

Como utilizar o MESA para calcular sequências

Ana Antonini

Outubro 2025
FIP10101- Evolução Estelar

Resumo

1. O que é o MESA
2. Instalando o MESA
3. Rodando uma Sequência
4. Ferramentas Para Lidar com Output
 - 4.1. mesa_reader -> para trabalhar quantidades de saída com Python
 - 4.2 [MESAexplorer](#) -> para gráficos

● O que é o MESA?

MESA é um código livre modulado e escrito em Fortran 95 para calcular modelos unidimensionais para diversos processos e objetos de interesse astrofísico

● Como são construídos os modelos?

Para construir um modelo, o código divide a estrela em camadas e resolve as equações de estrutura e transferência de energia para cada uma delas, além de equações adicionais que descrevem outros aspectos da estrela (ex: taxa de perda de massa, das reações, difusão, opacidade). Eq são resolvidas simultaneamente com Newton-Raphson

Modules for Experiments in Stellar Astrophysics (MESA)

Bill Paxton and Lars Bildsten

Kavli Institute for Theoretical Physics and Department of Physics, Kohn Hall, University of California, Santa Barbara, CA 93106 USA

Aaron Dotter¹ and Falk Herwig

Department of Physics and Astronomy, University of Victoria, PO Box 3055, STN CSC, Victoria, BC, V8W 3P6 Canada

Pierre Lesaffre

LERMA-LRA, CNRS UMR8112, Observatoire de Paris and Ecole Normale Supérieure, 24 Rue Lhomond, 75231 Paris cedex 05, France

Frank Timmes

School of Earth and Space Exploration, Arizona State University, PO Box 871404, Tempe, AZ, 85287-1404 USA

ABSTRACT

Stellar physics and evolution calculations enable a broad range of research in astrophysics. Modules for Experiments in Stellar Astrophysics (MESA) is a suite of open source, robust, efficient, thread-safe libraries for a wide range of applications in computational stellar astrophysics. A 1-D stellar evolution module, *MESA star*, combines many of the numerical and physics modules for simulations of a wide range of stellar evolution scenarios ranging from very-low mass to massive stars, including advanced evolutionary phases. *MESA star* solves the fully coupled structure and composition equations simultaneously. It uses adaptive mesh refinement and sophisticated timestep controls, and supports shared memory parallelism based on OpenMP. State-of-the-art modules provide equation of state, opacity, nuclear reaction rates, element diffusion data, and atmosphere boundary conditions. Each module is constructed as a separate Fortran 95 library with its own explicitly defined public interface to facilitate independent development. Several detailed examples indicate the extensive verification and

• Como se Escrevem as 4 Equações?

Conservação de Massa

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

Equilíbrio Hidrostático

$$P_{k-1} - P_k = \overline{dm}_k \left[-\frac{Gm_k}{4\pi r_k^4} - \frac{a_k}{4\pi r_k^2} \right]$$

Transporte de Energia

$$T_{k-1} - T_k = \overline{dm}_k \left[\nabla_{T,k} \left(\frac{dP}{dm} \right)_{\text{hydrostatic}} \frac{\overline{T}_k}{\overline{P}_k} \right]$$

Conservação de Energia

$$L_k - L_{k+1} = dm_k (\varepsilon_{nuc} - \varepsilon_{v,thermal} + \varepsilon_{grav})$$

