How we “look” inside stars:
stellar evolution codes &

Mathieu Renzo,
PhD student @ API, UvA

“Traditional scientific knowledge has generally taken the form of either theory or experimental data. However, where theory and experiment stumble, simulations may offer a third way.”
Simulation, Johannes Lenhard et al.
The most important thing

What is (Computational) “Stellar Astrophysics”? 

The **MESA** stellar evolution code

- Basic Assumptions
- Discretization
- Translation of the Physics for the Computer
- Example of input Physics: Nuclear Reaction Networks
- How the Computer Solves the Equations

What do I do with it?
The most important thing

What is (Computational) “Stellar Astrophysics”? 

The **MESA** stellar evolution code

- Basic Assumptions
- Discretization
- Translation of the Physics for the Computer
- Example of input Physics: Nuclear Reaction Networks
- How the Computer Solves the Equations

What do I do with it?
The most important thing

This is what should **not** happen

```
MY CODE WORKS
I HAVE NO IDEA WHY
```

```
MY CODE DOESN'T WORK
I HAVE NO IDEA WHY
```

**grep** is your friend! (see `man grep` on *nix)
The most important thing

What is (Computational) “Stellar Astrophysics”? 

The **MESA** stellar evolution code

- Basic Assumptions
- Discretization
- Translation of the Physics for the Computer
- Example of input Physics: Nuclear Reaction Networks
- How the Computer Solves the Equations

What do I do with it?
How can we “look” inside a star?

Figures Credits: NASA
How can we “look” inside a star?

We simply can’t!!

Other Q: How can we observe how one star evolves?
So what to do?

1. Build a theory from first principles;
2. Plug it in a computer;
3. Get out a model;
4. Find a smart way to compare it to what we can observe.

Advantages
- Full control over the parameters ⇒ Numerical Experiments;
- Allow to focus on interesting things (e.g. no reddening!);
- Allow to deal with long-lasting, rare, inaccessible phenomena;

Drawbacks
- Numerical errors;
- Limited computational resources;
- Nature ≫ Theory ≫ Model.

“All models are wrong, but some are useful” – G. Box
Outline

The most important thing

What is (Computational) “Stellar Astrophysics”?*

The **MESA** stellar evolution code

- Basic Assumptions
- Discretization
- Translation of the Physics for the Computer
- Example of input Physics: Nuclear Reaction Networks
- How the Computer Solves the Equations

What do I do with it?
The Stellar Evolution Code: **MESA**

is a *tool*, not a theory!

What does it stand for?

**Modules for Experiments in Stellar Astrophysics**

References:

- Paxton *et al.* 2013, ApJs208,4

mesa.sourceforge.net
mesastar.org

Open Source ⇔ Open Know How

“An algorithm must be seen to be believed” – D. Knuth

How to get MESA:

svn co -r 7624 svn://svn.code.sf.net/p/mesa/code/trunk mesa
## Modules overview

**MESA Module Definitions and Purposes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>alert</td>
<td>Utility</td>
<td>Error handling</td>
</tr>
<tr>
<td>atm</td>
<td>Microphysics</td>
<td>Gray and non-gray atmospheres; tables and integration</td>
</tr>
<tr>
<td>const</td>
<td>Utility</td>
<td>Numerical and physical constants</td>
</tr>
<tr>
<td>chem</td>
<td>Microphysics</td>
<td>Properties of elements and isotopes</td>
</tr>
<tr>
<td>diffusion</td>
<td>Macrophysics</td>
<td>Gravitational settling and chemical and thermal diffusion</td>
</tr>
<tr>
<td>eos</td>
<td>Microphysics</td>
<td>Equation of state</td>
</tr>
<tr>
<td>interp_1d</td>
<td>Numerics</td>
<td>One-dimensional interpolation routines</td>
</tr>
<tr>
<td>interp_2d</td>
<td>Numerics</td>
<td>Two-dimensional interpolation routines</td>
</tr>
<tr>
<td>ionization</td>
<td>Microphysics</td>
<td>Average ionic charges for diffusion</td>
</tr>
<tr>
<td>jina</td>
<td>Macrophysics</td>
<td>Large nuclear reaction nets using reaclib</td>
</tr>
<tr>
<td>kap</td>
<td>Microphysics</td>
<td>Opacities</td>
</tr>
<tr>
<td>karo</td>
<td>Microphysics</td>
<td>Alternative low-T opacities for C and N enhanced material</td>
</tr>
<tr>
<td>mlt</td>
<td>Macrophysics</td>
<td>Mixing length theory</td>
</tr>
<tr>
<td>mtx</td>
<td>Numerics</td>
<td>Linear algebra matrix solvers</td>
</tr>
<tr>
<td>net</td>
<td>Macrophysics</td>
<td>Small nuclear reaction nets optimized for performance</td>
</tr>
<tr>
<td>neu</td>
<td>Microphysics</td>
<td>Thermal neutrino rates</td>
</tr>
<tr>
<td>num</td>
<td>Numerics</td>
<td>Solvers for ordinary differential and differential-algebraic equations</td>
</tr>
<tr>
<td>package_template</td>
<td>Utility</td>
<td>Template for creating a new MESA module</td>
</tr>
<tr>
<td>rates</td>
<td>Microphysics</td>
<td>Nuclear reaction rates</td>
</tr>
<tr>
<td>screen</td>
<td>Microphysics</td>
<td>Nuclear reaction screening</td>
</tr>
<tr>
<td>star</td>
<td>Evolution</td>
<td>One-dimensional stellar evolution</td>
</tr>
<tr>
<td>utils</td>
<td>Utility</td>
<td>Miscellaneous utilities</td>
</tr>
<tr>
<td>weaklib</td>
<td>Microphysics</td>
<td>Rates for weak nuclear reactions</td>
</tr>
</tbody>
</table>
Numerical Methods: 1D (or 1.5D)

