# An exploration into the relative absence of magnetic massive binary stars: modelling the extraordinary system HD156324 using MESA

by Tom Wind
Student nr. 11413875

Report Bachelor Project Physics and Astronomy, size 15 EC
Conducted between 01-04-2020 and 31-07-2020
Submitted on 31-07-2020
Faculty of Science, University of Amsterdam

Massive stars group, API

*Supervisor:*
dr. Alex de Koter
*Daily Supervisor:*
dr Zsolt Keszthelyi

*Second Examiner:*
dr. Lex Kaper

ANTON PANNEKOEK
INSTITUTE

## SUMMARY

Non magnetic massive stars are overwhelmingly found in binary systems. However, for massive magnetic stars, this is not the case. In this research an attempt was made to find factors that could lead magnetic binary systems to be more rare. This was done via modelling of a specific magnetic binary system, HD156324. No conclusive evidence for factors leading to instabilities was found, but there was evidence found hinting to magnetic stars forming through merger, which would cause magnetic binaries to be more rare as they would have to consume a companion in order to form.



## POPULAR SUMMARY (DUTCH)

Sterren zijn de bronnen van Energie voor het universum. Onze ster, de zon, maakt leven op aarde mogelijk. Om het universum te begrijpen, is het bestuderen van sterren erg belangrijk. Sterren worden onderverdeeld in drie categorieën: licht, gemiddeld, en massief. Dit onderzoek focust zich op de massieve sterren. Massieve sterren worden vooral gevonden in dubbelsterren. Een dubbelster is een systeem waar twee of meer sterren door zwaartekracht aan elkaar gebonden zijn. Echter, voor massieve sterren met een magneetveld, blijkt dit niet het geval.

Dit project is op zoek naar factoren die ervoor kunnen zorgen dat magnetische dubbelsterren veel zeldzamer zijn dan gewone dubbelsterren. De methode daarvoor was het modelleren van een specifieke magnetische dubbelster, genaamd HD156324, met het modelleer programma MESA. Uit de modellen blijkt dat er geen duidelijke factor is die ervoor zorgt dat magnetische sterren geen companion kunnen hebben, maar geeft wel een hint voor een oorzaak waardoor ze zeldzamer zouden zijn: de magnetische ster in het HD156324 dubbelsysteem is mogelijk het resultaat van een fusie van twee sterren. Hierdoor is de magnetische ster jonger dan zijn companion.

Als het inderdaad zo is dat magnetische sterren voornamelijk vormen door fusie, betekent dit dat die magnetische sterren die door een fusie magnetisch zijn hervormd, initieel weldegelijk onderdeel van een dubbelster waren, maar dat nu niet meer zijn omdat ze hun companion hebben opgeslokt.

## ACKNOWLEDGEMENTS

I would like to thank Zsolt Keszthelyi and Alex de Koter for allowing me to work on this project, and for the time and help he has given me over the course of this project. I would like to thank Niels Ruijter for allowing me to use his template, especially the title page. I would like to thank my friends Felix Kram, Jelle van de Kerk and Wouter Bouma for proofreading this work, despite their lack of a background in astronomy. I would like to thank Dani van Enk for helping me get started with my project, and for the discussions we had on the topic. Some of the data used in this project was original data by Matt Schultz. This work has made use of the Dutch national e-infrastructure with support of the SURF Cooperative.

### CONTENTS

## 1 INTRODUCTION

Stars are the energy source of the universe. It is because of our star, the sun, that life on earth exists. This simple fact of life makes studying the stars in our galaxy tremendously important. Generally speaking, there are three types of stars, generally divided into categories based on their mass. Mass, however, is not the important factor that leads us to divide stars into groups. It is the evolution of a star over its lifespan that is important, and this happens to be mainly dependent on the mass of a star. The three types of stars are: small-sized Red Dwarf stars, medium-sized sun-like stars and large-sized Massive stars. It is the massive stars that this research will focus on.

### 1.1 Fossil magnetic fields

Unlike the magnetic field of our sun, which is generated by conductive plasma, magnetic fields in massive stars are speculated to be the result of magnetic flux conservation (Braithwaite & Nordlund 2006; Neiner et al. 2015). Flux conservation is given by:

$$B_{\mathrm{P}}(t)R(t)^2 \propto B_{\mathrm{P}}(t=0)R(t=0)^2. \tag{1}$$

The magnetic field in a massive star is thought not to be able to penetrate the outer layers of the star, meaning that any detected magnetic field has to be of a different origin. These are thought to be fossil fields, left intact from the formation of the star due to conservation of flux. These fossil fields could also be the result of stellar merger, where two stars merge to form a new star. Around 7% of stars like this are found to be magnetic (Morel et al. 2015; Fossati et al. 2015; Wade et al. 2016; Grunhut et al. 2017; Shultz et al. 2018b). The evolution and mass loss are affected by rotation. The rotation acts as a centrifugal force pushing the layers outwards, changing the hydrostatic equilibrium of the star. The star thus expands, making the outer layers even less gravitationally bound, resulting in more mass loss. The magnetic field interact with the star in two important ways. Some of the mass ejected in the wind gets funneled back onto the star through the magnetic field. This phenomenon is called mass loss quenching. Furthermore, the magnetic field slows the stellar rotation (Ud-Doula et al. 2009; Meynet et al. 2011; Keszthelyi et al. 2019, 2020). This phenomenon is called magnetic braking.

### 1.2 Stellar Wind

Magnetic stars differ from non-magnetic stars. To understand how, it is important to understand mass loss and how it affects a star. Normally, over their lifetime, massive stars lose a portion of their mass (Puls et al. 2008). The outer layers of the star feel less gravitational attraction, and are slowly getting scattered by the stellar radiation. This phenomenon is called the stellar wind. The amount of mass loss as a percentage of the initial mass grows with the mass of a star.

### 1.3 Roche Lobe overflow

In a binary system, there is a radius around each star where their gravitational forces are stronger than that of their counterpart. As a star evolves, it grows larger, eventually filling this radius entirely. When this happens, mass happens transfers from the expanded star onto the other. This is called Roche Lobe overflow.

### 1.4 Binary stars

A binary system is a star system where two stars are gravitationally bound to each other. A multiple system is essentially the same, but with more than two stars being gravitationally bound. Most massive stars are part of binary or multiple systems. If the magnetic field has no impact on the formation of these stars, one would expect this to be true for magnetic stars as well. This, however, is not the case (Sana et al. 2014). Currently, there are many more magnetic single stars known than magnetic binary stars(Alecian et al. 2013, 2014). This raises the central question of this research:

- Why are there fewer magnetic binary stars than there are magnetic single stars?

So far, there are three plausible reasons for this:

(i) The majority of magnetic stars form from mergers (Ferrario & Wickramasinghe 2008; Schneider et al. 2016). This would imply that while most of these massive stars form in binaries, the ones that retain their fossil field need to have formed in what is at least a stable triple system. This would greatly reduce the amount of magnetic binaries compared to magnetic single stars.

(ii) There is some factor that leads magnetic binary systems to be unstable or that makes it harder for magnetic stars to form in binary systems.

(iii) There are more magnetic binary stars than magnetic single stars, but it just so happened that so far more single stars have been found.

As the amount of magnetic stars in general that have been discovered is very small, the third option is not incredibly unlikely. If this is the case, it will be resolved in time. The first two are much more interesting. This research will mainly focus on option two, making the following question the research question of this work:

- What are possible factors that could lead magnetic binary systems to be rare relative to the amount of magnetic single stars?

If magnetic stars have some characteristics that make it harder to form in binary stars, this would greatly set them apart .

## 2 THE BIG PICTURE

As stated in the introduction, there are far fewer known magnetic binary systems than magnetic single star systems. A dataset of most known magnetic stars was taken from Shultz et al. (2018b). Most data used in this section comes from this paper, although some data has since been superseded, in which case the superseded data was used. To find differences in trends for characteristics of single and binary stars, a cross-correlation between the single and binary stars was made. Figures 1, 2 and 3 are essentially the same plots as figures 5 and 6 from Keszthelyi et al. (2020). The difference is that in figures 1, 2 and 3 the stars are differentiated by binarity, while Zsolt et al. differentiates based on stellar mass.

### 2.1 Rotation

Some stars form with high rotational velocities. As stated before, this affects their evolution. Magnetic stars, however, rapidly spin down due to magnetic braking. The loss of angular momentum due to magnetic braking is given by

$$\left(\frac{dJ}{dt}\right)_{\rm mb} \approx \frac{2}{3}\dot{M}\Omega R^2 \left(0.29 + (\eta_* + 0.25)^{1/4}\right)^2. \qquad (2)$$

as stated in Ud-Doula et al. (2009). Here, $\dot{M}$ is the mass loss rate of the star, $\eta = \frac{B_{\rm eq}^2 R^2}{\dot{M} v_\infty}$, where $B_{\rm eq}$ is the equatorial magnetic field which is equal to half of the polar field if the star has a dipolar magnetic field aligned with the rotational axis, and $v_\infty$ the terminal wind velocity of the stellar wind. Thus, magnetic stars with a high rotational velocity are generally young and less evolved, while stars with a slow rotation rate are usually older and more evolved. This does not hold in its entirety, as some stars form with a slow rotation rate, but it is a good predictor. Figure 1 shows the log(g) of the star sample plotted against their equatorial rotational velocities. Since stars usually expand over their lifetime, a higher log(g) is associated with a more evolved and thus older star. Figure 1 shows that generally, it holds that a lower $v_{\rm eq}$ is associated with a lower log(g), and thus that more evolved stars have spun down. There is one binary exception, however, but since it is the only one, this could very well be an outlier. Binary stars are on average less evolved and have higher spin velocities.

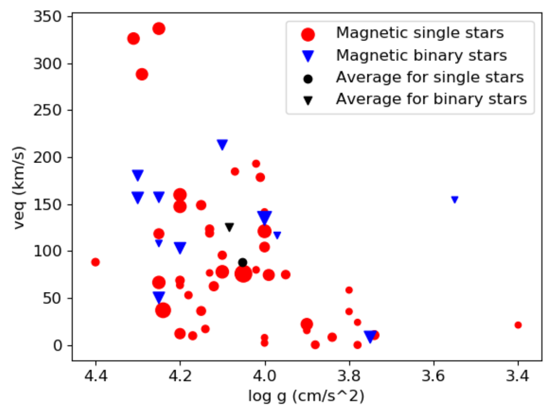



**Figure 1.** Log(g) versus equatorial rotational velocity. For binary stars, only those that are magnetic are shown. Stars are scaled to the size of their magnetic field.

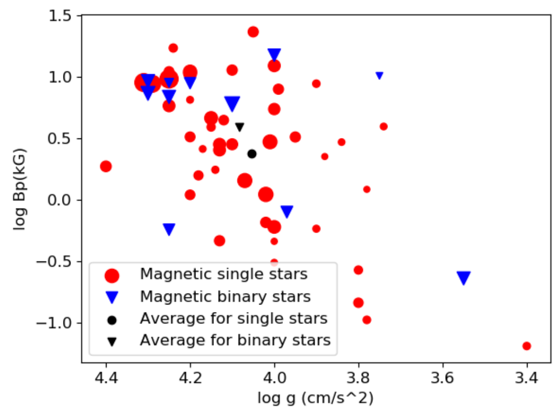



**Figure 2.** Log(g) versus magnetic field strength. For binary stars, only those that are magnetic are shown. Stars are scaled to their equatorial rotational velocity.

### 2.2 Evolution

Figure 2 shows the log(g) of the star sample plotted against their magnetic fields. This shows that the magnetic field strength generally decreases as a star evolves. This is to be expected; the star expands, but the total amount of magnetic flux is conserved, meaning the strength of the field should decrease.