variáveis podem ser definidas em células ou interfaces

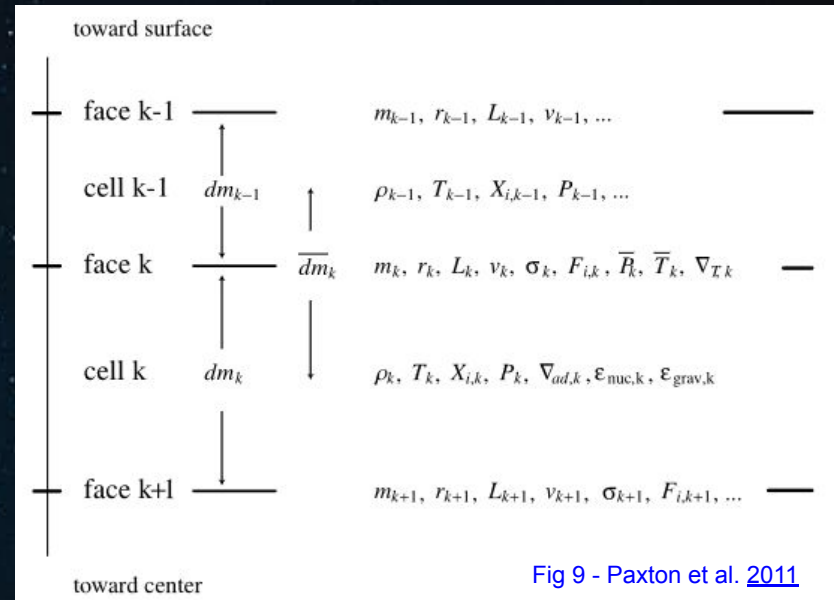


Fig 9 - Paxton et al. 2011

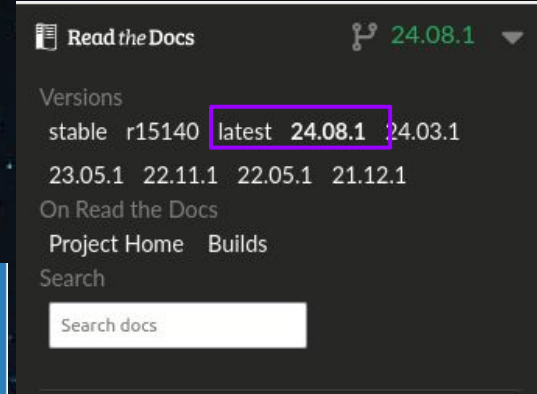
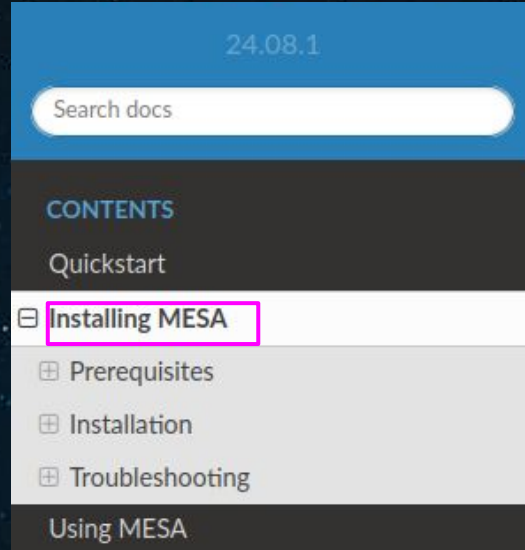
Instalando o MESA

1- Entrar no site do MESA → <https://docs.mesastar.org/>

2- Conferir que está na versão r24.08.1 →

3- Ir para “Installing MESA”

4- Conferir pré requisitos



Instalando o MESA

5- ir para página de download do [MESA SDK](#)

6- instalar pré requisitos do MESA SDK para seu SO

para Ubuntu



```
* sudo apt-get update  
* sudo apt dist-upgrade
```

```
sudo apt install binutils make perl libx11-dev tcsh zlib1g-dev
```

7- fazer download do SDK compatível com a versão do MESA utilizada

Instalando o MESA

8- Fazer download do MESA → [MESA r24.08.1](#)

9- Deszipar downloads na pasta de preferência

10- editar shell script

```
export MESASDK_ROOT=/home/antonini/Documents/MESA/mesasdk
source $MESASDK_ROOT/bin/mesasdk_init.sh
export MESA_DIR=/home/antonini/Documents/MESA/mesa-r24.03.1
PATH=$PATH:$MESA_DIR/scripts/shmesa

export OMP_NUM_THREADS=8
```

Para bash :

editar ~/.bashrc adicionando
as linhas ao lado ao arquivo

```
export MESASDK_ROOT=/home/anak/Documents/MESA/mesasdk
source $MESASDK_ROOT/bin/mesasdk_init.sh
export MESA_DIR=/home/anak/Documents/MESA/mesa-r24.08.1
PATH=$PATH:$MESA_DIR/scripts/shmesa

export OMP_NUM_THREADS=8
```

adequar ao seu computador

alterar a caixa em rosa para o
caminho até o diretório onde foi
feito o download do MESA e sdk

