Instructions for the MESA exercises

The exercises are designed to be run using the instructions below, without any prior knowledge of MESA. However, if you want to learn more about MESA, the structure of the code, the physics and the numerical methods are described in Paxton et al. (2011, 2013), and further information and examples on how to use MESA is provided at the website (mesa.sourceforge.net).

Preparations

Log in to one of the “bastet”, “hathor” or “sekhmet” machines (in the computer room) where MESA is installed.

Setup the MESA environment :

```
export MESA_DIR=/usr/lib/mesastar
export PGPLOT_DIR=/usr/lib/pgplot5
```

Make a directory on the scratch partition (note that the data stored here is only guaranteed to be kept until December 24). Download and unpack the exercise package :

```
mkdir /scratch/<hostname>/whatever
cd /scratch/<hostname>/whatever
wget ttt.astro.su.se/~maer0651/lse.tgz
tar xfz lse.tgz
```

The package contains six exercises in the following directories :

M4.0-lse (4 solar mass solar-metallicity star evolved to withe dwarf)
M15.0-lse (15 solar mass solar-metallicity star evolved to core-collapse)
M60.0-lse (60 solar mass solar-metallicity star evolved to core collapse)
M60.0-mdot-0-lse (60 solar mass solar-metallicity star with no massloss evolved to core collapse)
M150.0-lse (150 solar mass low-metallicity star evolved to pair-instability)
M150.0-Z-0.02-lse (150 solar mass solar-metallicity star evolved to core-collapse)

Each directory contains the ”mk”, ”rn” and ”re” scripts, the “inlist-MN.N-lse” and “inlist-MN.N-lse-pgstar” configuration files, and the “png”, “photos” and “LOGS” directories (among others).

Make all exercises using the top-level ”mk” script :

```
./mk
```

Running the excersises

Run the exercise using the ”rn” script :

```
./rn
```

Once the exercise is started three multi-panel windows shows up.

First window : Shows the abundance (upper panel) and nuclear power (lower panel) profiles for the current model.

Second window : Shows temperature versus density for the current model (upper panel), and the temperature, pressure, density and radius profiles for the current model (lower panel). The upper panel also shows the nuclear burning and electron degeneracy regions and, in the case of the 150 solar mass model, the pair production instability region where the adiabatic index decreases below 4/3.

Third window : Shows the evolution of effective temperature versus luminosity (HR-diagram; upper left panel), central temperature versus central density (upper right panel), neutrino luminosity, radiation luminosity, mass-loss rate and total mass (lower left panel). The lower right panel shows basic information about the current model (e.g. model number, age, mass) .

Once the run is finished all plots are stored in the “png” directory. Note that the profile plots are stored only for the last model.

The physical and numerical configuration is contained in the “inlist-MN.N-lse” file, and the logging and visualisation configuration in the “inlist-MN.N-lse-pgstar” file. Note that the visualisation configuration can be changed in real-time.

The run can be stopped by CRTL-C and then restarted with the ”re” script from a dump of the physical state in the ”photos” directory. These are stored once every thousand model in a file named “N000”, and once every hundred model (this is configurable) in a file called “xNNN”. To re-run model N000 :

```
./re N000
```

Loading and plotting the data

During the run data is logged to file at intervals specified in the “inlist-MN.N-lse-pgstar” file. The data is stored in line and space separated text-files in the “LOGS” directory, and which quantities to log is configured in the “history_columns.list” and “profile_columns.list” files. Here we describe how to load and plot the data using python.

Start python, load the numpy and pylab modules and turn on interactive plotting :

```
from numpy import *
from pylab import *
ion()
```

The evolution of the physical quantities are stored in the “history.data” file. To load the data from this file do the following :

```
history=genfromtxt(“LOGS/history.data”, skip_header=5, names=True)
```

The data is now stored in the structured numpy array “history”. To display the field names type “history.dtype.names”. To plot e.g. a HR-diagram do :

```
figure()
plot(history[“log_Teff”],history[“log_L”])
```

The radial profiles of the physical quantities are stored in the “profileN.data” files, where N is a sequence number and the mapping to model number is given in the “profiles.index” file. To load the data from the “profileN.data” files do the following :

```
profile=genfromtxt(“LOGS/profileN.data”, skip_header=5, names=True)
```

The data is now stored in the structured numpy array “profile”. To display the field names type “profile.dtype.names”. To plot e.g. a density profile do :

```
figure()
plot(profile[“mass”],profile[“logRho”])
```

Finally, if you take some time to learn about MESA (see above), you may play around with the physics, visualisation and logging by changing the configuration parameters.