Minilab 2: Explore the model Profile in Detail!

1 Using pgstar to gain physical insight

Often it is useful to compare `MESA` to analytic expectations. This is often done in post-processing, but can also be done real-time using `pgstar`!

We want to check if our stars are well-described by a polytrope, $P \propto \rho^\gamma$ where γ is the polytropic index. To do this, we want to stare at 2 pgstar plots: the T-rho profile that comes on by default, and a profile plot of P versus ρ.

In the `&pgstar` section of `inlist_pgstar`, set

```
Profile_Panels1_win_flag = .true.
```

By default the Profile panels x-axis is `'mass'` and the y-axis of the first row is `'logT'`. In your `inlist_pgstar`, change that by adding

```
Profile_Panels1_xaxis_name = 'logRho'
Profile_Panels1_yaxis_name(1) = 'logP'
```

and, to display only one panel, set

```
Profile_Panels1_num_panels = 1
Profile_Panels1_other_yaxis_name(1) = ''
```

Also, set your x- and y- axes:

```
Profile_Panels1_xmin = -12
Profile_Panels1_ymin(1) = 0
Profile_Panels1_xmax = 5
Profile_Panels1_ymax(1) = 20
```

Finally, we want to add a line of constant slope on the P-ρ profile to compare the `MESA` model with a possible polytrope. We can do this using the pgstar decorator in `run_star_extras`.

In order to get `MESA` to call the `my_Profile_Panels1_pgstar_decorator` subroutine at runtime, add the following line in the `&pgstar` section of `inlist_pgstar`:

```
Profile_Panels1_use_decorator = .true.
```

Within the subroutine `extras_controls` in `src/run_star_extras.f90`, add the line:

```
s% Profile_Panels1_pgstar_decorator => my_Profile_Panels1_pgstar_decorator
```

We now want to add a subroutine, which we will call `my_Profile_Panels1_pgstar_decorator`, in order to decorate the `Profile_Panels1` window with lines of constant slope in log P vs log ρ. This will allow you to visually compare your `MESA` model with polytropes of different γ's. You can find an example using `pgstar_decorator` in
`$MESA_DIR/star/other/pgstar_decorator.f90`.

Copy the commented out example subroutine from `pgstar_decorator.f90` into your `run_star_extras` and rename it from `Abundance_Pgstar_decorator` to `my_Profile_Panels1_pgstar_decorator`.

To save yourself an error, add

```
integer, intent(in) :: plot_num
```

---

to the top of the subroutine. For some reason, in the example, it's passed in but never declared!

The existing subroutine draws four boxes inside a do loop:

```
!call pgline(5,(/xcenter-a*dx,xcenter-a*dx,xcenter+a*dx,xcenter+a*dx,xcenter-a*dx/),&
!(/ycenter-a*dy,ycenter+a*dy,ycenter+a*dy,ycenter-a*dy,ycenter-a*dy/))
```

The number 5 is the number of points, and the enclosed values, `(/value,value,.../)`, are the x- and y- coordinates.

Modify the routine so that instead of drawing 4 boxes, it adds 2 lines showing $P \propto \rho^{4/3}$ and $P \propto \rho^{5/3}$. Draw lines from $\log \rho = -12$ to $\log \rho = 5$, and from $\log P = 0$ to higher pressures determined by $\log(P) = \gamma \log(\rho)$+Constant, with slopes given by γ =4/3 and 5/3. For information about pgline, see https://sites.astro.caltech.edu/~tjp/pgplot/subroutines.html#PGLINE. For valuable `MESA` bonus points, see if you can change the colors of the decorated lines.

To label your lines, follow the example line

```
call pgptxt(xcenter,ycenter, 0.0, 1.0,'Some added text on this plot')
```

but replace the suggested text with the string `'P/\gr\u4/3\d=const.'` (or `5/3`), and choose appropriate values to replace `xcenter` and `ycenter` (which give the x- and y- position of the right end of your string).

When you're ready to run, compile your `run_star_extras` (`./mk`), and run the model (`./rn`)! Your `Profile_panels1` should initially look similar to the figure on the next page.

Stare at the $T - \rho$ profile (generated by default). The grey dashed line labelled as $P_{\rm rad} = P_{\rm gas}$ has a slope of $\rho \propto T^3$, which is the same as a $P \propto \rho^{4/3}$ polytrope of the Eddington standard model. How does the slope compare to your expectations? Similarly, stare at your decorated `Profile_Panel1` plot.

At ZAMS, does your star better resemble a 4/3 or 5/3 polytrope?

If you're pressed for time, or need hints, a full version of the subroutine is given in the Dropbox.

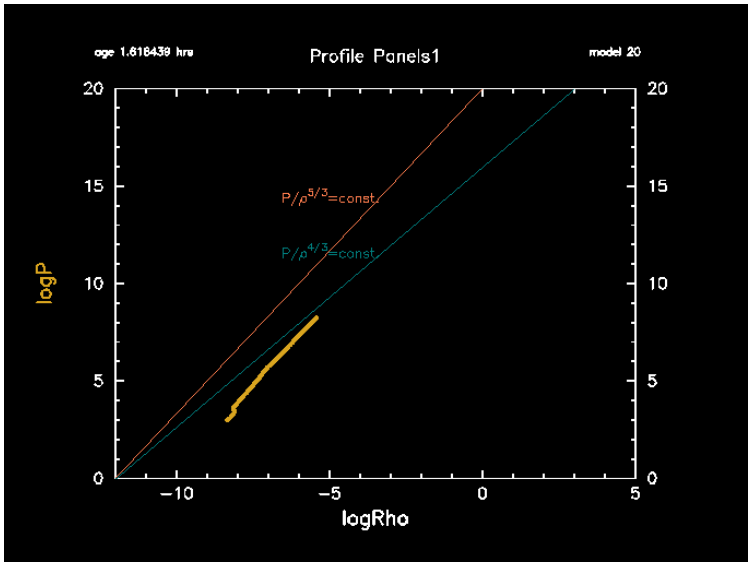

Figure 1: Example of a decorated `Profile_Panels1` plot generated by `pgstar`.

1.1 Troubleshooting notes & tips

If `(/ value, value /)` syntax causes your compiler to break, try without the backslashes, such as `( value, value )`. Two TAs (both using linux) verified that `(/value,value/)` works as in the example (and taking out the / causes compilation failure), but one TA (using a mac) needed to take out the backslashes. We have not identified the cause.

Sometimes pgplot has trouble with double precision floats, so when declaring variables, it may be better here to define them as single-precision (but not in general), e.g.:

```
real :: rhomin, rhomax, pmin, pmax1, pmax2, slope1, slope2
```

For label placement, we recommend setting `xcenter` and `ycenter` (or whatever variable you chose) to be equal to the midpoint of each line, e.g.

```
xcenter = (rhomax + rhomin)/2
ycenter = (pmax + pmin)/2
```

The line `call pgsci(clr_Coral)` calls the color Coral for every object below it, until a new color is called. Additional colors can be looked up with a grep search for `'clr_'` in `$MESA_DIR/star/private`, and will ultimately point you to a list of colors in `$MESA_DIR/star/private/pgstar_support.f90`. If you have time on your hands, a viable but unnecessary method is guessing color names until one of them works. You may be surprised at what works and what doesn't.