### 2.3 Alfvén radius and Kepler Corotation radius.

The Alfvén radius, $R_{\rm A}$ is the radius where the magnetic energy equals the kinetic energy of material ejected by the star. Within $R_{\rm A}$, material gets funneled back onto the star, and outside $R_{\rm A}$ material breaks free from the magnetic influence of the star. The Alfvén radius is given by

$$\frac{R_{\rm A}}{R_*} \approx 0.29 + (\eta_* + 0.25)^{1/4}, \qquad (3)$$

(ud-Doula & Owocki 2002). Here, $\dot{M}_{\rm B=0}$ is the mass loss rate in the absence of a magnetic field.

The Kepler Corotation radius is the distance at which the centrifugal and gravitational forces are equal to eachother (Petit et al. 2013). It is given by

$$\frac{R_{\rm K}}{R_*} = \left(\frac{v_{\rm rot}}{v_{\rm orb}}\right)^{-2/3} = \left(\frac{v_{\rm rot}}{\sqrt{GM_*/R_*}}\right)^{-2/3}. \qquad (4)$$

In general, if a stellar Alfvén radius is larger than its Kepler Corotation radius, the star is less evolved, and when a stellar Alfvén radius is smaller than its Kepler Corotation radius, the star is more evolved. This is because the Kepler Corotation radius increases as the stellar rotational velocity decreases, in other words as the star spins down. Figure 3 shows a plot of the Alfvén radii versus the Kepler Corotation radii of the sample stars. Figure 4 gives an overview of a stellar magnetosphere and wind with respect to the Alfvén and Kepler Corotation radii. There is a notable difference between single and binary stars visible: while few single stars are very evolved to the point where their Kepler Corotation radius exceeds their Alfvén radius, this is the case for none of the binary stars. Once again, it seems that binary stars are less evolved on average than single stars.

### 2.4 Findings and possible explanation

The data seems to suggest that magnetic binary stars are less evolved than their single star counterparts. However, this data ignores the fact that these binary stars are harder to

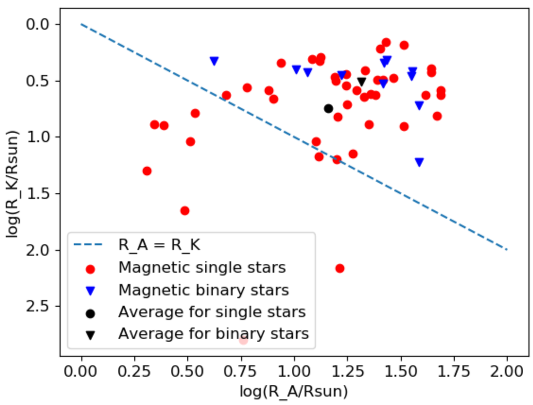



**Figure 3.** Alfvén radius versus Kepler Corotation radius. The dashed line shows where the Alfvén radius is equal to the Kepler Corotation radius.

spin down. Binary stars, per definition, have a large reservoir of angular momentum reservoir to draw from, namely the orbital angular momentum reservoir. This is obviously not the case for single stars. Thus, figures 1 and 3 show that binary stars are either less evolved on average or harder to spin down. As there is a known mechanism that would explain binary stars to spin down more slowly, this is the more likely answer. An example explanation for why magnetic binary stare are relatively unevolved would be that the magnetic braking interferes with the orbit of the system by removing angular momentum from the system, leading to a merger, or adding angular momentum to the orbit.

## 3 BINARY SYSTEM HD156324

For this research, modelling of a specific magnetic binary system is used to find characteristics of magnetic binary stars that would lead them to be relatively rare compared to single magnetic stars. The system used is HD156324. All details of the system are found in Shultz et al. (2018a). The configuration of the system is not precisely known, but consists of two well understood possible configurations. Configuration 1 would make the system a double binary, as shown in figure 5a. In this system, Aa is the primary magnetic star, with Ab being its close companion. Influence of the B-system is negligible. Configuration 2 would make the system a triple system as shown in figure 5b. In this configuration, A is the primary magnetic star, with B being a close companion and C being in a large orbit. Because component C is in a large orbit, and of similar mass to B, the effects on A and B would still be negligible. For this project, the focus will be on configuration 1, ignoring the B-system. The system will thus be treated as a simple binary.

The A-system has achieved mutual tidal locking and is circularized in orbit. The mass ratio of Aa to Ab is around 2. The orbital period is very short, and the magnetic field of the Aa component is very strong. Other important parameters of the system are found in table 1. Some of the parameters of the system are inferred instead of measured. For example, an upper limit for the magnetic field is derived as any value below this upper limit would make it currently undetectable

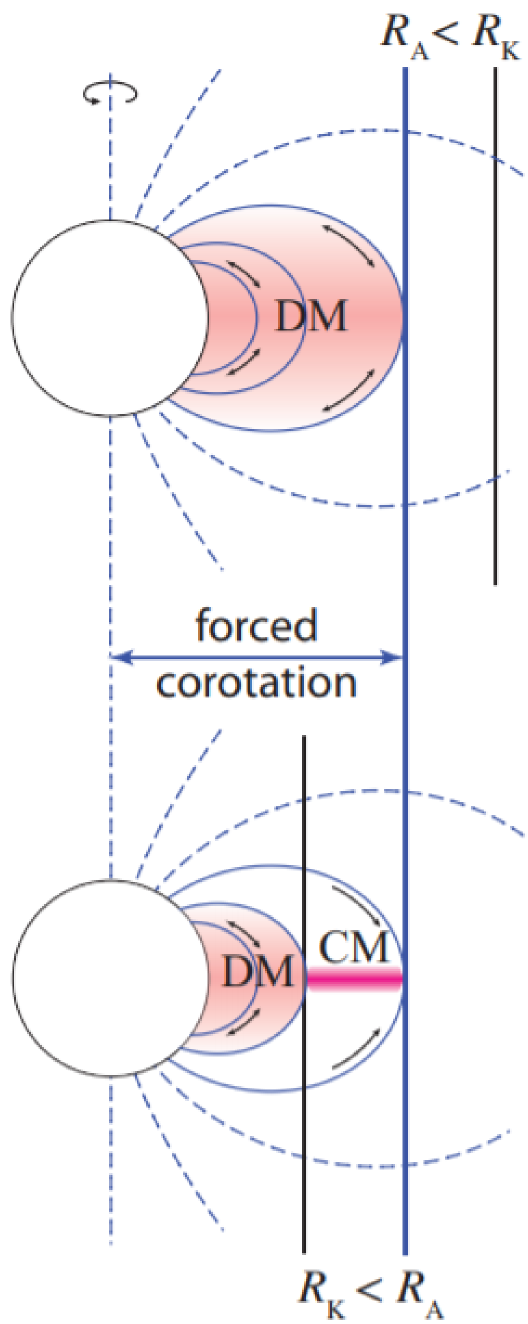
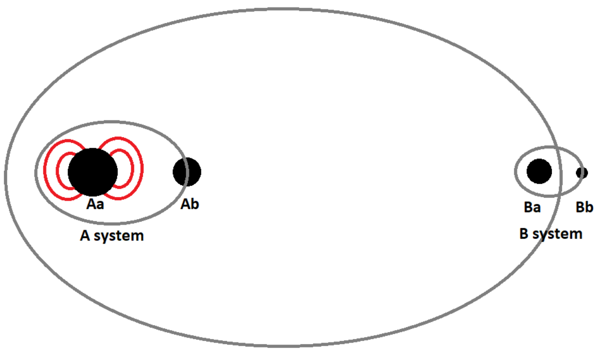
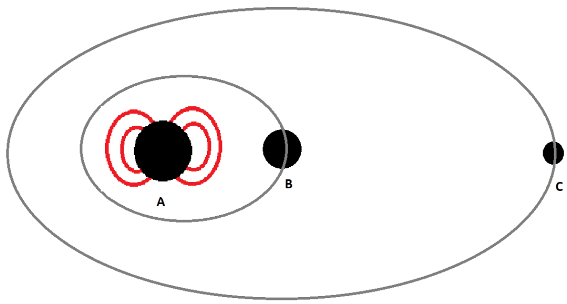



**Figure 4.** Overview of a star where $R_{\rm A} < R_{\rm K}$ and a star where $R_{\rm A} > R_{\rm K}$ from Petit et al. (2013). The regime where $R_{\rm K} < R_{\rm A}$ is called dynamical, where as the regime where $R_{\rm A} > R_{\rm K}$ is called centrifugal.

next to the magnetic field of Aa. This means that these are the parameters that are going to be looked into in more detail in the modelling section of this research. Because the magnetic field of Aa is very potent and the system is a very close binary with a separation of around 13 solar radii, the orbit of Ab lies well within the Alfvén radius of Aa, as it is around 22 solar radii. There is currently no way of modelling this in MESA, and thus it will be disregarded. This is however a detail that further research should experiment with.




(a) Schematic overview of configuration 1, the double binary.




(b) Schematic overview of configuration 2, the triple system.

**Figure 5.** Both possible configurations of HD156324. Stars are shown in black, with orbits shown in grey. Field lines in red to indicate the magnetic star. Stars, field lines and orbits are not to scale. The eccentricity shown in this figure is also not representative of the real system.

| Parameter | Aa | Ab |
|---|---|---|
| $P_{orb}(days)$ | 1.5805(1) | .. |
| a ($R_\odot$) | 13.9 (2.2) | .. e |
| < 0.01 | .. | |
| $T_{eff}(kK)$ | 22(3) | 15.5(1.5) |
| log(g) | 4.2(0.03) | 4.32(0.03) |
| $\log(\frac{L}{L_\odot})$ | 3.5(0.2) | 2.5(0.2) |
| Age (My) | .. | 7.3(3.2) |
| R ($R_\odot$) | 3.8(0.3) | 2.3(0.1) |
| M ($M_\odot$) | 8.5(1.5) | 4.1(0.3) |
| $B_d(kG)$ | 14(1.5) | < 2.6 |
| $R_A(R_\odot$ | 22(-3, +11) | 0 |

**Table 1.** Parameters of HD156324 as found in Shultz et al. (2018a).

## 4 MODELLING OF HD156324

To find characteristics of magnetic binaries that lead them to be uncommon or less evolved, a program called Modelling for Experiments in Stellar Astrophysics (Paxton et al. 2011, 2013, 2015, 2018, 2019), or MESA was used. The goal was to first find a best-fit model with starting parameters that best fit the current parameters of the system, followed by the modelling of this best-fit model to find peculiarities in the evolution of the magnetic primary component (Aa) and that of the orbit of the system. The secondary star (Ab) was also modelled, though this was mainly for the purpose of finding the best-fit[1].

### 4.1 MESA

MESA is a computer program that models stellar evolution in one dimension. The reason it models in 1D is because this much less intensive for computers, and can thus be used to create multiple models in a short amount of time. It also turns out that 1D is a good approximation for these stars and star systems. The 1D model used by MESA is essentially a line with different quantities at every point. Quantities like temperature, pressure, gravity or chemical composition are tracked and used to calculate every next step. However, because it models in 1D, certain quantities cannot be directly modeled. For example: stars rotate, and this affects their evolution. However, while it is not possible to implement these quantities directly, it fortunately is possible to implement their effects.

#### *4.1.1 Stellar wind*

A real star slowly loses mass over its lifetime, because radiation blasts away parts of the outer layer of the star. This ejected mass is called the stellar wind. Since it would be impossible for MESA to keep track of every small particle going into infinity, MESA instead removes a certain amount of mass at every step. Along with the mass, it also removes a proportional amount of angular momentum that leaves the star with the wind. The mass loss rate is also affected by rotation.