Prohibitive computational cost of 3D simulations

⇒ 1D, but stars are not spherical-symmetric!

Need of parametric approximations for:

- Rotation ⇒ “Shellular Approximation”;
- Magnetic Fields;
- Convection ⇒ Mixing Length Theory (MLT);
- (Some) mixing processes;
- ...

Beware of systematic errors!

(Recall: Nature ⇒ Model)
\[ \frac{dP}{dr} = - \frac{G m(r) \rho}{r^2} \]

... but stars are not necessarily static!

Other examples:
- He flash,
- Outburst and Eruptions,
- Impulsive mass loss,
- RLOF,
- ...

Figure: η Car, APOD.
For numerical solutions:

\[
\frac{df}{dx} \rightarrow \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k}
\]

⇒ Discretization of space (mesh or grid) and time (timesteps)

(Recall: Nature ≫ Model)
Spatial Discretization (Meshing)

- Intensive quantities (e.g. $T, \rho$) averaged by mass within each cell;
- Extensive quantities (e.g. $m, L$) calculated at outer cell boundary.

Figure: From Paxton et al. 2011, ApJs, 192, 3

Need to check that your physical results do not depend on the way you discretize space.
Numerical Methods: Timestep selection

\[ \Delta t_n: \text{Large enough, but } \lesssim \tau_{\text{KH}}, \tau_{\dot{M}}, \text{etc.} \]

Need to find the best \( \Delta t_n \) at each step – few \( \times 100 \lesssim \text{total } n \lesssim \text{few } \times 10^4 \)
Reformulation of the (1D–) Equations

Physical Theory:

\[ \frac{dP}{dr} = - \frac{G m(r) \rho}{r^2} \left( + \frac{F}{4\pi r^2} \right) \]

\[ \frac{dm}{dr} = 4\pi r^2 \rho \]

\[ \frac{dT}{dr} = - \frac{3}{16\pi ac} \frac{\kappa \rho L}{r^2 T^3} \]

\[ \frac{dL}{dr} = 4\pi r^2 \rho \varepsilon \]

\[ P \equiv P(\rho, \mu, T) \]

Numerical Implementation:

\[ \left. \frac{dX_i}{dt} \right|_r = \left[ \sum_j P_{j,i}(T, \rho) - \sum_k D_{i,k}(T, \text{rho}) \right] + \left[ \sigma_i \nabla^2 X_i \right] \]
Reformulation of the (1D–) Equations

Physical Theory:
\[
\frac{dP}{dr} = - \frac{Gm(r) \rho}{r^2} \left( + \frac{F}{4\pi r^2} \right)
\]
\[
\frac{dm}{dr} = 4\pi r^2 \rho
\]
\[
\frac{dT}{dr} = - \frac{3}{16\pi ac} \frac{\kappa \rho L}{r^2 T^3}
\]
\[
\frac{dL}{dr} = 4\pi r^2 \rho \varepsilon
\]

\[P \equiv P(\rho, \mu, T)\]