Instalando o MESA

11- entrar na pasta do MESA pelo terminal e dar `./install`

```
anak@stargayzer:~$ gedit .bashrc
anak@stargayzer:~$ cd $MESA_DIR
anak@stargayzer:~/Documents/MESA/mesa-r22.11.1$ ./install
```

após alguns minutos você deve receber a seguinte mensagem:

```
*****  
*****  
*****  
  
MESA installation was successful  
  
*****  
*****  
*****
```

12- Testar instalação fazendo uma cópia de `$MESA_DIR/star/work` para um ambiente **FORA** do diretório do MESA

```
anak@stargayzer:~/Documents/MESA$ cp -r $MESA_DIR/star/work teste
anak@stargayzer:~/Documents/MESA$ ls
mesa-r22.11.1  mesa-r22.11.1.zip  mesasdk  mesasdk-x86_64-linux-22.6.1.tar.gz  teste
anak@stargayzer:~/Documents/MESA$
```


Rodando uma Sequência de 1Msol - baixa SP a CO WD

1- copiar o diretório \$MESA_DIR/star/test_suite/1M_pre_ms_to_wd

```
anak@stargayzer:~$ cp -r $MESA_DIR/star/test_suite/1M_pre_ms_to_wd/ Documents/MESA/1M
anak@stargayzer:~$ ls
```

2- limpar e recompilar cópia (./clean , ./mk)

3- editar inlists

4- rodar o código (./rn)

[vídeo explicando como mexer editar pgstar se quiserem personalizar os gráficos/plotarem outras quantidades](#)

```

anak@stargayzer:~/Documents/MESA/1M$ ls
ck
clean
docs
history
columns.list
inlist
start
inlist_start
header
inlist_to_end
agb
inlist_to_end_agb
header
inlist_to_end_core_h_burn
inlist_to_end_core_h_burn_header
inlist_to_end_core_he_burn
inlist_to_end_core_he_burn_header
inlist_to_start
he_core_flash
inlist_to_start_he_core_flash_header
inlist_to_wd
inlist_to_wd_header
inlist_xtra_coeff_os
make
mk
profile_columns.list
re
README.rst
rn
rn1
rn_standard
src
standard_start_he_core_flash.mod

```

Rodando uma Sequência de 1Msol - editando inlists

Abrir cada um dos inlists e:

- remover o “!” na frente de “pgstar = .true.”
- acrescentar “!” na frente da linha “max_model_number”
- trocar “history_interval” para = 1
- adicionar a &controls a linha “max_years_for_timestep= x” em cada inlist

isso limita o passo de tempo máximo a x anos, aumentando a “resolução” da etapa.
Quanto maior o timestep menor a resolução temporal e menor o número de modelos totais calculados na etapa (“mais rápido” o cálculo)

limites recomendados:

Na SP $\rightarrow x \sim 5d6$ (pode ser aumentado um pouco mais, mas não passar de 1d8)

RGB, CHeB, AGB $\rightarrow x \sim 1d6$

WD CT $\rightarrow \sim 1d7$

Rodando uma Sequência de 1Msol - editando inlists

Abrir inlist_to_end_agb, ir para seção &controls:

- trocar o valor de “`blocker_scaling_factor`” para 0.5d0
isso reduzirá o fator de escala da equação de blocker (para perda de massa)
- acrescentar linha “`max_wind = 1d-4`”
impõem que o limite máximo para taxa de perda de massa seja 10^{-4} Msol/ano

Essas duas mudanças são para diminuir a perda de massa e permitir que a estrela experimente pulsos térmicos

caso a sequência fique presa nessa etapa, **aumentar novamente o fator de blocker**

Rodando uma Sequência de 1Msol - editando inlists

Optional:

abrir inlist_to_wd, ir a &controls e:

trocar valor de 'log_L_lower_limit' para -2

acrescentar as linhas:

```
do_element_diffusion = .true.
```

```
diffusion_use_full_net = .true.
```

isso esfriará a anã branca um pouco mais e ativará a difusão de elementos, produzindo um perfil de abundância final melhor

Rodando uma Sequência de 3Msol - alta SP a CO WD

1- copiar o diretório criado para 1M solar

```
antonini@chandra: ~/Documents/MESA/AULA_MESA$ shmesa cp 1M 3M
```

2- trocar a massa inicial de cada um dos inlists para 3.0 Msol e descomentar “pgstar = .true.”