#### *4.1.2 Rotation*

In a real star, rotation has a multitude of effects. The most important ones are:

(i) Chemical mixing, the mixing of elements through the stellar interior.
(ii) A centrifugal force, pointing outward on the rotating plane. This changes the hydrostatic equilibrium of the star, making it expand at and around the equator. This force also exacerbates the mass loss due to the stellar wind.
(iii) The so-called Spruit-Tayler dynamo. This is what gives stars like the sun their magnetic field.
(iv) In the case of a binary system, rotation also contributes to the total angular momentum of the system.

In 1D, it is obviously not possible to make the model rotate, but it is possible to add all of the above-mentioned effects. As the Spruit-Taylor dynamo is not relevant for this research, it will be ignored in this paper.

MESA calculates five different mixing processes: dynamical shear instability, Solberg-Høiland instability, secular shear instability, Eddington-Sweet circulation and Goldreich-Schubert-Fricke instability. More information on these processes is found in Paxton et al. (2013).

[1] It is possible for MESA to treat the secondary star as a point mass. However, it was decided that the evolution of Ab was relevant for this work.

#### 4.1.3 Magnetic fields

In a real star, the magnetic field alters a lot of aspects of a star. An example of this is that material from the outer layers and wind gets funneled through the field lines back onto the star in loops of plasma. This can, for obvious reasons, not be modeled in 1D. However, the effects of the magnetic field can. The two most important factors the magnetic field introduces are:

(i) Magnetic braking. This effect slows down the rotation rate of the star as given by 2.
(ii) Mass loss quenching. A part of the wind gets funneled back onto the star.

Mass loss quenching is introduced via the introduction of a scaling factor for the mass loss rate discussed above. This scaling factor depends solely on the strength of the magnetic field. Magnetic braking is introduced by adding an additional torque to the system. This torque slows the rotation rate of the star. Since just adding a flat torque to the system would not just stop the system, but eventually reverse its rotation, this torque depends not only on the strength of the magnetic field, but also on the rotation rate. On top of that, MESA prevents the star from spinning backwards. For binary stars, this is not actually a standard MESA setting, but an additional set of physics developed by Z. Keszthelyi and collaborators was used. This additional set of physics is found in Appendix A.

#### 4.1.4 Binarity

For binary systems, MESA gives two options. The first is to just introduce a point mass that does not evolve. The second option is the introduction of a second star that evolves just like the primary star. Orbital parameters like eccentricity and separation are added, and will also be calculated with every step.

### 4.2 HD156324

MESA models a star from ZAMS to TAMS (Zero Age Main Sequence to Terminal Age Main Sequence). This introduces a problem; we observe HD156324 at a point past its formation. As such, the initial parameters of the system are unknown. To find the initial parameters of the system, a number of parameter tests was conducted, with the goal of finding a model that fits the parameters we observe at roughly the same point in its evolution. The most important parameters to fit were:

(i) Polar magnetic field strength B,
(ii) The surface gravity log(g) and effective surface Temperature $T_{\rm eff}$ ,
(iii) the Orbital and Rotational periods $P_{\rm orb}$ and $P_{\rm rot}$ and where they match.
(iv) The mass ratio of the Aa and Ab star.

Less important parameters to fit were the individual masses of the stars, the age of the system and the luminosity of both stars.

After finding the best-fit model the goal was to find characteristics that would impact the orbital evolution of the system.

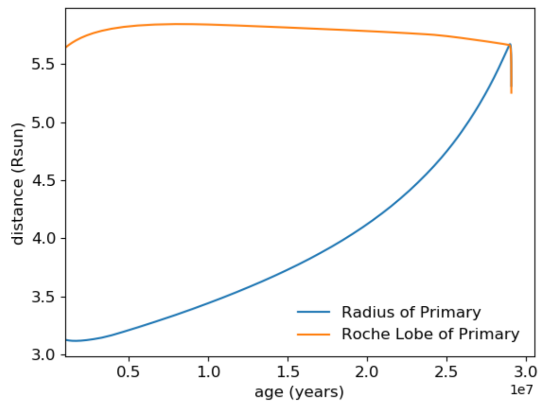



**Figure 6.** Stellar radius versus its Roche Lobe of the best fit model for Primary star Aa.

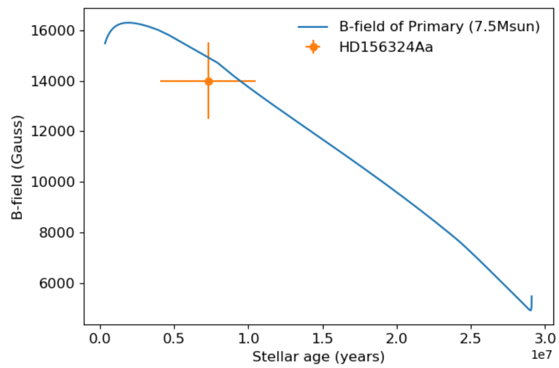



**Figure 7.** The magnetic field of the best-fit model over time.

## 5 RESULTS

The best-fit model was found via parameter tests and a fit-by-eye method of finding the best combined values.

### 5.1 Mass transfer at TAMS Aa

Something that is important to note before going into the relevant results is that in all models of the system, mass transfer via Roche Lobe overflow occurs. For the best-fit model, this is visible in figure 6. Mass transfer is beyond the scope of this research, so there will not be much more detail on this. However, this explains why the tail end of the evolutionary models behave strangely: at that point, mass transfer is occurring.

### 5.2 Best-fit parameters

As the magnetic field of a star evolves over time, a starting value of 9 kG was chosen. This is because at the start of a model, the evolution of the magnetic field would be impacted by the initial rotation of the system. The evolution of the magnetic field of Aa is shown in figure 7. The initial values for the best-fit model are shown in table 2. Figure 8 Shows the evolution of Aa and Ab. Aa seems to reach observed values at a much younger age than Ab. As shown in figure 9, the

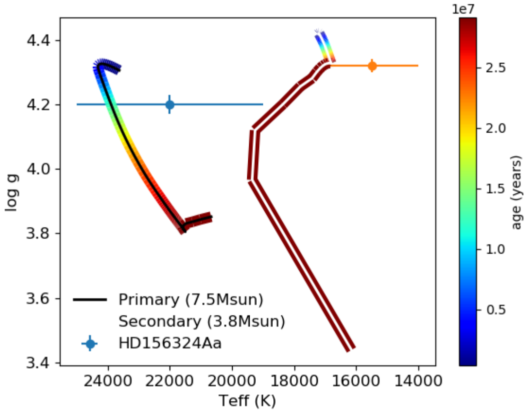



**Figure 8.** Evolutionary model of best-fit model. Aa seems to pass through the detected values at a much younger age than Ab.

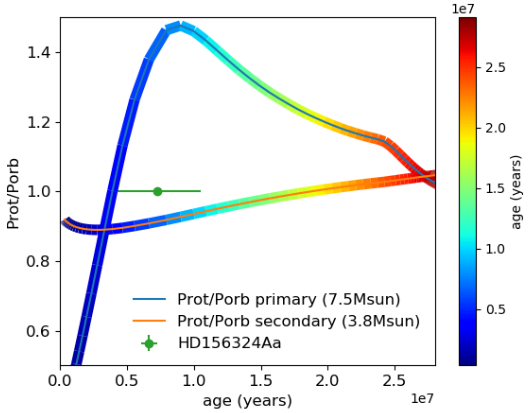



**Figure 9.** The rotational period divided by the orbital period of Aa and Ab. Aa seems to spin down very rapidly initially, but spins up a bit after a certain point.

| Parameter | Aa | Ab |
|---|---|---|
| $P_{orb}(days)$ | 1.45 | .. |
| M ($M_\odot$) | 7.5 | 3.8 |
| $v_{rot}(km/s)$ | 450 | tidally locked to orbit |
| $B_d(kG)$ | 9 | 0 |

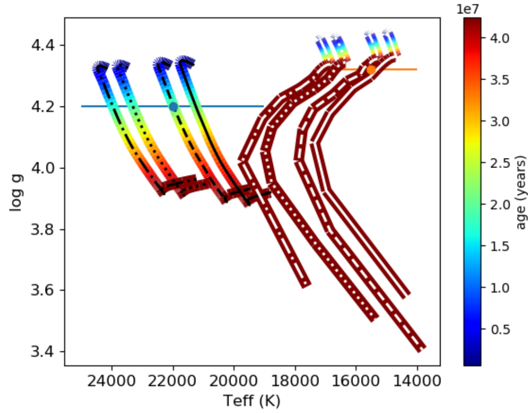
**Table 2.** Initial parameters of the Best-fit model.

system eventually moves towards tidal locking. Mass transfer only seems to occur after Aa reaches TAMS. Figure 10a shows four models with different masses but similar mass ratios. As in figure 8, Aa seems to pass observed values at a much younger age than Ab. An important note is that uncertainties in $log(g)$ can easily surpass 0.1 Dex, which could nullify this result.

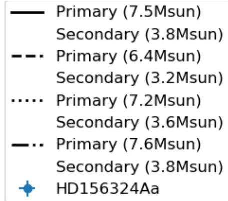
# 6 DISCUSSION

## 6.1 The Best-fit model

The best fit model does not seem to show any reason as to why the orbital evolution of the system would significantly




(a) Evolutionary models of four different masses but the same mass ratio as best-fit model. Just like the best-fit model, Aa seems to pass through observed values at a much younger age than Ab.




(b) Legend for figure 10a

**Figure 10.** Plot and legend of a parameter test where different masses were tested.

change due to the introduction a magnetic field. It shows a major spin-down of Aa as shown in figure 9, but this would only lead to the orbital period expanding slightly to compensate for the loss of spin angular momentum. The mass transfer occurs because Aa nears the TAMS and starts filling its Roche Lobe, not because of orbital decline. It proved to be hard to fit a model that eventually tidally locked. The introduction of magnetic braking always disrupted tidal locking. This is a problem because the orbital and rotational periods are known to be very precise. The best-fit model seems to approach tidal locking, but Aa reaches TAMS before it happens, so no true conclusion can be drawn that it actually will. An important detail is that, due to magnetic braking, at first Aa spins down to well below the orbital velocity in around 10 million years, and then begins to spin up again. The spin down is caused by magnetic braking, while the spin up is caused by Aa drawing angular momentum from the orbital reservoir. It can thus be said that on a short timescale of about 10 million years, the magnetic braking is the dominant force on the rotational evolution of the star, while on a larger timescale, in this case the 25 million years the system exists in its current state, the binarity becomes the dominant force acting on the stars rotation. There is a noticeable effect on the rotation rate of Ab due to the magnetic braking on Aa, but this effect is rather small. It is safe to say that even in such a close binary, the effects of magnetic braking on orbital evolution are small. It also proved to be very hard to fit both Aa and Ab through the their observed $T_{\rm eff}$ vs log(g) at around the same time frame. This only seems to be possible by vastly reducing the mass of Aa. The problem with that, however, is that the best-fit model already assumes the very lowest end of the mass ratio. This mass ratio is one of the more precisely known variables of the system as it influences the doppler shifts used to measure the period. Thus, the decision was made to fit with the current mass ratio of around 1.97. As shown in figures 8 and 10a, at this mass ratio, Aa passes through observed values at a much younger age than Ab. In fact, Ab never actually passes observed values, as Aa reaches TAMS well before. Shultz et al. (2018a) assumes that both stars formed at the same time. This is the standard assumption for binary stars. Another option is that while the system formed at the same time, Aa reformed from a stellar merger. It would also be possible that originally, Ab was the more massive star, and that mass transfer from Ab to Aa already happened, which would explain why Aa looks younger. However, not much is known about if this process could form a fossil magnetic field.