Numerical Implementation:
\[
\Rightarrow \quad \frac{P_{k-1} - P_k}{0.5(dm_{k-1} - dm_k)} = - \frac{Gm_k}{4\pi r_k^4} - \frac{a_k}{4\pi r_k^2}
\]
\[
\Rightarrow \quad \ln(r_k) = \frac{1}{3} \ln \left[ r_{k+1}^3 + \frac{3}{4\pi} \frac{dm_k}{\rho_k} \right]
\]
\[
\Rightarrow \quad \frac{T_{k-1} - T_k}{(dm_{k-1} - dm_k)/2} = - \nabla T, k \left( \frac{dP}{dm} \left|_{k} \right. \right) \frac{\tilde{T}_k}{\bar{P}_k}
\]
\[
\Rightarrow \quad L_k - L_{k+1} = dm_k \{ \varepsilon_{\text{nuc}} - \varepsilon_v + \varepsilon_{\text{grav}} \}
\]
\[
\Rightarrow \quad P \equiv P(\rho, \mu, T)
\]

\[
\left. \frac{dX_i}{dt} \right|_r = \left[ \sum_j P_{j,i}(T, \rho) - \sum_k D_{i,k}(T, \rho) \right] + \left[ \sigma_i \nabla^2 X_i \right]
\]
\[
\Updownarrow \quad X_{i,k}(t_n + \Delta t_{n+1}) = X_{i,k}(t_n) + \Delta t_{n+1} \left( \frac{dX_{i,k}}{dt} \right)_{\text{nuc}} + \frac{(X_{i,k} - X_{i,k-1}) \sigma_k \Delta t_{n+1}}{0.5(dm_{k-1} - dm_k)}
\]
Reformulation of the (1D–) Equations

Physical Theory:

\[
\frac{dP}{dr} = -\frac{Gm(r)\rho}{r^2} \left( + \frac{F}{4\pi r^2} \right)
\]

\[
\frac{dm}{dr} = 4\pi r^2 \rho
\]

\[
\frac{dT}{dr} = -\frac{3}{16\pi ac} \frac{\kappa \rho L}{r^2 T^3}
\]

\[
\frac{dL}{dr} = 4\pi r^2 \rho \epsilon
\]

\[P \equiv P(\rho, \mu, T)\]

Numerical Implementation:

\[
\frac{P_{k-1} - P_k}{0.5(dm_{k-1} - dm_k)} = - \frac{Gm_k}{4\pi r_k^4} - \frac{a_k}{4\pi r_k^2}
\]

\[
\ln(r_k) = \frac{1}{3} \ln \left[ r_{k+1}^3 + \frac{3}{4\pi} \frac{dm_k}{\rho_k} \right]
\]

\[
\frac{T_{k-1} - T_k}{(dm_{k-1} - dm_k)/2} = - \nabla T, k \left( \frac{dP}{dm} \bigg|_k \right) \frac{T_k}{P_k}
\]

\[L_k - L_{k+1} = dm_k \{ \epsilon_{\text{nuc}} - \epsilon_v + \epsilon_{\text{grav}} \} \]

\[
P \equiv P(\rho, \mu, T)
\]

\[
\left. \frac{dX_i}{dt} \right|_r = \left[ \sum_j P_{j,i}(T, \rho) - \sum_k D_{i,k}(T, \rho, \rho) \right] + \left[ \sigma_i \nabla^2 X_i \right]
\]

\[
X_{i,k}(t_n + \Delta t_{n+1}) = X_{i,k}(t_n) + \Delta t_{n+1} \left( \frac{dX_{i,k}}{dt} \right)_{\text{nuc}} + \frac{(X_{i,k} - X_{i,k-1})\sigma_k \Delta t_{n+1}}{0.5(dm_{k-1} - dm_k)}
\]
Reformulation of the (1D–) Equations

Physical Theory:

\[ \frac{dP}{dr} = -\frac{Gm(r)\rho}{r^2} \left( + \frac{F}{4\pi r^2} \right) \]

\[ \frac{dm}{dr} = 4\pi r^2 \rho \]

\[ \frac{dT}{dr} = -\frac{3}{16\pi ac} \frac{\kappa \rho L}{r^2 T^3} \]

\[ \frac{dL}{dr} = 4\pi r^2 \rho \varepsilon \]

\[ P \equiv P(\rho, \mu, T) \]

