

An Introduction to Modules for Experiments in Stellar Astrophysics



Meridith Joyce
Assistant Professor

Physical Science 109

(307) 766-6150

mjoyce8@uwyo.edu

A/Professor Meridith Joyce
Department of Physics and Astronomy
& School of Computing
University of Wyoming, USA
Marie Curie, Konkoly Observatory: '22–'24
MESA Developers: 2019 – present

What is MESA?

MESA, after astropy, is the most widely used, open-source software project in astronomy and astrophysics

MESA, after astropy, is the most widely used, open-source software project in astronomy and astrophysics

MESA and astropy were named in the Astro 2020 Decadal Survey as critical instruments for the future of astronomy in the next decade

MESA is an incredibly versatile
software instrument

MESA is an incredibly versatile
software instrument

MESA is a team of developers
and thousands of users worldwide

MESA is an incredibly versatile
software instrument

MESA is a team of developers
and thousands of users worldwide

MESA is a model for open-source
and community-driven science

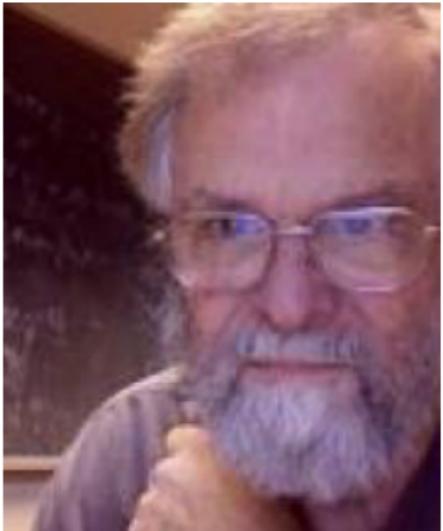
A Brief History of Time(steps)

A Brief History of Time(steps)

Our story
begins
with two
characters...

A Brief History of Time(steps)

Bill Paxton



Our story
begins
with two
characters...

Lars Bildsten



UC SANTA BARBARA
Kavli Institute for
Theoretical Physics



Bill
Paxton

■ Senior Fellow

KITP, Kohn Hall
University of California
Santa Barbara, CA 93106

Bill Paxton

Biographical Sketch



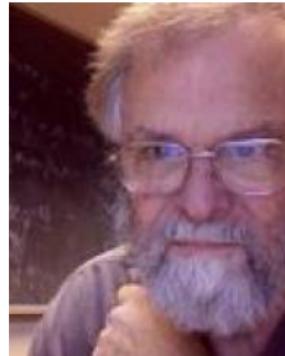
My academic and career background is in Computer Science. While I was in graduate school, I worked with **Doug Engelbart** at **SRI** where he and his group were busy inventing personal computing (check out the 40th Anniversary Celebration of the "**Mother of All Demos**"). As soon as I got my "union card" from Stanford (Ph.D., 1977), I went to work at the Xerox Palo Alto Research Center (**PARC**) where they were creating new technologies like Ethernet, networked personal computers, bitmap displays, graphical user-interfaces, and laser printers. Two of my colleagues at PARC, **Chuck Geschke** and **John Warnock**, eventually left to form **Adobe Systems**. I joined them soon after (1983), in time to help build the original PostScript and later become one of the Adobe recipients of the **ACM's Software System Award** (1989) for PostScript's design and implementation (among other things, I did the Type 1 font algorithms starting from some of John's great ideas).

Thanks to Adobe I've been "retired" since 1990, and now I'm having fun being an unofficial scholar at the University of California, Santa Barbara (**UCSB**). The people at the Kavli Institute for Theoretical Physics (**KITP**), **Lars Bildsten** in particular, have been most welcoming and tolerant of my eccentricities. The stellar evolution program **EZ** was created as part of a project with Lars. I'm now working on **MESA**, an open source set of modules for software experiments in stellar astrophysics.

Since a physicist can do "astro-physics", I imagine a Computer Scientist can do "astro-computing" or perhaps it could be called "computational-astro-physics". That seems to be a good description of what I'm up to these days, and I'm having a great time! But if I do happen to have a bad day, I can always turn to Calvin for inspiration.

A Brief History of Time(steps)

2011 – Instrument paper I Paxton et al.



Bill Paxton

2013 – Instrument paper II Paxton et al.



Lars Bildsten

2015 – Instrument paper III Paxton et al.

2018 – Instrument paper IV Paxton et al.

2019 – Instrument paper V Paxton et al.

2023 – Instrument paper VI Jermyn et al.

ADS library containing
these papers (+errata):

[https://ui.adsabs.harvard.edu/
user/libraries/
vT_uYj92TP6KMn4QWYBcVQ](https://ui.adsabs.harvard.edu/user/libraries/vT_uYj92TP6KMn4QWYBcVQ)



Beatrice M. Tinsley Prize

The Tinsley Prize recognizes an outstanding research contribution to astronomy or astrophysics, of an exceptionally creative or innovative character. The prize is normally awarded every two years.

2021 – Bill Paxton

For his inspired work on providing, maintaining, and supporting the use of open-source stellar-evolution codes that have seeped into the foundation of research and education efforts.

We have **6 instrument papers**, and they should each be cited separately if you use MESA in a paper

Citing MESA

You should cite all of the available MESA instrument papers at the time of the MESA version being used, as MESA is sum of this work. Currently, that is:

```
Modules for Experiments in Stellar Astrophysics  
\citet[MESA]{Paxton2011, Paxton2013, Paxton2015, Paxton2018, Paxton2019, Jermyn2023}.
```

A bibtex file containing these references is available here:
https://docs.mesastar.org/en/release-r24.03.1/using_mesa/best_practices.html

Also!! cite the works corresponding to significant infrastructure that has been shared with the project

MESA critically rests on the hard work of many researchers who have generated the input microphysics data that underpins the eos, kap, net, and neu modules. We therefore encourage users to briefly summarize these, including appropriate citations.

The MESA EOS is a blend of the OPAL \citep{Rogers2002}, SCVH \citep{Saumon1995}, FreeEOS \citep{Irwin2004}, HELM \citep{Timmes2000}, PC \citep{Potekhin2010}, and Skye \citep{Jermyn2021} EOSes.

Radiative opacities are primarily from OPAL \citep{Iglesias1993, Iglesias1996}, with low-temperature data from \citet{Ferguson2005} and the high-temperature, Compton-scattering dominated regime by \citet{Poutanen2017}. Electron conduction opacities are from \citet{Cassisi2007} and \citet{Blouin2020}.

Nuclear reaction rates are from JINA REACLIB \citep{Cyburt2010}, NACRE \citep{Angulo1999} and additional tabulated weak reaction rates \citet{Fuller1985, Oda1994, Langanke2000}. Screening is included via the prescription of \citet{Chugunov2007}. Thermal neutrino loss rates are from \citet{Itoh1996}.