### 6.2 Merger formation

As it was not the focal point of this research, there will not be a great amount of detail regarding merger formation. The age disparity between Aa and Ab is consistent with the hypothesis of merger formation. However, due to the uncertainty of $log(g)$ in Ab, if slightly different observations arise, the opposite conclusion would have to be drawn. Thus, while it a crucial part of information in finding out more about merger formations, it is inconclusive at best.

### 6.3 Magnetic field origin

The exact origin of the fossil magnetic field is still debated. As they are a result of flux conservation, they are thought to originate from either a rejuvenation process like in the case of a merger formation, or the simple amplification of a seed field into the observed fossil fields during star formation. Finding binaries where one star is younger than its companion could be a sign of rejuvenation. This work does seem to suggest such an age disparity, but that disparity is highly dependent on the accuracy of the measurements that the evolutionary targets are based on. For future research, this work can essentially be recreated for all other known binary systems with one or more magnetic components. As more information about HD156324 is revealed, it will also be possible to perform a better fitting model, creating a need to reproduce this work. Additionally, a very interesting angle into the relative absence of magnetic binary systems is to find evidence of the merger formation of the magnetic components. This information is crucial to be able to place more firm constraints on the origin of the magnetic field.

## 7 CONCLUSION

In this research, no evidence was found for magnetic fields significantly impacting binary evolution. Magnetic binaries do not significantly differ from their single star counterparts. Their rotational values do seem to differ from their single star counterparts, but this could also be explained by simple orbital mechanics. To add, the sample size of magnetic binaries is to small to draw definitive conclusions. A model was created to replicate binary system HD156324, and more accurately find values for initial parameters of the system via a best-fit model. This best-fit model did not suggest any impact on binary evolution linked to magnetic fields, and while it does hint at a possibility that the magnetic primary component of the binary to be the result of a merger formation, with present-day uncertainties, it is impossible to favour one scenario over the other. More measurements of the system combined with more research into merger formation seems to be the logical next step.

## References

Alecian E., Wade G. A., Catala C., Grunhut J. H., Landstreet J. D., Böhm T., Folsom C. P., Marsden S., 2013, MNRAS, 429, 1027
Alecian E., et al., 2014, A&A, 567, A28
Braithwaite J., Nordlund Å., 2006, A&A, 450, 1077
Ferrario L., Wickramasinghe D., 2008, MNRAS, 389, L66
Fossati L., et al., 2015, A&A, 582, A45
Grunhut J. H., et al., 2017, MNRAS, 465, 2432
Keszthelyi Z., Meynet G., Georgy C., Wade G. A., Petit V., David-Uraz A., 2019, MNRAS, 485, 5843
Keszthelyi Z., et al., 2020, MNRAS, 493, 518
Meynet G., Eggenberger P., Maeder A., 2011, A&A, 525, L11
Morel T., et al., 2015, in Meynet G., Georgy C., Groh J., Stee P., eds, IAU Symposium Vol. 307, New Windows on Massive Stars. pp 342–347 (arXiv:1408.2100), doi:10.1017/S1743921314007054
Neiner C., Mathis S., Alecian E., Emeriau C., Grunhut J., BinaMIcS MiMeS Collaborations 2015, in Nagendra K. N., Bagnulo S., Centeno R., Jesús Martínez González M., eds, IAU Symposium Vol. 305, Polarimetry. pp 61–66 (arXiv:1502.00226), doi:10.1017/S1743921315004524
Paxton B., Bildsten L., Dotter A., Herwig F., Lesaffre P., Timmes F., 2011, ApJS, 192, 3
Paxton B., et al., 2013, ApJS, 208, 4
Paxton B., et al., 2015, ApJS, 220, 15
Paxton B., et al., 2018, ApJS, 234, 34
Paxton B., et al., 2019, ApJS, 243, 10
Petit V., et al., 2013, MNRAS, 429, 398
Puls J., Vink J. S., Najarro F., 2008, A&ARv, 16, 209
Sana H., et al., 2014, ApJS, 215, 15
Schneider F. R. N., Podsiadlowski P., Langer N., Castro N., Fossati L., 2016, MNRAS, 457, 2355
Shultz M., Rivinius T., Wade G. A., Alecian E., Kochukhov O., BinaMIcS Collaboration 2018a, Contributions of the Astronomical Observatory Skalnate Pleso, 48, 298
Shultz M. E., et al., 2018b, MNRAS, 475, 5144
Ud-Doula A., Owocki S. P., Townsend R. H. D., 2009, MNRAS, 392, 1022
Wade G. A., et al., 2016, MNRAS, 456, 2
ud-Doula A., Owocki S. P., 2002, ApJ, 576, 413