3- abrir inlist_start,

em &star: trocar required_termination_code_string = 'Lnuc_div_L_zams_limit'

em &controls: apagar a linha “ log_L_lower_limit = 1” ,

acrescentar em seu lugar as linhas:

stop_near_zams = .true.

Lnuc_div_L_zams_limit = 0.9d0

Rodando uma Sequência de 3Msol - editando inlists

4- abrir inlist to start he core flash

em &stars : trocar “required termination code string” para ‘xa_central_lower_limit’

em &controls: apagar linha “power_he_burn_upper_limit = 10d0”

acrescentar linhas :

xa_central_lower_limit_species(1) = 'he4'

xa_central_lower_limit(1) = 0.9d0 →podem brincar com esse valor, 0.92,0.95, 0.97

5- editar inlist to end agb:

blocker_scaling_factor = 0.5

caso a sequência fique presa
nessa etapa, **aumentar o fator
de blocker até 1.0**

DETALHE: uma estrela com 3Msol, $Z=0.02$ e o esquema de convecção adotado **não** experimenta flash de He. O nome “inlist_to_start_he_core_flash” não foi trocado para não adicionar passos a mais a esse tutorial

Rodando uma Sequência de 10Msol - ONe WD

1- copiar o diretório \$MESA_DIR/star/test_suite/make_o_ne_wd

```
anak@stargayzer:~/Documents/MESA$ cp -r $MESA_DIR/star/test_suite/make_o_ne_wd 10M
anak@stargayzer:~/Documents/MESA$
```

2- limpar e recompilar cópia (./clean , ./mk)

3- editar inlists

4- rodar o código (./rn)

[vídeo explicando como mexer editar pgstar se quiserem personalizar os gráficos/plotarem outras quantidades](#)

```
anak@stargayzer:~/Documents/MESA/10M$ ls
ck                               inlist_remove_envelope         re
clean                           inlist_remove_envelope_header  README.rst
docs                             inlist_settle_envelope         rn
history_columns.list            inlist_settle_envelope_header  rn1
inlist_c_burn                   inlist_to_agb                  rn_all
inlist_c_burn_header            inlist_to_agb_header           src
inlist_common                   inlist_zams                     standard_agb.mod
inlist_o_ne_wd                  inlist_zams_header             standard_c_burn.mod
inlist_o_ne_wd_header           make                             standard_zams.mod
inlist_pgstar                   mk
```

Rodando uma Sequência de 10Msol - editando inlists

Abrir cada um dos inlists e:

- remover o “!” na frente de “pgstar = .true.”
isso fará com que o código produza gráficos enquanto vocês rodam a sequência
- acrescentar “!” na frente da linha “max_model_number”
- adicionar a &controls a linha “max_years_for_timestep= x” em cada inlist

isso limita o passo de tempo máximo a x anos, aumentando a “resolução” da etapa.
Quanto maior o timestep menor a resolução temporal e menor o número de modelos totais calculados na etapa (“mais rápido” o cálculo)

limites recomendados:

Na SP, RGB, CHeB $\rightarrow x \sim 1d5$ (pode ser aumentado um pouco mais, mas não passar de 1d6)

AGB $\rightarrow x \sim 1d3$

WD CT $\rightarrow \sim 1d6$

Rodando uma Sequência de 10Msol - editando inlists

Abrir `inlist_common`, ir para o fim da seção &controls:

- trocar `history_interval = 10` para `history_interval = 1`

isso fará com que o código salve informações no arquivo `history.data` a cada modelo calculado

Abrir `inlist_cburn`, ir para seção &controls:

- trocar o valor de “`blocker_scaling_factor`” para `0.5d0`

isso reduzirá o fator de escala da equação de blocker (para perda de massa)

- trocar o valor na linha “`max_wind =`” para `1d-6`

impõem que o limite máximo para taxa de perda de massa seja 10^{-6} Msol/ano

Rodando uma Sequência de 10Msol - editando inlists

Abrir `inlist_o_ne_wd`, ir para seção &controls:

- trocar o valor de “`blocker_scaling_factor`” para `10.0d0`
isso reduzirá o fator de escala da equação de bloquer (para perda de massa)
- acrescentar linha “`max_wind = 1d-2`”
impõem que o limite máximo para taxa de perda de massa seja 10^{-2} Msol/ano

