MESA

Modules for Experiments in Stellar Astrophysics

- MESA home
- code capabilities
- prereqs & installation
- getting started
- using pgstar
- using MESA output
- beyond inlists (extending MESA)
- troubleshooting
- FAQ
- star_job defaults
- controls defaults
- pgstar defaults
- binary_controls defaults
- news archive
- documentation archive

Latest News

- 21 Mar 2018
  » Release 10398
- 28 Jan 2018
  » New MESA SDK Version
- 13 Jan 2018
  » Summer School 2018
- 23 Oct 2017
  » Release 10108
- 23 Oct 2017
  » Instrument Paper 4
- 21 Sep 2017
  » New MESA SDK Version
- 12 Sep 2017
  » Release 10000
- 01 Sep 2017
  » Docker, Python and Web
- 14 Aug 2017
  » New mesa-users list
- 02 Aug 2017
  » New MESA SDK Version

This web documentation corresponds to the most recent MESA release (r10398). Documentation for past versions can be found here.

If you find errors or formatting issues, please email Josiah Schwab using this link.

This page documents the MESA options that are part of the star_job namelist. It is autogenerated from the file $MESA_DIR/star/defaults/star_job.defaults.

Boxes like

```
option_name = 'default'
```

show the default value of each option. To override the default values, add an entry to the star_job namelist in your inlist.

Contents

directories ¶

mesa_dir ¶

if set to the empty string, ‘’, then it defaults to using environment variable `$(MESA_DIR)`

```
mesa_dir = ''
```

chem_isotopes_filename ¶

this file is in `chem_data` in `mesa_data_dir`

```
chem_isotopes_filename = 'isotopes.data '
```

pause_before_terminate ¶

if true, then will pause before terminate run. this can be useful if you’d like a chance to look at the final model pgstar windows before they go away.

```
pause_before_terminate = .false.
```

cache directories ¶

eosDT_cache_dir ¶

eosPT_cache_dir ¶

eosDE_cache_dir ¶

ionization_cache_dir ¶

kap_cache_dir ¶

rates_cache_dir ¶

mesa uses caches to improve performance. the default location for these is in the mesa/data directory, but in some situations it is useful to keep the caches separately so, for example, multiple users can share the code and each can have a separate set of caches. ‘’ means use default location for cache.

The need for separate caches arises in cases where we need to put the main mesa directory in a location that is “read only” for a group of users (such as in a system directory that requires “root” or “superuser” to write). In that case the caches must be moved out of the main directory to locations that the user can write.

if you specify cache directories, use a separate one for each. e.g., something like this

```
eosDT_cache_dir = '/Users/bpaxton/mesa_caches/eosDT_cache'
eosPT_cache_dir = '/Users/bpaxton/mesa_caches/eosPT_cache'
eosDE_cache_dir = '/Users/bpaxton/mesa_caches/eosDE_cache'
ionization_cache_dir = '/Users/bpaxton/mesa_caches/ionization_cache'
kap_cache_dir = '/Users/bpaxton/mesa_caches/kap_cache'
rates_cache_dir = '/Users/bpaxton/mesa_caches/rates_cache'
```

If you give an empty string for the `cache_dir`, then if you have set the environment variable `MESA_CACHES_DIR`, then the cache is a subdirectory of that with one of the following names: `eosDT_cache`, `eosPT_cache`, `kap_cache`, `ionization_cache`, `rates_cache` if `MESA_CACHES_DIR` is not set or is the empty string, then the cache is a subdirectory of the corresponding data subdirectory, such as `data/rates_data/cache` for the rates cache.

```
eosDT_cache_dir = ''
eosPT_cache_dir = ''
eosDE_cache_dir = ''
ionization_cache_dir = ''
kap_cache_dir = ''
rates_cache_dir = ''
```

output ¶

echo_at_start ¶

```
echo_at_start = ''
```

echo_at_end ¶

```
echo_at_end = ''
```

save_star_job_namelist ¶

dumps all values for &star_job controls to file

```
save_star_job_namelist = .false.
```

star_job_namelist_name ¶

if empty, uses a default name

```
star_job_namelist_name = ''
```

show_log_description_at_start ¶

set this false if you want to skip the initial terminal output

```
show_log_description_at_start = .true.
```

show_net_species_info ¶

if true, then output a list of the species in the current net

```
show_net_species_info = .false.
```

show_net_reactions_info ¶

if true, then output information about the reactions in the current net

```
show_net_reactions_info = .false.
```

list_net_reactions ¶

if true, then output a simple list of the reactions in the current net

```
list_net_reactions = .false.
```

show_eqns_and_vars_names ¶

if true, then output a list of the names of the equations and variables

```
show_eqns_and_vars_names = .false.
```

pgstar_flag ¶

if true, activates pgplot output

```
pgstar_flag = .false.
```

disable_pgstar_for_relax ¶

if true, turn off pgstar during relax operations

```
disable_pgstar_for_relax = .false.
```

save_pgstar_files_when_terminate ¶

if true, then when the run terminates, pgstar outputs files for plots that have `file_flag = .true.` independently of the corresponding `file_interval`.

```
save_pgstar_files_when_terminate = .false.
```

history_columns_file ¶

if null string, use default.

```
history_columns_file = ''
```

profile_columns_file ¶

if null string, use default.

```
profile_columns_file = ''
```

save_model_number ¶

at any point during the run, you can save a model for later use

```
save_model_number = -111
```

save_model_when_terminate ¶

save final model when a run terminates

```
save_model_when_terminate = .false.
```

save_model_filename ¶

saved model root filename

```
save_model_filename = 'undefined'
```

save_photo_when_terminate ¶

if true, then save photo for last model before terminate the run

```
save_photo_when_terminate = .true.
```

profile_starting_model ¶

profile_model_number ¶

write profile for a specific model number

```
profile_starting_model = .false.
profile_model_number = -1111
```

write_profile_when_terminate ¶

filename_for_profile_when_terminate ¶

write profile to a given name upon termination

```
write_profile_when_terminate = .false.
filename_for_profile_when_terminate = ''
```

save_pulse_data_for_model_number ¶

save_pulse_data_when_terminate ¶

save_pulse_data_filename ¶

write pulsation info for the model (format given by `s% pulse_info_format`)

```
save_pulse_data_for_model_number = -111
save_pulse_data_when_terminate = .false.
save_pulse_data_filename = 'undefined'
```

save_stella_data_for_model_number ¶

save_stella_data_when_terminate ¶

save_stella_data_filename ¶

stella_min_surf_logRho ¶

stella_min_velocity ¶

stella_skip_inner_dm ¶

stella_num_points ¶

stella_nz_extra ¶

stella_mdot_years_for_wind ¶

stella_mdot_for_wind ¶

stella_v_wind ¶

stella_show_headers ¶

write stella info for the model

the mesa/star routine for creating stella files knows about the following species: h1, he3, he4, c12, n14, o16, ne20, na23, mg24, al27, si28, s32, ar36, ca40, ti44, cr48, cr60, fe52, fe54, fe56, co56, ni56 if your net includes other species, they will be ignored when creating the stella file. on the other hand, any species missing from this list will simply be given 0 abundance. see mesa/stella/README for details.

```
save_stella_data_for_model_number = -111
save_stella_data_when_terminate = .false.
save_stella_data_filename = 'undefined'
stella_num_points = 300
stella_nz_extra = 30
stella_min_surf_logRho = -1d2
stella_min_velocity = 0d0
stella_skip_inner_dm = 0.1
stella_mdot_years_for_wind = 1.2
stella_mdot_for_wind = 0.30
```

```
stella_v_wind = 12
stella_show_headers = .false.
```

internals_num ¶

write internals – for debugging only write if >= 0

```
internals_num = -1
```

report_retries ¶

report_backups ¶

in case you want some extra info about retries or backups

```
report_retries = .false.
report_backups = .false.
```

starting model ¶

By default at the start of a run a zams starting model is loaded, and then the `initial_mass`, `initial_z`, and `initial_y` are adjusted as necessary. However, there are alternatives. you can use a model you saved previously, or you can request the system to create a pre-main-sequence model.

BTW: the system finds the zams file by using the control called `zams_filename` the default zams file is for Z=0.02 and lives in `data/star_data/zams_models`. You can create your own zams file and use it instead – see `test_suite/create_zams`.

load_saved_model ¶

saved_model_name ¶

If `load_saved_model` is true, then use the specified initial model.

```
load_saved_model = .false.
saved_model_name = 'undefined'
```

create_pre_main_sequence_model ¶

If true, the code will create a starting model with uniform composition, a core temperature below 10^6 so no nuclear burning, and uniform contraction for enough luminosity to make it fully convective.

The mass is `initial_mass` from the controls namelist.

if `initial_y` is < 0 in the controls, then code uses `0.24 + 2*initial_z` for `initial_y`.

The h1 mass fraction is set to `1 - (initial_y + initial_z)`. The he3 and he4 mass fractions are set according to `initial_y` with relative amounts set according to the AG89 solar mass fractions (from chem_def).