## A MESA BINARY MAGNETIC BRAKING RUN_STAR_EXTRAS ROUTINE

```
module run_star_extras

  use const_lib
  use const_def
  use star_private_def
  use star_lib
  use star_def
  use chem_def, only: ih1, ihe4, ic12, ic13, in14, io16
  use binary_def

  implicit none

      ! Module variables:
      !==========================
      real(dp) :: R_init
      real(dp) :: Bp
      real(dp) :: B0
      real(dp) :: Beff
      real(dp) :: Deceff
      real(dp) :: vesc
      real(dp) :: vinf
      real(dp) :: etastar
      real(dp) :: Ra
      real(dp) :: Rc
      real(dp) :: f_B
      real(dp) :: logf_B
      real(dp) :: f_rot
      real(dp) :: Mdot_orig
      real(dp) :: w_vink
      real(dp) :: w_graefener
      real(dp) :: w_vanloon
      real(dp) :: w_dejager
      real(dp) :: Jbrake
      real(dp) :: J_surf
      real(dp) :: scale
      real(dp) :: core_h1_init
      real(dp) :: enclosedm
      integer  :: nin, k_max
      integer, parameter :: UNIFORM_TORQUE = 1
      integer, parameter :: SURFACE_TORQUE = 2
      integer, save  :: torque_method = 0
      integer, parameter :: CON = 1
      integer, parameter :: DEC = 2
      integer, save  :: fieldevolution = 0
      !==========================

 contains

  subroutine extras_controls(id, ierr)

     type (star_info), pointer :: s

    integer, intent(in) :: id
    integer, intent(out) :: ierr

    ierr = 0

    call star_ptr(id, s, ierr)
```

```
       if (ierr /= 0) then
          write(*,*) 'failed in star_ptr'
          return
       end if


  s% extras_startup => extras_startup
  s% extras_check_model => extras_check_model
  s% extras_finish_step => extras_finish_step
  s% extras_after_evolve => extras_after_evolve
  s% how_many_extra_history_columns =>
  how_many_extra_history_columns
  s% data_for_extra_history_columns =>
  data_for_extra_history_columns
  s% how_many_extra_profile_columns =>
  how_many_extra_profile_columns
  s% data_for_extra_profile_columns =>
  data_for_extra_profile_columns

  s% other_torque => MAG_braking
  s% other_wind => MAG_WIND
  s% other_am_mixing => MAG_MIX



  s% job% warn_run_star_extras =.false.


  if (s%job%extras_lrpar >= 1) then
     B0 = s%job%extras_rpar(1)
     Beff = s%job%extras_rpar(2)
     Deceff = s%job%extras_rpar(3)
  endif

  if (s%job%extras_lcpar >= 1) then

!  if (s%job%extras_lcpar == 1) then

     select case (s%job%extras_cpar(1))
     case ('UNIFORM_TORQUE')
        torque_method = UNIFORM_TORQUE
     case ('SURFACE_TORQUE')
        torque_method = SURFACE_TORQUE
    end select


!  if (s%job%extras_lcpar == 2) then

     select case (s%job%extras_cpar(2))
     case ('CON')
        fieldevolution = CON
     case ('DEC')
        fieldevolution = DEC
    end select


  endif

end subroutine extras_controls
```

```
  integer function extras_startup(id, restart, ierr)
         type (star_info), pointer :: s
         integer, intent(in) :: id
         logical, intent(in) :: restart
         integer, intent(out) :: ierr
         integer :: h1
         ierr = 0
         call star_ptr(id, s, ierr)
         call get_star_ptr(id,s,ierr)
         extras_startup = 0

         ! Usual startup stuff
         if (.not. restart) then
            call alloc_extra_info(s)
         else ! it is a restart
            call unpack_extra_info(s)
         end if


          h1 = s% net_iso(ih1)
          core_h1_init = s% xa(h1,s% nz)

          R_init = 0.0d0



      end function extras_startup




subroutine gammae(id,ierr)
  type (star_info), pointer :: s
  integer, intent(in) :: id
  integer, intent(out) :: ierr
  integer :: h1, he4, nz
  real(dp) :: alfa1, alfa2, L1, M1, T1, R1
  real(dp) :: I_he, YHe, sigma_e, Gamma_e
  real(dp) :: vinf1, vinf2, vinf3, logvinf
  real(dp) :: surface_h1, surface_he4
  real(dp) :: nom, denom, pow
  real(dp) :: f_rot1, f_rot2, f_rot3
  real(dp) :: X, Y, Z  ! H, He and metal mass fractions
  real(dp), parameter :: alpha = 0.66d0


! The calculations are partitioned into three Teff regimes.
  real(dp), parameter :: Teff_jump1 = 20.d3
! FIRST bi-stability jump temperature (from observations)
  real(dp), parameter :: Teff_jump2 = 10.d3
! SECOND bi-stability jump temperature (from observations)
  real(dp), parameter :: dT1 = 1.d3
! Interpolation width of the first bi-stability jump.
! Note that a small dT results in a steep change
! in Mdot while a larger dT yields a more gradual change.
  real(dp), parameter :: dT2 = 1.d3   ! second jump
! Interpolation width of the second bi-stability jump.

! --- Values related to terminal wind velocity ---

  real(dp), parameter :: fvinf1 = 2.6d0
  real(dp), parameter :: fvinf2 = 1.3d0
```

```
  real(dp), parameter :: fvinf3 = 0.7d0

  call get_star_ptr(id,s,ierr)

  L1 = s% photosphere_L        ! cgs
  M1 = s% star_mass * Msun     ! cgs
  T1 = s% Teff                 ! cgs
  R1 = s% photosphere_r * Rsun  ! cgs

!--- elements ----
  nz = s% nz
  h1 = s% net_iso(ih1)
  he4 = s% net_iso(ihe4)
  surface_h1 = s% xa(h1,1)
  surface_he4 = s% xa(he4,1)
  X = surface_h1
  Y = surface_he4
  Z = 1.d0 - (X + Y)

  YHe = Y/X / 4.d0
! surface Helium content by number,
! n_He/n_H; approximation: factor 4 should be m_He/m_H

! === PATH 1 ===
  ! --- vinf behaviour --- check for consistency with wind routine
 if (T1 .ge. Teff_jump1 - dT1) then
    I_he = 2.d0
    sigma_e = 0.398d0 * (1.d0 + YHe * I_he) /
    (1.d0 + 4.d0 * YHe)
    Gamma_e =  sigma_e * L1/(M1 * pi4 * standard_cgrav * clight)
! escape velocity corrected for electron scattering
    vesc = sqrt((1.d0 - Gamma_e) * 2.d0 * standard_cgrav * M1 / R1 )
! Eq. 15 KPW2017
    vinf1 = vesc * fvinf1
    vinf = vinf1
! rotational enhancement
    nom = 1.d0 - Gamma_e
    denom = 1.d0 - (4.d0/9.d0 * pow_cr
    (s%v_rot_avg_surf / s%v_crit_avg_surf, 2.0d0) ) - Gamma_e
    pow = (1.d0 / alpha) - 1.d0
    f_rot1 = pow_cr(nom, pow) / pow_cr(denom, pow)
    f_rot = f_rot1
endif
! === PATH 2 ===
 if (T1 .le. Teff_jump1 + dT1 .and. T1 .ge. Teff_jump2 - dT2) then
    I_he = 1.d0     ! cooler side of jump, Helium recombined to HeII
    sigma_e = 0.398d0 * (1.d0 + YHe * I_he) / (1.d0 + 4.d0 * YHe)
    Gamma_e =  sigma_e * L1/(M1 * pi4 * standard_cgrav * clight)
    vesc = sqrt(2.d0 * standard_cgrav * M1 / R1 * (1.d0-Gamma_e))
    vinf2 = vesc * fvinf2
    vinf = vinf2
! rotational enhancement
    nom = 1.d0 - Gamma_e
    denom = 1.d0 - (4.d0/9.d0 * pow_cr
    (s%v_rot_avg_surf / s%v_crit_avg_surf, 2.0d0) ) - Gamma_e
    pow = (1.d0 / alpha) - 1.d0
    f_rot2 = pow_cr(nom, pow) / pow_cr(denom, pow)
    f_rot = f_rot2
 endif
! --- INTERPOLATION for the FIRST bi-stability jump ---
 if (Teff_jump1 + dT1 > T1 .and. T1 > Teff_jump1 - dT1) then
    alfa1 = (T1 - (Teff_jump1 - dT1)) / (2.d0 * dT1)
```

```
    vinf = alfa1 * vinf1 + (1.d0 - alfa1) * vinf2
    f_rot = alfa1 * f_rot1 + (1.d0 - alfa1) * f_rot2
 endif
! === PATH 3 ===
 if (T1 .le. Teff_jump2 + dT2) then
    sigma_e = 0.398d0 * 1.d0 / (1.d0 + 4.d0 * YHe)
    Gamma_e =  sigma_e * L1/(M1 * pi4 * standard_cgrav * clight)
    vesc = sqrt(2.d0 * standard_cgrav * M1 / R1 * (1.d0-Gamma_e))
    vinf3 = vesc * fvinf3
    vinf = vinf3
! rotational enhancement
    nom = 1.d0 - Gamma_e
    denom = 1.d0 - (4.d0/9.d0 * pow_cr
    (s%v_rot_avg_surf / s%v_crit_avg_surf, 2.0d0) ) - Gamma_e
    pow = (1.d0 / alpha) - 1.d0
    f_rot3 = pow_cr(nom, pow) / pow_cr(denom, pow)
    f_rot = f_rot3
 endif
! --- INTERPOLATION for the SECOND bi-stability jump ---
 if (Teff_jump2 + dT2 > T1 .and. T1 > Teff_jump2 - dT2) then
    alfa2 = (T1 - (Teff_jump2 - dT2)) / (2.d0 * dT2)
    vinf = alfa2 * vinf2 + (1.d0 - alfa2) * vinf3
    f_rot = alfa2 * f_rot2 + (1.d0 - alfa2) * f_rot3
 endif



if (f_rot .le. 1.d-4 .OR. s%v_rot_avg_surf .le. 1.d0) then
 f_rot = 1.d0
endif

end subroutine gammae


subroutine magnetic(id,ierr)
  type (star_info), pointer :: s
  integer, intent(in) :: id
  integer, intent(out) :: ierr
  integer :: h1
  real(dp) :: L1, M1, T1, R1, core_h1
  real(dp) :: logB0, logBp, logetastar
  real(dp) :: logvesc, logvinf

!====  Initial magnetic flux can also be set.
!      Set only if B0 is not set.
!
!  real(dp), parameter :: F0 = 10**28
!
!====

 call gammae(id,ierr)  ! call in gammae for vinf routine
 call get_star_ptr(id,s,ierr)

! basic parameters
  L1 = s% photosphere_L ! cgs
  M1 = s% star_mass * Msun   ! cgs
  T1 = s% Teff      ! cgs
  R1 = s% photosphere_r * Rsun  ! cgs

  h1 = s% net_iso(ih1)
  core_h1 = s% xa(h1,s% nz)
```

```
 if (.not. s% doing_relax .AND. s% model_number > 1) then


  select case (fieldevolution)  ! set in inlist.

! ====================================
!  ===  MAGNETIC FLUX CONSERVATION ===
! ====================================
!
  case (CON)
   logBp = log10_cr(B0) + 2.d0 * log10_cr(R_init) - 2.d0 * log10_cr(R1)  ! cgs
   Bp = exp10_cr(logBp)

      WRITE(*,*) '::::::::::'
      WRITE(*,*) Bp
      WRITE(*,*) '::::::::::'

!
! ===================================
! === Equatorial magnetic confinement parameter ===                ! in cgs
  logetastar = 2.d0 * logBp + 2.d0 * log10_cr(R1)    &
               - log10_cr(Mdot_orig * Msun/secyer)   &