Numerical Implementation:

\[ \frac{P_{k-1} - P_k}{0.5(dm_{k-1} - dm_k)} = -\frac{Gm_k}{4\pi r_k^4} - \frac{a_k}{4\pi r_k^2} \]

\[ \ln(r_k) = \frac{1}{3} \ln \left[ r_{k+1}^3 + \frac{3}{4\pi} \frac{dm_k}{\rho_k} \right] \]

\[ \frac{T_{k-1} - T_k}{(dm_{k-1} - dm_k)/2} = -\nabla T, k \left( \frac{dP}{dm} \bigg|_{k} \right) \frac{\bar{T}_k}{P_k} \]

\[ L_k - L_{k+1} = dm_k \{ \varepsilon_{\text{nuc}} - \varepsilon_v + \varepsilon_{\text{grav}} \} \]

\[ P \equiv P(\rho, \mu, T) \]

\[ \left. \frac{dX_i}{dt} \right|_r = \left[ \sum_j P_{j,i}(T, \rho) - \sum_k D_{i,k}(T, \rho, \mu, T) \right] + \left[ \sigma_i \nabla^2 X_i \right] \]

\[ X_{i,k}(t_n + \Delta t_{n+1}) = X_{i,k}(t_n) + \Delta t_{n+1} \left( \frac{dX_{i,k}}{dt} \right)_{\text{nuc}} + \frac{(X_{i,k} - X_{i,k-1})\sigma_k \Delta t_{n+1}}{0.5(dm_{k-1} - dm_k)} \]
Reformulation of the (1D–) Equations

Physical Theory:

\[
\frac{dP}{dr} = -\frac{Gm(r)\rho}{r^2} \left( + \frac{F}{4\pi r^2} \right)
\]

\[
\frac{dm}{dr} = 4\pi r^2 \rho
\]

\[
\frac{dT}{dt} = -\frac{3}{16\pi ac} \frac{\kappa \rho L}{r^2 T^3}
\]

\[
\frac{dL}{dr} = 4\pi r^2 \rho \varepsilon
\]

\( P \equiv P(\rho, \mu, T) \)

Numerical Implementation:

\[
\Leftrightarrow \quad \frac{P_{k-1} - P_k}{0.5(dm_{k-1} - dm_k)} = -\frac{Gm_k}{4\pi r^4} - \frac{a_k}{4\pi r^2}
\]

\[
\Leftrightarrow \quad \ln(r_k) = \frac{1}{3} \ln \left[ r_{k+1}^3 + \frac{3}{4\pi} \frac{dm_k}{\rho_k} \right]
\]

\[
\Leftrightarrow \quad \frac{T_{k-1} - T_k}{(dm_{k-1} - dm_k)/2} = -\nabla T, k \left( \frac{dP}{dm} \right)_k \frac{\tilde{T}_k}{P_k}
\]

\[
\Leftrightarrow \quad L_k - L_{k+1} = dm_k \{ \varepsilon_{\text{nuc}} - \varepsilon_{\nu} + \varepsilon_{\text{grav}} \}
\]

\[
\Leftrightarrow \quad P \equiv P(\rho, \mu, T)
\]

\[
\frac{dX_i}{dt} \bigg|_r = \left[ \sum_j P_{j,i}(T, \rho) - \sum_k D_{i,k}(T, \rho, \cdot) \right] + \left[ \sigma_i \nabla^2 X_i \right]
\]

\[
\Leftrightarrow \quad X_{i,k}(t_n + \Delta t_{n+1}) = X_{i,k}(t_n) + \Delta t_{n+1} \left( \frac{dX_{i,k}}{dt} \right)_{\text{nuc}} + \frac{(X_{i,k} - X_{i,k-1})\sigma_k \Delta t_{n+1}}{0.5(dm_{k-1} - dm_k)}
\]
Numerical Methods: Nuclear Networks

What matters:

• Total Number of Isotopes $N_{iso}$;
• Which Isotopes;
• Number of Nuclear Reactions.

(Ex. of) tricks under the hood:

• Compound reactions, e.g. $3\alpha$:
  $\alpha + \alpha \rightarrow (^{8}\text{Be} + \alpha \rightarrow) ^{12}\text{C} + \gamma$;
• (Quasi Statistical Equilibrium Networks for advanced burning stages);

High impact on:

• Computational cost ($\propto N_{iso}^2$) ⇒ Run time;
• $\varepsilon_{\text{nuc}}$ ⇒ $L, T_c, \rho_c$, etc.;
• Free electrons ($Y_e$) ⇒ Final fate (BH, NS, WD, etc.)
Figure: From Paxton et al. 2013, ApJs, 208, 4. Black dots are non-zero entries.
Numerical Methods: Algorithm

- MESA solves simultaneously the fully coupled set for the structure and composition;
- Henyey code: varies all the quantities in each zone until an acceptable solution is found (≠ Shooting Method);
- Generalized Newton-Raphson solver (⇒ FIRST ORDER):

\[
0 = F(y) \approx F(y_i + \delta y_i) = F(y_i) + \left[ \frac{dF(y)}{dy} \right]_i \delta y_i + O((\delta y_i)^2);
\]

\[
\delta y_i \approx -\frac{F(y_i)}{\left[ \frac{dF(y)}{dy} \right]_i}
\]

\[
y_{i+1} = y_i + \delta y_i
\]
Numerical Methods: NR-Solver Iterations

Figure: Two models after the end of core hydrogen burning

$M = 9M_\odot, Z = Z_\odot,$
custom initial model,
TAMS

$T[10^7 K]$ vs. $M[M_\odot]$
Outline

The most important thing

What is (Computational) “Stellar Astrophysics”? 

