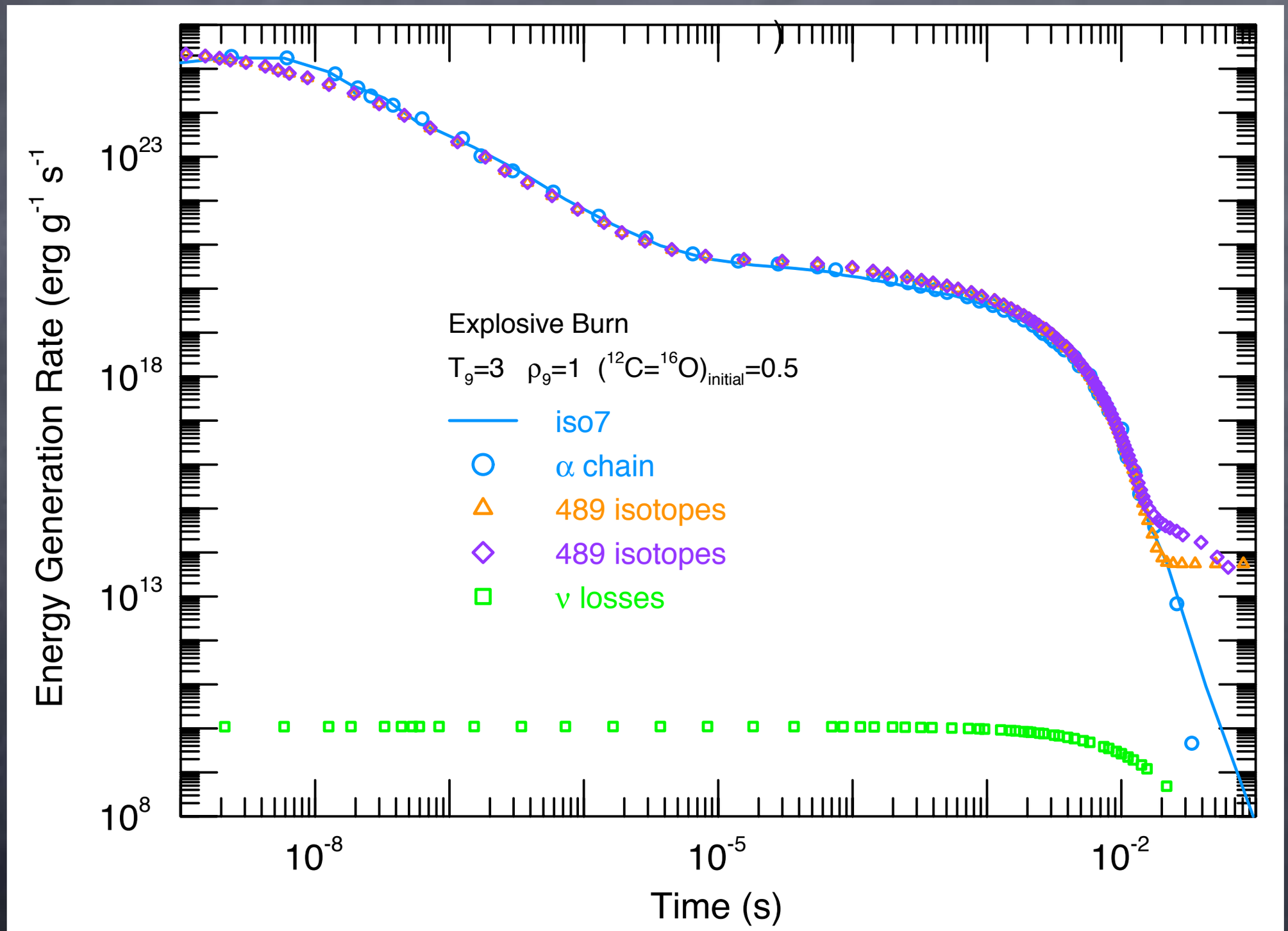


# Alpha-chain networks

- More importantly from a hydrodynamics standpoint, an α-chain reaction network gives a thermonuclear energy generation rate that is generally, but not always, within 20% of the energy generation rate given by larger reaction networks.

- In essence, one gets most of the energy generated for most thermodynamic conditions at a fraction of the computational cost (memory + CPU).

# Alpha-chain networks



# Alpha-chain networks

- A definition of what we mean by an α-chain reaction network is prudent.

- A strict α-chain reaction network is only composed of (α,γ) and (γ,α) links among the 13 isotopes $^{4}$He, $^{12}$C, $^{16}$O, $^{20}$Ne, $^{24}$Mg, $^{28}$Si, $^{32}$S, $^{36}$Ar, $^{40}$Ca, $^{44}$Ti, $^{48}$Cr, $^{52}$Fe, and $^{56}$Ni.

- It is essential, however, to include (α,p)(p,γ) and (γ,p)(p,α) links in order to obtain reasonably accurate energy generation rates and abundances when the temperature exceeds ~2.5 x $10^{9}$ K.