               - log10_cr(vinf) - log10_cr(4.0d0)
  etastar = exp10_cr(logetastar)

! === Alfve'n and closure radii ===
  Ra = 1.0d0 + pow_cr(etastar + 0.25d0, 0.25d0) - pow_cr(0.25d0, 0.25d0) ! this is: R_A / R_star
  Rc = 1.0d0 + 0.7d0 * (Ra - 1.0d0)                                      ! this is: R_c / R_star

! === Mass-loss quenching parameter ===    ! used to multiply original Mdot in wind routine
  f_B = 1.0d0 - sqrt(1.0d0 - 1.0d0/Rc)
  logf_B = log10_cr(f_B)

! For rotating models, this scaling will need to consider
! an updated version, using Eq. 22 from ud-Doula et al. (2009).
! The update is only needed if the star spends a significant time
! with a centrifugal magnetoshpere, and even then the correction factor is small.

 endif  ! for not doing relax.

end subroutine magnetic

! ===================================
!      === MAGNETIC BRAKING ===
! ===================================

subroutine MAG_braking(id, ierr)
        integer, intent(in) :: id
        real(dp) :: R1, M1, T1, Mdot, logvesc, modelm
        real(dp) :: logBp, logvinf, logetastar, Omega, Jtotnew, komega, qpart
        integer, intent(out)      :: ierr
        type (star_info), pointer :: s
        integer                   :: nink
        integer                   :: k, nz
        real(dp)                 :: J_lost, J_ex
        ierr = 0
        call star_ptr(id, s, ierr)
        if (ierr /= 0) return

 if (s% doing_relax) then
```

```
 write(*,*) 'still relaxing a bit - you should do the same'
 write(*,*) ''
 end if
!-------------------------------
 if (.not. s% doing_relax .AND. R_init> 0._dp .AND. s%mstar_dot < 0._dp) then

  R1 = s% photosphere_r * Rsun  ! cgs
  M1 = s% photosphere_m * Msun  ! cgs

  !!! find better definition
  !k_max = s%k_const_mass ! number of constant mass zone

  ! arbitrary threshold and scaling to establish how
  ! deeply the fossil field is anchored inside the star
  ! TEST: 0.1% of the total mass can lose AM:
  !
  modelm = 1.d0 ! M1/Msun * 1.d-1   !!! min(1.d0 - Bp/1.d4,1.d0) ! cgs

  do k = 1, s%nz ! find where braking will be applied
    if (DOT_PRODUCT(s%dm_bar(1:k), s%j_rot(1:k)) .GE. 0.10d0 * s%total_angular_momentum) then
       nin = k  ! index of the last zone where braking is applied
       exit
    endif
  end do
      WRITE(*,*) nin
      WRITE(*,*) s%total_angular_momentum
     ! WRITE(*,*) s%r(k_max), R1/Rsun
  !===
  !
  !
  ! FAS model - implemented simply for now,
  ! physically better approach should be considered.
  k_max = nin    !800 - max(int(100*s%star_age/1.d6),700)


  if (nin > 0.) then
      enclosedm = sum(s%dm_bar(1:k_max)) / Msun
    else
      enclosedm = 0.d0
  end if



  Mdot =  Mdot_orig * Msun / secyer          ! in cgs ! this is the original mass-loss rate for B=0.
  Omega = s% omega(1)   ! s% omega_avg_surf  ! in history this is 'surf_avg_omega', why???

  J_lost = s%angular_momentum_removed   ! by the mass loss

  ! === Total angular momentum loss due to magnetic fields ===
  ! This is dJ(field+gas)/dt from ud Doula, Owocki, Townsend (2009)
  ! Ra is in R_star units, so now it is converted to cgs.
  ! The negative sign will follow later to extract this quantity.

  Jbrake = (2.0d0 / 3.0d0) * Mdot * Omega * (Ra*R1)**2.d0

  J_ex = max(Jbrake * s% dt - J_lost,0._dp)   ! avoid double counting the AM removed by mass-loss.

  if (J_ex < 0._dp) then
   WRITE (*,*) '::::::::::::'
   WRITE (*,*) 'J ex cannot be negative.'
   WRITE (*,*) Mdot, Omega, s% v_rot_avg_surf, Ra, Jbrake, J_ex
   ierr = -1
```

```
   return
  end if

  if (J_ex >= 0._dp) then

! ===================================================================
! Two torque methods are introduced because the exact
! penetration depth of the fossil field remains unknown.
!
! UNIFORM will remove AM from the entire star, SURFACE will only brake the surface rotation.
! This is a simplifying assumption because we are not manipulating the angular momentum transport,
! only the loss.
!
    select case (torque_method)  ! set in inlist.
    case (UNIFORM_TORQUE)        ! convert J_ex to extra_j throughout the entire star.
  ! Jbrake converted to specific AM via scale:
  ! the same fraction of specific AM is removed from layer to layer but not the same amount.

    scale = J_ex / s% total_angular_momentum

      if (scale < 1._dp .AND. scale > 1.d-15) then
        do k = 1,s%nz
          s%extra_jdot(k) = - s% j_rot(k) *  scale / s% dt    ! negative sign added here: this is AM loss.
        end do
       else
        do k = 1, s%nz
         s% extra_jdot(k) = 0._dp
        end do
       end if

     case (SURFACE_TORQUE)        ! convert J_ex to extra_j in the near-surface of the star.

     J_surf = DOT_PRODUCT(s%dm_bar(1:k_max), s%j_rot(1:k_max))
     scale = Beff * J_ex  / J_surf ! consider efficiency parameter?
!
! apply magnetic torque if the scaleing is appropriate and the surface rotation
! is above 1 km s-1.
      if (scale < 1._dp .AND. scale > 1.d-15) then
        do k = 1,k_max  !!! nin
          s%extra_jdot(k) = - s% j_rot(k) * scale / s% dt    ! negative sign added here: this is AM loss.
        end do
       else
        do k = 1,k_max
         s% extra_jdot(k) = 0._dp
        end do
       end if

      case default
       stop 'Invalid torque method'

      end select  ! method
    end if        ! Jbrake not zero

   end if         ! not doing relaxation

end subroutine MAG_braking

! enforce a high viscosity term in layers
! where the magnetic torque is applied
subroutine MAG_MIX(id, ierr)
         integer, intent(in) :: id
         integer, intent(out) :: ierr
```

```
   type (star_info), pointer ::s
   integer :: k
        ierr = 0
        call star_ptr(id, s, ierr)
        if (ierr /= 0) return

 if (Bp > 0.d0) then
  do k=1,k_max-5    ! -5 added to introduce a few "transition" zones
    s%am_nu_rot(k) = 1.d28 !
    s%am_nu_non_rot(k) = 1.d28 !
  end do
 end if

end subroutine MAG_MIX



! ========================================
!   === MAGNETIC MASS-LOSS QUENCHING ===
!      === ROTATIONAL ENHANCEMENT ===
! ========================================

subroutine MAG_WIND(id, L_phot, M_phot, R_phot, T_phot, w, ierr)
  type (star_info), pointer :: s
  integer ::  h1, he4, nz
  real(dp) :: X, Y, Z, surface_h1, surface_he4, T1, L1, M1, R1
  real(dp), intent(in) :: L_phot, M_phot, R_phot, T_phot ! photospheric values (cgs)
  real(dp) :: w1, logMdot, gamma_edd, xsurf, beta, gammazero, lgZ ! for Gr. wind
  real(dp), parameter :: Zsolar = 0.019d0   ! for G. wind
  integer, intent(in) :: id
  integer, intent(out) :: ierr
  real(dp), intent(out) :: w
  logical, parameter :: dbg = .false.
  real(dp) :: log10wvan
  include 'formats'

  call get_star_ptr(id,s,ierr)  ! pointer to get data
  call magnetic(id,ierr)  ! quenching routine to obtain f_B
  call gammae(id,ierr)    ! call for rotational boost on mass-loss rates

  T1 = T_phot
  L1 = L_phot
  M1 = M_phot
  R1 = R_phot

  nz = s% nz
  h1 = s% net_iso(ih1)
  he4 = s% net_iso(ihe4)

  surface_h1 = s% xa(h1,1)
  surface_he4 = s% xa(he4,1)

  X = surface_h1
  Y = surface_he4
  Z = 1 - (X + Y)

if (T1 .GE. 10.d3 .AND. X > 0.3d0) then
 call Vink(id, L_phot, M_phot, R_phot, T_phot, w, ierr)
 Mdot_orig = w_vink
end if

! Grafener, G. & Hamann, W.-R. 2008, A&A 482, 945
```

```
! routine contributed by Nilou Afsari
if (T1 > 10.d3 .AND. X < 0.3d0) then
  xsurf = surface_h1
  gamma_edd = exp10_cr(-4.813d0)*(1+xsurf)*(L1/Lsun)*(Msun/M1)
  lgZ = log10_cr(Z/Zsolar)
  beta = 1.727d0 + 0.250d0*lgZ
  gammazero = 0.326d0 - 0.301d0*lgZ - 0.045d0*lgZ*lgZ
  logMdot = &
             + 10.046d0 &
              + beta*log10_cr(gamma_edd - gammazero) &
              - 3.5d0*log10_cr(T1) &
              + 0.42d0*log10_cr(L1/Lsun) &
              - 0.45d0*xsurf
  w1 = exp10_cr(logMdot)
  w_graefener = w1
  Mdot_orig = w_graefener
end if

! helium rich Wolf-Rayet star: Nugis & Lamers
if (T1 > 10.d3  .AND. X < 0.4d0 .AND. X > 0.3d0) then ! helium rich Wolf-Rayet star: Nugis & Lamers
  Mdot_orig =  1d-11 * pow_cr(L1/Lsun,1.29d0) * pow_cr(Y,1.7d0) * sqrt(Z)
end if

!if (T1 < 9.d3) then
 ! van Loon et al. 2005, A&A, 438, 273
!  log10wvan = -5.65d0 + 1.05*log10_cr(L1/(1d4*Lsun)) - 6.3d0*log10_cr(T1/35d2)
!  w_vanloon = exp10_cr(log10wvan)
!  Mdot_orig = w_vanloon
!end if

! de Jager, C., Nieuwenhuijzen, H., & van der Hucht, K. A. 1988, A&AS, 72, 259.
if (T1 < 10.d3) then
  logMdot = 1.769d0*log10_cr(L1/Lsun) - 1.676d0*log10_cr(T1) - 8.158d0
  w = exp10_cr(logMdot)
  w_dejager = w
  Mdot_orig = w_dejager
end if

! after the second bi-stability jump (the jump is not adopted)
 if (T1 < 11.d3 .AND. w_dejager > w_vink) then
  Mdot_orig = w_dejager
 end if

 ! ========================================
 ! === Final mass-loss rate of the star ===
 ! ========================================
if (f_B > 0.d0 .AND. f_rot > 0.d0) then
  w = Mdot_orig * f_B * f_rot  ! scaled by mass-loss quenching and rotational enhancement
 else
  w = Mdot_orig
  write(*,*) 'Either f_B or f_rot is zero.'
  write(*,*) f_B
  write(*,*) f_rot
end if
  if (dbg) write(*,*) 'wind', w

end subroutine MAG_WIND
```

