Gravity Darkening Corrections for 1-D Stellar Evolution Models

Aaron Dotter

July 13, 2017

Abstract

Rotating stars are subject to *gravity darkening*, in which the flux at the stellar surface is proportional to the local surface gravity and, thus, a rotating star appears both hotter and brighter at the poles than at the equator. This document describes a method to derive gravity darkening corrections that can be applied to stellar evolution models; these adjust the intrinsic model $T_{\rm eff}$, surface gravity, and luminosity in order to account for the effects of surface rotation as well as the angle between the stellar rotation axis and the line of sight. I provide a simple, fast Python implementation to apply these corrections.

1 The gravity darkening model

The gravity darkening model is derived by Espinosa Lara & Rieutord (2011, A&A, 533, 43; hereafter ELR). ELR assume the radiative flux is directed antiparallel to the effective surface gravity. The effective surface gravity is not uniform in a rotating star and, thus, neither is the flux. Since the scalar flux (F) is related to $T_{\rm eff}$ by the Stefan-Boltzmann law, the same argument applies to both F and $T_{\rm eff}$. The ELR model reduces to a differential equation characterized by two variables: the polar angle θ and the ratio of surface angular velocity to the Keplerian angular velocity (ELR eq. 10):

$$\omega = \Omega\sqrt{\frac{R_e}{GM}} = \frac{\Omega}{\Omega_K} \tag{1}$$

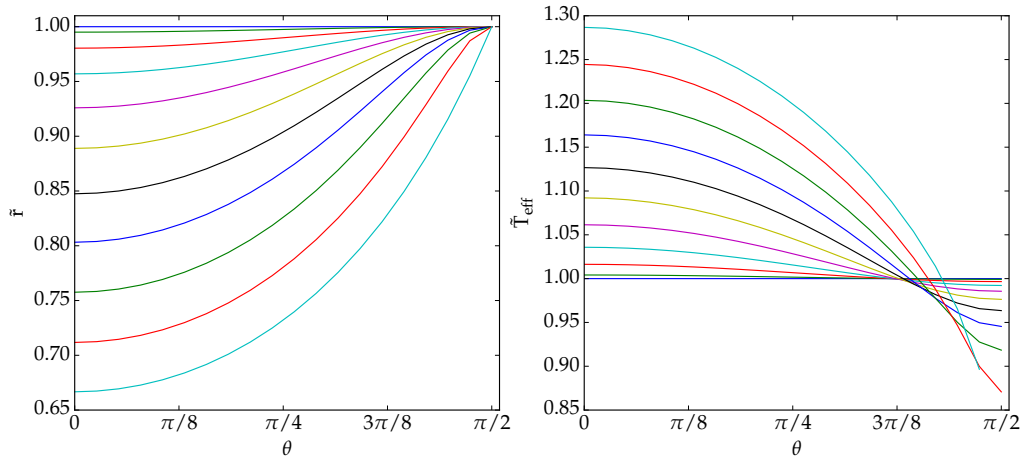

Figure 1: *Left*: the ratio of stellar radius to the equatorial radius R_e as a function of the polar angle θ for different values of ω. *Right*: Similar to the left panel but now showing the scaled $\tilde{\mathrm{T}}_{\mathrm{eff}}$ variable, see text for details.

where Ω is the angular velocity at the surface (radians per second, assumed uniform), R_e is the equatorial radius of the star, G is Newton's constant, and M is stellar mass. Ω_K is the Keplerian angular velocity. Note that ω is provided by `MESA` as '`surf_avg_omega_div_omega_crit`'.