A bibtex file containing these references is available here:
https://docs.mesastar.org/en/release-r24.03.1/using_mesa/best_practices.html



Josiah Schwab



Adam Jermyn



Meridith Joyce



Evan Bauer



Earl Bellinger



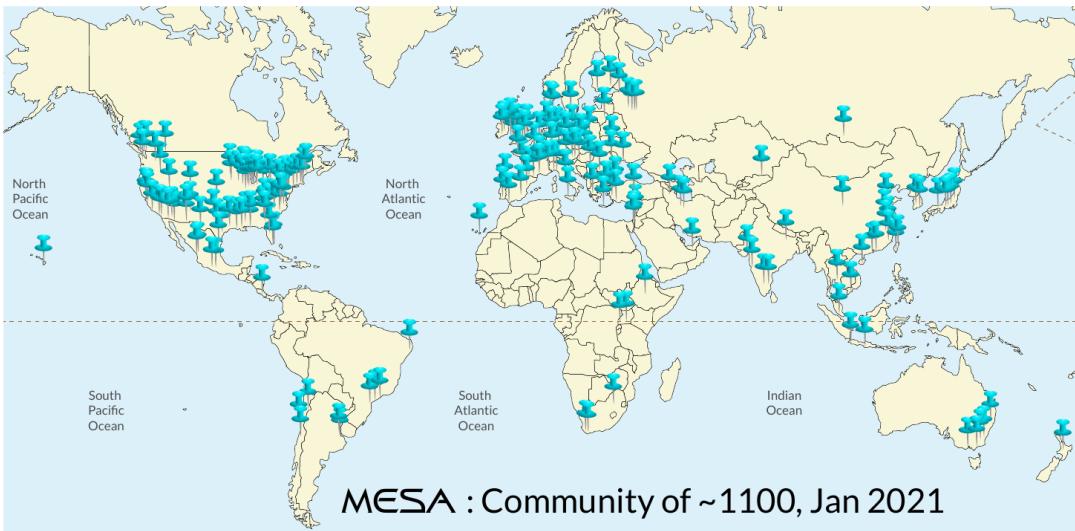
Anne Thoul



Radek Smolec



Rob Farmer



Bill Wolf



Pablo Marchant



Warrick Ball



Aaron Dotter



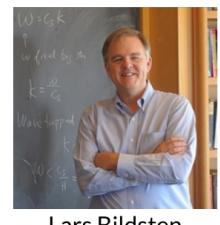
Rich Townsend



Frank Timmes



Bill Paxton



Lars Bildsten



Matteo Cantiello



Josiah Schwab



Adam Jermyn



Meridith Joyce



Evan Bauer



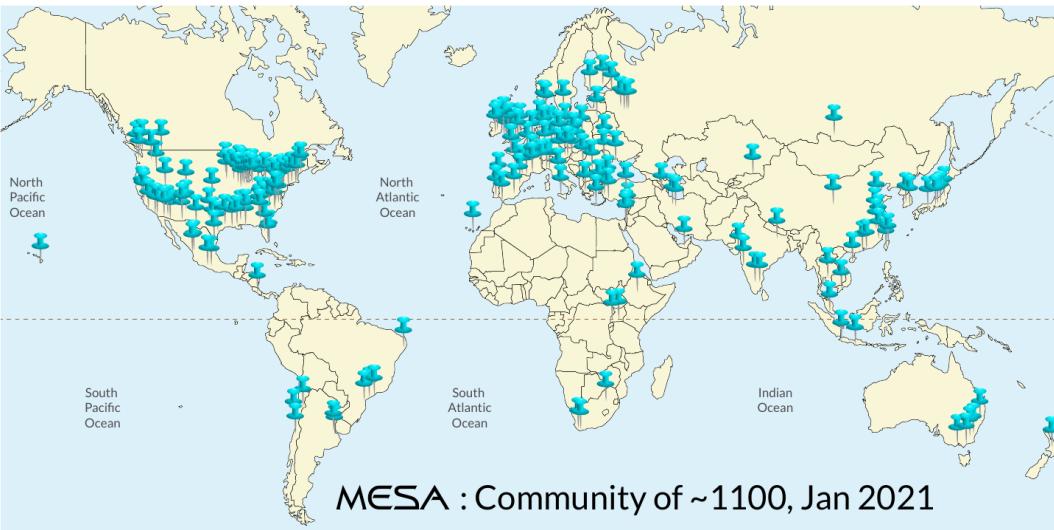
Earl Bellinger



Anne Thoul



Radek Smolec



Bill Wolf



Rob Farmer



Pablo Marchant



Warrick Ball



Aaron Dotter



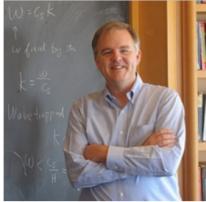
Rich Townsend



Frank Timmes



Bill Paxton



Lars Bildsten



Matteo Cantiello



Joey Mombarg



Ebraheem Farag



Meridith Joyce



Evan Bauer



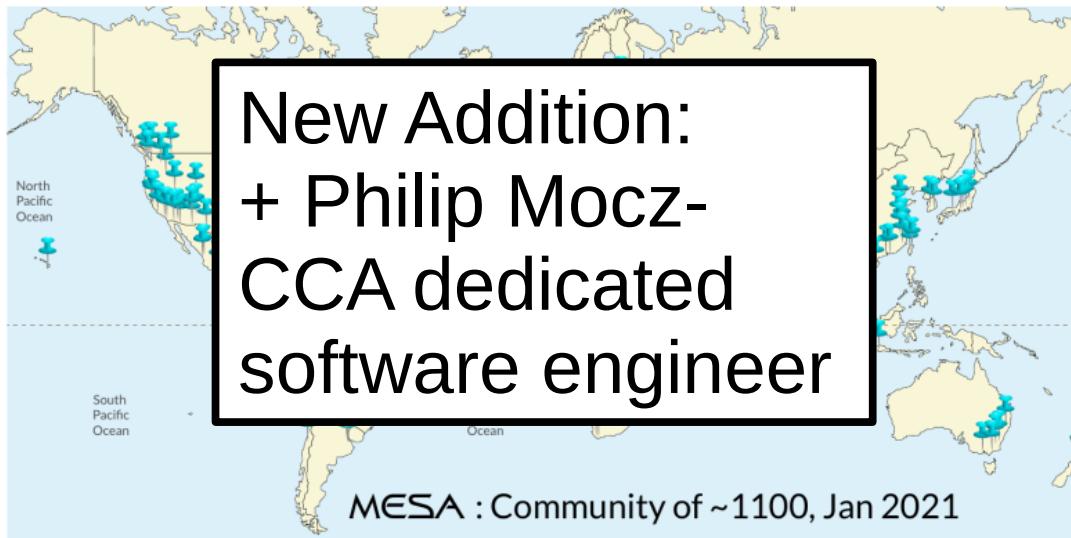
Earl Bellinger



Anne Thoul



Radek Smolec



Bill Wolf



Matthias Fabry



Pablo Marchant



Warrick Ball



Philip Mocz



Rich Townsend



Frank Timmes



Lars Bildsten



Matteo Cantiello

You can check out the past
and current developers at:

<https://docs.mesastar.org/en/release-r24.03.1/about.html#the-mesa-team>

MESA Summer Schools

Historically hosted at the University of California,
Santa Barbara (11 years)...