# B BINARY INLIST FILES

## B.1 inlist1

```
! inlist_test_rlo



&star_job

    show_log_description_at_start = .false.

    new_rotation_flag = .true.
    change_rotation_flag = .true.
    set_initial_surface_rotation_v = .true.
    !set rotational velocity to zero, tides will change this
    new_surface_rotation_v = 450
    num_steps_to_relax_rotation = 30

    !load_saved_model = .true.
    !saved_model_name = 'model1.mod'

    pgstar_flag = .false.
    pause_before_terminate = .true.

    set_initial_age = .true.
    initial_age = 0

    set_initial_model_number = .true.
    initial_model_number = 0

    extras_lrpar = 3
    extras_rpar(1) = 9.d3  !0.d3  ! B0 initial polar magnetic field strength in gauss
    extras_rpar(2) = 1.d0  !!!7.d0  ! Beff = efficiency of magnetic braking
    extras_rpar(3) = 1.d0   ! Deceff = efficiency of magnetic field decay B0*exp(- Deceff * star_age/12d6)
                           ! where t_ms = 12d6 is tailored for tau Sco's approx. ms lifetime.
    extras_lcpar = 2
    ! torque_method:
    extras_cpar(1) = 'UNIFORM_TORQUE'   !'SURFACE_TORQUE'   ! (see run star extras)
    ! magnetic field evolution
    extras_cpar(2) = 'CON'  ! CON or DEC for magnetic flux conservation or field decay

/ ! end of star_job namelist



&controls
    use_other_torque = .true.
    use_other_am_mixing = .true.
    use_other_wind = .true.
    mdot_omega_power = 0.0

    redo_limit = -1
    min_dq_for_xa = 1d-5

    newton_iterations_limit = 10

    max_logT_for_k_below_const_q = 100
    max_q_for_k_below_const_q = 0.995
    min_q_for_k_below_const_q = 0.995

    max_logT_for_k_const_mass = 100
```

```
    max_q_for_k_const_mass = 0.99
    min_q_for_k_const_mass = 0.99

    max_model_number = 5000

    fix_eps_grav_transition_to_grid = .true.

    varcontrol_target = 5d-4
    ! extra controls for timestep
    delta_lg_star_mass_limit = 2d-3
    delta_lg_star_mass_hard_limit = 4d-3
    ! these are to properly resolve core hydrogen depletion
    delta_lg_XH_cntr_limit = 0.04d0
    delta_lg_XH_cntr_max = 0.0d0
    delta_lg_XH_cntr_min = -3.0d0
    delta_lg_XH_cntr_hard_limit = 0.06d0
    ! these are to properly resolve core helium depletion
    delta_lg_XHe_cntr_limit = 0.04d0
    delta_lg_XHe_cntr_max = 0.0d0
    delta_lg_XHe_cntr_min = -3.0d0
    delta_lg_XHe_cntr_hard_limit = 0.06d0
    ! avoid large jumps in the HR diagram
    delta_HR_limit = 0.01d0
    ! use a less stric limit for L_He, to speed up things
    ! for the summer school
    delta_lgL_He_limit = 0.05

    photo_directory = 'photos1'
    log_directory = 'LOGS1'

    profile_interval = 50
    history_interval = 1
    terminal_interval = 1
    write_header_frequency = 10

    use_ledoux_criterion = .true.
    mixing_length_alpha = 1.5d0
    alpha_semiconvection = 1d0
    thermohaline_coeff = 1d0

  ! rotational mixing coeffs
    am_nu_factor = 1.0
    am_nu_GSF_factor = 1.0
    am_nu_ST_factor = 0.0
    am_nu_DSI_factor = 1.0
    am_nu_SSI_factor = 1.0
    am_nu_ES_factor = 1.0
    D_visc_factor = 0.0
    am_nu_SH_factor = 1.0
    D_ST_factor = 0.0
    D_SH_factor = 1.0
    D_GSF_factor = 1.0
    D_ES_factor = 1.0
    D_SSI_factor = 1.0
    D_DSI_factor = 1.0
    am_D_mix_factor = 0.333333d0
    am_gradmu_factor = 0.05d0
    num_cells_for_smooth_gradL_composition_term = 2

  ! premix omega to avoid doing the newton with crazily shearing material
    premix_omega = .true.
```

```
  ! wind options
    hot_wind_scheme = 'Dutch'
    cool_wind_RGB_scheme = 'Dutch'
    cool_wind_AGB_scheme = 'Dutch'
    Dutch_wind_lowT_scheme = 'de Jager'
    Dutch_scaling_factor = 0.8

    ! use implicit wind close to critical
    surf_avg_tau_min = 0
    surf_avg_tau = 10
    ! max_mdot_redo_cnt is set to 100 together with rlof
    max_mdot_redo_cnt = 0
    min_years_dt_for_redo_mdot = 0
    surf_w_div_w_crit_limit = 0.98d0
    surf_w_div_w_crit_tol = 0.02d0
    rotational_mdot_boost_fac = 1d10
    rotational_mdot_kh_fac = 1d10
    mdot_revise_factor = 1.2
    implicit_mdot_boost = 0.1

  ! Use type2 opacity tables
    use_Type2_opacities = .true.
    Zbase = 0.00085d0

  ! we use step overshooting
    step_overshoot_f_above_burn_h_core = 0.345
    overshoot_f0_above_burn_h_core = 0.01
    step_overshoot_D0_coeff = 1.0

    max_mdot_jump_for_rotation = 1.1

/ ! end of controls namelist


&pgstar
    pgstar_age_disp = 2.5
    pgstar_model_disp = 2.5

    !### scale for axis labels
    pgstar_xaxis_label_scale = 1.3
    pgstar_left_yaxis_label_scale = 1.3
    pgstar_right_yaxis_label_scale = 1.3

    Grid2_win_flag = .true.

    Grid2_win_width = 15
    Grid2_win_aspect_ratio = 0.65 ! aspect_ratio = height/width
    Grid2_title = 'STAR 1'

    Grid2_plot_name(4) = 'Mixing'

    ! file output
    !Grid2_file_flag = .true.
    Grid2_file_dir = 'png'
    Grid2_file_prefix = 'grid_'
    Grid2_file_interval = 1 ! output when mod(model_number,Grid2_file_interval)==0
    Grid2_file_width = -1 ! negative means use same value as for window
    Grid2_file_aspect_ratio = -1 ! negative means use same value as for window

    Grid2_num_cols = 7 ! divide plotting region into this many equal width cols
    Grid2_num_rows = 8 ! divide plotting region into this many equal height rows
```

```
Grid2_num_plots = 5 ! <= 10

Text_Summary1_txt_scale = 4
Text_Summary1_name(1,1) = 'model_number'
Text_Summary1_name(2,1) = 'log_star_age'
Text_Summary1_name(3,1) = 'log_dt'
Text_Summary1_name(4,1) = 'log_L'
Text_Summary1_name(5,1) = 'log_Teff'
Text_Summary1_name(6,1) = 'log_R'
Text_Summary1_name(7,1) = 'log_g'
Text_Summary1_name(8,1) = 'mass_conv_core'

Text_Summary1_name(1,2) = 'star_mass'
Text_Summary1_name(2,2) = 'log_abs_mdot'
Text_Summary1_name(3,2) = 'he_core_mass'
Text_Summary1_name(4,2) = 'log_cntr_T'
Text_Summary1_name(5,2) = 'log_cntr_Rho'
Text_Summary1_name(6,2) = 'center h1'
Text_Summary1_name(7,2) = 'center he4'
Text_Summary1_name(8,2) = 'surface he4'

Text_Summary1_name(1,3) = 'binary_separation'
Text_Summary1_name(2,3) = 'period_days'
Text_Summary1_name(3,3) = 'star_1_mass'
Text_Summary1_name(4,3) = 'star_2_mass'
Text_Summary1_name(5,3) = 'rl_1'
Text_Summary1_name(6,3) = 'star_1_radius'
Text_Summary1_name(7,3) = 'rl_2'
Text_Summary1_name(8,3) = 'star_2_radius'

Text_Summary1_name(1,4) = 'lg_t_sync_1'
Text_Summary1_name(2,4) = 'lg_t_sync_2'
Text_Summary1_name(3,4) = 'P_rot_div_P_orb_1'
Text_Summary1_name(4,4) = 'P_rot_div_P_orb_2'
Text_Summary1_name(5,4) = ''
Text_Summary1_name(6,4) = 'num_zones'
Text_Summary1_name(7,4) = 'num_retries'
Text_Summary1_name(8,4) = 'num_backups'

Grid2_plot_name(1) = 'HR'
Grid2_plot_row(1) = 1 ! number from 1 at top
Grid2_plot_rowspan(1) = 3 ! plot spans this number of rows
Grid2_plot_col(1) =  1 ! number from 1 at left
Grid2_plot_colspan(1) = 2 ! plot spans this number of columns
Grid2_plot_pad_left(1) = -0.05 ! fraction of full window width for padding on left
Grid2_plot_pad_right(1) = 0.05 ! fraction of full window width for padding on right
Grid2_plot_pad_top(1) = 0.00 ! fraction of full window height for padding at top
Grid2_plot_pad_bot(1) = 0.05 ! fraction of full window height for padding at bottom
Grid2_txt_scale_factor(1) = 0.65 ! multiply txt_scale for subplot by this

Grid2_plot_name(5) = 'Kipp'
Grid2_plot_row(5) = 4 ! number from 1 at top
Grid2_plot_rowspan(5) = 3 ! plot spans this number of rows
Grid2_plot_col(5) =  1 ! number from 1 at left
Grid2_plot_colspan(5) = 2 ! plot spans this number of columns
Grid2_plot_pad_left(5) = -0.05 ! fraction of full window width for padding on left
Grid2_plot_pad_right(5) = 0.05 ! fraction of full window width for padding on right
Grid2_plot_pad_top(5) = 0.03 ! fraction of full window height for padding at top
Grid2_plot_pad_bot(5) = 0.0 ! fraction of full window height for padding at bottom
Grid2_txt_scale_factor(5) = 0.65 ! multiply txt_scale for subplot by this
Kipp_title = ""
Kipp_xaxis_name = "star_age"
```

```
Kipp_xaxis_in_Myr = .true.

Grid2_plot_name(2) = 'Text_Summary1'
Grid2_plot_row(2) = 7 ! number from 1 at top
Grid2_plot_rowspan(2) = 2 ! plot spans this number of rows
Grid2_plot_col(2) = 1 ! number from 1 at left
Grid2_plot_colspan(2) = 4 ! plot spans this number of columns
Grid2_plot_pad_left(2) = -0.08 ! fraction of full window width for padding on left
Grid2_plot_pad_right(2) = -0.10 ! fraction of full window width for padding on right
Grid2_plot_pad_top(2) = 0.08 ! fraction of full window height for padding at top
Grid2_plot_pad_bot(2) = -0.04 ! fraction of full window height for padding at bottom
Grid2_txt_scale_factor(2) = 0.19 ! multiply txt_scale for subplot by this

Grid2_plot_name(3) = 'Profile_Panels3'
Profile_Panels3_title = 'Abundance-Power-Mixing'
Profile_Panels3_num_panels = 3
Profile_Panels3_yaxis_name(1) = 'Abundance'
Profile_Panels3_yaxis_name(2) = 'Power'
Profile_Panels3_yaxis_name(3) = 'Mixing'

Profile_Panels3_xaxis_name = 'mass'
Profile_Panels3_xaxis_reversed = .false.
Profile_Panels3_xmin = -101d0 ! only used if /= -101d0
Profile_Panels3_xmax = -101d0 ! 10 ! only used if /= -101d0

Grid2_plot_row(3) = 1 ! number from 1 at top
Grid2_plot_rowspan(3) = 6 ! plot spans this number of rows
Grid2_plot_col(3) = 3 ! plot spans this number of columns
Grid2_plot_colspan(3) = 3 ! plot spans this number of columns

Grid2_plot_pad_left(3) = 0.04 ! fraction of full window width for padding on left
Grid2_plot_pad_right(3) = 0.08 ! fraction of full window width for padding on right
Grid2_plot_pad_top(3) = 0.0 ! fraction of full window height for padding at top
Grid2_plot_pad_bot(3) = 0.0 ! fraction of full window height for padding at bottom
Grid2_txt_scale_factor(3) = 0.65 ! multiply txt_scale for subplot by this

Grid2_plot_name(4) = 'History_Panels1'
Grid2_plot_row(4) = 1 ! number from 1 at top
Grid2_plot_rowspan(4) = 8 ! plot spans this number of rows
Grid2_plot_col(4) =  6 ! number from 1 at left
Grid2_plot_colspan(4) = 2 ! plot spans this number of columns
Grid2_plot_pad_left(4) = 0.05 ! fraction of full window width for padding on left
Grid2_plot_pad_right(4) = 0.03 ! fraction of full window width for padding on right
Grid2_plot_pad_top(4) = 0.0 ! fraction of full window height for padding at top
Grid2_plot_pad_bot(4) = 0.0 ! fraction of full window height for padding at bottom
Grid2_txt_scale_factor(4) = 0.65 ! multiply txt_scale for subplot by this

History_Panels1_xaxis_name = 'star_1_mass'
History_Panels1_xaxis_reversed = .true.
History_Panels1_yaxis_name(1) = 'lg_mtransfer_rate'
History_Panels1_other_yaxis_name(1) = 'lg_wind_mdot_1'
History_Panels1_yaxis_name(2) = 'period_days'
History_Panels1_other_yaxis_name(2) = ''
History_Panels1_yaxis_name(3) = 'rl_relative_overflow_1'
History_Panels1_other_yaxis_name(3) = 'rl_relative_overflow_2'
History_Panels1_ymin(1) = -9
History_Panels1_ymax(1) = 0
History_Panels1_other_ymin(1) = -9
History_Panels1_other_ymax(1) = 0
History_Panels1_ymin(3) = -1d0
History_Panels1_ymax(3) = 0.4d0
History_Panels1_other_ymin(3) = -1d0
```

```
    History_Panels1_other_ymax(3) = 0.4d0

    Abundance_line_txt_scale_factor = 1.1 ! relative to other text
    Abundance_legend_txt_scale_factor = 1.1 ! relative to other text
    Abundance_legend_max_cnt = 0
    Abundance_log_mass_frac_min = -3.5 ! only used if < 0

    Text_Summary1_name(2,1) = 'star_age'
    Text_Summary1_name(3,1) = 'time_step'

    Grid2_file_flag = .true.
    Grid2_file_dir = 'png'
    Grid2_file_prefix = 'grid_'
    Grid2_file_interval = 50 ! 1 ! output when mod(model_number,Grid2_file_interval)==0
    Grid2_file_width = -1 ! negative means use same value as for window
    Grid2_file_aspect_ratio = -1 ! negative means use same value as for window

/ ! end of pgstar namelist
```