The metallicity is initial_z from the controls namelist with the metals fractions set according to the GS98 values (from chem_def).

to set the metals fractions, use initial_zfracs (described below)

```
create_pre_main_sequence_model = .false.
```

create_merger_model !!! EXPERIMENTAL ¶

saved_model_for_merger_1 ¶

saved_model_for_merger_2 ¶

If create_merger_model is true, then create a star by merging the two models provided.

```
create_merger_model = .false.
saved_model_for_merger_1 = 'undefined'
saved_model_for_merger_2 = 'undefined'
```

pre_ms_T_c ¶

Initial center temperature (must be below 1d6). If you have initial convergence problems creating a pre-ms model, you might try different values for pre_ms_T_c – that sometimes helps.

```
pre_ms_T_c = 3e5
```

pre_ms_guess_rho_c ¶

Guess for initial center density; set to 0 to let the code pick.

```
pre_ms_guess_rho_c = 0
```

pre_ms_d_log10_P ¶

Suggested spacing in pressure between points; set to 0 to let the code pick.

```
pre_ms_d_log10_P = 0
```

pre_ms_logT_surf_limit ¶

pre_ms_logP_surf_limit ¶

Model contruction is from inside out and stops when reaches either of the following limits.

```
pre_ms_logT_surf_limit = 3.7d0
pre_ms_logP_surf_limit = 3.5d0
```

pre_ms_relax_num_steps ¶

Let pre-ms model settle in for this many steps before changing anything else.

```
pre_ms_relax_num_steps = 100
```

create_initial_model ¶

This is an alternative to `create_pre_main_sequence_model`. If true, creates an adiabatic, contracting model for given mass and radius. Assumes no nuclear burning and constant entropy. Ignores radiation pressure. Uses star controls `initial_y` and `initial_z` to set X, Y, and Z. Uses `initial_zfracs` to set abundances of metals.

Note: if you'd like to do-it-yourself, then you can use `other_build_initial_model`. In that case, in addition to setting `create_initial_model`, also set star controls `use_other_build_initial_model`. Then your `run_star_extras` routine will be called instead of the standard one.

```
create_initial_model = .false.
```

radius_in_cm_for_create_initial_model ¶

mass_in_gm_for_create_initial_model ¶

Radius in cm and mass in grams.

```
radius_in_cm_for_create_initial_model = 0
mass_in_gm_for_create_initial_model = 0
```

center_logP_1st_try_for_create_initial_model ¶

entropy_1st_try_for_create_initial_model ¶

max_tries_for_create_initial_model ¶

abs_e01_tolerance_for_create_initial_model ¶

abs_e02_tolerance_for_create_initial_model ¶

```
center_logP_1st_try_for_create_initial_model = 10.9d0
entropy_1st_try_for_create_initial_model = 11.5d0
max_tries_for_create_initial_model = 100
abs_e01_tolerance_for_create_initial_model = 1d-4
abs_e02_tolerance_for_create_initial_model = 1d-4
```

initial_model_relax_num_steps ¶

Let initial model settle in for this many steps before changing anything else.

```
initial_model_relax_num_steps = 10
```

initial_model_eps ¶

Integration accuracy.

```
initial_model_eps = 0.05d0
```

when to stop ¶

steps_to_take_before_terminate ¶

If > 0, stop after taking this many steps. Sets `max_model_number` = `model_number` + `steps_to_take_before_terminate`. Ignore if <= 0.

```
steps_to_take_before_terminate = -1
```

stop_if_this_file_exists ¶

At each step, the code will try to open this file. If the file exists, it will terminate the run. If the file doesn't exist, it will keep going.

```
stop_if_this_file_exists = ''
```

modifications to model ¶

These controls enable one to alter the MESA model at the start of a run (`./rn`) or after a restart (`./re`). Controls that only apply to the first model have 'initial' in their names, and are ignored for restarts.

set_initial_age ¶

initial_age ¶

if true, set initial age in years

```
set_initial_age = .false.
initial_age = 0
```

set_initial_model_number ¶

initial_model_number ¶

if true, set initial model number

```
set_initial_model_number = .false.
initial_model_number = 0
```

set_initial_dt ¶

years_for_initial_dt ¶

seconds_for_initial_dt ¶

if true, set initial timestep, dt, in years

```
set_initial_dt = .false.
years_for_initial_dt = -1
seconds_for_initial_dt = -1
```

limit_initial_dt ¶

Like `set_initial_dt`, but does not increase current value for `dt_next`. Used in conjunction with `years_for_initial_dt` and `seconds_for_initial_dt`.

```
dt_next = min(dt_next, years_for_initial_dt*secyer)
```

```
limit_initial_dt = .false.
```

set_uniform_initial_composition ¶

Set uniform composition. This is useful with `create_pre_main_sequence_model`.

```
set_uniform_initial_composition = .false.
```

initial_h1 ¶

initial_h2 ¶

initial_he3 ¶

initial_he4 ¶

if `set_uniform_initial_composition` is true, then set hydrogen and helium mass fractions according to the following: If no h2 in current net, then this will be added to h1. If no he3 in current net, then this will be added to he4.

```
initial_h1 = -1
initial_h2 = -1
initial_he3 = -1
initial_he4 = -1
```

initial_zfracs ¶

if `set_uniform_initial_composition` is true, then set metal fractions z fractions – select one of the options defined in `chem/public/chem_def` :

- `AG89_zfracs = 1`
- `GN93_zfracs = 2`
- `GS98_zfracs = 3`
- `L03_zfracs = 4`
- `AGS05_zfracs = 5`
- `AGSS09_zfracs = 6`
- `L09_zfracs = 7`
- `A09_Prz_zfracs = 8`

for example, `initial_zfracs = 3` for `GS98_zfracs` or set `initial_zfracs = 0` to use the special list of z fractions specified in controls (i.e., `z_fraction_li`, `z_fraction_be`, `z_fraction_b`, etc.)

```
initial_zfracs = 3
```

dump_missing_metals_into_heaviest ¶

this controls the treatment metals that are not included in the current net. if this flag is true, then the mass fractions of missing metals are added to the mass fraction of the most massive metal included in the net. if this flag is false, then the mass fractions of the metals in the net are renormalized to make up for the total mass fraction of missing metals.

```
dump_missing_metals_into_heaviest = .true.
```

file_for_uniform_xa ¶

set_uniform_initial_xa_from_file ¶

set_uniform_xa_from_file ¶

an alternative to the above `set_uniform_initial_composition` method. if `set_uniform_initial_xa_from_file` is .true., read list of iso name and mass fraction pairs from file `file_for_uniform_xa` and use them to set uniform composition. E.g., to convert the star to pure fe56, a file with just the following line will work.

```
fe56   1.0
```

```
file_for_uniform_xa = ''
set_uniform_initial_xa_from_file = .false.
set_uniform_xa_from_file = .false.
```

mix_section ¶

mix_initial_section ¶

mix_section_nzlo ¶

mix_section_nzhi ¶

fully mix section of model

```
mix_section = .false.
mix_initial_section = .false.
mix_section_nzlo = -1
mix_section_nzhi = -1
```

mix_envelope_down_to_T ¶

mix_initial_envelope_down_to_T ¶

fully mix envelope from surface down to given temperature

```
mix_envelope_down_to_T = 0
mix_initial_envelope_down_to_T = 0
```

set_abundance ¶

set_initial_abundance ¶

chem_name ¶

new_frac ¶

set_abundance_nzlo ¶

set_abundance_nzhi ¶

given a `chem_name` from `chem_def`, set its abundance to be `new_frac` in a given range of cells, from `set_abundance_nzlo` to `set_abundance_nzhi`

```
set_abundance = .false.
set_initial_abundance = .false.
chem_name = 'he3'
new_frac = 0
```

```
set_abundance_nzlo = -1
set_abundance_nzhi = -1
```

replace_element ¶

replace_initial_element ¶

chem_name1 ¶

chem_name2 ¶

replace_element_nzlo ¶

replace_element_nzhi ¶

replace one iso by another in a given range of cells `chem_name1` and `chem_name2` from `chem_def`

```
replace_element = .false.
replace_initial_element = .false.
chem_name1 = 'he3'
chem_name2 = 'he4'
replace_element_nzlo = -1
replace_element_nzhi = -1
```

relax_initial_composition ¶

num_steps_to_relax_composition ¶

relax_composition_filename ¶

relax composition from current to specified over number of steps. `relax_composition_filename` holds the desired composition profile information file format for relax composition

```
1st line: num_points num_species
then 1 line for for each point where define desired composition
xq xa(1) ... xa(num_species)
xq = fraction of xmstar exterior to the point
where xmstar = mstar - M_center
the interpolation routines require that the xq values which
appear in your file must be monotonically increasing
xa(i) = mass fraction of i'th species
```

NOTE: it is up to you to ensure that the current net isotopes match the species in the composition file. You can set `show_net_species_info = .true.` to check the isotopes in the net.

```
relax_initial_composition = .false.
num_steps_to_relax_composition = 100
relax_composition_filename = ''
```

relax_initial_to_xaccrete ¶

Like `relax_initial_composition` (and uses `num_steps_to_relax_composition`), but new composition is set by current specification of accretion abundances.

```
relax_initial_to_xaccrete = .false.
```

some modifications must be done gradually over several steps in “pseudo” evolution these operations have “relax” in their names. many have an alternative, with “set” in name, that simply make the change all at once. the “set” version is fine if star can manage to converge the modified model. but for larger changes where that’s not possible, you’ll need to “relax” instead.