The ELR model is valid for $0 \leq \theta \leq \pi/2$; all other values of θ are mapped into this interval via symmetry arguments. The model requires a numerical solution, except at the extrema $\theta = 0$ (ELR eq. 27) and $\pi/2$ (ELR eq. 28) for which analytical solutions are provided. The numerical solution boils down to finding the value of $\tilde{r}$ $(= r/R_e)$ that satisfies (ELR eq. 30)

$$\frac{1}{\omega^2 \tilde{\mathrm{r}}} + \frac{1}{2}\tilde{\mathrm{r}}^2 \sin^2(\theta) = \frac{1}{\omega^2} + \frac{1}{2} \tag{2}$$

for given ω and θ. The values of $\tilde{\mathrm{r}}$, θ, and ω are then used to solve for the modified angle variable ϑ (ELR eq. 24)

$$\cos(\vartheta) + \ln\tan(\vartheta/2) = \frac{1}{3}\omega^2 \tilde{\mathrm{r}}^3 \cos^3(\theta) + \cos(\theta) + \ln\tan(\theta/2) \tag{3}$$

These are both easily solved with a root-find and, at this point, it is straightforward to calculate the scaled flux and $\mathrm{T}_{\mathrm{eff}}$ via ELR eq. 31.

Figure 1 shows the solution of the ELR model for $0 \leq \omega \leq 1$ in steps of 0.1 as a function of θ. The results show that the polar radius R_p range from R_e at $\omega = 0$ to 2/3 R_e at $\omega = 1$. The scaled temperature shown in Figure 1 is $\tilde{\mathrm{T}}_{\mathrm{eff}} = \mathrm{T}_{\mathrm{eff}} \left(\frac{\mathrm{L}}{4\pi\sigma \mathrm{R_e}^2}\right)^{-1/4}$. We can likewise define a scaled flux $\tilde{\mathrm{F}} = \mathrm{F} \left(\frac{\mathrm{L}}{4\pi \mathrm{R_e}^2}\right)^{-1}$. Azimuthal symmetry (no ϕ-dependence) makes it possible to compute $\tilde{\mathrm{F}}$ and $\tilde{\mathrm{T}}_{\mathrm{eff}}$ over the entire stellar surface with only ω and θ as inputs.

2 Oblate spheroids and projection effects

Now that we have a solution for the scaled flux ($\tilde{\mathrm{F}}$) we can proceed to compute the projected luminosity along the line of sight (LOS). The surface is an oblate spheroid and so we work in oblate spheroidal coordinates. The coordinate system is defined by three variables: polar angle ν, azimuthal angle ϕ, and $\mu = \mathrm{arctanh}(R_p/R_e)$.[1] It is important to distinguish here between ν in the oblate spheroidal coordinate system and the polar angle θ in the ELR model: $\nu = 0$ at the equator; $\nu = \pi/2$ at the pole where $\theta = 0$ and, hence, $\theta = \pi/2 - \nu$.

With the oblate spheroidal coordinate system thus defined, and properly connected to the ELR model, we can now integrate over the stellar surface Σ. However, the quantity that we want is that *projected along the LOS* and for that we need to define one additional angle i and the LOS unit vector $\hat{l}(i)$. i is chosen such that $i = 0$ at the equator. The *projected surface* is denoted by Σ_p.

A useful analytical formula to calculate Σ_p is given by Binngeli (1980). Note that Brandt & Huang (2015) use a different formula, which is only (trivially) correct at $i = 0$ and $\pi/2$. The Brandt & Huang formula leads to a maximum error of $\sim 8\%$ at $i = \pi/4$.

To calculate the projected luminosity $\mathrm{L_p}$ of a star requires the surface integral

$$\mathrm{L_p} = 4 \iint_{d\Sigma \cdot l > 0} F d\vec{\Sigma} \cdot \hat{l}(i) \tag{4}$$

where only terms projected toward the observer, i.e., with $d\vec{\Sigma} \cdot \hat{l}(i) > 0$, are kept. The factor of 4 arises because of the directional aspect of this integral

[1] `https://en.wikipedia.org/wiki/Oblate_spheroidal_coordinates`

and the relationship between astrophysical flux and specific intensity. **(Need some rigorous theory here!)** Once $\mathrm{L_p}$ and Σ_p are known, the projected effective temperature, $\mathrm{T_{eff,p}}$, is obtained from the Stefan-Boltzmann law,

$$\mathrm{T_{eff,p}} = \left(\frac{\mathrm{L_p}}{\sigma \Sigma_p} \right)^{1/4} \tag{5}$$

where σ is the Stefan-Boltzmann constant.