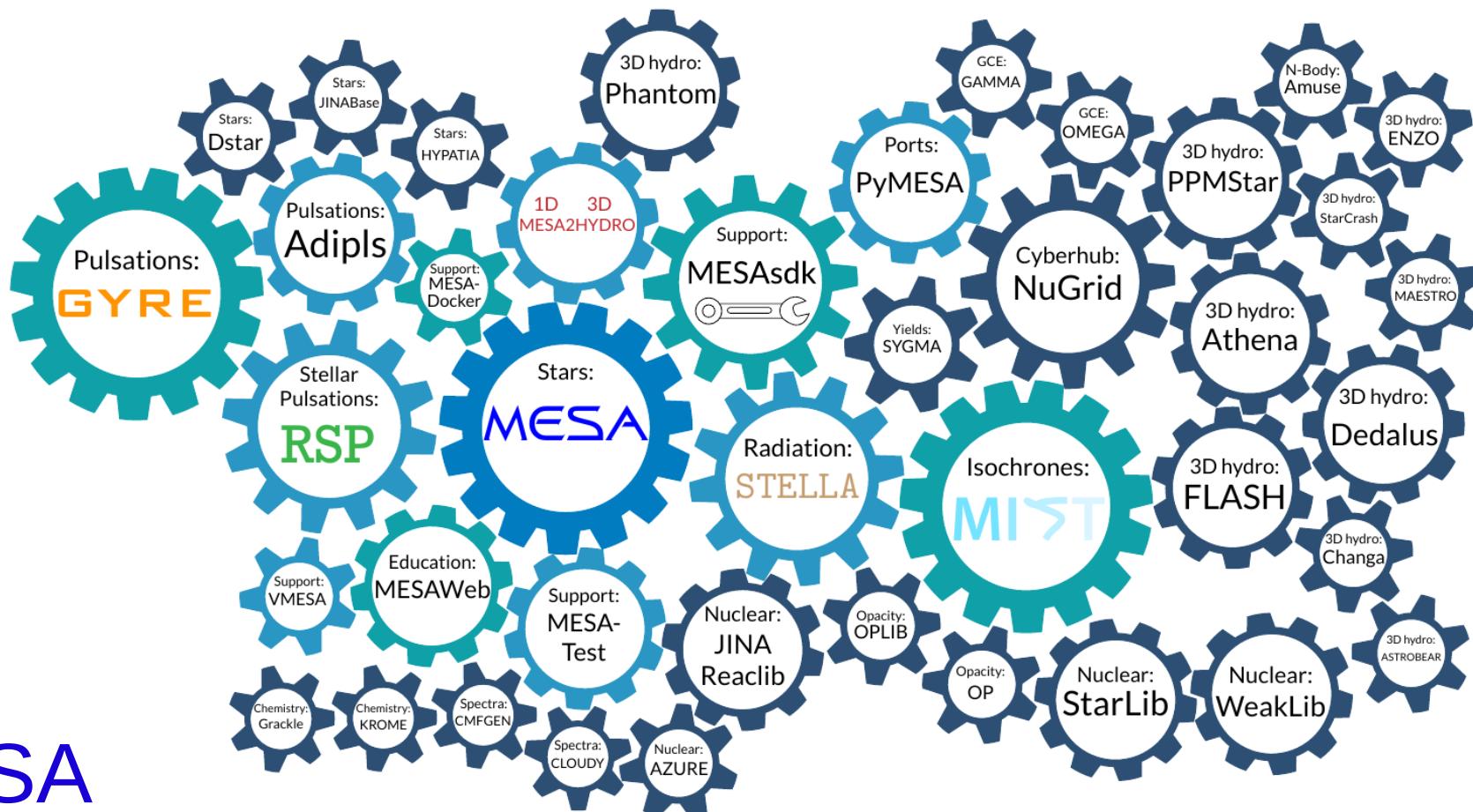
Last year: MESA@Konkoly



**MESA@Konkoly Summer School
Budapest, 2023.08.28-09.01**



Gaia LIGO SDSS Hubble JWST LSST TESS LCOGT NuSTAR



MESA Open knowledge philosophy

MESA Open knowledge philosophy

Open Source

The code is freely available, but that's only one piece of the picture.
In order for a product to be truly open source, it also has to be
usable

MESA Open knowledge philosophy

Open Source

The code is freely available, but that's only one piece of the picture. In order for a product to be truly open source, it also has to be usable

MESA's documentation is thorough, and extensive tutorials and pedagogical materials are freely available along with the software itself. The team is highly responsive to user questions

MESA Open knowledge philosophy

Open Knowledge

MESA Open knowledge philosophy

Open Knowledge

MESA “best practices” encourage the sharing of your parameter control files (inlists) as well as analysis scripts, numerical data, code for making figures, etc. The aspiration is *complete reproducibility* of science that uses MESA.

MESA Open knowledge philosophy

Open Knowledge

MESA “best practices” encourage the sharing of your parameter control files (inlists) as well as analysis scripts, numerical data, code for making figures, etc. The aspiration is *complete reproducibility* of science that uses MESA.

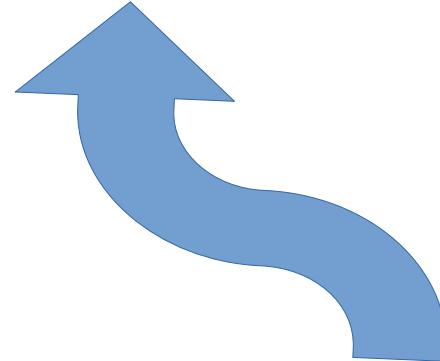
We can’t personally enforce this, but we hope the value of this level of scientific accountability is obvious. This is the way the world is heading, and grant agencies are adapting their funding priorities accordingly (e.g., open source/data sharing requirements in the EU)

Zenodo repository:

<https://zenodo.org/records/10783349>

Core Resources

Zenodo repository:
<https://zenodo.org/records/10783349>



Core Resources

This is where you should
download the code from!

Core Resources

zenodo.org/records/10783349

zenodo

Search records... 

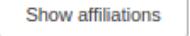
Communities My dashboard

MESA Modules for Experiments in Stellar Astrophysics (MESA)

Published March 5, 2024 | Version r24.03.01

Software  Open

Modules for Experiments in Stellar Astrophysics (MESA)

Paxton, Bill¹ 

Release versions of Modules for Experiments in Stellar Astrophysics (MESA) in ZIP format.

Files

mesa-r24.03.1.zip 

 mesa-r24.03.1.zip 

Core Resources

Zenodo repository:

<https://zenodo.org/records/10783349>

Github repository:

<https://github.com/MESAHub/mesa>

Software Development Kit (SDK):

<http://user.astro.wisc.edu/~townsend/static.php?ref=mesasdk>

The code itself:

Mesa-r24.03.1/star/test_suite

Mesa-r24.03.1/star/defaults/*.list ; *.defaults

The web-hosted documentation:

<https://docs.mesastar.org/en/release-r24.03.1/>

Core Resources

inlists used in academic papers:

https://cococubed.com/mesa_market/inlists.html

Past MESA Summer School lectures and labs, including solutions:

http://cococubed.asu.edu/mesa_market/education.html (2011-2022)

<https://mesahub.github.io/summer-school-2023/agenda/> (2023)

Mesa-users email list:

<https://lists.mesastar.org/mailman/listinfo/mesa-users>

py_mesa_reader by Bill Wolf:

https://github.com/wmwolf/py_mesa_reader

What can MESA do?

What can MESA do?

**MESA is BROAD, not
necessarily DEEP**

What can MESA do?

**MESA is BROAD, not
necessarily DEEP**

→ Depth comes from contributors
improving small components of
MESA relevant to their expertise

What can MESA do?