relax_Y ¶

change_Y ¶

relax_initial_Y ¶

change_initial_Y ¶

relax_Y_minq ¶

relax_Y_maxq ¶

new_Y ¶

`relax_Y = .true.` gradually changes average Y, reconverging at each step. `change_Y = .true.` changes abundances; doesn’t reconverge the model. note: `relax_dY` in the controls inlist determines the rate of change

```
relax_Y = .false.
change_Y = .false.
relax_initial_Y = .false.
change_initial_Y = .false.
relax_Y_minq = 0d0
relax_Y_maxq = 1d0
new_Y = -1
```

relax_Z ¶

change_Z ¶

relax_initial_Z ¶

change_initial_Z ¶

relax_Z_minq ¶

relax_Z_maxq ¶

new_Z ¶

`relax_Z = .true.` gradually changes average Z, reconverging at each step. `change_Z = .true.` simply changes abundances; doesn’t reconverge the model. note: `relax_dlnZ` in the controls inlist determines the rate of change

```
relax_Z = .false.
change_Z = .false.
relax_initial_Z = .false.
change_initial_Z = .false.
relax_Z_minq = 0d0
relax_Z_maxq = 1d0
new_Z = -1
```

relax_mass ¶

relax_initial_mass ¶

new_mass ¶

lg_max_abs_mdot ¶

Gradually change total mass by a wind to `new_mass`. `lg_max_abs_mdot = -4` means max abs mdot 1d-4 msun/year; Set <= -100 to let code pick.

```
relax_mass = .false.
relax_initial_mass = .false.
new_mass = -1
lg_max_abs_mdot = -100
```

relax_mass_scale ¶

relax_initial_mass_scale ¶

dlgm_per_step ¶

change_mass_years_for_dt ¶

Gradually rescale mass of star to `new_mass`. Rescales star mass without changing composition as function of m/mstar.

```
relax_mass_scale = .false.
relax_initial_mass_scale = .false.
dlgm_per_step = 1d-3
change_mass_years_for_dt = 1
```

relax_initial_angular_momentum ¶

max_steps_to_relax_angular_momentum ¶

timescale_for_relax_angular_momentum ¶

max_dt_for_relax_angular_momentum ¶

num_timescales_for_relax_angular_momentum ¶

relax_angular_momentum_filename ¶

relax angular momentum from current to specified over a certain amount of relaxation timescales. This is done by adding an extra torque term of the form

```
s% extra_jdot(k) =  &
    (1d0 - exp_cr(-s% dt/(s% job% timescale_for_relax_angular_momentum
    (desired_angular_momentum(k) - s% j_rot(k))/s% dt
```

and evolving the star without changing the composition for `num_timescales_for_relax_angular_momentum` times `timescale_for_relax_angular_momentum`. To circumvent convection we limit the acceleration of convective velocities using `min_T_for_acceleration_limited_conv_velocity = 0` (see controls.defaults), and the timescale for relaxation should be very short (less than a second).

`relax_angular_momentum_filename` holds the desired angular momentum profile information file format for relax angular momentum

```
1st line: num_points
then 1 line for for each point where define desired angular momentum
xq angular_momentum
xq = fraction of xmstar exterior to the point
where xmstar = mstar - M_center
angular_momentum = specific angular momentum in units of cm^2/s
```

```
relax_initial_angular_momentum = .false.
max_steps_to_relax_angular_momentum = 1000
timescale_for_relax_angular_momentum = 1d-10
max_dt_for_relax_angular_momentum = 1d-9
num_timescales_for_relax_angular_momentum = 1000
relax_angular_momentum_filename = ''
```

relax_initial_entropy ¶

max_steps_to_relax_entropy ¶

timescale_for_relax_entropy ¶

max_dt_for_relax_entropy ¶

num_timescales_for_relax_entropy ¶

relax_entropy_filename ¶

get_entropy_for_relax_from_eos ¶

relax entropy from current to specified over a certain amount of relaxation timescales. This is done by adding an extra heating term of the form

```
s% extra_heat(k) =  &
    (1d0 - exp_cr(s%lnS(k))/desired_entropy(k))*exp_cr(s%lnE(k))/(time
```

and evolving the star without changing the composition for `num_timescales_for_relax_entropy` times `timescale_for_relax_entropy`. To circumvent convection we limit the acceleration of convective velocities using `min_T_for_acceleration_limited_conv_velocity = 0` (see controls.defaults), and the timescale for relaxation should be very short (less than a second).

`relax_entropy_filename` holds the desired entropy profile information file format for relax entropy

```
1st line: num_points
then 1 line for for each point where define desired entropy
xq entropy
xq = fraction of xmstar exterior to the point
where xmstar = mstar - M_center
entropy = specific entropy in units of erg/gr/K
```

the interpolation routines require that the xq values which appear in your file must be monotonically increasing.

In case the entropy is not readily available, pairs of values of two other thermodynamic variables can be provided. The entropy is then computed using the eos module, and the composition of the stellar model (which can be set using `relax_initial_composition`). This is set by the option `get_entropy_for_relax_from_eos which` can take the values

- ’’: if empty, then input file directly specifies the entropy
- ‘eosDT’: input file includes density and temperature
- ‘eosPT’: input file includes gas pressure and temperature
- ‘eosDE’: input file includes density and specific internal energy

when any of the eos* options is used, then each line in the input file must contain three columns instead of two, specifying the values of the two thermodinamic variables used in the order specified above. So, for example, when using ‘eosDT’ the format of the input file is

```
1st line: num_points
then 1 line for for each point where define desired entropy
xq density temperature
xq = fraction of xmstar exterior to the point
where xmstar = mstar - M_center
density and temperature in cgs units
```

```
relax_initial_entropy = .false.
max_steps_to_relax_entropy = 1000
timescale_for_relax_entropy = 1d-9
max_dt_for_relax_entropy = 1d-9
num_timescales_for_relax_entropy = 100
relax_entropy_filename = ''
get_entropy_for_relax_from_eos = ''
```

relax_dxdt_nuc_factor ¶

relax_initial_dxdt_nuc_factor ¶

new_dxdt_nuc_factor ¶

dxdt_nuc_factor_multiplier ¶

Gradually rescale `dxdt_nuc_factor`. At each step, multiply `dxdt_nuc_factor` by `dxdt_nuc_factor_multiplier`, until reach `new_dxdt_nuc_factor`.

```
relax_dxdt_nuc_factor = .false.
relax_initial_dxdt_nuc_factor = .false.
new_dxdt_nuc_factor = 0
dxdt_nuc_factor_multiplier = 0
```

relax_eps_nuc_factor ¶

relax_initial_eps_nuc_factor ¶

new_eps_nuc_factor ¶

eps_nuc_factor_multiplier ¶

Gradually rescale `eps_nuc_factor`. At each step, multiply `eps_nuc_factor` by `eps_nuc_factor_multiplier` until reach `new_eps_nuc_factor`.

```
relax_eps_nuc_factor = .false.
relax_initial_eps_nuc_factor = .false.
new_eps_nuc_factor = 0
eps_nuc_factor_multiplier = 0
```

relax_opacity_max ¶

relax_initial_opacity_max ¶

new_opacity_max ¶

opacity_max_multiplier ¶

Gradually rescale `opacity_max`. At each step, multiply `opacity_max` by `opacity_max_multiplier` until reach `new_opacity_max`.

```
relax_opacity_max = .false.
relax_initial_opacity_max = .false.
new_opacity_max = 0
opacity_max_multiplier = 0
```

relax_fixed_L_for_BB_outer_BC ¶

relax_initial_fixed_L_for_BB_outer_BC ¶

steps_for_relax_fixed_L ¶

Gradually modify `fixed_L_for_BB_outer_BC`.

```
relax_fixed_L_for_BB_outer_BC = .false.
relax_initial_fixed_L_for_BB_outer_BC = .false.
steps_for_relax_fixed_L = 0
```

relax_max_surf_dq ¶

relax_initial_max_surf_dq ¶

new_max_surf_dq ¶

max_surf_dq_multiplier ¶

Gradually rescale `max_surface_cell_dq`. At each step, multiply `max_surface_cell_dq` by `opacity_max_multiplier` until reach `new_max_surf_dq`.

```
relax_max_surf_dq = .false.
relax_initial_max_surf_dq = .false.
new_max_surf_dq = 0
max_surf_dq_multiplier = 0
```

relax_to_this_tau_factor ¶

dlogtau_factor ¶

relax_tau_factor ¶

relax_initial_tau_factor ¶

relax_tau_factor_after_core_He_burn ¶

relax_tau_factor_after_core_C_burn ¶

`relax_to_this_tau_factor = 1` puts outer cell at photosphere; can go much larger or much smaller to move surface in or out from photosphere.

`dlogtau_factor` changes `log10(tau_factor)` by at most this amount per step

`relax_tau_factor` true gradually changes `tau_factor`, reconverging at each step.

`relax_tau_factor_after_core_He_burn` ignored if <= 0; change `tau_factor` when center H1 < 1e-4 and center He4 < `relax_tau_factor_after_core_He_burn`.