The following is based closely on what is done in the SYCLIST code (Georgy *et al* 2014, A&A, 566, 21). The double integral (4) is, essentially, a geometric factor that does not depend on the stellar temperature or luminosity. It only depends upon ω and i. Thus, it makes sense to derive correction factors which allow the projected quantities ($\mathrm{L_p}$, $\mathrm{T_{eff,p}}$) to be determined from the model intrinsic quantities (L, $\mathrm{T_{eff}}$) by a simple scaling factors, i.e.,

$$\mathrm{L_p} = C_L \mathrm{L} \tag{6}$$

$$\mathrm{T_{eff,p}} = C_T \mathrm{T_{eff}} \tag{7}$$

Defined in this way, the geometric factors C_T and C_L can be expressed as

$$C_L = \frac{4 \iint_{d\Sigma \cdot l > 0} F d\vec{\Sigma} \cdot \hat{l}(i)}{\iint_{\Sigma} F d\Sigma} \tag{8}$$

and

$$C_T = \left(\frac{C_L \Sigma}{4 \Sigma_p} \right)^{1/4} \tag{9}$$

Figure 2 shows how the values of C_L and C_T vary over the ω,i-plane. This can be compared with the upper panels of Georgy *et al* (2014). One can readily see that the effect of C_L is greater than that of C_T, which makes sense because of the $\frac{1}{4}$-power relationship between the two. $\mathrm{L_p}$ can vary by as much as -25 ($i = 0$) to $+50\%$ ($i = \pi/2$) at $\omega = 1$ while $\mathrm{T_{eff,p}}$ varies by $\pm 4\%$. When the LOS is at roughly $i = 34^\circ$ (0.6 radian) above the equatorial plane of the rotating star, both C_T, $C_L \approx 1$ for all ω.