A tour of the `test_suite`

After installation, navigate to:
`mesa-r24.03.1/star/test_suite/`

A tour of the test_suite

```
mjoyce@burbridge: ~/MESA/mesa-r24.03.1/star/test_suite$ ls
drwxrwxr-x 5 mjoyce mjoyce 4096 Mar  5 13:19 12M_pre_ms_to_core-collapse
mjoyce@burbridge:~/MESA/mesa-r24.03.1/star/test_suite$ ls
12M_pre_ms_to_core-collapse  custom_rates          make_co_wd           rsp_gyre
1.3M_ms_high_Z              debugging_stuff_for_inlists   make_env             rsp_RR_Lyrae
1.4M_ms_op_mono              diffusion_smoothness    make_he_wd           rsp_save_and_load_file
15M_dynamo                   do1_rsp_test_source   make_metals          rsp_Type_II_Cepheid
1.5M_with_diffusion         do1_test_source       make_o_ne_wd         semiconvection
16M_conv_premix              each_test_clean      make_planets         simplex_solar_calibration
16M_predictive_mix           each_test_compile    make_pre_ccsn_13bvn split_burn_big_net
1M_pre_ms_to_wd              each_test_do        make_sdb             starspots
1M_thermohaline              each_test_run        make_zams            test_case_template
20M_pre_ms_to_core Collapse extended_convective_penetration make_zams_low_mass test_memory
20M_z2m2_high_rotation       gyre_in_mesa_bcep     make_zams_ultra_high_mass test_suite_helpers
5M_cephheid_blue_loop        gyre_in_mesa_envelope ns_c
7M_prems_to_AGB              gyre_in_mesa_ms      ns_h
accreted_material_j           gyre_in_mesa_rsg     ns_he
adjust_net                   gyre_in_mesa_spb     other_physics_hooks tzo
build_and_run                 gyre_in_mesa_wd      pisn
c13_pocket                  hb_2M
carbon_kh                    high_mass
cburn_inward                 high_rot_darkening radiative_levitation wd_acc_small_dm
ccsn_IIP                     high_z
check_pulse_atm              hot_cool_wind      R_CrB_star           wd_aic
check_redo                   hse_riemann
conductive_flame              irradiated_planet relax_composition_j_entropy wd_c_core_ignition
conserve-angular_momentum    list_tests
conv_core_cpm                 low_z
count_tests                  magnetic_braking  report
custom_colors                 make_brown_dwarf   rsp_BEP
                           make_co_wd           rsp_BLAP
                           make_env             rsp_Cepheid
                           make_he_wd           rsp_Cepheid_6M
                           make_metals          rsp_check_2nd_crossing
                           make_o_ne_wd         rsp_Delta_Scuti
                           make_planets
                           make_sdb
                           make_zams
                           make_zams_low_mass
                           make_zams_ultra_high_mass
                           ns_c
                           ns_h
                           ns_he
                           other_physics_hooks
                           pisn
                           ppisn
                           radiative_levitation
                           R_CrB_star
                           relax_composition_j_entropy
                           report
                           rsp_BEP
                           rsp_BLAP
                           rsp_Cepheid
                           rsp_Cepheid_6M
                           rsp_check_2nd_crossing
                           rsp_Delta_Scuti
                           rsp_gyre
                           rsp_RR_Lyrae
                           rsp_save_and_load_file
                           rsp_Type_II_Cepheid
                           semiconvection
                           simplex_solar_calibration
                           split_burn_big_net
                           starspots
                           test_case_template
                           test_memory
                           test_suite_helpers
                           timing
                           T_tau_gradr
                           twin_studies
                           tzo
                           wd_acc_small_dm
                           wd_aic
                           wd_c_core_ignition
                           wd_cool_0.6M
                           wd_diffusion
                           wd_he_shell_ignition
                           wd_nova_burst
                           wd_stable_h_burn
                           zams_to_cc_80

mjoyce@burbridge:~/MESA/mesa-r24.03.1/star/test_suite$
```

Using MESA

Environment Variables

Environment variables

To run MESA, you must set *environment variables* that point your system to the correct paths for dependent packages

The variables are:

MESA_DIR – the location of your top-level MESA directory

MESASDK_ROOT – the location of the SDK

OMP_NUM_THREADS – the number of threads (= 2x number of cores) you choose to devote to MESA calculations.

*Note that most laptops have between 2 and 8 threads available, or 1 to 4 cores. Setting **OMP_NUM_THREADS** above this will result in overextending the capabilities of your system, causing the calculations to run very slowly or crash*

Environment Variables

The syntax for setting these variables differs between MacOS and Linux and depends on your shell environment (ex. I use Linux and bash shell)

Visit

https://www.youtube.com/watch?v=NmaLHFxpALg&ab_channel=FrankTimmes

for video guides on how to set up these variables for other systems

Other video installation guides also available at my website:

<https://www.meridithjoyce.com/talks.html>

Setting environment variables: the bad way

```
mjoyce@burbidge: ~/MESA/mesa-r24.03.1
drwxrwxr-x 11 mjoyce mjoyce 4096 Mar  5 13:19 binary
drwxrwxr-x  8 mjoyce mjoyce 4096 Mar  5 13:19 auto_diff
drwxrwxr-x  8 mjoyce mjoyce 4096 Mar  5 13:19 atm
drwxrwxr-x  9 mjoyce mjoyce 4096 Mar  5 13:19 astero
drwxrwxr-x  6 mjoyce mjoyce 4096 Mar  5 13:19 adipls
-rw-rw-r--  1 mjoyce mjoyce   172 Jun 16 15:27 testhub.yml
drwxrwxr-x 16 mjoyce mjoyce 4096 Jun 16 15:27 eos
drwxrwxr-x 11 mjoyce mjoyce 4096 Jun 16 15:33 kap
drwxrwxr-x  8 mjoyce mjoyce 4096 Jun 16 15:34 ionization
drwxrwxr-x 14 mjoyce mjoyce 4096 Jun 16 15:35 data
drwxrwxr-x  5 mjoyce mjoyce 4096 Jun 16 15:35 sample
drwxrwxr-x  6 mjoyce mjoyce 4096 Jun 16 15:35 gyre
drwxrwxr-x  2 mjoyce mjoyce 4096 Jun 16 15:39 lib
drwxrwxr-x  2 mjoyce mjoyce 4096 Jun 16 15:39 include
-rw-rw-r--  1 mjoyce mjoyce 67377 Jun 16 15:39 build.log
mjoyce@burbidge:~/MESA/mesa-r24.03.1$ export MESA_DIR=/home/mjoyce/MESA/mesa-r24.03.1
mjoyce@burbidge:~/MESA/mesa-r24.03.1$ export MESASDK_ROOT=/home/mjoyce/MESA/mesasdk_Jul2023
mjoyce@burbidge:~/MESA/mesa-r24.03.1$ source $MESASDK_ROOT/bin/mesasdk_init.sh
mjoyce@burbidge:~/MESA/mesa-r24.03.1$ export OMP_NUM_THREADS=2
mjoyce@burbidge:~/MESA/mesa-r24.03.1$
```

Setting environment variables the good way: `.bashrc` file

Since `.bashrc` (or equivalent) is read automatically each time you open a new terminal window, assigning environment variables in **`.bashrc` (or the equivalent profile of your local system)** means you do not need to assign them manually

Setting environment variables the good way: .bashrc file

```
.bashrc
1 # ~/.bashrc: executed by bash(1) for non-login shells.
2 # see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
3 # for examples
4
5 # If not running interactively, don't do anything
6 case $- in
7     *i*) ;;
8     *) return;;
9 esac
10
11 ## for screenshots only
12 export MESA_DIR=/home/mjoyce/MESA/mesa-r24031
13 export MESASDK_ROOT=/home/mjoyce/MESA/mesasdk
14 source ${MESASDK_ROOT}/bin/mesasdk_init.sh
15 export OMP_NUM_THREADS=2
16
```

Best way: functions in .bashrc

```
function mesa-24031 {  
export MESA_DIR=/home/mjoyce/MESA/mesa-r24.03.1  
export MESASDK_ROOT=/home/mjoyce/MESA/mesasdk_Jul2023  
source $MESASDK_ROOT/bin/mesasdk_init.sh  
export OMP_NUM_THREADS=8  
echo "environment set for MESA version 24.03.1"  
echo "SDK version 23.7.3 in use"  
echo "OMP_NUM_THREADS set to 8"  
}
```