`relax_tau_factor_after_core_C_burn` ignored if <= 0; change `tau_factor` when center H1 < 1e-4, He4 < 1e-4, and center C12 < `relax_tau_factor_after_core_C_burn`.

```
relax_to_this_tau_factor = -1
dlogtau_factor = 0.1d0
relax_tau_factor = .false.
relax_initial_tau_factor = .false.
relax_tau_factor_after_core_He_burn = -1
relax_tau_factor_after_core_C_burn = -1
```

set_to_this_tau_factor ¶

set_tau_factor ¶

set_initial_tau_factor ¶

set_tau_factor_after_core_He_burn ¶

set_tau_factor_after_core_C_burn ¶

As for `relax_to_this_tau_factor`, but changes `tau_factor` without reconverging.

```
set_to_this_tau_factor = -1
set_tau_factor = .false.
set_initial_tau_factor = .false.
set_tau_factor_after_core_He_burn = -1
set_tau_factor_after_core_C_burn = -1
```

adjust_tau_factor_to_surf_density ¶

base_for_adjust_tau_factor_to_surf_density ¶

if `adjust_tau_factor_to_surf_density`, then at start of each step set `tau_factor` to current Rho(1) divided by `base_for_adjust_tau_factor_to_surf_density`

```
adjust_tau_factor_to_surf_density = .false.
base_for_adjust_tau_factor_to_surf_density = 0d0
```

relax_to_this_opacity_factor ¶

d_opacity_factor ¶

relax_opacity_factor ¶

relax_initial_opacity_factor ¶

```
relax_to_this_opacity_factor = -1
d_opacity_factor = 0.1d0
relax_opacity_factor = .false.
relax_initial_opacity_factor = .false.
```

relax_to_this_Tsurf_factor ¶

dlogTsurf_factor ¶

relax_Tsurf_factor ¶

relax_initial_Tsurf_factor ¶

```
relax_to_this_Tsurf_factor = -1
dlogTsurf_factor = 0.1d0
relax_Tsurf_factor = .false.
relax_initial_Tsurf_factor = .false.
```

set_to_this_Tsurf_factor ¶

set_Tsurf_factor ¶

set_initial_Tsurf_factor ¶

As for `relax_to_this_Tsurf_factor`, but changes `Tsurf_factor` without reconverging.

```
set_to_this_Tsurf_factor = -1
set_Tsurf_factor = .false.
set_initial_Tsurf_factor = .false.
```

relax_mass_change ¶

relax_initial_mass_change ¶

relax_mass_change_min_steps ¶

relax_mass_change_max_yrs_dt ¶

relax_mass_change_init_mdot ¶

relax_mass_change_final_mdot ¶

`relax_mass_change_max_yrs_dt` in years `relax_mass_change_init_mdot` in Msun/year

```
relax_mass_change = .false.
relax_initial_mass_change = .false.
relax_mass_change_min_steps = 10
relax_mass_change_max_yrs_dt = 10
relax_mass_change_init_mdot = 0
relax_mass_change_final_mdot = 0
```

relax_irradiation ¶

relax_initial_irradiation ¶

relax_to_this_irrad_flux ¶

relax_irradiation_min_steps ¶

relax_irradiation_max_yrs_dt ¶

irrad_col_depth ¶

extra heat near surface to model irradiation. `relax_to_this_irrad_flux` is flux in erg s^-1 cm^-2 from companion. we capture `Pi*R^2` of that flux and distribute it uniformly in the outer `4*Pi*R^2*irrad_col_depth` grams of the star, where `irrad_col_depth` is in g cm^-2.

```
relax_irradiation = .false.
relax_initial_irradiation = .false.
relax_to_this_irrad_flux = 0
relax_irradiation_min_steps = 0
relax_irradiation_max_yrs_dt = -1
irrad_col_depth = -1
```

set_irradiation ¶

set_initial_irradiation ¶

set_to_this_irrad_flux ¶

as for `relax_irradiation` but sets values and does not reconverge

```
set_irradiation = .false.
set_initial_irradiation = .false.
set_to_this_irrad_flux = 0
```

change_lnPgas_flag ¶

change_initial_lnPgas_flag ¶

new_lnPgas_flag ¶

lnPgas variables `lnPgas_flag` is true if we are using lnPgas variables in place of lnd

```
change_lnPgas_flag = .false.
change_initial_lnPgas_flag = .false.
new_lnPgas_flag = .false.
```

change_RTI_flag ¶

change_initial_RTI_flag ¶

new_RTI_flag ¶

RTI variables `RTI_flag` is true if we are doing Rayleigh Taylor Instabilities

```
change_RTI_flag = .false.
change_initial_RTI_flag = .false.
new_RTI_flag = .false.
```

velocity variables ¶

change_v_flag ¶

change_initial_v_flag ¶

new_v_flag ¶

change whether MESA evolves a (radial) velocity variable, v, defined at cell boundaries

```
change_v_flag = .false.
change_initial_v_flag = .false.
new_v_flag = .false.
```

center_ye_limit_for_v_flag ¶

automatically turn on velocities if `center_ye` drops below this limit. this is useful for evolution leading up to core collapse.

```
center_ye_limit_for_v_flag = 0.45d0
```

gamma1_integral_for_v_flag ¶

automatically turn on velocities if `center_gamma1_integral` drops below this limit. this is useful for evolution leading up to pair instability core collapse. integral is sum over all cells of `(gamma1-4d0/3d0)` weighted by `dm*P/rho`

```
gamma1_integral_for_v_flag = 0d0
```

change_u_flag ¶

change_initial_u_flag ¶

new_u_flag ¶

change whether MESA evolves a (radial) velocity variable, u, defined at cell centers. this is an alternative to v at cell boundaries. can use one or the other, but not both.

Experimental – not ready for use.

```
change_u_flag = .false.
change_initial_u_flag = .false.
new_u_flag = .false.
```

rotation controls ¶

new_rotation_flag ¶

change_rotation_flag ¶

change_initial_rotation_flag ¶

rotation is enabled only if `rotation_flag` is true `new_rotation_flag` is only used if `change_rotation_flag` is true if `change_rotation_flag` true, then change `rotation_flag` to `new_rotation_flag`

NOTE: why 2 flags? because I want 3 options: set true, set false, and leave it alone. there are of course other ways to get 3 options, but this is what we have.

```
new_rotation_flag = .false.
change_rotation_flag = .false.
change_initial_rotation_flag = .false.
```

the following only apply when rotation is already on (i.e., when `rotation_flag` is true), including when you have just done `change_rotation_flag` true. all of these initialize the model to uniform omega (i.e. “solid body”)

new_omega ¶

set_omega ¶

set_initial_omega ¶

`new_omega` in rad/sec `set_omega` applies when do ./rn or ./re; if true, sets uniform omega = `new_omega` `set_initial_omega` only applies at start of run, not for restarts if true, sets uniform omega = `new_omega`

```
new_omega = 0
set_omega = .false.
```

```
set_initial_omega = .false.
```

new_omega_div_omega_crit ¶

set_omega_div_omega_crit ¶

set_initial_omega_div_omega_crit ¶

as above, but sets `omega/omega_crit` `omega_crit` is defined as:

```
gamma_factor = 1d0 - min(L_div_Ledd, 0.9999d0)
omega_crit = sqrt(gamma_factor*s% cgrav(k)*s% m_grav(k)/pow3(s% r(k)))
```

```
new_omega_div_omega_crit = 0
set_omega_div_omega_crit = .false.
set_initial_omega_div_omega_crit = .false.
```

new_surface_rotation_v = 0 ! (km sec^1) ¶

set_surface_rotation_v = .false. ¶

set_initial_surface_rotation_v = .false. ¶

as above, but sets surface velocity in km/sec

```
new_surface_rotation_v = 0
set_surface_rotation_v = .false.
set_initial_surface_rotation_v = .false.
```

the previous controls are “one shot” – they set omega once and are done. however you might need to set omega for several models in a row in order to give things a chance to adjust to the change. the following controls let you do that.

set_omega_step_limit ¶

if `model_number` is <= this, then do `set_omega`

```
set_omega_step_limit = -1
```

set_omega_div_omega_crit_step_limit ¶

if `model_number` is <= this, then do `set_omega_div_omega_crit`

```
set_omega_div_omega_crit_step_limit = -1
```

set_surf_rotation_v_step_limit ¶

if `model_number` is <= this, then do `set_surface_rotation_v`

```
set_surf_rotation_v_step_limit = -1
```

set_near_zams_omega_steps ¶

set_near_zams_omega_div_omega_crit_steps ¶

set_near_zams_surface_rotation_v_steps ¶

You might want to start a run at pre-ms but only turn on rotation when near zams rather than force you to stop the run near zams, change the inlist, and restart. The following will turn on rotation automatically. The working definition of “near zams” is `L_nuc_burn_total/L_phot >= Lnuc_div_L_upper_limit` `Lnuc_div_L_upper_limit` is in the controls part of the inlist.