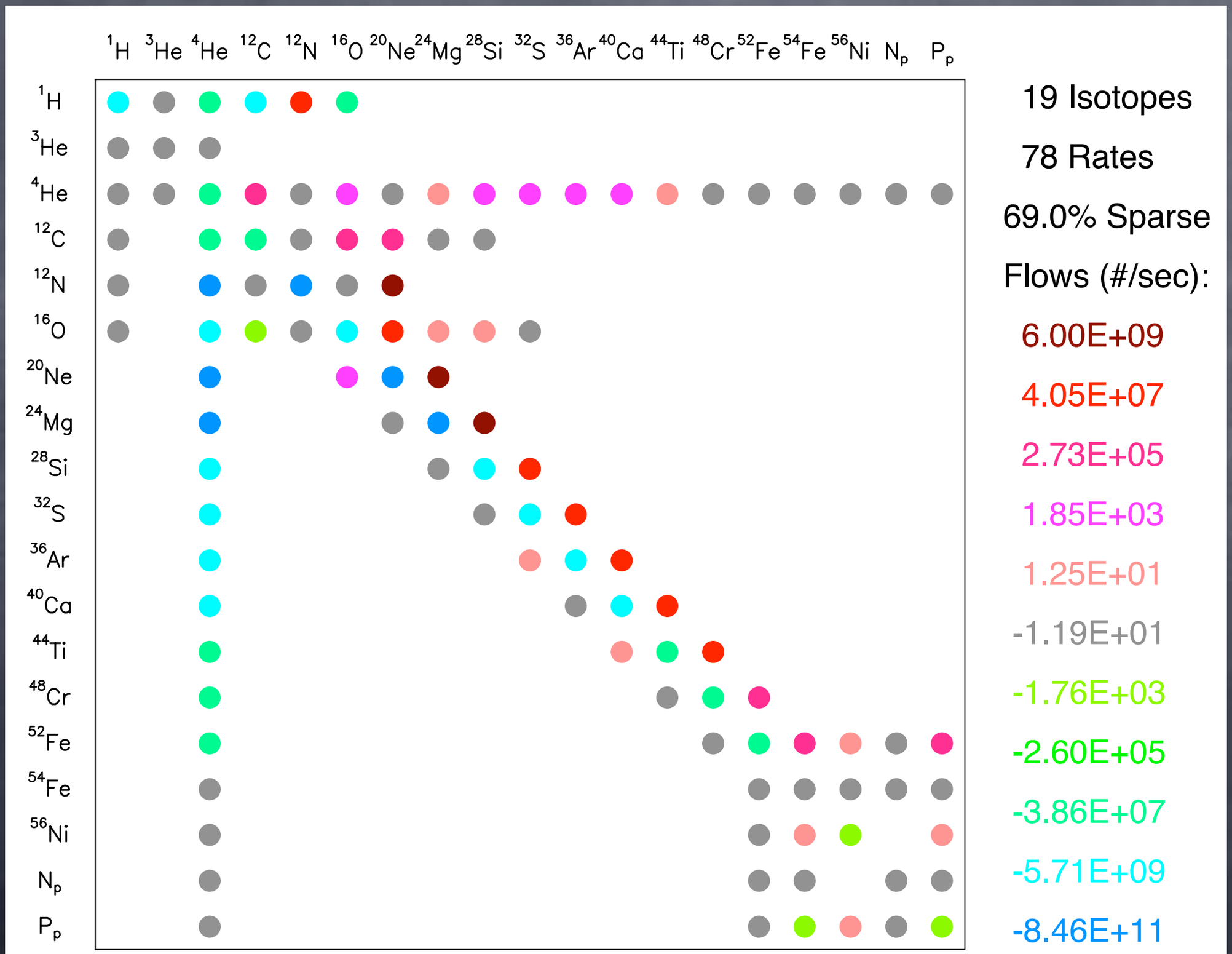
# Alpha-chain networks

- At these elevated temperatures the flows through the (α,p)(p,γ) sequences are faster than the flows through (α,γ) channels. An (α,p)(p,γ) sequence is, effectively, an (α,γ) reaction through an intermediate isotope.

- In the α-chain reaction network we'll use in these lectures, we include 8 (α,p)(p,γ) sequences plus the corresponding inverse sequences by assuming steady-state proton flows through the intermediate isotopes $^{27}$Al, $^{31}$P, $^{35}$Cl, $^{39}$K, $^{43}$Sc, $^{47}$V, $^{51}$Mn, and $^{55}$Co.

- This strategy permits inclusion of (α,p)(p,γ) sequences without evolving the proton or intermediate isotope abundances.

# Alpha-chain networks



# Tasks for the day

- Download, compile, and run the 13 isotope α-chain code from www.cococubed.com/code_pages/burn.shtml

- Verify the ODEs and the Jacobian matrix elements for the triple-alpha and the $^{12}C(\alpha,\gamma)^{16}O$ reactions.

- Run the code in hydrostatic mode for the initial conditions $T = 3 \times 10^9$ K, $\rho = 10^9$ g/cm$^3$, $X(^{12}C) = X(^{16}O) = 0.5$. Plot the abundance evolution. What isotope dominates when? Now run the code in its explosive (adiabatic) mode using the same initial conditions. Compare and contrast the results with the hydrostatic run.

# Tools and Toys in Nuclear Astrophysics