Because
then you
can toggle
between
MESA
versions, if
you ever
have reason
to do that...

```
371
372 function mesa-15140 {
373 export MESA_DIR=/home/mjoyce/MESA/mesa-r15140
374 export MESASDK_ROOT=/home/mjoyce/MESA/mesasdk_15140
375 source $MESASDK_ROOT/bin/mesasdk_init.sh
376 export OMP_NUM_THREADS=8
377 echo "environment set for MESA version 15140"
378 }
379
380
381 function mesa-22051 {
382 export MESA_DIR=/home/mjoyce/MESA/mesa-r22051
383 export MESASDK_ROOT=/home/mjoyce/MESA/mesasdk_15140
384 source $MESASDK_ROOT/bin/mesasdk_init.sh
385 export OMP_NUM_THREADS=8
386 echo "environment set for MESA version 22.05.1"
387 }
388
389
390 function mesa-23051 {
391 export MESA_DIR=/home/mjoyce/MESA/mesa-r23051
392 export MESASDK_ROOT=/home/mjoyce/MESA/mesasdk
393 source $MESASDK_ROOT/bin/mesasdk_init.sh
394 export OMP_NUM_THREADS=2
395 echo "environment set for MESA version 23.05.1"
396 }
397
```

```
mjoyce@burbridge: ~/MESA/mesa-r24.03.1
-rw-r--r-- 1 mjoyce mjoyce 461 Mar  5 13:19 each_package_do
drwxrwxr-x  7 mjoyce mjoyce 4096 Mar  5 13:19 const
drwxrwxr-x 11 mjoyce mjoyce 4096 Mar  5 13:19 colors
drwxrwxr-x  1 mjoyce mjoyce 5195 Mar  5 13:19 CODEOWNERS
drwxrwxr-x  1 mjoyce mjoyce 1910 Mar  5 13:19 CODE_OF_CONDUCT.rst
drwxrwxr-x  1 mjoyce mjoyce 356 Mar  5 13:19 clean
drwxrwxr-x  1 mjoyce mjoyce 21100 Mar  5 13:19 CITATIONS.bib
drwxrwxr-x  8 mjoyce mjoyce 4096 Mar  5 13:19 chem
drwxrwxr-x  8 mjoyce mjoyce 4096 Mar  5 13:19 binary
drwxrwxr-x  8 mjoyce mjoyce 4096 Mar  5 13:19 auto_diff
drwxrwxr-x  8 mjoyce mjoyce 4096 Mar  5 13:19 atm
drwxrwxr-x  9 mjoyce mjoyce 4096 Mar  5 13:19 astro
drwxrwxr-x  6 mjoyce mjoyce 4096 Mar  5 13:19 adipls
mjoyce@burbridge:~/MESA/mesa-r24.03.1$ source ~/.bashrc
mjoyce@burbridge:~/MESA/mesa-r24.03.1$ mesa-24031
environment set for MESA version 24.03.1
SDK version for July 2023 in use
OMP_NUM_THREADS set to 8
mjoyce@burbridge:~/MESA/mesa-r24.03.1$
```

Now it is straightforward to set up your **MESA** environment:

Microlab 0: Check and set environment variables (~3 minutes)

Microlab 0: Check and set environment variables

(~3 minutes)

Show of thumbs when complete!

Setup and output

Inlists – Fortran namelists that contain **value** definitions for all of the **parameters** of your run
ex) **history_filename = 'history_my_run.data'**

Parameter libraries- *all of the possible values for your parameters can be found in the module defaults files*

Mesa-r24.03.1/star/defaults/ contains

controls.defaults
star_job.defaults
pgstar.defaults
history_columns.list
profile_columns.list

Opacity defaults can be found in
mesa-r24.03.1/kap/defaults/kap.defaults

EOS defaults in
mesa-r24.03.1/eos/defaults/eos.defaults

Setup and output

Executable

star or **binary**; this is the program that is built by the compiler and runs your simulation

Scripts

clean, mk, rn, re – these are shell scripts that build and manipulate your program

Setup and output

By default, MESA keeps track of the full stellar structure of your model across evolutionary time

Output is stored in the **LOGS/** directory

history.data traces evolutionary quantities

profileX.data gives you the structural model at some time step dt .
you can adjust the frequency of these outputs in the inlists

profiles.index provides a mapping between the integer in the profile output names and the model number from the evolutionary run (in cases where a profile is not generated at every time step)

You can also store binary snapshots of the models: **photos**

Setup and output

History output should look something like this:

```
~/MESA/mesa-r15140/star/work/LOGS/history.data - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help
inlist_start_header | inlist_start | .bashrc | history.data
1 | 1 | 2 | 3
| 4 | 5 | 6
2 | version number | compiler | build
MESA_SDK version | math backend | date
burn_min1 | burn_min2 |
"15140" | "gfortran" | "CRMATH" | "10.2.0"
|x86_64-linux-20.12.1" | 1.000000000000000E+003 | "20211101"
5.000000000000000E+001
4 | 1 | 2 | 3
| 4 | 5 | 6
| 7 | 8 | 9
| 10 | 11 | 12
| 13 | 14 | 15
| 16 | 17 | 18
| 19 | 20 | 21
| 22 | 23 | 24
| 25 | 26 | 27
| 28 | 29 | 30
| 31 | 32 | 33
| 34 | 35 | 36
| 37 | 38 | 39
| 40 | 41 | 42
| 43 | 44 | 45
| 46 | 47 | 48
| 49 | 50 | 51
| 52 | 53 | 54
| 55 | 56 | 57
| 58
6 | model_number | num_zones | star_age | log_abs_mdot
log_dt | star_mass | log_xmstar | log_xmstar |
mass_conv_core | conv_mx1_top | conv_mx1_bot | conv_mx1_bot |
conv_mx2_top | conv_mx2_bot | mx2_top | mx2_top |
mx1_bot | log_LHe | log_LZ | log_LZ |
log_LH | pp | cno | epsnuc_M_2 |
log_Lnuc | epsnuc_M_1 | epsnuc_M_3 | epsnuc_M_5 |
tri_alfa | epsnuc_M_4 | epsnuc_M_6 | epsnuc_M_7 |
epsnuc_M_3 | epsnuc_M_8 | he_core_mass | o_core_mass |
epsnuc_M_6 | he_core_mass | fe_core_mass | neutron_rich_core_mass |
he_core_mass | log_L | log_R | log_g |
si_core_mass | log_cntr_P | log_cntr_rho | log_g |
log_Teff | center_mu | center_mu | center_ye |
v_div_csound_surf | center_h1 | center_h1 | center_h4 |
log_cntr_T | center_o16 | center_o16 | surface_c12 |
center_abar | total_mass_h1 | total_mass_h1 | total_mass_h4 |
center_c12 | num_iters | 492 | 1.000000000000000E-005 |
surface_o16 | num_iters | 492 | -4.999999999999991E+000 |
7 | 1.500000000000000E+001 | 3.4474597169416349E+001 | -9.899999999999986E+001 |
0.000000000000000E+000 | 9.9998828386135030E-001 | 3.0124054712287992E-003 |
0.000000000000000E+000 | 0.000000000000000E+000 | 9.9998828386135030E-001 |
3.0124054712287992E-003 | 0.000000000000000E+000 | 0.000000000000000E+000 |
-1.711367145755475E+001 | -9.899999999999986E+001 | -9.899999999999986E+001 |
-1.711367145755475E+001 | -1.711367145755475E+001 | -9.899999999999986E+001 |
-9.899999999999986E+001 | -2.000000000000000E+001 | -2.000000000000000E+001

Line 1, Column 1
Spaces: 3 Plain Text
```

Setup and output

History output should look something like this:

~/MESA/mesa-r15140/star/work/LOGS/history.data - Sublime Text (UNREGISTERED)

```
File Edit Selection Find View Goto Tools Project Preferences Help
inlist_start_header | inlist_start | .bashrc | history.data
1 | 1 | 2 | 3 |
| 4 | 5 | 6 |
2 | version number | compiler | build |
MESA_SDK version | math backend | date |
burn_min1 | burn_min2 | |
"15140" | "gfortran" | "10.2.0" |
"x86_64-linux-20.12.1" | "CRMATH" | "20211101" |
5.00000000000000E+001 | 1.00000000000000E+003 |
4 | 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |
| 10 | 11 | 12 |
| 13 | 14 | 15 |
| 16 | 17 | 18 |
| 19 | 20 | 21 |
| 22 | 23 | 24 |
| 25 | 26 | 27 |
| 28 | 29 | 30 |
| 31 | 32 | 33 |
| 34 | 35 | 36 |
| 37 | 38 | 39 |
| 40 | 41 | 42 |
| 43 | 44 | 45 |
| 46 | 47 | 48 |
| 49 | 50 | 51 |
| 52 | 53 | 54 |
| 55 | 56 | 57 |
| 58 | 59 | 60 |
6 | model_number | num_zones | star_age |
log_dt | star_mass | log_xmstar |
mass_conv_core | conv_mx1_top | conv_mx1_bot |
conv_mx2_top | conv_mx2_bot | mx1_top |
mx1_bot | mx2_top | mx2_bot |
log_LH | log_LHe | log_LZ |
log_Lnuc | pp | cno |
tri_alfa | epsnuc_M_1 | epsnuc_M_2 |
epsnuc_M_3 | epsnuc_M_4 | epsnuc_M_5 |
epsnuc_M_6 | epsnuc_M_7 | epsnuc_M_8 |
he_core_mass | fe_core_mass | o_core_mass |
si_core_mass | log_Teff | neutron_rich_core_mass |
log_Teff | v_div_csound_surf | log_R |
y_div_csound_surf | log_cntr_P | log_cntr_rho |
log_cntr_T | center_mu | center_ye |
center_abar | center_h1 | center_h4 |
center_c12 | center_o16 | surface_c12 |
surface_o16 | total_mass_h1 | total_mass_h4 |
num_iter | num_iter | log_g |
7 | 1 | 492 | 1.00000000000000E-005 |
| 1.50000000000000E+001 | 9.899999999999986E+001 |
| 0.00000000000000E+000 | -9.899999999999986E+001 |
| 0.00000000000000E+000 | 3.0124054712287992E-003 |
| 3.0124054712287992E-003 | 9.9998828386135030E-001 |
| -1.711367145755475E+001 | 0.00000000000000E+000 |
| -1.711367145755475E+001 | -9.899999999999986E+001 |
| -9.899999999999986E+001 | -1.711367145755475E+001 |
| -2.00000000000000E+001 | -2.00000000000000E+001 |
Line 1, Column 1 | Spaces: 3 | Plain Text
```

Microlab 1:

- 1) Explore your inlists
- 2) Run MESA out-of-the-box

(~10 minutes)

Microlab 1: Exploring MESA & inlists (10 min)

- (1) What is the initial mass of the model that runs when you launch a simulation in mesa-r24.03.1/star/work/ ?
- (2) What causes this model to stop?
- (3) Does MESA create a starting model for this simulation, or does it load an existing model?
- (4) Try to run this model using **`.clean`**; **`.mk`**; **`.rn`**
Ask a TA for help if you cannot! What do these scripts do?
- (5) How hot is the star at step number 35 of this run?

Using run_star_extras.f90

Using `run_star_extras.f90`, we can introduce our own
-project-specific physics, or
-additional functionality
without compromising the entire **MESA** source code base

File Edit Selection Find View Goto Tools Project Preferences Help

run_star_extras.f90 x

```
1 ! ****
2 Copyright (C) 2010-2019 Bill Paxton & The MESA Team
3
4 this file is part of mesa.
5
6 mesa is free software;
7 it under the terms of t
8 by the free software fo
9 (at your option) any la
10
11 mesa is distributed in
12 but without any warranty, without even the implied warranty of
13 merchantability or fitness for a particular purpose. see the
14 gnu library general public license for more details.
15
16 you should have received a copy of the gnu library general public license
17 along with this software; if not, write to the free software
18 foundation, inc., 59 temple place, suite 330, boston, ma 02111-1307 usa
19
20 ****
21
22 module run_star_extras
23
24 use star_lib
25 use star_def
26 use const_def
27 use math_lib
28
29 implicit none
30
31 ! these routines are called by the standard run_star check_model
32 contains
33
34 include 'standard_run_star_extras.inc'
35
36 end module run_star_extras
37
38
39
```

* Aa " " C≡

Find Find Prev Find All

Spaces: 6 Fortran (Modem)

Line 19, Column 42

run_star_extras.f90: default

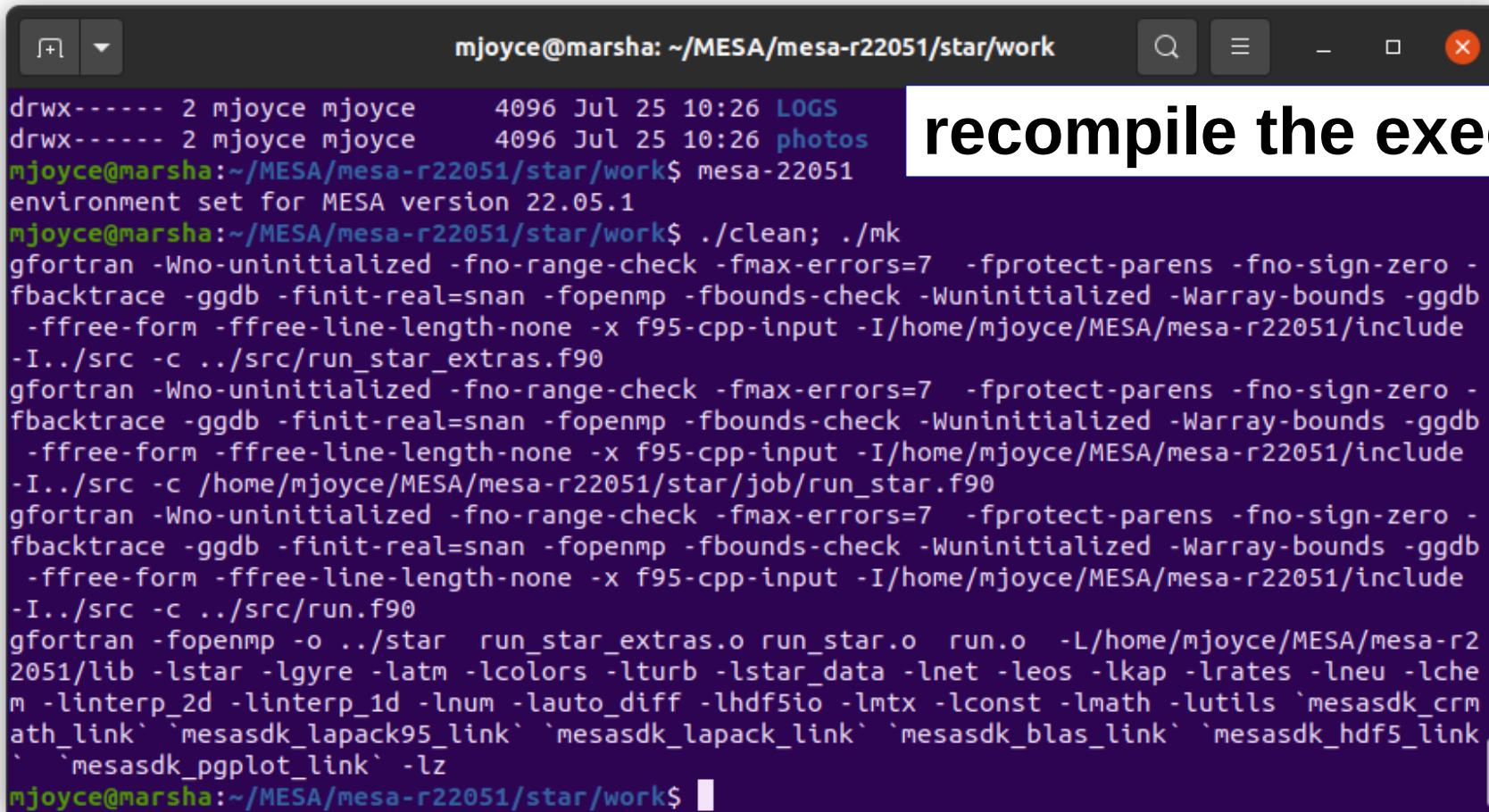
File Edit Selection Find View Goto Tools Project Preferences Help

```
22
23 module run_star_extras
24
25 use star_lib
26 use star_def
27 use const_def
28 use math_lib
29
30 implicit none
31
32 ! these routines are called by the standard run_star check_model
33 contains
34
35 subroutine extras_controls(id, ierr)
36     integer, intent(in) :: id
37     integer, intent(out) :: ierr
38     type (star_info), pointer :: s
39     ierr = 0
40     call star_ptr(id, s, ierr)
41     if (ierr /= 0) return
42
43 ! this is the place to set any procedure pointers you want to change
44 ! e.g., other_wind, other_mixing, other_energy (see star_data.inc)
45
46
47 ! the extras functions in this file will not be called
48 ! unless you set their function pointers as done below.
49 ! otherwise we use a null_version which does nothing (except warn).
50
51 s% extras_startup => extras_startup
52 s% extras_start_step => extras_start_step
53 s% extras_check_model => extras_check_model
54 s% extras_finish_step => extras_finish_step
55 s% extras_after_evolve => extras_after_evolve
56 s% how_many_extra_history_columns => how_many_extra_history_columns
57 s% data_for_extra_history_columns => data_for_extra_history_columns
58 s% how_many_extra_profile_columns => how_many_extra_profile_columns
59 s% data_for_extra_profile_columns => data_for_extra_profile_columns
60
61 s% how_many_extra_history_header_items => how_many_extra_history_header_items
62 s% data_for_extra_history_header_items => data_for_extra_history_header_items
63 s% how_many_extra_profile_header_items => how_many_extra_profile_header_items
64 s% data_for_extra_profile_header_items => data_for_extra_profile_header_items
```