The following apply when rotation is off and model satisfies the “near zams” test. Each turns on rotation and sets a step limit

only applies if > 0

```
set_omega_step_limit = model_number + set_near_zams_omega_steps - 1
```

```
set_near_zams_omega_steps = -1
```

only applies if > 0

```
set_omega_div_omega_crit_step_limit =
    model_number + set_near_zams_omega_div_omega_crit_steps - 1
```

```
set_near_zams_omega_div_omega_crit_steps = -1
```

only applies if > 0

```
set_surf_rotation_v_step_limit =  model_number + set_surf_rotation_v_s
```

```
set_near_zams_surface_rotation_v_steps = -1
```

num_steps_to_relax_rotation ¶

use `num_steps_to_relax_rotation` steps to relax omega to new value

```
num_steps_to_relax_rotation = 100
```

relax_omega_max_yrs_dt ¶

`relax_omega_max_yrs_dt` sets a maximum time step used during the relaxation process < 0 implies MESA chooses the step. Useful number is 1d4 if `num_steps_to_relax_rotation` > ~150

```
relax_omega_max_yrs_dt = -1
```

relax_omega ¶

relax_initial_omega ¶

near_zams_relax_omega ¶

if `relax_omega` true, relax to value of `new_omega`. applies when do ./rn or ./re `relax_initial_omega` only applies at start of run, not for restarts. `near_zams_relax`+omega applies when “near zams”. The working definition of “near zams” is `L_nuc_burn_total/L_phot >= Lnuc_div_L_upper_limit` `Lnuc_div_L_upper_limit` is in the controls part of the inlist.

```
relax_omega = .false.
relax_initial_omega = .false.
near_zams_relax_omega = .false.
```

relax_omega_div_omega_crit ¶

relax_initial_omega_div_omega_crit ¶

near_zams_relax_omega_div_omega_crit ¶

as above for `relax_omega`, but for `omega/omega_crit`

```
relax_omega_div_omega_crit = .false.
relax_initial_omega_div_omega_crit = .false.
near_zams_relax_omega_div_omega_crit = .false.
```

relax_surface_rotation_v ¶

relax_initial_surface_rotation_v ¶

near_zams_relax_initial_surface_rotation_v ¶

as above for `relax_omega`, but for surface speed

```
relax_surface_rotation_v = .false.
relax_initial_surface_rotation_v = .false.
near_zams_relax_initial_surface_rotation_v = .false.
```

new_D_omega_flag ¶

change_D_omega_flag ¶

change_initial_D_omega_flag ¶

```
new_D_omega_flag = .false.
change_D_omega_flag = .false.
change_initial_D_omega_flag = .false.
```

new_am_nu_rot_flag ¶

change_am_nu_rot_flag ¶

change_initial_am_nu_rot_flag ¶

use_D_omega_for_am_nu_rot ¶

if `am_nu_rot_flag` is true, use time and space smoothed `am_nu_rot` like `D_omega` else if `D_omega_flag` and `use_D_omega_for_am_nu_rot`, use `D_omega` for `am_nu_rot` else use `am_nu_rot` from current model with no smoothing

```
new_am_nu_rot_flag = .false.
change_am_nu_rot_flag = .false.
change_initial_am_nu_rot_flag = .false.
use_D_omega_for_am_nu_rot = .true.
```

relax_core ¶

relax_initial_core ¶

new_core_mass ¶

dlg_core_mass_per_step ¶

relax_core_years_for_dt ¶

core_avg_rho ¶

core_avg_eps ¶

controls for nonzero center M (mass), R (radius), L (luminosity) (e.g., to model neutron star envelope or rocky core planet) `new_core_mass` in Msun units. If you have convergence problems, you’ll need to reduce the mass/step `dlg_core_mass_per_step` and timestep `relax_core_years_for_dt` values. `core_avg_rho` in g/cm^3 and `core_avg_eps` in ergs/g/sec are just examples. Adjust them to values appropriate for your application.

```
relax_core = .false.
relax_initial_core = .false.
new_core_mass = 0
dlg_core_mass_per_step = 1d-3
relax_core_years_for_dt = 1
core_avg_rho = 10
core_avg_eps = 1d-6
```

relax_M_center ¶

relax_initial_M_center ¶

relax_M_center_dt ¶

Like `relax_mass_scale`, but all change in mass goes into `M_center`. NOTE: `new_mass` is new total mass for star, not the new `M_center` value. uses `dlgm_per_step` in same way as `relax_mass_scale`. `relax_M_center_dt` in seconds

Example: If you want to end up with total mass = 1.4 and `M_center` = 1.3, start with `star_mass` = total - center = 0.1 = mass exterior to center. Then `relax_M_center` with `new_mass` = 1.4. That will give a new total mass of 1.4 by changing `M_center`. The mass exterior to the center will stay = 0.1, so the final `M_center` will be 1.3.

```
relax_M_center = .false.
relax_initial_M_center = .false.
relax_M_center_dt = 3.1558149984d1
```

relax_R_center ¶

relax_initial_R_center ¶

new_R_center ¶

dlgR_per_step ¶

relax_R_center_dt ¶

as above for the mass, but for the radius. `new_R_center` in cm. `relax_R_center_dt` in seconds.

```
relax_R_center = .false.
relax_initial_R_center = .false.
new_R_center = 0
dlgR_per_step = 3d-3
relax_R_center_dt = 3.1558149984d1
```

zero_alpha_RTI ¶

zero_initial_alpha_RTI ¶

```
zero_alpha_RTI = .false.
zero_initial_alpha_RTI = .false.
```

set_v_center ¶

set_initial_v_center ¶

```
set_v_center = .false.
set_initial_v_center = .false.
```

relax_v_center ¶

relax_initial_v_center ¶

new_v_center ¶

dv_per_step ¶

relax_v_center_dt ¶

`new_v_center` in cm/s. `relax_v_center_dt` in seconds.

```
relax_v_center = .false.
relax_initial_v_center = .false.
new_v_center = 0
dv_per_step = 0
relax_v_center_dt = 0
```

set_L_center ¶

set_initial_L_center ¶

```
set_L_center = .false.
set_initial_L_center = .false.
```

relax_L_center ¶

relax_initial_L_center ¶

new_L_center ¶

dlgL_per_step ¶

relax_L_center_dt ¶

as above for the mass, but for the luminosity. `new_L_center` in Lsun. `relax_L_center_dt` in seconds.

```
relax_L_center = .false.
relax_initial_L_center = .false.
new_L_center = 0
dlgL_per_step = 5d-2
relax_L_center_dt = 3.1558149984d1
```

remove_initial_center_at_cell_k ¶

remove_initial_center_by_temperature ¶

remove_initial_center_by_mass_fraction_q ¶

remove_initial_center_by_delta_mass_gm ¶

remove_initial_center_by_delta_mass_msun ¶

remove_initial_center_by_mass_gm ¶

remove_initial_center_by_mass_msun ¶

remove_initial_center_by_radius_cm ¶

remove_initial_center_by_radius_Rsun ¶

remove_initial_center_by_si28 ¶

remove_initial_center_to_reduce_co56_ni56 ¶

remove_initial_center_by_ye ¶

remove_initial_center_by_entropy ¶

remove_initial_center_by_infall_kms ¶

allows the core to be removed. ignored if <= 0 value for si28 is mass fraction at which to make mass cut i.e. cut at first location going inward where mass fraction of si28 >= this limit value for ye is electron per baryon number for cut value for infall_kms is infall speed in km per sec to make the cut

```
remove_initial_center_at_cell_k = 0
remove_initial_center_by_temperature = 0
remove_initial_center_by_mass_fraction_q = 0
remove_initial_center_by_delta_mass_gm = 0
remove_initial_center_by_delta_mass_Msun = 0
remove_initial_center_by_mass_gm = 0
remove_initial_center_by_mass_Msun = 0
remove_initial_center_by_radius_cm = 0
remove_initial_center_by_radius_Rsun = 0
remove_initial_center_by_si28 = 0
```

```
remove_initial_center_to_reduce_co56_ni56 = 0
remove_initial_center_by_ye = 0
remove_initial_center_by_entropy = 0
remove_initial_center_by_infall_kms = 0
```

remove_center_at_cell_k ¶

remove_center_by_temperature ¶

remove_center_by_mass_fraction_q ¶

remove_center_by_delta_mass_gm ¶

remove_center_by_delta_mass_Msun ¶

remove_center_by_mass_gm ¶

remove_center_by_mass_Msun ¶

remove_center_by_radius_cm ¶

remove_center_by_radius_Rsun ¶

remove_center_by_si28 ¶

remove_center_to_reduce_co56_ni56 ¶

remove_center_by_ye ¶

remove_center_by_entropy ¶

remove_center_by_infall_kms ¶

allows the core to be removed. ignored if <= 0

```
remove_center_at_cell_k = 0
remove_center_by_temperature = 0
remove_center_by_mass_fraction_q = 0
remove_center_by_delta_mass_gm = 0
remove_center_by_delta_mass_Msun = 0
remove_center_by_mass_gm = 0
remove_center_by_mass_Msun = 0
remove_center_by_radius_cm = 0
remove_center_by_radius_Rsun = 0
remove_center_by_si28 = 0
remove_center_to_reduce_co56_ni56 = 0
remove_center_by_ye = 0
remove_center_by_entropy = 0
remove_center_by_infall_kms = 0
```

remove_initial_fe_core ¶

remove_fe_core ¶

```
remove_initial_fe_core = .false.
remove_fe_core = .false.
```

remove_initial_center_at_inner_max_abs_v ¶

remove_center_at_inner_max_abs_v ¶

```
remove_initial_center_at_inner_max_abs_v = .false.
remove_center_at_inner_max_abs_v = .false.
```

remove_fallback_at_each_step ¶

fallback_set_zero_v_center ¶

fallback_check_total_energy ¶

```
remove_fallback_at_each_step = .false.
fallback_set_zero_v_center = .false.
fallback_check_total_energy = .true.
```

retain_fallback_at_each_step ¶

```
retain_fallback_at_each_step = .false.
```

remove_center_adjust_L_center ¶

```
remove_center_adjust_L_center = .true.
```

limit_center_logP_at_each_step ¶

at start of each step remove center cells if necessary to keep logP at innermost cell >= this limit.

```
limit_center_logP_at_each_step = -1d99
```