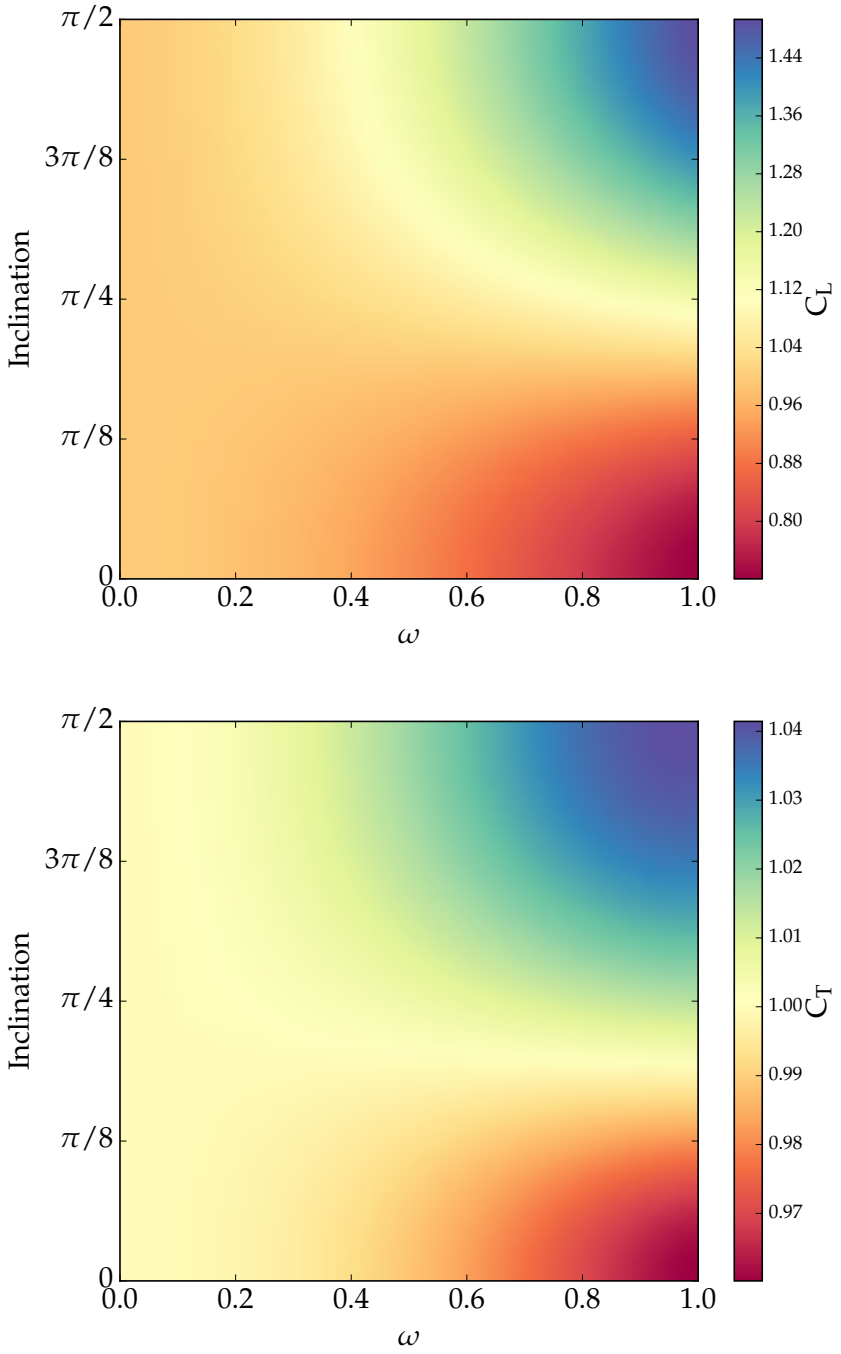

Figure 2: Variation of C_L and C_T over the ω,i-plane.

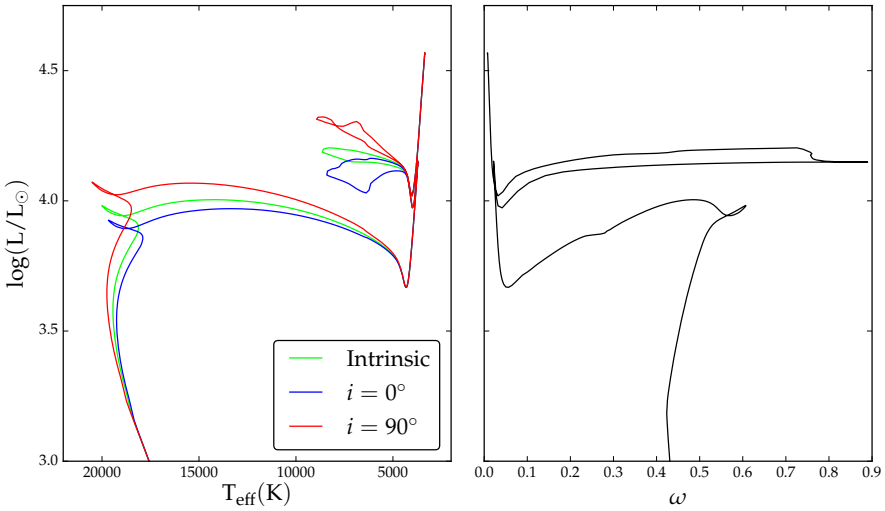

Figure 3: *Left*: MIST isochrone with [Fe/H]=0, log(Age[yr])=7.5, and ZAMS $v/v_{crit} = 0.4$ in the H-R diagram. The green line shows the intrinsic model values. The red line shows the gravity-darkened, projected values of L_p and $T_{eff,p}$ observed "face-on" ($i = 90°$). The blue line shows the projected values observed "edge-on" ($i = 0°$). *Right*: The variation of ω vs. luminosity for the same isochrone.

3 Python code for C_T and C_L

For fast access to C_L and C_T at arbitrary $\{\omega, i\}$ I have tabulated them over a 50×50 grid of $0 \leq \omega \leq 1$ and $0 \leq i \leq \pi/2$. The provided Python code reads in the tabulated data and returns separate instances of the `RectBivariateSpline` class[2] for C_L and C_T. These can subsequently be used to interpolate individual values or arrays of values. An example of the interpolated values applied to a MIST isochrone is demonstrated in Figure 3.

[2] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.RectBivariateSpline.html`