run_star_extras.f90: include

Using run_star_extras.f90

Every time you modify **run_star_extras**, you must

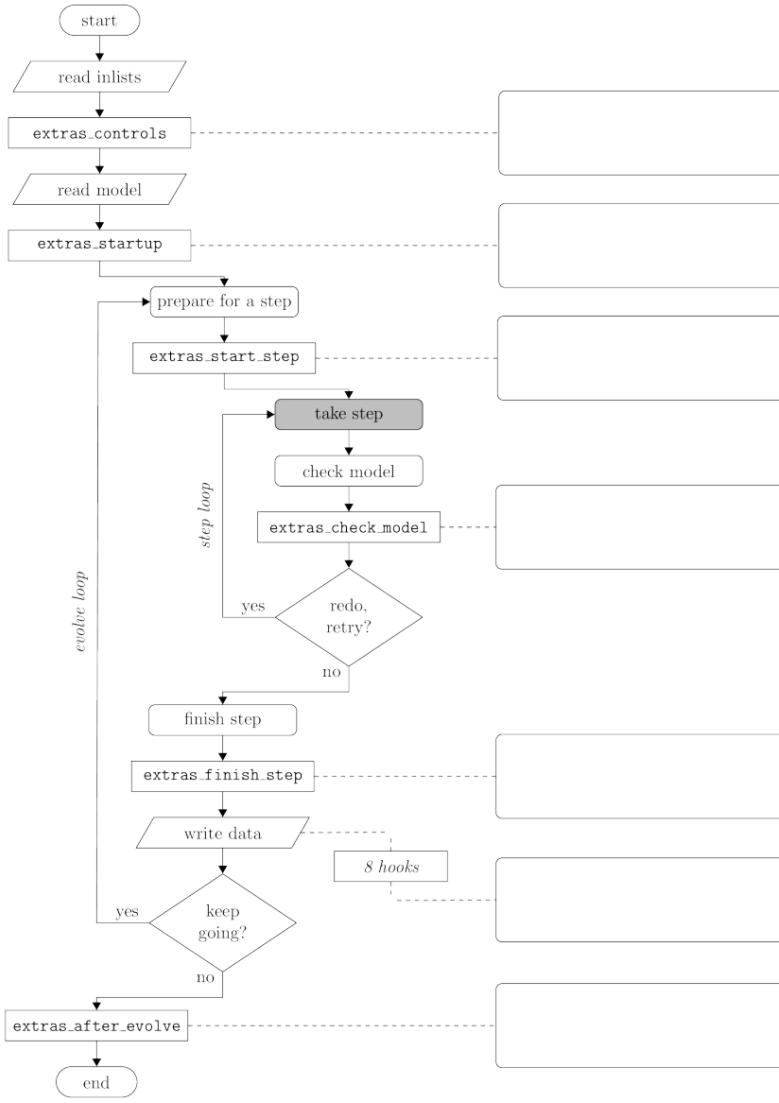


mjoyce@marsha: ~/MESA/mesa-r22051/star/work

```
drwx----- 2 mjoyce mjoyce 4096 Jul 25 10:26 LOGS
drwx----- 2 mjoyce mjoyce 4096 Jul 25 10:26 photos
mjoyce@marsha:~/MESA/mesa-r22051/star/work$ mesa-22051
environment set for MESA version 22.05.1
mjoyce@marsha:~/MESA/mesa-r22051/star/work$ ./clean; ./mk
gfortran -Wno-uninitialized -fno-range-check -fmax-errors=7 -fprotect-parens -fno-sign-zero -
-fbacktrace -ggdb -finit-real=snan -fopenmp -fbounds-check -Wuninitialized -Warray-bounds -ggdb
-fffree-form -ffree-line-length-none -x f95-cpp-input -I/home/mjoyce/MESA/mesa-r22051/include
-I../src -c ../src/run_star_extras.f90
gfortran -Wno-uninitialized -fno-range-check -fmax-errors=7 -fprotect-parens -fno-sign-zero -
-fbacktrace -ggdb -finit-real=snan -fopenmp -fbounds-check -Wuninitialized -Warray-bounds -ggdb
-fffree-form -ffree-line-length-none -x f95-cpp-input -I/home/mjoyce/MESA/mesa-r22051/include
-I../src -c /home/mjoyce/MESA/mesa-r22051/star/job/run_star.f90
gfortran -Wno-uninitialized -fno-range-check -fmax-errors=7 -fprotect-parens -fno-sign-zero -
-fbacktrace -ggdb -finit-real=snan -fopenmp -fbounds-check -Wuninitialized -Warray-bounds -ggdb
-fffree-form -ffree-line-length-none -x f95-cpp-input -I/home/mjoyce/MESA/mesa-r22051/include
-I../src -c ../src/run.f90
gfortran -fopenmp -o ../star run_star_extras.o run_star.o run.o -L/home/mjoyce/MESA/mesa-r2
2051/lib -lstar -lgyre -latm -lcolors -lturb -lstar_data -lnet -leos -lkap -lrates -lneu -lche
m -linterp_2d -linterp_1d -lnum -lauto_diff -lhdf5io -lmtx -lconst -lmath -lutils `mesasdk_crm
ath_link` `mesasdk_lapack95_link` `mesasdk_lapack_link` `mesasdk_blas_link` `mesasdk_hdf5_link
` `mesasdk_pgplot_link` -lz
mjoyce@marsha:~/MESA/mesa-r22051/star/work$
```

recompile the executable!!