zero_initial_inner_v_by_mass_msun ¶

zero_inner_v_by_mass_Msun ¶

```
zero_initial_inner_v_by_mass_Msun = 0
zero_inner_v_by_mass_Msun = 0
```

remove_center_logRho_limit ¶

```
remove_center_logRho_limit = -1d99
```

remove_initial_surface_at_cell_k ¶

remove_initial_surface_by_optical_depth ¶

remove_initial_surface_by_temperature ¶

remove_initial_surface_by_density ¶

remove_initial_surface_by_pressure ¶

remove_initial_surface_by_mass_fraction_q ¶

remove_initial_surface_by_mass_gm ¶

remove_initial_surface_by_mass_msun ¶

remove_initial_surface_by_radius_cm ¶

remove_initial_surface_by_radius_Rsun ¶

remove_initial_surface_by_v_surf_km_s ¶

allows the outer envelope to be removed. ignored if <= 0

```
remove_initial_surface_at_cell_k = 0
remove_initial_surface_by_optical_depth = 0
remove_initial_surface_by_temperature = 0
remove_initial_surface_by_density = 0
remove_initial_surface_by_pressure = 0
remove_initial_surface_by_mass_fraction_q = 0
remove_initial_surface_by_mass_gm = 0
remove_initial_surface_by_mass_Msun = 0
remove_initial_surface_by_radius_cm = 0
remove_initial_surface_by_radius_Rsun = 0
remove_initial_surface_by_v_surf_km_s = 0
```

remove_surface_at_cell_k ¶

remove_surface_by_optical_depth ¶

remove_surface_by_temperature ¶

remove_surface_by_density ¶

remove_surface_by_pressure ¶

remove_surface_by_mass_fraction_q ¶

remove_surface_by_mass_gm ¶

remove_surface_by_mass_Msun ¶

remove_surface_by_radius_cm ¶

remove_surface_by_radius_Rsun ¶

remove_surface_by_v_surf_km_s ¶

allows the outer envelope to be removed. ignored if <= 0

```
remove_surface_at_cell_k = 0
remove_surface_by_optical_depth = 0
remove_surface_by_temperature = 0
remove_surface_by_density = 0
remove_surface_by_pressure = 0
remove_surface_by_mass_fraction_q = 0
remove_surface_by_mass_gm = 0
remove_surface_by_mass_Msun = 0
remove_surface_by_radius_cm = 0
remove_surface_by_radius_Rsun = 0
remove_surface_by_v_surf_km_s = 0
```

repeat_remove_surface_for_each_step ¶

if true, then at each step removes surface as specified. e.g., if doing remove at specific density and expansion has lowered the surface density to below the limit, then remove surface mass down to the limit.

```
repeat_remove_surface_for_each_step = .false.
```

report_mass_not_fe56 ¶

reports mass that is not fe56

```
report_mass_not_fe56 = .false.
```

report_cell_for_xm ¶

in grams. if > 0 then write smallest k s.t. mass in cells 1 to k is >= `report_cell_for_xm`

```
report_cell_for_xm = -1
```

set_to_xa_for_accretion ¶

set_initial_to_xa_for_accretion ¶

set_nzlo ¶

set_nzhi ¶

changes the composition to the mass fractions `xa_for_accretion`. useful for creating a model with specific uniform composition. `set_to_xa_for_accretion` true, means do when start or restart. `set_initial_to_xa_for_accretion` true, means do for start but not for restarts. nzlo and nzhi determine the range of cells that will be changed. nzlo < 0 means change out to surface. nzhi < 0 or nzhi > number of cells means change to center.

```
set_to_xa_for_accretion = .false.
set_initial_to_xa_for_accretion = .false.
set_nzlo = -1
set_nzhi = -1
```

set_initial_cumulative_energy_error ¶

initial_cumulative_energy_error ¶

```
set_initial_cumulative_energy_error = .false.
set_cumulative_energy_error = .false.
new_cumulative_energy_error = 0d0
```

eos controls ¶

eos_file_prefix ¶

eosDT_Z1_suffix ¶

eosPT_Z1_suffix ¶

Modify this to select a different set of EoS tables.

```
eos_file_prefix = 'mesa'
eosDT_Z1_suffix = '_CO_1'
eosPT_Z1_suffix = '_CO_1'
```

set_HELM_OPAL_Zs ¶

Z_all_HELM ¶

Z_all_OPAL ¶

Max Z for using OPAL data instead of switching to HELM. If `set_HELM_OPAL_Zs` false, just use the eos default. Use HELM for Z > `Z_all_HELM`. Use OPAL for Z < `Z_all_OPAL`. Blend in the region `Z_all_OPAL < Z < Z_all_HELM`. The highest Z OPAL table in MESA is Z = 0.04; for Z > 0.04, the EOS will simply use the Z = 0.04 table for OPAL data.

```
set_HELM_OPAL_Zs = .false.
Z_all_HELM = 0.06d0
Z_all_OPAL = 0.05d0
```

set_HELM_SCVH_lgTs ¶

logT_low_all_HELM ¶

logT_low_all_SCVH ¶

Transition temperature zone for SCVH to HELM at low T. If `set_HELM_SCVH_lgTs` true, change limits.

- `logT_low_all_HELM` : use HELM for lgT <= this
- `logT_low_all_SCVH` : use SCVH for lgT >= this

```
set_HELM_SCVH_lgTs = .false.
logT_low_all_HELM = 2.2d0
logT_low_all_SCVH = 2.3d0
```

set_HELM_OPAL_lgTs ¶

logT_all_HELM ¶

logT_all_OPAL ¶

Transition temperature zone for OPAL to HELM at high T. If `set_HELM_OPAL_lgTs` true, change limits.

- `logT_all_HELM` : use HELM for lgT >= this
- `logT_all_OPAL` : use OPAL/SCVH for lgT <= this

```
set_HELM_OPAL_lgTs = .false.
logT_all_HELM = 7.7d0
logT_all_OPAL = 7.6d0
```

set_logRho_OPAL_SCVH_limits ¶

logRho1_OPAL_SCVH_limit ¶

logRho2_OPAL_SCVH_limit ¶

logRho_min_OPAL_SCVH_limit ¶

Transition density zones for OPAL/SCVH to HELM/PC. If `set_logRho_OPAL_SCVH_limits` true, change limits.

- `logRho1_OPAL_SCVH_limit` : don't use `OPAL_SCVH` for logRho > this
- `logRho2_OPAL_SCVH_limit` : full `OPAL_SCVH` okay for logRho < this
- `logRho_min_OPAL_SCVH_limit` : full `HELM` for logRho < this

Blend OPAL/SCVH with HELM/PC for logRho between these.

```
set_logRho_OPAL_SCVH_limits = .false.
logRho1_OPAL_SCVH_limit = 2.7d0
logRho2_OPAL_SCVH_limit = 2.5d0
logRho_min_OPAL_SCVH_limit = -14.299d0
```

set_eos_PC_parameters ¶

mass_fraction_limit_for_PC ¶

logRho1_PC_limit ¶

logRho2_PC_limit ¶

log_Gamma_all_HELM ¶

log_Gamma_all_PC ¶

PC_Gamma_start_crystal ¶

PC_Gamma_full_crystal ¶

PC_min_Z ¶

Change HELM PC limits. If `set_eos_PC_parameters` true, change limits.

- `mass_fraction_limit_for_PC` : skips species if abundance < this
- `logRho1_PC_limit` : use pure PC for logRho > this
- `logRho2_PC_limit` : don’t use PC for logRho < this (>= 2.8 or so because of PPT)
- `log_Gamma_all_HELM` : HELM for `log_Gamma` <= this (1.0d0 = log10(10))
- `log_Gamma_all_PC` : PC for `log_Gamma` >= this (1.3010299956d0 = log10(20))
- `PC_Gamma_start_crystal` : start releasing crystallization heat at Gamma > this
- `PC_Gamma_full_crystal` : finish releasing latent heat. Fully solid for Gamma > this
- `PC_min_Z` : don’t use PC for Z < this

Need `PC_Gamma_start_crystal` > `Gamma_lnS_eps_grav_full_on` for latent heat to be included in eps_grav properly.

```
set_eos_PC_parameters = .false.
mass_fraction_limit_for_PC = 1d-3
logRho1_PC_limit = 2.999d0
logRho2_PC_limit = 2.8d0
log_Gamma_all_HELM = 1.0d0
log_Gamma_all_PC = 1.3010299956d0
PC_Gamma_start_crystal = 150d0
PC_Gamma_full_crystal = 175d0
PC_min_Z = 0.999d0
```

set_HELM_ion_neutral_blends ¶

max_logRho_neutral_HELM ¶

logT_neutral_HELM ¶

logT_ion_HELM ¶

HELM does not explicitly include ionization. Therefore these transitions are imposed manually.

```
set_HELM_ion_neutral_blends = .false.
```

Transition between neutral and ionized at some density Fully neutral for `logRho <= max_logRho_neutral_HELM` Fully ionized for `logRho >= max_logRho_neutral_HELM`

```
max_logRho_neutral_HELM = 3d0
```