Essas duas mudanças são para diminuir a perda de massa e permitir que a estrela experimente alguns pulsos térmicos

caso a sequência fique presa nessa etapa, **aumentar novamente o fator de blocker e mudar `max_wind` para `=1`**

Rodando uma Sequência de 10Msol - editando inlists

Opcional:

abrir `inlist_settle_envelope`, ir a `&controls` e:

trocar valor de `'log_L_lower_limit'` para -2

retirar `“!”` da linha `“diffusion_use_full_net = .true.”`

isso esfriará a anã branca um pouco mais e ativará a difusão de elementos para todas as espécies da rede

Rodando uma Sequência de 12Msol - SNCC

1- copiar o diretório \$MESA_DIR/star/test_suite/make_pre_ccsn_13bcvn

```
antonini@chandra:~/Documents/MESA/AULA_MESA$ cp -r $MESA_DIR/star/test_suite/make_pre_ccsn_13bcvn 12M_CCSN
```

2- limpar e recompilar cópia (./clean , ./mk)

3- editar inlists

4- rodar o código (./rn)

[vídeo explicando como mexer editar pgstar se quiserem personalizar os gráficos/plótarem outras quantidades](#)

```
antonini@chandra:~/Documents/MESA/AULA_MESA/12M_CCSN$ ls
ck                               inlist_remove_header      re
clean                           inlist_to_cc              README.rst
history_columns.list            inlist_to_cc_header       rn
inlist_after_remove             inlist_to_post_si_burn    rn1
inlist_after_remove_header      inlist_to_post_si_burn_header
inlist_before_remove            inlist_to_zams             src
inlist_before_remove_header    inlist_to_zams_header     standard_after_si_burn.mod
inlist_massive_defaults        make                       standard_near_zams.mod
inlist_pgstar                  mk                         standard_ready.mod
inlist_remove                   profile_columns.list       standard_ready_to_remove.mod
                                standard_removed.mod
```


Rodando uma Sequência de 12Msol - editando inlists

Abrir cada um dos inlists e:

- em `&star` acrescentar “`save_pgstar_files_when_terminate = .true.`”
isso fará com que o código salve gráficos ao final de cada etapa

Abrir `inlist_massive_defaults`, ir para o fim da seção `&controls`:

- trocar `mesh_delta_coeff = 2` para `mesh_delta_coeff = 1`
isso passará a resolução espacial para o valor default do código

Tipos de arquivos gerados pelo MESA

history file (history.data)

- salva quantidades globais ao longo da evolução. Cada linha corresponde a um momento da evolução do modelo

profile file (profileX.data)

- salva quantidades locais em um momento da evolução. Cada linha corresponde a uma camada do modelo

photo file (X_____)

- espécie de “check-point”, o código salva informações necessárias para continuar rodando de pontos ao longo da evolução, caso seja interrompido. Não é compatível entre versões.

mod file (X.mod)

- salva informações “finais” que o código pode usar para começar novo run. É compatível entre versões

Ferramentas Para Lidar com Output

mesa_reader

- interpreta arquivos de saída do MESA para serem trabalhados com python
- para usar o mesa_reader basta fazer download do arquivo e chamá-lo com python. Importar como qualquer outro pacote python depois disso.

MESA explorer

- ferramenta online para visualizar comportamento de quantidades de saída do MESA
- para utilizá-lo basta fazer upload dos arquivos de interesse (history e profiles) e selecionar quais variáveis gostaria de plotar

Problemas Possíveis e como Resolvê-los

- MESA interrompe com mensagem “killed”

Em geral acontece por falta de RAM. Acrescentar a linha: `num_steps_for_garbage_collection = 500 !`

na seção “&star_job” dos inlists. Se continuar ocorrendo, diminuir ainda mais

- Mesa termina um dos inlists sem passar para o seguinte

A maneira mais fácil de lidar com esse problema será trocar o manualmente o nome do inlist lido pelo arquivo “inlist”

```
1 |
2 &star_job
3
4
5     read_extra_star_job_inlist(2) = .true.
6     extra_star_job_inlist_name(2) = 'inlist_to_start_he_core_flash'
```

isso deve ser feito para cada uma das seções &__