Code Organization



Go to Sunny Wong's Lab 3 for a high-resolution version of this chart!

https://courtcraw.github.io/mesadu_wdbinaries/lab3.html

Code Organization

There are some actions you will want to compute *once per evolutionary time step (evolve loop)*

there are others you may want to compute *once per Newton solver iteration (step loop)*

where one evolve step contains several solver iterations

Code Organization

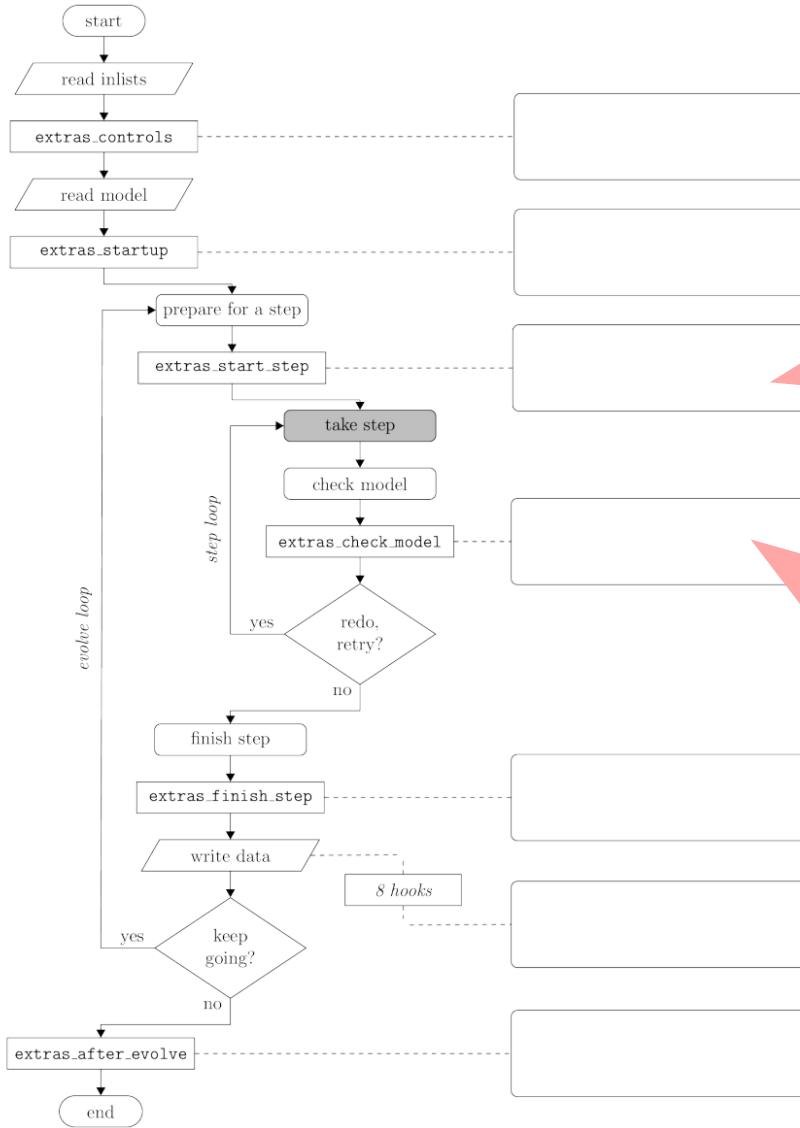
There are some actions you will want to compute *once per evolutionary time step (evolve loop)*

there are others you may want to compute *once per Newton solver iteration (step loop)*

where one evolve step contains several solver iterations

Example:

Checking whether your model has satisfied some global physical property (i.e., reaching a certain radius) can take place once per evolutionary step



```

integer function extras_start_step(id)
  integer, intent(in) :: id
  integer :: ierr
  type (star_info), pointer :: s
  ierr = 0
  call star_ptr(id, s, ierr)
  if (ierr /= 0) return
  extras_start_step = 0
end function extras_start_step

```

```

! returns either keep_going, retry, or terminate.
integer function extras_check_model(id)
  integer, intent(in) :: id
  integer :: ierr
  type (star_info), pointer :: s
  ierr = 0
  call star_ptr(id, s, ierr)
  if (ierr /= 0) return
  extras_check_model = keep_going
  if (.false. .and. s%star_mass_h1 < 0.35d0) then
    ! stop when star hydrogen mass drops to specified level
    extras_check_model = terminate
    write(*, *) 'have reached desired hydrogen mass'
    return
  end if

  if (extras_check_model == terminate) s%termination_code = t_extras_check_model
end function extras_check_model

```

Suppose we want MESA to stop when the star reaches a certain luminosity

- When during the **step** should this condition be checked?
- How often should this condition be checked?
- In which **subroutine** should we check this condition?

Getting involved in the MESA project

Getting involved in the MESA project

users-list engagement: asking and answering questions

Feature requests

Feature development and sharing your code

Large contributions

Becoming a **MESA developer**

“MESA doesn’t work!”

“MESA sucks at X!”

“I want MESA to do something it does not currently do!”

Raise an issue on github!

Raise an issue on github!

<https://github.com/MESAHub/mesa/issues>

[Website]

Becoming a MESA developer

- Membership to the MESA developers team is done by nomination
- Any MESA developer can nominate a new member. The existing members of the team have two weeks to approve the nomination or not
- Typically, nomination is discussed with a candidate before the formal nomination process

Becoming a MESA developer

The MESA Team

The missions of the MESA Team are:

- **Stewardship:** supporting contributors, maintaining the access and updates, seeking enabling funding, supporting MESA Summer Schools that allow for continued engagement, documenting MESA development in the refereed literature, and sustaining advanced development.
- **Interface with the User Community:** answering questions from users, developing or accepting new code in an integrated fashion, supporting MESA workshops and events, maintaining a user registry, and identifying new MESA Team members from those most active and engaged in the intelligent use of MESA.
- **Enable Scientific Research and Education:** promoting MESA and its goals, e.g., through scientific contributions at relevant conferences, identifying science opportunities that match MESA capabilities, facilitating and encouraging appropriate scientific collaborative

Becoming a MESA developer

Supporting the MESA project is voluntary service work, but it is a prestigious project that opens a lot of opportunities

Writing code—whether developing MESA directly, its support tools, or programs that integrate with it—is a core component of MESA development

But it is not the only way to contribute!



Joey Mombarg



Ebraheem Farag



Meridith Joyce



Evan Bauer



Earl Bellinger



Anne Thoul



Radek Smolec



Matthias Fabry



Bill Wolf



Pablo Marchant



Warrick Ball



Philip Mocz



Rich Townsend



Frank Timmes



Lars Bildsten



Matteo Cantiello



Evan Bauer



Evan Bauer



Meridith Joyce



Evan Bauer



Evan Bauer



Meridith Joyce



Evan Bauer



What it might as well be



Evan Bauer



MESA : Community of ~1100, Jan 2021



Evan Bauer



Evan Bauer



Evan Bauer



Evan Bauer



Evan Bauer



Evan Bauer



Evan Bauer

Becoming a MESA developer

As developers inevitably move on (or advance to professor, at which point they no longer have time to do anything), it is important to bring outstanding young astrophysicists into the team to keep MESA relevant and ensure that it continues to serve the needs of the research community

Becoming a MESA developer

As developers inevitably move on (or advance to professor, at which point they no longer have time to do anything), it is important to bring outstanding young astrophysicists into the team to keep MESA relevant and ensure that it continues to serve the needs of the research community

We are especially interested in **recruiting women**. If you would like to know more about what it means to be a **MESA developer**, please talk to me at this workshop!



PROGRAMS & MAJORS

ADMISSIONS

COST & FINANCIAL AID

CURRENT STUDENTS

MORE

APPLY TO UW



UW Homepage

School of Computing

People

Faculty

Meridith Joyce

UNIVERSITY OF WYOMING |
SCHOOL OF COMPUTING

MENU

RESEARCH SCIENTISTS

ELLEN AIKENS | ASSISTANT

PROFESSOR

MARGO BERENDSEN | GIS ANALYST

SEAN FIELD | ASSISTANT

PROFESSOR

MERIDITH JOYCE | ASSISTANT

PROFESSOR



Meridith Joyce

Assistant Professor | Physics | Astronomy | Mathematics | Statistics | School of Computing

Room 4083, Engineering

Laramie, WY

Phone: 307-766-5299

Email: mjoyce8@uwyo.edu

Talk to me about
PhD openings at
Wyo or how to apply
to postdocs and
fellowships in the
**United States and
Europe!**