### B.2 inlist2

```
! inlist_test_rlo



&star_job

    show_log_description_at_start = .false.

    new_rotation_flag = .true.
    change_rotation_flag = .true.
    set_initial_surface_rotation_v = .true.
    !set rotational velocity to zero, tides will change this
    new_surface_rotation_v = 0
    num_steps_to_relax_rotation = 30

    !load_saved_model = .true.
    !saved_model_name = 'model2.mod'

    extras_lrpar = 3
    extras_rpar(1) = 9000
    extras_rpar(2) = 1
    extras_rpar(3) = 1

    extras_lcpar = 2
    extras_cpar(1) = 'UNIFORM_TORQUE'
    extras_cpar(2) = 'CON'

    pgstar_flag = .false.
    pause_before_terminate = .true.

    set_initial_age = .true.
    initial_age = 0

    set_initial_model_number = .true.
    initial_model_number = 0


/ ! end of star_job namelist
```

```
&controls
    redo_limit = -1
    min_dq_for_xa = 1d-5

    newton_iterations_limit = 10

    max_logT_for_k_below_const_q = 100
    max_q_for_k_below_const_q = 0.995
    min_q_for_k_below_const_q = 0.995

    max_logT_for_k_const_mass = 100
    max_q_for_k_const_mass = 0.99
    min_q_for_k_const_mass = 0.99

    max_model_number = 5000

    fix_eps_grav_transition_to_grid = .true.

    varcontrol_target = 5d-4
    ! extra controls for timestep
    delta_lg_star_mass_limit = 2d-3
    delta_lg_star_mass_hard_limit = 4d-3
    ! these are to properly resolve core hydrogen depletion
    delta_lg_XH_cntr_limit = 0.04d0
    delta_lg_XH_cntr_max = 0.0d0
    delta_lg_XH_cntr_min = -3.0d0
    delta_lg_XH_cntr_hard_limit = 0.06d0
    ! these are to properly resolve core helium depletion
    delta_lg_XHe_cntr_limit = 0.04d0
    delta_lg_XHe_cntr_max = 0.0d0
    delta_lg_XHe_cntr_min = -3.0d0
    delta_lg_XHe_cntr_hard_limit = 0.06d0
    ! avoid large jumps in the HR diagram
    delta_HR_limit = 0.01d0
    ! use a less stric limit for L_He, to speed up things
    ! for the summer school
    delta_lgL_He_limit = 0.05

    photo_directory = 'photos2'
    log_directory = 'LOGS2'

    profile_interval = 50
    history_interval = 1
    terminal_interval = 200
    write_header_frequency = 10

    use_ledoux_criterion = .true.
    mixing_length_alpha = 1.5d0
    alpha_semiconvection = 1d0
    thermohaline_coeff = 1d0

  ! rotational mixing coeffs
    am_nu_ST_factor = 1.0
    D_visc_factor = 0.0
    am_nu_SH_factor = 0.0
    D_ST_factor = 0.0
    D_SH_factor = 0.0
    D_GSF_factor = 1.0
    D_ES_factor = 1.0
    D_SSI_factor = 1.0
```

```
    D_DSI_factor = 1.0
    am_D_mix_factor = 0.0333333d0
    am_gradmu_factor = 0.1d0
    num_cells_for_smooth_gradL_composition_term = 2

  ! premix omega to avoid doing the newton with crazily shearing material
    premix_omega = .true.

  ! wind options
    hot_wind_scheme = 'Dutch'
    cool_wind_RGB_scheme = 'Dutch'
    cool_wind_AGB_scheme = 'Dutch'
    Dutch_wind_lowT_scheme = 'de Jager'
    Dutch_scaling_factor = 1.0
    cool_wind_full_on_T = 0.8d4
    hot_wind_full_on_T = 1.2d4

    ! use implicit wind close to critical
    surf_avg_tau_min = 0
    surf_avg_tau = 10
    ! max_mdot_redo_cnt is set to 100 together with rlof
    max_mdot_redo_cnt = 0
    min_years_dt_for_redo_mdot = 0
    surf_w_div_w_crit_limit = 0.98d0
    surf_w_div_w_crit_tol = 0.02d0
    rotational_mdot_boost_fac = 1d10
    rotational_mdot_kh_fac = 1d10
    mdot_revise_factor = 1.2
    implicit_mdot_boost = 0.1

  ! Use type2 opacity tables
    use_Type2_opacities = .true.
    Zbase = 0.00085d0

  ! we use step overshooting
    step_overshoot_f_above_burn_h_core = 0.345
    overshoot_f0_above_burn_h_core = 0.01
    step_overshoot_D0_coeff = 1.0

    max_mdot_jump_for_rotation = 1.1

/ ! end of controls namelist


&pgstar
    pgstar_age_disp = 2.5
    pgstar_model_disp = 2.5

    !### scale for axis labels
    pgstar_xaxis_label_scale = 1.3
    pgstar_left_yaxis_label_scale = 1.3
    pgstar_right_yaxis_label_scale = 1.3

    Grid2_win_flag = .true.

    Grid2_win_width = 15
    Grid2_win_aspect_ratio = 0.65 ! aspect_ratio = height/width
    Grid2_title = 'STAR 2'

    Grid2_plot_name(4) = 'Mixing'

    ! file output
```

```
!Grid2_file_flag = .true.
Grid2_file_dir = 'png'
Grid2_file_prefix = 'grid_'
Grid2_file_interval = 1 ! output when mod(model_number,Grid2_file_interval)==0
Grid2_file_width = -1 ! negative means use same value as for window
Grid2_file_aspect_ratio = -1 ! negative means use same value as for window

Grid2_num_cols = 7 ! divide plotting region into this many equal width cols
Grid2_num_rows = 8 ! divide plotting region into this many equal height rows

Grid2_num_plots = 5 ! <= 10

Text_Summary1_txt_scale = 4
Text_Summary1_name(1,1) = 'model_number'
Text_Summary1_name(2,1) = 'log_star_age'
Text_Summary1_name(3,1) = 'log_dt'
Text_Summary1_name(4,1) = 'log_L'
Text_Summary1_name(5,1) = 'log_Teff'
Text_Summary1_name(6,1) = 'log_R'
Text_Summary1_name(7,1) = 'log_g'
Text_Summary1_name(8,1) = 'mass_conv_core'

Text_Summary1_name(1,2) = 'star_mass'
Text_Summary1_name(2,2) = 'log_abs_mdot'
Text_Summary1_name(3,2) = 'he_core_mass'
Text_Summary1_name(4,2) = 'log_cntr_T'
Text_Summary1_name(5,2) = 'log_cntr_Rho'
Text_Summary1_name(6,2) = 'center h1'
Text_Summary1_name(7,2) = 'center he4'
Text_Summary1_name(8,2) = 'surface he4'

Text_Summary1_name(1,3) = 'binary_separation'
Text_Summary1_name(2,3) = 'period_days'
Text_Summary1_name(3,3) = 'star_1_mass'
Text_Summary1_name(4,3) = 'star_2_mass'
Text_Summary1_name(5,3) = 'rl_1'
Text_Summary1_name(6,3) = 'star_1_radius'
Text_Summary1_name(7,3) = 'rl_2'
Text_Summary1_name(8,3) = 'star_2_radius'

Text_Summary1_name(1,4) = 'lg_t_sync_1'
Text_Summary1_name(2,4) = 'lg_t_sync_2'
Text_Summary1_name(3,4) = 'P_rot_div_P_orb_1'
Text_Summary1_name(4,4) = 'P_rot_div_P_orb_2'
Text_Summary1_name(5,4) = ''
Text_Summary1_name(6,4) = 'num_zones'
Text_Summary1_name(7,4) = 'num_retries'
Text_Summary1_name(8,4) = 'num_backups'

Grid2_plot_name(1) = 'HR'
Grid2_plot_row(1) = 1 ! number from 1 at top
Grid2_plot_rowspan(1) = 3 ! plot spans this number of rows
Grid2_plot_col(1) =  1 ! number from 1 at left
Grid2_plot_colspan(1) = 2 ! plot spans this number of columns
Grid2_plot_pad_left(1) = -0.05 ! fraction of full window width for padding on left
Grid2_plot_pad_right(1) = 0.05 ! fraction of full window width for padding on right
Grid2_plot_pad_top(1) = 0.00 ! fraction of full window height for padding at top
Grid2_plot_pad_bot(1) = 0.05 ! fraction of full window height for padding at bottom
Grid2_txt_scale_factor(1) = 0.65 ! multiply txt_scale for subplot by this

Grid2_plot_name(5) = 'Kipp'
Grid2_plot_row(5) = 4 ! number from 1 at top
```

```
Grid2_plot_rowspan(5) = 3 ! plot spans this number of rows
Grid2_plot_col(5) =  1 ! number from 1 at left
Grid2_plot_colspan(5) = 2 ! plot spans this number of columns
Grid2_plot_pad_left(5) = -0.05 ! fraction of full window width for padding on left
Grid2_plot_pad_right(5) = 0.05 ! fraction of full window width for padding on right
Grid2_plot_pad_top(5) = 0.03 ! fraction of full window height for padding at top
Grid2_plot_pad_bot(5) = 0.0 ! fraction of full window height for padding at bottom
Grid2_txt_scale_factor(5) = 0.65 ! multiply txt_scale for subplot by this
Kipp_title = ""
Kipp_xaxis_name = "star_age"
Kipp_xaxis_in_Myr = .true.

Grid2_plot_name(2) = 'Text_Summary1'
Grid2_plot_row(2) = 7 ! number from 1 at top
Grid2_plot_rowspan(2) = 2 ! plot spans this number of rows
Grid2_plot_col(2) = 1 ! number from 1 at left
Grid2_plot_colspan(2) = 4 ! plot spans this number of columns
Grid2_plot_pad_left(2) = -0.08 ! fraction of full window width for padding on left
Grid2_plot_pad_right(2) = -0.10 ! fraction of full window width for padding on right
Grid2_plot_pad_top(2) = 0.08 ! fraction of full window height for padding at top
Grid2_plot_pad_bot(2) = -0.04 ! fraction of full window height for padding at bottom
Grid2_txt_scale_factor(2) = 0.19 ! multiply txt_scale for subplot by this

Grid2_plot_name(3) = 'Profile_Panels3'
Profile_Panels3_title = 'Abundance-Power-Mixing'
Profile_Panels3_num_panels = 3
Profile_Panels3_yaxis_name(1) = 'Abundance'
Profile_Panels3_yaxis_name(2) = 'Power'
Profile_Panels3_yaxis_name(3) = 'Mixing'

Profile_Panels3_xaxis_name = 'mass'
Profile_Panels3_xaxis_reversed = .false.
Profile_Panels3_xmin = -101d0 ! only used if /= -101d0
Profile_Panels3_xmax = -101d0 ! 10 ! only used if /= -101d0

Grid2_plot_row(3) = 1 ! number from 1 at top
Grid2_plot_rowspan(3) = 6 ! plot spans this number of rows
Grid2_plot_col(3) = 3 ! plot spans this number of columns
Grid2_plot_colspan(3) = 3 ! plot spans this number of columns

Grid2_plot_pad_left(3) = 0.04 ! fraction of full window width for padding on left
Grid2_plot_pad_right(3) = 0.08 ! fraction of full window width for padding on right
Grid2_plot_pad_top(3) = 0.0 ! fraction of full window height for padding at top
Grid2_plot_pad_bot(3) = 0.0 ! fraction of full window height for padding at bottom
Grid2_txt_scale_factor(3) = 0.65 ! multiply txt_scale for subplot by this

Grid2_plot_name(4) = 'History_Panels1'
Grid2_plot_row(4) = 1 ! number from 1 at top
Grid2_plot_rowspan(4) = 8 ! plot spans this number of rows
Grid2_plot_col(4) =  6 ! number from 1 at left
Grid2_plot_colspan(4) = 2 ! plot spans this number of columns
Grid2_plot_pad_left(4) = 0.05 ! fraction of full window width for padding on left
Grid2_plot_pad_right(4) = 0.03 ! fraction of full window width for padding on right
Grid2_plot_pad_top(4) = 0.0 ! fraction of full window height for padding at top
Grid2_plot_pad_bot(4) = 0.0 ! fraction of full window height for padding at bottom
Grid2_txt_scale_factor(4) = 0.65 ! multiply txt_scale for subplot by this

History_Panels1_xaxis_name = 'star_1_mass'
History_Panels1_xaxis_reversed = .true.
History_Panels1_yaxis_name(1) = 'lg_mtransfer_rate'
History_Panels1_other_yaxis_name(1) = 'lg_wind_mdot_2'
History_Panels1_yaxis_name(2) = 'period_days'
```

```
    History_Panels1_other_yaxis_name(2) = ''
    History_Panels1_yaxis_name(3) = 'rl_relative_overflow_1'
    History_Panels1_other_yaxis_name(3) = 'rl_relative_overflow_2'
    History_Panels1_ymin(1) = -9
    History_Panels1_ymax(1) = 0
    History_Panels1_other_ymin(1) = -9
    History_Panels1_other_ymax(1) = 0
    History_Panels1_ymin(3) = -1d0
    History_Panels1_ymax(3) = 0.4d0
    History_Panels1_other_ymin(3) = -1d0
    History_Panels1_other_ymax(3) = 0.4d0

    Abundance_line_txt_scale_factor = 1.1 ! relative to other text
    Abundance_legend_txt_scale_factor = 1.1 ! relative to other text
    Abundance_legend_max_cnt = 0
    Abundance_log_mass_frac_min = -3.5 ! only used if < 0

    Text_Summary1_name(2,1) = 'star_age'
    Text_Summary1_name(3,1) = 'time_step'

    Grid2_file_flag = .true.
    Grid2_file_dir = 'png'
    Grid2_file_prefix = 'grid_'
    Grid2_file_interval = 50 ! 1 ! output when mod(model_number,Grid2_file_interval)==0
    Grid2_file_width = -1 ! negative means use same value as for window
    Grid2_file_aspect_ratio = -1 ! negative means use same value as for window
/ ! end of pgstar namelist
```

### B.3 inlist project

```
&binary_job

   inlist_names(1) = 'inlist1'
   inlist_names(2) = 'inlist2'

   evolve_both_stars = .true.

/ ! end of binary_job namelist

&binary_controls

   !Only need to provide initial orbital period!
   m1 = 7.6
   m2 = 3.8
   initial_period_in_days = 1.2

   do_tidal_circ = .true.
   do_tidal_sync = .true.
   do_j_accretion = .true.
   ! be 100% sure MB is always off
   do_jdot_mb = .false.
   do_jdot_missing_wind = .true.

   !mdot_scheme = "contact"

   ! timestep controls
   fr = 0.05
   fr_limit = 1d-2
   fr_dt_limit = 3000d0
   fm = 0.5
```

```
   fm_limit = 1d-1
   fj = 0.005
   dt_softening_factor = 0.4

   limit_retention_by_mdot_edd = .false.
   implicit_scheme_tolerance = 1d-3
   max_tries_to_achieve = 400
   min_change_factor = 1.025
   max_change_factor = 1.5d0
   initial_change_factor = 1.2d0
   change_factor_fraction = 0.8d0
   implicit_lambda = 0.4d0

   sync_mode_1 = "Uniform"
   sync_type_1 = "Hut_rad"
   Ftid_1 = 1
   sync_mode_2 = "Uniform"
   sync_type_2 = "Hut_rad"
   Ftid_2 = 1
   do_initial_orbit_sync_1 = .false.
   do_initial_orbit_sync_2 = .true.

   roche_min_mdot = 1d-10
   accretor_overflow_terminate = 100.0d0
   terminate_if_L2_overflow = .true.

/ ! end of binary_controls namelist
```