The [MESA] stellar evolution code

• Basic Assumptions
• Discretization
• Translation of the Physics for the Computer
• Example of input Physics: Nuclear Reaction Networks
• How the Computer Solves the Equations

What do I do with it?
$M_{\text{ZAMS}} \gtrsim 8 - 10 \, M_\odot$

- Nucleosynthesis
- Chemical Evolution of Galaxies
- Effects on Star Formation
- Re-ionization Epoch
- Observations of Farthest Galaxies
- Catastrophic Events
(Semi-)Empirical parametric models. Uncertainties encapsulated in efficiency factor:

\[ \dot{M}(L, T_{\text{eff}}, Z, R, M, \ldots) \]

\[ \eta \dot{M}(L, T_{\text{eff}}, Z, R, M, \ldots) \]

\( \eta \) is a free parameter:

\( \eta \in [0, +\infty) \)

Figure: From Smith 2014, ARA&A, 52, 487S
My current problem:

- I Want to see small effects $\Rightarrow$ need high spatial resolution ($\iff$ also high temporal resolution);
- I want to see them right-before the SN-explosion $\Rightarrow$ need to deal with advanced burning stages $\Rightarrow$ Need Large Nuclear Reaction Network;

Typically:

# cells $N_Z \sim 10^5 - 10^6\Rightarrow N \sim N_Z \times N_{iso} \sim 10^8 \Rightarrow$

# isotopes $N_{iso} \geq 200$

64bit float $\sim$ 8 bytes

$N^2 \times (8 \text{ bytes}) \sim 10^{17} \text{ bytes} \sim 10^8 \text{ Gb} !!$
My current problem:

- I want to see small effects \(\Rightarrow\) need high spatial resolution (\(\Leftrightarrow\) also high temporal resolution);
- I want to see them right-before the SN-explosion \(\Rightarrow\) need to deal with advanced burning stages \(\Rightarrow\) Need Large Nuclear Reaction Network;

Typically:

\[
\begin{align*}
\# \text{ cells } N_Z &\sim 10^5 - 10^6 \Rightarrow \mathcal{N} \sim N_Z \times N_{iso} \sim 10^8 \Rightarrow \\
\# \text{ isotopes } N_{iso} &\geq 200 \\
64\text{bit float} &\sim 8\text{ bytes}
\end{align*}
\]

\[
\mathcal{N}^2 \times (8\text{ bytes}) \sim 10^{17}\text{ bytes} \sim 10^8\text{ Gb} !!
\]

How can solve it?

Lower \(N_Z\) &
My current problem:

- I Want to see small effects ⇒ need high spatial resolution (⇔ also high temporal resolution);
- I want to see them right-before the SN-explosion ⇒ need to deal with advanced burning stages ⇒ Need Large Nuclear Reaction Network;

Typically:

# cells $N_z \sim 10^5 - 10^6 \Rightarrow N \sim N_z \times N_{iso} \sim 10^8 \Rightarrow \frac{N^2 \times (8 \text{ bytes})}{8}$

# isotopes $N_{iso} \geq 200$

64bit float $\sim 8$ bytes

$10^{17}$ bytes $\sim 10^8$ Gb !!

How can solve it?

Lower $N_z$ &

Thank you for your attention!
To choose the next timestep $\Delta t_{n+1}$:

1. $v_c \leq v_t \sim 10^{-4}$, $v_c$ unweighted average over all cells of the relative variations of $\log_{10}(R)$, $\log_{10}(T)$, $\log_{10}(\rho)$:

$$\Delta t_{n+1} = \Delta t_n \times g \left( \frac{g(v_t/v_{c,n})g(v_t/v_{c,n-1})}{g(\Delta t_n/\Delta t_{n-1})} \right)^{1/4}$$

$$g(x) \overset{\text{def}}{=} 1 + 2 \tan^{-1}(0.5(x-1)) \; ;$$

2. extra controls on relative variations of many quantities ($X_{i,k}$, $\varepsilon_{\text{nuc},k}$, $L_k$, $T_{\text{eff}}$, etc.);

It is always possible that you need to reduce $\Delta t_n$

If MESA fails: first retry then backup.