Transition between neutral and ionized over some temperature range Fully neutral for `logT <= logT_neutral_HELM` Fully ionized for `logT >= logT_ion_HELM`

```
logT_neutral_HELM = 5.0d0
logT_ion_HELM = 5.5d0
```

set_other_HELM_flags ¶

HELM_include_radiation ¶

HELM_always_skip_elec_pos ¶

HELM_always_include_elec_pos ¶

Flags for controlling parts of HELM. If HELM skips electron-positron, it also skips coulomb. Together, this reduces HELM to doing an ideal gas eos. If you’d like to see what happens using an ideal gas for all cases, then set these controls:

```
set_other_HELM_flags = .true.
HELM_always_skip_elec_pos = .true.
HELM_always_include_elec_pos = .false.
HELM_include_radiation = .true. or .false. depending on what you want
set_HELM_OPAL_Zs = .true.
Z_all_HELM = -0.1 ! switch to HELM for Z > this
PC_min_Z = 1.1 ! don't use PC for Z < this
```

```
set_other_HELM_flags = .false.
HELM_include_radiation = .true.
HELM_always_skip_elec_pos = .false.
HELM_always_include_elec_pos = .false.
```

eosDT_use_linear_interp_for_X ¶

```
eosDT_use_linear_interp_for_X = .false.
```

eosDT_use_linear_interp_to_HELM ¶

```
eosDT_use_linear_interp_to_HELM = .false.
```

nuclear reactions ¶

change_net ¶

new_net_name ¶

change_initial_net ¶

For switching reaction networks. `new_net_name` only used if `change_net` if true.

```
change_net = .false.
new_net_name = ''
change_initial_net = .false.
```

adjust_abundances_for_new_isos ¶

If false, new isos initial abundance set to 0.

```
adjust_abundances_for_new_isos = .true.
```

set_rates_preference ¶

new_rates_preference ¶

- 1 = NACRE rates
- 2 = jina reaclib rates

```
set_rates_preference = .false.
new_rates_preference = 2
```

set_rate_c12ag ¶

Empty string means ignore this control. Can be one of:

- ‘NACRE’
- ‘jina reaclib’
- ‘Kunz’
- ‘CF88’

(note: our CF88 is larger than the original by a factor of 1.7)

```
set_rate_c12ag = ''
```

set_rate_n14pg ¶

Empty string means ignore this control. Can be one of

- ‘NACRE’
- ‘jina reaclib’
- ‘CF88’

```
set_rate_n14pg = ''
```

set_rate_3a ¶

Empty string means ignore this control. Can be one of

- ‘NACRE’
- ‘jina reaclib’
- ‘CF88’
- ‘FL87’

FL87 is Fushiki and Lamb, Apj, 317, 368-388, 1987 and includes both strong screening and pyconuclear

```
set_rate_3a = ''
```

set_rate_1212 ¶

Empty string means ignore this control. Can be one of:

- `'CF88_basic_1212'` : the single rate approximation from CF88.
- `'CF88_multi_1212'` : combines the rates for the n, p, and a channels. c12(c12,n)mg23, c12(c12,p)na23, and c12(c12,a)ne20 and uses neutron branching from dayras, switkowski, and woosley, 1976.
- `'G05'` : based on Gasques, et al. Phys Review C, 72, 025806 (2005)

```
set_rate_1212 = ''
```

Users can also provide tabulated rates for any of the reactions. Tabulated rates automatically take priority over any other options for the reaction. e.g., if you provide a rate table for c12ag, those rates will be used if preference to the other options given in `set_rate_c12ag`.

To provide tabulated rates: create a file of (T8, rate) pairs as in `data/rates_data/rate_tables` You can give as many pairs as you want with any spacing in T8. The first uncommented line of the file should be a number giving the total number of (T8, rate) pairs in the subsequent lines. The following lines are your specified values of T8 and rate separated by a single space, one pair per line. Add the filename to `rate_list.txt` along with the name of the rate you want it to govern, either in `data/rates_data/rate_tables` or in a local directory specified with the `rate_tables_dir` control. Be aware that if you choose to put the modified `rate_list.txt` in `data/rates_data/rate_tables` rather than a local directory, your custom tabulated rate will override the rate for that reaction for all future MESA runs.

If the reaction you wish to control does not already have a name that MESA will recognize, you will also need to add it to the file speficed by `net_reaction_filename` (defaults to reactions.list). The default version of this file is located in `data/rates_data`. If you place a modified copy of this file in your work directory, it will take precedence.

num_special_rate_factors ¶

reaction_for_special_factor ¶

special_rate_factor ¶

For using other special rate factors. `num_special_rate_factors` must be <= `max_num_special_rate_factors`.

```
num_special_rate_factors = 0
reaction_for_special_factor(:) = ''
special_rate_factor(:) = 1
```

auto_extend_net ¶

h_he_net ¶

co_net ¶

adv_net ¶

If `auto_extend_net` true, then automatically extend the net as needed from `h_he_net` to `co_net` and then to `adv_net`.

```
auto_extend_net = .true.
h_he_net = 'basic.net'
co_net = 'co_burn.net'
adv_net = 'approx21.net'
```

enable_adaptive_network ¶

min_x_for_keep ¶

min_x_for_n ¶

min_x_for_add ¶

max_Z_for_add ¶

max_N_for_add ¶

max_A_for_add ¶

Heger-style adaptive network (Woosley, Heger, et al, ApJSS, 151:75-102, 2004). If `enable_adaptive_network` is true, then at each step, the system calculates a new set of isos according to the following rules:

```
for each iso in the current net:
  let Z = number of protons in the iso and N = number of neutrons.
  let x = max mass fraction for the iso in any cell in the model.
  if x >= `min_x_for_keep` then include the iso in new net.
  if x >= `min_x_for_n` then include following related isos:
    (Z,N+1)   (Z,N-1)     -- add or remove neutron
  if x >= min_x_for_add then include following related isos:
    (Z+1,N)   (Z-1,N)     -- add or remove proton
    (Z+2,N+2) (Z-2,N-2)   -- add or remove alpha
    (Z+2,N+1) (Z-2,N-1)   -- exchange neutron/alpha
    (Z+1,N+2) (Z-1,N-2)   -- exchange proton/alpha
    (Z+1,N-1) (Z-1,N+1)   -- exchange proton/neutron
    (Z+4,N+4) (Z+3,N+4)   -- extend alpha chain
```

Isos in the previous net can be dropped if they have x < `min_x_for_keep` and no other iso in the previous net causes them to be included in the new net. The new net has the included isos and all relevant reactions. The definition for the new net is saved to a text file in your local “nets” directory. The file name is composed of the model number and the number of species.

```
enable_adaptive_network = .false.
min_x_for_keep = 1d-5
min_x_for_n = 1d-4
min_x_for_add = 1d-4
max_Z_for_add = 999
max_N_for_add = 999
max_A_for_add = 999
```

net_reaction_filename ¶

Looks first in current directory, then in `mesa_data_dir/rates_data`.

```
net_reaction_filename = 'reactions.list'
```

jina_reaclib_filename ¶

Empty string means use current standard version. Else give name of file in directory `mesa/data/rates_data`, e.g., `jina_reaclib_results_20130213default2` (which is an 18.8 MB file of rates data). To use previous version, set to `jina_reaclib_results05301331`.

If you change reaclib version, you should clear the cache after making the change in order to ensure that cached rates from the default reaclib version are not being read. (You can use the script `empty_caches` in `$MESA_DIR`.)

In order to avoid this caching issue, one can also specify a local rates cache directory via the control `rates_cache_dir`.

```
jina_reaclib_filename = ''
```

jina_reaclib_min_T9 ¶

set jina reaclib rates to zero for T9 <= this. if this control is <= 0, then use the standard default from rates. need <= 3d-3 for pre-ms li7 burning if change this, must remove old cached rates from data/rates_data/cache

```
jina_reaclib_min_T9 = -1
```

rate_tables_dir ¶

When MESA looks for the files `rate_list.txt` and `weak_rate_list.txt`, it will look in a local directory with this name first. If doesn’t find one, it will use the one in `data/rates_data/rate_tables`.

```
rate_tables_dir = 'rate_tables'
```

rate_cache_suffix ¶

If this not empty, then use it when creating names for cache files for reaction rates from `rate_tables_dir`. If empty, the suffix will be ‘0’.

```
rate_cache_suffix = ''
```

T9_weaklib_full_off ¶

T9_weaklib_full_on ¶

Weak rates blend weaklib and reaclib according to temperature. These can be used to overwrite the defaults in `mesa/rates/public/rates_def`

- `T9_weaklib_full_off` : use pure reaclib for T <= this (ignore if <= 0)
- `T9_weaklib_full_on` : use pure weaklib for T >= this (ignore if <= 0)

```
T9_weaklib_full_off = 0.01d0
T9_weaklib_full_on = 0.02d0
```

weaklib_blend_hi_Z ¶

Ignore if <= 0. Blend for intermediate temperatures. For high Z elements, switch to reaclib at temp where no longer fully ionized. As rough approximation for this, we switch at Fe to higher values of T9.

```
weaklib_blend_hi_Z = 26
```

T9_weaklib_full_off_hi_Z ¶

T9_weaklib_full_on_hi_Z ¶

If input element has Z >= `weaklib_blend_hi_Z`, then use the following T9 limits:

- `T9_weaklib_full_off_hi_Z` : use pure reaclib for T <= this (ignore if <= 0)
- `T9_weaklib_full_on_hi_Z` : use pure weaklib for T >= this (ignore if <= 0)

```
T9_weaklib_full_off_hi_Z = 0.063d0
T9_weaklib_full_on_hi_Z = 0.073d0
```

use small net for newton iterations only ¶

change_small_net ¶

new_small_net_name ¶

change_initial_small_net ¶

For switching reaction networks for use as small net in newton iterations. small net is only used when also doing split mixing. if `small_net_name` is empty string, then newton uses the standard net rather than the small one. `new_small_net_name` only used if `change_small_net` if true.

```
change_small_net = .false.
new_small_net_name = ''
change_initial_small_net = .false.
```

controls for other weak rate sources ¶

use_suzuki_weak_rates ¶

If this is true, use the A=17-28 weak reaction rates from

```
Suzuki, Toki, and Nomoto (2016)
Electron-capture and $\beta$-decay rates for sd-shell nuclei in stella
http://adsabs.harvard.edu/abs/2016ApJ...817..163S
```

If you make use of these rates, please cite the above paper.

```
use_suzuki_weak_rates = .false.
```

use_special_weak_rates ¶

If this is true, calculate special weak rates using the approach described in Section 8 of Paxton et al. (2015).

```
use_special_weak_rates = .false.
```

special_weak_states_file ¶

File specifiying which states to include

Provide the low-lying energy levels of a given nucleus. These are needed to calcuate the partition function and to indicate which states have allowed transitions. Each isotope should have an entry of the form

```
<name> <nlevels>
<E_1> <J_1>
...
<E_n> <J_n>
```

where E = energy, J = spin.

```
special_weak_states_file = 'special_weak_rates.states'
```

special_weak_transitions_file ¶

File specifying to include

These are the transitions for electron capture / beta decay reactions that should be used.

Each reaction should have and entry of the form

```
<iso1> <iso2> <ntrans>
<si_1> <sf_1> <logft_1>
...
<si_n> <sf_n> <logft_n>
```

where si / sf are the n-th parent / daughter state, counting in the order that you specified in the states file. logft is the comparative half-life of that transition.

```
special_weak_transitions_file = 'special_weak_rates.transitions'
```

ion_coulomb_corrections ¶

select which expression for the ion chemical potential to use to calculate the energy shift associated with changing ion charge

- ‘none’: no corrections
- ‘DGC1973’: Dewitt, Graboske, & Cooper, M. S. 1973, ApJ, 181, 439
- ‘I1993’: Ichimaru, 1993, Reviews of Modern Physics, 65, 255
- ‘PCR2009’: Potekhin, Chabrier, & Rogers, 2009, Phys. Rev. E, 79, 016411

```
ion_coulomb_corrections = 'none'
```

electron_coulomb_corrections ¶

select which expression to use to calculate the shift in the electron chemical potential at the location of the nucleus

- ‘none’: no corrections
- ‘ThomasFermi’: Thomas-Fermi theory
- ‘Itoh2002’: Itoh et al., 2002, ApJ, 579, 380

```
electron_coulomb_corrections = 'none'
```

kap controls ¶

kappa_config_file ¶

Modify this to provide alternative set of X,Z tables For Type1 high- and low-T opacities. This file is a namelist. Leave blank to use the defaults.

```
kappa_config_file = ''
```

kappa_file_prefix ¶

Modify this to select a different set of opacity tables. alternatives:

- `'gn93'`
- `'gs98'`
- `'a09'`
- `'OP_gs98'`
- `'OP_a09'`

```
kappa_file_prefix = 'gs98'
```

kappa_lowT_prefix ¶

For lower temperatures. alternatives:

- `'lowT_Freedman11'`
- `'lowT_fa05_gs98'`
- `'lowT_fa05_gn93'`
- `'lowT_fa05_a09p'`
- `'lowT_af94_gn93'`

```
kappa_lowT_prefix = 'lowT_fa05_gs98'
```

kappa_blend_logT_upper_bdy ¶

<= 0 means use default. Actual upper boundary will be min of this and max logT for lowT tables. it is probably a good idea to keep the blend away from H ionization. logT upper of about 3.9 or a bit less will do

that. older version had default of 4.1 for `kappa_blend_logT_upper_bdy`.

```
kappa_blend_logT_upper_bdy = 3.88d0
```

kappa_blend_logT_lower_bdy ¶

<= 0 means use default Actual lower boundary will be max of this and min logT for highT tables. typical `logT_min` for tables is 3.75. check your tables to be sure. older version had default of 4.0 for `kappa_blend_logT_lower_bdy`.

```
kappa_blend_logT_lower_bdy = 3.80d0
```

kappa_CO_prefix ¶

For C/O enhanced (Type 2 OPAL opacities). alternatives:

- ‘gn93_co’
- ‘gs98_co’
- ‘a09_co’

```
kappa_CO_prefix = 'gs98_co'
```

kappa_min_logRho ¶

> = 100 means use default in mesa/kap otherwise if actual logRho < this, clip and set partial wrt logRho to zero

```
kappa_min_logRho = 1d99
```

ionization controls ¶

ionization_file_prefix ¶

ionization_Z1_suffix ¶

Prefix and suffix of ionization files.

```
ionization_file_prefix = 'ion'
ionization_Z1_suffix = ''
```

"extra" parameters ¶

For use by your `run_star_extras` routines.

extras_lipar ¶

extras_ipar ¶

`extras_lipar` number of integer parameters in `extras_ipar`. Must be <= `max_extras_params` (defined in `run_star_support`)

```
extras_lipar = 0
extras_ipar(:) = 0
```

extras_lrpar ¶

extras_rpar ¶

`extras_lrpar` number of real(dp) parameters in `extras_rpar`. Must be <= `max_extras_params` (defined in `run_star_support`)

```
extras_lrpar = 0
extras_rpar(:) = 0d0
```

extras_lcpar ¶

extras_cpar ¶

`extras_lcpar` number of string parameters in `extras_cpar`. Must be <= `max_extras_params` (defined in `run_star_support`).

```
extras_lcpar = 0
extras_cpar(:) = ''
```

extras_llpar ¶

extras_lpar ¶

`extras_llpar` number of logical parameters in `extras_lpar`. Must be <= `max_extras_params` (defined in `run_star_support`).

```
extras_llpar = 0
extras_lpar(:) = .false.
```

Color Files ¶

color_num_files ¶

color_file_names ¶

color_num_colors ¶

```
Filenames for each bolometric correction (BC) table to load
Must set the number of files to load
Must be <= `max_num_color_files` (defined in `colors_def.f90`).
Must set the number of BC's in each file (May be different for each fi
Must be <= `max_num_bcs_per_file` (defined in `colors_def.f90`).
Files should be structed as:
Teff log_g M_div_h filter1 filter2 ....
where filter1 is the name of the filter (No spaces allowed in name)
Names must be unique accross all files loaded and are case sensitive.
For a filter named filter1 history output will be bc_filter1 for bolom
and abs_mag_filter1 for absolute magnitude
```

```
color_num_files = -1
color_file_names(:) = ''
color_num_colors(:) = -1
```

Default file from Lejeune, Cuisinier, Buser (1998) A&AS 130, 65-75 Can be replaced if need be. The filter names are U B V R I J H K L Lprime M (case senistive)

```
color_num_files = 1
color_file_names(1) = 'lcb98cor.dat'
color_num_colors(1) = 11
```

Set of blackbody bolometric corrections in UBVRI Can be used at the same time as the lcb98cor.dat file Filter names bb_U bb_b bb_V bb_R bb_I color_num_files=2 color_file_names(2)='blackbody_johnson.dat' color_num_colors(2)=5

misc ¶

first_model_for_timing ¶

To get a breakdown of where the time is going set `first_model_for_timing` to determine when the clocks start. At the end of the run, there will be some output to the terminal showing times.

```
first_model_for_timing = -1
```

set_max_dt_to_frac_lifetime ¶

max_frac_of_lifetime_per_step ¶

limit max timestep. If true, set `max_timestep` and `max_years_for_timestep` according to expected lifetime as a function of mass. Use the Iben & Laughlin (1989) formula to estimate lifetime. Multiply that times the value of `max_frac_of_lifetime_per_step` to get `max_timestep`.

```
set_max_dt_to_frac_lifetime = .false.
max_frac_of_lifetime_per_step = -1
```

astero_just_call_my_extras_check_model ¶

Communications flag for astero and star.

```
astero_just_call_my_extras_check_model = .false.
```

include other inlists ¶

You can split your `star_job` inlist into pieces using the following controls. BTW: it works recursively, so the extras can read extras too.

read_extra_star_job_inlist{1..5} ¶

extra_star_job_inlist{1..5}_name ¶

if `read_extra_star_job_inlist{1..5}` is true, then read &star_job from this namelist file

```
read_extra_star_job_inlist1 = .false.
extra_star_job_inlist1_name = 'undefined'
read_extra_star_job_inlist2 = .false.
extra_star_job_inlist2_name = 'undefined'
read_extra_star_job_inlist3 = .false.
extra_star_job_inlist3_name = 'undefined'
read_extra_star_job_inlist4 = .false.
extra_star_job_inlist4_name = 'undefined'
read_extra_star_job_inlist5 = .false.
extra_star_job_inlist5_name = 'undefined'
```

private or experimental ¶

warn_run_star_extras ¶

Due to changing the `run_star_extras` functions to hooks, we break existing `run_star_extras` files. This flag sets a warning message and stops the MESA run until it is set to `.false.`. This way people will hopefully not be confused as to why their `run_star_extras` functions are not being called.

```
warn_run_star_extras = .true.
```